SILVACO
INTERNATIONAL

ATLAS User's Manual

DEVICE SIMULATION SOFTWARE

SILVACO International
4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 95054
Telephone (408) 567-1000
Internet: www.silvaco.com

December 7, 2006

ATLAS
User's Manual

SILVACO International.
4701 Patrick Henry Drive, Building #6
Santa Clara, CA 95054

Phone: (408) 567-1000
Web: www.silvaco.com

Notice

The information contained in this document is subject to change without notice.

SILVACO INTERNATIONAL MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

SILVACO INTERNATIONAL shall not be held liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

VIRTUAL WAFER FAB, VWF MANUFACTURING TOOLS, VWF AUTOMATION TOOLS, VWF INTERACTIVE TOOLS DECKBUILD, TONYPLOT, DEVEDIT, TONYPLOT3D, DEVEDIT3D, MASKVIEWS, ATHENA, SSUPREM4, MC IMPLANT, OPTOLITH, ELITE, MC DEPO/ETCH, SSUPREM3, SPDB, ATLAS, S-PISCES, BLAZE, GIGA2D/3D, MIXEDMODE2D/3D, SIC, FERRO, QUANTUM2D/3D, LUMINOUS2D/3D, LED, VCSELS, LASER, TFT2D/3D, OTFT, OLED, NOISE, DEVICE3D, THERMAL3D, ATLAS INTERPRETER, MERCURY, FASTBLAZE, FASTNOISE, FASTGIGA, FAST ATLAS C-INTERPRETER, MOCASIM, VICTORY, HARM, ZENITH, VISION, MIXSIM, TCAD DRIVEN CAD, SIMULATION STANDARD, CONNECTING TCAD TO TAPEOUT, AND TCAD OMNI are trademarks of Silvaco International.

All other trademarks mentioned in this manual are the property of their respective owners.

Style Conventions		
Font Style/Convention	**Description**	**Example**
•	This represents a list of items or terms.	• Bullet A • Bullet B • Bullet C
1. 2. 3.	This represents a set of directions to perform an action.	To open a door: 1. Unlock the door by inserting the key into keyhole. 2. Turn key counter-clockwise. 3. Pull out the key from the keyhole. 4. Grab the doorknob and turn clockwise and pull.
→	This represents a sequence of menu options and GUI buttons to perform an action.	**File→Open**
`Courier`	This represents the commands, parameters, and variables syntax.	`HAPPY BIRTHDAY`
New Century Schoolbook Bold	This represents the menu options and buttons in the GUI.	**File**
New Century Schoolbook Italics	This represents the equations.	*abc=xyz*
Note:	This represents the additional important information.	**Note:** Make sure you save often while running an experiment.
NEW CENTURY SCHOOLBOOK IN SMALL CAPS	This represents the names of the Silvaco Products.	ATHENA and ATLAS, .

Table of Contents

Chapter 1: Introduction

1.1: ATLAS Overview

ATLAS provides general capabilities for physically-based two (2D) and three-dimensional (3D) simulation of semiconductor devices. If you're new to ATLAS, read this chapter and Chapter 2: "Getting Started with ATLAS" to understand how ATLAS works. Once you've read these chapters, you can refer to the remaining chapters for a detailed understanding of the capabilities of each ATLAS product.

Those who have used earlier versions of ATLAS may find it helpful to review the updated version history in Appendix D: "ATLAS Version History".

ATLAS is designed to be used with the VWF INTERACTIVE TOOLS. The VWF INTERACTIVE TOOLS are DECKBUILD, TONYPLOT, DEVEDIT, MASKVIEWS, and OPTIMIZER. See their respective manuals on how to use these manuals. See Section 1.3: "Using ATLAS With Other Silvaco Software" for more information about using ATLAS with other Silvaco tools.

ATLAS is supplied with numerous examples that can be accessed through DECKBUILD. These examples demonstrate most of ATLAS's capabilities. The input files that are provided with the examples are an excellent starting point for developing your own input files. To find out how to access the example, see Chapter 2: "Getting Started with ATLAS", Section 2.4: "Accessing The Examples".

1.2: Features And Capabilities of ATLAS

1.2.1: Comprehensive Set of Models

ATLAS provides a comprehensive set of physical models, including:

- DC, AC small-signal, and full time-dependency.
- Drift-diffusion transport models.
- Energy balance and Hydrodynamic transport models.
- Lattice heating and heatsinks.
- Graded and abrupt heterojunctions.
- Optoelectronic interactions with general ray tracing.
- Amorphous and polycrystalline materials.
- General circuit environments.
- Stimulated emission and radiation
- Fermi-Dirac and Boltzmann statistics.
- Advanced mobility models.
- Heavy doping effects.
- Full acceptor and donor trap dynamics
- Ohmic, Schottky, and insulating contacts.
- SRH, radiative, Auger, and surface recombination.
- Impact ionization (local and non-local).
- Floating gates.
- Band-to-band and Fowler-Nordheim tunneling.
- Hot carrier injection.
- Quantum transport models
- Thermionic emission currents.

1.2.2: Fully Integrated Capabilities

ATLAS works well with other software from SILVACO. For example, ATLAS

- Runs in the DECKBUILD interactive run-time environment.
- Is interfaced to TONYPLOT, the interactive graphics and analysis package.
- Accepts input from the ATHENA and SSUPREM3 process simulators.
- Is interfaced to UTMOST parameter extraction and device modeling software.
- can be used in experiments with the VWF AUTOMATION TOOLS.

1.2.3: Sophisticated Numerical Implementation

ATLAS uses powerful numerical techniques, including:

- Accurate and robust discretization techniques.
- Gummel, Newton, and block-Newton nonlinear iteration strategies.
- Efficient solvers, both direct and iterative, for linear subproblems.
- Powerful initial guess strategies.
- Small-signal calculation techniques that converge at all frequencies.
- Stable and accurate time integration.

1.3: Using ATLAS With Other Silvaco Software

ATLAS is best used with the VWF INTERACTIVE TOOLS. These include DECKBUILD, TONYPLOT, DEVEDIT, MASKVIEWS, and OPTIMIZER. DECKBUILD provides an interactive run time environment. TONYPLOT supplies scientific visualization capabilities. DEVEDIT is an interactive tool for structure and mesh specification and refinement. MASKVIEWS is an IC Layout Editor. The OPTIMIZER supports black box optimization across multiple simulators.

ATLAS, however, is often used with the ATHENA process simulator. ATHENA predicts the physical structures that result from processing steps. The resulting physical structures are used as input by ATLAS, which then predicts the electrical characteristics associated with specified bias conditions. The combination of ATHENA and ATLAS makes it possible to determine the impact of process parameters on device characteristics.

The electrical characteristics predicted by ATLAS can be used as input by the UTMOST device characterization and SPICE modeling software. Compact models based on simulated device characteristics can then be supplied to circuit designers for preliminary circuit design. Combining ATHENA, ATLAS, UTMOST, and SMARTSPICE makes it possible to predict the impact of process parameters on circuit characteristics.

ATLAS can also be used as one of the simulators within the VWF AUTOMATION TOOLS. VWF makes it convenient to perform highly automated simulation-based experimentation. VWF is used in a way that reflects experimental research and development procedures using split lots. It therefore links simulation very closely to technology development, resulting in significantly increased benefits from simulation use.

1.4: The Nature Of Physically-Based Simulation

ATLAS is a physically-based device simulator. Physically-based device simulation is not a familiar concept for all engineers. This section will briefly describe this type of simulation.

Physically-based device simulators predict the electrical characteristics that are associated with specified physical structures and bias conditions. This is achieved by approximating the operation of a device onto a two or three dimensional grid, consisting of a number of grid points called nodes. By applying a set of differential equations, derived from Maxwell's laws, onto this grid you can simulate the transport of carriers through a structure. This means that the electrical performance of a device can now be modeled in DC, AC or transient modes of operation.

There are three physically-based simulation. These are:

- It is predictive.
- It provides insight.
- It conveniently captures and visualizes theoretical knowledge.

Physically-based simulation is different from empirical modeling. The goal of empirical modeling is to obtain analytic formulae that approximate existing data with good accuracy and minimum complexity. Empirical models provide efficient approximation and interpolation. They do not provide insight, or predictive capabilities, or encapsulation of theoretical knowledge.

Physically-based simulation has become very important for two reasons. One, it is almost always much quicker and cheaper than performing experiments. Two, it provides information that is difficult or impossible to measure.

The drawbacks of physically-based simulation are that all the relevant physics must be incorporated into a simulator. Also, numerical procedures must be implemented to solve the associated equations. These tasks have been taken care of for ATLAS users.

Those who use physically-based device simulation tools must specify the problem to be simulated. In ATLAS, specify device simulation problems by defining:

- The physical structure to be simulated.
- The physical models to be used.
- The bias conditions for which electrical characteristics are to be simulated.

The subsequent chapters of this manual describe how to perform these steps.

Chapter 2:
Getting Started with ATLAS

2.1: Overview

ATLAS is a physically-based two and three dimensional device simulator. It predicts the electrical behavior of specified semiconductor structures and provides insight into the internal physical mechanisms associated with device operation.

ATLAS can be used standalone or as a core tool in Silvaco's VIRTUAL WAFER FAB simulation environment. In the sequence of predicting the impact of process variables on circuit performance, device simulation fits between process simulation and SPICE model extraction.

This chapter will show you how to use ATLAS effectively. It is a source of useful hints and advice. The organization of topics parallels the steps that you go through to run the program. If you have used earlier versions of ATLAS, you will still find this chapter useful because of the new version.

This chapter concentrates on the core functionality of ATLAS. If you're primarily interested in the specialized capabilities of a particular ATLAS tool, read this chapter first. Then, read the chapters that describe the ATLAS tools you wish to use.

2.2: ATLAS Inputs and Outputs

Figure 2-1 shows the types of information that flow in and out of ATLAS. Most ATLAS simulations use two input files. The first input file is a text file that contains commands for ATLAS to execute. The second input file is a structure file that defines the structure that will be simulated.

ATLAS produces three types of output files. The first type of output file is the run-time output, which gives you the progress and the error and warning messages as the simulation proceeds. The second type of output file is the log file, which stores all terminal voltages and currents from the device analysis. The third type of output file is the solution file, which stores 2D and 3D data relating to the values of solution variables within the device at a given bias point.

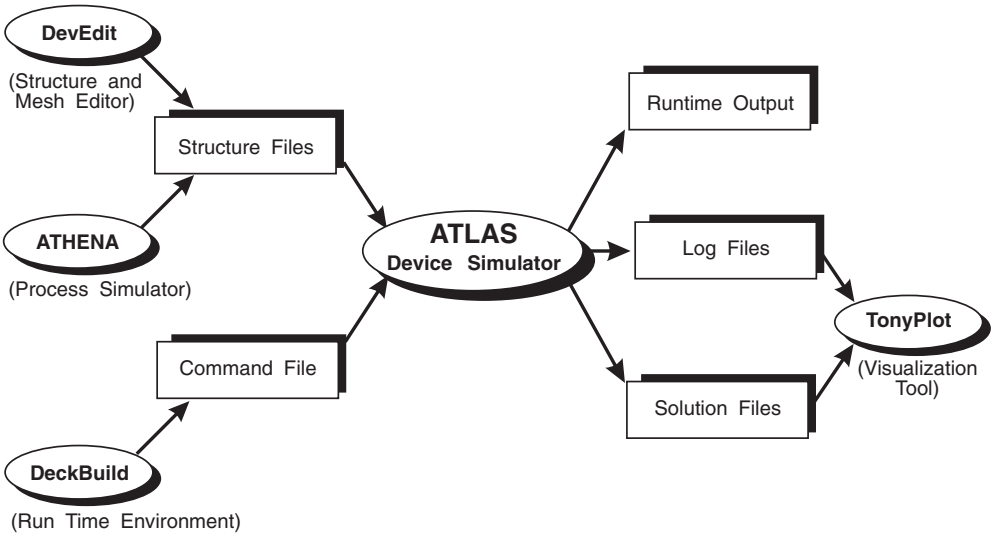


Figure 2-1: ATLAS Inputs and Outputs

2.3: Modes of Operation

ATLAS is normally used in conjunction with the DECKBUILD run-time environment, which supports both interactive and batch mode operation. We strongly recommend that you always run ATLAS within DECKBUILD. In this section, we present the basic information you need to run ATLAS in DECKBUILD. The DECKBUILD USER'S MANUAL provides a more detailed description of the features and capabilities of DECKBUILD.

2.3.1: Interactive Mode With DeckBuild

To start ATLAS in DECKBUILD, type:

```
deckbuild -as
```

at the UNIX system command prompt. The command line option, `-as`, instructs DECKBUILD to start ATLAS as the default simulator.

If you want to start from an existing input file, start DECKBUILD by typing:

```
deckbuild -as  <input filename>
```

The run-time output shows the execution of each ATLAS command and includes error messages, warnings, extracted parameters, and other important output for evaluating each ATLAS run. When ATLAS runs in this mode, the run-time output is sent to the output section of the DeckBuild Window and can be saved as needed. Therefore, you don't need to save the run-time output explicitly. The following command line, however, specifies the name of a file that will be used for storing the run-time output.

```
deckbuild -as  <input filename>  -outfile  <output filename>
```

In this case, the run-time output is sent to the output file and to the output section of the DeckBuild Window.

2.3.2: Batch Mode With DeckBuild

To use DECKBUILD in a non-interactive or batch mode, add the `-run` parameter to the command that invokes DECKBUILD. A prepared command file is required for running in batch mode. We advise you to save the run-time output to a file, since error messages in the run-time output would otherwise be lost when the batch job completes. For example:

```
deckbuild -run -as  <input filename>  -outfile  <output filename>
```

Using this command requires a local X-Windows system to be running. The job runs inside a DECKBUILD icon on the terminal and quits automatically when the ATLAS simulation is complete. You can also run DECKBUILD using a remote display. For example:

```
deckbuild -run -as <input file> -outfile <output file> -display<hostname>:0.0
```

2.3.3: No Windows Batch Mode With DeckBuild

For completely non-X Windows operation of DECKBUILD, use the `-ascii` parameter. For example:

```
deckbuild -run -ascii -as <input filename> -outfile <output filename>
```

This command directs DECKBUILD to run the ATLAS simulation without the DeckBuild Window or icon. This is useful for remote execution without an X Windows emulator or for replacing UNIX-based ATLAS runs within framework programs.

When using batch mode, use the UNIX command suffix, `&`, to detach the job from the current command shell. To run a remote ATLAS simulation under DECKBUILD without display and then logout from the system, use the UNIX command, `nohup` , before the DECKBUILD command line. For example:

```
nohup deckbuild -run -ascii -as <input filename> -outfile <output filename> &
```

2.3.4: Running ATLAS inside Deckbuild

Each ATLAS run inside DECKBUILD should start with the line:

```
go atlas
```

A single input file may contain several ATLAS runs each separated with a `go atlas` line. Input files within DECKBUILD may also contain runs from other programs such as ATHENA or DEVEDIT along with the ATLAS runs.

Running a given version number of ATLAS

The `go` statement can be modified to provide parameters for the ATLAS run. To run version 4.3.0.R, the syntax is:

```
go atlas simflags="-V 4.3.0.R"
```

Starting Parallel ATLAS

The `-P` option is used to set the number of processors to use in a parallel ATLAS run. If the number set by `-P` is greater than the number of processors available or than the number of parallel thread licenses, the number is automatically reduced to this cap number. To run on 4 processors, use:

```
go atlas simflags="-V 4.3.2.C -P 4"
```

2.3.5: Batch Mode Without DeckBuild

You can run ATLAS outside the DECKBUILD environment. But this isn't recommended by Silvaco. If you don't want the overhead of the DeckBuild Window, use the No Windows Mode. Many important features such as variable substitution, automatic interfacing to process simulation, and parameter extraction are unavailable outside the DECKBUILD environment. To run ATLAS directly under UNIX, use the command:

```
atlas <input filename>
```

To save the run-time output to a file, don't use the UNIX redirect command (>). Simply specify the name of the output file. For example:

```
atlas <input filename> -logfile  <output filename>
```

Note: The standard examples supplied with ATLAS will not run correctly outside of DECKBUILD.

2.3.6: TMA Compatibility Mode

You can add the `-TMA` command line flag to the `atlas` command to direct the ATLAS simulator to operate in TMA Compatibility Mode. In this mode, the default material models and parameters are altered to closely agree with those of TMA'S Medici© simulator.

2.4: Accessing The Examples

ATLAS has more than 300 standard examples that demonstrate how the program is used to simulate different technologies. These examples are a good starting point for creating your own simulations. The examples are accessed from the menu system in DECKBUILD. To select and load an example:

1. Start DECKBUILD with ATLAS as the simulator, which is described in the previous section.
2. Use left mouse button to pull down the **Main Control** menu.
3. Select **Examples**. An index will then appear in a Deckbuild Examples Window (see Figure 2-2).

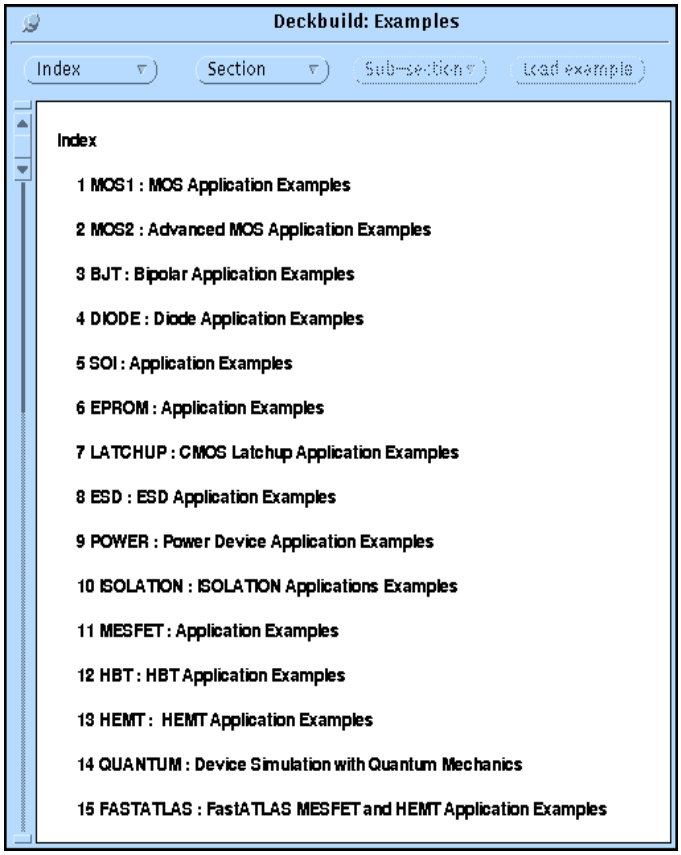


Figure 2-2: Examples Index in DeckBuild

The examples are divided by technology or technology group. For instance, the most common technologies are individually listed (e.g., MOS are under BJT), while others are grouped with similar devices (e.g., IGBT and LDMOS are under POWER, and solar cell and photodiode are under OPTOELECTRONICS).

4. Choose the technology by double clicking the left mouse button over that item. A list of examples for that technology will appear. These examples typically illustrate different devices, applications, or types of simulation.

   You can also search for an example by selecting the **Index** button. Wildcards can be used in the search.

5. Choose a particular example by double clicking the left mouse button over that item in the list. A text description of the example will appear in the window. This text describes the important physical mechanisms in the simulation and the details of the ATLAS syntax used. You should read this information before proceeding.

6. Press the **Load Example** button. The input command file for the example will be copied into your current working directory together with any associated files. A copy of the command file will be loaded into DECKBUILD. Note that the **Load Example** button remains faded out until this step is performed correctly.
7. Press the **Run** button in the middle frame of the DECKBUILD application window to run the example. Alternatively, most examples are supplied with results that are copied into the current working directory along with the input file. To view the results, select (highlight) the name of the results file and select **Tools-Plot**. See the TONYPLOT USER'S MANUAL for details on using TONYPLOT.

2.5: The ATLAS Syntax

An ATLAS command file is a list of commands for ATLAS to execute. This list is stored as an ASCII text file that can be prepared in DECKBUILD or in any text editor. Preparing an input file in DECKBUILD is preferred. You can create an input file by using the DeckBuild Commands menu in the DeckBuild Window.

2.5.1: Statements and Parameters

The input file contains a sequence of statements. Each statement consists of a keyword that identifies the statement and a set of parameters. The general format is:

```
<STATEMENT>    <PARAMETER>=<VALUE>
```

With a few exceptions, the input syntax is not case sensitive. One important exception is that commands described in this manual as being executed by DECKBUILD rather than ATLAS are case sensitive. These include `EXTRACT`, `SET`, `GO`, and `SYSTEM`. Also, filenames for input and output under UNIX are case sensitive.

For any `<STATEMENT>`, ATLAS may have four different types for the `<VALUE> parameter`. These are: Real, Integer, Character, and Logical.

An example of a statement line is:

```
DOPING UNIFORM N.TYPE CONCENTRATION=1.0e16 REGION=1 OUTFILE=my.dop
```

The statement is `DOPING`. All other items are parameters of the `DOPING` statement. `UNIFORM` and `N.TYPE` are logical parameters. Their presence on the line sets their values to `true`. Otherwise, they take their default values (usually `false`). `CONCENTRATION` is a Real parameter and takes floating point numbers as input values. `REGION` is an Integer parameter taking only integer numbers as input. `OUTFILE` is a Character parameter type taking strings as input.

The statement keyword must come first but the order of parameters within a statement is unimportant.

You only need to use enough letters of any parameter to distinguish it from any other parameter on the same statement. Thus, `CONCENTRATION` can be shortened to `CONC`. `REGION`. It can't be shortened to `R`, however, since there's a parameter called `RATIO` associated with the `DOPING` statement.

You can set logical parameters to `false` by preceding them with the ^ symbol. Any line beginning with a # is ignored. These lines are used as comments.

ATLAS can read up to 256 characters on one line. But it is better to spread long input statements over several lines to make the input file more readable. The \ character at the end of a line indicates continuation.

For more information about statements and parameters in ATLAS, see Chapter 19: "Statements".

2.5.2: The Order of ATLAS Commands

The order in which statements occur in an ATLAS input file is important. There are five groups of statements that must occur in the correct order (see Figure 2-3). Otherwise, an error message will appear, which may cause incorrect operation or termination of the program. For example, if the material parameters or models are set in the wrong order, then they may not be used in the calculations.

The order of statements within the mesh definition, structural definition, and solution groups is also important. Otherwise, it may also cause incorrect operation or termination of the program.

.

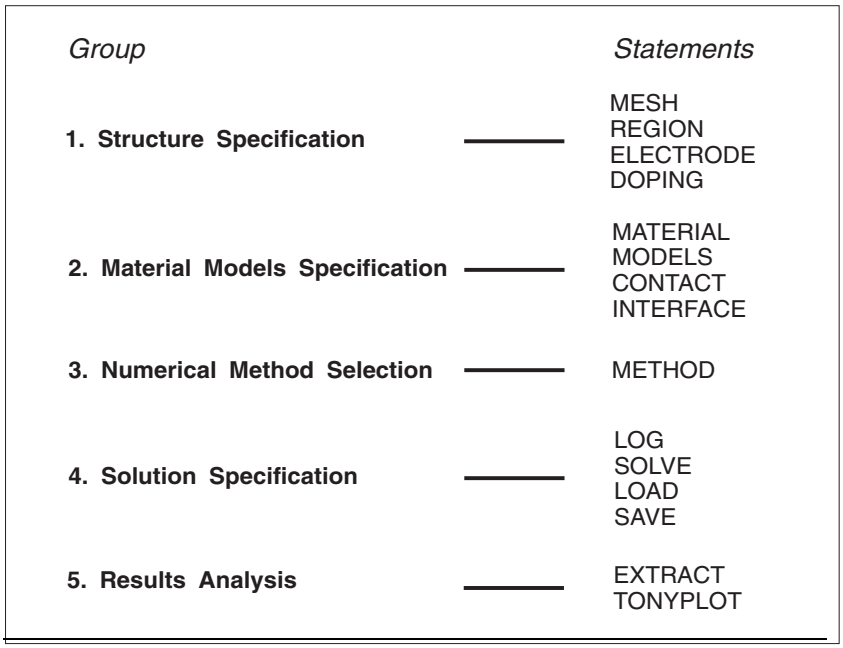


Figure 2-3: ATLAS Command Groups with the Primary Statements in each Group

2.5.3: The DeckBuild Command Menu

The DeckBuild Command Menu (Command Menu) can help you to create input files. This menu is found under the **Commands** button on DECKBUILD's main screen. The Commands Menu is configured for ATLAS whenever ATLAS is the currently active simulator in DECKBUILD. When ATLAS is active, which is indicated in the lower bar of the DeckBuild Window, an ATLAS command prompt will appear in the DECKBUILD output section.

The Command Menu gives you access to pop-up windows where you type information. When you select the **Write** button, syntactically correct statements are written to the DECKBUILD text edit region. The DeckBuild Command Menu does not support all possible ATLAS syntax, but aims to cover the most commonly used commands.

2.5.4: PISCES-II Quick Start

This section is a quickstart for those who may be familiar with the syntax and use of the Stanford University PISCES-II program or other device simulators derived from this program.

The major differences between ATLAS and PISCES-II are:

- all graphics are handled by a separate interactive graphics program, TONYPLOT. The PISCES-II graphics commands, such as `PLOT.1D`, `PLOT.2D`, `CONTOUR`, `VECTOR` are optional. By using TONYPLOT, you no longer need to run the device simulator simply to plot or alter graphics.
- no need to separate individual ATLAS simulations into separate input files. Multiple runs of ATLAS are possible in the same input file separated by the line `go atlas`. There's also no need to separate process and device simulation runs of Silvaco products into separate input files. A single file containing ATHENA and ATLAS syntax is permitted in DECKBUILD.
- the interface from process to device simulation is handled though a single file format compatible with other programs. The file read by ATLAS is the default output file format of ATHENA. No special file format for the interface is required.

- when defining a grid structure within ATLAS, the NODE and LOCATION syntax to define exact grid line numbers in X and Y is not recommended. A more reliable and easier to use syntax using LOCATION and SPACING is available.
- using the REGRID command is not recommended due to the creation of obtuse triangles. A standalone program, such as DEVEDIT, can be used as a grid pre-processor for ATLAS.
- all numerical method selection commands and parameters are on the METHOD statement. The SYMBOLIC statement is not used. Historically, SYMBOLIC and METHOD were used as a coupled pair of statements, but it is more convenient to use a single statement (METHOD) instead. Most of the old parameters of the SYMBOLIC statement have the same meaning and names, despite this move to a single statement. One notable change in ATLAS is that you can combine numerical methods together.

  See the “Pisces-II Compatibility” section on page 2-41 for more information concerning the translation of PISCES-II numerics statements.

- various general purpose commands are actually part of the DECKBUILD user environment. These include SET, EXTRACT, GO, SYSTEM, and SOURCE. These commands can be interspersed inside ATLAS syntax.
- Variable substitution is supported for both numerical and string variables using the SET statement and the $ symbol. To avoid confusion, the # symbol is preferred over the $ symbol for comment statements.

In addition to these changes, the physical models are generally different in ATLAS. Most of the original PISCES-II models have been preserved but often are not the default or the recommended models to use. See the on-line examples for technology specific information about models.

2.6: Defining A Structure

There are three ways to define a device structure in ATLAS.

The first way is to read an existing structure from a file. The structure is created either by an earlier ATLAS run or another program such as ATHENA or DEVEDIT. A MESH statement loads in the mesh, geometry, electrode positions, and doping of the structure. For example:

```
MESH INFILE=<filename>
```

The second way is to use the **Automatic Interface** feature from DECKBUILD to transfer the input structure from ATHENA or DEVEDIT.

The third way is create a structure by using the ATLAS command language. See Chapter 19: "Statements" for more information about the ATLAS syntax.

2.6.1: Interface From ATHENA

When ATHENA and ATLAS are run under DECKBUILD, you can take advantage of an automatic interface between the two programs. Perform the following steps to load the complete mesh, geometry, and doping from ATHENA to ATLAS.

1. Deposit and pattern electrode material in ATHENA.
2. Use the ELECTRODE statement in ATHENA to define contact positions. Specify the x and y coordinates as cross-hairs to pin-point a region. The whole region is then turned into electrode. In many cases, only the x coordinate is needed. For example:

   ```
   ELECTRODE NAME=gate X=1.3 [Y=-0.1])
   ```

   There is a special case to specify a contact on the bottom of the structure. For example:

   ```
   ELECTRODE NAME=substrate BACKSIDE
   ```

3. Save a structure file while ATHENA is still the active simulator. For example:

   ```
   STRUCTURE OUTF=nmos.str
   ```

4. Start ATLAS with the `go atlas` command written in the same input deck. This will automatically load the most recent structure from ATHENA into ATLAS.

If you need to load the structure saved in step 4 into ATLAS without using the auto-interface capability, use the MESH command. For example:

```
MESH INF=nmos.str
```

ATLAS inherits the grid used most recently by ATHENA. With a careful choice of initial mesh or by using the grid manipulation techniques in ATHENA, you can produce a final mesh from ATHENA that will give good results in ATLAS. But, a grid that is appropriate for process simulation isn't always appropriate for device simulation. If the final ATHENA mesh is inappropriate for ATLAS, use either DEVEDIT to re-mesh the structure or the REGRID command.

Note: There's no need to specify a MESH command in ATLAS when using the Automatic Interface of Deckbuild.

2.6.2: Interface From DevEdit

A 2D or 3D structure created by DEVEDIT can be read into ATLAS using the following statement.

```
MESH INF=<structure filename>
```

This statement loads in the mesh, geometry, electrode positions, and doping of the structure. ATLAS will automatically determine whether the mesh is 2D for S-PISCES or BLAZE, or 3D for DEVICE3D or BLAZE3D.

If the structure coming from DEVEDIT were originally created by ATHENA, then define the electrodes in ATHENA as described in the previous section. If the structure is created in DEVEDIT, the electrode regions should be defined in the **Region/Add** menu of DEVEDIT.

2.6.3: Using The Command Language To Define A Structure

To define a device through the ATLAS command language, you must first define a mesh. This mesh or grid covers the physical simulation domain. The mesh is defined by a series of horizontal and vertical lines and the spacing between them. Then, regions within this mesh are allocated to different materials as required to construct the device. For example, the specification of a MOS device requires the specification of silicon and silicon dioxide regions. After the regions are defined, the location of electrodes is specified. The final step is to specify the doping in each region.

When using the command language to define a structure, the information described in the following four sub-sections must be specified in the order listed.

Specifying The Initial Mesh

The first statement must be:

```
MESH SPACE.MULT=<VALUE>
```

This is followed by a series of X.MESH and Y.MESH statements.

```
X.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
Y.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
```

The SPACE.MULT parameter value is used as a scaling factor for the mesh created by the X.MESH and Y.MESH statements. The default value is 1. Values greater than 1 will create a globally coarser mesh for fast simulation. Values less than 1 will create a globally finer mesh for increased accuracy. The X.MESH and Y.MESH statements are used to specify the locations in microns of vertical and horizontal lines, respectively, together with the vertical or horizontal spacing associated with that line. You must specify at least two mesh lines for each direction. ATLAS automatically inserts any new lines required to allow for gradual transitions in the spacing values between any adjacent lines. The X.MESH and Y.MESH statements must be listed in the order of increasing x and y. Both negative and positive values of x and y are allowed.

Figure 2-4 illustrates how these statements work. On the left hand plot, note how the spacing of the vertical lines varies from 1 μm at x=0 and x=10 μm to 0.5 μm at x=5 μm. On the right hand plot, note how specifying the SPACE.MULT parameter to have a value of 0.5 has doubled the density of the mesh in both the x and y directions.

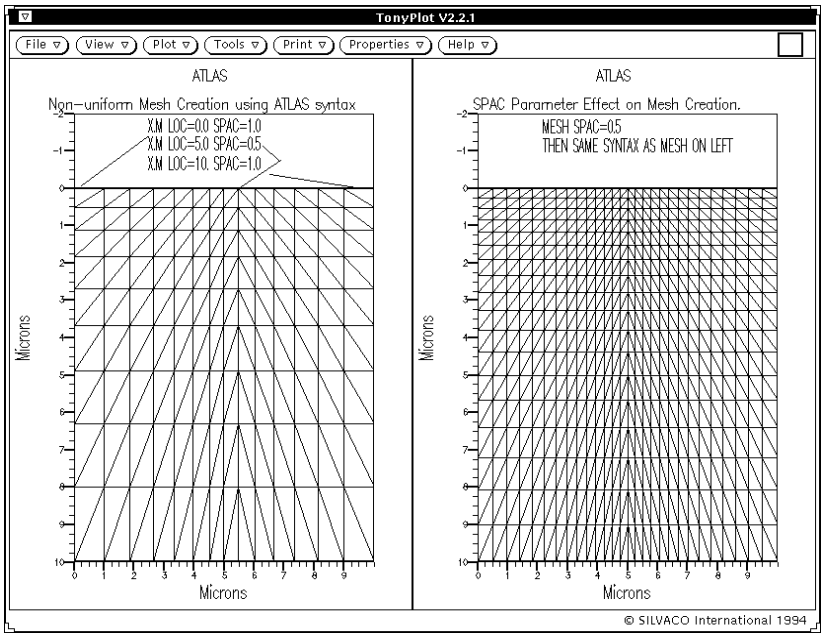


Figure 2-4: Non-uniform Mesh Creation using ATLAS Syntax

After an initial mesh has been defined, you can remove grid lines in specified regions. This is typically done in regions of the device where a coarse grid is expected to be sufficient such as the substrate. The removal of grid lines is accomplished using the ELIMINATE statement. The ELIMINATE statement removes every second mesh line in the specified direction from within a specified rectangle. For example, the statement:

```
ELIMINATE COLUMNS X.MIN=0 X.MAX=4 Y.MIN=0.0 Y.MAX=3
```

removes every second vertical grid line within the rectangle bounded by x=0, x=4, y=0 and y=3 microns.

Specifying Regions And Materials

Once the mesh is specified, every part of it must be assigned a material type. This is done with REGION statements. For example:

```
REGION number=<integer> <material_type> <position parameters>
```

Region numbers must start at 1 and are increased for each subsequent region statement. You can have up to 200 different regions in ATLAS. A large number of materials is available. If a composition-dependent material type is defined, the x and y composition fractions can also be specified in the REGION statement.

The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. If the position parameters of a new statement overlap those of a previous REGION statement, the overlapped area is assigned as the material type of the new region. Make sure that materials are assigned to all mesh points in the structure. If this isn't done, error messages will appear and ATLAS won't run successfully.

You can use the MATERIAL statement to specify the material properties of the defined regions. But you must complete the entire mesh and doping definition before any MATERIAL statements can be used. The specification of material properties is described in Section 2.7.2: "Specifying Material Properties".

Cylindrical Coordinates

Cylindrical coordinates are often used when simulating discrete power devices. In this mode, ATLAS operates with x=0 as the axis of symmetry around which the cylindrical geometry is placed. Many of the default units change when cylindrical coordinates are used. The calculated current is in Amps rather than the usual Amps per micron. External elements are specified in absolute units (e.g., Farads, not Farads/micron for capacitors).

The MESH statement must be used to specify cylindrical symmetry. The following statement creates a mesh, which contains cylindrical symmetry.

```
MESH NX=20 NY=20 CYLINDRICAL
```

There are 20 mesh nodes along the x-axis and 20 mesh nodes along the y-axis.

The following statement imports a mesh, which contains cylindrical symmetry.

```
MESH INF=mesh0.str CYLINDRICAL
```

Note: The CYLINDRICAL parameter setting isn't stored in mesh files. Therefore, this parameter must be specified each time a mesh file, which contains cylindrical symmetry, is loaded.

Specifying Electrodes

Once you have specified the regions and materials, define at least one electrode that contacts a semiconductor material. This is done with the ELECTRODE statement. For example:

```
ELECTRODE NAME=<electrode name> <position_parameters>
```

You can specify up to 50 electrodes. The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. Multiple electrode statements may have the same electrode name. Nodes that are associated with the same electrode name are treated as being electrically connected.

Some shortcuts can be used when defining the location of an electrode. If no y coordinate parameters are specified, the electrode is assumed to be located on the top of the structure. You also can use the RIGHT, LEFT, TOP, and BOTTOM parameters to define the location. For example:

```
ELECTRODE NAME=SOURCE LEFT LENGTH=0.5
```

specifies the source electrode starts at the top left corner of the structure and extends to the right for the distance LENGTH.

Specifying Doping

You can specify analytical doping distributions or have ATLAS read in profiles that come from either process simulation or experiment. You specify the doping using the DOPING statement. For example:

```
DOPING <distribution_type> <dopant_type> <position_parameters>
```

Analytical Doping Profiles

Analytical doping profiles can have uniform or Gaussian forms. The parameters defining the analytical distribution are specified in the DOPING statement. Two examples are shown below with their combined effect shown in Figure 2-5.

```
DOPING UNIFORM CONCENTRATION=1E16 N.TYPE REGION=1

DOPING GAUSSIAN CONCENTRATION=1E18 CHARACTERISTIC=0.05 P.TYPE \
       X.LEFT=0.0 X.RIGHT=1.0 PEAK=0.1
```

The first DOPING statement specifies a uniform n-type doping density of 10^{16} cm^{-3} in the region that was previously labelled as region #1. The position parameters X.MIN, X.MAX, Y.MIN, and Y.MAX can be used instead of a region number.

The second DOPING statement specifies a p-type Gaussian profile with a peak concentration of 10^{18} cm^{-3}. This statement specifies that the peak doping is located along a line from x = 0 to x = 1 microns. Perpendicular to the peak line, the doping drops off according to a Gaussian distribution with a standard deviation of $(0.05\sqrt{2})$ µm. At x < 0 or x > 1, the doping drops off laterally with a default standard deviation that is $(70\sqrt{2})$% of CHARACTERISTIC. This lateral roll-off can be altered with the RATIO.LATERAL parameter. If a Gaussian profile is being added to an area that was already defined with the opposite dopant type, you can use the JUNCTION parameter to specify the position of the junction depth instead of specifying the standard deviation using the CHARACTERISTIC parameter.

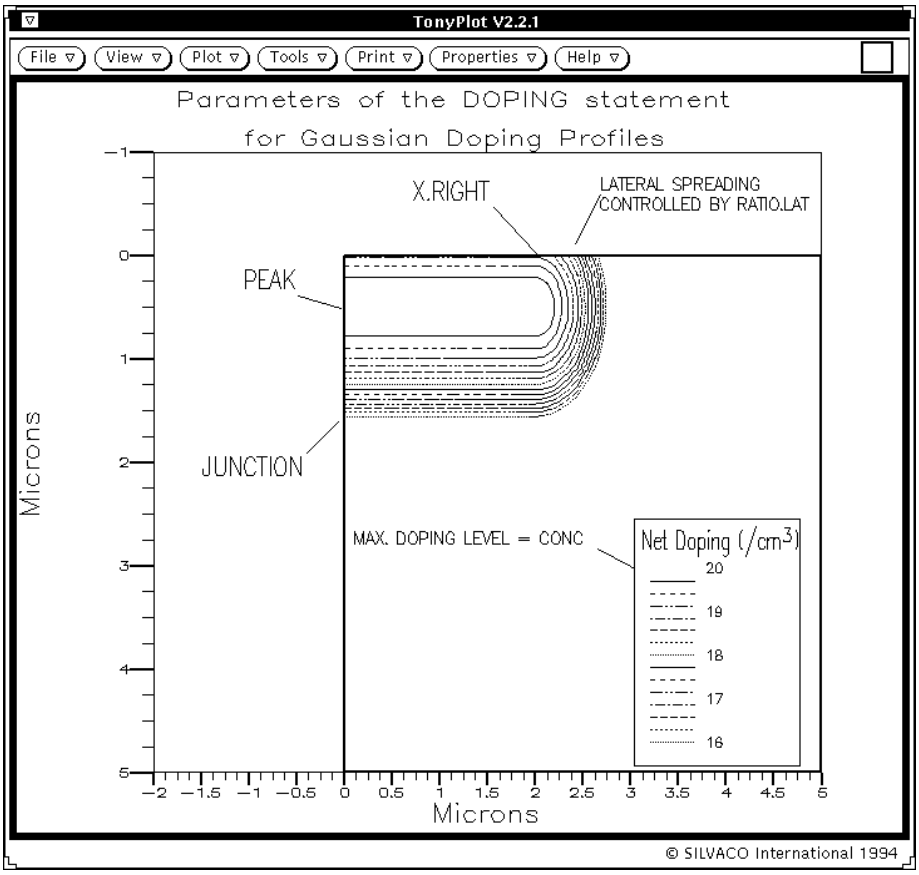


Figure 2-5: Analytical specification of a 2D Profile

Importing 1D SSUPREM3 Doping Profiles

One-dimensional doping profiles can be read into ATLAS from a SSUPREM3 output file. The doping data must have been saved from SSUPREM3 using the statement:

```
STRUCTURE OUTFILE=<output filename>
```

at the end of the SSUPREM3 run.

In ATLAS, the MASTER parameter of the DOPING statement specifies that a SSUPREM3 file will be read by ATLAS. Since this file will usually contain all the dopants from the SSUPREM3 simulation, the desired dopant type must also be specified. For example, the statement:

```
DOPING MASTER INFILE=mydata.dat BORON REGION=1
```

specifies that the boron profile from the file mydata.dat should be imported and used in region #1. SSUPREM3 profiles are imported into ATLAS one at a time (i.e., one DOPING statement is used for each profile or dopant). The statements:

```
DOPING MASTER INFILE=mydata.dat BORON OUTFILE=doping.dat
DOPING MASTER INFILE=mydata.dat ARSENIC X.RIGHT=0.8 RATIO=0.75
DOPING MASTER INFILE=mydata.dat ARSENIC X.LEFT=2.2 RATIO=0.75
```

offset the arsenic doping from boron to create a 2-D doping profile from a single SSUPREM3 result.

It is advisable to include the OUTFILE parameter on the first DOPING statement to create a 2-D doping file. This file will then be used in the next section to interpolate doping on a refined mesh after a REGRID. This file, however, can't be plotted in TONYPLOT. The position parameters and the RATIO.LATERAL parameter are used in the same manner as for analytical doping profiles to set the extent of the 1-D profile.

2.6.4: Automatic Meshing (Auto-meshing) Using The Command Language

Automatic meshing provides a simpler method for defining device structures and meshs than the standard method described in Section 2.6.3: "Using The Command Language To Define A Structure". Auto-meshing is particularly suited for epitaxial structures, especially device structures with many layers (for example, a VCSEL device). Auto-meshing unburdens you from the delicate bookkeeping involved in ensuring that the locations of mesh lines in the Y direction are consistently aligned with the edges of regions. This is done by specifying the locations of Y mesh lines in the REGION statements. The following sections will show how auto-meshing is done, using a few simple examples.

Specifying The Mesh And Regions

In the first example, we use a simple device to show you the fundimental concepts of auto-meshing. The first statements in this example are as follows:

```
MESH AUTO
X.MESH LOCATION=-1.0 SPACING=0.1
X.MESH LOCATION=1.0  SPACING=0.1
```

These statements are similar to the ones used in the standard method that described a mesh using the command language as shown in Section 2.6.3: "Using The Command Language To Define A Structure". There are, however, two key differences. The first difference is the inclusion of the AUTO parameter in the MESH statement. You need this parameter to indicate that you want to use auto-meshing. The second and more important difference is that in this example we will not specify any Y.MESH statements. This is because the locations of Y mesh lines will be automatically determined by the parameters of the REGION statements.

You can still specify one or more Y.MESH statements. Such defined mesh lines will be included in the mesh. But including Y.MESH statements is optional in auto-meshing.

In the next few statements of the example, we will show you several new concepts that will explain how auto-meshing works. The following four lines describe the regions in the example device:

```
REGION TOP    THICKNESS=0.02 MATERIAL=GaN   NY=5  DONOR=1E16
REGION BOTTOM THICKNESS=0.1  MATERIAL=AlGaN NY=5  DONOR=1E17    X.COMP=0.2
REGION TOP    THICKNESS=0.08 MATERIAL=AlGaN NY=4  ACCEPTOR=1E17 X.COMP=0.2
REGION BOTTOM THICKNESS=0.5  MATERIAL=AlGaN NY=10 DONOR=1E18    X.COMP=0.2
```

New Concepts

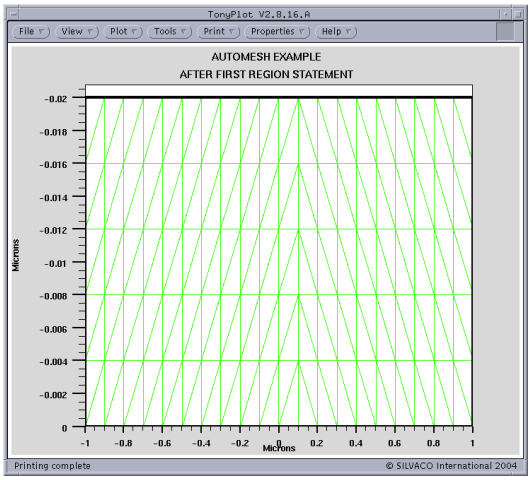
First, it appears that composition and doping are being specified in the REGION statement. This is the case for the DONOR, ACCEPTOR, X.COMPOSITION and Y.COMPOSITION parameters in the REGION statement that specify uniform doping or composition or both over the specified region. These parameters are also available to the standard methods described in Section 2.6.3: "Using The Command Language To Define A Structure", but are more amenable to specification of epitaxial structures such as we are describing in this example.

Next, you should notice several other new parameters. These are the TOP, BOTTOM, THICKNESS and NY parameters. All of these are used to describe the relative locations and thicknesses of the layers as well as the locations of the Y mesh lines. The most intuitive of these parameters is the THICKNESS parameter, which describes the thickness in microns, in the Y direction of each layer. As for the extent in the X direction, in the absence of any specified X.MIN or X.MAX parameters, it is assumed that the region extends over the full range of the X mesh described above in the X.MESH statements.

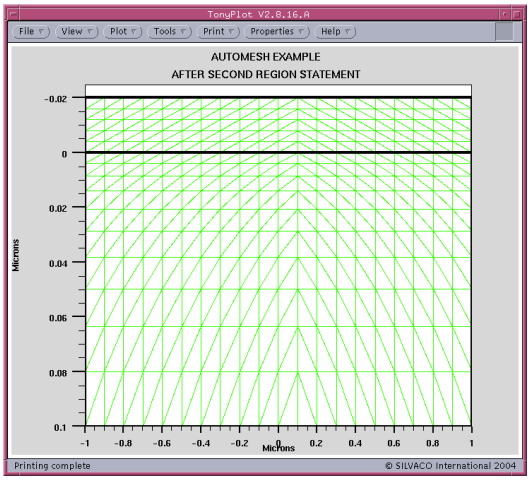
The NY parameter describes how many Y mesh lines are contained in the region so that the Y mesh lines are evenly spaced over the region. You can use the SY parameter instead of NY to specify the spacing in microns between Y mesh lines in the region. Make sure the value of SY does not exceed the value of THICKNESS. Generally, the relationship between SY, NY and THICKNESS can be expressed by the following:

```
SY = THICKNESS/NY
```

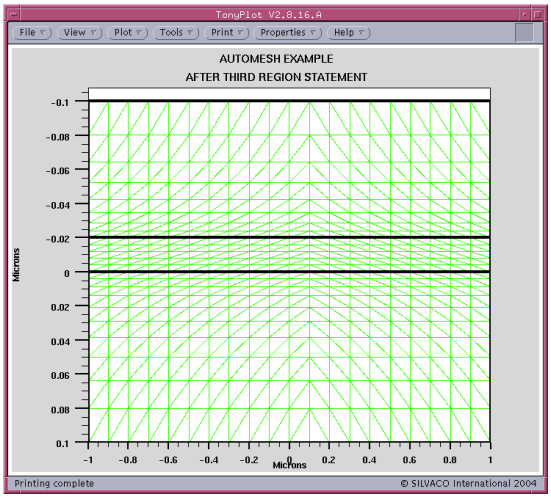
Figure 2-6 shows the meanings of the TOP and BOTTOM parameters.



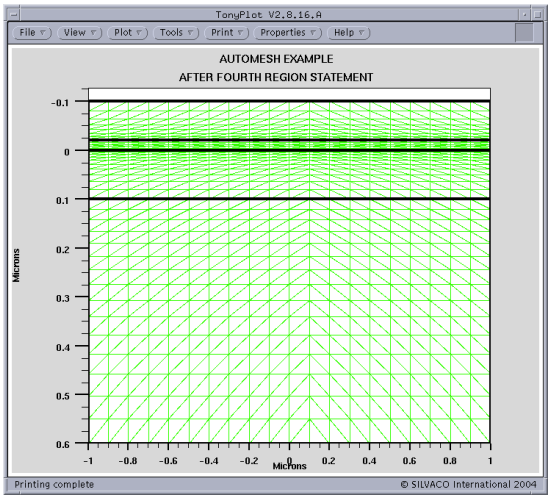
a) Structure after 1st REGION Statement



b) Structure after 2nd REGION Statement



c) Structure after 3rd REGION Statement



d) Structure after 4th REGION Statement

Figure 2-6: Simple Example of Auto-meshing Showing Sequence of Structure Development.

This figure shows a progression of representations of how the structure's mesh and region outlines appear after each REGION statement. This figure should give you an intuitive feel of how the regions are alternately placed on the top or bottom of the structure according to the specification of the TOP or BOTTOM parameters. It is important to keep in mind that the ATLAS coordinate convention for the Y axis is that positive Y is directed down into the device. This is similar to using the TOP and BOTTOM parameters of the ELECTRODE statement.

One thing you might notice in this figure is that the number of Y mesh lines in each region does not always match the number specified. This is because at each interface between regions, the Y mesh line spacing is ambiguously defined and the auto-meshing algorithm will always pick the smaller spacing between the two at each interface. Then, the spacing between Y mesh lines varies continuously between subsequent interfaces in a similar way as it does for mesh spacings specified by the LOCATION and SPACING parameters in the X.MESH and Y.MESH statements.

The auto-meshing algorithm maintains the "notion" of the current Y locations of the "top" and "bottom". Let's call these locations "Ytop" and "Ybottom".

Before any REGION statements are processed, "Ytop" and "Ybottom" are both defined to be equal to zero. As the REGION statements are processed, the following cases are addressed:

- If you place the region on the top, as specified by the TOP parameter, the region will extend from "Ytop" to "Ytop"-THICKNESS (remember positive Y points down) and "Ytop" will move to the new location "Ytop"-THICKNESS.
- If you place the region on the bottom, as specified by the BOTTOM parameter, the region will extend from "Ybottom" to "Ybottom"+THICKNESS and "Ybottom" will move to the new location "Ybottom"+THICKNESS.

The auto-meshing algorithm will ensure that all regions are perfectly aligned to their neighboring regions and there are no inconsistencies between the locations of Y mesh lines and the region edges that they resolve.

Non-uniformity In The X Direction and Auto-meshing

In some cases, you may want to define a device with material discontinuities in the X direction. Such discontinuities may represent etched mesa structures or oxide aperatures. There are a couple of ways to do this in auto-meshing. For example, you want to etch out a region from the structure of the previous example, from X=0 to the right and from Y=0 to the top. You can add the statement

```
REGION MATERIAL=Air X.MIN=0.0 Y.MAX=0.0
```

In this statement, we are not using the auto-meshing features but are using syntax of the standard meshing methods described in Section 2.6.3: "Using The Command Language To Define A Structure". ATLAS supports mixing the standard syntax with auto-meshing but be careful when doing this as we will explain shortly. Figure 2-7 shows the resulting structure and mesh after adding this statement. In this figure, we see that we have obtained the desired result.

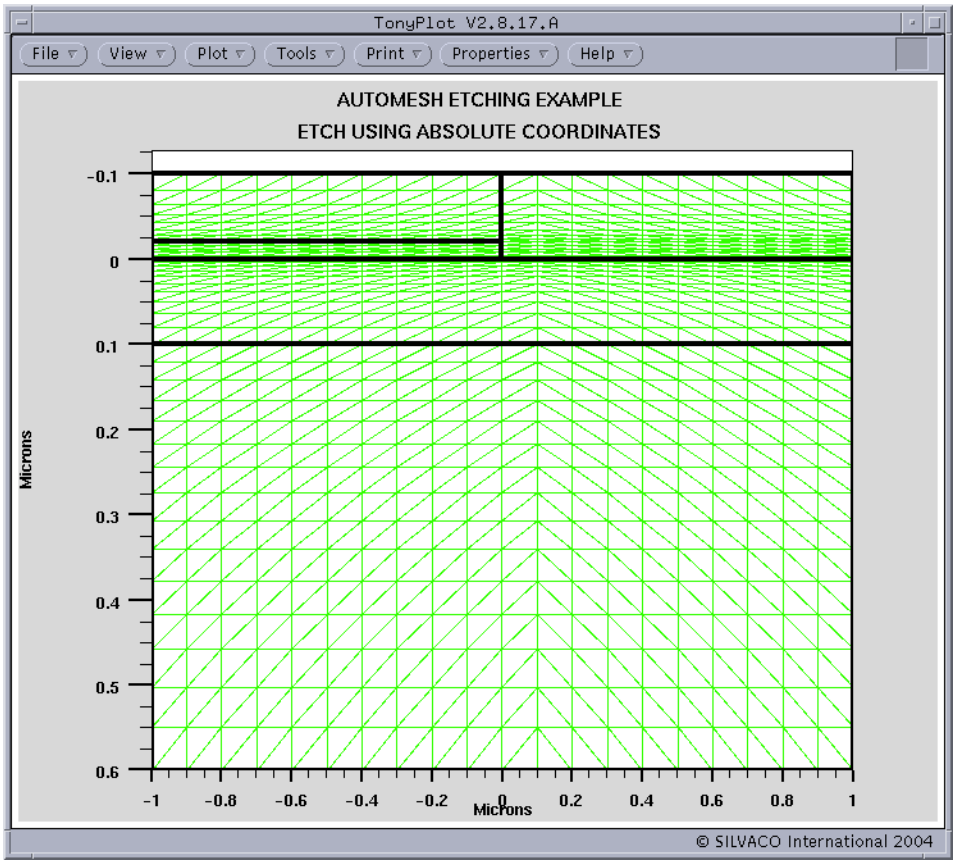


Figure 2-7: Structure Etch Example in Auto-meshing

The potential pitfall in using absolute Y coordinates in auto-meshing is that the location you choose, for example, by summing up thicknesses may not match the location your computer has calculated with its inherent numerical imprecision. The result is not only that the resulting structure may not exactly match what you desire. More importantly, you may end up accidently creating a mesh with Y mesh lines closely spaced (usually in the order of the machine precision). This can cause numerical instability (poor convergence) and can overflow or underflow in certain models. What's worse is that this situation is difficult to detect.

There is, however, one situation where we can absolutely predict the location of an edge or Y mesh line. That is at `Y=0`. This is exactly what we have done in our example. So if you want to use auto-meshing with specifications of an absolute value of Y, then arrange your device structure so that the specification of Y will be at zero.

This also applies to the location of a recessed electrode. Make sure it is located at `Y=0` if you are using auto-meshing.

There is another method of providing for discontinuities in material in the X direction that is absolutely safe from the previously discussed problems. In this approach, we use another parameter called `STAY` in the `REGION` statement in conjunction with the `TOP` or `BOTTOM` parameters.

The following describes the effect of the `STAY` parameter in the `REGION` statement.

- If you place the region on the top, as specified by the `TOP` parameter, and the `STAY` parameter is specified, the region will extend from `"Ytop"` to `"Ytop"-THICKNESS` and `"Ytop"` **will remain in its current position**.
- If you place the region on the bottom, as specified by the `BOTTOM` parameter, and the `STAY` parameter is specified, the region will extend from `"Ybottom"` to `"Ybottom"+THICKNESS` and `"Ybottom"` **will remain in its current position**.

The use of the `STAY` parameter can best be illustrated by the following example. In this example, we will reproduce the same structure discussed in the last example but this time using only `STAY` parameters and by not using any specification of absolute Y coordinates. The new `REGION` specifications are as follows:

```
REGION BOTTOM   THICKNESS=0.1  MATERIAL=AlGaN NY=5  DONOR=1E17    X.COMP=0.2
REGION BOTTOM   THICKNESS=0.5  MATERIAL=AlGaN NY=10 DONOR=1E18    X.COMP=0.2
REGION TOP STAY THICKNESS=0.02 MATERIAL=GaN   NY=5  DONOR=1E16
REGION TOP      THICKNESS=0.02 MATERIAL=Air   NY=5  X.MIN=0.0
REGION TOP STAY THICKNESS=0.08 MATERIAL=AlGaN NY=4  ACCEPTOR=1E17 X.COMP=0.2
REGION TOP      THICKNESS=0.08 MATERIAL=Air   NY=4  X.MIN=0.0
```

In this example, we slightly rearranged the `REGION` statements for clarity and split one region into two. Figure 2-8 shows the results.

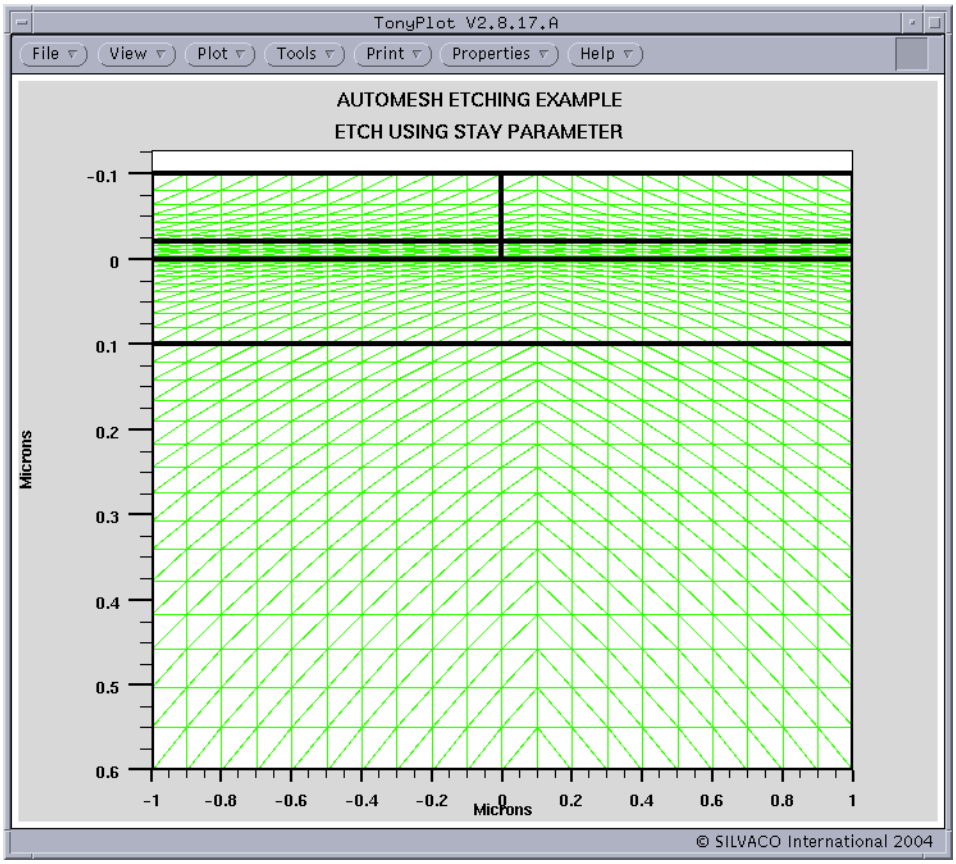


Figure 2-8: Sturcture Etching Example Using The STAY Parameter

The following describes the operation of the STAY parameter in this example:

- The STAY parameter in the third REGION statement means that the fourth REGION will start at the same Y coordinate as the third.
- Since the THICKNESS and NY parameters are the same in the third and fourth regions, they will have the same range of Y values and the same Y mesh.
- The specification of X.MIN in the fourth REGION statement means that the region will not completely overlap the third region but will only overlap to a minimum X value of 0.
- The lack of a STAY parameter in the fourth REGION statement means that the fifth region will lie directly atop the third and fourth regions.
- The STAY parameter in the fifth REGION statement means that the sixth REGION will start at the same Y coordinate as the fifth.
- Since the THICKNESS and NY parameters are the same in the fifth and sixth regions, they will have the same range of Y values and the same Y mesh.
- The specification of X.MIN in the sixth REGION statement means that the region will not completely overlap the fifth region but will only overlap to a minimum X value of 0.

As you can see, using the STAY parameter carefully avoids the problems of specifying values of Y coordinates and the potential problems involved. By doing so, you can specify arbitrary stepped structures.

Grading of Compositional and Doping

We have provided another method for specifying composition or doping or both in the REGION statement that is available for both auto-meshing and meshing using the standard method described in Section 2.6.3: “Using The Command Language To Define A Structure”. This method allows linear grading of rectangular regions. Eight new parameters of the REGION statement support this function. They are ND.TOP, ND.BOTTOM, NA.TOP, NA.BOTTOM, COMPX.TOP, COMPX.BOTTOM, COMPY.TOP, and COMPY.BOTTOM. In the syntax of these parameter names, "TOP" refers to the "top" or extreme Y coordinate in the negative Y direction, and "BOTTOM" refers to the "bottom" or extreme Y coordinate in the positive Y direction. With this in mind, the following rules apply:

- If you specify ND.TOP and ND.BOTTOM in a REGION statement, the donor doping in the region will vary linearly from the value specified by ND.TOP at the "top" of the device to the value specified by ND.BOTTOM at the "bottom" of the device.
- If you specify NA.TOP and NA.BOTTOM in a REGION statement, the acceptor doping in the region will vary linearly from the value specified by NA.TOP at the "top" of the device to the value specified by NA.BOTTOM at the "bottom" of the device.
- If you specify COMPX.TOP and COMPX.BOTTOM in a REGION statement, the X composition fraction in the region will vary linearly from the value specified by COMPX.TOP at the "top" of the device to the value specified by COMPX.BOTTOM at the "bottom" of the device.
- If you specify COMPY.TOP and COMPY.BOTTOM in a REGION statement, the Y composition fraction in the region will vary linearly from the value specified by COMPY.TOP at the "top" of the device to the value specified by COMPY.BOTTOM at the "bottom" of the device.

Note: Any subsequent DOPING statements will add to the doping specified by the doping related parameters on the REGION statement.

Superlattices and Distributed Bragg Reflectors DBRs

For auto-meshing, we have provided a convenient way of specifying a certain class of superlattices or as they are commonly used distributed Bragg reflectors (DBRs). This class of superlattice includes any superlattice that can be described as integer numbers of repetitions of two different layers. By different, we mean that the two layers may have different thicknesses, material compositions, or dopings, or all. These "conglomerate" structures are specified by the DBR or SUPERLATTICE statement. Actually, SUPERLATTICE is a synonym for DBR. Therefore in the following discussion, we will use the DBR syntax and recognize that SUPERLATTICE and DBR can be used interchangeably.

Most of the syntax of the DBR statement is similar to the syntax of the REGION statement, except the syntax is set up to describe two regions (or two sets of regions). As you will see, we differentiate these regions (or sets of regions) by the indices 1 and 2 in the parameter’s name.

The following example should make this concept clear.

```
MESH AUTO
X.MESH LOCATION=-1.0 SPACING=0.1
X.MESH LOCATION=1.0  SPACING=0.1

DBR TOP LAYERS=4 THICK1=0.1 THICK2=0.2 N1=2 N2=3 MAT1=GaAs MAT2=AlGaAs \
X2.COMP=0.4
DBR BOTTOM LAYERS=3 THICK1=0.05 THICK2=0.075 N1=3 N2=3 MAT1=AlGaAs \
MAT2=GaAs X1.COMP=0.4
```

In this example, we can see we are using auto-meshing because of the appearance of the AUTO parameter in the MESH statement. The DBR statements should be familiar since the functionality and syntax are similar to the REGION statement. We will not discuss those similarities here. We will only discuss the important differences. For more information about the similarities, see Chapter 19: "Statements" .

The main new parameter is the LAYERS parameter. This parameter specifies the total number of regions added. The region indexed "1" is always added first, then the region indexed "2" is added, and depending on the value of LAYERS, the sequence of regions continues 1, 2, 1, 2, 1, ...

Figure 2-9 shows the functionality of the DBR statement, which gives the resulting structure and mesh from the example.

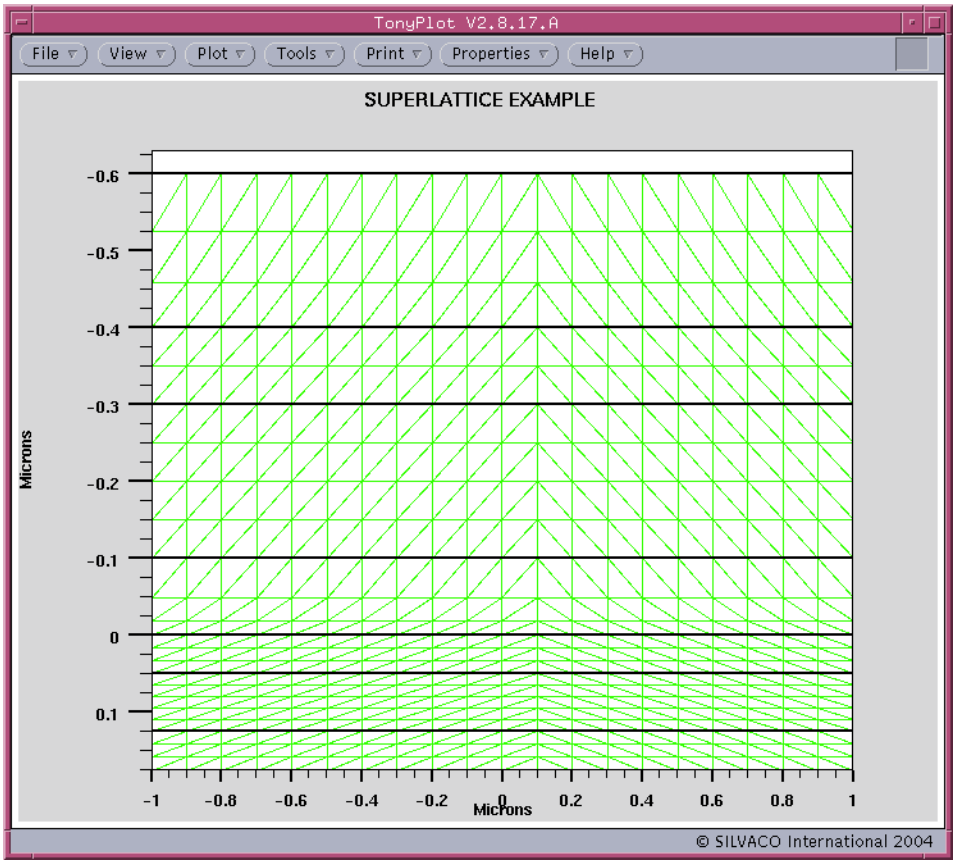


Figure 2-9: Supperlattice Example Structure

2.6.5: Modifying Imported Regions

If you import a device structure from a file using the INFILE parameter of the MESH statement, you may want to modify some of the characteristics of one or more of the regions in the structure. To do this, specify a REGION statement with the MODIFY parameter and the NUMBER or NAME parameter assigned to the region number of interest. You can specify/respecify any of the following REGION statement parameters: STRAIN, WELL.CNBS, WELL.VNBS, WELL.GAIN, POLAR.SCALE, QWELL, LED, WELL.FIELD, and POLARIZATION.

2.6.6: Remeshing Using The Command Language

It can be difficult to define a suitable grid when specifying a structure with the command language. The main problem is that the meshes required to resolve 2-D doping profiles and curved junctions are quite complicated. Simple rectangular meshes require an excessive number of nodes to resolve such profiles. If a device structure only includes regions of uniform doping, then there's usually no need to regrid. But when realistic 2-D doping profiles are present, a regrid may be necessary.

Note: The recommended solution for defining complex mesh structures for ATLAS is to use the standalone program, DEVEDIT.

Regrid On Doping

ATLAS includes a regridding capability that generates a fine mesh only in a localized region. You specify a quantity on which the regrid is to be performed. The mesh is then refined in regions where the specified quantity varies rapidly. Whenever a specified quantity (usually doping) changes quickly, the regridding will automatically grade the mesh accordingly. You can get a regrid on doping before any solutions are obtained. You can do this by using the statement:

```
REGRID LOGARITHM DOPING RATIO=2 SMOOTH.KEY=4 DOPFILE=<filename1> \
       OUTFILE=<filename2>
```

This statement must be used after the MESH, REGION, MATERIAL, ELECTRODE, and DOPING statements described previously. The effects of this REGRID statement on a simple diode structure are shown in Figure 2-10.

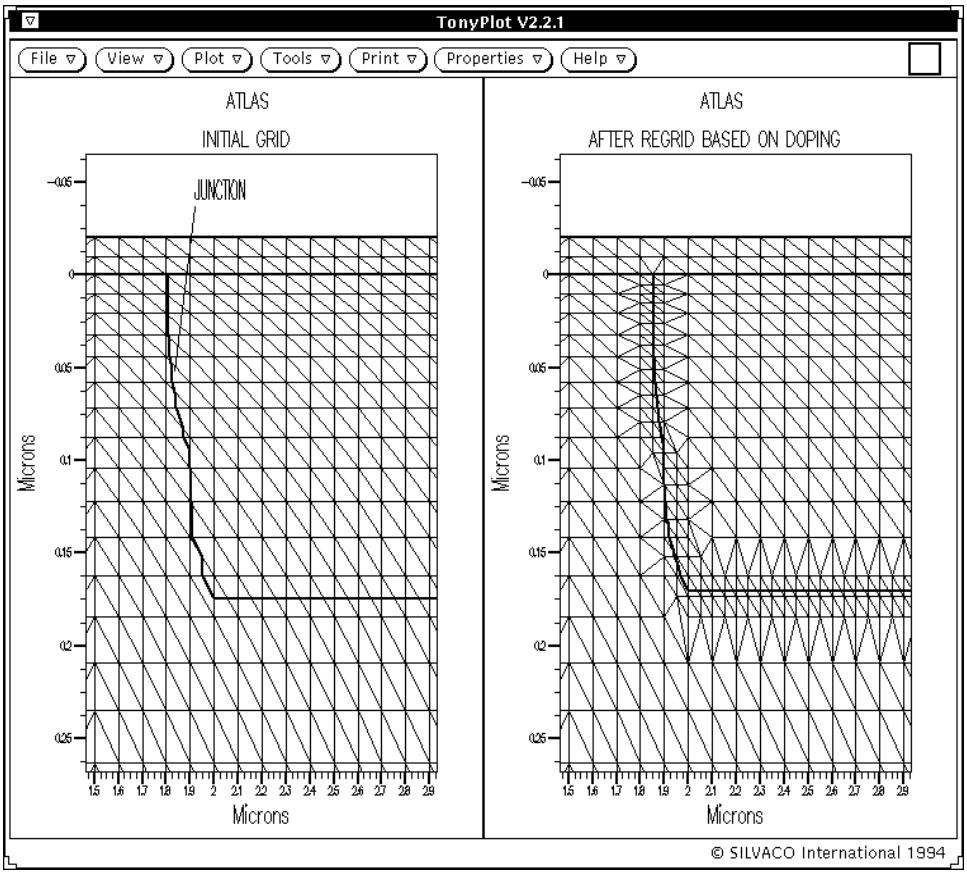


Figure 2-10: Regrid on doping provides improved resolution of junction

In this statement, the regridding will resolve doping profiles to two orders of magnitude in change.

The doping file, *filename1*, must be specified in the first DOPING statement with the OUTFILE parameter. The results of the regrid are saved in the file, *filename2*. The SMOOTH.KEY parameter value selects a smoothing algorithm. A value of 4 is typically best as this algorithm tends to produce the fewest obtuse triangles. For a complete description of the various smoothing algorithms, see Chapter 18: "Numerical Techniques".

Regrid Using Solution Variables

The REGRID statement can use a wide range of solution variables as the basis for mesh refinement. A regrid on solution variables can only be used after a solution has been obtained. After a regrid on a solution variable, the solution must be re-solved at the same bias in ATLAS.

For example:

```
REGRID POTENTIAL RATIO=0.2 MAX.LEVEL=1 SMOOTH.K=4 DOPFILE=<filename1>
SOLVE PREV
```

Regrid on potential is often used for high voltage power devices.

Note: You can use the REGRID statement any number of times on a structure. But we advise you to quit and restart ATLAS between regrids on electrical quantities. You can use the go atlas statement to do this. This should be followed by a MESH statement, loading the output file of the REGRID command, and a re-setting of all material and model parameters.

2.6.7: Specifying 3D Structures

The syntax for forming 3-D device structures is an extension of the 2-D syntax described in the previous section. The MESH statement should appear as:

```
MESH THREE.D
```

The THREE.D parameter tells ATLAS that a three dimensional grid will be specified. The other statements used to specify 3-D structures and grids are the same as for 2-D with the addition of z direction specifications. The statements:

```
MESH THREE.D X.MESH LOCATION=0 SPACING=0.15
X.MESH LOCATION=3 SPACING=0.15
Y.MESH LOCATION=0 SPACING=0.01
Y.MESH LOCATION=0.4 SPACING=0.01
Y.MESH LOCATION=3 SPACING=0.5
Z.MESH LOCATION=0 SPACING=0.1
Z.MESH LOCATION=3 SPACING=0.1
```

define a 3-D mesh that is uniform in the x and z directions and varies in the y direction.

Position parameters for the z direction (Z.MIN and Z.MAX) are also used on REGION, ELECTRODE, or DOPING statements.

2.6.8: General Comments Regarding Grids

Specifying a good grid is a crucial issue in device simulation but there is a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine grid that resolves the structure in solutions. Numerical efficiency is greater when fewer grid points are used. The critical areas to resolve are difficult to generalize because they depend on the technology and the transport phenomena. The only generalization possible is that most critical areas tend to coincide with reverse-biased metallurgical junctions. Typical critical areas are:

- High electric fields at the drain/channel junction in MOSFETs
- The transverse electric field beneath the MOSFET gate
- Recombination effects around the emitter/base junction in BJTs
- Areas of high impact ionization
- Around heterojunctions in HBT's and HEMTs.

The CPU time required to obtain a solution is typically proportional to N^{α}, where N is the number of nodes and α varies from 2 to 3 depending on the complexity of the problem. Thus, the most efficient way is to allocate a fine grid only in critical areas and a coarser grid elsewhere.

The three most important factors to look for in any grid are:

- Ensure adequate mesh density in high field areas
- Avoid obtuse triangles in the current path or high field areas
- Avoid abrupt discontinuities in mesh density

For more information about grids, see Chapter 18: "Numerical Techniques", Section 18.3: "Meshes".

2.6.9: Maximum Numbers Of Nodes, Regions, and Electrodes

ATLAS sets some limits on the maximum number of grid nodes that can be used. But this shouldn't be viewed as a bottleneck to achieving simulation results. In the default version, 2-D ATLAS simulations have a maximum node limit of 20,000. 3-D ATLAS simulations have an upper limit of 40,000,000 nodes with no more than 20,000 nodes in any one z-plane and a maximum number of z-planes of 2,000.

This limit is high enough that for almost all simulations of conventional devices, running out of nodes is never an issue. For most 2-D simulations, you can obtain accurate results with somewhere between 2,000 and 4,000 node points properly located in the structure.

If you exceed the node limits, error messages will appear and ATLAS will not run successfully. There are two options to deal with this problem. The first option is to decrease the mesh density because simulations with the maximum nodes take an extremely long time to complete. The second option is to contact your local Silvaco office (support@silvaco.com). Versions of 2-D ATLAS with up to 100,000 nodes have been distributed.

A node point limitation below these values might be seen due to virtual memory constraints on your hardware. For each simulation, ATLAS dynamically allocates the virtual memory. See the SILVACO INSTALLATION GUIDE for information about virtual memory requirements. The virtual memory used by the program depends on the number of nodes and on items, such as the models used and the number of equations solved.

Also, there is a node limit for the number of nodes in the X or Y directions in 2-D and 3-D ATLAS. In the standard version, this limit is 20,000 nodes. This is applicable to meshes defined in the ATLAS syntax using `X.MESH` and `Y.MESH` statements.

The maximum number of regions defined in both 2-D and 3-D ATLAS is 1,000. The maximum number of definable electrodes is 50. Again, if you need to include more than the maximum number of regions or electrodes, contact your local Silvaco office (support@silvaco.com).

2.7: Defining Material Parameters And Models

Once you define the mesh, geometry, and doping profiles, you can modify the characteristics of electrodes, change the default material parameters, and choose which physical models ATLAS will use during the device simulation. To accomplish these actions, use the CONTACT, MATERIAL, and MODELS statements respectively. Impact ionization models can be enabled using the IMPACT statement. Interface properties are set by using the INTERFACE statement.

Many parameters are accessible through the SILVACO C-INTERPRETER (SCI), which is described in Appendix A: "C-Interpreter Functions". This allows you to define customized equations for some models. For more information about the INTERFACE and MODELS statements, see Chapter 19: "Statements", Sections 19.18: "INTERFACE" and 19.29: "MODELS".

2.7.1: Specifying Contact Characteristics

Workfunction for Gates or Schottky Contacts

An electrode in contact with semiconductor material is assumed by default to be ohmic. If a work function is defined, the electrode is treated as a Schottky contact. The CONTACT statement is used to specify the metal workfunction of one or more electrodes. The NAME parameter is used to identify which electrode will have its properties modified.

The WORKFUNCTION parameter sets the workfunction of the electrode. For example, the statement:

```
CONTACT NAME=gate WORKFUNCTION=4.8
```

sets the workfunction of the electrode named gate to 4.8eV. The workfunctions of several commonly used contact materials can be specified using the name of the material. You can specify workfunctions for ALUMINUM, N.POLYSILICON, P.POLYSILICON, TUNGSTEN, and TU.DISILICIDE in this way. The following statement sets the workfunction for a n-type polysilicon gate contact.

```
CONTACT NAME=gate N.POLYSILICON
```

Aluminum contacts on heavily doped silicon is usually ohmic. For this situation, don't specify a workfunction. For example, MOS devices don't specify:

```
CONTACT NAME=drain ALUMINUM     /* wrong */
```

The CONTACT statement can also be used to specify barrier and dipole lowering of the Schottky barrier height. To enable barrier lowering, specify the BARRIER parameter, while specifying dipole lowering using the ALPHA parameter. For example, the statement:

```
CONTACT NAME=anode WORKFUNCTION=4.9 BARRIER ALPHA=1.0e-7‘
```

sets the work function of the Schottky contact named anode to 4.9eV enables barrier lowering and sets the dipole lowering coefficient to 1 nm.

Note: When a Schottky barrier is defined at a contact, we recommend that a fine y mesh is present just beneath the contact inside the semiconductor. This allows the Schottky depletion region to be accurately simulated.

Setting Current Boundary Conditions

The CONTACT statement is also used to change an electrode from voltage control to current control. Current controlled electrodes are useful when simulating devices, where the current is highly sensitive to voltage or is a multi-valued function of voltage (e.g., post-breakdown and when there is snap-back).

The statement:

```
CONTACT NAME=drain CURRENT
```

changes the drain electrode to current control. The BLOCK or NEWTON solution methods are required for all simulations using a current boundary condition. For more information about these methods, see Chapter 18: "Numerical Techniques", Section 18.5: "Non-Linear Iteration".

Defining External Resistors, Capacitors, or Inductors

Lumped resistance, capacitance, and inductance connected to an electrode can be specified using the RESISTANCE, CAPACITANCE, and INDUCTANCE parameters in the CONTACT statement. For example, the statement:

```
CONTACT NAME=drain RESISTANCE=50.0 CAPACITANCE=20e-12 INDUCTANCE=1e-6
```

specifies a parallel resistor and capacitor of 50 ohms and 20 pF respectively in series with a 1 μH inductor. Note that in 2D simulations, these passive element values are scaled by the width in the third dimension. Since in 2D ATLAS assumes a 1μm width, the resistance becomes 50 Ω-μm.

Distributed contact resistance for an electrode can be specified using the CON.RESIST parameter. For example, the statement:

```
CONTACT NAME=source CON.RESISTANCE=0.01
```

specifies that the source contact has a distributed resistance of 0.01 Ωcm^2.

Note: Simulations with external resistors, capacitors, or inductors must be solved using the BLOCK or NEWTON solution method.

Floating Contacts

The CONTACT statement is also used to define a floating electrode. There are two distinctly different situations where floating electrodes are important. The first situation is for floating gate electrodes used in EEPROM and other programmable devices. The second situation is for contacts directly onto semiconductor materials such as floating field plates in power devices.

To enable floating gates, specify the FLOATING parameter on the CONTACT statement. For example, the statement:

```
CONTACT NAME=fgate FLOATING
```

specifies that the electrode named fgate will be floating and that charge boundary conditions will apply.

For contacts directly onto semiconductor, the FLOATING parameter cannot be used. This type of floating electrode is best simulated by specifying current boundary conditions on the CONTACT statement. For example, the statement:

```
CONTACT NAME=drain CURRENT
```

specifies current boundary conditions for the electrode named drain. On subsequent SOLVE statements, the drain current boundary condition will default to zero current. Therefore, floating the contact.

You can also make a floating contact to a semiconductor using a very large resistor attached to the contact instead. For example:

```
CONTACT NAME=drain RESIST=1e20
```

Note that extremely large resistance values must be used to keep the current through the insignificant contact. Using a lumped resistor will allow the tolerance on potential to move slightly above zero. For example, if the tolerance is 10^{-5}V and the defined resistance was only 10MΩμm, then a current of 10^{-12} A/μm may flow through the contact, which may be significant in breakdown simulations.

Shorting Two Contacts Together

It is possible in ATLAS to tie two or more contact together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initially defined.

If the structure is defined using ATLAS syntax, you can have multiple ELECTRODE statements with the same NAME parameter defining separate locations within the device structure. In this case, the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both ELECTRODE areas.

Also, if two separate metal regions in ATHENA are defined using the ATHENA ELECTRODE statement to have the same name, then in ATLAS these two electrodes will be considered as shorted together.

If the electrodes are defined with different names, the following syntax can be used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base
.
SOLVE VBASE=0.1
```

Here, the electrode, base1, will be linked to the electrode, base. The applied 0.1V on base will then appear on base1. ATLAS, however, will calculate and store separate currents for both base and base1. This can be a useful feature. But in some cases, such as where functions of the currents are required in EXTRACT or TONYPLOT, it is undesirable. You can add the SHORT parameter to the CONTACT statement above to specify so that only a single base current will appear combining the currents from base and base1.

When loading a structure from ATHENA or DEVEDIT, where two defined electrode regions are touching, ATLAS will automatically short these and use the electrode name that was defined first.

Making an Open Circuit Contact

It is often required to perform a simulation with an open circuit on one of the defined electrodes. There are three different methods to make an open circuit contact. The first method is to entirely deleting an electrode from the structure file. The second method is to add an extremely large lumped resistance. For example, $10^{20}\Omega$ onto the contact to be made open circuit. The third method is to switch the boundary conditions on the contact to be made open circuit from voltage controlled to current controlled. Then, specifying a very small or zero current through that electrode.

Each of these methods are feasible. If a floating region, however, is created within the structure, then numerical convergence may be affected. As a result, we normally recommend that you use the second method because it ensures better convergence.

2.7.2: Specifying Material Properties

Semiconductor, Insulator, or Conductor

All materials are split into three classes: semiconductors, insulators and conductors. Each class requires a different set of parameters to be specified. For semiconductors, these properties include electron affinity, band gap, density of states and saturation velocities. There are default parameters for material properties used in device simulation for many materials.

Appendix B: "Material Systems" lists default material parameters and includes a discussion on the differences between specifying parameters for semiconductors, insulators, and conductors.

Setting Parameters

The MATERIAL statement allows you to specify your own values for these basic parameters. Your values can apply to a specified material or a specified region. For example, the statement:

```
MATERIAL MATERIAL=Silicon EG300=1.12 MUN=1100
```

sets the band gap and low field electron mobility in all silicon regions in the device. If the material properties are defined by region, the region is specified using the REGION or NAME parameters in the MATERIAL statement. For example, the statement:

```
MATERIAL REGION=2 TAUN0=2e-7 TAUP0=1e-5
```

sets the electron and hole Shockley-Read-Hall recombination lifetimes for region number two (see Chapter 3: "Physics", the "Shockley-Read-Hall (SRH) Recombination" section on page 3-79 for more information about this type of recombination). If the name, base, has been defined using the NAME parameter in the REGION statement, then the statement:

```
MATERIAL NAME=base NC300=3e19
```

sets the conduction band density of states at 300 K for the region named base.

The description of the MATERIAL statement in Chapter 19: "Statements", Section 19.24: "MATERIAL" provides a complete list of all the material parameters that are available.

Heterojunction Materials

The material properties of heterojunctions can also be modified with the MATERIAL statement. In addition to the regular material parameters, you can define composition dependent material parameters. For example, composition dependent band parameters, dielectric constants, and saturation velocities.

For heterojunction material systems, the bandgap difference between the materials is divided between conduction and valence bands. The ALIGN parameter specifies the fraction of this difference applied to the conduction band edge. This determines the electron and hole barrier height and overrides any electron affinity specification. For example, the statement:

```
MATERIAL MATERIAL=InGaAs ALIGN=0.36
MATERIAL MATERIAL=InP    ALIGN=0.36
```

specifies that 36% of the bandgap difference between InGaAs and InP is applied to the conduction band and 64% is applied to the valence band. For example, if the band gap difference (Δ Eg) for this material system is 0.6 eV, then the conduction band barrier height is 0.216 eV and the valence band barrier height is 0.384 eV.

For heterojunction devices, the transport models may be different for each material. You can specify these models and their coefficients for each material using the MODEL statement. See Section 2.7.4: "Specifying Physical Models" for a description of this option.

2.7.3: Specifying Interface Properties

The INTERFACE statement is used to define the interface charge density and surface recombination velocity at interfaces between semiconductors and insulators. For example, the statement:

```
INTERFACE QF=3e10
```

specifies that all interfaces between semiconductors and insulators have a fixed charge of $3.10^{10}cm^{-2}$. In many cases, the interface of interest is restricted to a specific region. This can be accomplished with the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters on the INTERFACE statement. These parameters define a rectangle, where the interface properties apply. For example, the statement:

```
INTERFACE QF=3e10 X.MIN=1.0 X.MAX=2 Y.MIN=0.0 Y.MAX=0.5
```

restricts the interface charge to the semiconductor-insulator boundary within the specified rectangle. In addition to fixed charge, surface recombination velocity and thermionic emission are enabled and defined with the INTERFACE statement. For more information about this statement, see Chapter 19: "Statements", Section 19.18: "INTERFACE".

2.7.4: Specifying Physical Models

Physical models are specified using the MODELS and IMPACT statements. Parameters for these models appear on many statements including: MODELS, IMPACT, MOBILITY, and MATERIAL. The physical models can be grouped into five classes: mobility, recombination, carrier statistics, impact ionization, and tunneling. Chapter 3: "Physics", Section 3.6: "Physical Models" contains details for each model.

Tables 2-1 to 2-5 give summary descriptions and recommendations on the use of each model. Table 2-6 is a guide for compatibility between models.

All models with the exception of impact ionization are specified on the MODELS statement. Impact ionization is specified on the IMPACT statement. For example, the statement:

```
MODELS CONMOB FLDMOB SRH FERMIDIRAC
IMPACT SELB
```

specifies that the standard concentration dependent mobility, parallel field mobility, Shockley-Read-Hall recombination with fixed carrier lifetimes, Fermi Dirac statistics and Selberherr impact ionization models should be used.

ATLAS also provides an easy method for selecting the correct models for various technologies. The MOS, BIP, PROGRAM, and ERASE parameters for the MODELS statement configure a basic set of mobility, recombination, carrier statistics, and tunneling models. The MOS and BIP parameters enable the models for MOSFET and bipolar devices, while PROGRAM and ERASE enable the models for programming and erasing programmable devices. For example, the statement:

```
MODELS MOS PRINT
```

enables the CVT, SRH, and FERMIDIRAC models, while the statement:

```
MODELS BIPOLAR PRINT
```

enables the CONMOB, FLDMOB, CONSRH, AUGER, and BGN.

Note: The PRINT parameter lists to the run time output the models and parameters, which will be used during the simulation. This allows you to verify models and material parameters. We highly recommend that you include the PRINT parameter in the MODEL statement.

Physical models can be enabled on a material by material basis. This is useful for heterojunction device simulation and other simulations where multiple semiconductor regions are defined and may have different characteristics. For example, the statement:

```
MODEL MATERIAL=GaAs FLDMOB EVSATMOD=1 ECRITN=6.0e3 CONMOB
MODEL MATERIAL=InGaAs SRH FLDMOB EVSATMOD=1 \
ECRITN=3.0e3
```

change both the mobility models and critical electric field used in each material. For devices based on advanced materials, these model parameters should be investigated carefully.

Energy Balance Models

The conventional drift-diffusion model of charge transport neglects non-local effects, such as velocity overshoot and reduced energy dependent impact ionization. ATLAS can model these effects through the use of an energy balance model, which uses a higher order approximation of the Boltzmann Transport Equation (see Chapter 3: "Physics", Section 3.1.3: "The Transport Equations"). In this equation, transport parameters, such as mobility and impact ionization, are functions of the local carrier temperature rather than the local electric field.

To enable the energy balance transport model, use the HCTE, HCTE.EL, or HCTE.HO parameters in the MODELS statement. These parameters enable the energy transport model for both carriers, electrons only, or holes only respectively. For example, the statement:

```
MODELS MOS HCTE
```

enables the energy balance transport model for both electrons and holes in addition to the default MOSFET models.

2.7.5: Summary Of Physical Models

Table 2-1. Carrier Statistics Models		
Model	**Syntax**	**Notes**
Boltzmann	BOLTZMANN	Default model
Fermi-Dirac	FERMI	Reduced carrier concentrations in heavily doped regions (statistical approach).
Incomplete Ionization	INCOMPLETE	Accounts for dopant freeze-out. Typically, it is used at low temperatures.
Silicon Ionization Model	IONIZ	Accounts for full ionization for heavily doped Si. Use with INCOMPLETE.
Bandgap Narrowing	BGN	Important in heavily doped regions. Critical for bipolar gain. Use Klaassen Model.

Table 2-2. Mobility Models		
Model	**Syntax**	**Notes**
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora's Model	ARORA	Alternative to ANALYTIC for Si
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity saturation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.
Klaassen Model	KLA	Includes N, T, and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices

Table 2-2. Mobility Models		
Model	**Syntax**	**Notes**
Shirahata Model	SHI	Includes N, $E_{\perp}$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_{\perp}$ effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E//, and $E_{\perp}$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_{\perp}$ effects. Only calibrated for 300K.

Table 2-3. Recombination Models		
Model	**Syntax**	**Notes**
Shockley-Read-Hall	SRH	Uses fixed minority carrier lifetimes. Should be used in most simulations.
Concentration Dependent	CONSRH	Uses concentration dependent lifetimes. Recommended for Si.
Auger	AUGER	Direct transition of three carriers. Important at high current densities.
Optical	OPTR	Band-band recombination. For direct materials only.
Surface	S.N S.P	Recombination at semiconductor to insulator interfaces. This is set in the INTERFACE statement.

Table 2-4. Impact Ionization		
Model	**Syntax**	**Notes**
Selberherr's Model	IMPACT SELB	Recommended for most cases. Includes temperature dependent parameters.
Grant's Model	IMPACT GRANT	Similar to Selberherr's model but with different coefficients.
Crowell-Sze	IMPACT CROWELL	Uses dependence on carrier scattering length.
Toyabe Model	IMPACT TOYABE	Non-local model used with Energy Balance. Any IMPACT syntax is accepted.

Table 2-4. Impact Ionization		
Model	**Syntax**	**Notes**
Concannon	N.CONCAN P.CONCAN	Non-local model developed in Flash EEPROM technologies.

Table 2-5. Tunneling Models and Carrier Injection Models		
Model	**Syntax**	**Notes**
Band-to-Band (standard)	BBT.STD	For direct transitions. Required with very high fields.
Concannon Gate Current Model	N.CONCAN P.CONCAN	Non-local gate model consistent with Concannon substrate current model.
Direct Quantum tunneling (Electrons)	QTUNN.EL	Quantum tunneling through conduction band barrier due to an insulator.
Direct Quantum tunneling (Hole)	QTUNN.HO	Quantum tunneling through valence band barrier due to an insulator.
Fowler-Nordheim (electrons)	FNORD	Self-consistent calculation of tunneling through insulators. Used in EEPROMs.
Fowler-Nordheim (holes)	FNHOLES	Same as FNORD for holes.
Klaassen Band-to-Band	BBT.KL	Includes direct and indirect transitions.
Hot Electron Injection	HEI	Models energetic carriers tunneling through insulators. Used for gate current and Flash EEPROM programming.
Hot Hole Injection	HHI	HHI means hot hole injection.

Note: In the notes in Tables 2-1 through 2-5, n is electron concentration, p is hole concentration, T is lattice temperature, N is dopant concentration, Ell is parallel electric field, and E$\perp$ is perpendicular electric field.

Table 2-6. Model Compatibility Chart

	CONMOB	FLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURMOB	LATTICE H	E.BALANCE
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	CV		—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	2

Key To Table Entries

MODEL ABBREVIATION = The model that supersedes when a combination is specified. In some cases, a warning message is issued when a model is ignored.

OK = This combination is allowed.

NO = This combination is not allowed.

NOTES:

1.Uses internal model similar to FLDMOB

2.When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

Using the C-Interpreter to Specify Models

One of the ATLAS products is a C language interpreter that allows you to specify many of the models used by ATLAS. To use these functions, implement the model in C as equations in a special file called an *ATLAS lib file*. You can access the default ATLAS template file by typing:

```
atlas -T <filename>
```

at the UNIX command prompt. This creates a default template file with the name specified by the `<filename>` parameter. See Appendix A: "C-Interpreter Functions" for a listing of the default C-INTERPRETER functions. To use the interpreter functions, give the corresponding parameters in the statements containing the name of the C language file with the model given as the parameter value. For example, the statement:

```
MATERIAL NAME=Silicon F.MUNSAT=munsat.lib
```

specifies that the file, *munsat.lib*, contains the C-INTERPRETER function for the specification of the parallel field dependent electron mobility model.

2.8: Choosing Numerical Methods

2.8.1: Numerical Solution Techniques

Several different numerical methods can be used for calculating the solutions to semiconductor device problems. Numerical methods are given in the METHOD statements of the input file. Some guidelines for these methods will be given here. For full details, however, see Chapter 18: "Numerical Techniques".

Different combinations of models will require ATLAS to solve up to six equations. For each of the model types, there are basically three types of solution techniques: (a) decoupled (GUMMEL), (b) fully coupled (NEWTON) and (c) BLOCK. The GUMMEL method will solve for each unknown in turn keeping the other variables constant, repeating the process until a stable solution is achieved. The NEWTON method solve the total system of unknowns together. The BLOCK methods will solve some equations fully coupled while others are de-coupled.

Generally, the GUMMEL method is useful where the system of equations is weakly coupled but has only linear convergence. The NEWTON method is useful when the system of equations is strongly coupled and has quadratic convergence. The NEWTON method may, however, spend extra time solving for quantities, which are essentially constant or weakly coupled. NEWTON also requires a more accurate initial guess to the problem to obtain convergence. Thus, a BLOCK method can provide for faster simulations times in these cases over NEWTON. GUMMEL can often provide better initial guesses to problems. It can be useful to start a solution with a few GUMMEL iterations to generate a better guess. Then, switch to NEWTON to complete the solution. Specification of the solution method is carried out as follows:

```
METHOD GUMMEL BLOCK NEWTON
```

The exact meaning of the statement depends on the particular models it applied to. This will be discussed in the following sections.

Basic Drift Diffusion Calculations

The isothermal drift diffusion model requires the solution of three equations for potential, electron concentration, and hole concentration. Specifying GUMMEL or NEWTON alone will produce simple Gummel or Newton solutions as detailed above. For almost all cases, the NEWTON method is preferred and it is the default.

Specifying:

```
METHOD GUMMEL NEWTON
```

will cause the solver to start with GUMMEL iterations. Then, switch to NEWTON if convergence is not achieved. This is a robust but a more time consuming way of obtaining solutions for any device. This method, however, is highly recommended for all simulations with floating regions such as SOI transistors. A floating region is defined as an area of doping, which is separated from all electrodes by a pn junction.

BLOCK is equivalent to NEWTON for all isothermal drift-diffusion simulations.

Drift Diffusion Calculations with Lattice Heating

When the lattice heating model is added to drift diffusion, an extra equation is added. The BLOCK algorithm solves the three drift diffusion equations as a NEWTON solution. Follows this with a GUMMEL solution of the heat flow equation. The NEWTON algorithm solves all four equations in a coupled manner. NEWTON is preferred once the temperature is high. BLOCK, however, is quicker for low temperature gradients. Typically, the combination used is:

```
METHOD BLOCK NEWTON
```

Energy Balance Calculations

The energy balance model requires the solution of up to 5 coupled equations. GUMMEL and NEWTON have the same meanings as with the drift diffusion model. In other words, GUMMEL specifies a de-coupled solution and NEWTON specifies a fully coupled solution.

But BLOCK performs a coupled solution of potential, carrier continuity equations followed by a coupled solution of carrier energy balance, and carrier continuity equations.

You can switch from BLOCK to NEWTON by specifying multiple solution methods on the same line. For example:

```
METHOD BLOCK NEWTON
```

will begin with BLOCK iterations. Then, switch to NEWTON if convergence is still not achieved. This is the most robust approach for many energy balance applications.

The points where the algorithms switch is pre-determined, but can be changed in the METHOD statement, the default values set by Silvaco work well for most circumstances.

Energy Balance Calculations with Lattice Heating

When non-isothermal solutions are performed in conjunction with energy balance models, a system of up to six equations must be solved. GUMMEL or NEWTON solve the equations iteratively or fully coupled respectively. BLOCK initially performs the same function as with energy balance calculations, then solves the lattice heating equation in a de-coupled manner.

Setting the Number Of Carriers

ATLAS can solve both electron and hole continuity equations, or only for one or none. You can make this choice by using the CARRIERS parameter. For example:

```
METHOD CARRIERS=2
```

specifies a solution for both carriers is required. This is the default. With one carrier, the ELEC or HOLE parameter is needed. For example, for hole solutions only:

```
METHOD CARRIERS=1 HOLE
```

To select a solution for potential, only specify:

```
METHOD CARRIERS=0
```

Note: Setting the number of carriers using the syntax, MODEL NUMCARR=<n>, is obsolete and should not be used.

Important Parameters Of the METHOD Statement

You can alter all of the parameters relevant to the numerical solution process. This, however, isn't recommended unless you have expert knowledge of the numerical algorithms. All of these parameters have been assigned optimal values for most solution conditions. For more information about numerical algorithms, see Chapter 18: "Numerical Techniques".

The following parameters, however, are worth noting at this stage:

- CLIMIT or CLIM.DD specify minimal values of concentrations to be resolved by the solver. Sometimes, you need to reduce this value to aid solutions of breakdown characteristics. A value of CLIMIT=1e-4 is recommended for all simulations of breakdown, where the pre-breakdown current is small. CLIM.DD is equivalent to CLIMIT but uses the more convenient units of cm^{-3} for the critical concentration. CLIM.DD and CLIMIT are related by the following expression.

$$\text{CLIM.DD} = \text{CLIMIT}^* \sqrt{N_c N_v} \qquad \text{2-1}$$

- DVMAX controls the maximum update of potential per iteration of Newton's method. The default corresponds to 1V. For power devices requiring large voltages, you may need an increased value of DVMAX. DVMAX=1e8 can improve the speed of high voltage bias ramps.
- CLIM.EB controls the cut-off carrier concentration below, which the program will not consider the error in the carrier temperature. This is applied in energy balance simulations to avoid excessive calculations of the carrier temperature at locations in the structure where the carrier concentration is low. Setting this parameter too high, where ATLAS ignores the carrier temperature errors for significant carrier concentrations, will lead to unpredictable and mostly incorrect results .

Restrictions on the Choice of METHOD

The following cases require METHOD NEWTON CARRIERS=2 to be set for isothermal drift-diffusion simulations:

- current boundary conditions
- distributed or lumped external elements
- AC analysis
- impact ionization

Both BLOCK or NEWTON or both are permitted for lattice heat and energy balance.

Note: Simulations using the GUMMEL method in these cases may lead to non-convergence or incorrect results.

Pisces-II Compatibility

Previous releases of ATLAS (2.0.0.R) and other PISCES-II based programs use the SYMBOLIC command to define the solution method and the number of carriers included in the solution. In this version of ATLAS, the solution method is specified completely on the METHOD statement.

The COMB parameter, which was available in earlier ATLAS versions, is no longer required. It has been replaced with either the BLOCK method or the combination of GUMMEL and NEWTON parameters. Table 2-7 identifies direct translations of old syntax to new.

Note: These are direct translations and not necessarily the best choices of numerical methods.

Table 2-7. Parameter Syntax Replacements

Old Syntax (V2.0.0.R)	New Syntax
`symbolic newton carriers=2`	`method newton`
`symbolic newton carriers=1 elec`	`method newton carriers=1 electron`
`symbolic gummel carriers=0`	`method gummel carriers=0`
`symbolic newton carriers=2` `method comb`	`method gummel newton`

`models lat.temp`
`symbolic newton carriers=2`
`method comb`
	`models lat.temp`
	`method block`

`models hcte`
`symbolic gummel carriers=2`
`method comb`
	`models hcte`
	`method block`

2.9: Obtaining Solutions

ATLAS can calculate DC, AC small signal, and transient solutions. Obtaining solutions is similar to setting up parametric test equipment for device tests. You usually define the voltages on each of the electrodes in the device. ATLAS then calculates the current through each electrode. ATLAS also calculates internal quantities, such as carrier concentrations and electric fields throughout the device. This is information that is difficult or impossible to measure.

In all simulations, the device starts with zero bias on all electrodes. Solutions are obtained by stepping the biases on electrodes from this initial equilibrium condition. As will be discussed, due to the initial guess strategy, voltage step sizes are limited. This section concentrates on defining solution procedures. To save results, use the LOG or SAVE statements. Section 2.10: "Interpreting The Results" on how to analyze and display these results.

2.9.1: DC Solutions

In DC solutions, the voltage on each electrode is specified using the SOLVE statement. For example, the statements:

```
SOLVE VGATE=1.0
SOLVE VGATE=2.0
```

solves a single bias point with 1.0V and then 2.0V on the gate electrode. One important rule in ATLAS is that when the voltage on any electrode is not specified in a given SOLVE statement, the value from the last SOLVE statement is assumed.

In the following case, the second solution is for a drain voltage of 1.0V and a gate voltage of 2.0V.

```
SOLVE VGATE=2.0
SOLVE VDRAIN=1.0
```

When the voltage on a particular electrode is never defined on any SOLVE statement and voltage is zero, you don't need to explicitly state the voltage on all electrodes on all SOLVE statements. For example, in a MOSFET, if VSUBSTRATE is not specified, then Vbs defaults to zero.

Sweeping The Bias

For most applications, a sweep of one or more electrodes is usually required. The basic DC stepping is inconvenient and a ramped bias should be used. To ramp the base voltage from 0.0V to 1.0V with 0.05V steps with a fixed collector voltage of 2.0V, use the following syntax:

```
SOLVE VCOLLECTOR=2.0
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base
```

The NAME parameter is required and the electrode name is case-sensitive. Make sure the initial voltage, VSTEP and VFINAL, are consistent. A badly specified ramp from zero to 1.5V in 0.2V steps will finish at 1.4V or 1.6V.

Generating Families Of Curves

Many applications such as MOSFET Id/Vds and bipolar Ic/Vce simulations require that a family of curves is produced. This is done by obtaining solutions at each of the stepped bias points first, and then solving over the swept bias variable at each stepped point. For example, in MOSFET Id/Vds curves, solutions for each Vgs value are obtained with Vds=0.0V. The output from these solutions are saved in ATLAS solution files. For each gate bias, the solution file is loaded and the ramp of drain voltage performed.

The family of curves for three 1V gate steps and a 3.3V drain sweep would be implemented in ATLAS as follows:

```
SOLVE VGATE=1.0  OUTF=solve_vgate1
SOLVE VGATE=2.0  OUTF=solve_vgate2
SOLVE VGATE=3.0  OUTF=solve_vgate3

LOAD INFILE=solve_vgate1
LOG OUTFILE=mos_drain_sweep1
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate2
LOG OUTFILE=mos_drain_sweep2
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate3
LOG OUTFILE=mos_drain_sweep3
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3
```

The LOG statements are used to save the Id/Vds curve from each gate voltage to separate files. We recommend that you save the data in this manner rather than to a single LOG file (see Section 2.10: “Interpreting The Results”).

2.9.2: The Importance Of The Initial Guess

To obtain convergence for the equations used, supply a good initial guess for the variables to be evaluated at each bias point. The ATLAS solver uses this initial guess and iterates to a converged solution. For isothermal drift diffusion simulations, the variables are the potential and the two carrier concentrations. If a reasonable grid is used, almost all convergence problems in ATLAS are caused by a poor initial guess to the solution.

During a bias ramp, the initial guess for any bias point is provided by a projection of the two previous results. Problems tend to appear near the beginning of the ramp when two previous results are not available. If one previous bias is available, it is used alone. This explains why the following two examples eventually produce the same result. The first will likely have far more convergence problems than the second.

```
1.  SOLVE VGATE=1.0 VDRAIN=1.0 VSUBSTRATE=-1.0
2.  SOLVE VGATE=1.0
    SOLVE VSUBSTRATE=-1.0
    SOLVE VDRAIN=1.0
```

In the first case, one solution is obtained with all specified electrodes at 1.0V. In the second case, the solution with only the gate voltage at 1.0V is performed first. All other electrodes are at zero bias. Next, with the gate at 1.0V, the substrate potential is raised to -1.0V and another solution is obtained. Finally, with the substrate and the gate biased, the drain potential is added and the system solved again. The advantage of this method over the first case is that the small incremental changes in voltage allow for better initial guesses at each step.

Generally, the projection method for the initial guess gives good results when the I-V curve is linear. But it may encounter problems if the IV curve is highly non-linear or if the device operating mode is changing. Typically, this may occur around the threshold or breakdown voltages. At these biases, smaller voltage steps are required to obtain convergence. As will be described, ATLAS contains features such as the TRAP parameter and the curve tracer to automatically cut the voltage steps in these highly non-linear area.

Numerical methods are described in Section 2.8: "Choosing Numerical Methods" and Chapter 18: "Numerical Techniques".

In many cases, these methods are designed to overcome the problems associated with the initial guess. This is particularly important in simulations involving more than the three drift diffusion variables. Generally, coupled solutions require a good initial guess, whereas de-coupled solutions can converge with a poor initial guess.

The Initial Solution

When no previous solutions exist, the initial guess for potential and carrier concentrations must be made from the doping profile. This is why the initial solution performed must be the zero bias (or thermal equilibrium) case. This is specified by the statement:

```
SOLVE INIT
```

But if this syntax isn't specified, ATLAS automatically evaluates an initial solution before the first SOLVE statement. To aid convergence of this initial guess, the solution is performed in the zero carrier mode solving only for potential.

The First and Second Non-Zero Bias Solutions

From experience with ATLAS, it is found that the first and second non-zero bias solutions are the most difficult to obtain good convergence. Once these two solutions are obtained, the projection algorithm for the initial guess is available and solutions should all have a good initial guess.

These first two solutions, however, must use the result of the initial solution as the basis of their initial guess. Since the initial solution is at zero bias, it provides a poor initial guess.

The practical result of this is that the first and second non-zero bias solutions should have very small voltage steps. In the following example, the first case will likely converge whereas the second case may not.

```
1.  SOLVE INIT
    SOLVE VDRAIN=0.1
    SOLVE VDRAIN=0.2
    SOLVE VDRAIN=2.0
2.  SOLVE INIT
    SOLVE VDRAIN=2.0
```

The Trap Parameter

Although ATLAS provides several methods to overcome a poor initial guess and other convergence problems, it is important to understand the role of the initial guess in obtaining each solution. The simplest and most effective method to overcome poor convergence is by using:

```
METHOD TRAP
```

This is enabled by default. Its effect is to reduce the bias step if convergence problems are detected. Consider the example from the previous section:

```
SOLVE INIT
SOLVE VDRAIN=2.0
```

If the second SOLVE statement does not converge, TRAP automatically cuts the bias step in half and tries to obtain a solution for Vd = 1.0V. If this solution does not converge, the bias step will be halved again to solve for Vd = 0.5V. This procedure is repeated up to a maximum number of tries set by the METHOD parameter MAXTRAPS. Once convergence is obtained, the bias steps are increased again to solve up to 2.0V. The default for MAXTRAPS is 4 and it is not recommended to increase it, since changing the syntax to use smaller bias steps is generally much faster.

This trap facility is very useful during bias ramps in overcoming convergence difficulties around transition points such as the threshold voltage. Consider the following syntax used to extract a MOSFET Id/Vgs curve.

```
SOLVE VGATE=0.0 VSTEP=0.2 VFINAL=5.0 NAME=gate
```

Assume the threshold voltage for the device being simulated is 0.7V and that ATLAS has solved for the gate voltages up to 0.6V. The next solution, at 0.8V, may not converge at first. This is because the initial guess was formed from the two sub-threshold results at Vgs=0.4V and 0.6V and the solution has now become non-linear. The trap facility will detect the problems in the 0.8V solution and cut the bias step in half to 0.7V and try again. This will probably converge. The solution for 0.8V will then be performed and the bias ramp will continue with 0.2V steps.

2.9.3: Small-Signal AC Solutions

Specifying AC simulations is a simple extension of the DC solution syntax. AC small signal analysis is performed as a post-processing operation to a DC solution. Two common types of AC simulation in ATLAS are outlined here. The results of AC simulations are the conductance and capacitance between each pair of electrodes. Tips on interpreting these results is described in Section 2.10: “Interpreting The Results”.

Single Frequency AC Solution During A DC Ramp

The minimum syntax to set an AC signal on an existing DC ramp is just the AC flag and the setting of the small signal frequency. For example:

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6
```

Other AC syntax for setting the signal magnitude and other parameters are generally not needed as the defaults suffice. One exception is in 1D MOS capacitor simulations. To obtain convergence in the inversion/deep depletion region, add the DIRECT parameter to access a more robust solution method.

Ramped Frequency At A Single Bias

For some applications, such as determining bipolar gain versus frequency, you need to ramp the frequency of the simulation. This is done using the following syntax:

```
1.  SOLVE VBASE=0.7 AC FREQ=1e9 FSTEP=1e9 NFSTEPS=10
2.  SOLVE VBASE=0.7 AC FREQ=1e6 FSTEP=2 MULT.F NFSTEPS=10
```

The first case ramps the frequency from 1GHz to 11GHz in 1GHz steps. A linear ramp of frequency is used and FSTEP is in Hertz. In the second example, a larger frequency range is desired and so a geometrical step of frequency is used. The MULT.F parameter is used to specify that FSTEP is a unitless multiplier for the frequency. This doubles the frequency in successive steps from 1MHz to 1.024GHz.

You can combine the following syntax for ramping the frequency of the AC signal with the syntax for ramping the bias. The frequency ramps are done at each bias point during the DC ramp.

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6 \
      FSTEP=2 MULT.F NFSTEPS=10
```

2.9.4: Transient Solutions

Transient solutions can be obtained for piecewise-linear, exponential, and sinusoidal bias functions. Transient solutions are used when a time dependent test or response is required. To obtain transient solutions for a linear ramp, specify the TSTART, TSTOP, TSTEP, and RAMPTIME parameters.

Figure 2-11 shows the syntax of the transient voltage ramp. The TSTART parameter specifies the time that the linear ramp should start. The RAMPTIME specifies the time that the linear ramp should obtain its final value. TSTOP specifies the time that solutions will stop. TSTEP specifies the initial step size.

Subsequent time steps are calculated automatically by ATLAS. For example, the statement:

```
SOLVE VGATE=1.0 RAMPTIME=1E-9 TSTOP=10e-9 TSTEP=1e-11,
```

specifies that the voltage on the gate electrode will be ramped in the time domain from its present value to 1.0V over a period of 1 nanoseconds.

Time domain solutions are obtained for an additional 9 nanoseconds. An initial time step of 10 picoseconds is specified. Note that if you specify subsequent transient solutions, don't reset the time to zero.

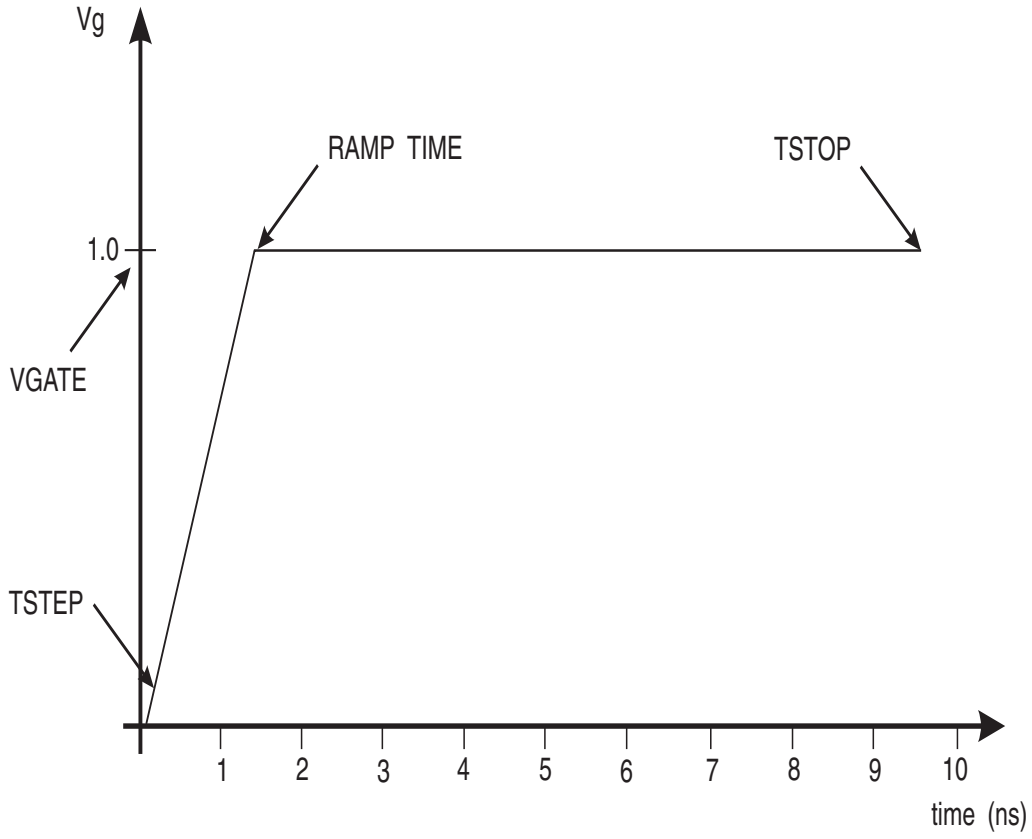


Figure 2-11: Diagram showing syntax of Transient Voltage Ramp in ATLAS

2.9.5: Advanced Solution Techniques

Obtaining Solutions Around The Breakdown Voltage

Obtaining solutions around the breakdown voltage can be difficult using the standard ATLAS approach. It requires special care when choosing voltage steps and interpreting the results. The curve tracer described in "The Curvetrace Capability" section on page 2-48 is the most effective method in many cases.

A MOSFET breakdown simulation might be performed using the following standard syntax for ramping the drain bias. Note that the setting of CLIMIT is recommended for breakdown simulations when the pre-breakdown leakage is low.

```
IMPACT SELB
METHOD CLIMIT=1e-4
SOLVE VDRAIN=1.0 VSTEP=1.0 VFINAL=20.0 NAME=drain
```

If the breakdown were 11.5V, then convergence problems will be expected for biases higher than 11.0V. Although it depends on technology used, it is common for the breakdown curve to be flat up to a voltage very close to breakdown and then almost vertical. The current changes by orders of magnitude for very small bias increments.

This produces some problems for ATLAS using the syntax described above. If the breakdown occurs at 11.5V, there are no solutions for voltages greater than this value. ATLAS is trying to ramp to 20.0V. Therefore, it is inevitable that ATLAS will fail to converge at some point. This is usually not a problem since by that point the breakdown voltage and curve have been obtained.

Above 11V, bias step reduction will take place due to the TRAP parameter. ATLAS will continually try to increase the drain voltage above 11.5V and those points will fail to converge. But it will solve points asymptotically approaching Vds=11.5V until it reaches the limit set by the MAXTRAPS parameter. If you use the default of 4 traps, the minimum allowed voltage step will be $1.0\times(0.5)^4$ or 0.004V. This is normally enough accuracy for determining the breakdown point. But the simulation might not allow the current to reach a sufficiently high level before MAXTRAPS is needed.

Typically in device simulation, the breakdown point is determined once the current is seen to increase above the flat pre-breakdown leakage value by two orders of magnitude in a small voltage increment. If you want to trace the full breakdown curve up to high current values, apply more advanced techniques than the simple voltage ramp. Two of these techniques are described in the following subsections. These methods may use extra CPU time.

Using Current Boundary Conditions

In all of the examples considered in the basic description of the SOLVE statement, it was assumed that voltages were being forced and currents were being measured. ATLAS also supports the reverse case through current boundary conditions. The current through the electrode is specified in the SOLVE statement and the voltage at the contact is calculated. Current boundary conditions are set using the CONTACT statement as described in Section 2.7: "Defining Material Parameters And Models".

The syntax of the SOLVE statement is altered once current boundary conditions are specified.

```
SOLVE IBASE=1e-6
```

The syntax above specifies a single solution at a given current.

```
SOLVE IBASE=1e-6 ISTEP=1e-6 IFINAL=5e-6 NAME=base
```

This sets a current ramp similar in syntax to the voltage ramp described earlier.

```
SOLVE IBASE=1e-10 ISTEP=10 IMULT IFINAL=1e-6 NAME=base
```

This is similar to the previous case. The IMULT parameter, however, is used to specify that ISTEP should be used as a multiplier for the current rather than a linear addition. This is typical for ramps of current since linear ramps are inconvenient when several orders of magnitude in current may need to be covered.

Important points to remember about current boundary conditions are that the problems of initial guess are more acute when very small (noise level) currents are used. Often, it is best to ramp the voltage until the current is above 1pA/μm and then switch to current forcing.

When interpreting the results, it is important to remember the calculated voltage on the electrode with current boundary conditions is stored as the internal bias. In other words, base int.bias in TONYPLOT or vint.base in DECKBUILD's extract syntax.

The Compliance Parameter

Compliance is a parameter used to limit the current or voltage through or on an electrode during a simulation. You can set an electrode compliance. After reaching this limit, the bias sweep will stop. This is similar to parametric device testing when we stop a device from being over-stressed or destroyed. The compliance refers to the maximum resultant current or voltage present after obtaining a solution. If you set an electrode voltage, the compliance will then refer to the electrode current. If there are current boundary conditions, you can set a voltage compliance.

The statements:

```
SOLVE VGATE=1.0
SOLVE NAME=drain VDRAIN=0 VFINAL=2 VSTEP=0.2 COMPL=1E-6 CNAME=drain
```

solve for IV on the gate and then ramps the drain voltage towards 2V in 0.2V steps. The simulation will stop if it reaches 1μA/μm of drain current before Vd=2V. Thus, as in parametric testing, you can define a particular level and set the simulation to solve up to that point. Once ATLAS reaches the compliance limit, it will simulate the next statement line in the command file.

The Curvetrace Capability

The automatic curve tracing algorithm can be invoked to enable ATLAS to trace out complex IV curves. The algorithm can automatically switch from voltage to current boundary conditions and vice versa. You can use a single SOLVE statement to trace out complex IV curves, such as breakdown curves and CMOS latch-up including the snapback region and second breakdown. The algorithm is based upon a dynamic load line approach.

For example, typical CURVETRACE and SOLVE statements to trace out an IV curve for the breakdown of a diode would look like:

```
CURVETRACE CONTR.NAME=cathode STEP.INIT=0.5 NEXST.RATIO=1.2 \
           MINCUR=1e-12 END.VAL=1e-3 CURR.CONT
SOLVE CURVETRACE
```

CONTR.NAME specifies the name of the electrode, which is to be ramped. STEP.INIT specifies the initial voltage step. NEXST.RATIO specifies the factor used to increase the voltage step in areas on the IV curve away from turning points. MINCUR sets a small current value above, which activates the dynamic load line algorithm. Below the MINCUR level, using the STEP.INIT and NEXST.RATIO determines the next solution bias. END.VAL stops the tracing if the voltage or current of the ramped electrode equals or exceeds END.VAL. Using VOLT.CONT or CURR.CONT specify whether END.VAL is a voltage or current value.

When you plot the log file created by the CURVETRACE statement in TONYPLOT, select the internal bias labeled int.bias for the ramped electrode instead of plotting the applied bias, which is labeled Voltage.

2.9.6: Using DeckBuild To Specify SOLVE Statements

The DeckBuild Solve menu can be used generate SOLVE statements. The menu has a spreadsheet style entry. To access this menu, select the **Command/Solutions/Solve...** button in DECKBUILD. To define a test, click the right mouse button in the worksheet and select the **Add new row** option. This will add a new row to the worksheet. This procedure should be repeated once per electrode in your device structure. The entry for each cell can then be edited to construct a SOLVE statement.

Some cells require the selection using a pop menu or the entry of numerical values. The electrode name is specified in the first cell. To edit the electrode name, use a popup menu by pressing the right mouse button on the cell. The second cell specifies whether the electrode will be a voltage (V), current (I) or charge (Q) controlled. The third cell specifies whether the SOLVE statement is to be a single DC solve (CONST), a swept DC variable (VAR1), a stepped DC variable (VAR2), or a transient solution (PULSE). The remaining cells specify the parameter values that are required for the type of solution desired.

The pop-up window to specify the solution file names are accessed through the **Props...** button. You can construct several SOLVE statements to create solve sequences, which define a test. This test may be saved in a file and read in using the **Save...** and **Load...** buttons.

2.10: Interpreting The Results

As already described in Section 2.2: "ATLAS Inputs and Outputs", ATLAS produces three different types of output files. These files are also described in the following sections.

2.10.1: Run-Time Output

Run-time output is provided at the bottom of the DeckBuild Window. If it's run as a batch job, the run-time output can be stored to a file.

Errors occurring in the run-time output will be displayed in this window. Note that not all errors will be fatal (as DECKBUILD tries to interpret the file and continue). This may cause a statement to be ignored, leading to unexpected results. We recommend that you check the run-time output of any newly created input file the first time it runs to intercept any errors.

If you specify the PRINT option within the MODELS statement, the details of material parameters and constants and mobility models will be specified at the start of the run-time output. This is a useful way of checking what parameters values and models are being applied in the simulation. We recommend that you always specify MODELS PRINT in input files.

During SOLVE statements, the error numbers of each equation at each iteration are displayed (this is a change from the previous ATLAS version). It isn't vital for you to understand the iteration information. It can, however, provide important insights in the case of convergence problems.

The output can be interpreted as follows:

```
proj     psi    n     p       psi     n         p
direct   x      x     x       rhs     rhs       rhs

  i   j  m  -5.00*  -5.00*  -5.00*  -26.0*  -17.3*  -17.3*
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 1     N  -1.932 -2.934 -1.932  -25.2  -10.19 -9.876
 2     N  -4.741 -5.64* -4.267  -28.8* -16.67 -15.47
 3     A  -11.3* -11.7* -9.63*  -28.8* -16.67 -18.0*

 Electrode  Va(V)        Jn(A/um)     Jp(A/um)     Jc(A/um)     Jt(A/um)
 ========================================================================
 gate       0.000e+00    -0.000e+00   -0.000e+00   0.000e+00    0.000e+00
 source     0.000e+00    -3.138e-13   -1.089e-35   -3.138e-13   -3.138e-13
 drain      1.000e-01    3.139e-13    1.076e-23    3.139e-13    3.139e-13
 substrate  0.000e+00    -6.469e-19   -8.853e-17   -8.918e-17   -8.918e-17
```

The top left value, proj, indicates the initial guess methodology used. The default projection method is used here. Alternatives are previous, local, or init. The second value, direct, indicates the solver type. This will either be direct or iterative.

The first three column headings: i, j, and m indicate the iteration numbers of the solution and the solution method. i indicates the outer loop iteration number for decoupled solutions. j indicates the inner loop number. m indicates the solution method by a single letter, which are:

- G = gummel
- B = block
- N = newton
- A = newton with autonr
- S = coupled Poisson-Schrodinger solution

The remaining column headings indicate which column lists the XNORM and RHSNORM errors for the equations being solved. See Chapter 18: "Numerical Techniques", Section 18.5.7: "Error Measures" for a full description of these errors. The values printed in each error column under the hashed line are the logarithm to base 10 of the error. Earlier PISCES versions would print the floating point value. The values printed above the dashed line in each column are the tolerances used.

When the star * symbol appears as the least significant digit in the number, it means this error measure has met its tolerance.

After achieving convergence, ATLAS lists the results by electrode. The column, *Va*, lists the voltage at the contact surface. This will differ from the applied voltage if external resistors or the curvetracer are used. All relevant current components are listed. Here, only electron, hole, conduction, and total currents are given. In other modes of simulation, these columns may differ.

The output of AC analysis, MIXEDMODE, and 3-D simulations differ from this standard. See Chapter 12: "MixedMode: Mixed Circuit and Device Simulator" for more information about MIXEDMODE.

ATLAS may produce a very large amount of run-time output for complex simulations. You can save run-time output to a file as shown in Section 2.3: "Modes of Operation".

2.10.2: Log Files

Log files store the terminal characteristics calculated by ATLAS. These are current and voltages for each electrode in DC simulations. In transient simulations, the time is stored. In AC simulations, the small signal frequency and the conductances and capacitances are saved. For example, the statement:

```
LOG OUTF=<FILENAME>
```

is used to open a log file. Terminal characteristics from all SOLVE statements after the LOG statement are then saved to this file along with any results from the PROBE statement.

To not save the terminal characteristics to this file, use another LOG statement with either a different log filename or the OFF parameter.

Typically, a separate log file should be used for each bias sweep. For example, separate log files are used for each gate bias in a MOS Id/Vds simulation or each base current in a bipolar Ic/Vce simulation. These files are then overlaid in TONYPLOT.

Log files contain only the terminal characteristics. They are typically viewed in TONYPLOT. Parameter extraction of data in log files can be done in DECKBUILD. Log files cannot be loaded into ATLAS to re-initialize the simulation.

Units Of Currents In Log files

Generally, the units of current written into the log file. Therefore, seen in TONYPLOT is Amperes per micron. This is because ATLAS is a two-dimensional simulator. It sets the third dimension (or z-direction) to be one micron. Thus, if you compare ATLAS 2D simulation results for a MOSFET versus the measured data from a MOSFET of width 20 micron, you need to multiply the current in the log file by 20.

There are four exceptions:

- In the 3D modules of ATLAS, the width is defined in the 3-D structure. Therefore, the units of the current are Amperes.
- In MIXEDMODE, set the width of devices. The current is also in Amperes.
- When cylindrical coordinates are used, the current written to the log file is integrated through the cylinder and is also in Amperes.
- When the WIDTH parameter in the MESH statement is used, the current is scaled by this factor and is in Amperes.

Similar rules apply for the capacitance and conductance produced by AC simulations. These are usually in 1/(ohms.microns) and Farads/micron respectively.

UTMOST Interface

ATLAS log files can be read directly into the batch mode, UTMOST. The following commands in UTMOST are used to read in a set of IV curves stored in separate log files.

```
INIT INF=<filename> MASTER
INIT INF=<filename> MASTER APPEND
```

Note: The UTMOST statement in ATLAS is no longer recommended for interfacing to UTMOST.

2.10.3: Parameter Extraction In DeckBuild

The EXTRACT command is provided within the DECKBUILD environment. It allows you to extract device parameters. The command has a flexible syntax that allows you to construct specific EXTRACT routines. EXTRACT operates on the previous solved curve or structure file. By default, EXTRACT uses the currently open log file. To override this default, supply the name of a file to be used by EXTRACT before the extraction routine. For example:

```
EXTRACT INIT INF="<filename>"
```

A typical example of using EXTRACT is the extraction of the threshold voltage of an MOS transistor. In the following example, the threshold voltage is extracted by calculating the maximum slope of the I_d / V_g curve, finding the intercept with the x-axis and then subtracting half of the applied drain bias.

```
EXTRACT NAME="nvt" XINTERCEPT(MAXSLOPE(CURVE (V."GATE", (I."DRAIN"))) \
-(AVE(V."DRAIN"))/2.0)
```

The results of the extraction will be displayed in the run-time output and will be by default stored in the file results.final. To store the results in a different file at the end of EXTRACT command, use the following option:

```
EXTRACT....DATAFILE="<filename>"
```

Cut-off frequency and forward current gain are of particular use as output parameters. These functions can be defined as follows:

```
# MAXIMUM CUTOFF FREQUENCY
EXTRACT NAME="FT_MAX" MAX(G."COLLECTOR""BASE"/(6.28*C."BASE""BASE"))

#FORWARD CURRENT GAIN
EXTRACT NAME="PEAK GAIN" MAX(I."COLLECTOR"/ I."BASE")
```

Note: Over 300 examples are supplied with ATLAS to provide many practical examples of the use of the EXTRACT statement.

AC Parameter Extraction

You can perform basic analysis of the capacitance and conductance matrix produced by ATLAS using DECKBUILD or TONYPLOT. The capacitance between gate and drain will be labeled as `Cgate>drain` in TONYPLOT or `c."gate""drain"` in DECKBUILD'S EXTRACT.

The total capacitance on any electrode is defined as `Celectrode>electrode`. Thus, the magnitude of `Cgate>gate` is the total gate capacitance.

The `LOG` statement also includes options for small-signal, two-port RF analysis including s-parameter extraction. The solutions are saved into the log file and in the run-time output. The list of options for RF analysis are:

```
s.param, y.param, h.param, z.param, abcd.param gains
```

Terminal impedance and parasitics are accounted for by adding any of the following:

```
impedance=<val>, rin=<val>, rout=<val>, rcommon=<val> or
rground=<val>, lin=<val>, lout=<val>,
lcommon=<val> or lground=<val>, width=<val>
```

The width defaults to 1μm and impedance defaults to 50Ω All parasitics default to zero.

The Stern stability factor, `k`, is calculated along with current gain (h21), GUmax, and GTmax when the `GAINS` option is added to the `LOG` statement.

The run-time output for AC analysis has been modified to only list the analysis frequency and electrode conductance/capacitance values. If one of the two-port options is added to the `LOG` statement (e.g., `S.PARAM`), the two-port parameters are also included in the run-time output.

Note: AC parameter conversion utilities in the `UTMOST` statement have been discontinued.

Additional Functions

`EXTRACT` has two additional important functions. The first function is to provide data for the VWF database. In other words, to store device parameters to the VWF database, you must evaluate them using `EXTRACT`. The second function is when using the DeckBuild Optimizer to tune parameters, use the `EXTRACT` statements as the optimization targets.

2.10.4: Functions In TonyPlot

The **Functions** Menu in TONYPLOT allows you to specify and plot functions of the terminal characteristics in the **Graph Function** text fields. For example, you can calculate transconductance using the following function:

```
dydx (drain current, gate voltage)
```

Current gain can be evaluated as:

```
collector current / base current
```

When creating functions, the key to correct syntax is that the name for any variable in a function is the same as that in the **Y Quantities** list on the **Display** menu.

2.10.5: Solution Files

Solution files or structure files provide an image of the device at a particular bias point (DC solution or transient solution point). This gives you the ability to view any evaluated quantity within the device structure in question, from doping profiles and band parameters to electron concentrations and electric fields. These files should be plotted using TONYPLOT.

The syntax used to generate these files is of two forms:

```
(1)   SAVE OUTFILE=<filename>
(2)   SOLVE .... OUTFILE=<filename>.sta MASTER [ONEFILEONLY]
```

Here in the second form, a file named `<filename>` will be saved with data from the previously solved bias point.

In this case, a structure file will be saved at each bias point solved in the solve statement. The last letter of the file name will be automatically incremented alphabetically (i.e., `*.sta`, `*.stb`, `*.stc...`, and so on). If you need the solution for the last bias point, you can add the `onefileonly` parameter to the command. The `<filename>.sta` file will be overwritten at each solution point.

Structure files can be large (1 - 2 MB), depending on the mesh density and quantities saved. It is recommended that unwanted structure files be deleted.

If many solution files have been written from a long simulation, it is often confusing to find out which solution file belongs to which bias point or transient time. The solution files should be plotted in TONYPLOT. In TONYPLOT, you can create 2-D contour plots and 1-D cutlines. To find out the bias of any solution file in TONYPLOT, select the plot, and press **b** on the keyboard.

Interpreting Contour Plots

Some quantities saved in the solution files are not evaluated at the node points during solutions. They are evaluated at the center of the sides of each triangle in the mesh. Values of quantities at each node are derived from averaging the values from the sides of triangles connected to that node. The weighting method used to do the averaging can be selected with options in the `OUTPUT` statement. It is possible that for some meshes, smoother contour plots can be obtained by choosing a non-default averaging method.

When interpreting the contour plots, it's important to remember that the solution file contains values only at each node point. The color fills seen in TONYPLOT are simply interpolations based on the node values. This may lead to occasional strange contour values. In these cases, check the node values using the probe in TONYPLOT.

The primary solution variables (potential, carrier concentration, lattice temperature and carrier temperatures) are calculated on the nodes of the ATLAS mesh. Therefore, they are always correct in TONYPLOT. But since ATLAS doesn't use nodal values of quantities, such as electric field and mobility, the actual values used in calculations cannot be determined from TONYPLOT or the structure files. The `PROBE` statement allows you to directly probe values at given locations in the structure. This provides the most accurate way to determine the actual values used in ATLAS calculations.

Customizing Solution Files (OUTPUT Statement)

Several quantities are saved by default within a structure file. For example, doping, electron concentration, and potential. You can also specify additional quantities (such as conduction band potential) by using the OUTPUT statement. This must precede the SAVE statement in question. For example, to save the conduction and valence band potentials, the following command would be used at some point before the relevant SAVE.

```
OUTPUT CON.BAND VAL.BAND
```

Saving Quantities from the Structure at each Bias Point (PROBE statement)

Structure files provide all data from the structure at a single bias point. The log files provide terminal characteristics for a set of bias points. Use the PROBE statement to combine these and allow certain structural quantities to be saved at each bias point.

The PROBE statement allows you to specify quantities to be saved at given XY locations. You can also save the maximum or minimum of certain quantities. The value from the PROBE at each bias point in DC or timestep in transient mode is saved to the log file. The syntax:

```
PROBE NAME=mycarriers N.CONC X=1 Y=0.1
```

saves the electron concentration at (1, 0.1) for each solution in the log file. When TONYPLOT displays the log file, the value will be labelled mycarriers. You can then plot mycarriers versus terminal bias or current or other probed quantities.

Certain direction dependent quantities, such as electric field and mobility, can be probed. In these cases, specify a direction for the vector quantity using the DIR parameter.

The PROBE statement provides the only way to extract the actual values of quantities, which are calculated along the sides of each triangle in ATLAS. The PROBE statement actually stored the triangle side value closest to the probed location, while taking into account the direction for vector quantities.

Note: Specifying the probe location exactly at a material or region interface will often lead to erroneous results. It is best to slightly offset the location of the probe inside the material or region of interest.

Re-initializing ATLAS at a Given Bias Point

Each SOLVE statement will begin with the device biased at the previous value solved. To begin a solution at a previously solved bias point, re-load the structure file saved at that point. This is accomplished in the following manner:

```
LOAD INFILE=<filename> MASTER
```

Information about that solution point will be displayed in the Output Window.

This command is useful for solving a set of I/V curves. For example, to solve a family of Id / Vd (at various Vg), ramp the gate with zero drain bias. A structure file is then saved at each desired value of Vg. These structure files can be reloaded in turn while a Vd sweep is performed.

Note: An ATLAS input file cannot start with a LOAD statement. Before loading the structure file, use the MESH statement to load the device mesh for the same structure. Also, the same MODELS, MATERIAL, and CONTACT settings are required when the files are saved by ATLAS.

2.10.6: Technology Specific Issues in ATLAS

This chapter was designed to give an overview to the basic use of ATLAS without regard to the details required for a given technology. Requirements for ATLAS simulation vary considerably. The needs of the sub-micron MOS device engineer, the 1000V power device engineer, and the III-V RF device engineer differ and cannot all be covered in one chapter. Silvaco provides many references to individual technology problems using ATLAS. These are:

- A library of over 500 examples that can be accessed on-line from DECKBUILD. Look at these examples not only for their technology but also related ones. For example, different aspects of high frequency analysis is covered in the MESFET and silicon bipolar example sections.
- The chapters for each ATLAS module in this manual.
- The Simulation Standard, a newsletter distributed by Silvaco. To make sure you're on the mailing list, contact your local Silvaco office or go to www.silvaco.com.
- The Silvaco website also provides detailed information. It contains on-line versions of the articles in our newsletter, on-line searchable index of the examples, links to other TCAD web sites, and a section on solutions to known problems with all Silvaco programs.
- For more information about suggested technology specific strategies, contact your local Silvaco support engineer at support@silvaco.com.

Chapter 3: Physics

3.1: Basic Semiconductor Equations

Years of research into device physics have resulted in a mathematical model that operates on any semiconductor device [121]. This model consists of a set of fundamental equations which link together the electrostatic potential and the carrier densities, within some simulation domain. These equations, which are solved inside any general purpose device simulator, have been derived from Maxwell's laws and consist of Poisson's Equation (see Section 3.1.1: "Poisson's Equation"), the continuity equations (see Section 3.1.2: "Carrier Continuity Equations") and the transport equations (see Section 3.1.3: "The Transport Equations"). Poisson's Equation relates variations in electrostatic potential to local charge densities. The continuity and the transport equations describe the way that the electron and hole densities evolve as a result of transport processes, generation processes, and recombination processes.

This chapter describes the mathematical models implemented in ATLAS. Note that a discretization of the equations is also performed so that they can be applied to the finite element grid used to represent the simulation domain.

3.1.1: Poisson's Equation

Poisson's Equation relates the electrostatic potential to the space charge density:

$$div(\varepsilon \nabla \psi) = -\rho \qquad 3\text{-}1$$

where ψ is the electrostatic potential, ε is the local permittivity, and ρ is the local space charge density. The reference potential can be defined in various ways. For ATLAS, this is always the intrinsic Fermi potential ψ_i which is defined in the next section. The local space charge density is the sum of contributions from all mobile and fixed charges, including electrons, holes, and ionized impurities.

The electric field is obtained from the gradient of the potential (see Equation 3-2).

$$\vec{E} = -\nabla \psi \qquad 3\text{-}2$$

3.1.2: Carrier Continuity Equations

The continuity equations for electrons and holes are defined by equations:

$$\frac{\partial n}{\partial t} = \frac{1}{q} div \overrightarrow{J_n} + G_n - R_n \qquad 3\text{-}3$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} div \overrightarrow{J_p} + G_p - R_p \qquad 3\text{-}4$$

where n and p are the electron and hole concentration, $\overrightarrow{J_n}$ and $\overrightarrow{J_p}$ are the electron and hole current densities, G_n and G_p are the generation rates for electrons and holes, R_n and R_p are the recombination rates for electrons and holes, and q is the magnitude of the charge on an electron.

By default ATLAS includes both Equations 3-3 and 3-4. In some circumstances, however, it is sufficient to solve only one carrier continuity equation. The specification of which continuity equations are to be solved is performed in the METHOD statement by turning off any equation that is not to be solved. The syntax, ^ELECTRONS or ^HOLES, turns off the electron continuity equation and the hole continuity equation respectively.

3.1.3: The Transport Equations

Equations 3-1, 3-3, and 3-4 provide the general framework for device simulation. But further secondary equations are needed to specify particular physical models for: $\overrightarrow{J_n}$, $\overrightarrow{J_p}$, G_n, R_n, G_p and R_p.

The current density equations, or charge transport models, are usually obtained by applying approximations and simplifications to the Boltzmann Transport Equation. These assumptions can result in a number of different transport models such as the drift-diffusion model, the Energy Balance Transport Model or the hydrodynamic model. The choice of the charge transport model will then have a major influence on the choice of generation and recombination models.

The simplest model of charge transport that is useful is the Drift-Diffusion Model. This model has the attractive feature that it does not introduce any independent variables in addition to ψ, n and p. Until recently, the drift-diffusion model was adequate for nearly all devices that were technologically feasible. The drift-diffusion approximation, however, becomes less accurate for smaller feature sizes. More advanced energy balance and hydrodynamic models are therefore becoming popular for simulating deep submicron devices. ATLAS supplies both drift-diffusion and advanced transport models.

The charge transport models and the models for generation and recombination in ATLAS make use of some concepts associated with carrier statistics. These concepts are summarized in a further section of this chapter that deals with the carrier statistics.

Drift-Diffusion Transport Model

Derivations based upon the Boltzmann transport theory have shown that the current densities in the continuity equations may be approximated by a drift-diffusion model [142]. In this case, the current densities are expressed in terms of the quasi-Fermi levels ϕ_n and ϕ_p as:

$$\overrightarrow{J_n} = -q\mu_n n \nabla \phi_n \qquad 3\text{-}5$$

$$\overrightarrow{J_p} = -q\mu_p n \nabla \phi_p \qquad 3\text{-}6$$

where μ_n and μ_p are the electron and hole mobilities. The quasi-Fermi levels are then linked to the carrier concentrations and the potential through the two Boltzmann approximations:

$$n = n_{ie} exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \qquad 3\text{-}7$$

$$p = n_{ie} exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \qquad 3\text{-}8$$

where n_{ie} is the effective intrinsic concentration and T_L is the lattice temperature. These two equations may then be re-written to define the quasi-Fermi potentials:

$$\phi_n = \psi - \frac{kT_L}{q} ln \frac{n}{n_{ie}} \qquad 3\text{-}9$$

$$\phi_p = \psi + \frac{kT_L}{q} ln \frac{p}{n_{ie}} \qquad 3\text{-}10$$

By substituting these equations into the current density expressions, the following adapted current relationships are obtained:

$$\overrightarrow{J_n} = qD_n\nabla n - qn\mu_n\nabla\psi - \mu_n n(kT_L\nabla(ln n_{ie})) \quad 3\text{-}11$$

$$\overrightarrow{J_p} = -qD_p\nabla p - qp\mu_p\nabla\psi + \mu_p p(kT_L\nabla(ln n_{ie})) \quad 3\text{-}12$$

The final term accounts for the gradient in the effective intrinsic carrier concentration, which takes account of bandgap narrowing effects. Effective electric fields are normally defined whereby:

$$\vec{E}n = -\nabla\left(\psi + \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}13$$

$$\vec{E}p = -\nabla\left(\psi - \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}14$$

which then allows the more conventional formulation of drift-diffusion equations to be written (see Equations 3-15 and 3-16).

$$\overrightarrow{J_n} = qn\mu_n\overrightarrow{E_n} + qD_n\nabla n \quad 3\text{-}15$$

$$\overrightarrow{J_p} = qp\mu_p\overrightarrow{E_p} - qD_p\nabla p \quad 3\text{-}16$$

It should be noted that this derivation of the drift-diffusion model has tacitly assumed that the Einstein relationship holds. In the case of Boltzmann statistics this corresponds to:

$$D_n = \frac{kT_L}{q}\mu_n \quad 3\text{-}17$$

$$D_p = \frac{kT_L}{q}\mu_p \quad 3\text{-}18$$

If Fermi-Dirac statistics are assumed for electrons, Equation 3-17 becomes:

$$D_n = \frac{\left(\frac{kT_L}{q}\mu_n\right)F_{1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn} - \varepsilon_C]\right\}}{F_{-1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn} - \varepsilon_C]\right\}} \quad 3\text{-}19$$

where F_α is the Fermi-Dirac integral of order α and ε_{Fn} is given by $-q\phi_n$. An analogous expression is used for holes with Fermi-Dirac statistics.

Note: See Section 3.2.9: "Bandgap Narrowing" for more information on the effects resulting from bandgap narrowing and their implementation.

Energy Balance Transport Model

A higher order solution to the general Boltzmann Transport Equation consists of an additional coupling of the current density to the carrier temperature, or energy. The current density expressions from the drift-diffusion model are modified to include this additional physical relationship. The electron current and energy flux densities are then expressed as:

$$\overrightarrow{J_n} = qD_n \nabla n - q\mu_n n \nabla \psi + qnD_n^T \nabla T_n \quad \text{3-20}$$

$$\overrightarrow{S_n} = -K_n \nabla T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}_n T_n \quad \text{3-21}$$

$$\overrightarrow{J_p} = -qD_p \nabla p - q\mu_p p \nabla \psi - qpD_p^T \nabla T_p \quad \text{3-22}$$

$$\overrightarrow{S_p} = -K_p \nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}_p T_p \quad \text{3-23}$$

where T_n and T_p represent the electron and hole carrier temperatures and S_n and S_p are the flux of energy (or heat) from the carrier to the lattice. The energy balance transport model includes a number of very complex relationships and therefore a later section of this chapter has been devoted to this model.

3.1.4: Displacement Current Equation

For time domain simulation, the displacement current is calculated and included in the structure, log file and the run time output. The expression for displacement current is given as:

$$\overrightarrow{J_{dis}} = \varepsilon\left(\frac{\partial \overline{E}}{\partial t}\right) \quad . \quad \text{3-24}$$

3.2: Basic Theory of Carrier Statistics

3.2.1: Fermi-Dirac and Boltzmann Statistics

Electrons in thermal equilibrium at temperature T_L with a semiconductor lattice obey Fermi-Dirac statistics. That is the probability $f(\varepsilon)$ that an available electron state with energy ε is occupied by an electron is:

$$f(\varepsilon) = \frac{1}{1 + exp\left(\frac{\varepsilon - E_F}{kT_L}\right)} \quad 3\text{-}25$$

where E_F is a spatially independent reference energy known as the Fermi level and k is Boltzmann's constant.

In the limit, $\varepsilon - E_F >> kT_L$, Equation 3-25 can be approximated as:

$$f(\varepsilon) = exp\left(\frac{E_F - \varepsilon}{kT_L}\right) \quad 3\text{-}26$$

Statistics based on the use of Equation 3-26 are referred to as Boltzmann statistics [191, 78]. The use of Boltzmann statistics instead of Fermi-Dirac statistics makes subsequent calculations much simpler. The use of Boltzmann statistics is normally justified in semiconductor device theory, but Fermi-Dirac statistics are necessary to account for certain properties of very highly doped (degenerate) materials.

The Fermi-Dirac statistics have been implemented in ATLAS in a similar form to Boltzmann statistics.

The remainder of this section outlines derivations and results for the simpler case of Boltzmann statistics which are the default in ATLAS. You can have ATLAS use Fermi-Dirac statistics by specifying the `FERMIDIRAC` parameter in the `MODEL` statement.

3.2.2: Effective Density of States

Integrating the Fermi-Dirac statistics over a parabolic density of states in the conduction and valence bands, whose energy minimum is located at energies E_C and E_V respectively, yields the following expressions for the electron and hole concentrations:

$$n = N_C F_{1/2}\left(\frac{E_F - E_C}{kT_L}\right) \quad 3\text{-}27$$

$$p = N_V F_{1/2}\left(\frac{E_v - E_F}{kT_L}\right) \quad 3\text{-}28$$

where $F_{1/2}(\eta)$ is referred to as the Fermi-Dirac integral of order $1/2$. If Equation 3-26 is a good approximation, then Equations 3-27 and 3-28 can be simplified to

$$n = N_C exp\left(\frac{E_F - E_C}{kT_L}\right) \quad 3\text{-}29$$

$$p = N_V exp\left(\frac{E_v - E_F}{kT_L}\right) \quad 3\text{-}30$$

which are referred to as the Boltzmann approximation.

N_C and N_V are referred to as the effective density of states for electrons and holes and are given by:

$$N_C(T_L) = 2\left(\frac{2\pi m_e^* kT_L}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_L}{300}\right)^{\mathrm{NC\cdot F}} \mathrm{NC300} \quad 3\text{-}31$$

$$N_V(T_L) = 2\left(\frac{2\pi m_h^* kT_L}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_L}{300}\right)^{\mathrm{NV\cdot F}} \mathrm{NV300} \quad 3\text{-}32$$

where NC300 and NV300 are user-definable on the MATERIAL statement as shown in Table 3-1.

In some circumstances, the lattice temperature, T_L, is replaced by the electron temperature, T_n, in Equation 3-31 and hole temperature, T_p, in Equation 3-32.

Table 3-1. User-Definable Parameters for the Density of States

Statement	Parameter	Default	Units
MATERIAL	NC300	2.8×10^{19}	cm^{-3}
MATERIAL	NV300	1.04×10^{19}	cm^{-3}
MATERIAL	NCF	1.5	
MATERIAL	NVF	1.5	

3.2.3: Intrinsic Carrier Concentration

Multiplying Equations 3-29 and 3-30 yields:

$$np = n_{ie}^2 \quad 3\text{-}33$$

where n_{ie} is the intrinsic carrier concentration and is given for Boltzmann statistics by:

$$n_{ie} = \sqrt{N_C N_V}\exp\left(\frac{-E_g}{2kT_L}\right) \quad 3\text{-}34$$

$E_g = E_C - E_V$ is the band-gap energy.

For intrinsic (undoped) material, $p = n$. By equating Equations 3-29 and 3-30 and solving for E_F yields:

$$E_F = E_i = -q\psi_i = \frac{E_C + E_V}{2} + \left(\frac{kT_L}{2}\right)ln\left(\frac{N_v}{N_c}\right) \quad 3\text{-}35$$

where E_i is the Fermi level for intrinsic doped silicon, and ψ_i is the intrinsic potential. Equation 3-35 also defines the intrinsic potential under non-equilibrium conditions. As indicated previously, for ATLAS the ψ used in Equation 3-1 is the intrinsic potential.

The electron and hole concentrations can be expressed in terms of the intrinsic carrier concentration as:

$$n = n_{ie} exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \quad 3\text{-}36$$

$$p = n_{ie} exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \quad 3\text{-}37$$

where ψ is the intrinsic potential and ϕ is the potential corresponding to the Fermi level (i.e., $E_F = q\phi$).

The expression for intrinsic carrier concentration, n_{ie}, can be generalized to Fermi-Dirac statistics using Equations 3-27 and 3-28. Specifying the NI.FERMI parameter in the MODELS statement will cause ATLAS to calculate n_{ie} using Fermi-Dirac statistics.

3.2.4: Evaluation of Fermi-Dirac Integrals

In addition to the Fermi-Dirac integral of order $\frac{1}{2}$ as used in Equations 3-27 and 3-28, ATLAS also needs to evaluate the Fermi-Dirac integrals of order $\frac{-3}{2}, \frac{-1}{2}, \frac{3}{2}$. There are simple, quickly calculated, but relatively poor approximations available for these integrals. You can also evaluate them to machine precision; however, the amount of computation required makes the evaluations relatively slow. ATLAS uses a Rational Chebyshev approximation scheme, which is efficient in terms of CPU use and also has accuracy close to the machine precision. In the worst case, the approximation differs from the actual values by 1 part in 10^{10} and is typically much better.

3.2.5: Rules for Evaluation of Energy Bandgap

In the ATLAS simulator, there are several ways to evaluate the value of energy bandgap for a given material. To uniquely define the approach used, there is a set of rules that are followed in a heirarchical manner. The following is a description of this heirarchy:

1. For each material or material system, there may be a default approach. To determine the default for a given material or material system, refer to the descriptions given in Section 5.3: “Material Dependent Physical Models”. If the material system is not listed in this section, then look for default energy bandgap model constants in Appendix B: “Material Systems”.
2. If you set the EG300 parameter of the MATERIAL statement to a value greater than zero and you do not set the PASSLER parameter of the MODELS statement, the universal energy bandgap model described in Section 3.2.6: “The Universal Energy Bandgap Model” will be used.
3. If you set the EG300 parameter of the MATERIAL statement to a value greater than zero and you set the PASSLER parameter of the MODELS statement, then Passler's model for temperature dependent energy bandgap described in Section 3.2.7: “Passler's Model for Temperature Dependent Bandgap” will be used.
4. If you specify the K.P parameter of the MODELS statement for materials in the InAlGaN system, then the bandgap is taken as the minimum transition energy calculated following the approach given by Chuang's 3 band model for gain and radiative recombination in Section 3.9.8: “Chuang's Three Band Model for Gain and Radiative Recombination in Wurtzite Materials”.
5. If you assign the F.BANDCOMP parameter of the MATERIAL statement to a valid filename, the C interpreter function for energy bandgap will be used.
6. If you assign the EG12BOW parameter to a non-zero value, the general terniary bowing model described in Section 3.2.8: “General Ternary Bandgap Model with Bowing” will be used.

3.2.6: The Universal Energy Bandgap Model

The temperature dependence of the bandgap energy is modeled in ATLAS as follows [156]:

$$E_g(T_L) = E_g(0) - \frac{\mathrm{EGALPHA}\left(T_L^2\right)}{T_L + \mathrm{EGBETA}} = \mathrm{EG300} + \mathrm{EGALPHA}\left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}}\right] \quad \text{3-38}$$

You can specify EG300, EGALPHA and EGBETA parameters in the MATERIAL statement (see Table 3-2).

Table 3-2. User-Specifiable Parameters for Equation 3-38

Statement	Parameter	Default	Units
MATERIAL	EG300	1.08	eV
MATERIAL	EGALPHA	4.73×10^{-4}	eV/K
MATERIAL	EGBETA	636	K
MATERIAL	NC300	2.8×10^{19}	cm^{-3}
MATERIAL	NV300	1.04×10^{19}	cm^{-3}

The default values are material dependent, which can be found in Appendix B: "Material Systems". Table 3-2 displays the defaults for Silicon only.

3.2.7: Passler's Model for Temperature Dependent Bandgap

An alternative temperature dependent bandgap model by Passler [118] can be enabled by the PASSLER parameter of the MODELS statement.

The bandgap is given by

$$Eg(T) = Eg(0) - \frac{\mathrm{A.PASSLER} \cdot \mathrm{T.PASSLER}}{2} \quad \text{3-39}$$

$$\left\{\left[\left(\frac{2 \cdot T}{\mathrm{T.PASSLER}}\right)^{1/\mathrm{P.PASSLER}} + 1\right]^{\mathrm{P.PASSLER}} - 1\right\}$$

where *Eg(T)* is the lattice temperature dependent bandgap, *T* is the lattice temperature, A.PASSLER, T.PASSLER and P.PASSLER are user-specified parameters on the MATERIAL statement and *Eg(0)* is given by

$$Eg(0) = \mathrm{EG300} - \frac{\mathrm{A.PASSLER} \cdot \mathrm{T.PASSLER}}{2} \quad \text{3-40}$$

$$\left\{\left[\left(\frac{2 \cdot 300}{\mathrm{T.PASSLER}}\right)^{1/\mathrm{P.PASSLER}} + 1\right]^{\mathrm{P.PASSLER}} - 1\right\}$$

where EG300 is a user specified parameter on the MATERIAL statement.

3.2.8: General Ternary Bandgap Model with Bowing

For Ternary compound semiconductors, you can use a bandgap model that continuously and non-linearly interpolates between the bandgaps of the two binary extremes in composition. To enable this model, you must specify the EG12BOW, EG1300 and EG2300 parameters of the MATERIAL statement. EG1300 is the 300K bandgap of the binary compound at a composition fraction of *x=0*. EG2300 is the 300K bandgap of the binary compound at a composition fraction of *x=1*. EG12BOW is the bowing factor. The bandgap as a function of composition is given by Equation 3-41.

$$Eg = \texttt{EG2300}*x + \texttt{EG1300}*(1-x) - \texttt{EG12BOW}*X*(1-x) \quad \text{3-41}$$

where x is the composition fraction.

You can include the effects of temperature in the Bowing model by specifying the EG1ALPH, EG1BETA, EG2ALPH and EG2BETA parameters of the MATERIAL statement. These parameters are used in Equations 3-51 through 3-53 to calculate the bandgap.

$$Eg_1 = \texttt{EG1300} + \texttt{EG1ALPH}\left[\frac{300^2}{300 + \texttt{EG1BETA}} - \frac{T_L^2}{T_L + \texttt{EG1BETA}}\right] \quad \text{3-42}$$

$$Eg_2 = \texttt{EG2300} + \texttt{EG2ALPH}\left[\frac{300^2}{300 + \texttt{EG2BETA}} - \frac{T_L^2}{T_L + \texttt{EG2BETA}}\right] \quad \text{3-43}$$

$$Eg = Eg_2*x + Eg_1(1-x) - \texttt{EG12BOW}*X*(1-x) \quad \text{3-44}$$

3.2.9: Bandgap Narrowing

In the presence of heavy doping, greater than 10^{18}cm^{-3}, experimental work has shown that the pn product in silicon becomes doping dependent [151]. As the doping level increases, a decrease in the bandgap separation occurs, where the conduction band is lowered by approximately the same amount as the valence band is raised. In ATLAS this is simulated by a spatially varying intrinsic concentration n_{ie} defined according to Equation 3-45.

$$n_{ie}^2 = n_i^2 exp\left(\frac{\Delta Eg}{kT}\right) \quad \text{3-45}$$

Bandgap narrowing effects in ATLAS are enabled by specifying the BGN parameter of the MODELS statement. These effects may be described by an analytic expression relating the variation in bandgap, ΔE_g, to the doping concentration, N. The expression used in ATLAS is from Slotboom and de Graaf [152]:

$$\Delta E_g = \texttt{BGN.E}\left\{ ln\frac{N}{\texttt{BGN}\cdot\texttt{N}} + \left[\left(ln\frac{N}{\texttt{BGN.N}}\right)^2 + \texttt{BGN.C}\right]^{\frac{1}{2}}\right\} \quad \text{3-46}$$

You can specify the BGN.E, BGN.N, and BGN.C parameters in the MATERIAL statement. The default values from Slotboom [152] and Klaassen [88] are shown in Table 3-3. The Klaassen defaults will be used if you specify the BGN.KLA parameter in the MODELS statement. Otherwise, the Slotboom values will be used (by default). You can select a second set of defaults by specifying BGN.KLASSEN.

Table 3-3. User-Definable Parameters of Bandgap Narrowing Model

Statement	Parameter	Defaults (Slotboom)	Defaults (Klaassen)	Units
MATERIAL	BGN.E	9.0×10^{-3}	6.92×10^{-3}	V
MATERIAL	BGN.N	1.0×10^{17}	1.3×10^{17}	cm^{-3}
MATERIAL	BGN.C	0.9	0.5	—

The variation in bandgap is introduced to the other physical models by subtracting the result of Equation 3-47 from the bandgap, E_g. In addition an adjustment is also made to the electric field terms in the transport models as described earlier. The adjustment takes the form:

$$\vec{E}n = -\nabla\left(\psi + \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}47$$

$$\vec{E}p = -\nabla\left(\psi - \frac{kT_L}{q} ln n_{ie}\right) \quad 3\text{-}48$$

The variation in the energy bandgap is also applied partially to the electron affinity, χ. The effective electron affinity, χ_{eff} given as follows:

$$\chi_{eff} = \chi + \Delta E_g \times \text{ASYMMETRY} \quad 3\text{-}49$$

where ASYMMETRY is a user-specifiable asymmetry factor. You can specify the value of the asymmetry factor using the ASYMMETRY parameter of the MATERIAL statement.

Note: In addition to this in-built model for bandgap narrowing, ATLAS allows you to use its C-INTERPRETER module. You can write an external file with C-code that defines an equation for bandgap narrowing. The filename is specified with the F.BGN parameter in the MODEL statement. See Appendix A: "C-Interpreter Functions" for more information on C-INTERPRETER.

3.2.10: The Universal Bandgap Narrowing Model

A universal bandgap narrowing model has been suggested [122]. This model given in Equation 3-50 below relates the change in bandgap to the material effective masses, m_c and m_v, and the static dielectric constant ε_s. The UBGN.C and UBGN.B parameters are fitting parameters to define in the MATERIAL statement. To enable the model, specify UBGN on the MODEL statement.

$$\Delta E_g = -\mathrm{UBGN.C}\left[\frac{\varepsilon_s}{N}\left(m_0\frac{m_c+m_v}{m_c m_v}+\mathrm{UBGN.B}T^2\frac{\varepsilon_s}{N}\right)\right]^{-\frac{1}{4}} \quad \text{3-50}$$

3.3: Space Charge from Incomplete Ionization, Traps, and Defects

Poisson's Equation (Equation 3-1) including the carrier concentrations, the ionized donor and acceptor impurity concentrations N_D^+ and N_A^-, charge due to traps and defects, Q_T, has the form:

$$div(\varepsilon\nabla\psi) = q\left(n - p - N_D^+ + N_A^-\right) - Q_T \quad \text{3-51}$$

In ATLAS the default is to assume full impurity ionization (i.e., $N_D^+ = N_{D,total}$ and $N_A^- = N_{A,total}$), and to set Q_T equal to zero.

ATLAS also provides the options of accounting for incomplete ionization of impurities and accounting for additional charge associated with traps and defects.

3.3.1: Incomplete Ionization of Impurities

ATLAS can account for impurity freeze-out [77] with appropriate degeneracy factors GCB and GVB for conduction and valence bands. The ionized donor and acceptor impurity concentrations are then given by:

$$N_D^+ = \frac{N_D}{1 + \mathrm{GCB}\,exp\left(\dfrac{\varepsilon_{F_n} - (E_C - \mathrm{EDB})}{kT_L}\right)} \quad \text{3-52}$$

$$N_A^- = \frac{N_A}{1 + \mathrm{GVB}\,exp\left(\dfrac{E_V + \mathrm{EAB} - \varepsilon_{F_p}}{kT_L}\right)} \quad \text{3-53}$$

where EDB and EAB are the dopant activation energies and N_D and N_A are net compensated n-type and p-type doping, respectively. Net compensated doping is defined as follows:

If

$$N_{total} \equiv (N_{D,total} - N_{A,total}) > 0 \quad \text{3-54}$$

then

$$N_D = \mid N_{total} \mid \text{ and } N_A = 0 \quad \text{3-55}$$

Otherwise

$$N_D = 0 \text{ and } N_A = \quad \mid Ntotal \mid) \quad \text{3-56}$$

The INCOMPLETE parameter of the MODELS statement is used to select incomplete ionization and the parameters.

Table 3-4. User-Specifiable Parameters for Equations 3-52 and 3-53		
Statement	**Parameter**	**Units**
MATERIAL	GCB	
MATERIAL	EDB	eV
MATERIAL	GVB	
MATERIAL	EAB	eV

To properly handle incomplete ionization in silicon for high doping levels, a new incomplete ionization model has been added.

The models that form incomplete ionization of impurities given by Equations 3-52 and 3-53 give good physical results for low to moderately doped semiconductors. For heavily (greater than $3\times10^{18}/cm^3$) doped semiconductors, these models fail to predict experimental results of complete ionization. For silicon, an optional model has been introduced that better matches experimental results. This model is set by the IONIZ parameter of the MODELS statement.

In this model, the activation energies of the dopants in Equations 3-52 and 3-53 have been modified for doping dependence as given in the following equations:

$$\mathrm{EDB}(eV) = \mathrm{A.EDB} - \mathrm{B.EDB}N_D^{1/3} \quad \text{3-57}$$

$$\mathrm{EAB}(eV) = \mathrm{A.EAB} - \mathrm{B.EAB}N_A^{1/3} \quad \text{3-58}$$

At doping concentrations above 3×10^{18} cm^{-3}, the model predicts complete ionization. At doping concentrations between 10^{18} cm^{-3} and 3×10^{18} cm^{-3}, the model predicts a linearly interpolated value between the above expressions and complete ionization.

The default value for the parameters of Equations 3-57 and 3-58 were chosen to represent reasonable values for silicon and are given in Table 3-5.

Table 3-5. User-Specifiable Parameters for Equations 3-57 and 3-58		
Statement	**Parameter**	**Units**
MATERIAL	A.EDB	0.044
MATERIAL	B.EDB	3.6e-8
MATERIAL	A.EAB	0.0438
MATERIAL	B.EAB	3.037e-5

For general semiconductors, you can use Equations 3-57 and 3-58 without the transitions to complete ionization. To enable this model, specify INC.ION in the MODEL statement.

3.3.2: Low Temperature Simulations

In conjunction with Fermi-Dirac statistics and impurity freeze-out, ATLAS simulates device behavior under low operating temperatures. In general, simulations can be performed at temperatures as low as 50K without loss of quadratic convergence. Below this temperature, carrier and ionization statistics develop sharp transitions which cause slower convergence. Since many more iterations will probably be required if temperatures below 50K are specified, the ITLIMIT parameter of the METHOD statement should be increased.

Due to the limited exponent range on some machines, ATLAS can have trouble calculating the quasi-Fermi level of minority carriers. As the temperature decreases, the intrinsic carrier concentration also decreases. In quasi-neutral regions, the minority carrier concentration can easily underflow. Such situations were handled in the past by setting these concentrations to zero. This method does not allow an accurate subsequent calculation of minority carrier quasi-Fermi levels. In order to accurately calculate quasi-Fermi levels, majority carrier concentration and the relation, $np=n_{ie}^2$ is used to obtain minority carrier concentrations in case of an underflow. Despite these efforts, spurious glitches are occasionally observed at low temperatures in the minority quasi-Fermi levels.

3.3.3: Traps and Defects

Semiconductor materials exhibit crystal flaws, which can be caused by dangling bonds at interfaces or by the presence of impurities in the substrate. The presence of these defect centers, or traps, in semiconductor substrates may significantly influence the electrical characteristics of the device. Trap centers, whose associated energy lies in a forbidden gap, exchange charge with the conduction and valence bands through the emission and capture of electrons. The trap centers influence the density of space charge in semiconductor bulk and the recombination statistics.

Device physics has established the existence of three different mechanisms, which add to the space charge term in Poissons' equation in addition to the ionized donor and acceptor impurities. These are interface fixed charge, interface trap states and bulk trap states. Interface fixed charge is modelled as a sheet of charge at the interface and therefore is controlled by the interface boundary condition. Interface traps and bulk traps will add space charge directly into the right hand side of Poisson's equation. This section describes the definition of bulk trap states and the implementation of these bulk trap states into ATLAS for both steady state and transient conditions.

A donor-type trap can be either positive or neutral like the donor. An acceptor-type trap can be either negative or neutral like the acceptor. A donor-like trap is positively charged (ionized) when empty and neutral when filled (with an electron). An empty donor-type trap, which is positive, can capture an electron or emit a hole. A filled donor-type trap, which is neutral, can emit an electron or capture a hole. An acceptor-like trap is neutral when empty and negatively charged (ionized) when filled (with an electron). A filled acceptor-like trap can emit an electron or capture a hole. An empty acceptor-like trap can capture an electron or emit a hole. Unlike the donor, donor-like traps usually lie near the valence band. Likewise, acceptor-like traps usually lie near the conduction band.

Figure 3-1 shows the terminology used within ATLAS to define the type of trap. The position of the trap is defined relative to the conduction or valence bands using E.LEVEL so for instance, an acceptor trap at 0.4eV would be 0.4eV below the conduction band.

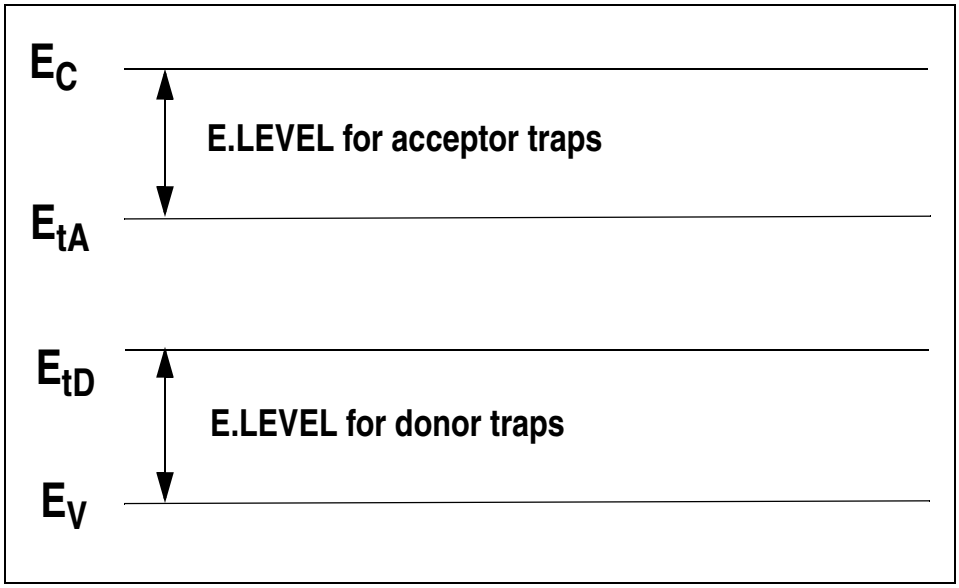


Figure 3-1: Definition of the trap energy level for acceptor and donor traps in reference to the conduction and valence band edges.

Calculation of Trapped Charge in Poisson's Equation

The total charge caused by the presence of traps is subtracted from the right hand side of Poisson's equation. The total charge value is defined by:

$$Q_T = q\left(N_{tD}^{+} - N_{tA}^{-}\right) \quad \text{3-59}$$

where N_{tD}^{+} and N_{tA}^{-} are the densities of ionized donor-like and acceptor-like traps respectively. The ionized density depends upon the trap density, DENSITY, and its probability of ionization, F_{TA} and F_{TD}. For donor-like and acceptor-like traps respectively, the ionized densities are calculated by the equations:

$$N_{tD}^{+} = \texttt{DENSITY} \times F_{TD} \quad \text{3-60}$$

$$N_{tA}^{-} = \texttt{DENSITY} \times F_{TA} \quad \text{3-61}$$

In the case where multiple traps at multiple trap energy levels are defined the total charge becomes:

$$N_{tD}^{+} = \sum_{\alpha = 1}^{k} N_{tD\alpha}^{+} \quad , N_{tA}^{-} = \sum_{\beta = 1}^{m} N_{tA\alpha}^{-} \quad \text{3-62}$$

where k is the number of donor-like traps and m is the number of acceptor-like traps.

The probability of ionization assumes that the capture cross sections are constant for all energies in a given band and follows the analysis developed by Simmons and Taylor [150]. The probability of ionization is given by the following equations for acceptor and donor-like traps.

$$F_{tA} = \frac{v_n \texttt{SIGN}\, n + e_{pA}}{v_n \texttt{SIGN}\, n + v_p \texttt{SIGP}\, p + e_{nA} + e_{pA}} \quad \text{3-63}$$

$$F_{tD} = \frac{v_p \texttt{SIGP}\, p + e_{nD}}{v_n \texttt{SIGN}\, n + v_p \texttt{SIGP}\, p + e_{nD} + e_{pD}} \quad \text{3-64}$$

where SIGN and SIGP are the carrier capture cross sections for electrons and holes respectively, v_n and v_p are the thermal velocities for electrons and holes. For donor like traps, the electron and hole emission rates, e_{nD} and e_{pD}, are defined by:

$$e_{nD} = \frac{1}{\texttt{DEGEN.FAC}}\, v_n\, \texttt{SIGN}\, n_i exp\frac{E_t - E_i}{kT_L} \quad \text{3-65}$$

$$e_{pD} = \texttt{DEGEN.FAC}\, v_p\, \texttt{SIGP}\, n_i exp\frac{E_i - E_t}{kT_L} \quad \text{3-66}$$

where E_i is the intrinsic Fermi level position, E_t is the trap energy level as defined by E.LEVEL and DEGEN.FAC is the degeneracy factor of the trap center. The latter term takes into account that spin degeneracy will exist, that is the "empty" and "filled" conditions of a defect will normally have different spin and orbital degeneracy choices. For acceptor like traps, the electron and hole emission rates, e_{nA} and e_{pA}, are defined by:

$$e_{nA} = \texttt{DEGEN.FAC}\, v_n\, \texttt{SIGN}\, n_i exp\frac{E_t - E_i}{kT_L} \quad \text{3-67}$$

$$e_{pA} = \frac{1}{\texttt{DEGEN.FAC}}\, v_p\, \texttt{SIGP}\, n_i exp\frac{E_i - E_t}{kT_L} \quad \text{3-68}$$

Table 3-6. User-Specifiable Parameters for Equations 3-60 to 3-66

Statement	Parameter	Units
TRAP	E.LEVEL	eV
TRAP	DENSITY	cm^{-3}
TRAP	DEGEN.FAC	
TRAP	SIGN	cm^2
TRAP	SIGP	cm^2

Trap Implementation into Recombination Models

To maintain self-consistency, you need to take into account that electrons are being emitted or captured by the donor and acceptor-like traps. Therefore, the concentration of carriers will be affected. This is accounted for by modifying the recombination rate in the carrier continuity equations.

The standard SRH recombination term (see "Shockley-Read-Hall (SRH) Recombination" section on page 3-79) is modified as follows:

$$R = \sum_{\alpha = 1}^{l} R_{D_\alpha} + \sum_{\beta = 1}^{m} R_{A_\beta} \quad \text{3-69}$$

where l is the number of donor-like traps, m is the number of acceptor-like traps. For donor-like traps, the function R is:

$$R_{D\alpha} = \frac{pn - n_{ie}^2}{\mathrm{TAUN}\left[p + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \mathrm{TAUP}\left[n + \frac{1}{\mathrm{DEGEN.FAC}}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-70}$$

For acceptor like traps the function R is:

$$R_{A\beta} = \frac{pn - n_{ie}^2}{\mathrm{TAUN}\left[p + \frac{1}{\mathrm{DEGEN.FAC}} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \mathrm{TAUP}\left[n + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-71}$$

The electron and hole lifetimes TAUN and TAUP are related to the carrier capture cross sections SIGN and SIGP through the equations:

$$\mathrm{TAUN} = \frac{1}{\mathrm{SIGN}\, v_n \mathrm{DENSITY}} \quad \text{3-72}$$

$$\mathrm{TAUP} = \frac{1}{\mathrm{SIGP}\, v_p\, \mathrm{DENSITY}} \quad \text{3-73}$$

The thermal velocities v_n and v_p are calculated from the following electron and hole effective masses.

$$v_n = (3kT/m_e)^{1/2} \quad \text{3-74}$$

$$v_p = (3kT/m_n)^{1/2} \quad \text{3-75}$$

To specify the effective masses directly, use the M.VTHN and M.VTHP parameters from the MATERIAL statement. If M.VTHN or M.VTHP or both are not specified, the density of states effective mass is extracted from the density of states (N_c or N_v) using Equations 3-31 and 3-32. In the case of silicon if M.VTHN or M.VTHP are not specified, the effective masses are calculated from:

$$m_e = 1.045 + 4.5 \times 10^{-4} T \quad \text{3-76}$$

$$m_n = 0.523 + 1.4 \times 10^{-3} T - 1.48 \times 10^{-6} T^2 \quad \text{3-77}$$

where T is the lattice temperature.

Table 3-7. User-Specifiable Parameters for Equations 3-70-3-73		
Statement	**Parameter**	**Units**
TRAP	TAUN	s
TRAP	TAUP	s

The TRAP statement activates the model and is used to:

- Specify the trap type DONOR or ACCEPTOR
- Specify the energy level E.LEVEL parameter
- Specify the density of the trap centers DENSITY
- Specify the degeneracy factor DEGEN.FAC
- Specify either the cross sections, SIGN and SIGP, or the electron and hole lifetimes TAUN and TAUP.

Trap-Assisted Tunneling

Trap-Assisted Tunneling models the trap-to-band phonon-assisted tunneling effects for Dirac wells. At high electric fields, tunneling of electrons from the valence band to the conduction band through trap or defect states can have an important effect on the current.

Trap-assisted tunneling is modeled by including appropriate enhancement factors [70] (Γ_n^{DIRAC} and Γ_p^{DIRAC}) in the trap lifetimes in Equation 3-70. These enhancement factors modify the lifetimes so that they include the effects of phonon-assisted tunneling on the emission of electrons and holes from a trap. This model is enabled by specifying TRAP.TUNNEL in the MODELS statement.

For donor like traps, the recombination term for traps becomes:

$$R_D = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{1+\Gamma_n^{DIRAC}}\left[p + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{1+\Gamma_p^{DIRAC}}\left[n + \frac{1}{\mathrm{DEGEN.FAC}}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-78}$$

For acceptor like traps, the recombination term becomes:

$$R_A = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{1+\Gamma_n^{DIRAC}}\left[p + \frac{1}{\mathrm{DEGEN.FAC}} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{1+\Gamma_p^{DIRAC}}\left[n + \mathrm{DEGEN.FAC}\, n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad \text{3-79}$$

The field-effect enhancement term for electrons is given by:

$$\Gamma_n^{DIRAC} = \frac{\Delta En}{kT_L}\int_0^1 exp\left(\frac{\Delta En}{kT_L}u - K_n u^{3/2}\right) du \quad \text{3-80}$$

while the field-effect enhancement term for hole is:

$$I_p^{DIRAC} = \frac{\Delta Ep}{kT_L}\int_0^1 exp\left(\frac{\Delta En}{kT_L}u - K_p u^{3/2}\right)du \quad 3\text{-}81$$

where u is the integration variable, ΔE_n is the energy range where tunneling can occur for electrons, ΔE_p is the tunneling energy range for holes, and K_n and K_p are defined as:

$$K_n = \frac{4}{3}\frac{\sqrt{2m_0 \text{MASS.TUNNEL}\Delta E_n^3}}{3q\hbar|E|} \quad 3\text{-}82$$

$$K_p = \frac{4}{3}\frac{\sqrt{2m_0 \text{MASS.TUNNEL}\Delta Ep^3}}{3q\hbar|E|} \quad 3\text{-}83$$

$\hbar$ is the reduced Planck's constant, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. You can specify MASS.TUNNEL by setting the MASS.TUNNEL parameter in the MODELS statement.

Table 3-8 shows the user-specifiable parameter for Equations 3-82 and 3-83.

Table 3-8. User-Specifiable Parameters for Equations 3-82 and 3-83

Statement	Parameter	Default	Units
MODELS	MASS.TUNNEL	0.25	

Non-local Trap Assisted Tunneling

The Trap-assisted-tunneling model in the previous section uses values of the field effect enhancement factors I_n^{DIRAC} and I_p^{DIRAC} which are approximate. They are evaluated using an approximation for the tunneling probability. It is assumed that the value at each node point depends only on the local field strength there, and that the field is assumed to be constant over the range of the tunneling. This gives Airy function solutions for the tunneling probabilities. A further approximation is also made by replacing the Airy function with the leading term of its high field asymptotic behavior. The integration over allowed tunneling energies is also evaluated approximately. A more accurate expression for I_n^{DIRAC} is

$$I_n^{DIRAC} = \frac{1}{KT}\int_0^{\Delta E_n} e^{E/kT}T(E)dE \quad 3\text{-}84$$

where *T(E)* is the probability that an electron with energy E tunnels elastically to the conduction band from the trap position.

The exponential term gives the probability that the electron will gain energy ($\Delta E_n - E$) from phonon interactions before tunneling. In Equation 3-84, the reference of energy is the conduction band edge and ΔE_n is the maximum energy below this for which the tunneling is possible, or the trap depth if less than this maximum (see Figure 3-2).

The trap energy is assumed to be at the midgap unless the TAT.NLDEPTH parameter is specified on the MODELS statement. The specified value of TAT.NLDEPTH gives the depth of the trap below the conduction band edge in eV.

$$E_T = E_c - \text{TAT.NLDEPTH} \qquad 3\text{-}85$$

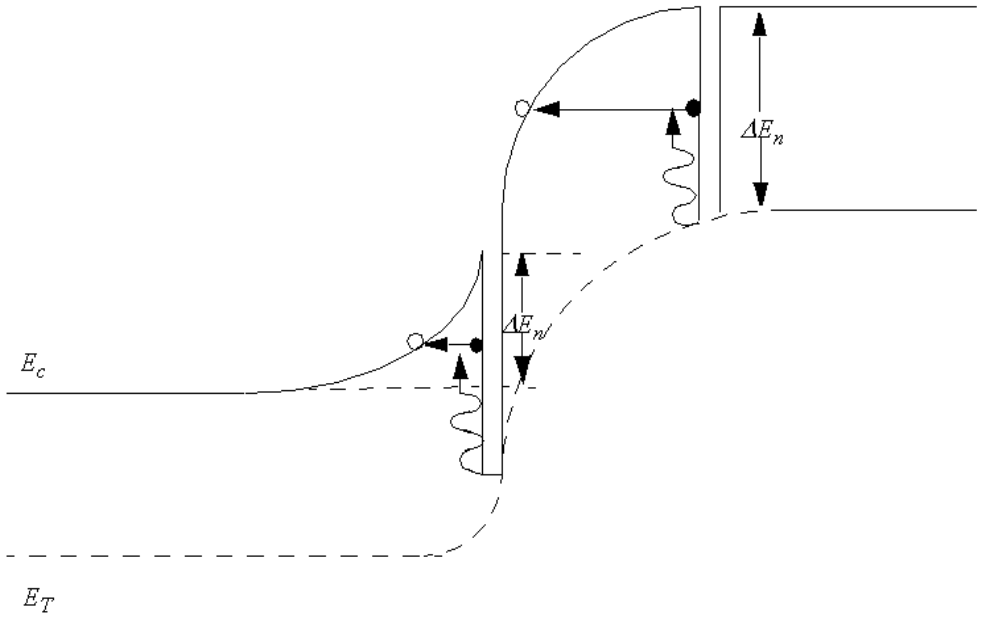


Figure 3-2: Schematic of Phonon assisted electron tunneling from a trap to the conduction band

There is an similar expression for holes. The tunneling probability is evaluated using a Wentzel-Kramers-Brillouin (WKB) method. The integral in Equation 3-84 is evaluated numerically in order to give the I_n^{DIRAC} and I_p^{DIRAC} terms.

These terms are then used in Equations 3-78 and 3-79. The values of wavevector used in the tunneling probability calculation are evaluated using the density of states effective mass. To use different masses, then use the ME.TUNNEL and MH.TUNNEL parameters on the MATERIAL statement. Using ME and MV on the MATERIAL statement will also change the masses used but may affect the behavior of other models. For example, if you specify both ME and ME.TUNNEL, the evaluation of the Γ factor will use the value of ME.TUNNEL.

To use this model, specify an extra rectangular mesh encompassing the region where the Γ terms are to be evaluated. To position the special mesh, use the QTX.MESH and QTY.MESH statement. You must then set the required direction of the quantum tunneling using the QTUNN.DIR parameter on the MODELS statement. Outside this mesh, the local model of the previous section will be used to evaluate the Γ terms.

For example

```
qtx.mesh loc=0.0 spac=0.01
qtx.mesh loc=1.0 spac=0.01

qty.mesh loc=1.0 spac=1.0
qty.mesh loc=9.0 spac=1.0
```

will set up a mesh in the rectangle bounded by x=0.0, x=1.0, y=1.0 and y=9.0. The tunneling will be assigned to be in the x-direction using the QTUNN.DIR (=1) parameter on the MODELS statement. All the required values are interpolated from the values on the ATLAS mesh. In this case, they are interpolated onto each slice in the x-direction.

To enable the model, specify TAT.NONLOCAL on the MODELS statement. You also need to specify QTUNN.DIR on the MODELS statement to determine the direction of tunneling.

Table 3-9. Non-local Trap Assisted Tunneling Parameters for the MODELS Statements

Parameter	Type	Default
TAT.NONLOCAL	Logical	False
TAT.NLDEPTH	Real	$(E_c - E_v)/2$
QTUNN.DIR	Real	0

Table 3-10. Non-local Trap Assisted Tunneling Parameters for the MATERIAL Statement

Parameter	Type	Default
ME.TUNNEL	Real	
MH.TUNNEL	Real	

Poole-Frenkel Barrier Lowering for Coulombic Wells

The Poole-Frenkel barrier lowering effect enhances the emission rate for trap-to-band phonon-assisted tunneling as well as pure thermal emissions at low electric fields. The Poole-Frenkel effect occurs when the Coulombic potential barrier is lowered due to the electric field, and only occurs when there is a Coulomb interaction between the trap and the carrier.

The following interactions are Coulombic.

- Between an empty (positive) donor-type trap and an electron.
- Between a filled (negative) acceptor-type trap and a hole.

The following interactions are short-range (Dirac).

Between a filled (neutral) donor-type trap and a hole.

Between an empty (neutral) acceptor-type trap and an electron.

The Poole-Frenkel effect is modeled by including field-effect enhancement for Coulombic wells (Γ_n^{COUL} and Γ_p^{COUL}) and thermal emission (χ_F) [100] in the trap lifetimes in Equation 3-70. The trap-assisted tunneling effects for Dirac wells remains unmodified. To enable this model, specify TRAP.COULOMBIC in the MODELS statement.

The recombination term for traps now becomes:

$$R_A = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{\chi_F + \Gamma_n^{COUL}}\left[p + \frac{1}{\mathrm{DEGEN.FAC}} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{1 + \Gamma_p^{DIRAC}}\left[n + \mathrm{DEGEN.FAC}\ n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}86$$

$$R_D = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUN}}{1 + \Gamma_n^{DIRAC}}\left[p + \mathrm{DEGEN.FAC} n_{ie} exp\left(\frac{E_i - E_t}{kT_L}\right)\right] + \frac{\mathrm{TAUP}}{\chi_F + \Gamma_p^{COUL}}\left[n + \frac{1}{\mathrm{DEGEN.FAC}}\ n_{ie} exp\left(\frac{E_t - E_i}{kT_L}\right)\right]} \quad 3\text{-}87$$

where the Poole-Frenkel thermal emission enhancement factor, χ_F, is given by:

$$\chi_F = exp\left(\frac{\Delta E_{fp}}{kT_L}\right) \quad 3\text{-}88$$

ΔE_{fp} is the barrier lowering term for a Coulombic well (see Equation 3-89).

$$\Delta E_{fp} = \sqrt{\frac{q|E|}{\pi\varepsilon}} \quad 3\text{-}89$$

The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_p^{COUL}, are defined as:

$$\Gamma_n^{COUL} = \frac{\Delta En}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} exp\left(\frac{\Delta En}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_n}\right)^{5/3}\right]\right) du \quad 3\text{-}90$$

$$\Gamma_p^{COUL} = \frac{\Delta E_p}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} exp\left(\frac{\Delta E_p}{KT_L} u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_p}\right)^{5/3}\right]\right) du \quad 3\text{-}91$$

Transient Traps

In the time domain the acceptor and donor traps do not reach equilibrium instantaneously but require time for electrons to be emitted or captured. This is taken account of inside ATLAS by solving an additional differential rate equation whenever a transient simulation is performed. These rate equations for donor and acceptor traps are given in Equations 3-92 and 3-93 [179]:

$$\frac{dN_{tD}^+}{dt} = \mathrm{DENSITY}\left\{\upsilon_p \mathrm{SIGP}\left[p(1 - F_{tD}) - F_{tD} n_i \mathrm{DEGEN.FAC}\ exp\frac{E_i - E_t}{kT}\right]\right. \quad 3\text{-}92$$
$$\left. - \upsilon_n \mathrm{SIGN}\left[nF_{tD} - \frac{(1 - F_{tD}) n_i exp}{\mathrm{DEGEN.FAC}}\ \frac{E_t - E_i}{kT}\right]\right\}$$

$$\frac{dN_{tA}^{-}}{dt} = \texttt{DENSITY}\left\{\upsilon_n \texttt{SIGN}\left[n(1-F_{tA}) - \texttt{DEGEN.FAC}\, F_{tA} n_i exp\frac{E_t - E_i}{kT}\right]\right. \quad \text{3-93}$$

$$\left. - \upsilon_p \texttt{SIGP}\left[pF_{tA} - \frac{(1-F_{tA})n_i}{\texttt{DEGEN.FAC}}\, exp\frac{E_i - E_t}{kT}\right]\right\}$$

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. If this is the case, a flag (FAST) on the TRAP statement will neglect the trap rate equation from the simulation.

3.4: The Energy Balance Transport Model

The conventional drift-diffusion model of charge transport neglects non-local transport effects such as velocity overshoot, diffusion associated with the carrier temperature and the dependence of impact ionization rates on carrier energy distributions. These phenomena can have a significant effect on the terminal properties of submicron devices. As a result ATLAS offers two non-local models of charge transport, the energy balance, and hydrodynamic models.

The Energy Balance Transport Model follows the derivation by Stratton [153,154] which is derived starting from the Boltzmann Transport Equation. By applying certain assumptions, this model decomposes into the hydrodynamic model [108,11,12].

The Energy Balance Transport Model adds continuity equations for the carrier temperatures, and treats mobilities and impact ionization coefficients as functions of the carrier temperatures rather than functions of the local electric field.

3.4.1: The Energy Balance Equations

The Energy Balance Transport Model introduces two new independent variables T_n and T_p, the carrier temperature for electrons and holes. The energy balance equations consist of an energy balance equation with the associated equations for current density and energy flux $S_{n,p}$.

For electrons the Energy Balance Transport Model consists of:

$$div\vec{S}n = \frac{1}{q}\vec{J}_n \cdot \vec{E} - W_n - \frac{3k}{2}\frac{\partial}{\partial t}\left(\lambda_n^* nT_n\right) \quad 3\text{-}94$$

$$\vec{J}n = qD_n \nabla n - q\mu_n n\nabla\psi + qnD_n^T \nabla T_n \quad 3\text{-}95$$

$$\vec{S}n = -K_n \vec{\nabla} T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}nT_n \quad 3\text{-}96$$

and for holes:

$$div\vec{S}p = \frac{1}{q}\vec{J}p \cdot \vec{E} - W_p - \frac{3k}{2}\frac{\partial}{\partial t}\left(\lambda_p^* pT_p\right) \quad 3\text{-}97$$

$$\vec{J}p = -qD_p \nabla p - q\mu_p p \nabla\psi - qpD_p^T \nabla T_p \quad 3\text{-}98$$

$$\vec{S}p = -K_p \nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}pT_p \quad 3\text{-}99$$

where $\vec{S}n$ and $\vec{S}p$ are the energy flux densities associated with electrons and holes, and μ_n and μ_p are the electron and hole mobilities.

The remaining terms, D_n and D_p, are the thermal diffusivities for electrons and holes as defined in Equations 3-100 and 3-107 respectively. W_n and W_p are the energy density loss rates for electrons and holes as defined in Equations 3-123 and 3-124 respectively. K_n and K_p are the thermal conductivities of electrons and holes as defined in Equations 3-104 and 3-111 respectively.

$$D_n = \frac{\mu_n k T_n}{q}\lambda_n^* \quad (3\text{-}100)$$

$$\lambda_n^* = \frac{F(1/2)(\eta_n)}{F-1/2(\eta_n)}, \qquad \eta_n = \frac{^{\varepsilon}F_n - \varepsilon c}{kT_n} = F_{1/2}^{-1}\left(\frac{n}{N_c}\right) \quad (3\text{-}101)$$

$$D_n^T = \left(\mu_{2n} - \frac{3}{2}\lambda_n^*\mu_n\right)\frac{k}{q} \quad (3\text{-}102)$$

$$\mu_{2n} = \mu_n\left(\frac{5}{2} + \xi_n\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)} \quad (3\text{-}103)$$

$$K_n = q\ n\ \mu_n\left(\frac{k}{q}\right)^2 \Delta_n T_n \quad (3\text{-}104)$$

$$\Delta_n = \delta_n\left[\left(\xi_n + \frac{7}{2}\right)\frac{F_{\xi_n} + 5/2(\eta_n)}{F_{\xi_n} + 3/2(\eta_n)} - \left(\xi_n + \frac{5}{2}\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)}\right] \quad (3\text{-}105)$$

$$\delta_n = \frac{\mu 2n}{\mu_n} \quad (3\text{-}106)$$

Similar expressions for holes are as follows:

$$D_p = \frac{\mu_p k T_p}{q}\lambda_p^* \quad (3\text{-}107)$$

$$\lambda_p^* = \frac{F(1/2)(\eta_p)}{F-1/2(\eta_p)}, \qquad \eta_p = \frac{^{\varepsilon}v - \varepsilon f_p}{kT_p} = F_{1/2}^{-1}\left(\frac{p}{N_V}\right) \quad (3\text{-}108)$$

$$D_p^T = \left(\mu_{2p} - \frac{3}{2}\lambda_p^*\mu_p\right)\frac{k}{q} \quad (3\text{-}109)$$

$$\mu_{2p} = \mu_p\left(\frac{5}{2} + \xi_p\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)} \quad (3\text{-}110)$$

$$K_p = q\ p\ \mu_p\left(\frac{k}{q}\right)^2 \Delta_p T_p \quad (3\text{-}111)$$

$$\Delta_p = \delta_p\left[\left(\xi_p + \frac{7}{2}\right)\frac{F_{\xi_p} + 5/2(\eta_p)}{F_{\xi_p} + 3/2(\eta_p)} - \left(\xi_p + \frac{5}{2}\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)}\right] \quad (3\text{-}112)$$

$$\delta_p = \frac{\mu 2p}{\mu_p} \quad (3\text{-}113)$$

If Boltzmann statistics are used in preference to Fermi statistics, the above equations simplify to:

$$\lambda_n^* = \lambda_p^* = 1 \quad \text{3-114}$$

$$\Delta_n = \delta_n = \left(\frac{5}{2} + \xi_n\right) \quad \text{3-115}$$

$$\Delta_p = \delta_p = \left(\frac{5}{2} + \xi_p\right) \quad \text{3-116}$$

$$\xi_n = \frac{d(ln\,\mu_n)}{d(ln\,T_n)} = \frac{T_n}{\mu_n}\frac{\partial \mu_n}{\partial T_n} \quad \text{3-117}$$

$$\xi_p = \frac{d(ln\,\mu_p)}{d(ln\,T_p)} = \frac{T_p}{\mu_p}\frac{\partial \mu_p}{\partial T_p} \quad \text{3-118}$$

The parameters ξ_n and ξ_p are dependent on the carrier temperatures. Different assumptions concerning ξ_n and ξ_p correspond to different non-local models. In the high-field saturated-velocity limit, that corresponds to velocity saturation, the carrier mobilities are inversely proportional to the carrier temperatures.

$$\xi_n = \xi_p = -1 \quad \text{3-119}$$

and this corresponds to the Energy Balance Transport Model.

If instead the choice ξ_n=ξ_p=0 was chosen this would correspond to the simplified Hydrodynamic Model. The parameters, ξ_n and ξ_p, can be specified using the KSN and KSP parameters on the MODELS statement.

Boundary conditions for n, p, and ψ are the same as for the drift diffusion model. Energy balance equations are solved only in the semiconductor region. Electron and hole temperatures are set equal to the lattice temperature on the contacts. On the other part of the boundary, the normal components of the energy fluxes vanish.

Hot carrier transport equations are activated by the MODELS statement parameters: HCTE.EL (electron temperature), HCTE.HO (hole temperature), and HCTE (both carrier temperatures).

3.4.2: Density of States

The calculation of the effective density of states is modified for the Energy Balance Transport Model. The electron and hole temperatures replace the lattice temperature in Equations 3-31 and 3-32. For example:

$$N_c = \left(\frac{2\pi m_e^* kT_n}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_n}{300}\right)^{\frac{3}{2}} NC(300) \quad \text{3-120}$$

$$N_v = \left(\frac{2\pi m_n^* kT_p}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_p}{300}\right)^{\frac{3}{2}} NV(300) \quad \text{3-121}$$

3.4.3: Energy Density Loss Rates

The energy density loss rates define physical mechanisms by which carriers exchange energy with the surrounding lattice environment. These mechanisms include carrier heating with increasing lattice temperature as well as energy exchange through recombination processes (SRH and Auger) and generation processes (impact ionization). If the net generation-recombination rate is written in the form:

$$U = R_{srh} + R_n^A + R_p^A - G_n - G_p \quad \text{3-122}$$

where R_{srh} is the SRH recombination rate, R_n^A are R_p^A Auger recombination rates related to electrons and holes, G_n and G_p are impact ionization rates, then the energy density loss rates in Equations 3-94 and 3-97 can be written in the following form:

$$W_n = \frac{3}{2} n \frac{k(T_n - T_L)}{\texttt{TAUREL.EL}} \lambda_n + \frac{3}{2} k T_n \lambda_n R_{SRH} + E_g \left(G_n - R_n^A \right) \quad \text{3-123}$$

$$W_p = \frac{3}{2} p \frac{k(T_p - T_L)}{\texttt{TAUREL.HO}} \lambda_p + \frac{3}{2} k \ T_p \ \lambda_p R_{SRH} + E_g \left(G_p - R_p^A \right) \quad \text{3-124}$$

where

$$\lambda_n = F_{\frac{3}{2}}(\eta_n) / F_{\frac{1}{2}}(\eta_n) \quad \text{3-125}$$

$$\lambda_p = F_{\frac{3}{2}}(\eta_p) / F_{\frac{1}{2}}(\eta_p) \quad \text{3-126}$$

(and are equal to 1 for Boltzmann statistics), TAUREL.EL and TAUREL.HO are the electron and hole energy relaxation times, E_g is the bandgap energy of the semiconductor. The relaxation parameters are user-definable on the MATERIAL statement, which have their defaults shown in Table 3-11.

The relaxation times are extremely important as they determine the time constant for the rate of energy exchange and therefore precise values are required if the model is to be accurate. But, this is an unmeasurable parameter and Monte Carlo analysis is the favored method through which values can be extracted for the relaxation time.

It's also important to take into consideration that different materials will have different values for the energy relaxation time but within ATLAS, the relaxation time will always default to the value for silicon.

Table 3-11. User-Specifiable Parameters for Equations 3-123 and 3-124

Statement	Parameter	Default	Units
MATERIAL	TAUREL.EL	2.5×10^{-13}	s
MATERIAL	TAUREL.HO	2.5×10^{-13}	s

3.4.4: Temperature Dependence of Relaxation Times

ATLAS doesn't provide an explicit default model for the temperature dependence of energy relaxation times. Two methods exist, however, to make the relaxation time a function of carrier energy.

For the first method, use a built-in model by specifying the TRE.T1, TRE.T2, TRE.T3, TRE.W1, TRE.W2, and TRE.W3 parameters in the MATERIAL statement. Then, activate the electron temperature (electron energy) dependent energy relaxation time by using E.TAUR.VAR in the MODELS statement. Electron energy relaxation time will then be:

$$\tau_e = \begin{cases} \texttt{TRE.T1}, W < \texttt{TRE.W1} \\ \texttt{TRE.T2}, W = \texttt{TRE.W2} \\ \texttt{TRE.T3}, W > \texttt{TRE.W3} \end{cases} \quad \text{3-127}$$

where:

$$W = \frac{3}{2} k T_n \quad \text{3-128}$$

For TRE.W1<W<TRE.W2 the energy relaxation time varies quadratically between TRE.T1 and TRE.T2. For TRE.W2<W<TRE.W3 energy relaxation time varies quadratically between TRE.T2 and TRE.T3. The corresponding parameter for hole energy relaxation time in the MODELS statement is H.TAUR.VAR. Other parameters are listed in Table 3-12.

Table 3-12. User- Specifiable Parameters for Variable Energy Relaxation Time.

Statement	Parameter	Default	Units
MATERIAL	TRE.T1	8×10^{-13}	s
MATERIAL	TRE.T2	1.92×10^{-12}	s
MATERIAL	TRE.T3	1×10^{-12}	s
MATERIAL	TRE.W1	0.06	eV
MATERIAL	TRE.W2	0.3	eV
MATERIAL	TRE.W3	0.45	eV
MATERIAL	TRH.T1	1×10^{-12}	s
MATERIAL	TRH.T2	1×10^{-12}	s
MATERIAL	TRH.T3	1×10^{-12}	s
MATERIAL	TRH.W1	1×10^{10}	eV
MATERIAL	TRH.W2	1×10^{10}	eV
MATERIAL	TRH.W3	1×10^{10}	eV

For the second method, use the C-INTERPRETER to apply a user-defined model for energy relaxation time as a function of carrier energy. In the MODELS statement, assign the F.TAURN and F.TAURP parameters with the names of external files that contain the user-defined C-INTERPRETER function for the energy relaxation time. You should also specify E.TAUR.VAR and H.TAUR.VAR flags in the MODELS statement when using C-INTERPRETER functions for the energy relaxation times.

3.4.5: Energy Dependent Mobilities

The Energy Balance Transport Model requires the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated, which causes the carriers in a homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields, $E_{eff,n}$ and $E_{eff,p}$, are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\text{TAUMOB.EL}} \quad \text{3-129}$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\text{TAUMOB.HO}} \quad \text{3-130}$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived from the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introduced into the relevant field dependent mobility model. A full description of the available models is given in Section 3.6.1: “Mobility Modeling”.

ATLAS2D provides a general C-interpreter function allowing you to specify carrier mobility as a function of Perpendicular field, carrier temperature, Lattice temperature, carrier concentration and donor and acceptor concentrations.

For electron mobility, the parameter is F.ENMUN, which can be set in either the MATERIAL or MOBILITY statement. The corresponding parameter for hole mobility is F.ENMUP, which can be set in either the MATERIAL or MOBILITY statement. The respective C-functions are endepmun() and endepmup(). The examples of these functions are provided in ATLAS.

3.5: Boundary Physics

ATLAS supports several boundary conditions: Ohmic contacts, Schottky contacts, insulated contacts, and Neumann (reflective) boundaries. Voltage boundary conditions are normally specified at contacts. Current boundary conditions can also be specified. Additional boundary conditions have been implemented to address the needs of specific applications. You can connect lumped elements between applied biases and semiconductor device contacts. A true distributed contact resistance is included to account for the finite resistivity of semiconductor contacts.

3.5.1: Ohmic Contacts

Ohmic contacts are implemented as simple Dirichlet boundary conditions, where surface potential, electron concentration, and hole concentrations (ψ_s, n_s, p_s) are fixed. Minority and majority carrier quasi-Fermi potentials are equal to the applied bias of the electrode (i.e., ϕ_n=ϕ_p=$V_{applied}$). The potential ψ_s is fixed at a value that is consistent with space charge neutrality. For example:

$$n_s + N_A^- = p_s + N_D^+ \quad \text{3-131}$$

Equation 3-131 can be solved for ψ_s, n_s, and p_s, since ϕ_n and ϕ_p are known. If Boltzmann statistics are used, the substitution of Equations 3-36 and 3-37 into Equation 3-131 will yield:

$$n_s = \frac{1}{2}\left[\left(N_D^+ - N_A^-\right) + \sqrt{\left(N_D^+ - N_A^-\right)^2 + 4n_{ie}^2}\right] \quad \text{3-132}$$

$$p_s = \frac{n_{ie}^2}{n_s} \quad \text{3-133}$$

$$\psi_s = \phi_n + \frac{kT_L}{q} ln\frac{n_s}{n_{ie}} = \phi_p - \frac{kT_L}{q} ln\frac{p_s}{n_{ie}} \quad \text{3-134}$$

Note: If you don't specify a work function, the contacts will be Ohmic regardless of its material.

3.5.2: Schottky Contacts

To specify a Schottky contact [129], specify a workfunction using the WORKFUN parameter of the CONTACT statement. The surface potential of the Schottky contact is given by:

$$\psi_s = \text{AFFINITY} + \frac{E_g}{2q} + \frac{kT_L}{2q} ln\frac{N_C}{N_V} - \text{WORKFUN} + V_{applied} \quad \text{3-135}$$

where AFFINITY is the electron affinity of the semiconductor material, E_g is the bandgap, N_C is the conduction band density of states, N_V is the valence band density of states, and T_L is the ambient temperature. In practice, the workfunction is defined as:

$$\text{WORKFUN} = \text{AFFINITY} + \phi_B \quad \text{3-136}$$

where ϕ_B is the barrier height at the metal-semiconductor interface in eV.

Table 3-13. User-Specifiable Parameters for Equation 3-136		
Statement	**Parameter**	**Units**
CONTACT	WORKFUN	eV
MATERIAL	AFFINITY	eV

For example, if the Schottky contact were aluminum with a workfunction difference to the silicon of 4.2 eV and a barrier height of 0.7eV, then you would define the Schottky contact with the statement:

```
CONTACT NAME=GATE WORKFUN=4.9
```

Note: You can also define the workfunction using a C-Interpreter function. This function is referenced by the F.WORKF parameter of the CONTACT statement. The function defines workfunction as a function of electric field.

You can enable the surface recombination model by specifying any of the following parameters of the CONTACT statement: SURF.REC, E.TUNNEL, VSURFN, VSURFP, or BARRIERL. In this case, the quasi-Fermi levels, ϕ_n and ϕ_p, are no longer equal to $V_{applied}$. Instead, these parameters are defined by a current boundary conditions at the surface [42]:

$$J_{sn} = q\mathrm{VSURFN}(n_s - n_{eq})exp\left(\frac{\Delta\phi_b}{kT}\right) \quad \text{3-137}$$

$$J_{sp} = q\mathrm{VSURFP}(p_s - p_{eq})exp\left(\frac{\Delta\phi_b}{kT}\right) \quad \text{3-138}$$

Table 3-14. User-Specifiable Parameters for Equations 3-137 to 3-138		
Statement	**Parameter**	**Units**
CONTACT	VSURFN	cm/s
CONTACT	VSURFP	cm/s

where J_{sn} and J_{sp} are the electron and hole currents at the contact, n_s, is the surface electron concentration and p_s is the surface hole concentrations. The terms, n_{eq} and p_{eq}, are the equilibrium electron and hole concentrations assuming infinite surface recombination velocity ($\phi_n = \phi_p = V_{applied}$). If VSURFN and VSURFP are not specified on the CONTACT statement, their values will be calculated using:

$$\mathrm{VSURFN} = \frac{\mathrm{ARICHN}\ T_L^{\,2}}{q\ N_C} \quad \text{3-139}$$

$$\mathrm{VSURNP} = \frac{\mathrm{ARICHP}\ T_L^{\,2}}{q\ N_V} \quad \text{3-140}$$

Table 3-15. User-Specifiable Parameters for Equations 3-139 to 3-140			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	ARICHN	110	$A/cm^2/K^2$
MATERIAL	ARICHP	30	$A/cm^2/K^2$

Here, ARICHN and ARICHP are the effective Richardson constants for electrons and holes, taking account of quantum mechanical reflections and tunneling, N_C and N_V are the conduction and valence band density of states. The ARICHN and ARICHP parameters are user-definable as shown in Table 3-15 and N_C and N_V are functions of the lattice temperature, T_L, according to Equations 3-31 and 3-32.

The Surface Recombination Schottky Model also accounts for field-dependent barrier-lowering mechanisms. These mechanisms are caused by image forces and possible static dipole layers at the metal-semiconductor interface [156]. If the barrier height is defined as:

$$\phi_{bn} = \text{WORKFUN} - \text{AFFINITY} \quad 3\text{-}141$$

$$\phi_{bp} = \text{AFFINITY} + \frac{E_g}{q} - \text{WORKFUN} \quad 3\text{-}142$$

With barrier lowering, the amount of energy by which these barrier heights are lowered is defined by:

$$\Delta\phi_b = \text{BETA}\left[\frac{q}{4\pi\varepsilon_s}\right]^{\frac{1}{2}} E^{1/2} + \text{ALPHA} \times E^{\text{GAMMA}} \quad 3\text{-}143$$

Table 3-16. User-Specifiable Parameters for Equation 3-143			
Statement	**Parameter**	**Default**	**Units**
CONTACT	ALPHA	0.0	cm
CONTACT	BETA	1.0	
CONTACT	GAMMA	1.0	

Here, E is the magnitude of the electric field at the interface and ALPHA is the linear, dipole barrier lowering coefficient. The Barrier Lowering Model can be turned on with the BARRIER parameter in the CONTACT statement. Typical values of ALPHA may be found in [9]. Note that the term with the square root dependence on electric field corresponds to the image force, while the linear term corresponds to the Dipole Effect [156].

Surface recombination is implemented on a triangle-by-triangle basis, which means using the surface recombination velocity and geometrical data. A recombination component is then calculated for each triangle so that an element of interest is connected. Using the electric field for each triangle, an adjusted recombination term can be computed if barrier lowering is incorporated. This is in contrast to where a single field value for the electrode node is used to compute total recombination value [135].

You can take electron tunneling through the barrier into account by specifying the E.TUNNEL parameter in the CONTACT statement. The electron tunneling current (J_{tn}) is given by the Tsu-esaki Model (Equation 3-144) and is applied to the contact as a current boundary condition.

$$J_{tn} = -\frac{4\,\pi\,q\,\mathrm{ME.TUNNEL}\,m_0\,k\,T_L}{h^3}\int_0^{\phi_{bn}} P(E_Z)N(E_Z)dEz \qquad 3\text{-}144$$

Here, ϕ_{bn} is the barrier height, ME.TUNNEL is the relative effective mass for electrons, m_0 is the electron rest mass, $\hbar$ is Planck's constant, k is Boltzmann's constant, q is the electron charge, and $N(E_Z)$ is given by:

$$N(E_Z) = ln\left(\frac{1+e^{\left(\frac{E_F-E_Z}{kT_L}\right)}}{1+e^{\left(\frac{E_F-E_Z-qV}{kT_L}\right)}}\right) \qquad 3\text{-}145$$

The transmission probability $P(E_Z)$ is given by the *WKB* [156] approximation for a triangular barrier and is expressed as:

$$P(E_Z) = exp\left(\frac{-8\pi(2\mathrm{ME.TUNNEL}\,m^*)^{\frac{1}{2}}(\phi_{bn}-E_Z)^{\frac{3}{2}}}{3qhE}\right) \qquad 3\text{-}146$$

The ME.TUNNEL parameter is user-definable in the CONTACT statement.

For example, to specify a Thermoinic Emission Schottky Contact with tunneling and barrier lowering, the command would be:

```
CONTACT NAME=GATE SURF.REC E.TUNNEL BARRIER
```

Note: At room temperature, both methods for Schottky contacts should yield similiar results. If the ambient temperature changes, then you should apply the Thermoinic Emission Model. Also, when applying either Schottky contact model, we recommend that you place a fine grid underneath the contact to model the depletion region and that you specify FERMI statistics in the MODELS statement.

Note: Tunneling is implemented as an addition to Equations 3-137 and 3-138.

Parabolic Field Emission Model

For wide bandgap materials such as SiC the parabolic field emission model is suggested [41,52]. Equation 3-147 describes the current as a function of applied bias voltage *V*. Equation 3-148 describes the tunneling probability *T(E)*.

$$J = \frac{A^*T}{k_B}\int_0^{\infty} T(E)\,ln\left[\frac{1 + exp\left(\frac{-E-E_n}{k_BT}\right)}{1 + exp\left(\frac{-E-qV-E_n}{k_BT}\right)}\right] \quad 3\text{-}147$$

$$T(E) = \begin{cases} exp\,\frac{2}{hq}\sqrt{\frac{me}{N}}\left[E\,ln\left(\frac{\sqrt{E_b^*} + \sqrt{E_b^* - E}}{\sqrt{E}}\right) - \sqrt{E_b^*}\,\sqrt{E_b^* - E_b}\right] & for\ (E < E_b^*) \\ 1 & for\ (E > E_b^*) \end{cases} \quad 3\text{-}148$$

Here A^* is Richardson's constant, E_b is the barrier height on the metalside, N is the semiconductor doping concentration, $En=Ec-Ef=kT^*ln(Nc/N)$ is the quasi fermi level energy $Vbi=(Eb-En)/q$ is the built in voltage , and $Eb^* = q(Vbi-V) = Eb-En-qV$ is the barrier height on the semiconductor side.

Richardson's constant A^* is given by $\frac{4\pi mqk^2}{h^3}$ where m is the effective mass.

The effective mass is generally taken from the density of states. You can, however, specify a mass for this tunneling calculation only by the ME.TUNNEL and MH.TUNNEL parameters of the MATERIAL statement. To enable the model, use the PARABOLIC parameter of the CONTACT statement. To turn on the tunneling components independently for electrons and holes, specify the E.TUNNEL and H.TUNNEL parameters of the CONTACT statement. To enable the model, specify the NSURF.REC or PSURF.REC parameters of the CONTACT statement.

Universal Schottky Tunneling (UST) Model

In the Universal Schottky Tunneling (UST) Model [74, 105], the tunneling current is represented by localized tunneling rates at grid locations near the Schottky contact. For electrons, this is illustrated in Figure 3-3. The tunneling component of current can be described by Equation 3-149.

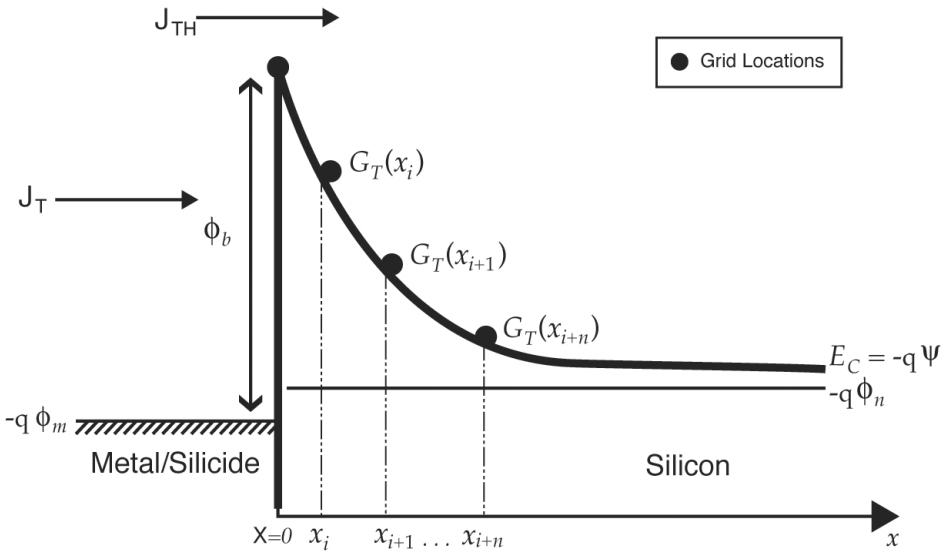


Figure 3-3: Local tunneling generation rate representation of the universal Schottky tunneling model

$$J_T = \frac{A^*T}{k}\int_E^{\infty} \Gamma(E')\ln\left[\frac{1+fs(E')}{1+fm(E')}\right]dE' \qquad 3\text{-}149$$

Here, J_T is the tunneling current density, A^* is the effective Richardson's coefficient, T is the temperature, $\Gamma(E)$ is the tunneling probability, $fs(E)$ and $fm(E)$ are the Maxwell-Boltzmann distribution functions in the semiconductor and metal and E is the carrier energy.

You can then apply the transformation [74] given in Equation 3-150 to obtain the localized tunneling rates, G_T.

$$G_T = \frac{1}{q}\nabla J_T \qquad 3\text{-}150$$

Applying the transformation given in Equations 3-149 and 3-150, you can obtain the expression given in Equation 3-151.

$$G_T = \frac{A^*T\vec{E}}{k}\Gamma(x)\ln\left[\frac{1+n/\gamma_n N_c}{1+\exp[-(E_c-E_{FM})/kT]}\right] \qquad 3\text{-}151$$

Here, $\vec{E}$ is the local electric field, n is the local electron concentration, N_c is the local conduction band density of states, γ_n is the local Fermi-Dirac factor, E_c is the local conduction band edge energy and E_{FM} is the Fermi level in the contact.

The tunneling probability $\Gamma(x)$ can be described by Equation 3-152.

$$\Gamma(x) = \exp\left[\frac{-2\sqrt{2m}}{\hbar}\int_0^x (E_c(x') - E_c(x))dx'\right] \qquad 3\text{-}152$$

In Equation 3-152, m is the electron effective mass for tunneling, and $E_c(x)$ is the conduction band edge energy as a function of position.

Assuming linear variation of E_c around a grid location, Equation 3-152 can be reduced to Equation 3-153.

$$\Gamma(x) = exp\left[\frac{-4\sqrt{2m}x}{3h}(E_{FM} + q\phi b - E_c(x))\right]^{1/2} \quad \text{3-153}$$

In Equation 3-153, ϕb is the barrier height. Similar expressions to Equations 3-149 through 3-153 exist for holes.

To enable the universal Schottky tunneling model you should specify UST on the MODELS statement. You are also required to specify SURF.REC on the CONTACT statement for each contact that you wish the model to apply. Once enabled, this model automatically applies to both electrons and holes. For a given grid point, however, the model will only apply to one carrier according to the relative direction of the local electric field.

Once you enable the model, for each electrode with thermionic emission, the model will be applied to all grid points within a specified distance of the electrode, To specify this distance, use D.TUNNEL parameter of the MATERIAL statement.

3.5.3: Floating Contacts

A contact that isn't connected to a current or voltage source and is totally insulated by dielectric is called a floating contact. ATLAS accounts for floating contacts such as floating gates in EPROM devices by using a distributed charge boundary condition, which is applied to all nodes of the floating electrode (see Equation 3-154).

$$\int_s D\, dS = Q_{FG} \quad \text{3-154}$$

where D is the electric displacement vector, s represents the external surface of the floating gate, and Q_{FG} is the injected charge.

ATLAS performs an integration over the entire surface of the electrode and forces the potential on the floating nodes to produce the correct total charge on the electrode. The total charge when performing a simulation is by default zero but can be defined using the SOLVE statement. The total charge may change if you activate a Charge Injection Model. For more information about this model, see Section 3.6.6: "Gate Current Models". To define a contact as a floating contact, use:

```
CONTACT NAME=fgate FLOATING
```

Note: When specifying a floating contact, use the Newton scheme as the numerical technique. See Chapter 18: "Numerical Techniques", Section 18.5.1: "Newton Iteration".

3.5.4: Current Boundary Conditions

In some devices, the terminal current is a multi-valued function of the applied voltage. This means that for some voltage boundary conditions, the solution that is obtained depends on the initial guess. An example of this is the CMOS latch-up trigger point. At the trigger point the I-V curve changes from being flat to vertical and may exhibit a negative slope. The solution will then have three different solutions of current for one applied bias. The particular solution which the model finishes in will depend upon the initial conditions.

This trigger point is difficult to determine using a simple voltage boundary condition. In addition, it is almost impossible to compute any solutions in the negative resistance regime when using voltage boundary conditions. Some of these problems can be overcome using current boundary conditions.

Calculation of current boundary conditions is activated by the CURRENT parameter in the CONTACT statement.

The voltage boundary condition should be used in regions where dI/dV is small. The current boundary condition may be preferable for operating regimes where dI/dV is large. It is common for the negative resistance regime of a device to have a slope dI/dV very close to 0. Such behavior should be considered when using a current source to trace out an entire I-V curve. In these cases, use the CURVETRACE option in ATLAS.

Note: When a current boundary condition has been specified, choose the Newton numerical scheme on the METHOD statement. You can perform ac small signal analysis with current boundary conditions.

3.5.5: Insulating Contacts

Insulating contacts are contacts that are completely surrounded by insulator. They may be connected to a voltage source (tied contact) or they may be floating.

Insulating contacts that are connected to a voltage source generally have a work function that dictates a value for ψ_s similar to that given by Equation 3-135. Electron and hole concentrations within the insulator and at the insulating contact are forced to be zero (i.e., n_s=p_s=0).

3.5.6: Neumann Boundaries

Along the outer (non-contact) edges of devices, homogeneous (reflecting) Neumann boundary conditions are imposed so that current only flows out of the device through the contacts. In the absence of surface charge along such edges, the normal electric field component becomes zero. Current isn't permitted to flow from the semiconductor into an insulating region except via oxide tunneling models. At the interface between two different materials, the difference between the normal components of the respective electric displacements must be equal to any surface charge according to:

$$\hat{n} \cdot \varepsilon_1 \nabla \psi_1 - \hat{n} \cdot \varepsilon_2 \nabla \psi_2 = \rho_s \qquad \text{3-155}$$

where n is the unit normal vector, ε_1 and ε_2 are the permittivities of the materials on either side of the interface and ρ_s is the sheet charge at the interface.

3.5.7: Lumped Element Boundaries

ATLAS supports some simple configurations of lumped elements. These are indicated in Figure 3-4.

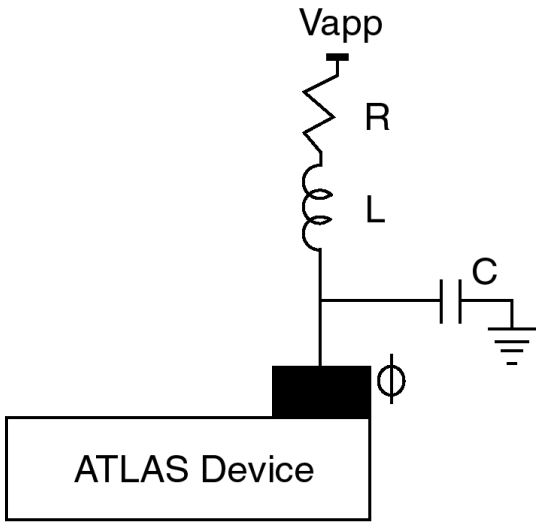


Figure 3-4: The Lumped elements supported by ATLAS

Lumped elements can be extremely useful when simulating CMOS or MOSFET structures. For example, the p-tub contact in the CMOS cross-section might be tens or hundreds of microns away from the active area of an embedded vertical npn bipolar transistor, which may only be 10-20 μm on a side. If the whole structure were simulated, a tremendous number of grid points (probably more than half) are essentially wasted in accounting for a purely resistive region of the device.

In the case of a MOSFET substrate, you would not want to include grid points all the way to the back side of a wafer. In either of these cases, a simple lumped resistance can be substituted [127].

Table 3-17 shows the user-specifiable parameters for Figure 3-4.

Table 3-17. User-Specifiable Parameters for Figure 3-2.

Symbol	Statement	Parameter	Units
C	CONTACT	CAPACITANCE	F/μm
R	CONTACT	RESISTANCE	Ωμm
L	CONTACT	INDUCTANCE	Hμm

Resistance is specified in $\Omega\mu$m, capacitance is specified in F/μm, and inductance is specified in Hμm.

The following combinations are possible: resistance only, resistance and capacitance, inductance and resistance and inductance, capacitance and resistance. For the case of inductance or capacitance only, ATLAS adds a small resistance as well. For more complicated circuit configurations, use the MIXEDMODE module of ATLAS.

Note: Capacitance increases with device width (into the z-plane) while resistance decreases. Except for the case of extremely large resistances where the arrangement becomes similar to a pure current source, no convergence degradation has been observed for a lumped element boundary in comparison to a simple Ohmic contact. Transient simulation therefore becomes easier and is more well-defined.

You should use the simulator to calculate any resistance (or capacitance) components that might be included as lumped elements. When performing CMOS simulations, you could simulate just the p-tub with Ohmic contacts at either end. From the plot of terminal current (in A/μm) versus voltage, resistance can be directly extracted from the slope. Be very careful to consider any three-dimensional effects (e.g., current spreading) before using a resistance value in further simulations.

When looking at the results of simulation with lumped elements it's important to distinguish between the applied voltage (V_{app}) and the internal bias (ϕ) in the log file produced by ATLAS.

Note: AC small signal analysis can't be performed when any lumped elements have been specified. AC small signal analysis, however, can be performed when you specify the resistance in the LOG statement. Also, when lumped elements have been defined, use the Newton numerical scheme in the METHOD statement.

3.5.8: Distributed Contact Resistance

Since contact materials have finite resistivities, the electrostatic potential isn't always uniform along the metal-semiconductor surface. To account for this effect, a distributed contact resistance can be associated with any electrode. This is in contrast to the lumped contact resistance described in the previous section.

ATLAS implements this distributed contact resistance by assigning a resistance value to each surface element associated with the contact. If each surface element is a different length, a different value for the contact resistance will be assigned. ATLAS calculates a resistance value of R_i for each surface element from the value of CON.RESIST, as specified on the CONTACT statement. The units of CON.RESIST are Ωcm^2 and R_i calculated as:

$$R_i = \frac{\text{CON.RESIST}}{d_i \ \text{WIDTH}} \qquad \text{3-156}$$

where R_i is the resistance at node i, P_c is specified by the CON.RESIST parameter, d_i is the length of the contact surface segment associated with node i and WIDTH is the width of the device. The effect of the resistance, R_i is to add an extra equation to be satisfied to node i. This equation is given by:

$$\frac{1}{R_i}\left[V_{applied} - \left(\psi_i \pm \frac{kT_L}{q} ln(N/n_{ie})\right)\right] - (I_n + I_p + I_{disp}) = 0 \qquad \text{3-157}$$

where $V_{applied}$ is the external applied voltage, Ψ_i is the surface potential, N is the net doping, n_i is the intrinsic electron concentration, and I_n, I_p, I_{disp} are the electron, hole and displacement currents at node i. This equation simply balances the current in and out of the resistor added to each i node.

As with the case for lumped elements, ATLAS can print out a value of contact resistance for each contact in the run time output. Since the actual value depends on the length of each surface segment for distributed contacts, ATLAS prints out the value of CON.RESIST/WIDTH which is the same for all contact surface segments. This runtime output is enabled by adding the PRINT option on the MODELS statement.

Table 3-18. User-Specifiable Parameters for Equation 3-156		
Statement	**Parameter**	**Units**
CONTACT	CON.RESIST	Ω-cm^2
MESH	WIDTH	μm

Note: AC small signal analysis cannot be performed when any distributed contact resistance has been specified. Also, only the NEWTON numerical scheme should be chosen on the METHOD statement.

3.5.9: Energy Balance Boundary Conditions

When the Energy Balance Transport Model is applied, special boundary conditions are applied for carrier temperatures. By default at the contacts, Dirichlet boundary conditions are used for carrier temperatures:

$$T_n = T_p = T_L \qquad \text{3-158}$$

You can treat contacts as Neumann (reflective) boundaries with respect to carrier temperature by specifying REFLECT on the CONTACT statement.

Elsewhere on the boundary, the normal components of the energy fluxes vanish. The boundary conditions for (ψ, n, p) are the same as for the drift-diffusion model.

3.6: Physical Models

3.6.1: Mobility Modeling

Electrons and holes are accelerated by electric fields, but lose momentum as a result of various scattering processes. These scattering mechanisms include lattice vibrations (phonons), impurity ions, other carriers, surfaces, and other material imperfections. Since the effects of all of these microscopic phenomena are lumped into the macroscopic mobilities introduced by the transport equations these mobilities are therefore functions of the local electric field, lattice temperature, doping concentration, and so on.

Mobility modeling is normally divided into: (i) low field behavior, (ii) high field behavior, (iii) bulk semiconductor regions and (iv) inversion layers.

The low electric field behavior has carriers almost in equilibrium with the lattice and the mobility has a characteristic low-field value that is commonly denoted by the symbol $\mu_{n0,p0}$. The value of this mobility is dependent upon phonon and impurity scattering. Both of which act to decrease the low field mobility.

The high electric field behavior shows that the carrier mobility declines with electric field because the carriers that gain energy can take part in a wider range of scattering processes. The mean drift velocity no longer increases linearly with increasing electric field, but rises more slowly. Eventually, the velocity doesn't increase any more with increasing field but saturates at a constant velocity. This constant velocity is commonly denoted by the symbol ν_{sat}. Impurity scattering is relatively insignificant for energetic carriers, and so ν_{sat} is primarily a function of the lattice temperature.

Modeling mobility in bulk material involves: (i) characterizing μ_{n0} and μ_{p0} as a function of doping and lattice temperature, (ii) characterizing ν_{sat} as a function of lattice temperature, and (iii) describing the transition between the low field mobility and saturated velocity regions.

Modeling carrier mobilities in inversion layers introduces additional complications. Carriers in inversion layers are subject to surface scattering, extreme carrier-carrier scattering, and quantum mechanical size quantization effects. These effects must be accounted for in order to perform accurate simulation of MOS devices. The transverse electric field is often used as a parameter that indicates the strength of inversion layer phenomena.

You can define multiple non-conflicting mobility models simultaneously. You also need to know which models are over-riding when conflicting models are defined.

Low Field Mobility Models

The low field carrier mobility can be defined in five different ways.

The first way is use the MUN and MUP parameters to set constant values for electron and hole mobilities. The second way is by using a look-up table model (CONMOB) to relate the low field mobility at 300K to the impurity concentration. The third way is by choosing the analytic low field mobility models, ANALYTIC, ARORA, or MASETTI, to relate the low field carrier mobility to impurity concentration and temperature.The fourth way is by choosing a carrier-carrier scattering model (CCSMOB, CONWELL, or BROOKS) that relates the low field mobility to the carrier concentrations and temperature. The fifth way is to use a unified low field mobility model (KLAASSEN) that relates the low field mobility to donor, acceptor, lattice, carrier-carrier scattering, and temperature.

Constant Low Field Mobility Model

In ATLAS, the choice of mobility model is specified on the MODELS statement. The parameters associated with mobility models are specified on a separate MOBILITY statement. One or more mobility models should always be specified explicitly. The default is to use constant low field mobilities within each region of a device. This default model is independent of doping concentration, carrier densities and electric field. It does account for lattice scattering due to temperature according to:

$$\mu_{n0} = \mathrm{MUN}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUN}} \quad 3\text{-}159$$

$$\mu_{p0} = \mathrm{MUP}\left(\frac{T_L}{300}\right)^{-\mathrm{TMUP}} \quad 3\text{-}160$$

where T is the lattice temperature. The low field mobility parameters: MUN, MUP, TMUN and TMUP can be specified in the MOBILITY statement with the defaults as shown in Table 3-19.

Table 3-19. User-Specifiable Parameters for the Constant Low Field Mobility Model

Statement	Parameter	Default	Units
MOBILITY	MUN	1000	$cm^2/(V \cdot s)$
MOBILITY	MUP	500	$cm^2/(V \cdot s)$
MOBILITY	TMUN	1.5	
MOBILITY	TMUP	1.5	

Concentration-Dependent Low Field Mobility Tables

ATLAS provides empirical data for the doping dependent low-field mobilities of electrons and holes in silicon at T_L=300K only. This data is used if the CONMOB parameter is specified in the MODELS statement. The data that is used is shown in Table 3-20.

Table 3-20. Mobility of Electrons and Holes in Silicon at T=300K

Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
1.0×1014	1350.0	495.0
2.0×1014	1345.0	495.0
4.0×1014	1335.0	495.0
6.0×1014	1320.0	495.0
8.0×1014	1310.0	495.0
1.0×1015	1300.0	491.1
2.0×10^{15}	1248.0	487.3

Table 3-20. Mobility of Electrons and Holes in Silicon at T=300K		
Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
4.0×10^{15}	1200.0	480.1
6.0×10^{15}	1156.0	473.3
8.0×10^{15}	1115.0	466.9
1.0×10^{16}	1076.0	460.9
2.0×10^{16}	960.0	434.8
4.0×10^{16}	845.0	396.5
6.0×10^{16}	760.0	369.2
8.0×10^{16}	720.0	348.3
1.0×10^{17}	675.0	331.5
2.0×10^{17}	524.0	279.0
4.0×10^{17}	385.0	229.8
6.0×10^{17}	321.0	2103.8
8.0×10^{17}	279.0	186.9
1.0×10^{18}	252.0	178.0
2.0×10^{18}	182.5	130.0
4.0×10^{18}	140.6	90.0
6.0×10^{18}	113.6	74.5
8.0×10^{18}	99.5	66.6
1.0×10^{19}	90.5	61.0
2.0×10^{19}	86.9	55.0
4.0×10^{19}	83.4	53.7
6.0×10^{19}	78.8	52.9
8.0×10^{19}	71.6	52.4
1.0×10^{20}	67.8	52.0
2.0×10^{20}	52.0	50.8
4.0×10^{20}	35.5	49.6

Table 3-20. Mobility of Electrons and Holes in Silicon at T=300K		
Concentration (cm^{-3})	Mobility ($cm^2/V \cdot s$)	
	Electrons	Holes
6.0×10^{20}	23.6	48.9
8.0×10^{20}	19.0	48.4
1.0×10^{21}	17.8	48.0

Analytic Low Field Mobility Model

The following analytic function based upon the work of Caughey and Thomas [32, 143] can be used to specify doping- and temperature-dependent low-field mobilities.

$$\mu_{n0} = \texttt{MU1N\_CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{ALPHAN.CAUG}} + \frac{\texttt{MU2N\_CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{BETAN} \cdot \texttt{CAUG}} - \texttt{MU1N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{ALPHAN} \cdot \texttt{CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\texttt{GAMMAN.CAUG}} \cdot \left(\frac{N}{\texttt{NCRITN\_CAUG}}\right)^{\texttt{DELTAN.CAUG}}} \quad \text{3-161}$$

$$\mu_{p0} = \texttt{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{ALPHAP} \cdot \texttt{CAUG}} + \frac{\texttt{MU2P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{BETAP} \cdot \texttt{CAUG}} - \texttt{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\texttt{ALPHAP} \cdot \texttt{CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\texttt{GAMMAP} \cdot \texttt{CAUG}} \cdot \left(\frac{N}{\texttt{NCRITP\_CAUG}}\right)^{\texttt{DELTAP} \cdot \texttt{CAUG}}} \quad \text{3-162}$$

where N is the local (total) impurity concentration in cm^{-3} and T_L is the temperature in degrees Kelvin.

This model is activated by specifying both the CONMOB and ANALYTIC parameters in the MODELS statement. The parameters of this model are specified in the MOBILITY statement. The default parameters are for silicon at T_L = 300K.

Table 3-21. User-Specifiable Parameters for Equations 3-161 and 3-162			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MU1N.CAUG	55.24	$cm^2/(V \cdot s)$
MOBILITY	MU1P.CAUG	49.7	$cm^2/(V \cdot s)$
MOBILITY	MU2N.CAUG	1429.23	$cm^2/(V \cdot s)$
MOBILITY	MU2P.CAUG	479.37	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.CAUG	0.0	arbitrary
MOBILITY	ALPHAP.CAUG	0.0	arbitrary
MOBILITY	BETAN. CAUG	-2.3	arbitrary
MOBILITY	BETAP. CAUG	-2.2	arbitrary
MOBILITY	GAMMAN.CAUG	-3.8	arbitrary
MOBILITY	GAMMAP.CAUG	-3.7	arbitrary
MOBILITY	DELTAN.CAUG	0.73	arbitrary
MOBILITY	DELTAP.CAUG	0.70	arbitrary
MOBILITY	NCRITN.CAUG	1.072×10^{17}	cm^{-3}
MOBILITY	NCRITP.CAUG	1.606×10^{17}	cm^{-3}

Arora Model for Low Field Mobility

Another analytic model for the doping and temperature dependence of the low field mobility is available in ATLAS. This model, which is due to Arora [15], has the following form:

$$\mu_{n0} = \mathrm{MU1N.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAN.ARORA}} + \frac{\mathrm{MU2N.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{BETAN.ARORA}}}{1 + \frac{N}{\mathrm{NCRITN.ARORA}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{GAMMAN.ARORA}}}} \quad 3\text{-}163$$

$$\mu_{p0} = \mathrm{MU1P.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAP.ARORA}} + \frac{\mathrm{MU2P.ARORA}\left(\frac{T_L}{300}\right)^{\mathrm{BETAP.ARORA}}}{1 + \frac{N}{\mathrm{NCRITP.ARORA}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{GAMMAP.ARORA}}}} \quad 3\text{-}164$$

This model is used if CONMOB and ARORA are specified in the MODELS statement. The parameters of the model are specified in the MOBILITY statement. The default parameters are for silicon at T_L=300K.

Table 3-22. User-Specifiable Parameters for Equations 3-163 and 3-164			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MU1N.ARORA	88.0	$cm^2/(V \cdot s)$
MOBILITY	MU1P.ARORA	54.3	$cm^2/(V \cdot s)$
MOBILITY	MU2N.ARORA	1252.0	$cm^2/(V \cdot s)$
MOBILITY	MU2P.ARORA	407.0	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.ARORA	-0.57	
MOBILITY	ALPHAP.ARORA	-0.57	
MOBILITY	BETAN.ARORA	-2.33	
MOBILITY	BETAP.ARORA	-2.33	
MOBILITY	GAMMAN.ARORA	2.546	
MOBILITY	GAMMAP.ARORA	2.546	
MOBILITY	NCRITN.ARORA	1.432×10^{17}	cm^{-3}
MOBILITY	NCRITP.ARORA	2.67×10^{17}	cm^{-3}

Masetti Model For Low Field Mobility

Masetti et al. [104] modelled the dependence of mobility on carrier concentration over a range of 8 orders of magnitude in carrier concentration from approximately 10^{13} cm^{-3} to 10^{21} cm^{-3}. They found that their model required different parameter sets for the electron mobility in Arsenic and Phosphorous n-doped Silicon. The model is optimized for room temperature, although it does model temperature dependence to some extent.

The functional form of the electron mobility is

$$\left(\mu_n = \mathrm{MTN.MIN1}\, exp\left(-\frac{\mathrm{MTN.PC}}{N}\right)\right) + \frac{\mathrm{MTN.MAX} - \mathrm{MTN.MIN2}}{1 + \left(\frac{N}{\mathrm{MTN.CR}}\right)^{\mathrm{MTN.ALPHA}}} - \frac{\mathrm{MTN.MU1}}{1 + \left(\frac{\mathrm{MTN.CS}}{N}\right)^{\mathrm{MTN.BETA}}} \quad 3\text{-}165$$

where N is the total or ionized doping level. The functional form of the hole mobility is

$$\left(\mu_p = \mathrm{MTP.MIN1}\, exp\left(-\frac{\mathrm{MTP.PC}}{N}\right)\right) + \frac{\mathrm{MTP.MAX} - \mathrm{MTP.MIN2}}{1 + \left(\frac{N}{\mathrm{MTP.CR}}\right)^{\mathrm{MTP.ALPHA}}} - \frac{\mathrm{MTP.MU1}}{1 + \left(\frac{\mathrm{MTP.CS}}{N}\right)^{\mathrm{MTP.BETA}}} \quad 3\text{-}166$$

where MTN.MAX and MTP.MAX are given a lattice temperature dependence as in Equations 3-159 and 3-160 respectively. These equations are functionally equivalent to the bulk mobility term in the Lombardi CVT model (Equations 3-214 and 3-215). To use it as a stand alone mobility model, use the MASETTI parameter on the MODELS statement for both electron and hole mobility.

N.MASETTI on the MOBILITY statement enables it for electrons. P.MASETTI on the MOBILITY statement enables it for holes.

For electron mobility, there is a choice of two default parameter sets. The set for arsenic doping are used unless you set the MSTI.PHOS parameter on the MOBILITY statement to use the set corresponding to Phosphorous. The following tables show default sets for electron mobility and for Boron doping (p-type).

Table 3-23. Parameter Set for Electron Mobility with Arsenic Doping in Silicon.

Statement	Parameter	Default	Units
MOBILITY	MTN.MIN1	52.2	$cm^2/(Vs)$
MOBILITY	MTN.MIN2	52.2	$cm^2/(Vs)$
MOBILITY	MTN.MAX	1417	$cm^2/(Vs)$
MOBILITY	MTN.MU1	43.4	$cm^2/(Vs)$
MOBILITY	MTN.PC	0.0	cm^{-3}
MOBILITY	MTN.CR	9.68×10^{16}	cm^{-3}
MOBILITY	MTN.CS	3.34×10^{20}	cm^{-3}
MOBILITY	MTN.ALPHA	0.68	
MOBILITY	MTN.BETA	2.0	

Table 3-24. Parameter Set for Electron Mobility with Phosphorous Doping in Silicon (MSTI.PHOS set)

Statement	Parameter	Default	Units
MOBILITY	MTN.MIN1	68.5	$cm^2/(Vs)$
MOBILITY	MTN.MIN2	68.5	$cm^2/(Vs)$
MOBILITY	MTN.MAX	1414	$cm^2/(Vs)$
MOBILITY	MTN.MU1	56.1	$cm^2/(Vs)$
MOBILITY	MTN.PC	0.0	cm^{-3}
MOBILITY	MTN.CR	9.2×10^{16}	cm^{-3}
MOBILITY	MTN.CS	3.41×10^{20}	cm^{-3}
MOBILITY	MTN.ALPHA	0. 711	
MOBILITY	MTN.BETA	1.98	

Table 3-25. Parameter Set for Hole Mobility with Boron doping in Silicon.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MTP.MIN1	44.9	$cm^2/(Vs)$
MOBILITY	MTP.MIN2	0.0	$cm^2/(Vs)$
MOBILITY	MTP.MAX	470.5	$cm^2/(Vs)$
MOBILITY	MTP.MU1	29.0	$cm^2/(Vs)$
MOBILITY	MTP.PC	9.23×10^{16}	cm^{-3}
MOBILITY	MTP.CR	2.23×10^{17}	cm^{-3}
MOBILITY	MTP.CS	6.1×10^{20}	cm^{-3}
MOBILITY	MTP.ALPHA	0.719	
MOBILITY	MTP.BETA	2.0	

Carrier-Carrier Scattering Models For Low Field Mobility

Dorkel and Leturcq Models

The Dorkel and Leturcq Model [46] for low field mobility includes the dependence on temperature, doping, and carrier-carrier scattering. This model is activated by specifying the CCSMOB parameter of the MODELS statement. The parameters of the model are specified in the MOBILITY statement. This model has the form:

$$\mu_{n0,\,p0} = \mu^{L}_{n,\,p}\left(\frac{1.025}{1+\left[1.126\left(\frac{\mu^{L}_{n,\,p}}{\mu^{IC}_{n,\,p}}\right)\right]^{0.715}}\right)^{-0.025} \qquad 3\text{-}167$$

where L is the lattice scattering, I is the ionized impurity scattering, and C is the carrier-carrier scattering. Here, $\mu^{IC}_{n,\,p}$ is defined as:

$$\mu^{IC}_{n,\,p} = \left[\frac{1}{\mu^{C}} + \frac{1}{\mu^{I}_{n,\,p}}\right]^{-1} \qquad 3\text{-}168$$

where:

$$\mu^{C} = \frac{1.04\cdot10^{21}\left(\frac{T_L}{300}\right)^{3/2}}{\sqrt{np}\,ln\left[1+7.45\cdot10^{13}\left(\frac{T_L}{300}\right)^{2}(np)^{-1/3}\right]} \qquad 3\text{-}169$$

$$\mu_n^I = \frac{\texttt{AN.CCS}\left(\frac{T_L}{300}\right)^{3/2}}{N_T} f\left[\frac{\texttt{BN.CCS}\left(\frac{T_L}{300}\right)^2}{n+p}\right] \quad 3\text{-}170$$

$$\mu_p^I = \frac{\texttt{AP.CCS}\left(\frac{T_L}{300}\right)^{3/2}}{N_T} f\left[\frac{\texttt{BP.CCS}\left(\frac{T_L}{300}\right)^2}{n+p}\right] \quad 3\text{-}171$$

Here, N_T is the total concentration, T_L is the lattice temperature and n,p are the electron and hole carrier concentrations respectively.

$$f(x) = \left[ln(1+x) - \frac{x}{1+x}\right]^{-1} \quad 3\text{-}172$$

The values of the lattice scattering terms, $\mu_{N,P}{}^L$ are defined by Equations 3-159 and 3-160.

Table 3-26. User-Specifiable Parameters for Equations 3-170 and 3-171

Statement	Parameter	Default	Units
MOBILITY	AN.CCS	4.61×10^{17}	cm^{-3}
MOBILITY	AP.CCS	1.0×10^{17}	cm^{-3}
MOBILITY	BN.CCS	1.52×10^{15}	cm^{-3}
MOBILITY	BP.CCS	6.25×10^{14}	cm^{-3}

Conwell-Weisskopf Model

This model adapts Conwell-Weisskopf theory to carrier-carrier scattering [35]. According to this model, the carrier-carrier contribution to mobility is

$$\mu_{ccs} = \frac{\texttt{D.CONWELL}\left(\frac{T}{300}\right)^{3/2}}{\sqrt{pn}\left(ln\left(1+\frac{\texttt{F.CONWELL}}{(pn)^{(1/3)}}\left(\frac{T}{300}\right)^{3/2}\right)\right)} \quad 3\text{-}173$$

where T is the Lattice temperature in Kelvin, n is the electron concentration, and p is the hole concentration. This is then combined with other enabled low field mobility models (if any) using Matthiessens rule, giving overall low field mobilities as follows

$$\frac{1}{\mu_n} = \frac{1}{\mu_{n0}} + \frac{1}{\mu_{ccs}} \quad 3\text{-}174$$

$$\frac{1}{\mu_p} = \frac{1}{\mu_{p0}} + \frac{1}{\mu_{ccs}} \quad 3\text{-}175$$

To enable the model for both electrons and holes, specify CONWELL on the MODELS statement. Alternatively, N.CONWELL on the MOBILITY statement enables it for electron mobility. P.CONWELL on the MOBILITY statement enables it for hole mobility.

Table 3-27. MOBILITY Statement Parameters

Parameter	Type	Default	Units
D.CONWELL	Real	1.04×10^{21}	$(cmV\cdot s)^{-1}$
F.CONWELL	Real	7.452×10^{13}	cm^{-2}
N.CONWELL	Logical	False	
P.CONWELL	Logical	False	

Brooks-Herring Model

Like the Conwell-Weisskopf model, this adds in the effects of carrier-carrier scattering to the low field mobility using Matthiesen's rule. The carrier-carrier contribution to mobility is

$$\mu_{ccs} = \frac{\text{A.BROOKS}\left(\frac{T}{300}\right)^{\frac{3}{2}}}{\sqrt{pn}\ \ \phi(\eta)} \quad \text{3-176}$$

where

$$\phi(n) = log(1+n) - \frac{n}{1+n} \quad \text{3-177}$$

and

$$n(T) = \frac{\text{B.BROOKS}\left(\frac{T}{300}\right)^{2}}{N_c F_{-1/2}\left(\frac{n}{N_c}\right) + N_v F_{-1/2}\left(\frac{p}{N_v}\right)} \quad \text{3-178}$$

$F_{-1/2}$ is the Fermi-Dirac function of order $-1/2$, N_c and N_v are the conduction band and valence band effective densities of states, n is the electron concentration, p is the hole concentration, and T is the Lattice temperature.

To enable the model for both electrons and holes, specify BROOKS on the MODELS statement. Alternatively, N.BROOKS on the MOBILITY statement enables it for electron mobility. P.BROOKS on the MOBILITY statement enables it for hole mobility.

Table 3-28. MOBILITY Statement Parameters			
Parameter	**Type**	**Default**	**Units**
A.BROOKS	Real	1.56×10^{21}	$(cmV\cdot s)^{-1}$
B.BROOKS	Real	7.63×10^{19}	cm^{-3}
N.BROOKS	Logical	False	
P.BROOKS	Logical	False	

Incomplete Ionization and Doping Dependent Mobility

The default in ATLAS is to use the total doping (N_A+N_D) in the formulae for the doping concentration dependence of low field mobility (e.g., ARORA model). As an alternative, ATLAS can use the ionised dopant concentration.

The ionized doping concentration is obtained automatically and used in calculating the charge density when you set the INCOMPLETE parameter on the MODELS statement. See Section 3.3.1: “Incomplete Ionization of Impurities” for more information.

ATLAS has a MOB.INCOMPL parameter on the MODELS statement. This allows you to use the ionized dopant concentration to calculate doping dependent mobilities. Specifying this parameter automatically sets the INCOMPLETE parameter. This ensures that the solution is consistent with an ionized rather than total doping concentration.

For high doping levels, the difference between total doping and ionized doping concentrations can be large and so this can result in quite different mobility values.

If it is required to use the ionized dopant concentration for mobility calculations and the total dopant concentration for calculating the charge density in Poisson's equation, then specify MOB.INCOMPL ^INCOMPLETE on the MODELS statement to explicitly clear this flag. This combination of parameters is not recommended.

MOB.INCOMPL affects the ANALYTIC, ARORA, MASETTI, YAMAGUCHI, CVT, KLAASSEN, ALBRECHT and tabulated Low Field Mobility Models.

Klaassen’s Unified Low Field Mobility Model

The model by D. B. M. Klaassen [86, 87], provides a unified description of majority and minority carrier mobilities. In so doing, it includes the effects of lattice scattering, impurity scattering (with screening from charged carriers), carrier-carrier scattering, and impurity clustering effects at high concentration. The model shows excellent agreement between the modeled and empirical data for:

- majority electron mobility as a function of donor concentration over the range of 10^{14} cm^{-3} to 10^{22} cm^{-3}
- minority electron mobility as a function of acceptor concentration over the range of 10^{17} cm^3 to 10^{20} cm^{-3}
- minority hole mobility as a function of donor concentration from 10^{17} cm^{-3} to 10^{20} cm^{-3}
- temperature dependence over the range of 70 K to 500 K

The Klaassen Model accounts for a broader set of effects and has been calibrated over a wider range of conditions than any other of the low field bulk mobility models. This is the recommended model for both MOS and bipolar simulation and is the default model for silicon when you set MOS2 or BIPOLAR2 in the MODELS statement. You can enable or disable the model by using the KLA parameter in0 the

MODELS statement, or independently for electrons and holes by the KLA.N and KLA.P parameters of the MOBILITY statement.

The total mobility can be described by its components using Matthiessen's rule as:

$$\mu_{n0}^{-1} = \mu_{nL}^{-1} + \mu_{nDAP}^{-1} \quad \text{3-179}$$

$$\mu_{p0}^{-1} = \mu_{pL}^{-1} + \mu_{pDAP}^{-1} \quad \text{3-180}$$

μ_n and μ_p are the total low field electron and hole mobilities, μ_{nL} and μ_{pL} are the electron and hole mobilities due to lattice scattering, μ_{nDAP} and μ_{pDAP} are the electron and hole mobilities due to donor (D), acceptor (A), screening (P) and carrier-carrier scattering.

The lattice scattering components, μ_{nL} and μ_{pL} are given as:

$$\mu_{nL} = \texttt{MUMAXN.KLA}\left(\frac{300}{T_L}\right)^{\texttt{THETAN.KLA}} \quad \text{3-181}$$

$$\mu_{pL} = \texttt{MUMAXP.KLA}\left(\frac{300}{T_L}\right)^{\texttt{THETAP.KLA}} \quad \text{3-182}$$

where T_L is the temperature in degrees Kelvin. MUMAXN.KLA, MUMAXP.KLA, THETAN.KLA, and THETAP.KLA are user-definable model parameters which can be specified as shown in Table 3-29.

Table 3-29. User-Specifiable Parameters for Equations 3-181 and 3-182

Statement	Parameter	Default	Units
MOBILITY	MUMAXN.KLA	1417.0	$cm^2/(V \cdot s)$
MOBILITY	MUMAXP.KLA	470.5	$cm^2/(V \cdot s)$
MOBILITY	THETAN.KLA	2.285	
MOBILITY	THETAP.KLA	2.247	

The impurity-carrier scattering components of the total mobility are given by:

$$\mu_{nDAP} = \mu_{N,n}\frac{N_{ncs}}{N_{nsc,eff}}\left(\frac{\texttt{NREF1N.KLA}}{N_{nsc}}\right)^{\texttt{ALPHA1N.KLA}} + \mu_{nc}\left(\frac{n+p}{N_{nsc,eff}}\right) \quad \text{3-183}$$

$$\mu_{pDAP} = \mu_{N,p}\frac{N_{psc}}{N_{psc,eff}}\left(\frac{\texttt{NREF1P.KLA}}{N_{psc}}\right)^{\texttt{ALPHA1P.KLA}} + \mu_{pc}\left(\frac{n+p}{N_{psc,eff}}\right) \quad \text{3-184}$$

Table 3-30. User-Specifiable Parameters for Equations 3-183 and 3-184			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALPHA1N.KLA	0.68	
MOBILITY	ALPHA1P.KLA	0.719	
MOBILITY	NREF1N.KLA	9.68×10^{16}	cm^3
MOBILITY	NREF1P.KLA	2.23×10^{17}	cm^3

The impurity scattering components, $\mu_{N,n}$ and $\mu_{N,p}$, are given by:

$$\mu_{N,n} = \frac{\text{MUMAXN.KLA}^2}{\text{MUMAXN.KLA} - \text{MUMINN.KLA}}\left(\frac{T_L}{300}\right)^{3\text{ALPHA1N.KLA} - 1.5} \quad 3\text{-}185$$

$$\mu_{N,p} = \frac{\text{MUMAXP.KLA}^2}{\text{MUMAXP.KLA} - \text{MUMINP.KLA}}\left(\frac{T_L}{300}\right)^{3\text{ALPHA1P.KLA}-1.5} \quad 3\text{-}186$$

where T_L is the temperature in degrees Kelvin. MUMINN.KLA and MUMINP.KLA are user-defined parameters shown in Table 3-31, and the other parameters are as described in Tables 3-29 and 3-30.

Table 3-31. User-Specifiable Parameters for Equations 3-185 and 3-186			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MUMINN.KLA	52.2	cm^2(V·s)
MOBILITY	MUMINP.KLA	44.9	cm^2(V·s)

The carrier-carrier scattering components, μ_{nc} and μ_{pc}, are given by:

$$\mu_{nc} = \frac{\text{MUMINN.KLA} \times \text{MUMAXN.KLA}}{\text{MUMAXN.KLA} - \text{MUMINN.KLA}}\left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}187$$

$$\mu_{pc} = \frac{\text{MUMINP.KLA} \times \text{MUMAXP.KLA}}{\text{MUMAXP.KLA} - \text{MUMINP.KLA}}\left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}188$$

The N_{nsc} and N_{psc} parameters of Equations 3-183 and 3-184 are given by:

$$N_{nsc} = N_D + N_A + p \quad 3\text{-}189$$

$$N_{psc} = N_D + N_A + n \quad 3\text{-}190$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3}, n is the electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}.

The parameters of Equations 3-185 and 3-186 are given by:

$$N_{nsc,eff} = N_D + G(P_n)N_A + \left(\frac{p}{F(P_n)}\right) \quad 3\text{-}191$$

$$N_{psc,eff} = N_A + G(P_p)N_D + \left(\frac{n}{F(P_p)}\right) \quad 3\text{-}192$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3} and n is the electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}. The two functions, *G(P)* and *F(P)*, are functions of the screening factors, P_n and P_p, for electrons and holes. The function, *G(P)*, in Equations 3-191 and 3-192 are given by:

$$G(P_n) = 1 - \frac{\texttt{S1.KLA}}{\left[\texttt{S2.KLA} + P_{BH,n}\left(\frac{m_\circ}{m_e}\frac{T_L}{300}\right)^{\texttt{S4.KLA}}\right]^{\texttt{S3.KLA}}} + \frac{\texttt{S5.KLA}}{\left[P_{BH,n}\left(\frac{m_\circ 300}{m_e T_L}\right)^{\texttt{S7.KLA}}\right]^{\texttt{S6.KLA}}} \quad 3\text{-}193$$

$$G(P_p) = 1 - \frac{\texttt{S1.KLA}}{\left[\texttt{S2.KLA} + P_{BH,p}\left(\frac{m_\circ}{m_h}\frac{T_L}{300}\right)^{\texttt{S4.KLA}}\right]^{\texttt{S3.KLA}}} + \frac{\texttt{S5.KLA}}{\left[P_{BH,p}\left(\frac{m_\circ 300}{m_h T_L}\right)^{\texttt{S7.KLA}}\right]^{\texttt{S6.KLA}}} \quad 3\text{-}194$$

Here, T_L is the temperature in degrees Kelvin, m_e and m_h are the electron and hole masses and the parameters `S1.KLA` through `S7.KLA` are user-specifiable model parameters as shown in Table 3-32.

.

Table 3-32. User-Specifiable Parameters for Equations 3-193 and 3-194

Statement	Parameter	Default	Units
MOBILITY	S1.KLA	0.89233	
MOBILITY	S2.KLA	0.41372	
MOBILITY	S3.KLA	0.19778	
MOBILITY	S4.KLA	0.28227	
MOBILITY	S5.KLA	0.005978	
MOBILITY	S6.KLA	1.80618	
MOBILITY	S7.KLA	0.72169	

The functions, $F(P_n)$ and $F(P_p)$, in Equations 3-191 and 3-192 are given by:

$$F(P_n) = \frac{\texttt{R1.KLA}\, P_n^{\texttt{R6.KLA}} + \texttt{R2.KLA} + \texttt{R3.KLA}\,\frac{m_e}{m_h}}{P_n^{\texttt{R6.KLA}} + \texttt{R4.KLA} + \texttt{R5.KLA}\,\frac{m_e}{m_h}} \quad 3\text{-}195$$

$$F(P_p) = \frac{\texttt{R1.KLA}\, P_p^{\texttt{R6.KLA}} + \texttt{R2.KLA} + \texttt{R3.KLA}\,\dfrac{m_h}{m_e}}{P_p^{\texttt{R6.KLA}} + \texttt{R4.KLA} + \texttt{R5.KLA}\,\dfrac{m_h}{m_e}} \quad \text{3-196}$$

where the parameters, R1.KLA through R6.KLA, are user-specifiable as shown in Table 3-33.

Table 3-33. User-Specifiable Parameters for Equations 3-195 and 3-196

Statement	Parameter	Default	Units
MOBILITY	R1.KLA	0.7643	
MOBILITY	R2.KLA	2.2999	
MOBILITY	R3.KLA	6.5502	
MOBILITY	R4.KLA	2.3670	
MOBILITY	R5.KLA	-0.8552	
MOBILITY	R6.KLA	0.6478	

The screening parameters, P_n and P_p, used in Equations 3-195 and 3-196 are given by:

$$P_n = \left[\frac{\texttt{FCW.KLA}}{P_{CW,n}} + \frac{\texttt{FBH.KLA}}{P_{BH,n}}\right]^{-1} \quad \text{3-197}$$

$$P_p = \left[\frac{\texttt{FCW.KLA}}{P_{CW,p}} + \frac{\texttt{FBH.KLA}}{P_{BH,p}}\right]^{-1} \quad \text{3-198}$$

Here, the FCW.KLA and FBH.KLA parameters are user-specifiable model parameters as shown in Table 3-34.

Table 3-34. User-Specifiable Parameters for Equations 3-197 and 3-198

Statement	Parameter	Default	Units
MOBILITY	FCW.KLA	2.459	
MOBILITY	FBH.KLA	3.828	

The functions, $P_{BH,n}$ and $P_{BH,p}$, $P_{CW,n}$, and $P_{CW,p}$ are given by the following equations.

$$P_{BH,n} = \frac{1.36\times10^{20}}{n}\left(\frac{m_e}{m_0}\right)\left(\frac{T_L}{300}\right)^2 \quad \text{3-199}$$

$$P_{BH,p} = \frac{1.36\times10^{20}}{p}\left(\frac{m_h}{m_0}\right)\left(\frac{T_L}{300}\right)^2 \quad \text{3-200}$$

$$P_{CW,n} = 3.97 \times 10^{13} \left\{ \frac{1}{Z_n^3 N_D} \left(\frac{T_L}{300} \right)^3 \right\}^{\frac{2}{3}} \quad 3\text{-}201$$

$$P_{CW,p} = 3.97 \times 10^{13} \left\{ \frac{1}{Z_p^3 N_A} \left(\frac{T_L}{300} \right)^3 \right\}^{\frac{2}{3}} \quad 3\text{-}202$$

where T_L is the temperature in degrees Kelvin, m_e/m_0 and m_h/m_0 are the normalized carrier effective masses, and n and p are the electron and hole concentrations in cm^{-3}.

Also here, N_D and N_A are the donor and acceptor concentrations in cm^{-3}, T_L is the temperature in degrees Kelvin, and Z_n and Z_p are clustering functions given by:

$$Z_n = 1 + \frac{1}{\mathtt{CD.KLA} + \left(\frac{\mathtt{NREFD.KLA}}{N_D} \right)^2} \quad 3\text{-}203$$

$$Z_p = 1 + \frac{1}{\mathtt{CA.KLA} + \left(\frac{\mathtt{NREFA.KLA}}{N_A} \right)^2} \quad 3\text{-}204$$

where N_D and N_A are the donor and acceptor concentrations in cm^{-3} and CD.KLA, CA.KLA, NREFD.KLA, and NREFA.KLA are user-definable parameters as given in Table 3-35.

Table 3-35. User-Specifiable Parameters for Equations 3-203 and 3-204

Statement	Parameter	Default	Units
MOBILITY	CD.KLA	0.21	
MOBILITY	CA.KLA	0.50	
MOBILITY	NREFD.KLA	4.0×10^{20}	cm^3
MOBILITY	NREFA.KLA	7.2×10^{20}	cm^3

Note: When the Klaassen low field mobility is used, remember that it has been calibrated to work with Klaassen's models for bandgap narrowing, KLAAUG recombination, and KLASRH recombination. These models are described in the Section 3.6.3: "Carrier Generation-Recombination Models".

Uchida's Low Field Model for Ultrathin SOI

The model by Uchida et.al. [161], provides a mobility limit in ultrathin-body MOSFET transistors with SOI thickness less than 4 nm. This mobility limit is due to thickness fluctuations in the nano scale SOI film. The model for electrons and holes is given by Equations 3-205 and 3-206.

$$\mu_{nu} = \texttt{CN.UCHIDA} \cdot \texttt{TN.UCHIDA}^{6} \quad \text{3-205}$$

$$\mu_{nu} = \texttt{CP.UCHIDA} \cdot \texttt{TP.UCHIDA}^{6} \quad \text{3-206}$$

The parameters CN.UCHIDA, CP.UCHIDA are calibrated from the reference to a default value of 0.78125 $cm^2/V_s\mu m^6$. The parameters TN.UCHIDA and TP.UCHIDA represent the thickness of the SOI in µm. This value should as close as possible to match the physical thickness of the SOI layer in the simulated structure file. To enable this model for electrons and holes, specify values for TN.UCHIDA and TP.UCHIDA, and specify the logical parameters UCHIDA.N or UCHIDA.P or both on the MOBILITY statement. Table 3-36 lists the user specifiable parameters.

Table 3-36. User-Specifiable Parameters for Equations 3-187 and 3-188

Statement	Parameter	Default	Units
MOBILITY	CN.UCHIDA	0.78125	$cm^2/Vs\mu m^6$
MOBILITY	CP.UCHIDA	0.78125	$cm^2/Vs\mu m^6$
MOBILITY	TN.UCHIDA	0.0	µm
MOBILITY	TP.UCHIDA	0.0	µm

Inversion Layer Mobility Models

To obtain accurate results for MOSFET simulations, you need to account for the mobility degradation that occurs inside inversion layers. The degradation normally occurs as a result of the substantially higher surface scattering near the semiconductor to insulator interface.

This effect is handled within ATLAS by three distinct methods:

- a surface degradation model SURFMOB
- a transverse electric field model SHIRAHATA
- specific inversion layer mobility models CVT, YAMAGUCHI, and TASCH

The CVT, YAMAGUCHI, and TASCH models are designed as stand-alone models which incorporate all the effects required for simulating the carrier mobility.

Lombardi CVT Model

The inversion layer model from Lombardi [99] is selected by setting CVT on the MODEL statement. This model overrides any other mobility models which may be specified on the MODELS statement. In the CVT model, the transverse field, doping dependent and temperature dependent parts of the mobility are given by three components that are combined using Matthiessen's rule. These components are μ_{AC}, μ_{sr} and μ_b and are combined using Matthiessen's rule as follows:

$$\mu_T^{-1} = \mu_{AC}^{-1} + \mu_b^{-1} + \mu_{sr}^{-1} \quad \text{3-207}$$

The first component, μ_{AC}, is the surface mobility limited by scattering with acoustic phonons:

$$\mu_{AC,\,n} = \frac{\mathrm{BN.CVT}}{E_{\perp}^{\mathrm{EN.CVT}}} + \frac{\mathrm{CN.CVT}\,N^{\mathrm{TAUN.CVT}}}{T_L E_{\perp}^{\mathrm{DN.CVT}}} \quad 3\text{-}208$$

$$\mu_{AC,\,p} = \frac{\mathrm{BP.CVT}}{E_{\perp}^{\mathrm{EP.CVT}}} + \frac{\mathrm{CP.CVT}\,N^{\mathrm{TAUP.CVT}}}{T_L E_{\perp}^{\mathrm{DP.CVT}}} \quad 3\text{-}209$$

where T_L is the temperature, $E_{\perp}$ is the perpendicular electric field, and N is the total doping concentration. In this discussion, you should assume that $E_{\perp}$ is scaled by a factor of 1V/cm and N is scaled by a factor of $1\mathrm{cm}^{-3}$. You can define the parameters BN.CVT, BP.CVT, CN.CVT, CP.CVT, DN.CVT, DP.CVT, TAUN.CVT, and TAUP.CVT in the MOBILITY statement (see Table 3-37 for their defaults).

The second component, μ_{sr}, is the surface roughness factor and is given by:

$$\mu_{sr,\,n} = \frac{\mathrm{DELN.CVT}}{\mathrm{KN.CVT}} \quad 3\text{-}210$$
$$E_{\perp}$$

$$\mu_{sr,\,p} = \frac{\mathrm{DELP.CVT}}{\mathrm{KP.CVT}} \quad 3\text{-}211$$
$$E_{\perp}$$

You can also specify parameters FELN.CVT and FELP.CVT. These modify the expression for μ_{sr}, which can now be written as

$$\frac{1}{\mu_{sr,\,n}} = \frac{E_{\perp}^{\mathrm{KN.CVT}}}{\mathrm{DELN.CVT}} + \frac{E_{\perp}^{3}}{\mathrm{FELN.CVT}} \quad 3\text{-}212$$

for electrons and

$$\frac{1}{\mu_{sr,\,p}} = \frac{E_{\perp}^{\mathrm{KP.CVT}}}{\mathrm{DELP.CVT}} + \frac{E_{\perp}^{3}}{\mathrm{FELP.CVT}} \quad 3\text{-}213$$

for holes.

The default values of FELN.CVT and FELP.CVT are set so high that the second terms are negligible. The KN.CVT, KP.CVT, DELN.CVT, DELP.CVT, FELN.CVT and FELP.CVT parameters are user-definable (see Table 3-37 for their defaults).

The third mobility component, μ_b, is the mobility limited by scattering with optical intervalley phonons. This component is given by:

$$\mu_{b,n} = \mathrm{MU0N.CVT}\,exp\left(\frac{-\mathrm{PCN.CVT}}{N}\right) + \frac{\left[\mathrm{MUMAXN.CVT}\left(\frac{T_L}{300}\right)^{-\mathrm{GAMN.CVT}} - \mathrm{MU0N.CVT}\right]}{1+\left(\frac{N}{\mathrm{CRN.CVT}}\right)^{\mathrm{ALPHN.CVT}}} - \frac{\mathrm{MU1N.CVT}}{1+\left(\frac{\mathrm{CSN.CVT}}{N}\right)^{\mathrm{BETAN.CVT}}} \quad \text{3-214}$$

$$\mu_{b,p} = \mathrm{MU0P.CVT}\,exp\left(\frac{-\mathrm{PCP.CVT}}{N}\right) + \frac{\left[\mathrm{MUMAXP.CVT}\left(\frac{T_L}{300}\right)^{-\mathrm{GAMP.CVT}} - \mathrm{MU0P.CVT}\right]}{1+\left(\frac{N}{\mathrm{CRP.CVT}}\right)^{\mathrm{ALPHP.CVT}}} - \frac{\mathrm{MU1P.CVT}}{1+\left(\frac{\mathrm{CSP.CVT}}{N}\right)^{\mathrm{BETAP.CVT}}} \quad \text{3-215}$$

Here, N is the total density of impurities and T_L is the temperature in degrees Kelvin.

Because μ_{AC} and μ_{sr} are related to interaction with an interface, you can specify a typical distance from the interface over which these components are significant. If you specify the N.LCRIT or P.LCRIT parameters or both then factors

$$FE = exp(-1/\mathrm{N.LCRIT}) \quad \text{3-216}$$

and

$$FH = exp(-1/\mathrm{P.LCRIT}) \quad \text{3-217}$$

where l is the shortest distance to the nearest interface from the point at which the mobility will be calculated. These factors are then used to modify the Matthiesen's rule of Equations 3-165 and 3-166 to

$$\frac{1}{\mu_T} = \frac{FE}{\mu_{AC}} + \frac{FE}{\mu_{sr}} + \frac{1}{\mu_b} \quad \text{3-218}$$

for electrons and

$$\frac{1}{\mu_T} = \frac{FH}{\mu_{AC}} + \frac{FH}{\mu_{sr}} + \frac{1}{\mu_b}$$

for holes. Far from the interface, the mobility is the same as the bulk mobility.

Table 3-37. User-Specifiable Parameters for Equations 3-208 to 3-215			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALPHN.CVT	0.680	
MOBILITY	ALPHP.CVT	0.71	
MOBILITY	BETAN.CVT	2.00	
MOBILITY	BETAP.CVT	2.00	
MOBILITY	BN.CVT	4.75×10^{7}	$cm^2/(V\cdot s)$
MOBILITY	BP.CVT	9.925×10^{6}	$cm^2/(V\cdot s)$
MOBILITY	CN.CVT	1.74×10^{5}	
MOBILITY	CP.CVT	8.842×10^{5}	
MOBILITY	CRN.CVT	9.68×10^{16}	cm^{-3}
MOBILITY	CRP.CVT	2.23×10^{17}	cm^{-3}
MOBILITY	CSN.CVT	3.43×10^{20}	cm^{-3}
MOBILITY	CSP.CVT	6.10×10^{20}	cm^{-3}
MOBILITY	DELN.CVT	5.82×10^{14}	$cm^2/(V\cdot s)$
MOBILITY	DELP.CVT	2.0546×10^{14}	$cm^2/(V\cdot s)$
MOBILITY	DN.CVT	0.333	
MOBILITY	DP.CVT	0.333	
MOBILITY	EN.CVT	1.0	
MOBILITY	EP.CVT	1.0	
MOBILITY	FELN.CVT	1.0×10^{50}	$V^2/(cm\cdot s)$
MOBILITY	FELP.CVT	1.0×10^{50}	$V^2/(V\cdot s)$
MOBILITY	KN.CVT	2.0	
MOBILITY	KP.CVT	2.0	
MOBILITY	GAMN.CVT	2.5	
MOBILITY	GAMP.CVT	2.2	
MOBILITY	MU0N.CVT	52.2	$cm^2/(V\cdot s)$
MOBILITY	MU0P.CVT	44.9	$cm^2/(V\cdot s)$
MOBILITY	MU1N.CVT	43.4	$cm^2/(V\cdot s)$
MOBILITY	MU1P.CVT	29.0	cm2/(V·s)

Table 3-37. User-Specifiable Parameters for Equations 3-208 to 3-215			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MUMAXN.CVT	1417.0	$cm^2/(V \cdot s)$
MOBILITY	MUMAXP.CVT	470.5	$cm^2/(V \cdot s)$
MOBILITY	N.LCRIT	1.0	cm
MOBILITY	P.LCRIT	1.0	cm
MOBILITY	PCN.CVT	0.0	cm^{-3}
MOBILITY	PCP.CVT	9.23×10^{16}	cm^{-3}
MOBILITY	TAUN.CVT	0.125	
MOBILITY	TAUP.CVT	0.0317	

Note: The CVT model when activated will also, by default, apply the Parallel Electric Field Mobility Model which is described in the "Parallel Electric Field-Dependent Mobility" section on page 3-72. In this model, the low field mobility is supplied from the CVT model.

Darwish CVT Model

The CVT model already described has been successfully used in many device simulations. Lately, an improved version of this model has been proposed by Darwish et al. [44] where several modifications have been implemented. The first modification is that the bulk mobility is calculated using Klaassen's model (see "Klaassen's Unified Low Field Mobility Model" section on page 3-51) to take into account coulomb screening effects. The second modification is a new expression for surface roughness is used.

The new model for surface roughness replaces Equations 3-210 and 3-211 with:

$$\frac{1}{\mu_{sr,\,n}} = \frac{E_{\perp}^{\gamma_n}}{\mathrm{DELN.CVT}} + \frac{E_{\perp}^{3}}{\mathrm{FELN.CVT}} \qquad 3\text{-}219$$

$$\frac{1}{\mu_{sr,\,p}} = \frac{E_{\perp}^{\gamma_p}}{\mathrm{DELP.CVT}} + \frac{E_{\perp}^{3}}{\mathrm{FELP.CVT}} \qquad 3\text{-}220$$

where:

$$\gamma_n = \mathrm{AN.CVT} + \frac{\mathrm{ALN.CVT} \times N}{\mathrm{ETAN.CVT}} \qquad 3\text{-}221$$

N_{TOT}

$$\gamma_p = \texttt{AP.CVT} + \frac{\texttt{ALP.CVT} \times N}{\texttt{ETAP.CVT}} \qquad 3\text{-}222$$
$$N_{TOT}$$

Here, N_{TOT} is the total doping density *($N_D + N_A$)* and N is the total carrier concentration *(n+p)*.

Table 3-38 shows the default parameters for the new equations. You should assume that N_{TOT} is scaled by a factor of 1 cm^{-3}.

Table 3-38. Default parameters for the surface roughness components of new CVT model.

Parameter	Statement	Default	Units
AN.CVT	MOBILITY	2.58	
AP.CVT	MOBILITY	2.18	
ALN.CVT	MOBILITY	6.85×10^{-21}	cm^3
ALP.CVT	MOBILITY	7.82×10^{-21}	cm^3
ETAN.CVT	MOBILITY	0.0767	
ETAP.CVT	MOBILITY	0.123	

Next, in the Darwish model, the expressions for acoustic phonon scattering are modified as shown in Equations 3-223 and 3-224.

$$\mu_{AC,\,n} = \frac{\texttt{BN.CVT}}{E_{\perp}^{\texttt{EN.CVT}}} + \frac{\texttt{CN.CVT}\ N^{\texttt{TAU.CVT}}}{T_n' E_{\perp}^{\texttt{DN.CVT}}} \qquad 3\text{-}223$$

$$\mu_{AC,\,p} = \frac{\texttt{BP.CVT}}{E_{\perp}^{\texttt{EP.CVT}}} + \frac{\texttt{CP.CVT}\ N^{\texttt{TAUP.CVT}}}{T_p' E_{\perp}^{\texttt{DP.CVT}}} \qquad 3\text{-}224$$

where

$$T_n' = (T_L/300)^{\texttt{KAPPAN.CVT}} \qquad 3\text{-}225$$

$$T_p' = (T_L/300)^{\texttt{KAPPAP.CVT}} \qquad 3\text{-}226$$

Table 3-39 shows the default values for KAPPAN.CVT and KAPPAP.CVT.

Table 3-39. Default Parameter Values for Equations 3-225 and 3-226.

Statement	Parameter	Default	Units
MOBILITY	KAPPAN.CVT	1.7	
MOBILITY	KAPPAP.CVT	0.9	

Finally, the default values for many of the paremeters are changed from those listed Table 3-37. Table 3-31 lists the defaults used for the Darwish model.

Table 3-40. Default Parameters for the Darwish model.

Statement	Parameter	Default	Units
MOBILITY	BN.CVT	3.61×10^{7}	cm/s
MOBILITY	BP.CVT	1.51×10^{7}	cm/s
MOBILITY	CN.CVT	1.7×10^{4}	
MOBILITY	CP.CVT	4.18×10^{3}	
MOBILITY	TAUN.CVT	0.0233	
MOBILITY	TAUP.CVT	0.0119	
MOBILITY	DELN.CVT	3.58×10^{18}	$cm^2/(V\cdot s)$
MOBILITY	DELP.CVT	4.1×10^{15}	$cm^2/(V\cdot s)$

You can enable the Darwish model for electrons and holes by specifying NEWCVT.N or NEWCVT.P on the MOBILITY statement.

Yamaguchi Model

The Yamaguchi Model [186] is selected by setting YAMAGUCHI in the MODELS statement. This model overrides any mobility model specifications other than the CVT model. The model consists of calculating the low field, doping dependent mobility. Surface degradation is then accounted for based upon the transverse electric field before including the parallel electric field dependence.

The low field part of the Yamaguchi Model is given as follows:

$$\mu_{n0} = \mathrm{MULN.YAMA}\left[1 + \frac{N_i}{\frac{N_i}{\mathrm{SN.YAMA}} + \mathrm{NREFP.YAMA}}\right]^{-\frac{1}{2}} \quad 3\text{-}227$$

$$\mu_{p0} = \mathrm{MULP.YAMA}\left[1 + \frac{N_i}{\frac{N_i}{\mathrm{SP.YAMA}} + \mathrm{NREFP.YAMA}}\right]^{-\frac{1}{2}} \quad 3\text{-}228$$

where N_i is the total impurity concentration. The equation parameters: MULN.YAMA, MULP.YAMA, SN.YAMA, SP.YAMA, NREFP.YAMA, and NREFP.YAMA are user-definable in the MOBILITY statement (see Table 3-41 for their defaults).

The transverse electric field dependence is accounted for as follows:

$$\mu_{s,\,n} = \mu_{n0}(1 + \mathrm{ASN.YAMA}\,E_{\perp})^{-\frac{1}{2}} \quad 3\text{-}229$$

$$\mu_{s,p} = \mu_{p0}(1 + \mathrm{ASP.YAMA}\, E_{\perp})^{-\frac{1}{2}} \quad 3\text{-}230$$

where $E_{\perp}$ is the perpendicular electric field and the equation parameters, ASN.YAMA and ASP.YAMA, are user-definable in the MOBILITY statement (see Table 3-41 for their defaults).

The final calculation of mobility takes into account the parallel electric field dependence which takes the form:

$$\mu_n = \mu_{s,n}\left[1 + \left(\frac{\mu_{s,n}E}{\mathrm{ULN.YAMA}}\right)^2 \left(\mathrm{GN.YAMA} + \frac{\mu_{s,n}E}{\mathrm{ULN.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,n}E}{\mathrm{VSN.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad 3\text{-}231$$

$$\mu_p = \mu_{s,p}\left[1 + \left(\frac{\mu_{s,p}E}{\mathrm{ULP.YAMA}}\right)^2 \left(\mathrm{GP.YAMA} + \frac{\mu_{s,p}E}{\mathrm{ULP.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,p}E}{\mathrm{VSP.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad 3\text{-}232$$

where E is the parallel electric field and the equation parameters: ULN.YAMA, ULP.YAMA, VSN.YAMA, VSP.YAMA, GN.YAMA, and GP.YAMA are user-definable in the MOBILITY statement (see Table 3-41 for their defaults).

Table 3-41. User-Specifiable Parameters for Equations 3-227 to 3-232

Statement	Parameter	Default	Units
MOBILITY	SN.YAMA	350.0	
MOBILITY	SP.YAMA	81.0	
MOBILITY	NREFP.YAMA	3.0×10^{16}	cm^{-3}
MOBILITY	NREFP.YAMA	4.0×10^{16}	cm^{-3}
MOBILITY	MULN.YAMA	1400.0	$cm^2/(V\cdot s)$
MOBILITY	MULP.YAMA	480.0	$cm^2/(V\cdot s)$
MOBILITY	ASN.YAMA	1.54×10^{-5}	cm/V
MOBILITY	ASP.YAMA	5.35×10^{-5}	cm/V
MOBILITY	VSN.YAMA	1.036×10^{7}	cm/s
MOBILITY	VSP.YAMA	1.2×10^{7}	cm/s
MOBILITY	ULN.YAMA	4.9×10^{6}	cm/s
MOBILITY	ULP.YAMA	2.928×10^{6}	cm/s
MOBILITY	GN.YAMA	8.8	
MOBILITY	GP.YAMA	1.6	

The Tasch Model

S-PISCES includes an improved local field-dependent mobility model. This model, which was originally derived and published by Tasch et. al [146,147] has been designed explicitly for MOSFETs. It defines the mobility as a function of the perpendicular and parallel electric fields, the interface charge, the lattice temperature and the doping concentration. To activate this model, is activated use the TASCH parameter on the MODELS statement. This mobility model is given by the following expressions:

$$\mu_n = \Gamma_n + (E_{perp} - E_0)\frac{d\Gamma_n}{dE_{perp}} \quad 3\text{-}233$$

$$\mu_p = \Gamma_p + (E_{perp} - E_0)\frac{d\Gamma_p}{dE_{perp}} \quad 3\text{-}234$$

where *Eperp* is the transverse electric field and E_0 is the transverse electric field at the edge of the inversion layer. The functions $\Gamma_{n,p}$ are defined as:

$$\Gamma_n = \frac{\mu_{eff,n}}{\left(1 + \left(\frac{\mu_{eff,n} E_{\parallel}}{\texttt{VSATN}}\right)^{\texttt{BETAN}}\right)^{1/\texttt{BETAN}}} \quad 3\text{-}235$$

$$\Gamma_p = \frac{\mu_{eff,p}}{\left(1 + \left(\frac{\mu_{eff,p} E_{\parallel}}{\texttt{VSATP}}\right)^{\texttt{BETAP}}\right)^{1/\texttt{BETAP}}} \quad 3\text{-}236$$

The carrier mobilities $\mu_{eff,n}$ and $\mu_{eff,p}$ are defined by three components μ_{ph}, μ_{sr} and μ_c which are combined by Mathiessen's rule according to:

$$\mu_{eff} = \left[\frac{1}{\mu_{ph}} + \frac{1}{\mu_{sr}} + \frac{1}{\mu_c}\right]^{-1} \quad 3\text{-}237$$

The term μ_{ph} takes account of the degradation in mobility due to acoustic phonon scattering through the expressions:

$$\mu_{ph,n}^{-1} = \left(\texttt{MUBN.TAS}\left(\frac{T_L}{300}\right)^{-\texttt{TMUBN.TAS}}\right)^{-1} + \left(Z_n/\texttt{DN.TAS}\; Y_n\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}238$$

$$\mu_{ph,p}^{-1} = \left(\texttt{MUBP.TAS}\left(\frac{T_L}{300}\right)^{-\texttt{TMUBP.TAS}}\right)^{-1} + \left(Z_p/\texttt{DP.TAS}\; Y_p\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}239$$

The function, $Z_{n,p}$, is defined as:

$$Z_n = \texttt{Z11N.TAS}\left(\frac{T_L}{300}\right)E_{eff,n}^{-1} + \texttt{Z22N.TAS}\; E_{eff,n}^{-\frac{1}{3}} \quad 3\text{-}240$$

$$Z_p = \texttt{Z11P.TAS}\left(\frac{T_L}{300}\right)E_{eff,p}^{-1} + \texttt{Z22P.TAS}\; E_{eff,p}^{-\frac{1}{3}} \quad 3\text{-}241$$

where:

$$E_{eff,\,n} = \frac{(E_{perp} + (\mathrm{RN.TAS} - 1)\cdot E_0)}{\mathrm{RP.TAS}} \quad 3\text{-}242$$

$$E_{eff,\,p} = \frac{(E_{perp} + (\mathrm{RN.TAS} - 1)\cdot E_0)}{\mathrm{RP.TAS}} \quad 3\text{-}243$$

Also, the function $Y_{n,p}$, is defined as:

$$Y_n = \mathrm{P1N.TAS}\left(\frac{T_L}{300}\right)^{\mathrm{B1N.TAS}} + \mathrm{P2N.TAS}\; n^{\;+\mathrm{B2N.TAS}}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}244$$

$$Y_p = \mathrm{P1P.TAS}\left(\frac{T_L}{300}\right)^{\mathrm{B1P.TAS}} + \mathrm{P2P.TAS}\; p^{\;+\mathrm{B2P.TAS}}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}245$$

Mobility degradation due to surface roughness is accounted for by the term μ_{sr} which is calculated according to:

$$\mu_{sr,\,n} = \left(\frac{\mathrm{ESRN.TAS}}{E_{eff,\,n}}\right)^{\mathrm{BETAN.TAS}} \quad 3\text{-}246$$

$$\mu_{sr,\,p} = \left(\frac{\mathrm{ESRP.TAS}}{E_{eff,\,p}}\right)^{\mathrm{BETAP.TAS}} \quad 3\text{-}247$$

The final term, μ_C, models Coulombic scattering with the expressions:

$$\mu_{C,\,n} = \frac{\mathrm{N2N.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_A\; ln(1+\gamma_{BH,\,n}) - \frac{\gamma_{BH,\,n}}{(1+\gamma_{BH,\,n})}} \quad 3\text{-}248$$

$$\mu_{C,\,p} = \frac{\mathrm{N2P.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_D\; ln(1+\gamma_{BH,\,p}) - \frac{\gamma_{BH,\,p}}{(1+\gamma_{BH,\,p})}} \quad 3\text{-}249$$

Here:

$$\gamma_{BH,\,n} = \frac{\mathrm{N1N.TAS}}{n}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAN.TAS}} \quad 3\text{-}250$$

$$\gamma_{BH,\,p} = \frac{\mathrm{N1P.TAS}}{p}\cdot\left(\frac{T_L}{300}\right)^{\mathrm{ALPHAP.TAS}} \quad 3\text{-}251$$

T_L is the lattice temperature in degrees Kelvin, N_f is the fixed interface charge at the gate dielectric-silicon interface (cm^{-2}), N_A is the channel acceptor doping concentration in cm^{-3}, N_D is the channel donor doping concentration in cm^{-3}, n and p are the electron and hole concentrations per unit volume

in the inversion layer (cm^{-3}). The default parameters within each equation is defined in the MOBILITY statement. The default parameters are shown in Table 3-41.

Table 3-42. Parameters for Equations 3-233 through 3-251

Statement	Parameter	Default	Units
MOBILITY	RN.TAS	2	
MOBILITY	RP.TAS	3	
MOBILITY	BETAN	2	
MOBILITY	BETAP	1	
MOBILITY	MUBN.TAS	1150	
MOBILITY	MUBP.TAS	270	
MOBILITY	TMUBN.TAS	2.5	
MOBILITY	TMUBP.TAS	1.4	
MOBILITY	DN.TAS	3.2×10^{-9}	
MOBILITY	DP.TAS	2.35×10^{-9}	
MOBILITY	P1N.TAS	0.09	
MOBILITY	P1P.TAS	0.334	
MOBILITY	B1N.TAS	1.75	
MOBILITY	B1P.TAS	1.5	
MOBILITY	P2N.TAS	4.53×10^{-8}	
MOBILITY	PEP.TAS	3.14×10^{-7}	
MOBILITY	B2N.TAS	-0.25	
MOBILITY	B2P.TAS	-0.3	
MOBILITY	Z11N.TAS	0.0388	
MOBILITY	Z11P.TAS	0.039	
MOBILITY	Z22N.TAS	1.73×10^{-5}	
MOBILITY	Z22P.TAS	1.51×10^{-5}	
MOBILITY	ESRN.TAS	2.449×10^{7}	
MOBILITY	ESRP.TAS	1.0×10^{8}	
MOBILITY	BETAN.TAS	2	
MOBILITY	BETAP.TAS	1	
MOBILITY	N2N.TAS	1.1×10^{21}	

Table 3-42. Parameters for Equations 3-233 through 3-251			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	N2P.TAS	1.4×10^{18}	
MOBILITY	N1N.TAS	2.0×10^{19}	
MOBILITY	N1P.TAS	8.4×10^{16}	
MOBILITY	ALPHAN.TAS	2	
MOBILITY	ALPHAP.TAS	3.4	

Perpendicular Electric Field-Dependent Mobility

The Watt Model

A surface mobility model derived by J.T.Watt [173] is available in ATLAS. This mobility model is activated when the parameter SURFMOB is specified on the MODELS statement. The default model parameters are tuned to describe the measured mobilities in silicon at 300K. You can modify model parameters by using the MOBILITY statement.

The Watt model takes into consideration the following primary scattering mechanisms in the inversion layer:

1. Phonon scattering which results primarily from the interaction between two-dimensional inversion layer carriers and bulk phonons.
2. Surface roughness scattering caused by the interaction between inversion layer carriers and deviations from ideal planarity at the interface.
3. Charged impurity scattering caused by the interaction between inversion layer carriers and ions located in the oxide, at the interface, or in the bulk

The phonon and surface roughness components are functions of effective electric field. The charged impurity component is a function of the channel doping density.

The effective mobilities for electrons and holes are given by Equations 3-252 and 3-253.

$$\frac{1}{\mu_{eff,n}} = \frac{1}{\mathrm{MREF1N.WATT}}\left(\frac{1}{E_{eff,n}}\right)^{\mathrm{AL1N.WATT}} + \frac{1}{\mathrm{MREF2N.WATT}}\left(\frac{1}{E_{eff,n}}\right)^{\mathrm{AL2N.WATT}} + \frac{1}{\mathrm{MREF3N.WATT}}\left(\frac{1}{N_B}\right)^{-1}\left(\frac{1}{N_i}\right)^{\mathrm{AL3N.WATT}} \quad \text{3-252}$$

$$\frac{1}{\mu_{eff,p}} = \frac{1}{\mathrm{MREF1P.WATT}}\left(\frac{1}{E_{eff,p}}\right)^{\mathrm{AL1P.WATT}} + \frac{1}{\mathrm{MREF2P.WATT}}\left(\frac{1}{E_{eff,p}}\right)^{\mathrm{AL2P.WATT}} + \frac{1}{\mathrm{MREF3P.WATT}}\left(\frac{1}{N_B}\right)^{-1}\left(\frac{1}{N_i}\right)^{\mathrm{AL3N.WATT}} \quad \text{3-253}$$

Here, N_B is the surface trapped charge density, N_i is the inversion layer charge density and E_{eff} is the effective electric field given by:

$$E_{eff,n} = E_{\perp} + \texttt{ETAN.WATT}(E_0 - E_{\perp}) \quad \text{3-254}$$

$$E_{eff,p} = E_{\perp} + \texttt{ETAP.WATT}(E_0 - E_{\perp}) \quad \text{3-255}$$

Here, $E_{\perp}$ is the electric field perpendicular to the current flow and E_0 is the perpendicular electric field at the insulator-semiconductor interface. The equation parameters and their defaults are listed in Table 3-43.

The terms on the right side of Equations 3-252 and 3-253, describe (in order) the three scattering mechanisms previously discussed. Each component contains two constants: a pre-exponential factor and the exponent of the principal independent parameter. The charge impurity scattering component is assumed to be inversely proportional to doping density.

The expression for effective mobility contains a number of normalizing constants. These normalizing constants are included to allow easy comparison of constants. The first two terms in this mobility model are dependent on E_{eff} and represent the universal mobility-field relationship. The third term accounts for deviation from the universal relationship resulting from charged impurity scattering.

Table 3-43. User-Specifiable Parameters for Equations 3-252 - 3-255

Statement	Parameter	Default	Units
MOBILITY	ETAN.WATT	0.50	
MOBILITY	ETAP.WATT	0.33	
MOBILITY	MREF1N.WATT	481.0	cm2/(V·s)
MOBILITY	MREF1P.WATT	92.8	cm2/(V·s)
MOBILITY	MREF2N.WATT	591.0	cm2/(V·s)
MOBILITY	MREF2P.WATT	124.0	cm2/(V·s)
MOBILITY	MREF3N.WATT	1270.0	cm2/(V·s)
MOBILITY	MREF3P.WATT	534.0	cm2/(V·s)
MOBILITY	AL1N.WATT	-0.16	
MOBILITY	AL1P.WATT	-0.296	
MOBILITY	AL2N.WATT	-2.17	
MOBILITY	AL2P.WATT	-1.62	
MOBILITY	AL3N.WATT	1.07	
MOBILITY	AL3P.WATT	1.02	

Modifications to the Watt's Model

By default the Watt mobility model is a surface model that applies, only to those grid points on the silicon/ oxide interface. A modification has been added that now applies the Watt model to points below the interface. This extension to the Watt model is enabled using the MOD.WATT.N and MOD.WATT.P parameters of the MOBILITY statement.

The distance over which the model is applied can be controlled by using the YMAXN.WATT and the YMAXP.WATT parameters of the MOBILITY statement. These parameters specify the maximum value of the y-coordinate over which the model is applies below the interface for electrons and holes respectively.

The XMINN.WATT, XMINP.WATT, XMAXN.WATT and XMAXP.WATT of the MOBILITY statement can be used to limit the range of the model in the x-direction, to prevent the model from applying to the source and drain regions. The MIN.SURF parameter of the MODELS statement can also be used for this purpose. When enabled, the MIN.SURF parameter will ensure that the Watt model will only apply to minority regions.

The logical parameters EXP.WATT.N and EXP.WATT.P of the MOBILITY statement can also be used to enable an additional modification to the Watt model. When these parameters are enabled the effective normal electric field becomes a function of the depth beneath the silicon/oxide interface according to:

$$E_{\perp, n} = E_y exp \frac{-(y - y_{int})}{\texttt{YCHARN.WATT}} \quad \text{3-256}$$

$$E_{\perp, p} = E_y exp \frac{-(y - y_{int})}{\texttt{YCHARP.WATT}} \quad \text{3-257}$$

where $E_\perp$ is the perpendicular electric field, E_y is the perpendicular electric field at the interface, y is the local y-coordinate and y_{int} is the y-coordinate of the silicon/oxide interface. The YCHARN.WATT and YCHARP.WATT parameters are user-definable in the MOBILITY statement.

Table 3-44. User-Specifiable Parameters for Equations 3-194 and 3-195

Statement	Parameter	Default	Units
MOBILITY	XMINN.WATT	-1.0×10^{32}	microns
MOBILITY	XMAXN.WATT	1.0×10^{32}	microns
MOBILITY	YMAXN.WATT	-1.0×10^{32}	microns
MOBILITY	XMINP.WATT	-1.0×10^{32}	microns
MOBILITY	XMAXP.WATT	1.0×10^{32}	microns
MOBILITY	YMAXP.WATT	-1.0×10^{32}	microns
MOBILITY	YCHARN.WATT	1.0×10^{32}	microns
MOBILITY	YCHARP.WATT	1.0×10^{32}	microns

Shirahata's Mobility Model

The Shirahata Mobility Model [148] is a general purpose MOS mobility model that takes into account screening effects in the inversion layer as well as improved perpendicular field dependence for thin gate oxides. In the original paper, the authors present the model as a combination of portions of Klaassen's model for low field mobility contributions and an empirically fit expression for the perpendicular field dependent mobility in the inversion layer. In this implementation, for any given location the lesser of the low field mobility and the mobility due to the Shirahata Mobility Model is used. If the Klaassen Low-Field Model is used with the Shirahata model, the lattice scattering term in the Klaassen model is omitted.

The Shirahata Mobility Model is enabled by the SHI parameter of the MODELS statement or can be enabled individually for electrons and holes using the SHI.N and SHI.P parameters of the MOBILITY statement.

The Shirahata models for electrons and holes are given by:

$$\mu_n = \frac{\mathrm{MU0N.SHI}\left(\frac{T_L}{300}\right)^{-\mathrm{THETAN.SHI}}}{\left[1+\frac{|E_\perp|}{\mathrm{E1N.SHI}}\right]^{\mathrm{P1N.SHI}} + \left[\frac{|E_\perp|}{\mathrm{E2N.SHI}}\right]^{\mathrm{P2N.SHI}}} \quad \text{3-258}$$

$$\mu_p = \frac{\mathrm{MU0P.SHI}\left(\frac{T_L}{300}\right)^{-\mathrm{THETAP.SHI}}}{\left[1+\frac{|E_\perp|}{\mathrm{E1P.SHI}}\right]^{\mathrm{P1P.SHI}} + \left[\frac{|E_\perp|}{\mathrm{E2P.SHI}}\right]^{\mathrm{P2P.SHI}}} \quad \text{3-259}$$

where $E_\perp$ is the perpendicular electric and the equation parameters: MU0N.SHI, MU0P.SHI, E1N.SHI, E1P.SHI, E2N.SHI, E2P.SHI, P1N.SHI, P1P.SHI, P2N.SHI, P2P.SHI, THETAN.SHI and THETAP.SHI are user-definable in the MOBILITY statement (see Table 3-45 for their defaults).

Table 3-45. User-Specifiable Parameters for Equations 3-258 and 3-259

Statement	Parameter	Default	Units
MOBILITY	MU0N.SHI	1430.0	$cm^2/(V \cdot s)$
MOBILITY	MU0P.SHI	500.0	$cm^2/(V \cdot s)$
MOBILITY	E1N.SHI	6.3×10^3	V/cm
MOBILITY	E1P.SHI	8.0×10^3	V/cm
MOBILITY	E2N.SHI	0.77×10^6	V/cm
MOBILITY	E2P.SHI	3.9×10^5	V/cm
MOBILITY	P1N.SHI	0.28	
MOBILITY	P1P.SHI	0.3	
MOBILITY	P2N.SHI	2.9	

Table 3-45. User-Specifiable Parameters for Equations 3-258 and 3-259			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	P2P.SHI	1.0	
MOBILITY	THETAN.SHI	2.285	
MOBILITY	THETAP.SHI	2.247	

Note: If the maximum low field mobility has been user-defined, then it's important to also define this value inside the Shirahata model with the MU0N.SHI and MU0P.SHI parameters in the MOBILITY statement.

Parallel Electric Field-Dependent Mobility

As carriers are accelerated in an electric field their velocity will begin to saturate when the electric field magnitude becomes significant. This effect has to be accounted for by a reduction of the effective mobility since the magnitude of the drift velocity is the product of the mobility and the electric field component in the direction of the current flow. The following Caughey and Thomas Expression [32] is used to implement a field-dependent mobility. This provides a smooth transition between low-field and high field behavior where:

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}E}{\mathrm{VSATN}}\right)^{\mathrm{BETAN}}}\right]^{\frac{1}{\mathrm{BETAN}}} \quad 3\text{-}260$$

$$\mu_p(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}E}{\mathrm{VSATP}}\right)^{\mathrm{BETAP}}}\right]^{\frac{1}{\mathrm{BETAP}}} \quad 3\text{-}261$$

Here, E is the parallel electric field and μ_{n0} and μ_{p0} are the low field electron and hole mobilities respectively. The low field mobilities are either set explicitly in the MOBILITY statement or calculated by one of the low field mobility models. The BETAN and BETAP parameters are user-definable in the MOBILITY statement (see Table 3-46 for their defaults).

The saturation velocities are calculated by default from the temperature-dependent models [139]:

$$\mathrm{VSATN} = \frac{\mathrm{ALPHAN.FLD}}{1+\mathrm{THETAN.FLD}\,exp\left(\frac{T_L}{\mathrm{TNOMN.FLD}}\right)} \quad 3\text{-}262$$

$$\mathrm{VSATP} = \frac{\mathrm{ALPHAP.FLD}}{1+\mathrm{THETAP.FLD}\,exp\left(\frac{T_L}{\mathrm{TNOMP.FLD}}\right)} \quad 3\text{-}263$$

But, you can set them to constant values on the MOBILITY statement using the VSATN and VSATP parameters.

In this case, no temperature dependence is implemented. Specifying the FLDMOB parameter on the MODELS statement invokes the field-dependent mobility. FLDMOB should always be specified unless one of the inversion layer mobility models (which incorporate their own dependence on the parallel field) are specified.

You can invoke a C-INTERPRETER function for the saturation velocities. The F.VSATN and F.VSATP parameters in the MATERIAL statement can be set to provide the filenames of two text files containing the particular functions. These functions allow you to include the temperature dependence. See Appendix A: "C-Interpreter Functions" for more details.

Canali Modification

The Canali model [30] has been implemented as an alternative to using fixed values of BETAN and BETAP in the Caughey-Thomes model. This uses the exponent BETA, which depends on lattice temperature (TL), and will calculate the values of BETAN and BETAP as

$$\mathrm{BETAN} = \mathrm{N.BETA0} * (TL/300)^{(\mathrm{N.BETAEXP})} \quad \text{3-264}$$

$$\mathrm{BETAP} = \mathrm{P.BETA0} * (TL/300)^{(\mathrm{P.BETAEXP})} \quad \text{3-265}$$

The Canali model can be used for Silicon up to 430 K.

To enable the model, use the N.CANALI and P.CANALI parameters on the MOBILITY statement. You should also set the FLDMOB flag on the MODELS statement. The EVSATMOD and HVSATMOD should also have their default values.

Table 3-46. User-Definable Parameters in the Field-Dependent Mobility Model

Statement	Parameter	Default	Units
MOBILITY	ALPHAN.FLD	2.4×10^7	cm/s
MOBILITY	ALPHAP.FLD	2.4×10^7	cm/s
MOBILITY	BETAN	2.0	
MOBILITY	BETAP	1.0	
MOBILITY	N.BETAEXP	0.66	
MOBILITY	N.BETA0	1.109	
MOBILITY	N.CANALI	False	
MOBILITY	P.BETAEXP	0.17	
MOBILITY	P.BETA0	1.213	
MOBILITY	P.CANALI	False	
MOBILITY	THETAN.FLD	0.8	
MOBILITY	THETAP.FLD	0.8	
MOBILITY	TNOMN.FLD	600.0	K

Table 3-46. User-Definable Parameters in the Field-Dependent Mobility Model			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	VSATN		cm/s
MOBILITY	VSATP		cm/s
MOBILITY	TNOMP.FLD	600.0	K

Note: Equation 3-263, which was derived for the drift-diffusion approximation, ensures that velocity overshoot cannot occur. To model velocity overshoot in silicon, apply the Energy Balance Transport Model. This model follows the above implementation but with the electric field term replaced by a new "effective" field calculated from the carrier temperature see the following section for more details.

Note: BLAZE includes a different field dependent mobility model that does simulate velocity overshoot in GaAs. See Chapter 5: "Blaze: Compound Material 2D Simulator" for more information.

Carrier Temperature Dependent Mobility

The Energy Balance Transport Model allows the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated, which is the uniform electric field value that causes the carriers in a homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields, $E_{eff,n}$ and $E_{eff,p}$, are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\texttt{TAUMOB.EL}} \quad \text{3-266}$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\texttt{TAUMOB.HO}} \quad \text{3-267}$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived from the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introduced into the relevant field dependent mobility model. The TAUMOB.EL and TAUMOB.HO parameters can be set on the MODELS statement and have the default values shown in Table 3-47.

Note: The TAUMOB.EL and TAUMOB.HO parameters are distinct from the energy balance relaxation time parameters: TAUREL.EL and TAUREL.HO. But, if the E.TAUR.VAR and H.TAUR.VAR parameters are set on the MODELS statement, then TAUREL.EL and TAUREL.HO are used in place of TAUMOB.EL and TAUMOB.HO.

Table 3-47. User-Specifiable Parameters for Equations 3-266 and 3-267			
Statement	**Parameter**	**Default**	**Units**
MODELS	TAUMOB.EL	2.5×10^{-13}	s
MODELS	TAUMOB.HO	2.5×10^{-13}	s

Four different models have been implemented into the ATLAS Energy Balance Transport Model which can be chosen by the parameter EVSATMOD on the MODELS statement. These models shall be described next.

Setting EVSATMOD=0 implements, the default model for silicon based upon the Caughey-Thomas field-dependent mobility model in Equation 3-260. The resultant relationship between the carrier mobility and the carrier temperature is in the forms:

$$\mu_n = \frac{\mu_{n0}}{\left(1 + X_n^{\mathrm{BETAN}}\right)^{\frac{1}{\mathrm{BETAN}}}} \quad 3\text{-}268$$

$$\mu_p = \frac{\mu_{p0}}{\left(1 + X_p^{\mathrm{BETAP}}\right)^{\frac{1}{\mathrm{BETAP}}}} \quad 3\text{-}269$$

$$X_n^{\mathrm{BETAN}} = \frac{1}{2}\left(\alpha_n^{\mathrm{BETAN}}(T_n - T_L)^{\mathrm{BETAN}} + \sqrt{\alpha_n^{2\mathrm{BETAN}}(T_n - T_L)^{2\mathrm{BETAN}} - 4\alpha_n^{\mathrm{BETAN}}(T_n - T_L)^{\mathrm{BETAN}}}\right) \quad 3\text{-}270$$

$$X_p^{\mathrm{BETAP}} = \frac{1}{2}\left(\alpha_p\text{\^{}}\mathrm{BETAP}(T_n - T_L)^{\mathrm{BETAP}} + \sqrt{\alpha_p^{2\mathrm{BETAP}}(T_p - T_L)^{2\mathrm{BETAP}} - 4\alpha_p^{\mathrm{BETAP}}(T_p - T_L)^{\mathrm{BETAP}}}\right) \quad 3\text{-}271$$

$$\alpha_n = \frac{3}{2}\frac{k_B\mu_{n0}}{q\,\mathrm{VSATN}^2(\ \mathrm{TAUREL.EL})} \quad 3\text{-}272$$

$$\alpha_p = \frac{3}{2}\frac{k_B\mu_{p0}}{q\,\mathrm{VSATP}^2(\ \mathrm{TAUREL.HO})} \quad 3\text{-}273$$

where μ_{n0} and μ_{p0} are the low field carrier mobilities and VSATN and VSATP are the saturated velocities for electrons and holes. The VSATN, VSATP, BETAN, and BETAP parameters are user-definable in the MOBILITY statement. The terms, TAUREL.EL and TAUREL.HO, are the energy relaxation times for electrons and holes and can be defined in the MATERIAL statement.

Setting EVSATMOD=0 with the additional parameter, MOBTEM.SIMPL, allows you to apply a simplified form of the above model. This form is:

$$\mu_n = \frac{\mu_{n0}}{\sqrt{1 + \alpha_n^2 (T_n - T_L)^2}} \quad 3\text{-}274$$

$$\mu_p = \frac{\mu_{p0}}{\sqrt{1 + \alpha_p^2 (T_p - T_L)^2}} \quad 3\text{-}275$$

where μ_{n0} and μ_{p0} are again the low field carrier mobilities and $\alpha_{n,p}$ are as defined above.

Setting EVSATMOD=1 implements the GaAs carrier temperature dependent mobility model. See Chapter 5: "Blaze: Compound Material 2D Simulator" for more information about this model.

Setting EVSATMOD=2 will apply the simple velocity limiting model based upon the electric field. In other words, the temperature dependent mobility is turned off and the standard electric field based mobility model is applied.

Note: If the YAMAGUCHI or TASCH mobility models are chosen in the MODEL statement, then no energy dependence is applied. No energy dependence is included in any perpendicular electric field model, such as SHIRAHATA or SURFMOB.

Table 3-48. User-Specifiable Parameters for Equations 3-268– 3-275

Statement	Parameter	Units
MOBILITY	MUN	$cm^2/(V \cdot s)$
MOBILITY	MUP	$cm^2/(V \cdot s)$
MATERIAL	VSATN	cm/s
MATERIAL	VSATP	cm/s

Meinerzhagen-Engl Model

The Meinerzhagen-Engl model is another velocity saturation model in which the mobility depends on the carrier temperature [109].

$$\mu_n = \frac{\mu_{n0}}{\left(1 + \alpha_n^{\beta} (T_n - T_L)^{\beta}\right)^{1/\beta}} \quad 3\text{-}276$$

$$\mu_p = \frac{\mu_{n0}}{\left(1 + \alpha_p^{\beta} (T_p - T_L)^{\beta}\right)^{1/\beta}} \quad 3\text{-}277$$

where α_n and α_p are the same as in Equations 3-272 and 3-273. If β=2, then you have the same result as in Equations 3-274 and 3-275. The exponent β, however, may now depend on lattice temperature (T_L).

As in the Canali model, the values of BETAN and BETAP are calculated as

$$\mathtt{BETAN} = \mathtt{N.BETA0} * (TL/300)^{(\mathtt{N.BETAEXP})} \quad \text{3-278}$$

$$\mathtt{BETAP} = \mathtt{P.BETA0} * (TL/300)^{(\mathtt{P.BETAEXP})} \quad \text{3-279}$$

although the default values differ from the Canali model (see Table 3-49).

To activate the model, specify FLDMOB on the MODELS statement and additionally N.MEINR or P.MEINR or both on the MOBILITY statement. You must at least solve for one of the electron or hole temperatures. The exponent β is constant unless you enable Lattice temperature too.

Table 3-49. User specifiable parameters for Meinerzhagen-Engl model.

Statement	Parameter	Default
MOBILITY	N.MEINR	False
MOBILITY	P.MEINR	False
MOBILITY	N.BETA0	0.6
MOBILITY	P.BETA0	0.6
MOBILITY	N.BETAEXP	0.01
MOBILITY	P.BETAEXP	0.01

3.6.2: Mobility Model Summary

Tables 3-50 and 3-51 shows a brief description of the mobility models.

Table 3-50. Mobility Models Summary

Model	Syntax	Notes
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora's Model	ARORA	Alternative to ANALYTIC for Si.
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity saturation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.
Klaassen Model	KLA	Includes N, T, and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices
Shirahata Model	SHI	Includes N, $E_{\perp}$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_{\perp}$ effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E// and $E_{\perp}$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_{\perp}$ effects. Only for 300K.

Table 3-51. Mobility Models Summary

	CONMOB	FLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	OK		—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	[2]

Table 3-51. Mobility Models Summary											
	CONMOB	FLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE

Key to Table Entries:

MODEL ABBREVIATION = The model that supersedes when a combination is specified. In some cases, but not all, a warning message is issued when a model is ignored.

OK = This combination is allowed.

NO = This combination isn't allowed.

NOTES:

1. Uses internal model similar to FLDMOB
2. When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

3.6.3: Carrier Generation-Recombination Models

Carrier generation-recombination is the process through which the semiconductor material attempts to return to equilibrium after being disturbed from it. If we consider a homogeneously doped semiconductor with carrier concentrations n and p to the equilibrium concentrations n_0 an p_0 then at equilibrium a steady state balance exists according to:

$$n_0 p_0 = n_i^2 \qquad 3\text{-}280$$

Semiconductors, however, are under continual excitation whereby n and p are disturbed from their equilibrium states: n_0 and p_0. For instance, light shining on the surface of a p-type semiconductor causes generation of electron-hole pairs, disturbing greatly the minority carrier concentration. A net recombination results which attempts to return the semiconductor to equilibrium. The processes responsible for generation-recombination are known to fall into six main categories:

- phonon transitions
- photon transitions
- Auger transitions
- surface recombination
- impact ionization
- tunneling

The following sections describes the models implemented into ATLAS that attempts the simulation of these six types of generation-recombination mechanisms.

Shockley-Read-Hall (SRH) Recombination

Phonon transitions occur in the presence of a trap (or defect) within the forbidden gap of the semiconductor. This is essentially a two step process, the theory of which was first derived by Shockley and Read [149] and then by Hall [64]. The Shockley-Read-Hall recombination is modeled as follows:

$$R_{SRH} = \frac{pn - n_{ie}^2}{\mathrm{TAUP0}\left[n + n_{ie}\, exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right)\right] + \mathrm{TAUN0}\left[p + n_{ie}\, exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]} \qquad 3\text{-}281$$

where ETRAP is the difference between the trap energy level and the intrinsic Fermi level, T_L is the lattice temperature in degrees Kelvin and TAUN0 and TAUP0 are the electron and hole lifetimes. This model is activated by using the SRH parameter of the MODELS statement. The electron and hole lifetime parameters, TAUN0 and TAUP0, are user-definable in the MATERIAL statement. The default values for carrier lifetimes are shown in Table 3-52. Materials other than silicon will have different defaults. A full description of these parameters are given in Appendix B: "Material Systems".

Table 3-52. User-Specifiable Parameters for Equation 3-281

Statement	Parameter	Default	Units
MATERIAL	ETRAP	0	eV
MATERIAL	TAUN0	1×10^{-7}	s
MATERIAL	TAUP0	1×10^{-7}	s

Note: This model only presumes one trap level which, by default, is ETRAP=0 and it corresponds to the most efficient recombination centre. If the TRAP statement is used to define specific trap physics then separate SRH statistics are implemented as described earlier in "Trap Implementation into Recombination Models" section on page 3-17.

SRH Concentration-Dependent Lifetime Model

The constant carrier lifetimes that are used in the SRH recombination model above can be made a function of impurity concentration [133,95,55] using the following equation:

$$R_{SRH} = \frac{pn - n_{ie}^2}{\tau_p\left[n + n_{ie}exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right)\right] + \tau_n\left[p + n_{ie}exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}282$$

where:

$$\tau_n = \frac{\mathrm{TAUN0}}{\mathrm{AN} + \mathrm{BN}\left(\frac{Ntotal}{\mathrm{NSRHN}}\right) + \mathrm{CN}\left(\frac{Ntotal}{\mathrm{NSRHN}}\right)^{\mathrm{EN}}} \quad 3\text{-}283$$

$$\tau_p = \frac{\mathrm{TAUP0}}{\mathrm{AP} + \mathrm{BP}\left(\frac{Ntotal}{\mathrm{NSRHP}}\right) + \mathrm{CN}\left(\frac{Ntotal}{\mathrm{NSRHP}}\right)^{\mathrm{EP}}} \quad 3\text{-}284$$

Here, N is the local (total) impurity concentration. The TAUN0, TAUP0, NSRHN, and NSRHP parameters can be defined on the MATERIAL statement (see Table 3-53 for their default values). This model is activated with the CONSRH parameter of the MODELS statement.

Table 3-53. User-Specifiable Parameters for Equations 3-282 to 3-284			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	TAUN0	1.0×10^{-7}	s
MATERIAL	NSRHN	5.0×10^{16}	cm^{-3}
MATERIAL	TAUP0	1.0×10^{-7}	s
MATERIAL	NSRHP	5.0×10^{16}	cm^{-3}
MATERIAL	AN	1.0	
MATERIAL	AP	1.0	
MATERIAL	BN	0.0	
MATERIAL	BP	1.0	
MATERIAL	CN	0.0	
MATERIAL	CP	0.0	
MATERIAL	EN	0.0	
MATERIAL	EP	0.0	

Klaassen's Concentration Dependent Lifetime Model

The Klaassen Concentration and Temperature-dependent SRH Lifetime Model [87] is enabled by setting the KLASRH logical parameter in the MODELS statement. The lifetimes for electrons and holes in this model are given by the equations:

$$\mathrm{TAUN0}^{-1} = (\mathrm{KSRHTN}^{-1} + \mathrm{KSRHCN} \times N)\left(\frac{300}{T_L}\right)^{\mathrm{KSRHGN}} \quad \text{3-285}$$

$$\mathrm{TAUP0}^{-1} = (\mathrm{KSRHTP}^{-1} + \mathrm{KSRHCP} \times \mathrm{N})\left(\frac{300}{\mathrm{T}_L}\right)^{\mathrm{KSRHGP}} \quad \text{3-286}$$

Here, N is the local (total) impurity concentration. You can define the KSRHTN, KSRHTP, KSRHCN, KSRHCP, KSRHGN, and KSRHGP parameters in the MATERIAL statement. Their default values are given in Table 3-54.

Table 3-54. User-Specifiable Parameters for Equations 3-285 to 3-286			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	KSRHTN	2.5×10^{-3}	s
MATERIAL	KSRHTP	2.5×10^{-3}	s
MATERIAL	KSRHCN	3.0×10^{-13}	cm^3/s

Table 3-54. User-Specifiable Parameters for Equations 3-285 to 3-286			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	KSRHCP	11.76×10^{-13}	cm^3/s
MATERIAL	KSRHGN	1.77	
MATERIAL	KSRHGP	0.57	

Trap-Assisted Tunneling

In a strong electric field, electrons can tunnel through the bandgap via trap states. This trap-assisted tunneling mechanism is enabled by specifying TRAP.TUNNEL on the MODELS statement and is accounted for by modifying the Schockley-Read-Hall recombination model.

$$R_{SRH} = \frac{pn - n_{ie}^2}{\frac{\mathrm{TAUP0}}{1 + \Gamma_p^{DIRAC}}\left[n + n_{ie}exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right)\right] + \frac{\mathrm{TAUN0}}{1 + \Gamma_n^{DIRAC}}\left[p + n_{ie}exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}287$$

Here, Γ_n^{DIRAC} is the electron field-effect enhancement term for Dirac wells, and Γ_p^{DIRAC} is the hole field-effect enhancement term for Dirac wells. Γ_n^{DIRAC} and Γ_p^{DIRAC} are defined in Equations 3-80 and 3-81.

Poole-Frenkel Barrier Lowering for Coulombic Wells

The Poole-Frenkel effect can enhance the emission rate for Coulombic wells. *I* the TRAP.COULOMBIC parameter is specified in the MODELS statement, the Schockley-Read-Hall electron and hole recombination model becomes:

$$R_{n,\,SRH} = \frac{p_n - n_{ie}^2}{\frac{\mathrm{TAUP0}}{1 + \Gamma_p^{DIRAC}}\left[n + n_{ie}exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right) + \frac{\mathrm{TAUN0}}{\chi_F + \Gamma_n^{COUL}\left[p + n_{ie}exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]}\right]}. \quad 3\text{-}288$$

$$R_{p,\,SRH} = \frac{p_n - n_{ie}^2}{\frac{\mathrm{TAUP0}}{\chi_F + \Gamma_n^{COUL}}\left[n + n_{ie}exp\left(\frac{\mathrm{ETRAP}}{kT_L}\right) + \frac{\mathrm{TAUN0}}{1 + \Gamma_p^{DIRAC}\left[p + n_{ie}exp\left(\frac{-\mathrm{ETRAP}}{kT_L}\right)\right]}\right]} \quad 3\text{-}289$$

The Poole-Frenkel thermal emission factor, χ_F, is defined in Equation 3-88. The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_P^{COUL} are defined in Equations 3-90 and 3-91.

Optical Generation/Radiative Recombination

The next physical mechanisms we have to consider for generation/recombination are photon transition. This mechanism occurs primarily in one step and is therefore a direct generation/ recombination mechanism. There are two partial processes involved. For radiative recombination, an electron loses energy on the order of the band gap and moves from the conduction band to the valence band. For optical generation, an electron moves from the valence band to the conduction. In silicon, band to band generation/recombination is insignificant. This effect, however, is important for narrow gap semiconductors and semiconductors whose specific band structure allows direct transitions. By assuming a capture rate C_c^{OPT} and an emission rate C_e^{OTP}, the involved partial processes can be written as

$$R_{np}^{OPT} = C_c^{OPT} np, \quad \text{3-290}$$

for recombination and

$$G_{np}^{OPT} = C_e^{OPT} \quad \text{3-291}$$

for generation.

These rates must be equal in thermal equilibrium so that

$$C_{np}^{OPT} = C_c^{OPT} n_{ie}^2 \ . \quad \text{3-292}$$

The total band to band generation/recombination is the difference of the partial rates, which equates to

$$R_{np}^{OPT} = C_c^{OPT} (np - n_{ie}^2) . \quad \text{3-293}$$

In ATLAS, COPT is used express C_c^{OPT} and can be defined away from default values on the materials statement or implemented using a C-Interpreter routine. To turn on the optical recombination/ generation model, define the OPTR keyword on the MODELS statement.

Auger Recombination

Auger recombination occurs through a three particle transition whereby a mobile carrier is either captured or emitted. The underlying physics for such processes is unclear and normally a more qualitative understanding is sufficient [142].

Standard Auger Model

Auger Recombination is commonly modeled using the expression [47]:

$$R_{Auger} = \text{AUGN} \left(pn^2 - nn_{ie}^2 \right) + \text{AUGP} \left(np^2 - pn_{ie}^2 \right) \quad \text{3-294}$$

where the model parameters AUGN and AUGP are user-definable in the MATERIAL statement (see Table 3-55 for its default value). You can activate this model with the AUGER parameter from the MODELS statement.

Table 3-55. User-Specifiable Parameters for Equation 3-294			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	AUGN	8.3×10^{-32}	cm^6/s
MATERIAL	AUGP	1.8×10^{-31}	cm^6/s

Klaassen's Temperature-Dependent Auger Model

The Klaassen Auger Recombination Model [88] is activated by specifying the KLAAUG parameter of the MODELS statement. The form of this model is:

$$R_{Auger} = C_n\left(pn^2 - nn_{ie}^2\right) + C_p\left(np^2 - pn_{ie}^2\right) \quad 3\text{-}295$$

where the Auger coefficients are temperature dependent according to:

$$C_n = \text{KAUGCN}\left(\frac{T_L}{300}\right)^{\text{KAUGDN}} \quad 3\text{-}296$$

$$C_p = \text{KAUGCP}\left(\frac{T_L}{300}\right)^{\text{KAUGDP}} \quad 3\text{-}297$$

Here, the KAUGCN, KAUGCP, KAUGDN, and KAUGDP parameters are user-definable in the MATERIAL statement and have the defaults shown in Table 3-56.

Table 3-56. User-Specifiable Parameters for Equation 3-296 and 3-297			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	KAUGCN	1.83×10^{-31}	cm^6/s
MATERIAL	KAUGCP	2.78×10^{-31}	cm^6/s
MATERIAL	KAUGDN	1.18	
MATERIAL	KAUGDP	0.72	

Narrow Bandgap Auger Model

An alternative model for the Auger recombination coefficients that is more suitable for modelling Auger processes in narrow bandgap semiconductors can be enabled by setting the parameters AUGKN and AUGKP on the MODELS statement. The model in ATLAS is a simplification of that by Beattie [21] and takes the form:

$$R_{Auger} = C_n\left(pn^2 - nn_{ie}^2\right) + C_p\left(np^2 - pn_{ie}^2\right) \quad 3\text{-}298$$

where the Auger coefficients are concentration dependent according to:

$$C_n = \frac{\texttt{AUGN}}{1 + \texttt{AUGKN}\ n} \quad \text{3-299}$$

$$C_p = \frac{\texttt{AUGP}}{1 + \texttt{AUGKP}\ p} \quad \text{3-300}$$

Here, n and p are the electron and hole carrier concentrations and the new parameters, AUGKN and AUGKP, are user-definable on the MATERIALS statement.

Surface Recombination

In addition to generation-recombination within the bulk of the semiconductor, electrons or holes may recombine or be generated at interfaces. The rate of surface recombination may be even greater than within the bulk. The standard method is to model interface recombination in a similar manner as the bulk generation-recombination rate [61] where:

$$R_{surf} = \frac{pn - n_{ie}^2}{\tau_p^{eff}\left[n + n_{ie} exp\ \left(\frac{\texttt{ETRAP}}{kT_L}\right)\right] + \tau_n^{eff}\left[p + n_{ie} exp\left(\frac{-\texttt{ETRAP}}{kT_L}\right)\right]} \quad \text{3-301}$$

Here:

$$\frac{1}{\tau_n^{eff}} = \frac{1}{\tau_n^i} + \frac{d_i}{A_i}\texttt{S.N} \quad \text{3-302}$$

and

$$\frac{1}{\tau_p^{eff}} = \frac{1}{\tau_p^i} + \frac{d_i}{A_i}\texttt{S.P} \quad \text{3-303}$$

τ^i_n is the bulk lifetime calculated at node i along the interface and which may be a function of the impurity concentration as well. The d_i and A_i parameters are the length and area of the interface for node *i*. The S.N and S.P parameters are the recombination velocities for electrons and holes respectively, which are user-definable in the INTERFACE statement. The X.MIN, X.MAX, Y.MIN, and Y.MAX parameters can also be set in the INTERFACE statement to define the region, where the specified values of the surface recombination velocities apply. This model is activated by the presence of the recombination velocities in the INTERFACE statement.

Table 3-57. User-Specifiable Parameters for Equations 3-302 to 3-303

Statement	Parameter	Default	Units
INTERFACE	S.N	0	cm/s
INTERFACE	S.P	0	cm/s

3.6.4: Impact Ionization Models

In any space charge region with a sufficiently high reverse bias, the electric field will be high enough to accelerate free carriers up to a point where they will have acquired sufficient energy to generate more free carriers when in collision with the atoms of the crystal. In order to acquire sufficient energy, two principle conditions must be met.

First, the electric field must be sufficiently high. Then, the distance between the collisions of the free carrier must be enough to allow acceleration to a sufficiently high velocity

In other words, the carrier must gain the ionization energy E_i between collisions. If the generation rate of these free carriers is sufficiently high this process will eventually lead to avalanche breakdown.

The general impact ionization process is described by the Equation 3-304.

$$G = \alpha_n \left|\vec{J}\right|_n + \alpha_p \left|\vec{J}\right|_p \quad \text{3-304}$$

Here, G is the local generation rate of electron-hole pairs, $\alpha_{n,p}$ are the ionization coefficient for electrons and holes and $J_{n,p}$ are their current densities. The ionization coefficient represents the number of electron-hole pairs generated by a carrier per unit distance travelled. The accurate calculation of this parameter has been researched because it is vital if the effects related to impact ionization, such as substrate current and device breakdown, are to be simulated. These models can be classified into two main types: local and non-local models.

The former assume that ionization at any particular point within the device is a function only of the electric field at that position. Non-local models, however, perform a more rigorous approach by taking into account the energy that the carrier gains.

Geometrical Considerations for Impact Ionization Models

In all the available models discussed in the following sections, the ionization coefficients are dependent on electric field. There are several different ways to consider the interaction between electric field and current implied by Equation 3-304.

During simulation, currents and fields are calculated both as scalar values on the edges of triangles and as vector quantities on the triangles themselves. ATLAS allows three different ways of looking at Equation 3-304.

The first model uses Equation 3-305.

$$G = \alpha_n(\left|E_{tri}\right|)J_{ntri} + \alpha_p(\left|E_{tri}\right|)J_{ptri} \quad \text{3-305}$$

Here, E_{tri} is the vector field on the triangle, J_{ntri} is the electron current vector on the triangle, and J_{ptri} is the hole current vector on the triangle. In ATLAS3D, electric field is combined with the z-component of the field at each corner of the prism to give an overall field modules at each node. To select this model, specify `E.VECTOR` of the `IMPACT` statement.

A simpler model can be selected by specifying `E.SIDE` on the `IMPACT` statement (see Equation 3-306).

$$G = \alpha_n(\left|E_{side}\right|)J_{nside} + \alpha_p(\left|E_{side}\right|)J_{pside} \quad \text{3-306}$$

Here, E_{side} is the scalar field along the side of a triangle, J_{nside} is the electron current along the side and J_{pside} is the hole current along the side. This model is the most non-physical but has the advantages of better robustness and calculation speed and is compatible with older device simulators. This model is only supported in ATLAS2D.

The most complex and physically sound model is selected by specifying E.DIR on the IMPACT statement (see Equation 3-307).

$$G = \alpha_n\left(\left|\frac{E_{tri}\cdot J_{ntri}}{J_{ntri}}\right|\right)J_{ntri} + \alpha_p\left(\left|\frac{E_{tri}\cdot J_{ntri}}{J_{ntri}}\right|\right)J_{ntri} \quad 3\text{-}307$$

In this model, the ionization coefficients are a function of the field in the direction of the current. If the dot product of E and J is negative, then the field component is taken as 0. Consequently, impact ionization may only occur when a current is dominated by the drift term. This model is the most physically sound and is the default model for the field dependance of the impact ionization coefficients.

Another option is to abandon the use of the Electric field and adopt the gradient of the Quasi-Fermi levels to use when calculating the ionization coefficients.

$$G = \alpha_n(\left|\underline{\nabla}\phi_n\right|)J_n + \alpha_p(\left|\underline{\nabla}\phi_p\right|)J_p \quad 3\text{-}308$$

The modulus of the quasi-Fermi level gradients across each triangle are used, which is similar to the E.VECTOR electric field model. Using the gradient of quasi-Fermi level has the advantage that built-in electric fields (such as those existing across highly doped *n-p* junctions) do not result in an artificially high ionization rate at low contact biases. In the situation where the current is dominated by drift rather than diffusion, the gradient of quasi-fermi level will be essentially equal to the electric field. The impact ionization coefficients calculated from gradient of quasi-fermi level are in that case the same as the E.DIR or E.VECTOR options. To enable this model, specify GRADQFL on the IMPACT statement.

Local Electric Field Models for Impact Ionization

Selberherr's Impact Ionization Model

The ionization rate model proposed by Selberherr [142] is a variation of the classical Chynoweth model [39]. Activate this model by using the SELB parameter of the IMPACT statement, which is based upon the following expressions [165]:

$$\alpha_n = \mathrm{AN}\,exp\left[-\left(\frac{\mathrm{BN}}{E}\right)^{\mathrm{BETAN}}\right] \quad 3\text{-}309$$

$$\alpha_p = \mathrm{AP}\,exp\left[-\left(\frac{\mathrm{BP}}{E}\right)^{\mathrm{BETAP}}\right] \quad 3\text{-}310$$

Here, E is the electric field in the direction of current flow at a particular position in the structure and the parameters AN, AP, BN, BP, BETAN, and BETAP are defined on the IMPACT statement and have the default values shown in Table 3-58. In the case of AN, AP, BN, and BP you can define a value of electric field, EGRAN V/cm, where for electric fields, >EGRAN V/cm, the parameters are: AN1, AP1, BN1, BP1, while for electric fields, <EGRAN V/cm, the parameters become AN2, AP2, BN2, and BP2.

The AN and BN parameters are also a function of the lattice temperature in this model [103]. The temperature dependence of these coefficients is defined as follows:

$$\mathrm{AN} = AN_{1,2}\left(1 + \mathrm{A.NT}\left[\left(\frac{T_L}{300}\right)^{\mathrm{M.ANT}} - 1\right]\right) \quad 3\text{-}311$$

$$\mathrm{AP} = AP_{1,2}\left(1 + \mathrm{A.PT}\left[\left(\frac{T_L}{300}\right)^{\mathrm{M.APT}} - 1\right]\right) \quad 3\text{-}312$$

$$\text{BN} = BN_{1,2}\left(1 + \text{B.NT}\left[\left(\frac{T_L}{300}\right)^{\text{M.BNT}} - 1\right]\right) \quad 3\text{-}313$$

$$\text{BP} = BP_{1,2}\left(1 + \text{B.PT}\left[\left(\frac{T_L}{300}\right)^{\text{M.BPT}} - 1\right]\right) \quad 3\text{-}314$$

The parameters associated with these equations are shown in Table 3-59.

An alternative model for temperature dependence of AN and AP is given by the following expressions:

$$\text{AN} = AN_{1,2} + \text{CN2} \cdot \text{T} + \text{DN2} \cdot \text{T}^2 \quad 3\text{-}315$$

$$\text{AP} = AP_{1,2} + \text{CP2} \cdot \text{T} + \text{DP2} \cdot \text{T}^2 \quad 3\text{-}316$$

where T is temperature and CN2, CP2, DN2, and DP2 are user-specifiable parameters on the IMPACT statement. By default, the temperature model of Equations 3-281 and 3-282 are used and the values of CN2, CP2, DN2 and DP2 are all zero. You can use the temperature dependence models described in Equations 3-312 or 3-313 or both by specifying non-zero values for CN2, CP2, DN2 and DP2.

The critical fields given by BN and BP may be modeled based on band gap and optical phonon mean free paths using the following expressions:

$$\text{BN} = \frac{E_g}{q\lambda_n^0} \quad 3\text{-}317$$

$$\text{BP} = \frac{E_g}{q\lambda_p^0} \quad 3\text{-}318$$

where $q\lambda_n^0$ and $q\lambda_p^0$ are the optical phonon mean free paths for electrons and holes and Eg is the local temperature dependent band gap. The free paths are modeled using the following expressions:

$$\lambda_n^o = \text{LAMDAH}\frac{tanh[q\text{OPPHE}/2kT_L]}{tanh[q\text{OPPHE}/2k300]} \quad 3\text{-}319$$

$$\lambda_n^o = \text{LAMDAE}\frac{tanh[q\text{OPPHE}/2kT_L]}{tanh[q\text{OPPHE}/2k300]} \quad 3\text{-}320$$

where T is the lattice temperature and LAMDAE, LAMDAH, OPPHE are user-specifiable parameters listed in Table 3-60. To enable the models described by Equations 3-314, 3-315, 3-316 and 3-317, either specify $\text{BN}_{1,2}$ or $\text{BP}_{1,2}$ or both as zero.

Table 3-58. User-Definable Parameters in the Selberherr Impact Ionization Model

Statement	Parameter	Default
IMPACT	AN1	7.03×10^{5} cm^{-1}
IMPACT	AN2	7.03×10^{5} cm^{-1}
IMPACT	AP1	6.71×10^{5} cm^{-1}
IMPACT	AP2	1.58×10^{6} cm^{-1}
IMPACT	BN1	1.231×10^{6}V/cm
IMPACT	BN2	1.231×10^{6} V/cm
IMPACT	BP1	1.693×10^{6} V/cm
IMPACT	BP2	2.036×10^{6} V/cm
IMPACT	BETAN	1.0
IMPACT	BETAP	1.0
IMPACT	EGRAN	4×10^{5} V/cm

Table 3-59. Temperature Coefficient Parameters of the Selberherr Impact Ionization Model for Silicon in Equations 3-311 to 3-314

Statement	Parameter	Default
IMPACT	A.NT	0.588
IMPACT	B.NT	0.248
IMPACT	A.PT	0.588
IMPACT	B.PT	0.248
IMPACT	M.ANT	1.0
IMPACT	M.BNT	1.0
IMPACT	M.APT	1.0
IMPACT	M.BPT	1.0

Table 3-60. User specifiable parameters for the optical phonon mean free path model in Equations 3-319 and 3-320.

Statement	Parameter	Default	Units
IMPACT	LAMDAE	6.2e-7	cm
IMPACT	LAMDAH	3.8e-7	cm
IMPACT	OPPHE	0.063	eV

Van Overstraeten - de Man Impact Ionization Model

Based on the Chynoweth law [39], this model is very similar to the Selberherr model. The differences, however, are that the exponents BETAN and BETAP are set equal to unity. The lattice temperature dependence of the coefficients is also different. The functional forms of the ionization rate are

$$\alpha_n = \gamma_n \mathrm{AN} exp\left[\frac{-\gamma_n \mathrm{AN}}{E}\right] \quad 3\text{-}321$$

$$\alpha_p = \gamma_p \mathrm{AN} exp\left[\frac{-\gamma_p \mathrm{AP}}{E}\right] \quad 3\text{-}322$$

where the γ factors depend on Lattice temperature, or device temperature if GIGA is not enabled. They are calculated as

$$\gamma_n = \frac{tanh\left(\frac{\mathrm{N.VANHW}}{2K300}\right)}{tanh\left(\frac{\mathrm{N.VANHW}}{2KT}\right)} \quad 3\text{-}323$$

$$\gamma_p = \frac{tanh\left(\frac{\mathrm{P.VANHW}}{2K300}\right)}{tanh\left(\frac{\mathrm{P.VANHW}}{2KT}\right)} \quad 3\text{-}324$$

and will be unity if the device is at a uniform 300 K. The same parameters are used as in the Selberherr model but the BETAN and BETAP are fixed at unity. The other parameters are in Table 3-61. To activate the model, use the VANOVERS parameter on the IMPACT statement.

Table 3-61. van Overstraeten - de Man model Parameters

Statement	Parameter	Default	Units
IMPACT	N.VANHW	0.063	eV
IMPACT	P.VANHW	0.063	eV
IMPACT	VANOVERS	False	--

Valdinoci Impact Ionization Model

Valdinoci et al. [163] reported on a measurement technique to calibrate the temperature dependence of impact ionization models and proposed a new model based on their study. Their model of both electron and hole ionization coefficients was calibrated for silicon for temperature ranging from 25 to 400°C. This model is based on the following:

$$\alpha_{n,p} = \frac{E}{a_{n,p}(T_L) + b_{n,p}(T_L) exp[d_{n,p}(T_L)/(E + c_{n,p}(T_L))]} \quad 3\text{-}325$$

where E is the electric field along the current flow lines. The parameters in Equation 3-325 depend on temperature as follows:

$$a_n(T_L) = \mathrm{VAL.AN0} + \left(\mathrm{VAL.AN1} \cdot T_L^{\mathrm{VAL.AN2}}\right) \quad (3\text{-}326)$$

$$b_n(T_L) = \mathrm{VAL.BN0}\, exp\left(\mathrm{VAL.BN1} \cdot T_L\right) \quad (3\text{-}327)$$

$$c_n(T_L) = \mathrm{VAL.CN0} + \left(\mathrm{VAL.CN1} \cdot T_L^{\mathrm{VAL.CN2}}\right) + \mathrm{VAL.CN3} \cdot T_L^2 \quad (3\text{-}328)$$

$$d_n(T_L) = \mathrm{VAL.DN0} + \mathrm{VAL.DN1} \cdot T_L + \mathrm{VAL.DN2} \cdot T_L^2 \quad (3\text{-}329)$$

$$a_p(T_L) = \mathrm{VAL.AP0} + \left(\mathrm{VAL.AP1} \cdot T_L^{\mathrm{VAL.AP2}}\right) \quad (3\text{-}330)$$

$$b_p(T_L) = \mathrm{VAL.BP0}\, exp\left(\mathrm{VAL.BP1} \cdot T_L\right) \quad (3\text{-}331)$$

$$c_p(T_L) = \mathrm{VAL.CP0} + \left(\mathrm{VAL.CP1} \cdot T_L^{\mathrm{VAL.CP2}}\right) + \mathrm{VAL.CP3} \cdot T_L^2 \quad (3\text{-}332)$$

$$d_p(T_L) = \mathrm{VAL.DP0} + \mathrm{VAL.DP1} \cdot T_L + \mathrm{VAL.DP2} \cdot T_L^2 \quad (3\text{-}333)$$

Our default parameters for silicon are shown in Table 3-62. These parameters are also taken from Valdinoci et al. [163]. To enable this model, specify the logical parameter, VALDINOCI, in the IMPACT statement.

Table 3-62. Default Parameters for Valdinoci Impact Ionization Model

Statement	Parameter	Type	Default	Units
IMPACT	VAL.AN0	Real	4.3383	
IMPACT	VAL.AN1	Real	-2.42e-12	
IMPACT	VAL.AN2	Real	4.1233	
IMPACT	VAL.BN0	Real	0.235	
IMPACT	VAL.BN1	Real	0.0	
IMPACT	VAL.CN0	Real	1.6831e4	
IMPACT	VAL.CN1	Real	4.3796	
IMPACT	VAL.CN2	Real	1.0	
IMPACT	VAL.CN3	Real	0.13005	

Table 3-62. Default Parameters for Valdinoci Impact Ionization Model				
Statement	**Parameter**	**Type**	**Default**	**Units**
IMPACT	VAL.DN0	Real	1.233735e6	
IMPACT	VAL.DN1	Real	1.2039e3	
IMPACT	VAL.DN2	Real	0.56703	
IMPACT	VAL.AP0	Real	2.376	
IMPACT	VAL.AP1	Real	0.01033	
IMPACT	VAL.AP2	Real	1.0	
IMPACT	VAL.BP0	Real	0.17714	
IMPACT	VAL.BP1	Real	-0.002178	
IMPACT	VAL.CP0	Real	0.0	
IMPACT	VAL.CP1	Real	0.00947	
IMPACT	VAL.CP2	Real	2.4924	
IMPACT	VAL.CP3	Real	0.0	
IMPACT	VAL.DP0	Real	1.4043e6	
IMPACT	VAL.DP1	Real	2.9744e3	
IMPACT	VAL.DP2	Real	1.4829	

Zappa's Model for Ionization Rates in InP

A model for temperature and field dependent ionization rates in InP was proposed by Zappa et. al. [192]. To enable this model, specify ZAPPA in the IMPACT statement. The model is described for electrons and holes by Equations 3-334 through 3-391:

$$\alpha_n = \frac{qE}{\mathrm{ZAP.AN}} exp\left\{ 0.217\left(\frac{\mathrm{ZAP.AN}}{E_n}\right)^{1.14} - \left[\left(0.217\left(\frac{\mathrm{ZAP.AN}}{E_n}\right)^{1.14}\right)^2 + \left(\frac{\mathrm{ZAP.AN}}{qE\lambda_n}\right)^2\right]^{0.5}\right\} \quad 3\text{-}334$$

$$\lambda_n = \mathrm{ZAP.BN} tanh\left(\frac{\mathrm{ZAP.CN}}{2kT}\right) \quad 3\text{-}335$$

$$E_n = \mathrm{ZAP.DN} tanh\left(\frac{\mathrm{ZAP.EN}}{2kT}\right) \quad 3\text{-}336$$

$$\alpha_p = \frac{qE}{\mathrm{ZAP.AP}} exp\left\{ 0.217\left(\frac{\mathrm{ZAP.AP}}{E_p}\right)^{1.14} - \left[\left(0.217\left(\frac{\mathrm{ZAP.AP}}{E_p}\right)^{1.14}\right)^2 + \left(\frac{\mathrm{ZAP.AP}}{qE\lambda_p}\right)^2\right]^{0.5}\right\} \quad 3\text{-}337$$

$$\lambda_p = \texttt{ZAP.BP} tanh\left(\frac{\texttt{ZAP.CP}}{2kT}\right) \quad 3\text{-}338$$

$$E_p = \texttt{ZAP.DP} tanh\left(\frac{\texttt{ZAP.EP}}{2kT}\right) \quad 3\text{-}339$$

where E is the local electric field and T is the local temperature. Table 3-63 describes the user-specifiable parameters for Zappa's model.

Table 3-63. User-Specifiable parameters for Zappa's Ionization Rate Model

Statement	Parameter	Type	Default	Units
IMPACT	ZAP.AN	Real	1.9	eV
IMPACT	ZAP.BN	Real	41.7	angstroms
IMPACT	ZAP.CN	Real	46.0	meV
IMPACT	ZAP.DN	Real	46.0	meV
IMPACT	ZAP.EN	Real	46.0	meV
IMPACT	ZAP.AP	Real	1.4	eV
IMPACT	ZAP.BP	Real	41.3	angstroms
IMPACT	ZAP.CP	Real	36.0	meV
IMPACT	ZAP.DP	Real	36.0	meV
IMPACT	ZAP.EP	Real	36.0	meV

Grant's Impact Ionization Model

The second ionization model has the same form as the Selberherr model but a simpler implementation where:

$$\alpha_n = \texttt{AN} exp\left[-\left(\frac{\texttt{BN}}{E}\right)\right] \quad 3\text{-}340$$

$$\alpha_p = \texttt{AP} exp\left[-\left(\frac{\texttt{BP}}{E}\right)\right] \quad 3\text{-}341$$

This implementation has three key differences:

- The model has a low field, an intermediate field and a high field region.
- The coefficients for silicon are different.
- There is no temperature dependence.

This model was developed after investigations by Baraff [19] suggested the existence of a low, intermediate and high field response region for electron and hole ionization rates. The coefficients implemented into this model match the experimental data of Grant [48], which suggested that the three different regions existed.

This model is activated with the GRANT parameter of the IMPACT statement. The model parameters: AN, AP, BN, and BP aren't user-definable. Instead, the three electric field regions have in-built values as follows:

1) **Low Electric Field** $E < 2.4\times10^5\ V/cm$ 3-342

$AN = 2.6\times10^6 \quad AP = 2.0\times10^6$

$BN = 1.43\times10^6 \quad BN = 1.97\times10^6$

2) **Intermediate Electric Field** $2.4\times10^5 > E > 5.3\times10^5\ V/cm$ 3-343

$AN = 6.2\times10^5 \quad AP = 2.0\times10^6$

$BN = 1.08\times10^6 \quad BP = 1.97\times10^6$

3) **High Electric Field** $E > 5.3\times10^5\ V/cm$ 3-344

$AN = 5.0\times10^5 \quad AP = 5.6\times10^5$

$BN = 9.9\times10^6 \quad BP = 1.32\times10^6$

Crowell-Sze Impact Ionization Model

Crowell and Sze [43] have proposed a more physical relationship between the electric field and the ionization rates. This model represents ionization coefficients as follows:

$$\alpha_{n,\,p} = \frac{1}{\lambda_{n,\,p}} exp\left[C_o(r) + C_1(r)x + C_2(r)x^2\right] \quad 3\text{-}345$$

$$C_0 = -1.92 + 75.5r - 75.7r^2 \quad 3\text{-}346$$

$$C_1(r) = -1.75\times10^{-2} - 11.9r + 46r^2 \quad 3\text{-}347$$

$$C_2(r) = 3.9\times10^{-4} - 1.17r + 11.5r^2 \quad 3\text{-}348$$

where:

$$r = \frac{\mathrm{OPPHE}}{E_i};\quad x = \frac{E_i}{q\lambda_{n,\,p}E} \quad 3\text{-}349$$

$$E_i = \begin{cases} 1.1eV & for\ electrons \\ 1.8eV & for\ holes \end{cases} \quad 3\text{-}350$$

$$\lambda_n^o = \mathrm{LAMDAE}\frac{tanh[q\,\mathrm{OPPHE}/2kT_L]}{tanh[q\,\mathrm{OPPHE}/2k300]} \quad 3\text{-}351$$

$$\lambda_n^o = \mathrm{LAMDAH}\frac{tanh[qE_r/2kT_L]}{tanh[qE_r/2k300]} \quad 3\text{-}352$$

The Crowell-Sze Model for impact ionization is selected by setting the CROWELL parameter of the IMPACT statement.

Table 3-64. Crowell-Sze Impact Ionization Model Parameters.		
Statement	**Parameter**	**Default**
IMPACT	LAMDAE	6.2×10^{-7}cm
IMPACT	LAMDAH	3.8×10^{-7} cm

Okuto-Crowell Impact Ionization Model

This is another empirical model with temperature dependent coefficients where the ionization co-efficients take the following forms:

$$\alpha_n = \mathrm{ANOKUTO}(1 + \mathrm{CNOKUTO}(T - 300)) \cdot F^{\mathrm{NGAMOKUTO}} \times exp\left(-\left(\frac{\mathrm{BNOKUTO}(1 + \mathrm{DNOKUTO}(T - 300))}{F}\right)^{\mathrm{NDELOKUTO}}\right) \quad \text{3-353}$$

$$\alpha_p = \mathrm{APOKUTO}(1 + \mathrm{CPOKUTO}(T - 300)) \cdot F^{\mathrm{PGAMOKUTO}} \times exp\left(-\left(\frac{\mathrm{BPOKUTO}(1 + \mathrm{DPOKUTO}(T - 300))}{F}\right)^{\mathrm{PDELOKUTO}}\right) \quad \text{3-354}$$

where F is the effective electric field.

The parameters are optimized to fit in the Electric Field Range 10^5 to 10^6 V/cm but the same ones are used for the whole range of field. It uses a relatively large set of parameters and so you can easily adjust the default values (applicable to silicon) to fit other materials [116].

To enable the model, use the OKUTO parameter on the IMPACT statement. The adjustable model parameters are given in Table 3-65.

Table 3-65. Okuto-Crowell Model Parameters			
Statement	**Parameter**	**Default**	**Units**
IMPACT	ANOKUTO	0.426	/V
IMPACT	APOKUTO	0.243	/V
IMPACT	BNOKUTO	4.81×10^5	V/cm
IMPACT	BPOKUTO	6.53×10^5	V/cm
IMPACT	CNOKUTO	3.05×10^{-4}	/K
IMPACT	CPOKUTO	5.35×10^{-4}	/K

Table 3-65. Okuto-Crowell Model Parameters			
Statement	**Parameter**	**Default**	**Units**
IMPACT	DNOKUTO	6.86×10^{-4}	/K
IMPACT	DPOKUTO	5.67×10^{-4}	/K
IMPACT	NGAMOKUTO	1.0	
IMPACT	PGAMOKUTO	1.0	
IMPACT	NDELOKUTO	2.0	
IMPACT	PDELOKUTO	2.0	

Note: If NGAMOKUTO or PGAMOKUTO is different from 1.0, the units of ANOKUTO or APOKUTO will be different from /V.

Lackner Impact Ionization Model

This is another, more recent, impact ionization model based on Chynoweth's law [91]. It is similar to the van Overstraeten - de Man model but with an extra factor Z, dividing the ionisation rate. The factor Z and the ionization coefficients are

$$Z = 1 + \frac{\mathrm{GAMMAN.BNLACKNER}}{F} exp\left(-\frac{\mathrm{GAMMAN.BNLACKNER}}{F}\right) + \frac{\mathrm{GAMMAP.BPLACKNER}}{F} exp\left(-\frac{\mathrm{GAMMAP.BPLACKNER}}{F}\right) \quad 3\text{-}355$$

$$\alpha_n = \frac{\mathrm{GAMMAN.ANLACKNER}}{Z} exp\left(-\frac{\mathrm{GAMMAN.BNLACKNER}}{F}\right) \quad 3\text{-}356$$

$$\alpha_p = \frac{\mathrm{GAMMAP.APLACKNER}}{Z} exp\left(-\frac{\mathrm{GAMMAP.BPLACKNER}}{F}\right) \quad 3\text{-}357$$

$$\mathrm{GAMMAN} = \frac{tanh\left(\frac{\mathrm{N.LACKHW}}{2K300}\right)}{tanh\left(\frac{\mathrm{N.LACKHW}}{2KT}\right)}, \mathrm{GAMMAP} = \frac{tanh\left(\frac{\mathrm{P.LACKHW}}{2K300}\right)}{tanh\left(\frac{\mathrm{P.LACKHW}}{2KT}\right)} \quad 3\text{-}358$$

To enable the model, use the LACKNER parameter on the IMPACT statement. The adjustable model parameters are given in Table 3-66.

Table 3-66. LACKNER Model Parameters			
Statement	**Parameter**	**Default**	**Units**
IMPACT	ANLACKNER	1.316×10^6	/cm
IMPACT	APLACKNER	1.818×10^6	/cm
IMPACT	BNLACKNER	1.474x10^6	V/cm
IMPACT	BPLACKNER	2.036x10^6	V/cm
IMPACT	N.LACKHW	0.063	eV
IMPACT	P.LACKHW	0.063	eV

Non-Local Carrier Energy Models for Impact Ionization

All local electric field based models will normally overestimate the rate of impact ionization. This occurs because lucky electron theory inherently assumes that a carrier is traveling through a constant electric field E. As a result it will predict a distance $\Delta x=E_i/qE$ over which the carrier will gain the ionization energy E_i. In real devices, however, the electric field is never constant but is normally sharply peaked at metallurgical junctions. Therefore, as a carrier passes through the peaked electric field the lucky electron model will predict the ionization distance Δx to be too small. As a result the ionization rate is overestimated. The effect of this is that all the simulated breakdown voltages will be underestimated and substrate currents overestimated.

The Energy Balance Transport Model can be used to improve the simulation of impact ionization by implementing ionization models based upon the carrier temperature instead of the electric field. The carrier temperature is a more meaningful basis as the velocity-field relationship is more closely modeled. This allows a non-local dependence on the electric field within the impact ionization model. Energy Balance Transport Models will therefore result in more accurate simulations of breakdown voltage and substrate current. Two different impact ionization models have been implemented into ATLAS, the first is based upon the classical Chynoweth relationship, modified to include carrier temperature, but the second is a more advanced non-Maxwellian approach based upon carrier temperatures.

When the energy balance transport model is applied, only two impact ionization models are available. These are the Toyabe model and the Concannon model.

Toyabe Impact Ionization Model

The temperature dependent impact ionization model is founded on the Selberherr model and is similar to that suggested by Toyabe [80]. The carrier temperature is used to calculate an effective electric field based upon the homogeneous temperature-field relation. To maintain self-consistency within the Energy Balance Transport Model this is the same relationship used for the effective electric field within the carrier temperature dependent mobility. This model is the default model for impact ionization with energy balance transport and is activated with the TOYABE or SELB parameters on the IMPACT statement. The ionization rates now have the form:

$$\alpha_n = \mathrm{AN} exp\left(\frac{-\mathrm{BN}}{E_{eff,\,n}}\right) \quad \text{3-359}$$

$$\alpha_p = \mathrm{AP} exp\left(\frac{-\mathrm{BP}}{E_{eff,\,p}}\right) \quad \text{3-360}$$

where the model parameters: AN, AP, BN, and BP are user-definable in the IMPACT statement (see Table 3-58 for their default values).

The effective electric field is calculated according to:

$$E_{eff,n} = \frac{3}{2}\frac{kT_n}{q\,\mathrm{LREL.EL}} \quad 3\text{-}361$$

$$E_{eff,p} = \frac{3}{2}\frac{kT_p}{q\,\mathrm{LREL.HO}} \quad 3\text{-}362$$

where the energy relaxation lengths, LREL.EL and LREL.HO, can be explicitly defined in the IMPACT statement, or can be calculated according to:

```
LREL.EL = VSATN* TAUSN                    3-363
LREL.HO = VSATP* TAUSP                    3-364
```

VSATN and VSATP are the saturation velocities for electrons and holes, and TAUSN and TAUSP correspond to the electron energy relaxation times (TAUREL.EL and TAUREL.HO) in Equations 3-272 and 3-273. You must set the LENGTH.REL flag to use the values of LREL.EL and LREL.HO specified in the IMPACT statement.

Table 3-67. User-Specifiable Parameters for Equations 3-361-3-364

Statement	Parameter	Units
IMPACT	LREL.EL	µm
IMPACT	LREL.HO	µm
MATERIAL	VSAT	cm/s
MATERIAL	VSATN	cm/s
MATERIAL	VSATP	cm/s
IMPACT	TAUSN	s
IMPACT	TAUSP	s

Note: As an added level of flexibility, the relaxation times used for the energy balance equation and those used in the impact ionization model are separated into two user-definable parameters. In contrast to TAUREL.EL and TAUREL.HO, which are used in different formulae, the TAUSN and TAUSP parameters are only applicable in the impact ionization expression in Equations 3-360 and 3-361. By default, TAUREL.EL=TAUSN and TAUREL.HO=TAUSP.

It can also be argued that the AN, AP, BN, and BP parameters should also be a function of the carrier temperature. But, no clear theoretical basis for this has been proposed and accepted. Instead, the C-INTERPRETER within ATLAS has been extended to include two C-INTERPRETER functions. These functions are specified via the F.EDIIN and F.EDIIP parameters of the IMPACT statement. These parameters specify the filename of a text file containing a C-INTERPRETER function that describes the dependence of the model parameters AN, AP, BN and BP as a function of the carrier temperatures. These values will then be used within Toyabe's energy dependent impact ionization model.

Concannon's Impact Ionization Model

The previous non-local impact ionization model inherently assumes a Maxwellian shape to the distribution of hot carriers. Recent work by Fiegna et. al. [54] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the energy distribution function. To accurately model these effects, a non-Maxwellian based model from Concannon [40] has been implemented. Based upon this energy distribution the model calculates the probability of a carrier having sufficient energy to cause impact ionization. The model results show good agreement with measured results for a 0.9μm flash EPROM device [40].

To enable the Concannon substrate current for the electron and hole continuity equations, specify the N.CONCANNON and P.CONCANNON parameters in the IMPACT statement.

The generation rate is a function of the carrier temperature and concentration is given by:

$$G_n(x,y) = \texttt{CSUB.N} \times n \int_{\texttt{ETH.N}}^{\infty} F(\varepsilon, T_n(x,y)))d\varepsilon \quad \text{3-365}$$

$$G_p(x,y) = \texttt{CSUB.P} \times p \int_{\texttt{ETH.P}}^{\infty} F(\varepsilon, T_p(x,y)))d\varepsilon \quad \text{3-366}$$

where $n(x,y)$ and $p(x,y)$ are the electron and hole carrier concentrations within the semiconductor, ε is energy, $T_n(x,y)$ and $T_p(x,y)$ are the electron and hole carrier temperatures in the semiconductor, F is given in Equation 3-263, CSUB.N, CSUB.P, ETH.N, and ETH.P are user-specifiable parameters as given in Table 3-68.

Table 3-68. User-Specifiable Parameters for Equations 3-365 and 3-366

Statement	Parameter	Default	Units
IMPACT	CSUB.N	2.0×10^{14}	
IMPACT	CSUB.P	4.0×10^{14}	
IMPACT	ETH.N	1.8	eV
IMPACT	ETH.P	3.5	eV

The function, $F(\varepsilon,T)$, in Equations 3-365 and 3-366 is given by the product of the density of states function, g(ε), and the energy distribution function $f(\varepsilon)$ as:

$$F = \frac{g(\varepsilon)f(\varepsilon)}{\int_0^{\infty} g(\varepsilon)f(\varepsilon)} \quad \text{3-367}$$

The density of states function is given by:

$$g(\varepsilon) = \varepsilon^{1.25} \quad \text{3-368}$$

The energy distribution functions for electrons and holes are:

$$f_n(\varepsilon) = \left[exp\left(\frac{-\mathrm{CHIA}\ \varepsilon^3}{T_n^{1.5}} \right) + \mathrm{C0}\, exp\left(\frac{-\mathrm{CHIB}\ \varepsilon^3}{T_n^{1.5}} \right) \right] \quad \text{3-369}$$

$$f_p(\varepsilon) = exp\left(\frac{-\mathrm{CHI.HOLES}\ \varepsilon^3}{T_p^{1.5}} \right) \quad \text{3-370}$$

Here, ε is energy, $T_{n,p}$ are the carrier temperatures, CHIA, CHIB, and C0 are user-specifiable parameters (see Table 3-69).

Table 3-69. User-Definable Parameters for the Energy Distribution Functions

Statement	Parameter	Default	Units
IMPACT	CHIA	3.0×10^5	
IMPACT	CHIB	5.0×10^4	
IMPACT	C0	2.5×10^{-10}	
IMPACT	CHI.HOLES	4.6×10^4	

Two other parameters in the IMPACT statement are user-definable, which may affect the result of the numeric integration. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter sets the upper limit of the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

Note: To maintain self-consistent results, implement this model if the Concannon model is being used for the simulation of gate current.

Other Hydrodynamic Models

You can also use the Lackner, Okuto-Crowell, and van Overstraeten - de Man models with the hydrodynamic model. To use this model, set the command line flag -ISE instead of the ATLAS default model (Equations 3-361 and 3-362). The ATLAS default model assumes the effective field used to calculate impact ionization rate is directly proportional to carrier Temperature.

A more physical relationship between effective field and carrier temperature is given by

$$E_{eff,\,n} = \frac{1.5K_B(T_n - T_L)}{\mathrm{TAUREL.EL\ NISELAMBDA\ VSATN}} \quad \text{3-371}$$

$$+ \alpha_n(E_{eff,\,n})\mathrm{N.UPSILON}(EG(T_L) + \mathrm{NISEDELTA}K_B T_n)$$

$$E_{eff,\,p} = \frac{1.5K_B(T_p - T_L)}{\texttt{TAUREL.HO PISELAMBDA VSATP}} + \alpha_p(E_{eff,\,p})\texttt{P.UPSILON}(EG(T_L) + \texttt{PISEDELTA}K_B T_p) \quad \text{3-372}$$

The parameters N.UPSILON and P.UPSILON can be either 0(unset) or 1(set). The default for Silicon is 1. It is 0 for other materials. In the case where UPSILON is unity, the equation becomes a transcendental one because it contains α as a function of E_{eff}. In this case, the equation is solved iteratively in order to obtain α. If no solution is found, then an error message will output and you must set N.UPSILON=0 or P.UPSILON=0. The user adjustable parameters are given in Table 3-70.

Table 3-70. Hydrodynamic Model Parameters

Statement	Parameter	Default	Units
IMPACT	N.UPSILON	True (in silicon)	False (otherwise)
IMPACT	P.UPSILON	True (in silicon)	False (otherwise)
IMPACT	NISELAMBDA	1.0	
IMPACT	PISELAMBDA	1.0	
IMPACT	NISEDELTA	1.5	
IMPACT	PISEDELTA	1.5	

3.6.5: Band-to-Band Tunneling

If a sufficiently high electric field exists within a device local band bending may be sufficient to allow electrons to tunnel, by internal field emission, from the valence band into the conduction band. An additional electron is therefore generated in the conduction band and a hole in the valence band. This generation mechanism is implemented into the right-hand side of the continuity equations. The tunneling generation rate is [69,70,71,85] as:

$$G_{BBT} = \texttt{BB.A}\ E^{\texttt{BB.GAMMA}} exp\left(-\frac{\texttt{BB.B}}{E}\right) \quad \text{3-373}$$

where E is the magnitude of an electric field and BB.A, BB.B, and BB.GAMMA are user-definable parameters. In ATLAS there are three different sets of values that may be applied to the model parameters.

The model parameters can be set to the standard model [81] by specifying BBT.STD on the MODELS statement. The parameter defaults for the standard model are as follows:

BB.A = 9.6615e18 BB.B= 3.0e7 V/cm BB.GAMMA= 2.0

The model parameters may also be set to the Klaassen Model [69,85,70] by specifying BBT.KL on the MODELS statement. The parameter defaults for the Klaassen model are as follows:

BB.A = 4.00e14 BB.B = 1.9e7 V/cm BB.GAMMA= 2.5

In application, use the standard model with direct transitions while using the Klaassen model with indirect transitions.

You can modify the basic band-to-band tunneling given in Equation 3-373 by the generation rate GBBT with the D factor as suggested in [69]. The D factor is given by:

$$D = (exp[\phi_p - q\psi / KT] + 1)^{-1} - (exp[\phi_n - q\psi / KT] + 1)^{-1} \quad 3\text{-}374$$

where ϕ_n and ϕ_p are the electron and hole quasi-Fermi levels as given in Equations 3-9 and 3-10 and ψ is the electrostatic potential. To enable this modification, specify BBT.HURKX in the MODELS statement.

Another modification allows these model parameters to be calculated from the first principles by specifying the AUTOBBT parameter in the MODELS statement. In this case, the parameters are calculated according to:

$$\mathtt{BB.A} = \frac{q^2\sqrt{(2 \times \mathtt{MASS.TUNNEL}\, m_0)}}{h^2\sqrt{\mathtt{EG300}}} \quad 3\text{-}375$$

$$\mathtt{BB.B} = \frac{\pi^2\, \mathtt{EG300}^{\frac{3}{2}}\sqrt{\frac{\mathtt{MASS.TUNNEL}\, m_0}{2}}}{qh} \quad 3\text{-}376$$

Here:

$$\mathtt{BB.GAMMA} = 2 \quad 3\text{-}377$$

where q is the electronic charge, $\hbar$ is Planck's constant, E_g is the energy bandgap, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. The parameter MASS.TUNNEL may be set on the MODELS statement and the bandgap at 300K, EG300, is defined on the MATERIAL statement.

Table 3-71. User-Definable Parameters in the Band-to-Band Tunneling Model

Statement	Parameter	Default	Units
MODEL	BB.A	4.0×10^{14}	
MODEL	BB.B	1.9×10^{7}	V/cm
MODEL	BB.GAMMA	2.5	

Schenk Band to Band Tunnelling Model

A comprehensive study of band-to-band tunnelling was carried out by Schenk [138]. A rigorous theory was developed and then an approximate result suitable for use in device simulations was derived. The result shows that phonon assisted band-to-band tunnelling rate is generally dominant compared to the band-to-band tunnelling that doesn't involve a phonon scattering event. The direct band-to-band tunnelling is thus neglected. The model also assumes that the electric field is constant over the tunnelling length. Therefore, it is a local model.

The recombination-generation rate is given by

$$G_{BBT}^{SCHENK} = A.BBT.SCHENK F^{7/2} S \left(\frac{(A^{\mp})^{-3/2} exp\left(\frac{A^{\mp}}{F}\right)}{exp\frac{(HW.BBT.SCHENK)}{KT} - 1} + \frac{(A^{\pm})^{-3/2} exp\left(\frac{A^{\pm}}{F}\right)}{1 - exp\frac{(-HW.BBT.SCHENK)}{KT}} \right) \quad 3\text{-}378$$

where $A^{\pm}$ =B.BBT.SCHENK($HW^{\pm}$ HW.BBT.SCHENK) and S is a statistical fact or dependent on carrier concentrations.

You can set parameters from Table 3-72 on the MODELS or MATERIAL statement. To enable the model, specify SCHENK.BBT on the MODELS statement.

Table 3-72. Schenk Band to Band Tunelling Model Parameters

Statement	Parameter	Type	Default	Units
MATERIAL/MODELS	A.BBT.SCHENK	Real	8.977E20	$cm^2s^{-1}V^{-2}$
MATERIAL/MODELS	B.BBT.SCHENK	Real	2.14667E7	$eV^{(-3/2)}Vcm^{-1}$
MATERIAL/MODELS	HW.BBT.SCHENK	Real	0.0186	eV

Non-local Band-to-Band Tunneling

The band-to-band tunneling model in the previous section uses the electric field value at each node to give a Generation rate at that point due to the tunneling. In reality, the tunneling process is non-local and is necessary to take into acount the spatial profile of the energy bands. It is also necessary to take into account the spatial separation of the electrons generated in the conduction band from the holes generated in the valence band.

A model for this process has been created for ATLAS. It assumes that the tunneling can be modelled as being one-dimensional in nature so that it can be calculated using a special rectangular mesh superimposed over and coupled to the ATLAS mesh. This mesh needs to include the junction region of interest and the direction of the Band-to-Band tunneling must be specified.

Adjust the extent and resolution of the mesh to obtain the best results. To position the special mesh, use the QTX.MESH and QTY.MESH statements. You must then set the required direction of the quantum tunneling using the QTUNN.DIR parameter on the MODELS statement.

For example

```
qtx.mesh loc=0.0 spac=0.25
qtx.mesh loc=1.0 spac=0.25

qty.mesh loc=1.5 spac=0.01
qty.mesh loc=3.5 spac=0.01
```

will set up a mesh in the rectangle bounded by `x=0.0`, `x=1.0`, `y=1.5` and `y=3.5`. This mesh has a much finer spacing in the y-direction and is therefore suitable for modelling quantum tunneling in the y-direction. The mesh spacing does not have to be uniform. We recommend you to have as fine a mesh as possible across the region where tunneling is happening.

In order to explain how the tunneling current is calculated, let us consider an Energy band profile along each slice in the tunneling direction when applying a reverse bias across the junction. Figure 3-5 shows a schematic of this. The allowed range of valence band electron energy for which tunneling is permitted is indicated.

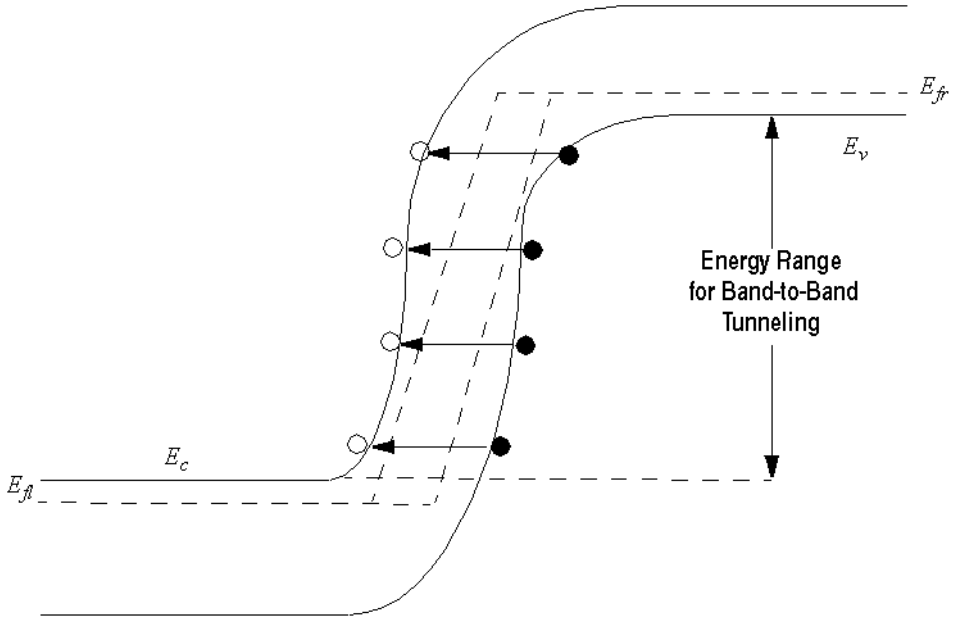


Figure 3-5: Schematic of non-local band to band tunneling in reverse bias

If the junction doping levels are sufficiently high, then this energy range may also exist in forward bias. If we consider only elastic scattering mechanisms, then electrons from anywhere in the permitted energy range can tunnel from the valence band to the conduction band.

ATLAS considers each energy in the allowed range and determines the spatial start and end positions for the tunneling at each energy, E, which we label x_{beg} and $x_{end.}$

The contribution to tunneling current for an electron in the energy range E - $\Delta E/2$ to E + $\Delta E/2$ (where ΔE is a small energy increment) is

$$J(E) = \frac{qKTm^*}{2\pi^2\hbar^3}T(E)ln\left\{\frac{1+exp[E_{Fl}-E]/kT}{1+exp[E_{Fr}-E]/kT}\right\}\Delta E \qquad 3\text{-}379$$

where *T(E)* is the tunneling probability and

$$m^* = m_0\sqrt{m_e(x_{end})m_h(x_{beg})} \quad 3\text{-}380$$

The sign of the current is *-ve* if $E_{fl} < E_{fr}$ and *+ve* if $E_{fl} > E_{fr}$. In equilibrium, $E_{fl} = E_{fr}$ and the current is zero as expected.

This contribution to the tunneling current is calculated and coupled into the ATLAS mesh at x_{beg} and x_{end}. This is repeated for the whole range of allowed tunneling energies. Therefore, it modifies the continuity equations with generation terms (reverse bias) or recombination terms (forward bias). In forward bias, it was found to be necessary to apply the recombination term at the ends of the QT mesh to avoid numerical instabilities.

The tunneling probability was calculated using a two band approximation for the evanescent wavevector, namely

$$k = \frac{k_e k_h}{\sqrt{k_e^2 + k_h^2}} \quad 3\text{-}381$$

where

$$k_e = \frac{1}{i\hbar}\sqrt{2m_0 m_e(E - E_c)} \quad 3\text{-}382$$

$$k_h = \frac{1}{i\hbar}\sqrt{2m_0 m_h(E_v - E)} \quad 3\text{-}383$$

which ensures that the tunneling is electron-like near the conduction band, hole-like near the valence band and is mixed at energies near the middle of the energy gap. ATLAS uses a transmission matrix method to calculate the tunneling probability for direct quantum tunneling simulations through an insulator. In the case of band-to-band tunneling, however, a carefully applied WKB method was found to give equivalent results and is computationally more efficent.

The effective masses used in the calculation are the density of states effective masses for the conduction band and valence band. They will also be the default values calculated by ATLAS. You can set them using the MC and MV parameters on the MATERIAL statement. Changing these parameters, however, may also change the behavior of other models. Therefore, we recommend that you modify the effective masses used in the band-to-band tunneling calculation using the ME.TUNNEL and MH.TUNNEL parameters of the MATERIAL statement.

To enable non-local Band to Band tunneling, use the BBT.NONLOCAL parameter on the MODELS statement. You must also define the QT mesh. A local band-to-band tunneling model can also be specified simultaneously. The local band-to-band tunneling model will be applied to nodes outside the QT mesh. The non-local band-to-band model will be applied to nodes inside the QT mesh. To do this, specify BBT.STD or BBT.KL and BBT.NONLOCAL on the MODELS statement .

The tunneling current depends on the values of the band edge at all points along its tunneling path. Ideally, the derivatives of the tunneling current with respect to these values should be included in the calculation. ATLAS includes these derivatives if you set the BBT.NLDERIVS parameter on the MODELS statement. This also slows down the calculation. Therefore, we only recommend this if the solution fails to converge. Convergence without this parameter enabled is good in general.

As mentioned above, instabilities can occur in forward bias due to the inclusion of tunneling recombination terms in the junction itself.

The parameter BBT.FORWARD moves the recombination to the ends of the QT mesh and improves convergence. We recommend you specify this parameter in order to model the forward tunneling current of a Zener diode.

Table 3-73. Non-Local Band-to-Band Tunneling Parameters for the MODELS Statement

Parameter	Type	Default
BBT.NONLOCAL	Logical	False
BBT.NLDERIVS	Logical	False
BBT.FORWARD	Logical	False

Table 3-74. Non-Local Band-to-Band Tunneling Parameters for the MATERIAL Statement

Parameter	Type	Units
ME	Real	None
MV	Real	None
ME.TUNNEL	Real	None
MH.TUNNEL	Real	None

Note: If you want to fine tune the Band to Band tunneling, we recommend you use ME.TUNNEL and MH.TUNNEL parameters because they will only modify the effective masses used in tunneling.

3.6.6: Gate Current Models

In devices that have a metal-insulator-semiconductor (MIS) formation, the conductance of the insulating film would ideally be considered as zero. But, for the sub 0.5um generation of MOS devices there is now considerable conductance being measured on the gate contacts. This gate current has resulted in two major consequences, one negative and one positive.

On the negative side, the gate current is responsible for the degradation in device operating characteristics with time. This reliability issue is important because the lifetime of electronic parts has to be guaranteed. You can simulate reliability within the Silvaco suite of tools for device level reliability which are described in a later chapter.

On the positive side, the existence of this gate current has caused the proliferation of the non-volatile memory market. These devices use the existence of gate current to program and erase the charge on a "floating" contact. This concept has resulted in a variety of different devices such as FLASH, FLOTOX, and EEPROM. All such devices rely on the physics of the gate current process for their existence.

There are a variety of different conduction mechanisms within an insulating layer [156], but in the case of nonvolatile memory, only two mechanism are relevant: Fowler-Nordheim tunneling and hot carrier injection. Models for these two injection processes are described in the following sections. In the case of hot electron injection, two models are available: the lucky electron model and the Concannon gate current model.

Fowler-Nordheim Tunneling

If the electric field across an insulator is sufficiently high, then it may cause tunneling of electrons from the semiconductor (or metal) Fermi level into the insulator conduction band. This process is strongly dependent on the applied electric field but is independent of the ambient temperature.

The Fowler-Nordheim Equation [101] expresses tunnel current density through the oxide as:

$$J_{FN} = \texttt{F.AE}\ E^2 exp\left(-\frac{\texttt{F.BE}}{E}\right) \quad \text{3-384}$$

$$J_{FP} = \texttt{F.AH}\ E^2 exp\left(-\frac{\texttt{F.BH}}{E}\right) \quad \text{3-385}$$

where E specifies the magnitude of the electric field in the oxide. The model parameters: F.AE, F.AH, F.BE, and F.BH can be defined on the MODELS statement. The default values for these parameters, obtained from Keeney, Piccini, and Morelli [81], are shown in Table 3-75.

Table 3-75. User-Specifiable Parameters for Equations 3-384 and 3-385

Symbol	Statement	Parameter	Default Values
A_{FN}	MODELS	F.AE	1.82×10^{-7}
B_{FN}	MODELS	F.BE	1.90×10^{8}
A_{FH}	MODELS	F.AH	1.82×10^{-7}
B_{FH}	MODELS	F.BH	1.90×10^{8}

The Fowler-Nordheim model in ATLAS has been implemented in two ways. In the post-processing implementation, the tunnelling current is calculated from the solution to the device equations at each bias step. In the self-consistent implementation, the tunnelling current is included directly in the current-continuity equations. Thus, it is part of the solution to the equations and gives accurate current continuity for the device.

To enable the post processing version, use the MODELS statement parameters FNPP for electron current and FNHPP for hole current. To enable the self-consistent solution, use the MODELS statement parameters FNORD for electrons and FNHOLES for holes. Setting the parameter FN.CUR is the same as specifying FNORD and FNHOLES.

For either model, the implementation scheme is the same. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment, the Fowler-Nordheim current is calculated as described above. This current is then added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the Fowler-Nordheim current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, receives the Fowler-Nordheim current.

A second model may be chosen using the NEARFLG parameter of the MODEL statement. In this case, the electrode-insulator segment found closest to the semiconductor-insulator segment receives the Fowler-Nordheim current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary.

Note: Since Fowler-Nordheim tunneling current is responsible for EPROM and EEPROM cell erasure, this model should always be specified when performing erasure simulation. We also recommend that you model the band-to-band tunneling model if you model Fowler-Norheim tunneling.

Note: When simulating EPROM erasure in a transient analysis with this model, the floating contact charge becomes a function of the gate current. In this case, the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of the charge is then used as the boundary condition for the next time step.

Lucky Electron Hot Carrier Injection Model

In the Lucky-Electron Hot Carrier Injection Model it is proposed that an electron is emitted into the oxide by first gaining enough energy from the electric field in the channel to surmount the insulator/semiconductor barrier. Once the required energy to surmount the barrier has been obtained the electrons are redirected towards the insulator/semiconductor interface by some form of phonon scattering. When these conditions are met, the carrier travelling towards the interface will then have an additional probability that it will not suffer any additional collision through which energy could be lost.

The model implemented into ATLAS is a modified version of the model proposed by Tam [159] and is activated by the parameters HEI and HHI, for electron and hole injection respectively, on the MODELS statement. The gate electrode-insulator interface is subdivided into a number of discrete segments which are defined by the mesh. For each segment the lucky electron model is used to calculate the injected current into that segment. The total gate current is then the sum of all of the discrete values.

If we consider a discrete point on the gate electrode-insulator boundary we can write a mathematical formula for the current injected from the semiconductor. The formula calculates the injected gate current contribution from every node point within the semiconductor according to:

$$Iinj = \int\int P_n(x,y)\left|\vec{J}n(x,y)\right|dx\ dy + \int\int P_p(x,y)\left|\vec{J}p(x,y)\right|dx\ dy \qquad 3\text{-}386$$

where $J_{n,p}$ (x,y) are the electron and hole current densities at a point (x,y) within the semiconductor, and $P_{n,p}$ (x,y) are the probabilities that a fraction of this current reaches the gate oxide and is injected across into the gate electrode. The total probability $P_{n,p}$ (x,y) is defined by:

$$P_n(x,y) = P_{\phi B,n} P_{1,n} P_{2,n}^{/\mathtt{IG.ELINR}} \qquad 3\text{-}387$$

$$P_p(x,y) = P_{\phi B,p} P_{1,p} P_{2,p}^{/\mathtt{IG.HLINR}} \qquad 3\text{-}388$$

where E is the electric field parallel to the current flow, IG.ELINR and IG.HLINR are the electron and hole mean free path lengths between redirecting collisions. The three probability factors will now be described.

The probability $P\phi_B$ is the probability of a carrier gaining the energy ϕ_B by moving in, and parallel to, an electric field E, without suffering energy loss by optical phonon scattering and is given by:

$$P_{\phi_{B,n}} = 0.25\left(\frac{E \text{ IG.ELINF}}{\phi_{B,n}}\right) exp\left(-\frac{\phi_{B,n}}{E \text{ IG.ELINF}}\right) \quad 3\text{-}389$$

$$P_{\phi_{B,p}} = 0.25\left(\frac{E \text{ IG.HLINF}}{\phi_{B,p}}\right) exp\left(-\frac{\phi_{B,p}}{E \text{ IG.HLINF}}\right) \quad 3\text{-}390$$

where IG.ELINF and IG.HLINF are the mean free path lengths of electrons and holes for scattering by optical phonons. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B,n} = \text{IG.EB0} - \text{IG.EBETA}\sqrt{E_\perp} - \text{IG.EETAE}\,E_\perp^{2/3} - \Delta\psi(x,y) \quad 3\text{-}391$$

$$\phi_{B,p} = \text{IG.HB0} - \text{IG.HBETA}\sqrt{E_\perp} - \text{IG.HETAE}\,E_\perp^{2/3} - \Delta\psi(x,y) \quad 3\text{-}392$$

where $E_\perp$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0, are reduced to take account of three effects. The first effect is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second effect takes account of tunneling through the gate oxide by reducing the barrier height. The third effect takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default, this last effect is disabled. But you can enable it by specifying the E.BENDING and H.BENDING parameters for electrons and holes respectively.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected, and is given by:

$$P_{1,n} \sim exp\left(-\frac{r}{\text{IG.ELINF}}\right) \quad 3\text{-}393$$

$$P_{1,p} \sim exp\left(-\frac{r}{\text{IG.HLINF}}\right) \quad 3\text{-}394$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by:

$$P_{2,n} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\text{PATH.N}}\right) \quad \text{for } \theta > \text{THETA.N} \quad 3\text{-}395$$

$$P_{2,n} = 0 \quad \text{for } \theta < \text{THETA.N} \quad 3\text{-}396$$

$$P_{2,p} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\text{PATH.P}}\right) \quad \text{for } \theta > \text{THETA.P} \qquad 3\text{-}397$$

$$P_{2,p} = 0 \quad \text{for } \theta < \text{THETA.P} \qquad 3\text{-}398$$

Here, PATH.N and PATH.P are the electron and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [171]. His experiments indicate a critical rejection angle, THETA.N and THETA.P, between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle then the electrons are repelled back to the substrate.

Table 3-76 lists the user-definable model parameters that can be set in the MODELS statement, their default values, and their units.

Table 3-76. User-Definable Parameters in Concannon's Gate Current Model

Statement	Parameter	Default	Units
MODELS	IG.EB0	3.2	eV
MODELS	IG.ELINR	6.16×10^{-6}	cm
MODELS	IG.HLINR	6.16×10^{-6}	cm
MODELS	IG.ELINF	9.2×10^{-7}	cm
MODELS	IG.HB0	4.0	eV
MODELS	IG.HLINF	9.2×10^{-7}	cm
MODELS	IG.EBETA	2.59×10^{-4}	$(Vcm)^{1/2}$
MODELS	IG.HBETA	2.59×10^{-4}	$(Vcm)^{1/2}$
MODELS	IG.EETA	2.0×10^{-5}	$V^{1/3}cm^{2/3}$
MODELS	IG.HETA	2.0×10^{-5}	$V^{1/3}cm^{2/3}$
MODELS	IG.LRELE	3.0×10^{-6}	[Q=1] cm
MODELS	IG.LRELH	2.0×10^{-6}	[Q=1] cm
MODELS	PATH.N	3.4×10^{-7}	cm
MODELS	PATH.P	2.38×10^{-7}	cm
MODELS	THETA.N	60	degrees
MODELS	THETA.P	60	degrees

The implementation of this model is similar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments, which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the MODELS statement. In this case, the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary.

The lucky electron hot carrier injection model can be used to include the electron or hole carrier temperature in the solution because the carrier temperature does not directly enter the equations.

The one exception is in ATLAS2D where the electric field parallel to the current flow is calculated as

$$E = 1.5\ K_B\ T_n\ /\ IG.LRELE$$

for electrons if HCTE.EL is specified and

$$E = 1.5\ K_B\ T_p\ /\ IG.LRELH$$

for holes if HCTE.HO is specified. K_B is Boltzmanns constant in units of eV/Kelvin. If IG.LRELE is set to zero then E will be calculated the same way as if HCTE.EL is unspecified. The same is true for IG.LRELH and HCTE.HO. In ATLAS3D, this model is unavailable and E is calculated the same way regardless if HCTE is specified.

Note: When simulating EPROM programming with this model, the floating contact charging is simulated in the transient mode. In this case, the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of charge is then used as the boundary condition for the next time step.

Concannon's Injection Model

The implicit assumption in the lucky electron approach is a Maxwellian shape for the energy distribution of the hot carriers. Recent work by Fiegna [54] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the distribution function. To accurately model these effects, a non-Maxwellian based model from Concannon [40] has been implemented. This model requires the solution to the energy balance equation for the carrier temperatures but has been implemented in a similar manner to the lucky electron model. The Concannon gate injection model may be specified with the parameters N.CONCANNON and P.CONCANNON on the MODELS statement. This choice of parameters automatically activates the Energy Balance Transport Model.

The Concannon injection model has a similar form to the lucky electron model. The injected current is calculated according to:

$$Iinj = \int\int P_n(x, y)\ n(x, y) dx\ dy + \int\int P_p(x, y)\ p(x, y) dx\ dy \qquad 3\text{-}399$$

where $n(x,y)$ and $p(x,y)$ are the carrier concentrations within the semiconductor. The probability functions $P_n(x,y)$ and $P_p(x,y)$ are now defined by:

$$P_n(x,y) = -q \text{ CGATE.N } P_{\phi_{B,n}} P_{1,n} P_{2,n} \quad 3\text{-}400$$

$$P_p(x,y) = q \text{ CGATE.P } P_{\phi_{B,p}} P_{1,p} P_{2,p} \quad 3\text{-}401$$

where q is the electronic charge and the parameters CGATE.N and CGATE.P are user-definable on the MODEL statement. The three probability functions in Equations 3-399 and 3-400 shall now be described.

The probability that a carrier has sufficient energy to surmount the insulator-semiconductor barrier of height ϕ_B is now defined as a function of energy. The probability now has the form:

$$P_{\phi_{B,n}} = \int_{\phi_{B,n}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_n(x,y)) d\varepsilon \quad 3\text{-}402$$

$$P_{\phi_{B,p}} = \int_{\phi_{B,p}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_p(x,y)) d\varepsilon \quad 3\text{-}403$$

Here, $v_{\perp}(\varepsilon)$ is the perpendicular velocity of a hot carrier and defines the probability of a hot carrier with an energy ε travelling in the direction of the insulator-semiconductor. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B,n} = \text{IG.EB0} - \text{IG.EBETA}\sqrt{E_{\perp}} - \text{IG.EETA } E_{\perp}^{2/3} - \Delta\psi(x,y) \quad 3\text{-}404$$

$$\phi_{B,p} = \text{IG.HB0} - \text{IG.HBETA}\sqrt{E_{\perp}} - \text{IG.HETA } E_{\perp}^{2/3} - \Delta\psi(x,y) \quad 3\text{-}405$$

where $E_{\perp}$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0, are reduced to take account of three effects. The first effect is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second effect takes account of tunneling through the gate oxide by reducing the barrier height. The third effect takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default, this last effect is disabled. But you can enable it by specifying the E.BENDING and H.BENDING parameters for electrons and holes respectively.

The carrier velocity model follows the approach of Fiegna et. al. [54] where velocity is proportional to energy according to:

$$v_{\perp} \sim \varepsilon^{0.25} \quad 3\text{-}406$$

The function, $F(\varepsilon, T_{n,p}(x,y))$, is determined by the density of states and the energy distribution function according to:

$$F(\varepsilon, T_n(x,y)) \sim \frac{g(\varepsilon) f(\varepsilon)}{\int_{\circ}^{\infty} g(\varepsilon) f(\varepsilon) d\varepsilon} \quad 3\text{-}407$$

The density of states $g(\varepsilon)$ follows the analysis of Cassi [33] where:

$$g(\varepsilon) \sim \varepsilon^{1.25} \quad 3\text{-}408$$

Finally, the energy distribution functions for electrons and holes are defined by:

$$f_n(\varepsilon) \sim \left[exp\left(\frac{-\mathtt{CHIA}\,\varepsilon^3}{T_n^{1.5}} \right) + \mathtt{C0}\, exp\left(\frac{-\mathtt{CHIB}\,\varepsilon^3}{T_n^{1.5}} \right) \right] \quad (3\text{-}409)$$

$$f_p(\varepsilon) \sim exp\left(\frac{-\mathtt{CHI.HOLES}\,\varepsilon^3}{T_p^{1.5}} \right) \quad (3\text{-}410)$$

where CHIA, CHIB, CHI.HOLES, and C0 are user-definable constants found from fitting to measured data. The terms: T_n and T_p are the mean carrier temperatures for electrons and holes, which are calculated from the Energy Balance Transport Model.

Normalization in all of the above equations is accounted for in the combined constants of proportionality, CGATE.N and CGATE.P.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected and is given by:

$$P_{1,n} \sim exp\left(-\frac{r}{\mathtt{IG.ELINF}} \right) \quad (3\text{-}411)$$

$$P_{1,p} \sim exp\left(-\frac{r}{\mathtt{IG.HLINF}} \right) \quad (3\text{-}412)$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by:

$$P_{2,n} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathtt{PATH.N}} \right) \quad \text{for } \theta > \mathtt{THETA.N} \quad (3\text{-}413)$$

$$P_{2,n} = 0 \quad \text{for } \theta > \mathtt{THETA.N} \quad (3\text{-}414)$$

$$P_{2,p} = exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathtt{PATH.P}} \right) \quad \text{for } \theta > \mathtt{THETA.P} \quad (3\text{-}415)$$

$$P_{2,p} = 0 \quad \text{for } \theta < \mathtt{THETA.P} \quad (3\text{-}416)$$

Here, PATH.N and PATH.P are the electron and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [171]. His experiments indicate a critical rejection angle, THETA.N and THETA.P between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle, then the electrons are repelled back to the substrate.

Note: The current implementation of the Concannon model for hot carrier injection is that only carriers along the semiconductor-insulator interface are significant and as a result the probability P_1 is assumed unity. This also means that the integration is only applied to those node points along the semiconductor-insulator interface.

Two other parameters of the MODELS statement that may affect the result of the numeric integration are user-definable. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter sets the upper limit of the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

The implementation of this model is similar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the MODELS statement. In this case the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segments around the electrode boundary.

Note: To maintain self-consistent results, it's important that this model is implemented if the Concannon model is being used for the simulation of substrate current.

Direct Quantum Tunneling Model

For deep submicron devices, the thickness of the insulating layers can be very small. For example, gate oxide thicknesses in MOS devices can be as low as several nanometers. In this case, the main assumptions of the Fowler-Nordheim approximation are generally invalid and you need a more accurate expression for tunneling current. The one ATLAS uses is based on a formula, which was introduced by Price and Radcliffe [126] and developed by later authors. It formulates the Schrodinger equation in the effective mass approximation and solves it to calculate the transmission probability, *T(E)*, of an electron or hole through the potential barrier formed by the oxide layer. The incident (perpendicular) energy of the charge carrier, *E*, is a parameter. It is assumed that the tunneling process is elastic. After taking into account carrier statistics and integrating over lateral energy, the formula

$$J = \frac{qkT}{2\pi^2\hbar^3}\sqrt{m_y m_z}\int T(E) ln\left\{\frac{1+exp[(E_{Fl}-E)/kT]}{1+exp[(E_{Fr}-E)/kT]}\right\}dE \quad 3\text{-}417$$

is obtained, which gives the current density J (A/m^2) though the barrier. The effective masses m_y and m_z are the effective masses in the lateral direction in the semiconductor. For example, for a direct bandgap material, where the Γ valley is isotropic, both m_y and m_z are the same as the density of states effective mass. The logarithmic term includes the carrier statistics and E_{fl} and E_{fr} are the quasi-Fermi levels on either side of the barrier (see Figure 3-6). The range of integration is determined according to the band edge shape at any given contact bias.

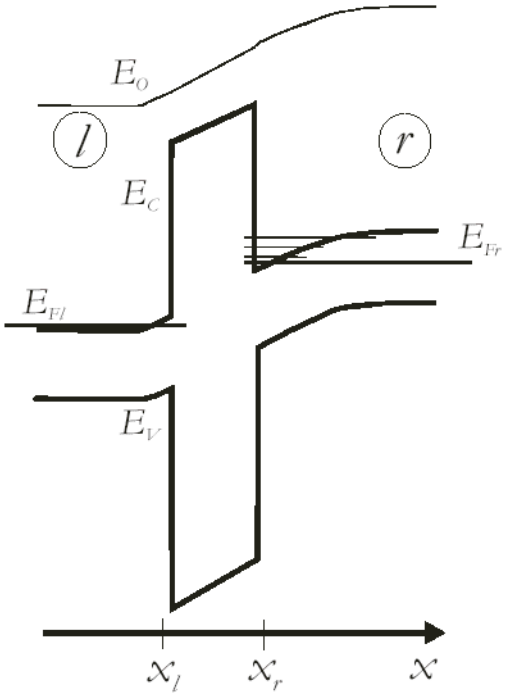


Figure 3-6: Typical conduction and valence band profiles of a MOS capacitor. The right region represents the MOS bulk, the left region represents the gate.

For indirect bandgap materials, Equation 3-418 is applied to each valley and the resulting sum gives the tunneling current. For the conduction band of silicon, for example, summing over the 6 valleys gives

$$J = \frac{qkT}{2\pi^2\hbar^3}(2m_t + 4\sqrt{m_l m_t})\int T(E) ln\left\{\frac{1+exp[(E_{Fl}-E)/kT]}{1+exp[(E_{Fr}-E)/kT]}\right\}dE \quad 3\text{-}418$$

where m_t is the transverse effective mass and m_l the longitudinal mass. For a valence band, you need to sum the light hole and heavy hole contributions separately. For the valence band, the light hole and heavy contributions are summed to give

$$J = \frac{qkT}{2\pi^2\hbar^3}(m_{lh} + m_{hh})\int T(E) ln\left\{\frac{1+exp[(E_{Fl}-E)/kT]}{1+exp[(E_{Fr}-E)/kT]}\right\}dE \quad 3\text{-}419$$

In equilibrium, $E_{fl} = E_{fr}$ and the logarithmic term and consequently J is identically zero.

Tunneling Types

ATLAS can account for several types of tunneling. For clarity, we assume the material on the left hand side of Figure 3-6 is polysilicon. Electron tunneling occurs when electrons tunnel through the insulator energy barrier from one conduction band to the other. Hole tunneling occurs when holes tunnel from one valence band to the other. If the bias applied to the contact is sufficiently large, then the situation illustrated in Figure 3-7 can occur. In this case, an electron in the Silicon valence band tunnels through the insulator to the polysilicon conduction band.

If the bias is reversed, then the opposite occurs in which an electron in the polysilicon valence band tunnels to the Silicon conduction band. Both of these cases are referred to as band-to-band Tunnelling. These should not be confused, however, with the other band-to-band tunnelling models implemented in ATLAS, which apply to regions of high electric field within a semiconductor. All these types of tunneling can be used as post-processing calculations or self-consistent calculations. In the former case, the tunneling currents are calculated after ATLAS has found a converged solution at a given bias. ATLAS then outputs the calculated tunnelling currents to the logfile. In the latter case, the tunneling currents are coupled with the solution of the current continuity equations in the semiconductor. Therefore, the terminal currents are self-consistent.

In situations where there is a strong quantum confinement in the semiconductor, ATLAS can include charge quantization effects. To do this, use the Schrodinger-Poisson solver in the Silicon, which causes the tunneling current to be calculated as a sum over eigenstates. The relevant formulae are

$$J = \frac{qkT}{\pi\hbar^2}\sqrt{m_y m_z}\sum_i \upsilon_r(E_i)T(E_i)ln\left\{\frac{1+exp[(E_{Fl}-E_i)/kT]}{1+exp[(E_{Fr}-E_i)/kT]}\right\}dE \quad 3\text{-}420$$

where the sum is over all bound state energies, E_i, for a particular band minimum.

For electrons in silicon, with 6 conduction band minima, we obtain

$$J = \frac{2qkT}{\pi\hbar^2}m_t\sum_i \upsilon_{ril}T(E_{il})ln\left\{\frac{1+exp[(E_{Fl}-E_{il})/kT]}{1+exp[(E_{Fr}-E_{il})/kT]}\right\} \quad 3\text{-}421$$

$$+\frac{4qkT}{\pi\hbar^2}\sqrt{m_t m_l}\sum_i \upsilon_{rit}T(E_{it})ln\left\{\frac{1+exp[(E_{Fl}-E_{it})/kT]}{1+exp[(E_{Fr}-E_{it})/kT]}\right\}$$

where the sums over longitudinal bound eigenstates and transverse bound eigenstates are done separately. For holes, the current is calculated using the equation

$$J = \frac{qkT}{\pi\hbar^2}m_{lh}\sum_i \upsilon_{ri\_lh}T(E_{i\_lh})ln\left\{\frac{1+exp[(E_{i\_lh}-E_{Fl})/kT]}{1+exp[(E_{i\_lh}-E_{Fr})/kT]}\right\} \quad 3\text{-}422$$

$$+\frac{qkT}{\pi\hbar^2}m_{hh}\sum_i \upsilon_{ri\_hh}T(E_{i\_hh})ln\left\{\frac{1+exp[(E_{i\_hh}-E_{Fl})/kT]}{1+exp[(E_{i\_hh}-E_{Fr})/kT]}\right\}$$

where the sums over light hole and heavy hole bound eigenstates are done separately.

The expression υ is called the attempt frequency and is given by

$$\upsilon_r = \frac{\hbar k}{4m}\left[\left|\Psi(x_r)\right|^2 + \left|\frac{d\Psi}{dx}(x_r)\right|^2\frac{1}{k^2}\right] \quad 3\text{-}423$$

It is evaluated at the semiconductor-oxide interface if quantum confinement occurs there. This quantum confined modification of the tunneling will be applied when you set the SCHRODINGER parameter and set quantum tunneling. This requires you to set CARRIERS=0 on the METHOD statement. Therefore, this model cannot be used if it is required to solve the current continuity equations.

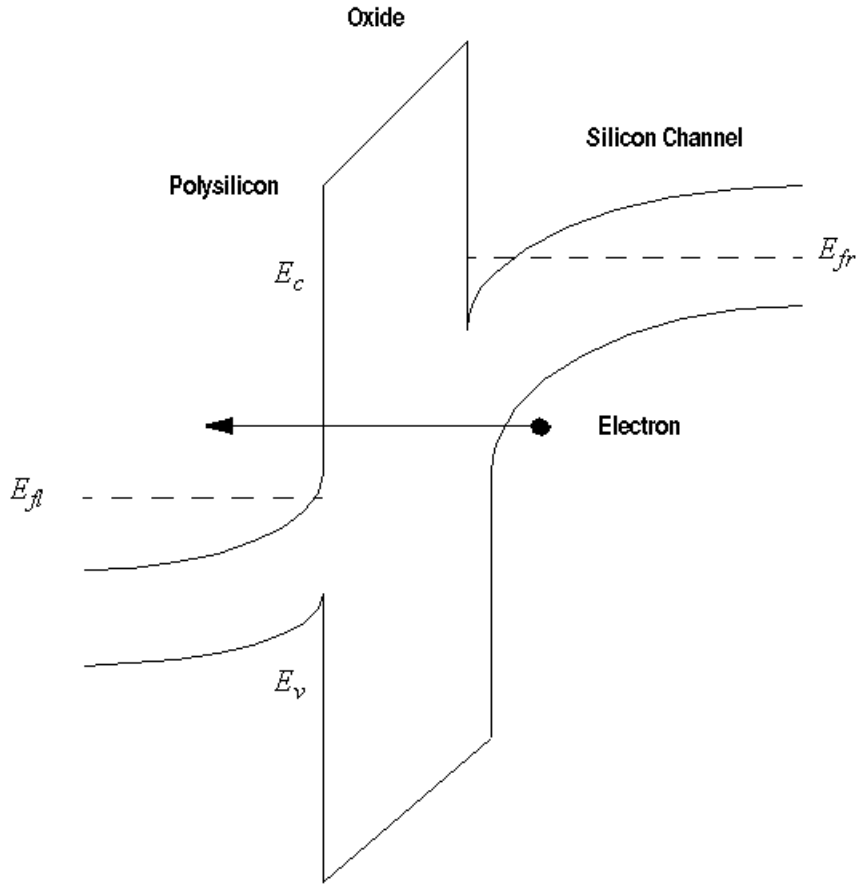


Figure 3-7: Schematic of band-to-band tunneling across a gate insulator with a polysilicon gate.

Enabling the Models

To enable the quantum tunneling model for electrons, specify QTUNN.EL in the MODELS statement. To enable the quantum tunneling model for holes, specify QTUNN.HO in the MODELS statement. To enable Band-to-Band tunneling, specify QTUNN.BBT in the MODELS statement. In order to enable all three options, specify QTUNN in the MODELS statement.

Table 3-77. MODELS Statement: Quantum Tunneling Parameters

Parameter	Type	Default
QTUNN	Logical	False
QTUNN.BBT	Logical	False
QTUNN.EL	Logical	False
QTUNN.HO	Logical	False

QTUNN.EL and QTUNN.HO are available with the CARRIERS=0, CARRIERS=1, and CARRIERS=2 options on the METHOD statement. With CARRIERS=0, the semi-classical and quantum-confined variants are available. QTUNN.BBT is only available as part of a semi-classical calculation with either CARRIERS=0, CARRIERS=1 or CARRIERS=2.

To enable the self-consistent versions of the quantum tunneling models, specify the QTNLSC.EL and QTNLSC.HO parameters on the MODELS statement for electron and hole tunneling respectively. The Band-to-Band tunneling option is enabled using QTNLSC.BBT. You can enable all three models using the QTUNNSC parameter on the MODELS statement.

Table 3-78. MODELS Statement: Self-consistent quantum tunneling parameters.

Parameter	Type	Default
QTNL.DERIVS	Logical	false
QTNLSC.EL	Logical	false
QTNLSC.HO	Logical	false
QTNLSC.BBT	Logical	false
QTUNNSC	Logical	false

For the self-consistent implementations, convergence can be somewhat poor in some circumstances, particularly for high values of tunneling current. In this case, you can set the QTNL.DERIVS parameter on the MODELS statement. This will include extra terms in the Jacobian matrix, which should improve convergence. It will also increase the time needed to solve each iteration.

Mesh Considerations

The tunneling current model is quasi one-dimensional and the tunneling current is evaluated on a series of parallel or nearly parallel slices through the insulator. Implementation details therefore depend on the type of mesh. If the mesh was generated by ATLAS meshing commands, then everything including the direction of tunneling is determined automatically. If the mesh was created by another Silvaco product, for example ATHENA or DEVEDIT, then there are two options.

The first option is to use the QTX.MESH and QTY.MESH commands to create a rectangular mesh on which all quantum tunneling is calculated. Interpolation is used to couple between this supplementary mesh and the device mesh. By default, the tunneling will be assumed to be in the y-direction. You can change this by setting the QTUNN.DIR parameter on the MODELS statement to 1 instead of its default value of 0. Place the QT mesh to enclose the insulator and to not overextend into the conductor or semiconductor. The second option is to allow ATLAS to automatically generate slices through the oxide layer. This is to be preferred if the oxide geometry is non-planar.

There are a choice of two algorithms for determining the slices. The default is to calculate the currents on slices constructed as being locally perpendicular to each semiconductor-oxide segment on the interface. To enable the alternative algorithm, use the SHAPEOX parameter on the MODELS statement. In this case, slices giving the shortest distance to the contact are constructed from each node on the semiconductor-insulator interface.

If you specified the Schrodinger equation using the NEW.SCHRODINGER parameter and solved on a rectangular mesh specified by the SPX.MESH and SPY.MESH commands, then the Quantum tunneling current must also be calculated on the rectangular mesh defined by the QTX.MESH and QTY.MESH commands. Best results will be obtained if the mesh lines in the direction of tunneling are roughly coincident and if the SP mesh ends at the semiconductor-insulator interface.

Table 3-79. QTX.MESH and QTY.MESH Statements

Parameter	Type	Units	Default
NODE	Int		-999
LOCATION	Real	microns	-999
X	Real	microns	-999
RATIO	Real	microns	1.0
SPACING	Real	microns	-999

Table 3-80. MODELS Statement

Parameter	Type	Default
QTUNN.DIR	Real	0
SHAPEOX	Logical	False

Contact Specifications

Carriers that tunnel through an oxide are added to the current of the electrode into or from which they flow. If they tunnel into a polysilicon region with several contacts attached, then the tunnel current is added to the electrode that is nearest to the segment of the oxide/polysilicon interface across which the current is being calculated. The NEARFLG parameter in the MODELS statement is automatically set for quantum tunneling. Therefore, the algorithm used to obtain the nearest electrode is the same as when you set NEARFLG. To exclude any electrode from this algorithm, set the EXCLUDE_NEAR flag in the CONTACTS statement.

You can also set the effective mass to use inside each contact for the tunneling current calculation using the QTUNN.CMASS (electrons) and QTUNN.VMASS (holes) parameters in the CONTACTS statement. If the contact is polysilicon, then the default effective mass is either the conduction band or valence band density of states effective mass depending on its dopant specification.

Table 3-81. CONTACTS Statement

Parameter	Type	Units	Default
EXCLUDE_NEAR	Logical		False
QTUNN.CMASS	Real		1.0
QTUNN.VMASS	Real		1.0

To calibrate the tunneling current, use the effective mass in the oxide region. You can set this by using either the MC or ME.TUNNEL parameters on the MATERIAL statement for electrons and the MV or MH.TUNNEL parameters on the MATERIAL statement for holes.

For example

```
MATERIAL MATERIAL=OXIDE MC=0.6 MV=0.2
```

In addition to this direct tunneling model and Fowler Nordheim model, there are several quantum tunnelling models included in ATLAS for compatability with other products. These are mutually exclusive and the tunneling current outputs to a variable I_{tnl} (A) or J_{tnl} (A/um) regardless of model chosen.

Schenk Oxide Tunnelling model

Another approximate tunnelling model is based on the Gundlach model [62] and includes the effects of barrier lowering due to the image force potentials [137]. As such, it is especially suited for tunnelling through ultra-thin gate oxides. It involves mapping from the energy barrier arising from the actual barrier profile plus the correction due to the image force and to an effective trapezoidal barrier. The exact Transmission Coefficient for a trapezoidal barrier can then be obtained from

$$T(E) = \frac{2}{1+g(E)} \qquad 3\text{-}424$$

where

$$g(E) = \frac{\pi^2}{2}\left[\frac{m_s k_c}{m_c k_s}(Bi'_d Ai_o - Ai'_d Bi_o)^2 + \frac{m_c k_s}{m_s k_c}(Bi_d Ai'_o - Ai_d Bi'_o)^2 \right. \qquad 3\text{-}425$$

$$\left. + \frac{m_c m_s}{\lambda_o^2 m_{cx}^2 k_c k_s}(Bi'_d Ai'_o - Ai'_d Bi'_o)^2 + \frac{\lambda_o^2 m_{ox}^2 k_c k_s}{m_c m_s}(Bi_d Ai_o - Ai_d Bi_o)^2\right]$$

where k_c is the wavevector in the contact, k_s is the wavevector in the semiconductor, m_c is the effective mass in the contact, m_{ox} is the effective mass in the oxide and m_s is the effective mass in the semiconductor.

Complication arises because the image potential is not of a simple functional form, it is given by

$$E_{im}(x) = \frac{q^2}{16\pi\varepsilon_{ox}} \sum_{n=0}^{\infty} (k_1 k_2)^n \times \left[\frac{k_1}{nd + x} + \frac{k_2}{d(n+1) - x} + \frac{2k_1 k_2}{d(n+1)} \right] \quad \text{3-426}$$

with

$$k_1 = \frac{\varepsilon_{ox} - \varepsilon_M}{\varepsilon_{ox} - \varepsilon_M} = -1, \qquad k_2 = \frac{\varepsilon_{ox} - \varepsilon_s}{\varepsilon_{ox} - \varepsilon_s} \quad \text{3-427}$$

where ε is the relative dielectric permittivity (of the material indicated by the subscript), d is the oxide thickness and x is the position in the barrier. The sum can be evaluated numerically and the potential added to the barrier potential. This allows one to evaluate the action of an electron of incident energy E when moving through this barrier by numerically integrating the barrier energy minus the electron energy as a function of distance between the classical turning points of the motion.

This can be equated with the action of the carrier moving through a trapezoidal barrier with the barrier height as a fitting parameter.

$$S_{eff}(E) = S_{im} + actual(E) \quad \text{3-428}$$

This results in an effective trapezoidal barrier height as a function of electron incident energy E. This is evaluated at three different energies and these values are used to calculate the effective barrier height as a function of electron energy. The interpolation formula used is

$$\Phi_B(E) = \Phi_B(E_0) + \frac{\Phi_B(E_2) - \Phi_B(E_0)}{(E_2 - E_0)(E_1 - E_2)}(E - E_0)(E_1 - E) \quad \text{3-429}$$
$$- \frac{\Phi_B(E_2) - \Phi_B(E_0)}{(E_2 - E_0)(E_1 - E_2)}(E - E_0)(E_2 - E)$$

For a given electron incident energy, you use Equation 3-429 to calculate the effective barrier height and then use Equation 3-425 to calculate the Transmission Probability, *T(E)*. This is placed in an equation like Equation 3-417 and the integration over the range of tunnelling energies is carried out to give the tunnelling current.

The model has been implemented as both a post-processing step and alternatively, as being solved self-consistently with the current continuity equations. To enable the post-processing option, specify SCHENK on the MODELS statement. To enable the self-consistent version, use SCHKSC with the MODELS statement. You can enable them separately for electrons and holes if required using the parameters SCHENK.EL, SHENCK.HO (post-processing) and SCHKSC.EL, SCHKSC.HO (self-consistent) on the MODELS statement. The values of effective mass and permittivity will affect the quantity of tunnelling current as will the electron affinities and work functions of the materials involved.

Table 3-82. Schenk Oxide Tunnelling Flags			
Parameter	**Type**	**Default**	**Units**
SCHENK.EL	Logical	False	
SCHENK.HO	Logical	False	
SCHENK	Logical	False	
SCHKSC.EL	Logical	False	
SCHKSC.HO	Logical	False	
SCHKSC	Logical	False	
SCHENK.BBT	Logical	False	

Gate Tunnelling Model for SONOS Type Structures

One approach to obtaining Non-Volatile Memories with low operating Voltages is to use a SONOS structure. The (S)emiconducting channel has a thin layer of tunnel (O)xide grown on it, followed by a thin layer of silicon (N)itride, and then followed by a thicker blocking or capping layer of (O)xide, and finally a (S)emiconducting polysilicon gate. The nitride has trapping levels located within it and the nitride-oxide band offset allows charge to be accumulated in the nitride layer.

ATLAS presently has a model to allow the charge-erase behavior of this class of structure to be simulated. To enable it, use the FNONOS parameter on the MODELS statment. Currently, it requires you to either set the Nitride layer as a floating contact or embed floating contacts in the nitride layer. For each point in the Channel-Oxide interface, the nearest distance to the nitride layer is calculated. The tunnelling current for this point is then calculated as

$$J_n = \mathrm{F.AE}\ \mathrm{E}^2 / factor1 \exp(-\mathrm{F.BE}\ factor2/\mathrm{E}) \quad \text{3-430}$$

where

$$factor1 = \left[1-(1-\mathrm{DV}/\mathrm{BH.FNONOS})^{\frac{1}{2}}\right]^2 \quad \text{3-431}$$

and

$$factor2 = \left[1-(1-\mathrm{DV}/\mathrm{BH.FNONOS})^{\frac{3}{2}}\right] \quad \text{3-432}$$

DV is the potential drop across the tunnel oxide layer and is calculated automatically by ATLAS. The value BH.FNONOS is the Barrier height and if not specified directly ATLAS will calculate it. This formula is calculated using WKB theory for the tunnelling co-efficient through a trapezoidal barrier.

If DV > BH.FNONOS, then *factor1* and *factor2* are both set to be unity. In this case, Equation 3-430 is the same as the Fowler-Nordheim Expression (Equation 3-384).

ETA.FNONOS times the tunnelling current is added to the nearest floating electrode, where ETA.FNONOS is the capture efficiency. A factor (1-ETA.FNONOS) times the tunnelling current is added to the next nearest (gate) electrode. The efficiency can depend on the Charge state of the floating electrode itself. To enable this, set the NT.FNONOS parameter on the MODELS statement to a positive value. The efficiency is modified by an extra term

$$1.0 - Q_{floating} / (q\text{NT\_FNONOS}) \qquad \text{3-433}$$

where $Q_{floating}$ is the Floating gate charge density in C/micron. NT.FNONOS is the integrated trap density /micron. If this term is negative, then zero is used instead. This allows you to model the phenomenon of trap saturation.

Table 3-83. MODELS Statement

Parameter	Type	Default	Units
FNONOS	Logical	False	
ETA.FNONOS	Real	1.0	
BH.FNONOS	Real	3.07	eV
NT.FNONOS	Real	0.0	/um

3.6.7: Device Level Reliability Modeling

Hansch MOS Reliability Model

The Hansch Reliability Model [66,4,131] can be used to simulate MOS transistor degradation under stress conditions. The causes of device characteristic degradation are the hot electron (hole) injection into gate oxide, and the trapping of electron (hole) charge on the effective interface acceptor (donor) like traps.

The model calculates hot electron (hole) injection current according to the lucky electron model. The arbitrary position-dependent distributions of acceptor and donor-like traps are specified on the oxide-semiconductor interface with corresponding capture cross sections. The device degradation is calculated as a function of stress time by performing transient calculations. The trap rate equation is solved on every time-step and thus the trapped electron (hole) concentration is calculated. The rate of electron trapping can be described by the equations:

$$\frac{dN_n(x,t)}{dt} = \frac{\texttt{SIGMAE}}{q} \cdot J_{inj,\,n}(x,t) \cdot (\texttt{NTA}(x) - N(x,t)) \qquad \text{3-434}$$

$$\frac{dN(x,t)}{dt} = \frac{\texttt{SIGMAH}}{q} \cdot J_{inj,\,p}(x,t) \cdot (\texttt{NTD}(x) - N(x,t)) \qquad \text{3-435}$$

where N(x,t) represents the trapped electron (hole) density, at the interface point x, at time=t during a transient simulation. The NTA and NTD parameters represent the acceptor and donor-like trap densities at time=0. The $J_{inj,n}$(x,t) and $J_{inj,p}$(x,t) parameters are the injected electron and hole current densities, SIGMAE and SIGMAH are the capture cross section of electrons and holes.

To activate this model, use the DEVDEG, DEVDEG.E, and DEVDEG.H parameters in the MODELS statement (to account for both hot electron and hole injection, hot electron or hot hole injection, respectively). The model parameters are user-definable on the DEGRADATION statement.

Table 3-84. User-Definable Parameters for Equations 3-434 and 3-435

Statement	Parameter	Units
DEGRADATION	SIGMAE	cm^2
DEGRADATION	SIGMAH	cm^2
DEGRADATION	NTA/F.NTA	cm^2
DEGRADATION	NTD/F.NTD	cm^2

The results of stress simulation can be used to calculate the characteristics of the degraded device (the shift of the threshold voltage, transconductance degradation, and so on). You can view the distribution of traps, hot electron (hole) current density, and trapped electron (hole) distribution by using TONYPLOT.

The model parameters: NTA, NTD, SIGMAE, and SIGMAH can also be defined through the C-INTERPRETER functions: F.NTA, F.NTD, F.SIGMAE, and F.SIGMAH. This allows you to define these values as functions of their position (x,y) along the insulator-semiconductor interface. These C-function libraries are also defined on the DEGRADATION statement. More information on the C-INTERPRETER functions can be found in Appendix A: "C-Interpreter Functions".

3.6.8: The Ferroelectric Permittivity Model

Ferroelectric materials exhibit high dielectric constants, polarization and hysteresis. Such materials are finding more and more applications in integrated memory devices. To simulate these effects, a modified version of the ferroelectric model from Miller [110] has been implemented.

To enable the Ferroelectric Model, set the FERRO parameter in the MODELS statement. In this model the permittivity used in Poisson's Equation (Equation 3-1) is given the following functional form:

$$\varepsilon(E) = \mathrm{FERRO.EPSF} + \mathrm{FERRO.PS} \bullet 2\delta \bullet sech^{\,2}\left[\frac{E - \mathrm{FERRO.EC}}{2\delta}\right] \qquad 3\text{-}436$$

where FERRO.EPSF is the permittivity, E is the electric field and δ is given as follows:

$$\delta = \mathrm{FERRO.EC}\left[log\frac{1 + \mathrm{FERRO.PR/FERRO.PS}}{1 - \mathrm{FERRO.PR/FERRO.PS}}\right]^{-1} \qquad 3\text{-}437$$

The FERRO.EPSF, FERRO.PS, FERRO.PR, and FERRO.EC parameters can be modified in the MATERIAL statement (see Table 3-85).

The permittivity in Equation 3-435 can be replaced with a user-defined expression with the C-INTERPRETER. The F.FERRO parameter of the MATERIAL statement (see Chapter 19: "Statements", Section 19.24: "MATERIAL") defines the file that contains the C-function. This function allows the permittivity to be position and field dependent.

For more information about C-Interpreter, see Appendix A: "C-Interpreter Functions".

The derivative of the dipole polarization with respect to electric field is given by:

$$\frac{dP_d}{dE} = \Gamma\frac{dP_{sat}}{dE} \qquad 3\text{-}438$$

where P_d is the position dependent dipole polarization. A numeric integration of this function is carried out in ATLAS to determine the position dependent dipole polarization.

For saturated loop polarization, the Γ function is equal to unity, which corresponds to the default model. If you specify the UNSAT.FERRO parameter in the MODELS statement, the Γ function will take on a more general form suitable for simulation of unsaturated loops. In this case, the Γ function is given by:

$$\Gamma = 1 - tanh\left[\left(\frac{P_d - P_{sat}}{\xi P_s - P_d}\right)^{1/2}\right] \qquad 3\text{-}439$$

where $\xi = 1$ for increasing fields and $\xi = -1$ for decreasing fields.

Table 3-85. User-Specifiable Parameters for Equations 3-436 to 3-437

Statement	Parameter	Default	Units
MATERIAL	FERRO.EC	0.0	V/cm
MATERIAL	FERRO.EPS	1.0	
MATERIAL	FERRO.PS	0.0	C/sqcm
MATERIAL	FERRO.PR	0.0	C/sqcm

3.6.9: Epitaxial Strain Tensor Calculation in Wurtzite

The strain tensor in epitaxial layers is used to calculate piezoelectric polarization (Section 3.6.10: "Polarization in Wurtzite Materials [23]") or in gain modeling (Section 3.9.8: "Chuang's Three Band Model for Gain and Radiative Recombination in Wurtzite Materials") or both. In epitaxial layers, the strain tensor can be represented by ε_{xx}, ε_{yy}, ε_{zz}, ε_{xy}, ε_{yz} and ε_{zx}. The relationship between the various components of the strain tensor are given by Equations 3-92 through 3-387 as follows:

$$\varepsilon_{xx} = \varepsilon_{yy} = \frac{a_s - a_0}{a_0} \quad \text{3-440}$$

$$\varepsilon_{zz} = -2\frac{C_{13}}{C_{33}}\varepsilon_{xx} \quad \text{3-441}$$

$$\varepsilon_{xy} = \varepsilon_{yz} = \varepsilon_{zx} = 0 \quad \text{3-442}$$

where C_{13} and C_{33} are elastic constants, which can be specified by the parameters C_{13} and C_{33} on the MATERIAL statement. The default values for C_{13} and C_{33} are given for the GaN system in Section B.8: "Material Defaults for GaN/InN/AlN System".

The principal value of strain, ε_{xx}, can be specified or calculated. To specify ε_{xx}, assign the desired value to the STRAIN parameter of the REGION statement.

In Equation 3-318 a_s is the lattice constant in the layer in question. The parameter a_0 is the lattice constant in the "substrate". You can specify the lattice constant in the given layer by the ALATTICE parameter of the MATERIAL statement.

Default values for ALATTICE can be found for the GaN/AlN/InN system in Section B.8: "Material Defaults for GaN/InN/AlN System".

The "substrate" is more ambiguously defined so there are several ways to specify the "substrate" or the "substrate" lattice constant. First, you can specify a region as the "substrate" for strain calculations by specifying the logical parameter SUBSTRATE on the associated REGION statement. You can then specify the lattice constant for that region using the ALATTICE parameter of the corresponding MATERIALstatement.

Alternatively, you can directly specify the "substrate" lattice constant in the REGION statement, which make the strain calculations, using the ASUB parameter.

If the "substrate" lattice constant is not otherwise specified through the ASUB or SUBSTRATE parameters. The "substrate" lattice constant is taken as the average of the lattice constants of the two adjacent epitaxial layers (region above and below the region in question). If there is only one adjacent region, the lattice constant of that region is used as the "substrate" lattice constant.

3.6.10: Polarization in Wurtzite Materials [23]

Polarization in wurtzite materials is characterized by two conponents, spontaneous polarization, P_{sp}, and piezoelectric polarization, P_{pi}. Therefore, the total polarization, P_t, is given by:

$$P_t = \mathrm{PSP} + P_{pi} \quad \text{3-443}$$

where PSP is specified on the MATERIAL statement and specifies the total spontaneous polarization, P_{sp}, for the given material(s). The piezoelectric polarization, P_{pi}, is given by:

$$P_{pi} = 2\frac{a_s - a_0}{a_0}\left(\mathrm{E31} - \frac{\mathrm{C13}}{\mathrm{C33}}\mathrm{E33}\right) \quad \text{3-444}$$

where E31 and E33 are piezoelectric constants, and C13 and C33 are elastic constants all specified in the MATERIAL statement. The a_0 parameter is the lattice constant of the material layer in question, which can be specified by the ALATTICE parameter of the MATERIAL statement. The a_s parameter is the average value of the lattice constants of the layers directly above and below the layer in question.

To enable the polarization model, specify POLARIZATION in the REGION statement for the region for which you wish to characterize polarization effects. Typically, this will be a quantum well layer or active layer.

The polarization enters into the simulation as a positive and negative fixed charges appearing at the top (most negative Y coordinate) and bottom (most positive Y coordinate) of the layer in question. By default, the positive charge is added at the bottom and the negative charge is added at the top. You can modify the sign and magnitude of this charge by specifying POLAR.SCALE in the REGION statement. This parameter is multiplied by the polarization determined by Equation 3-443 to obtain the applied charge. The default value for POLAR.SCALE is 1.0.

In some cases, the introduction of polarization charges may introduce difficulties with convergence due to problems with initial guess. If these problems arise, you can use the PIEZSCALE parameter of the SOLVE statement to gradually introduce the effects of polarization. This parameter defaults to 1.0 and is multiplied by the net charge given by the product of the results of Equation 3-444 and the POLAR.SCALE parameter.

Table 3-86 shows the parameters of the wurtzite polarization model.

Table 3-86. User Specifiable Parameters of the Wurtzite Polarization Model

Statement	Parameter	Default	Units
MATERIAL	PSP	see Tables B-17-B-23	cm^{-2}
MATERIAL	ALATTICE	see Tables B-17-B-23	Å
MATERIAL	E13	see Tables B-17-B-23	cm^{-2}
MATERIAL	E33	see Tables B-17-B-23	cm^{-2}
MATERIAL	C13	see Tables B-17-B-23	GPa
MATERIAL	C33	see Tables B-17-B-23	GPa

3.6.11: Stress Effects on Bandgap in Si

Mechanical stress causes change in the band edges in silicon. These band edge shifts are given by the deformation potential theory [59]. The shifts for the conduction band edges are given by:

$$\Delta E_c^{(i)} = \texttt{D.DEFPOT}(\texttt{EPS11} + \texttt{EPS22} + \texttt{EPS33}) + \texttt{U.DEFPOT*EPSii} \quad \text{3-445}$$

where $\Delta E_c^{(i)}$ is the shift in the band edge of the ith ellipsoidal conduction band minima. The parameters U.DEFPOT and D.DEFPOT are the user-definable dialation and shear deformation potentials for the conduction band.

The parameters EPS11, EPS22, EPS33 are the user-definable diagonal components of the strain tensor. If you omit the definition of the strain tensor, it is automatically calculated using the expressions in Equations 3-402, 3-403 and 3-404.

The valence band edges are calculated by:

$$\Delta E_v^{(hl)} = \texttt{A.DEFPOT}(\texttt{EPS11} + \texttt{EPS22} + \texttt{EPS33}) \pm \sqrt{\xi} \quad \text{3-446}$$

where $\Delta E_v^{(hl)}$ are the band edge shifts in the light and heavy hole valence band maxima. The ξ parameter is given by:

$$\xi = \frac{\texttt{B.DEFPOT}^2}{2}\{(\texttt{EPS11}-\texttt{EPS22})^2 + (\texttt{EPS22}-\texttt{EPS33})^2 + (\texttt{EPS33}-\texttt{EPS11})^2\} \quad \text{3-447}$$

$$+ \texttt{C.DEFPOT}(\texttt{EPS12}^2 + \texttt{EPS23}^2 + \texttt{EPS31}^2)$$

where EPS12, EPS13 and EPS23 are the user-definable off diagonal components of the strain tensor. If these parameters are not defined, they are calculated using the expressions in Equations 3-440, 3-441 and 3-442.

In Equations 3-445 and 3-446, A.DEFPOT, B.DEFPOT and C.DEFPOT are user-definable valence band deformation portntial constants.

The user definable parameters are shown in Table 3-87.

Table 3-87. User Definable Parameters for Strained Silicon Band Gap

Statement	Parameter	Default	Units
MATERIAL	A.DEFPOT	2.1	eV
MATERIAL	B.DEFPOT	-2.33	eV
MATERIAL	C.DEFPOT	-4.75	eV
MATERIAL	D.DEFPOT	1.1	eV
MATERIAL	U.DEFPOT	10.5	eV
MATERIAL	EPS11	*	
MATERIAL	EPS22	*	

Table 3-87. User Definable Parameters for Strained Silicon Band Gap			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	EPS33	*	
MATERIAL	EPS12	*	
MATERIAL	EPS13	*	
MATERIAL	EPS23	*	

Note: * If unspecified, Equations 3-440, 3-441 and 3-442 calculate these parameters.

The net changes in the band edges, under Boltzman's statistics, are given by Equations 3-448 and 3-449.

$$\Delta E_c = kT ln\left[\sum_{=1}^{3} \frac{exp\left(-\frac{\Delta E_c^{(i)}}{kT}\right)}{3}\right] \quad 3\text{-}448$$

$$\Delta E_c = k_T ln\left[\frac{r}{1+r} exp\left(-\frac{\Delta E_v^{(i)}}{kT}\right) + \frac{1}{1+r} exp\left(-\frac{\Delta E_v^{(h)}}{kT}\right)\right] \quad 3\text{-}449$$

Here, the parameter r is given by Equation 3-450.

$$r = (m_l / m_h)^{3/2} \quad 3\text{-}450$$

In Equation 3-450, m_l and m_h are the effective masses of light and heavy holes as described in other places in this manual.

To enable the model for stress dependent band gap in silicon, specify the STRESS parameter of the MODELS statement.

3.6.12: Low Field Mobility in Strained Silicon

For Boltzman's statistics, the following expressions can be used for strain dependent electron and hole low field mobilities in silicon [49]:

$$\mu_n = \mu_{n\phi}\left\{1 + \frac{1 - ML/MT1}{1 + 2(ML/MT1)}\left[exp\left(\left(\frac{\Delta E_c - \Delta E_c^{(i)}}{kT}\right) - 1\right)\right]\right\} \quad \text{3-451}$$

$$\mu_p = \mu_{p\phi}\left\{1 + (\texttt{EGLEY.R} - 1)\frac{(MLH/MHH)^{1.5}}{1 + (MLH/MHH)^{1.5}}\left[exp\left(\left(\frac{\Delta E_v^{(l)} - \Delta E_v{}^{(h)}}{kT}\right) - 1\right)\right]\right\} \quad \text{3-452}$$

where $\Delta E_c^{(i)}$, ΔE_c, $\Delta E_v^{(l)}$, and $\Delta E_v{}^{(h)}$ are given by Equations 3-445, 3-448, and 3-446 respectively. Table 3-88 shows user-definable parameters for this model.

Table 3-88. User Definable Parameters for the Strained Silicon Low Field Mobility Model

Statement	Parameter	Default	Units
MATERIAL	ML	0.916	
MATERIAL	MT1	0.191	
MATERIAL	MLH	0.16	
MATERIAL	MHH	0.49	
MOBILITY	EGLEY.R	2.79	

To enable the strained silicon low field mobility model, specify EGLEY.N for electrons and EGLEY.P for holes on the MOBILITY statement .

3.7: Quasistatic Capacitance - Voltage Profiles

Quasistatic Capacitance is calculated by specifying the QSCV parameter in the SOLVE statement. The quasistatic capacitance is obtained for the electrode (whose bias is being ramped) by subtracting the electrode charge on the electrode at one bias from that at the adjacent bias and dividing by the Voltage increment. The charge density is calculated by applying Gauss' Flux Theorem to the electrode. This gives the capacitance at the midpoint between the two bias points. ATLAS does not correct the output for this because the voltage increment should be sufficient fine. Therefore, this small shift is negligible. A fine voltage increment will also give a good approximation to the continuous derivative.

QSCV will work with CARRIERS=0, 1 or 2 specified in the METHOD statement. If you link electrodes using the COMMON parameter of the CONTACT statement, then the capacitance for each one and the sum of capacitances will be calculated. If you specify MULT and FACTOR for a linked electrode, then the capacitance will be calculated for the appropriate bias points, but will be stored as a function of the bias applied to the COMMON electrode. In this case, scaling or shifting of the C-V curve may be necessary.

3.8: Conductive Materials

In certain cases, it may be advantageous to simulate metal conductivity directly rather than handling electrodes as boundaries. You can define metal regions as "conductive" by specifying the CONDUCTOR parameter of the REGION statement. This might be useful, for example in simulating self-heating in metal regions.

When the metal is treated as a conductor, the conduction equation for all points in the region are solved as follows:

$$J = E/RESISTIVITY \qquad 3\text{-}453$$

where J is the current density, E is the electric field, and RESISTIVITY is the metal resistivity in metal resistivity in $\mu\Omega\cdot$cm. To specify the metal resistivity, use the RESISTIVITY parameter from the MATERIAL statement. To specify the thermal coefficient of resistivity, use the DRHODT parameter from the MATERIAL statement.

3.8.1: Conductors with Interface Resistance

You can add interface contact resistance at interfaces between conductor and semiconductor regions. To do this, tassign a positive value to the INT.RESIST parameter of the INTERFACE statement. This value corresponds to the interface resistivity in units of Ωcm^2. To acheive the desired effect, also specify S.C on the INTERFACE statement. The interface resistance in completely analogous to using contact resistance (the CON.RESIST parameter of the CONTACT statement) only for conductor-semiconductor interfaces.

3.9: Optoelectronic Models

This section discusses various physical models used to simulate optoelectronic devices. These models predict fundamental optoelectronic processes, such as absorption and gain and radiative recombination rates versus material composition, temperature and optical wavelength.

These models are based on various band theories and account for the following:

- the existence of multiple valence bands supporting multiple optical transitions,
- asymmetry in conduction band effective masses,
- the effects of strain on the band parameters,
- the effects of quantum confinement on allowable transitions.

In the following paragraphs, we will discuss three optoelectronic models.

With respect to their various principal investigators they are named the Yan model, the Li model and the Chuang model. To activate these models, specify YAN, LI or CHUANG in the MODELS statement. Yan and Li's models are applicable to zincblende materials while Chuang's model applies to wurtzite materials. Yan's model is based on a single valence band, Li's is based on two valence bands (light holes and heavy holes), and Chuang's model is based on three valence bands (light holes, heavy holes and crystal split-off holes).

In addition to the band structure based models, there are certain, more general but less physical models that are also mentioned. All of these models are used in the following applications:

- General drift-diffusion to account for radiative recombination for any semiconductor device,
- Laser and VCSEL simulation in both stimulated emission gain and spontaneous emission (see Chapters 8: “Laser: Edge Emitting Simulator” and 9: “VCSEL Simulator”),
- LED light emission (see Chapter 11: “LED: Light Emitting Diode Simulator”).

3.9.1: The General Radiative Recombination Model

At the most, fundamental level the radiative recombination model described in Equation 3-454 describes all the most salient features of radiative recombination except spectral content. You can, however, use this model in all three applications mentioned above. It has the advantages of being fast, simple and easily calibrated.

To enable the general model, specify OPTR in the MODEL statement. This will enable radiative recombination using the general model in the drift diffusion part of the simulation. To enable this model in LASER or VCSEL, disable the default model for LASER and VCSEL by specifying ^SPONTANEOUS in the LASER statement. To use the general model for LED, make sure no other competing mechanisms are enabled. The disadvantage of using this model for any light emission application is that it lacks spectral information.

When used in LASER and VCSEL the spontaneous recombination rate is given by:

$$r(r,z) = \frac{\texttt{EMISSION\_FACTOR} \cdot \texttt{COPT} \cdot \left(n \cdot p - n_i^2\right)}{N_l} \qquad \text{3-454}$$

where n and p are the electron and hole concentrations, n_i is the intrinsic concentration, N_l is the number of longitudinal modes, EMISSION_FACTOR and COPT are user-defined parameters on the MATERIAL statement. COPT accounts for the radiative rate in all directions and energies. EMISSION_FACTOR represents the fraction of energy coupled into the direction of interest and in the energy range of interest. Note that Equation 3-454 contains no spectral information.

3.9.2: The Default Radiative Recombination Model

The default spontaneous radiative recombination model is given by

$$r_{sp_m}(r,z) = \texttt{ESEP} \cdot \texttt{EMISSION\_FACTOR}\frac{c}{\texttt{NEFF}} \cdot D(E) \cdot \texttt{GAIN0} \cdot \sqrt{\frac{\hbar\omega - E_g}{kT}} \cdot f\left(E_c - E_{fn} + \texttt{GAMMA}\frac{(\hbar\omega - E_g)}{kT}\right) \cdot \left[1 - f\left(E_v - E_{fp} - \frac{(1-\texttt{GAMMA})(\hbar\omega - E_g)}{kT}\right)\right] \quad 3\text{-}455$$

where:

- c is the speed of light,
- $\hbar$ Planck's constant,
- k Boltzman's constant,
- E_g is the energy bandgap,
- E_c and E_v are the conduction and valence band edge energies,
- T is the lattice temperature,
- E_{fn} and E_{fp} are the electron and hole quasi-Fermi energies,
- ω is the emission frequency that corresponds to the transition energy E,
- *D(E)* is the optical mode density,
- EMISSION.FACTOR, GAIN0 and GAMMA are user defined parameters from the MATERIAL statement,
- ESEP and NEFF are user-defined parameters specified from the LASER statement.

The optical mode density *D(E)* is given by

$$D(E) = \frac{n^3 E^2}{\pi^2 \hbar^3 c^3} \quad 3\text{-}456$$

where n is the index of refraction.

The function f(x) is given by

$$f(x) = \frac{1}{1 + exp(x)}. \quad 3\text{-}457$$

3.9.3: The Standard Gain Model

The standard gain model [115] is enabled by specifying G.STANDARD in the MODELS statement. The standard gain model is given by

$$g(r,z) = \texttt{GAIN0}\sqrt{\frac{\hbar\omega - E_g}{kT}}\left[f\left(\frac{E_c - E_{fn} + \texttt{GAMMA}(\hbar\omega - E_g)}{kT}\right) - f\left(\frac{E_v - E_{fp} - (1-\texttt{GAMMA})(\hbar\omega - E_g)}{kT}\right)\right] \quad 3\text{-}458$$

where:

- $\hbar$ is Planck's constant,
- E_g is the energy bandgap,
- k is Boltzman's constant,
- T is the lattice temperature,
- E_{fn} and E_{fp} are the electron and hole quasi-Fermi energies,

- ω is the emission frequency,
- E_v and E_c are the valence and conduction band edge energies,
- the function f is defined in Equation 3-457,
- GAMMA and GAIN0 user-definable parameters specified on the MATERIAL statement.

Table 3-89 describes the user defined parameters of Equation 3-458.

Table 3-89. User-Specifiable Parameters for Equation 3-458

Statement	Parameter	Default	Units
MATERIAL	GAIN0	2000.0	cm^{-1}
MATERIAL	GAMMA		

If GAMMA in Equation 3-458 is not specified, it is automatically calculated from the following expression:

$$\mathrm{GAMMA} = \frac{1}{\left(\frac{N_c}{N_v}\right)^{\frac{2}{3}} + 1} \qquad 3\text{-}459$$

where N_c and N_v are the conduction and valence band densities of states.

3.9.4: The Empirical Gain Model

The emipircal gain model is enabled by specifying G.EMPIRICAL in the MODELS statement. The model is described by the following expression:

$$g(r,z) = \mathrm{GAIN00} + \mathrm{GAIN1N} \cdot n + \mathrm{GAIN1P} \cdot p + \mathrm{GAIN2NP} \cdot np + \mathrm{GAIN1MIN} \cdot min(n,p) \qquad 3\text{-}460$$

where n and p are the electron and hole concentrations, and GAIN00, GAIN1N, GAIN1P, GAIN2NP and GAIN1MIN are user-specified parameters from the MATERIAL statement. Note that the empirical model contains no spectral dependency and should not be used for LASER or VCSEL simulations with multiple logitudinal modes. Table 3-90 shows the user-definable parameters for Equation 3-460.

Table 3-90. User-Specifiable Parameters for Equation 3-460

Statement	Parameter	Default	Units
MATERIAL	GAIN00	-200.0	cm^{-1}
MATERIAL	GAIN1P	0	cm^2
MATERIAL	GAIN2NP	0	cm^2
MATERIAL	GAIN1NP	0	cm^5
MATERIAL	GAIN1MIN	3.0×10^{-16}	cm^2

3.9.5: Tayamaya's Gain Model

Tayamaya's gain model [128] is enabled by specifying TAYAMAYA in the MODELS statement. Tayamaya's model is described by the following expression:

$$g(r,z) = \mathrm{GN1}\left(n(r,z) - \mathrm{NTRANSPARENT}\right) \qquad n > \mathrm{NTRANSPARENT}$$
$$g(r,z) = \mathrm{GN2}\left(n(r,z) - \mathrm{NTRANSPARENT}\right) \qquad n <= \mathrm{NTRANSPARENT} \qquad \text{3-461}$$

where n is the electron concentration and GN1, GN2 and NTRANSPARENT are user-specified parameters from the MATERIAL statement. As in the empirical gain model, this model contains no spectral dependency and should not be used to simulate multiple longitudinal modes in LASER or VCSEL. Table 3-91 shows the user definable parameters for Equation 3-461.

Table 3-91. User-specified values for Equation 3-461

Statement	Parameter	Default	Units
MATERIAL	GN1	3.0×10^{-16}	cm^2
MATERIAL	GN2	4.0×10^{-15}	cm^2
MATERIAL	NTRANSPARENT	2.0×10^{18}	cm^{-3}

3.9.6: Band Stucture Dependent Optoelectronic Models

Although the simple models described in Sections 3.9.1: "The General Radiative Recombination Model" through 3.9.5: "Tayamaya's Gain Model" are useful for modeling the principal effects, they suffer from one or more of the following drawbacks:

- they lack spectral dependencies,
- they are not generally calibrated to all tools,
- they do not account for effects of strain,
- they do not account for the effects of quantum confinement,
- the lack physical basis.

The following sections will describe more physically based models. Figure 3-8 illustrates the simulation flow for these models.

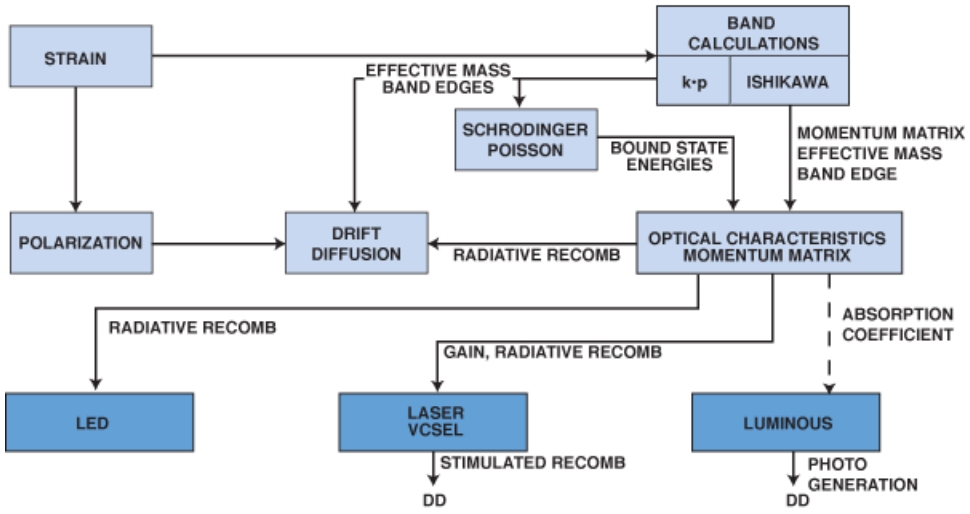


Figure 3-8: Simulation Flow For Physically Based Optoelectronic Models

First, strain is introduced either directly or calculated from lattice mismatch and used to calculate the strain effects on band calculations. The strain is also introduced to polarization calculations that enter directly into the drift diffusion calculations as polarization fields.

Next, the band parameters (band edges and effective masses) are used directly in the drift diffusion simulations as well as feeding into the solutions of Schrodinger's equations to calculate the bound state energies. The band parameters and bound state energies are then used to calculate gain, radiative recombination rate and optical absorption.

3.9.7: Yan's and Li's Models for Gain and Radiative Recombination in Zincblende Materials

You can select one of two models for optical gain and spontaneous recombination. If the LI parameter is specified, the model by Li [96] will be used. If the YAN parameter is specified the model by Yan [187] will be used. The difference between the models is that in the YAN Model only one valence band is accounted for, whereas the LI Model accounts for both light and heavy holes in the valence band.

The first step in calculating the gain and radiative recombination is to calculate the bound state energies. The band edge energies and effective masses are used to calculate the quantum well bound state energies through the Schrodinger's Equation (See Chapter 13: “Quantum: Quantum Effect Simulator”, Equations and 13-2). This calculation is performed in the same manner as the Self-Consistent Coupled Schrodinger Poisson Model. The calculation of the bound state energies is performed over a discrete domain that isn’t the same as the device simulation mesh. The discrete domain is specified by the WELL.NX and WELL.NY parameters in the REGION statement.

The WELL.NX and WELL.NY parameters specify the number of uniform mesh locations in the bounding box where the solution of Schrodinger's Equation is performed to extract bound state energies. Typically, WELL.NY should be set to a large number so that there will be several samples per well. WELL.NX should be comparable to the number of grid lines in the device mesh over the same extent in the X direction.

Once the bound state energies are calculated, the bulk momentum matrix element is calculated as shown in Equation 3-462.

$$\left|M_{avg}\right|^2 = \frac{m_0}{6}\left(\frac{m_0}{m^*} - 1\right)\frac{E_g(E_g + \mathrm{DELTA})}{\left(E_g + \frac{2}{3}\mathrm{DELTA}\right)} \qquad 3\text{-}462$$

where m^* is the effective mass, E_g is the bandgap, and WELL.DELTA is the user-specifiable spin-orbital splitting energy as described in Table 3-92.

Table 3-92. User-Specifiable parameters for Equation 3-462

Statement	Parameter	Default	Units
MATERIAL	WELL.DELTA	0.341	eV

Next, the bulk momentum matrix element is used to calculate the quantum well matrix element. If the YAN Model is specified, Equation 3-463 will be used to calculate the quantum well momentum matrix element.

$$\left|M_{qw}\right|^2 = \frac{3}{4}\left|M_{avg}\right|^2 \qquad 3\text{-}463$$

If the LI Model is specified the matrix elements for light and heavy holes are given by Equation 3-464.

$$\begin{aligned} M_{hh} &= A_{hh}M_{avg} \\ M_{lh} &= A_{lh}M_{avg} \end{aligned} \qquad 3\text{-}464$$

where *Ahh* and *Alh* are anisotropy factors for heavy and light holes.

The anisotropy factors depend upon whether TM or TE modes are the dominant modes [96]. The dominant mode is user-specifiable by using the TE.MODES parameter of the MATERIAL statement. When this parameter is true (default), the TE mode models are used and when false the TM mode model is used.

For TE modes, the values of the anisotropy factors are given by Equation 3-465.

$$\begin{aligned} A_{hh} &= \frac{3+3E_{ij}/E}{4}, \qquad A_{lh} = \frac{5-3E_{ij}/E}{4} \qquad for(E > E_{ij}) \\ A_{hh} &= 3/2, \qquad A_{lh} = 1/2 \qquad for(E \le E_{ij}) \end{aligned} \qquad 3\text{-}465$$

For TM modes, the values of the anisotropy factors are given by Equation 3-466.

$$\begin{aligned} A_{hh} &= \frac{3-3E_{ij}/E}{2} \qquad A_{hh} = \frac{1+3E_{ij}/E}{2} \qquad for(E > E_{ij}) \\ A_{hh} &= 0 \qquad Alh = 2 \qquad for(E < E_{ij}) \end{aligned} \qquad 3\text{-}466$$

Here, *E* is the transition energy and *Eij* is the energy difference between the *ith* conduction bound state and the *jth* valence band state.

The bound state energies and effective masses are then used to calculate the Fermi functions given in Equation 3-467.

$$\begin{aligned} f_i &= \left\{1 + exp\left[\left(E_i - \frac{m_{ij}}{m_i}(E - E_{ij}) - E_{fp}\right)/kT\right]\right\}^{-1} \\ f_j &= \left\{1 + exp\left[\left(E_j - \frac{m_{ij}}{m_j}(E - E_{ij}) - E_{fn}\right)/kT\right]\right\}^{-1} \end{aligned} \qquad 3\text{-}467$$

Here, *Efp* is the hole Fermi level, *Efn* is the electron Fermi level, E_i is the valence band energy, and E_j is the conduction band energy.

Next, the reduced effective masses are used to calculate the 2D density of states as in Equation 3-468.

$$\rho = \frac{m}{\pi \hbar^2 t} \tag{3-468}$$

where t is the quantum well width.

Next, the optical mode density is calculated as given by Equation 3-469.

$$D(E) = \frac{n^3 E^2}{\pi^2 \hbar^3 c^3} \tag{3-469}$$

Optical Gain Models

For the LI Model, the optical gain is given by Equation 3-470.

$$g_{ij}\left(E_{ij}^0\right) = \left(\frac{2\pi}{\hbar}\right)\left|H_{ij}\right|^2 (f_j' - f_i')\left(\frac{\varepsilon_1}{nc}\right)\rho \cdot \texttt{WELL.GAIN}$$
$$\left|H_{ij}\right|^2 = \left(\frac{q}{m_0}\right)^2 \left(\frac{2\hbar\omega}{4\varepsilon_1 \varepsilon_0 \omega^2}\right) M_{ij}^2 O_{ij}^2 \tag{3-470}$$

where H_{ij} is the Hamiltonian matrix element, w is the frequency, M_{ij} is either M_{hh} or M_{eh}, O_{ij} is the overlap integral, and `WELL.GAIN` is a user-specifiable scale factor on the `REGION` statement.

For the YAN Model, the optical gain is given by Equation 3-471.

$$g(E) = \left(\frac{\pi e^2 \hbar}{\varepsilon_0 n c m_0^2 E}\right)\left|M_{qw}\right|^2 \rho \cdot D(E)(f_j' - f_i') \tag{3-471}$$

Spontaneous Recombination Models

For the LI Model, the spontaneous recombination is given by Equation 3-472.

$$r_{sp}(E) = \sum_{i,j}\left(\frac{2\pi}{\hbar}\right)\left|H_{ij}\right|^2 f_j' (1 - f_i')D(E)\rho \tag{3-472}$$

For the YAN Model, the spontaneous recombination is given by Equation 3-473.

$$r_{sp}(E) = \left(\frac{2\pi}{\hbar}\right)\left(\frac{e}{m_0}\right)^2 \left(\frac{2\hbar\omega/\varepsilon}{4\omega^2}\right)\left|M_{qw}\right|^2 \rho \cdot D(E) \cdot f_j' (1 - f_i') \tag{3-473}$$

3.9.8: Chuang's Three Band Model for Gain and Radiative Recombination in Wurtzite Materials

Chuang's model [36,37,90] is derived from the k*p method for three valence bands in wurtzite crystalline structure. Given the assumptions of parabolic bands, no valence band mixing and momentum approaching zero, the following approach can be used. First, we calculate the parameters from Equations 3-474 and 3-475.

$$\theta_\varepsilon = D_3\varepsilon_{zz} + D_4(\varepsilon_{xx} + \varepsilon_{yy}) \quad 3\text{-}474$$

$$\lambda_\varepsilon = D_1\varepsilon_{zz} + D_2(\varepsilon_{xx} + \varepsilon_{yy}) \quad 3\text{-}475$$

Here, D_1, D_2, D_3, and D_4 are shear deformation potentials and ε_{xx}, ε_{yy}, and ε_{zz}, are taken from the strain tensor calculations described in Section 3.6.9: "Epitaxial Strain Tensor Calculation in Wurtzite". Next, we can calculate the valence band energies from Equations 3-476 through 3-478.

$$E_{hh}^0 = E_v^0 + \Delta_1 + \Delta_2 + \theta_\varepsilon + \lambda_\varepsilon \quad 3\text{-}476$$

$$E_{lh}^0 = E_v^0 + \frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2} + \lambda_\varepsilon + \sqrt{\left(\frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2}\right)^2 + 2\frac{\Delta^2}{3}} \quad 3\text{-}477$$

$$E_{ch}^0 = E_v^0 + \frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2} + \lambda_\varepsilon + \sqrt{\left(\frac{\Delta_1 - \Delta_2 + \theta_\varepsilon}{2}\right)^2 + 2\frac{\Delta^2}{3}} \quad 3\text{-}478$$

Here, Δ_1, Δ_2 and Δ_3 are split energies and E_v is the valence band reference level. Next, we can calculate the hydrostatic energy shift from Equation 3-479.

$$P_{c\varepsilon} = a_{cz}\varepsilon_{zz} + a_{ct}(\varepsilon_{xx} + \varepsilon_{yy}) \quad 3\text{-}479$$

Here, a_{cz} and a_{ct} are hydrostatic deformation potentials. From which we can calculate the conduction band energy as given in Equation 3-480.

$$E_c^0 = E_v^0 + \Delta_1 + \Delta_2 + E_g + P_{c\varepsilon} \quad 3\text{-}480$$

Here, E_g is the energy bandgap.

Next, we can calculate the effective masses in the various bands using the expressions in Equations 3-481 through 3-486.

$$m_{hh}^z = -m_0(A_1 + A_3)^{-1} \quad 3\text{-}481$$

$$m_{hh}^t = -m_0(A_2 + A_4)^{-1} \quad 3\text{-}482$$

$$m_{lh}^z = -m_0\left[A_1 + \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right)A_3\right]^{-1} \quad 3\text{-}483$$

$$m_{lh}^{t} = -m_0\left[A_2 + \left(\frac{E_{lh}^{0} - \lambda_{\varepsilon}}{E_{lh}^{0} - E_{ch}^{0}}\right)A_4\right]^{-1} \quad 3\text{-}484$$

$$m_{ch}^{z} = -m_0\left[A_1 + \left(\frac{E_{ch}^{0} - \lambda_{\varepsilon}}{E_{ch}^{0} - E_{lh}^{0}}\right)A_3\right]^{-1} \quad 3\text{-}485$$

$$m_{lh}^{t} = -m_0\left[A_2 + \left(\frac{E_{lh}^{0} - \lambda_{\varepsilon}}{E_{lh}^{0} - E_{ch}^{0}}\right)A_4\right]^{-1} \quad 3\text{-}486$$

Here, m_{ϕ} is the free space mass of an electron, m_{hh}^{z} , m_{hh}^{t}, m_{lh}^{z}, m_{lh}^{t}, m_{ch}^{z} and m_{ch}^{t} are the effective masses for heavy holes, light holes and crystal split off holes in the axial and transverse directions and A_1, A_2, A_3 and A_4 are hole effective mass parameters.

The momentum matrix elements for the various transitions can be calculated by Equations 3-499 through 3-492.

$$m_{hh}11 = 0 \quad 3\text{-}487$$

$$m_{hh}11 = b^2\left(\frac{m_0}{2}E_{pz}\right) \quad 3\text{-}488$$

$$m_{ch}11 = a^2\left(\frac{m_0}{2}E_{pz}\right) \quad 3\text{-}489$$

$$m_{hh}\perp = \frac{m_0}{4}E_{px} \quad 3\text{-}490$$

$$m_{lh}\perp = a^2\left(\frac{m_0}{4}\right)E_{px} \quad 3\text{-}491$$

$$m_{ch}\perp = b^2\left(\frac{m_0}{4}\right)E_{px} \quad 3\text{-}492$$

Here, E_{px} and E_{pz} are given by Equations 3-493 and 3-494, a^2 and b^2 are given by Equations 3-497 and 3-498. The values of P_1^2 and P_2^2 are given by Equations 3-495 and 3-496.

$$E_{px} = \left(2m_0/\hbar^2\right)P_2^2 \quad 3\text{-}493$$

$$E_{pz} = \left(2m_0/\hbar^2\right)P_1^2 \quad 3\text{-}494$$

$$P_1^2 = \frac{\hbar^2}{2m_0}\left(\frac{m_0}{m_e^z} - 1\right)\frac{(E_g + \Delta_2 + \Delta_2)(E_g + 2\Delta_2) - 2\Delta_3^2}{(E_g + 2\Delta_2)} \quad 3\text{-}495$$

$$P_2^2 = \frac{\hbar^2}{2m_0}\left(\frac{m_0}{m_e^t} - 1\right)\frac{E_g[(E_g + \Delta_1 + \Delta_2)(E_g + 2\Delta_2) - 2\Delta_3^2]}{(E_g + \Delta_1 + \Delta_2)(E_g + \Delta_2) - \Delta_3^2} \quad 3\text{-}496$$

$$a^2 = \left(\frac{E_{lh}^0 - \lambda_\varepsilon}{E_{lh}^0 - E_{ch}^0}\right) \quad 3\text{-}497$$

$$b^2 = \left(\frac{E_{ch}^0 - \lambda_\varepsilon}{E_{ch}^0 - E_{lh}^0}\right) \quad 3\text{-}498$$

In Equations 3-493 through 3-498, m_e^t and m_e^z are the transverse and axial conduction band effective masses.

The default values for the parameters of Equations 3-474 through 3-498 are shown in Appendix B: "Material Systems", Section B.8: "Material Defaults for GaN/InN/AlN System".

3.9.9: Lorentzian Gain Broadening

Gain broadening due to intra-band scattering can be introduced by specifying LORENTZ in the MODEL statement. The following equation describes gain broadening when it's applied.

$$g = \int g(E)L(E - E_{ij})dE \quad 3\text{-}499$$

The Lorentzian shape function is given by Equation 3-500.

$$L(E - E_{ij}) = \frac{1}{\pi} \cdot \frac{\text{WELL.GAMMA0}}{(E - E_{ij})^2 + \text{WELL.GAMMA0}^2} \quad 3\text{-}500$$

where WELL.GAMMA0 is user-specifiable in the MATERIAL statement.

3.9.10: Ishikawa's Strain Effects Model

The strained layer InGaAsP and InGaAlAs quantum well material models from [75] are implemented in the ATLAS simulator. To enable these models, specify the ISHIKAWA parameter in the MQW statement (see Chapter 19: "Statements", Section 19.30: "MQW"). Strain percentages are specified by the STRAIN parameter in the MQW statement. STRAIN between a substrate material with lattice constant, d_s, and an epitaxial material without STRAIN, with lattice constant, d_e, is $(d_e - d_s) | d_s$.

When you enable this model, the band edge parameters for materials in the InGaAsP and InGaAlAs systems are calculated using Equations 3-501, 3-502,and 3-503.

InGaAs

$$E_c = 1.040 - 0.0474\,\mathrm{STRAIN} + 0.003303\,\mathrm{STRAIN}^2$$

$$E_{v,HH} = 0.3331 + 0.05503\,\mathrm{STRAIN} - 0.002212\,\mathrm{STRAIN}^2$$

$$E_{v,LH} = 0.331 - 0.01503\varepsilon - 0.003695\,\mathrm{STRAIN}^2 \qquad \text{3-501}$$

InGaAsP

$$a)\,\mathrm{STRAIN} < 0$$

$$E_c = (0.6958 + 0.4836E_g) - 0.03031\,\mathrm{STRAIN}$$

$$E_{v,LH} = (0.5766 - 0.3439E_g) - 0.3031\,\mathrm{STRAIN} \qquad \text{3-502}$$

$$b)\,\mathrm{STRAIN} > 0$$

$$E_c = (0.6958 + 0.4836E_g) + 0.003382\,\mathrm{STRAIN}$$

$$E_{v,HH} = (0.6958 - 0.5164E_g) + 0.003382\,\mathrm{STRAIN}$$

InGaAlAs

$$a)\,\mathrm{STRAIN} < 0$$

$$E_c = (0.5766 + 0.6561E_g) - 0.02307\,\mathrm{STRAIN}$$

$$E_{v,LH} = (0.5766 - 0.3439E_g) - 0.2307\,\mathrm{STRAIN} \qquad \text{3-503}$$

$$b)\,\mathrm{STRAIN} > 0$$

$$E_c = (0.5766 + 0.06561E_g) + 0.01888\,\mathrm{STRAIN}$$

$$E_{v,HH} = (0.5766 - 0.3439E_g) + 0.01888\,\mathrm{STRAIN}$$

A STRAIN parameter has also been added to the REGION statement to account for strain in the bulk materials. Note that for the InGaAs material system, the equations ignore composition fraction and variation in the parameters is accounted strictly through strain (see Equation 3-501). The compositional variation can be seen in *Figure 5.a* from [96].

These band edge parameters are used through all subsequent calculations. Most importantly, the band edges are used in solving the Schrodinger's Equation (See Chapter 13: "Quantum: Quantum Effect Simulator", Equations and 13-2) to obtain the bound state energies in multiple quantum wells.

If you enable the Ishikawa Model, this would include the effects of strain in the valence and conduction band effective masses as described in [96]. The effects of strain are introduced in Equation 3-504.

$$\frac{1}{m_{lh}} = \mathrm{ASTR} + \mathrm{BSTRSTRAIN} + \mathrm{CSTRSTRAIN}^2$$
$$\frac{1}{m_{hh}} = \mathrm{DSTR} + \mathrm{ESTRSTRAIN} + \mathrm{FSTRSTRAIN}^2 \qquad 3\text{-}504$$

The ASTR, BSTR, CSTR, DSTR, ESTR, and FSTR parameters are user-definable in the MQW statement. You can also choose the appropriate values for these parameters from Tables 3-93 or 3-94.

Table 3-93. In-Plane Effective Mass of InGaAsP Barrier (1.2µm)/InGaAs(P) Well and InGaAlAs Barrier (1.2µm)/InGa(Al)As Well System for 1.55µm Operation [96]

–2.0<Strain(%)<–0.5				0<Strain(%)<2.0		
	ASTR	BSTR	CSTR	DSTR	ESTR	FSTR
Lattice-matched InGaAsP barrier/InGaAs well	–4.238	–6.913	–1.687	4.260	2.253	–0.584
InGaAsP (1.6-µm) well	–0.936	–4.825	–1.754	2.986	5.505	–1.736
Strain-compensated InGaAsP barrier/InGaAs well	–7.761	–13.601	–5.100	4.166	1.933	–0.558
Lattice-matched InGaAlAs barrier/InGaAs well	–3.469	–6.133	–1.481	3.9134	2.336	–0.633
InGaAlAs(1.6-µm) well	–1.297	–5.598	–2.104	2.725	6.317	–1.766
Strain-compensated InGaAlAs barrier/InGaAs well	–5.889	–9.467	–2.730	4.193	1.075	–-0.155

Table 3-94. In-Plane Effective Mass of InGaAsP Barrier (1.1µm)/InGaAs(P) Well and InGaAlAs Barrier (1.1µm)/InGa(Al)As Well System for 1.30µm Operation [96]

–2.0<Strain(%)<–0.5				–0.5<Strain(%)<2.0		
	ASTR	BSTR	CSTR	DSTR	ESTR	FSTR
Lattice-matched InGaAsP barrier/InGaAs well	–9.558	–8.634	–1.847	4.631	1.249	–0.313
InGaAsP (1.4-µm) well	–2.453	–4.432	–1.222	3.327	3.425	–1.542
Strain-compensated InGaAsP barrier/InGaAs well				4.720	0.421	–0.014
Lattice-matched InGaAlAs barrier/InGaAs well	–8.332	–8.582	–2.031	3.931	1.760	–0.543
InGaAlAs(1.4-µm) well	–3.269	–5.559	–1.594	3.164	4.065	–1.321
Strain-compensated InGaAlAs barrier/InGaAs well				4.078	0.516	–0.0621

To choose these values, use the STABLE parameter in the MQW statement. You can set this parameter to the row number in the tables. The rows are numbered sequentially. For example, the first row in Table 3-93 is selected by specifying STABLE=1. The first row of Table 3-94 is selected by specifying STABLE=7.

You can also choose to specify the effective masses directly. The conduction band effective mass is specified by the MC parameter in the MQW statement. There are two ways to specify the valence band. One way, is to use the MV parameter to specify a net effective mass. The other way, is to use the MLH and MHH parameters to specify the light and heavy hole effective masses individually.

If you don't specify the effective masses by using one of these methods, then the masses will be calculated from the default density of states for the given material as described in Section 3.4.2: "Density of States".

The effective masses described are also used to solve the Schrodinger Equation (see Chapter 13: "Quantum: Quantum Effect Simulator", Equations and 13-2).

3.10: Optical Index Models

The index of refraction of materials are used in several places in the ATLAS simulation environment. For LUMINOUS and LUMINOUS3D, the index is used to calculate reflections during raytrace. For LASER and VCSEL, the index is key to determining optical confinement. For LED, the index is used to calculate reflections for reverse raytrace.

In the ATLAS environment there are a variety of ways the optical index of refraction can be introduced as well as resonable energy dependent defaults for many material systems. For some these defaults are tabular (see Table B-36). For others there exist analytic functions. Here we describe some of those analytic models.

3.10.1: The Sellmeier Dispersion Model

To enable the Sellmeier dispersion model, specify NDX.SELLMEIER on the MATERIAL statement. The Sellmeier dispersion model is given in Equation 3-505.

$$n_r(\lambda) = \sqrt{\mathtt{S0SELL} + \frac{\mathtt{S1SELL}\lambda^2}{\lambda^2 - \mathtt{L1SELL}^2} + \frac{\mathtt{S2SELL}\lambda^2}{\lambda^2 - \mathtt{L2SELL}^2}} \quad \text{3-505}$$

In this equation, λ is the wavelength in microns, and S0SELL, S1SELL, S2SELL, L1SELL and L2SELL are all user-definable parameters on the MATERIAL statement. Default values for these parameters of various materials are given in Table B-37.

3.10.2: Adachi's Dispersion Model

To enable a model by Adachi [1], specify NDX.ADACHI on the MATERIAL statement. This model is given by Equations 3.429, 3-430 and 3-431.

$$n_r(w)^2 = \mathtt{AADACHI}\left[f(x_1) + 0.5\left(\frac{E_g}{E_g + \mathtt{DADACHI}}\right)^{1.5} f(x_2)\right] + \mathtt{BADACHI} \quad \text{3-506}$$

$$f(x_1) = \frac{1}{x_1^2}\left(2 - \sqrt{1+x_1} - \sqrt{1-x_1}\right) \quad , x_1 = \frac{\hbar\omega}{E_g} \quad \text{3-507}$$

$$f(x_2) = \frac{1}{x_2^2}\left(2 - \sqrt{1+x_2} - \sqrt{1-x_2}\right) \quad , x_2 = \frac{\hbar\omega}{E_g + \mathtt{DADACHI}} \quad \text{3-508}$$

In these equations, E_g is the band gap, $\hbar$ is plank's constant, ω is the optical frequency, and AADACHI, BADACHI and DADACHI are user-specifiable parameters on the MATERIAL statement. Default values for these parameters for various binary III-V materials are given in Table B-38. The compositionally dependent default values for the ternary combinations listed in Table B-38 are linearly interpolated from the binary values listed in the same table.

For nitride compounds, you can use a modified version of this model, which is described in Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.3.7: "GaN, InN, AlN, AlGaN, and InGaN System".

3.11: Carrier Transport in an Applied Magnetic Field

Magnetic field effects have been used in semiconductor characterization measurements and exploited in semiconductor device applications. ATLAS can model the effect of applied magnetic fields on the behavior of a semiconductor device.

The basic property, which is measured is the Hall coefficient R_H. For a uniformly doped device with a current flowing in the x-direction and a magnetic field in the z-direction, an electric field (the Hall field) is set up in the y-direction. The Hall coefficient is then obtained by combining these measured quantities in the ratio

$$R_H = \frac{E_y}{J_x B_z} \quad \text{3-509}$$

The Hall co-efficient also has a theoretical value

$$R_H = \frac{r(p - s^2 n)}{q(p + sn)^2} \quad \text{3-510}$$

where s is the electron mobility divided by hole mobility. The Hall scattering factor, r, can be obtained from the mean free time between carrier collisions and is typically not very different from unity.

Another quantity that is measured experimentally is the Hall Mobility

$$\mu^* = \left| R_H \sigma \right| \quad \text{3-511}$$

where σ is the electrical conductivity. We will consider the electron and hole currents separately. Therefore, taking the limits ($n >> p$) and then ($p >> n$) in Equation 3-511, we obtain the Hall mobilities for electrons and holes respectively as

$$\mu^*_n = r\mu_n$$
$$\mu^*_p = r\mu_p \quad \text{3-512}$$

where μ_n and μ_p are the carrier mobilities. The Hall scattering factor for electrons can be different to that for holes in general. Published values for silicon are 1.2 and 1.2 [6] and 1.1 and 0.7 [130] for electrons and holes respectively. The default in ATLAS is the latter. You can set the values on the MODELS statement using the RH.ELEC and RH.HOLE parameters.

The effect of the magnetic field on a carrier travelling with velocity v is to add a term called the Lorentz force

$$q(\underline{v} \mathbf{x} \underline{B}) \quad \text{3-513}$$

to the force it feels. The magnetic field density B is a vector (B_x, B_y, B_z) and is in units of Tesla (V . s/m^2) in ATLAS.

The consequence of this extra force in a semiconductor can be calculated by including it in a relaxation time based transport equation and making a low field approximation [6]. You can relate the current density vector obtained with a magnetic field applied to that obtained with no magnetic field using a linear equation

$$\underline{J}_B = \underset{\sim}{M} \underline{J}_0 \quad \text{3-514}$$

The Matrix M can be written

$$\underset{\sim}{M} = \frac{1}{1 + a^2 + b^2 + c^2}\begin{pmatrix} 1 + a^2 & ab - c & ca + b \\ c + ab & 1 + b^2 & bc - a \\ ca - b & a + bc & 1 + c^2 \end{pmatrix} \quad 3\text{-}515$$

where $a = \mu^* B_x$, $b = \mu^* B_y$, $c = \mu^* B_z$. The matrix takes this form for both electrons and holes. Using this form, you can include the magnetic field effects in ATLAS. The model is derived from an expansion in powers of Magnetic field and is only accurate if the weak field condition

$$\mu^* |B| \ll 1 \quad 3\text{-}516$$

is satisfied. Therefore, for example, a field of 1 Tesla in Silicon with a typical mobility of 0.1 $m^2/(\mathrm{V.s})$, the product is 0.1 so that the condition is satisfied.

To invoke this model in ATLAS2D, simply supply a value for BZ on the MODELS statement. Nonzero values of BX and BY are not permitted because they would generally result in current in the z-direction, thereby voiding the assumption of two-dimensionality. In ATLAS3D, any combination of BX, BY or BZ can be specified as long as condition (Equation 3-516) is satisfied.

Table 3-95 shows the parameters used for the carrier transport in a magnetic field.

Table 3-95. MODELS Statement Parameters

Parameter	Type	Default	Units
BX	Real	0.0	Tesla
BY	Real	0.0	Tesla
BZ	Real	0.0	Tesla
RH.ELEC	Real	1.1	
RH.HOLE	Real	0.7	

3.12: Anisotropic Relative Dielectric Permittivity

In the general case, the relationship between the electric displacement vector and the electric field is given by

$$\vec{D} = \varepsilon_0 \underset{\sim}{\varepsilon} \vec{E} \quad \text{3-517}$$

where ε, the relative dielectric permittivity, is a symmetric second order tensor represented by a 3×3 matrix. Since it is symmetric, it can be written as

$$\underset{\sim}{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yx} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{pmatrix} \quad \text{3-518}$$

In the isotropic case, the off-diagonal terms are zero and the diagonal terms all have the same value as each other. In the anisotropic case, you can always set the off-diagonal elements to zero by a suitable transformation of coordinates. This coordinate system may not match the coordinate system used by ATLAS for the simulation. Therefore, it is necessary to allow ATLAS to model anisotropic permittivity for the relationship Equation 3-518 where the off-diagonal elements are non-zero.

The discretization of the flux term in the Poisson equation

$$\varepsilon_0 \nabla \cdot \left(\underset{\sim}{\varepsilon} \vec{E} \right) \quad \text{3-519}$$

requires a more general treatment in the anisotropic case. For example, the x-component of the electric displacement vector will now contain a contribution from all the components of the electric field

$$D_x = \varepsilon_{xx} E_x + \varepsilon_{xy} E_y + \varepsilon_{xz} E_z \quad \text{3-520}$$

and so the discretization is more complicated and uses more CPU time. A simplification can be made when

- the permittivity matrix is diagonal.
- the mesh used in modelling the device is rectangular.

In this case, discretization will occur only in the directions of the coordinate axes and simply using a scalar permitivity for each direction is correct. This retains the simplicity and speed of the discretization for isotropic materials.

ATLAS allows you to specify a value of anisotropic relative dielectric permittivity using the PERM.ANISO parameter on the MATERIAL statement. By default, this is the value of (ε_{yy}) if the simulation is in 2D and (ε_{zz}) if it is in 3D. The off-diagonal elements are assumed to be zero. In the case of a 3D simulation, you can apply the value of PERM.ANISO to the y-direction instead of the z-direction by specifying YDIR.ANISO or ^ZDIR.ANISO on the MATERIAL statement. You can specify the value of (ε_{xx}) by using the PERMITTIVITY parameter on the MATERIAL statement. The simpler version of discretization will be used in this case by default.

If the coordinate system in which the permittivity tensor is diagonal and the ATLAS coordinate system are non-coincident, then the coordinate transformation can be specified as a set of vectors

```
X = (X.X, X.Y, X.Z)

Y = (Y.X, Y.Y, Y.Z)
```

```
Z = (Z.X, Z.Y, Z.Z)
```

where the vector components can be specified on the MATERIAL statement.

The default is that

```
X = (1.0, 0.0, 0.0)
```

```
Y = (0.0, 1.0, 0.0)
```

```
Z = (0.0, 0.0, 1.0)
```

and you should specify all necessary components. You do not need to normalize the vectors to unit length. ATLAS will normalize them and transform the relative dielectric tensor according to the usual rules. If you specify any component of X, Y or Z, then the more complete form of discretization will be used. You can also enable the more complete form of electric displacement vector discretization by using the E.FULL.ANISO parameter on the MATERIAL statement.

Table 3-96 shows the parameters used in the Anisotropic Relative Dielectric Permittivity.

Table 3-96. MATERIAL Statement Parameters

Parameter	Type	Default
PERM.ANISO	Real	-999.0
YDIR.ANISO	Logical	False
ZDIR.ANISO	Logical	True
E.FULL.ANISO	Logical	False
X.X	Real	1.0
X.Y	Real	0.0
X.Z	Real	0.0
Y.X	Real	0.0
Y.Y	Real	1.0
Y.Z	Real	0.0
Z.X	Real	0.0
Z.Y	Real	0.0
Z.Z	Real	1.0

3.13: Single Event Upset Simulation

The capability of single event upset/photogeneration transient simulation is included in 2D and 3D using the SINGLEEVENTUPSET statement (see Chapter 19: "Statements", Section 19.43: "SINGLEEVENTUPSET" for more information). It allows you to specify the radial, length, and time dependence of generated charge along tracks. There can be a single particle strike or multiple strikes. Each track is specified by an Entry Point Location (x0,y0,z0) and an Exit Point Location (x1,y1,z1). This is assumed to be a cylinder with the radius defined with the RADIUS parameter. In 2D, z0 and z1 should be neglected.

The entry and exit points are specified by the ENTRYPOINT and EXITPOINT parameters of the SINGLEEVENTUPSET statement. These are character parameters that represent the ordered triplet coordinates of the entry and exit points of the particle track.

The electron/hole pairs generated at any point is a function of the radial distance, r, from the center of the track to the point, the distance l along the track and the time, t. The implementation into ATLAS allows you to define the generation as the number of electron-hole pairs per cm^3 along the track according to the equation:

$$G(r, l, t) = (\texttt{DENSITY}*L1(l) + S*\texttt{B.DENSITY}*L2(l)*R(r)*T(t)) \quad \text{3-521}$$

where DENSITY and B.DENSITY are defined as the number of generated electron/hole pairs per cm^3.

In radiation studies, the ionizing particle is typically described by the linear charge deposition (LCD) value, which defines the actual charge deposited in units of pC/μm. You can use this definition within ATLAS by specifying the PCUNITS parameter in the SINGLEEVENTUPSET statement. If the user-defined parameter, PCUNITS, is set in the SINGLEEVENTUPSET statement, then B.DENSITY is the generated charge, in pC/μm, and the scaling factor S is:

$$S = \frac{1}{q\pi\,\texttt{RADIUS}^2} \quad \text{3-522}$$

where RADIUS is defined on the SINGLEEVENTUPSET statement. If the PCUNITS parameter isn't set, then B.DENSITY is the number of generated electron/hole pairs in cm^{-3} and the scaling parameter, S, is unity.

Another common measure of the loss of energy of the SEU particle, as it suffers collisions in a material, is the linear energy transfer (LET) value, which is given in units of MeV/mg/cm^2 (or MeV-cm^2/mg). In silicon, you can convert energy to charge by considering you need approximately 3.6 eV of energy to generate an electron-hole pair. The conversion factor from the LET value to the LCD value is then approximately 0.01 for silicon. So for instance, a LET value of 25 MeV-cm^2/mg is equivalent to 0.25 pC/μm, which can then be defined with the B.DENSITY and PCUNITS parameters.

The factors, *L1* and *L2*, in Equation 3-521 define the variation of charge or carrier generation along the path of the SEU track. These variations are defined by the equations:

$$L1(l) = \texttt{A1} + \texttt{A2}\cdot l + \texttt{A3}\,exp(\texttt{A4}\cdot l) \quad \text{3-523}$$

$$L2(l) = \texttt{B1}(\texttt{B2} + l\cdot\texttt{B3})^{\texttt{B4}} \quad \text{3-524}$$

where the A1, A2, A3, A4, B1, B2, B3 and B4 parameters are user-definable as shown in Table 3-97.

Table 3-97. User-Specifiable Parameters for Equations and 3-524			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	A1	1	
SINGLEEVENTUPSET	A2	0	cm^{-1}
SINGLEEVENTUPSET	A3	0	
SINGLEEVENTUPSET	A4	0	cm^{-1}
SINGLEEVENTUPSET	B1	1	
SINGLEEVENTUPSET	B2	1	
SINGLEEVENTUPSET	B3	0	cm^{-1}
SINGLEEVENTUPSET	B4	0	

Note: The default parameters in Table 3-97 were chosen to result in constant carrier or charge generation as a function of distance along the particle track.

The factor *R(r)* is the radial parameter, which is defined by one of two equations. The default is:

$$R(r) = exp\left(-\frac{r}{\mathrm{RADIUS}}\right) \quad \text{3-525}$$

where *r* is the radial distance from the center of the track to the point and RADIUS is a user-definable parameter as shown in Table 3-98. You can choose an alternative expression if you specify the RADIALGAUSS parameter on the SINGLEEVENTUPSET statement. In this case, *R(r)* is given by:

$$R(r) = exp-\left(\frac{r}{\mathrm{RADIUS}}\right)^2 \quad \text{3-526}$$

The time dependency of the charge generation *T(t)* is controlled with the TC parameter through two functions.

For TC=0:

$$T(t) = \mathrm{deltafunction(t-T0)} \quad \text{3-527}$$

For TC>0:

$$T(t) = \frac{2e-\left(\frac{t-\mathrm{TO}}{\mathrm{TC}}\right)^2}{\mathrm{TC}\sqrt{\pi}erfc\left(\frac{-\mathrm{TO}}{\mathrm{TC}}\right)} \quad \text{3-528}$$

where TO and TC are parameters of the SINGLEEVENTUPSET statement.

Table 3-98. User-Specifiable Parameters for Equations 3-521, 3-522, 3-527, and 3-528			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	DENSITY	0.0	cm^{-3}
SINGLEEVENTUPSET	B.DENSITY	0.0	cm^{-3} or pC/μm
SINGLEEVENTUPSET	TO	0.0	s
SINGLEEVENTUPSET	TC	0.0	s
SINGLEEVENTUPSET	RADIUS	0.05	μm

The following example shows a particle strike that is perpendicular to the surface along the z-plane. Therefore, only the z-coordinates change has a radius of 0.1 μm and a LET value of 20 MeV-cm^2/mg (B.DENSITY=0.2). The strike has a delay time of 60 ps and the Gaussian profile has a characteristic time of 10 ps. The generation track for this striking particle is from z=0 to z=10 μm and carrier generation occurs along its entire length.

```
single    entry="5.5,0.0,0.0" exit="5.5,0.0,10" radius=0.1 \
pcunits b.density=0.2 t0=60.e-12 tc=10.e-12
```

User-defined SEU

In addition to the model described by the SINGLEEVENTUPSET statement, you can use the C-INTERPRETER to specify an arbitrary generation profile. This is specified using the BEAM statement. In 2D, use the F.RADIATE parameter. In 3D, use the F3.RADIATE parameter.

```
BEAM NUM=1 F3.RADIATE=myseu.c
.
SOLVE B1=1.0 RAMPTIME=1e-12 DT=1e-14 TSTOP=1e-7
```

The F3.RADIATE parameter indicates an external C-language sub-routine conforming to the template supplied. The file, *myseu.c*, returns a time and position dependent value of carrier generation in 3-D. The value returned by *myseu.c*, is multiplied by the B1 parameter at each timestep. For more information about the C-Interpreter, see Appendix A: “C-Interpreter Functions”.

This page is intentionally left blank.

Chapter 4:
S-Pisces: Silicon Based 2D Simulator

4.1: Overview

S-PISCES is a powerful and accurate two-dimensional device modeling program that simulates the electrical characteristics of silicon based semiconductor devices including MOS, bipolar, SOI, EEPROM, and power device technologies.

S-PISCES calculates the internal distributions of physical parameters and predicts the electrical behavior of devices under either steady-state, transient, or small signal AC conditions. This is performed by solving Poisson's Equation (see Chapter 3: "Physics", Section 3.1.1: "Poisson's Equation") and the electron and hole carrier continuity equations in two dimensions (see Chapter 3: "Physics", Section 3.1.2: "Carrier Continuity Equations").

S-PISCES solves basic semiconductor equations on non-uniform triangular grids. The structure of the simulated device can be completely arbitrary. Doping profiles and the structure of the device may be obtained from analytical functions, experimentally measured data, or from process modeling programs SSUPREM3 and ATHENA.

For more information on semiconductor equations, see Chapter 3: "Physics", Section 3.1: "Basic Semiconductor Equations".

You should be familiar with ATLAS before reading this chapter any further. If not, read Chapter 2: "Getting Started with ATLAS".

4.2: Simulating Silicon Devices Using S-PISCES

4.2.1: Simulating MOS Technologies

Physical Models for MOSFETs

S-PISCES provides special physical models tuned for simulating MOS devices. Most of these models are accessed from the MODEL statement. The MOS parameter of the MODEL statement can be specified to turn on a default set of physical models that are most useful for MOS simulation. The MOS parameter enables Shockley-Read-Hall (SRH), Fermi Statistics (FERMI), and the Lombardi Mobility model (CVT) for transverse field dependence. To set the default MOS simulation models , use:

```
MODEL MOS PRINT
```

The transverse field dependent mobility models are of particular importance for simulating MOS devices. S-PISCES currently supports several different transverse field dependent mobility models. The CVT parameter selects the Lombardi CVT model. The YAMA parameter selects the Yamaguchi model. The TASCH parameter selects the Tasch model. The WATT parameter selects the Watt Surface Model, which can be operated in a more accurate mode with the extra parameter, MOD.WATT, on the MOBILITY statement.

You will find that the MOBILITY statement can be used to modify some of the parameters of the various models, to apply different models to different regions, or to apply different models to electrons and holes.

Meshing for MOS Devices

In device simulation of MOS devices, the key areas for a tight mesh are:

- very small vertical mesh spacing in the channel under the gate. The exact size of mesh required depends on the transverse field or surface mobility model chosen. See Figure 4-1 for an example of the effect of mesh size on drain current.
- lateral mesh spacing along the length of the channel for deep sub-micron devices. This is required to get the correct source-drain resistance and to resolve the channel pinch-off point.
- lateral mesh at the drain/channel junction for breakdown simulations. This is required to resolve the peak of electric field, carrier temperature and impact ionization.
- several vertical grid spacings inside the gate oxide when simulating gate field effects such as gate induced drain leakage (GIDL) or using any hot electron or tunneling gate current models

Figures 4-1 and 4-2 show the effect of mesh size in the MOS channel on IV curves. In Figure 4-1, the mesh density in the vertical direction is increased. As the mesh density increases, the resolution of the electric field and carrier concentration is improved. This example uses the CVT mobility model. Improvements in transverse electric field resolution lead to a reduced mobility in the channel and a stronger IV roll-off.

But Figure 4-2 shows the effect of surface channel mesh in MOSFETs is model dependent. This result shows the current at Vds=3.0V and Vgs=0.1V versus the size of the first grid division into the silicon. Results vary for each model but note that for all models, a very fine grid is required to reduce the grid dependence to acceptable levels.

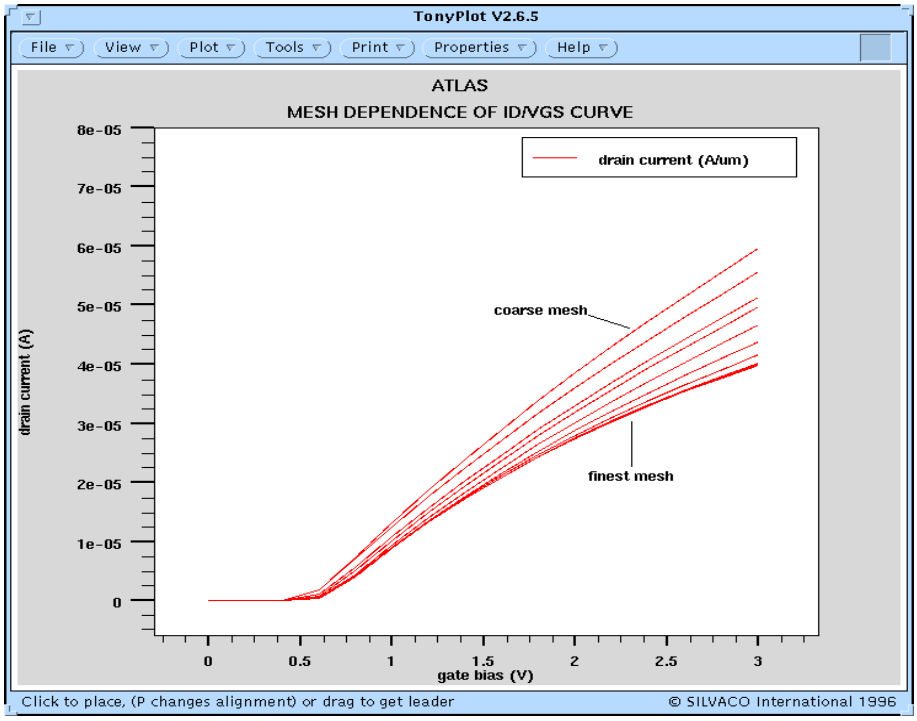


Figure 4-1: Effect on MOS IV curve of progressive refinement of the vertical mesh spacing at the surface of the MOS channel

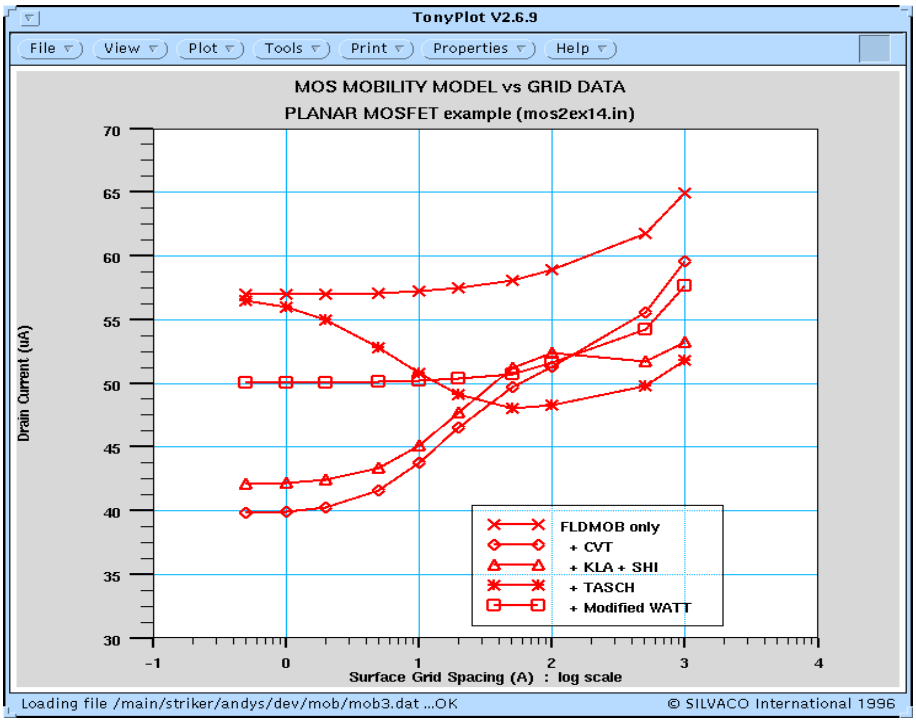


Figure 4-2: Effect of surface mesh spacing on simulated current for several MOS Mobility Models

MOS Electrode Naming

For MOS simulation, S-PISCES allows you to use standard electrode names to reduce confusion with the use of electrode indices. These names include: source, drain, gate, and substrate. Electrode names can be defined in ATHENA or DEVEDIT or in the ELECTRODE statement in ATLAS. These names can be used in the SOLVE statements for setting bias voltages such as:

```
SOLVE VGATE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=GATE
```

Gate Workfunction

In MOS simulations, the workfunction of the gate is an important parameter. This must be set in each input deck using the WORK parameter of the CONTACT statement. For example:

```
CONTACT NAME=GATE WORK=4.17
```

would set the workfunction on the gate at 4.17 eV.

Certain material names can also be used to set the workfunction of common gate materials. For example:

```
CONTACT NAME=GATE N.POLY
```

would set the workfunction on the gate to that of n type polysilicon.

Note: The gate workfunction should be set on a CONTACT statement even though the material or workfunction might be set from ATHENA or DEVEDIT.

Interface Charge

For accurate simulation of MOS devices, the interface charge at the oxide, specify semiconductor interface. To do this, set the QF parameters for the INTERFACE statement. Typically, a value of 3×10^{10} cm^{-2} is representative for the interface charge found in silicon MOS devices. The proper syntax for setting the value for this interface fixed charge is:

```
INTERFACE QF=3e10
```

You can also try to model the fixed charge more directly by using interface traps to simulate the surface states. To do this, use the INTTRAP statement. But this is rarely done in practice.

Single Carrier Solutions

Frequently for MOS simulation, you can choose to simulate only the majority carrier. This will significantly speed up simulations where minority carrier effects can be neglected. This can be done by turning off the minority carrier. To do this, use the ATLAS negation character, ^ , and one of the carrier parameters (ELECTRONS or HOLES) in the METHOD statement. For example, to simulate electrons only, you can specify one of the following:

```
METHOD CARRIERS=1 ELECTRONS
METHOD ^HOLES
```

Single carrier solutions should not be performed where impact ionization, any recombination mechanism, lumped element boundaries, or small signal AC analysis are involved.

Energy Balance Solutions

As MOS devices become smaller and smaller, non-local carrier heating effects becomes important. You can perform accurate simulation of non-local carrier heating effects using the energy balance model (EBM). As a general rule, you can use the gate length as a gauge to predict when non-local effects are important. Generally, for drain currents, energy balance should be applied for gate lengths less than 0.5 microns. For substrate currents, energy balance should be applied for gate lengths less than 1.0 micron.

To enable energy balance for electrons/holes, set either the HCTE.EL or HCTE.HO parameters on the MODEL statement. For example:

```
MODEL HCTE.EL
```

enables the energy balance model for electrons.

ESD Simulation

In some cases, lattice heating may be important to MOS simulation. This typically occurs in cases with very high currents, just like the case with ESD simulation. In these cases, GIGA should be used to simulate the heat-flow in the device. To enable heat flow simulation, set the LAT.TEMP parameter of the MODEL statement (a license for GIGA is required). For example, the statement:

```
MODEL LAT.TEMP
```

enables heat-flow simulation.

4.2.2: Simulating Silicon Bipolar Devices

Physical Models for BJTs

S-PISCES provides special physical models for bipolar device simulation. You can select these models using the MODEL statement. The BIPOLAR parameter of the MODEL statement enables a reasonable default set of bipolar models. These include: concentration dependent mobility (CONMOB), field dependent mobility (FLDMOB), bandgap narrowing (BGN), concentration-dependent lifetime (CONSRH) and Auger recombination (AUGER).

For the most accurate bipolar simulations, the recommended mobility model is the Klaassen model (KLA). This includes doping, temperature and carrier dependence. It applies separate mobility expressions for majority and minority carriers. You should also use this model with Klaassen's Auger model (KLAAUG) and Klaassen's concentration dependent SRH model (KLASRH). Combine the mobility model with FLDMOB to model velocity saturation. For surface (or lateral) bipolar devices, you can use the Shirahata model (SHI) to extend the Klaassen model with a transverse electric field dependence. The most accurate and appropriate model statement for bipolar devices is therefore:

```
MODELS KLA FLDMOB KLASRH KLAAUG BGN FERMI PRINT
```

You can choose this set of models by using the BIPOLAR2 parameter from the MODEL statement.

Note: For a complete syntax including description of models and method for simulating polysilicon emitter bipolar devices, see the BJT directory in the on-line examples.

Meshing Issues for BJTs

The most important areas to resolve in bipolar transistors are the emitter/base and base/collector junctions. Typically, a very fine mesh throughout the base region is required. The gain of the bipolar device is determined primarily by the recombination at the emitter/base junction or inside the emitter. Therefore, these regions need to be resolved with a fine mesh.

BJT Electrode Naming

S-PISCES also provides special electrode names for bipolar simulation that can be used to ease confusion over electrode indices. These electrode names include: "emitter", "base", "collector", "anode", and "cathode". Electrode names can be defined in ATHENA or DEVEDIT or in the ELECTRODE statement in ATLAS. The electrode names are used on SOLVE statement. For example:

```
SOLVE VBASE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=BASE
```

Dual Base BJTs

It is possible in S-PISCES to tie two or more contacts together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initially defined.

If the structure is defined using ATLAS syntax, it is possible to have multiple ELECTRODE statements with the same NAME parameter defining separate locations within the device structure. In this case, the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both ELECTRODE areas.

Similarly, if two separate metal regions in ATHENA are defined using the ATHENA ELECTRODE statement to have the same name, then in ATLAS these two electrodes will be considered as shorted together.

If the electrodes are defined with different names, the following syntax can be used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base
.
SOLVE VBASE=0.1
```

Here, the electrode, base1, will be linked to the electrode, base. Then, the applied 0.1V on base will also appear on base1. But ATLAS will calculate and store separate currents for both base and base1. This can be a useful feature. In some cases, however, such as where functions of the currents are required in EXTRACT or TONYPLOT, it is undesirable. You can add the SHORT parameter to the CONTACT statement above to specify that only a single base current will appear combining the currents from base and base1.

When loading a structure from ATHENA or DEVEDIT where two defined electrode regions are touching, ATLAS will automatically short these and use the first defined electrode name.

Creating an Open Circuit Electrode

It is often required to perform a simulation with an open circuit, such as for an open-base breakdown voltage simulation, on one of the defined electrodes. There are three different methods that will accomplish this. The first method is to delete an electrode from the structure file. The second method is to add an extremely large lumped resistance, for example $10^{20}\Omega$, onto the contact to be made open circuited. The third method is to first switch the boundary conditions on the contact to be made open circuited from voltage controlled to current controlled. Then, specify a very small current through that electrode.

Each of these methods are feasible. But if a floating region is created within the structure while using these methods, then numerical convergence may be affected. As a result, it is normally recommended to use the second method to ensure that no floating region is created.

Solution Techniques for BJTs

To obtain bipolar solutions, you almost always need to simulate using two carriers. This is due to the importance of minority carriers to device operation.

In certain cases, non-local carrier heating may be important to accurately simulate bipolar devices. In these cases, use the energy balance model. To model non-local carrier heating for electrons/holes, set the HCTE.EL and HCTE.HO parameters in the MODEL statement. For example, the statement:

```
MODEL HCTE.EL
```

invokes the carrier heating equation for electrons.

4.2.3: Simulating Non-Volatile Memory Technologies (EEPROMs, FLASH Memories)

To simulate non-volatile memory devices, you must become familiar with the basics of MOSFET simulation (see Section 4.2.1: “Simulating MOS Technologies”).

Defining Floating Gates

To simulate non-volatile memory technologies, such as EEPROMs or FLASH EEPROMs, specify one or more electrodes as floating gates. To do this, set the FLOATING parameter of the CONTACT statement. For example:

```
CONTACT NAME=fgate FLOATING
```

This specifies that the electrode named fgate is simulated as a floating gate. This means that the charge on the floating gate is calculated as the integral of the gate current at that gate during a transient simulation.

Modeling the correct coupling capacitance ratio between the floating gate and control gate often requires adding an extra lumped capacitor from the floating gate to the control gate or one of the other device terminals. This is often required since S-PISCES is performing a 2-D simulation whereas the coupling of the gates is often determined by their 3-D geometry. Parameters on the CONTACT statement are used to apply these extra lumped capacitances. For example, to add a capacitor of 1fF/mm between the control and floating gates the syntax is:

```
CONTACT NAME=fgate FLOATING FG1.CAP=1.0e-15 EL1.CAP=cgate
```

Gate Current Models

You can supply the gate currents for the floating gate structure by one of three sources: hot electron injection (HEI or N.CONCAN), hot hole injection (HHI or P.CONCAN) and Fowler-Nordheim tunneling current (FNORD).

These currents are of importance, depending on whether electrons are being moved onto the gate or off the floating gate. In the case of placing electrons on the floating gate, use hot electron injection and Fowler-Nordheim tunneling. In the case of removing electrons from the floating gate, set hot hole injection and Fowler-Nordheim tunneling.

In drift diffusion simulations, perform hot electron injection by setting the HEI parameter of the MODELS statement. To simulate hot hole injection, use the HHI parameter of the MODELS statement. To enable Fowler-Nordheim tunneling, set the FNORD parameter of the MODELS statement. The following example demonstrated the proper setting for Flash EPROM programming.

```
MODELS MOS HEI PRINT
```

This next example is appropriate for EEPROM erasure.

```
MODELS MOS HHI FNORD PRINT
```

With energy balance simulations, use the Concannon Models for EPROM programming and erasing. For more information about these models, see Chapter 3: “Physics”, the “Concannon’s Injection Model” section on 3-111.

Note: Writing and erasure of floating gate devices should be done using transient simulation.

Gate Current Assignment (NEARFLG)

The actual calculation of floating gate current magnitude is done at the silicon-oxide interface. The question of distribution of oxide currents to the various electrodes near the interface is resolved using one of two models. The actual flow of carriers in oxides is not well known. Accepted physical models of carrier flow in oxides are still under research. As such, S-PISCES provides two heuristic models to choose from. The default is to distribute currents calculated at points along the interface to the electrode in the direction of highest contributing field. This model is somewhat analogous to a purely drift model of oxide carrier transport. The alternative is to set the NEARFLG parameter of the MODEL statement. In this case, the currents calculated at points along the interface are distributed to the geometrically closest electrode. This model is analogous to a purely diffusion model of carrier transport in oxide.

4.2.4: Simulating SOI Technologies

Silicon substrates are now being produced that contain an oxide layer buried below the surface of the silicon at some predefined depth. The existence of this buried oxide layer has resulted in a change not only in the fabrication process used to manufacture a device in the surface silicon but also in the challenges facing device simulation.

All of the issues raised previously about MOS device simulation should be considered with some extra SOI specific problems.

The most common device technology that uses these SOI substrates is the SOI MOSFET. This section summarizes the simulation requirements for SOI using this particular technology as a reference.

Meshing in SOI devices

The mesh requirements for SOI MOSFETs is very similar to that described in the previous section for bulk MOS transistors. In addition to these requirements, there are some additional points to meshing these devices. These are:

- Two channel regions may exist: one underneath the top (front) gate oxide and the other above the buried (back gate) oxide.
- Inside the buried oxide layer, the mesh constraints can be relaxed considerably compared with the top gate oxide.
- The active silicon underneath the top gate can act as the base region of a bipolar transistor and as such may require a finer mesh when compared to bulk MOS transistors.

Physical Models for SOI

SOI MOSFET simulations are based upon the physical operation of the device, which exhibits both MOS and bipolar phenomena. As a result a more complex set of physical models will be required than for either MOS or bipolar technologies. Table 4-1 shows these models.

Table 4-1: SOI Physical Models

Model	Description
Mobility	Klaassens Model (KLA) is recommended to account for lattice scattering, impurity scattering, carrier-carrier scattering and impurity clustering effects at high concentration. The Shirahata Mobility Model (SHI) is needed to take into account surface scattering effects at the silicon/oxide interface, which is a function of the transverse electric field. High electric field velocity saturation is modelled through the field dependent mobility model (FLDMOB). You can tune model parameters using the MOBILTY statement syntax.
Interface Charge	In SOI transistors, there exist two active silicon to oxide interfaces on the wafer. The top interface, under the top gate, is similar to MOS technology. The bottom interface is quite different and typically contains significantly more charge. You can set different interface charges in SPISCES using the INTERFACE statement with region specific parameters.
Recombination	To take account of recombination effects, we recommend the use of the Shockley-Read-Hall (SRH) model. This simulates the leakage currents that exist due to thermal generation. You may also need to simulate the presence of interface traps at the silicon/oxide interface. Then, turn on the Direct Recombination Model (AUGER). The parameters for both models can be tuned in the MOBILITY statement.
Bandgap Narrowing	This model (BGN) is necessary to correctly model the bipolar current gain when the SOI MOSFET behaves like a bipolar transistor. Specify the parameters for this model in the MODELS statement.

Table 4-1: SOI Physical Models	
Model	**Description**
Carrier Generation	Impact ionization significantly modifies the operation of SOI MOSFETs. To account for this phenomena, switch on the impact ionization model (IMPACT) and calibrate for SOI technology. The calibration parameters are set in the IMPACT statement.
Lattice Heating	When you switch a device on, there can be significant current density within the silicon. This could generate a significant amount of heat. In bulk MOS devices, the silicon substrates behaves like a good heat conductor. This generated heat is then quickly removed. But this isn't the case with SOI substrates as the buried oxide layer allows this generated heat to be retained. For SOI MOSFETs, this can be a significant amount and can drastically affect the operation of the device. In such cases, take account of this by using the GIGA module. Note that when you switch on lattice heating by using the LAT.TEMP parameter in the MODELS statement, you also need to specify a thermal boundary condition with the THERMCONTACT statement. See Chapter 7: "Giga: Self-Heating Simulator" for more details.

Carrier Heating	In deep submicron designs, you may need to switch on the additional energy balance equations. These take into account the exchange of energy between carriers and between the carriers and the lattice. See Section 4.2.1: "Simulating MOS Technologies" for more information.
	
	An example of a set of typical models for a partially depleted SOI MOSFET could be:
	
	`MODEL KLA SHI FLDMOB SRH AUGER BGN LAT.TEMP`
	`INTERFACE QF=1e10 Y.MAX=0.05`
	`INTERFACE QF=1e11 Y.MIN=0.05`
	`THERMCONTACT NUM=1 ELEC.NUM=4 EXT.TEMP=300`
	`IMPACT SELB`

Numerical Methods for SOI

One important issue with SOI device simulation is the choice of numerical methods. In SOI technology, the potential in the channel (or body) region is commonly referred to as "floating". This occurs because there exists no direct contact to it by any electrode. As a result, when a bias is applied, or increased, on a contact there may be some convergence problem. This occurs because the guess used in the numerical solution scheme for (ψ, n, p) may be poor, particularly in the "floating" region. This is particularly true if impact ionization is used. To overcome the problem of poor initial guess, use the following numerical methods syntax in isothermal drift-diffusion simulations.

```
METHOD GUMMEL NEWTON
```

This method initially performs a GUMMEL iteration to obtain an improved initial guess for the NEWTON solution scheme. Although this method is more robust, this is slower than using the NEWTON scheme alone. For more information on the numerical schemes available, see Chapter 18: "Numerical Techniques".

SOI Physical Phenomena

The physical models and the numerical schemes described above should allow S-PISCES to study all the important SOI phenomena. These include:

- full or partial depletion effects
- threshold voltage
- subthreshold slopes
- front to back gate coupling effects
- leakage current analysis
- high frequency analysis
- device breakdown
- snapback effects
- the kink effect in the output Ids-Vds characteristics
- negative differential resistance

This page is intentionally left blank.

Chapter 5:
Blaze: Compound Material 2D Simulator

5.1: Overview

Before continuing to the sections that follow, become familiar with ATLAS. If not, read Chapter 2: "Getting Started with ATLAS" before proceeding with this chapter.

BLAZE is a general purpose 2-D device simulator for III-V, II-VI materials, and devices with position dependent band structure (i.e., heterojunctions). BLAZE accounts for the effects of positionally dependent band structure by modifications to the charge transport equations. BLAZE is applicable to a broad range of devices including: HBTs, HEMTs, LEDs, heterojunction photodetectors (APDs, solar cells, and so on) and heterojunction diodes.

This chapter is composed of several sections. Section 5.1.1: "Basic Heterojunction Definitions" diagrams the basic heterojunction band parameters and includes a section on heterojunction alignment. Heterojunction charge transport is covered in Section 5.1.3: "The Drift Diffusion Transport Model". This section includes the details of how BLAZE modifies the basic transport models to simulate heterodevices. Section 5.2: "The Physical Models" covers the physical models unique to BLAZE. Detailed information about the material systems encountered in heterojunction simulation is covered in Section 5.3: "Material Dependent Physical Models". This includes the relationships between the compound elemental concentrations and bandgap, dielectric constant, low field mobility, and other important material and transport parameters. Finally, Section 5.4: "Simulating Heterojunction Devices with Blaze" covers how to define materials and models for heterojunction devices with BLAZE.

Appendix B: "Material Systems" has the defaults for these parameters.

5.1.1: Basic Heterojunction Definitions

Figure 5-1 shows the band diagrams and band parameters for a basic p-n heterojunction device under equilibrium conditions. This diagram illustrates two materials with different bandgaps and electron affinities and a Schottky barrier in contact to the n-type material.

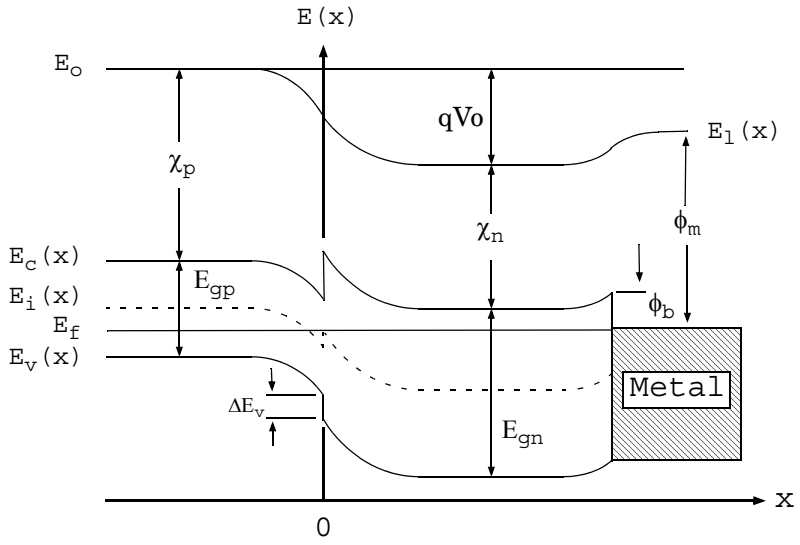


Figure 5-1: Band Diagram of p-n Heterojunction

Referring to Figure 5-1:

- $E_c(x)$, $E_v(x)$ and $E_i(x)$ are the spatially dependent conduction band, valence band and intrinsic energy levels respectively.
- E_f and E_o are the Fermi and Vacuum level energies.
- E_{gp} and E_{gn} are the p-type and n-type material bandgaps.
- ΔE_v is the portion of the difference between E_{gp} and E_{gn} that appears between the valence bands at the heterojunction interface.
- ΔE_c (not labelled) is the portion of the difference between E_{gp} and E_{gn} that appears between the conduction bands at the heterojunction interface.
- qVo is the built-in potential of the heterojunction.
- χ_p and χ_n are the electron affinities of the p-type and n-type materials.
- ϕ_m is the work function of the metal.
- ϕ_b is the barrier height between the metal and the semiconductor.

The basic band parameters for defining heterojunctions in BLAZE are bandgap parameter, EG300, electron parameter, AFFINITY, and the conduction and valence band density of states, NC300 and NV300. To define these parameters for each material, use the MATERIAL statement. Other transport parameters relating these basic definitions to compound elemental concentrations (x.COMPOSITION and Y.COMPOSITION) can also be defined.

See the MATERIALS section of this chapter for a description of these relationships for each material system and the Section 5.4: "Simulating Heterojunction Devices with Blaze" for their usage. The work function of metals is defined using the CONTACT statement.

5.1.2: Alignment

As shown from Figure 5-1, the difference in the two material bandgaps creates conduction and valence band discontinuities. How the bandgap difference is distributed between the conduction and valence bands has a large impact on the charge transport in these heterodevices. There are two methods for defining the conduction band alignment for a heterointerface The first method is the Affinity Rule, which uses the ALIGN parameter on the MATERIAL statement. The second method is to adjust the material affinities using the AFFINITY parameter on the MATERIAL statement.

The Affinity Rule

This is the default method in BLAZE for assigning how much of the bandgap difference appears as the conduction band discontinuity. The affinity rule assigns the conduction band discontinuity equal to the difference between the two materials electron affinities (AFFINITY on the MATERIAL statement). The affinity rule method is used by default for all materials where the ALIGN parameter has not been defined on the MATERIAL statement.

Using the ALIGN parameter in the MATERIAL statement

Experimental measurements of the actual band discontinuities can differ from what is assigned using the affinity rule with the standard material electron affinities. Therefore, BLAZE allows ΔE_c to be calculated by specifying the ALIGN parameter on the MATERIAL statement. ALIGN specifies the fraction of the bandgap difference, which will appear as the conduction band discontinuity. This bandgap difference is between the material, which is specified by the ALIGN parameter, and the smallest bandgap material in the overall structure (the reference material). Internally, BLAZE creates the desired conduction band offset by modifying the electron affinity of the material, which is specified by the ALIGN parameter.

In many applications, the Schottky barriers or more than two different semiconductor materials are present. Keep the reference material bandgap and these assigned affinities in mind when defining offsets for multiple materials or Schottky barrier heights. Examples for multiple materials and Schottky barriers are given in the examples section.

Manually Adjusting Material Affinity

You can use the AFFINITY parameter on the MATERIAL statement in conjunction with the default affinity rule alignment method to adjust the conduction band offset. In this case, the electron affinity of the larger bandgap material is adjusted so that the difference between the two materials affinity equals the desired conduction band offset. When there are more than two different materials, you can adjust each material affinity in this manner. This is the easiest method for handling multiple materials and heterojunctions.

The following examples describe the procedure for aligning heterojunctions using these two methods in BLAZE.

EXAMPLE 1

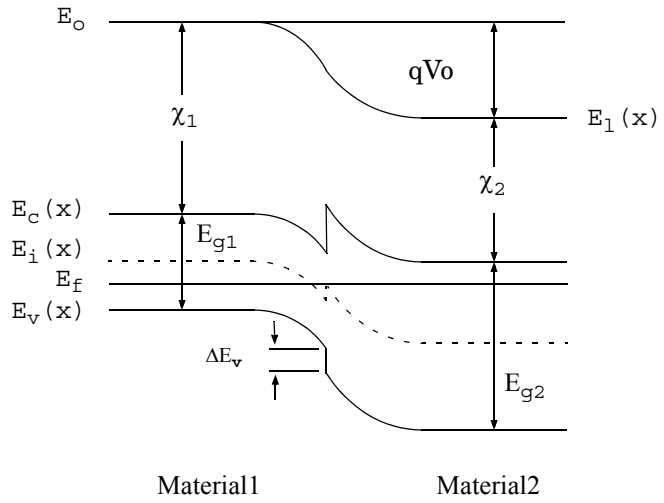


Figure 5-2: Band diagram of heterojunction with band offset.

Figure 5-2 shows a heterojunction consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities χ_1 and χ_2. This example is similar to the bandstructure of a HFET or HEMT. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_c = \chi_1 - \chi_2 \qquad 5\text{-}1$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \qquad 5\text{-}2$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface. ΔE_v is the amount of the valence band discontinuity.

Note: The Affinity Rule is used to calculate the conduction band offset for a material if the ALIGN parameter is not specified on the MATERIAL statement for that material.

Using the ALIGN parameter on the MATERIAL statement

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset. Then, define the ALIGN parameter on the MATERIAL statement for Material2 using

```
MATERIAL NAME=Material2 ALIGN=0.80
```

Then:

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 \quad \text{5-3}$$

Internally, the affinity of Material2 is adjusted so that ΔE_c equals this value. This value of electron affinity will override any electron affinity specification for Material2. This has an impact on any calculation where this material's electron affinity is used and considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

```
MATERIAL NAME=Material2 AFFINITY=VALUE
```

where VALUE is adjusted to provide for the desired conduction band offset as calculated by the affinity rule, that is, relative to Material1. This value of electron affinity will override any electron affinity specification for Material2. This has an impact on any calculation where this material's electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations

Note: If you do not specify the ALIGN parameter on the MATERIAL statement, BLAZE will use the Affinity Rule and either the default electron affinity or the affinity assigned using the AFFINITY parameter on the MATERIAL statement to calculate the conduction band offsets.

EXAMPLE 2

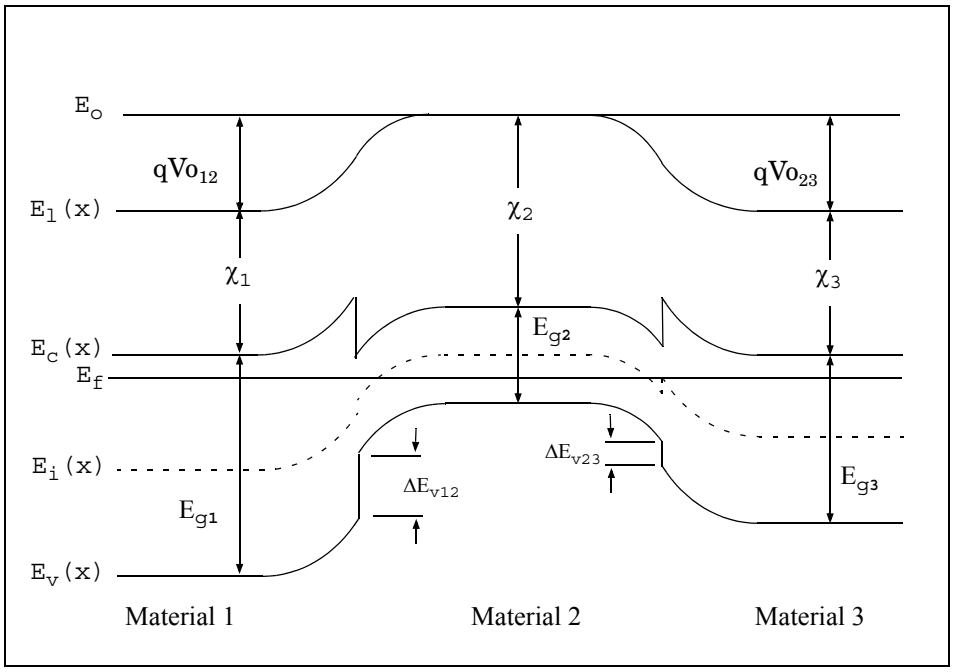


Figure 5-3: Band diagram of three material system (lowest E_g in center)

Figure 5-3 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1}, E_{g2} and E_{g3} and electron affinities χ_1, χ_2 and χ_3. This is similar to the band diagram of a Double Heterojunction Bipolar Transistor. For this example, $E_{g1} > E_{g2} < E_{g3}$ and $\chi_1 < \chi_2 > \chi_3$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_{c12} = \chi_2 - \chi_1 \quad \text{5-4}$$

and:

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \quad \text{5-5}$$

for the heterojunction between `Material1` and `Material2` and:

$$\Delta E_{c23} = \chi_2 - \chi_3 \quad \text{5-6}$$

and:

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \quad \text{5-7}$$

for the heterojunction between `Material2` and `Material3` using the `ALIGN` parameter on the `MATERIAL` statement:

Notice the reference material and the material with the smallest bandgap in this case, `Material2`, is located between the two larger bandgap materials, `Material1`, and `Material3`.

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset for this heterojunction. Then, define the ALIGN parameter on the MATERIAL statement for Material 1 using:

```
MATERIAL NAME=Material1 ALIGN=0.8
```

then:

$$\Delta E_{c12} = (E_{g1} - E_{g2}) \cdot 0.80 \quad \text{5-8}$$

Internally, the affinity of Material1 is adjusted so that ΔE_{c12} equals this value.

Let's assign 70% of the bandgap difference between Material3 and Material2 to the conduction band offset for this heterojunction. Defining the ALIGN parameter on the MATERIAL statement for Material3 using:

```
MATERIAL NAME=Material3 ALIGN=0.70
```

then:

$$\Delta E_{c23} = (E_{g3} - E_{g2}) \cdot 0.70 \quad \text{5-9}$$

Internally, the affinity of Material3 is adjusted so that ΔE_{c23} equals this value.

These new values of electron affinity for Material1 and Material3 will override any electron affinity specification for these materials. This has an impact on any calculation where these materials's electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

Assigning the conduction band offsets for each heterojunction is accomplished by setting the electron affinities for Material1 and Material3 using the AFFINITY parameter on the MATERIAL statement. The electron affinity for Material1 and Material3 are adjusted relative to Material2 by the amount of the desired conduction band offset for each heterojunction. Since Material2 affinity is larger than that for Material1 and Material3, the affinities for Material1 and Material3 are reduced (relative to Material2) to provide the desired conduction band offset.

Let's assume an electron affinity for Material2 of 4eV (~ that of GaAs). Let's decide that between Material1 and Material2, the conduction band offset is 0.3eV and that between Material3 and Material2, the conduction band offset is 0.2eV. Then, for Material1:

```
MATERIAL NAME=Material1 AFFINITY=3.7
```

and for Material3:

```
MATERIAL NAME=Material3 AFFINITY=3.8
```

This is the easiest method to define the conduction band offsets for multiple materials. This value of electron affinity will override any electron affinity specification. This has an impact on any calculation where this material's electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 3

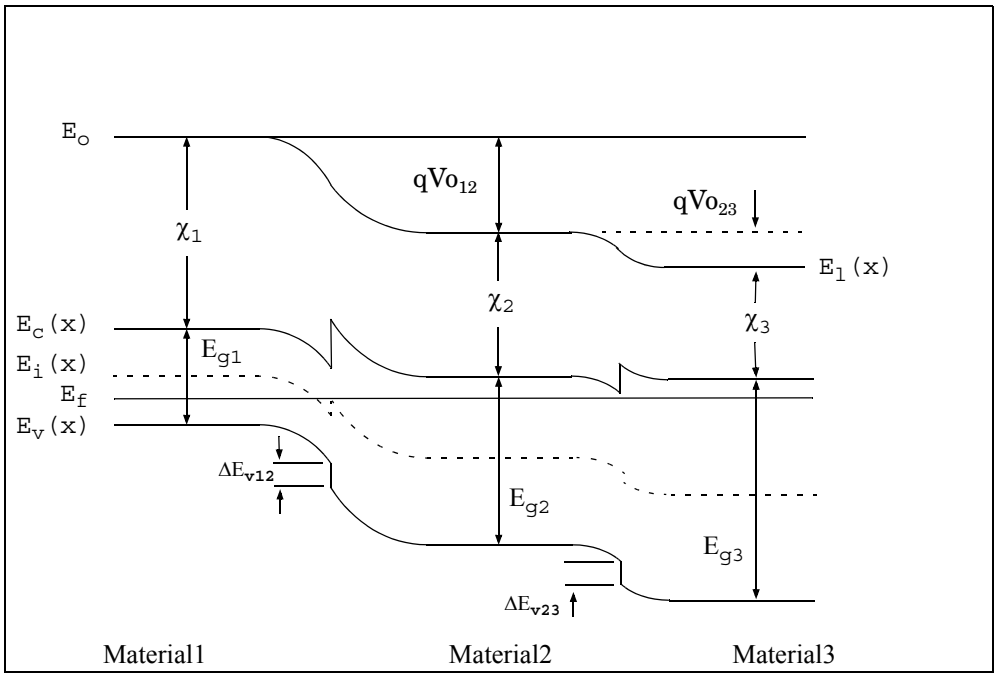


Figure 5-4: Band diagram of three material system (lowest E_g not in center)

Figure 5-4 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1}, E_{g2} and E_{g3} and electron affinities χ_1, χ_2 and χ_3. This is similar to the "EXAMPLE 2" on page 5-5, except that the narrow bandgap material is not located in between the other larger bandgap materials. This will add extra complexity to the conduction and valence band offset calculations. For this example, $E_{g1} < E_{g2} < E_{g3}$ and $\chi_3 < \chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_{c12} = \chi_1 - \chi_2 \quad \text{5-10}$$

and

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \quad \text{5-11}$$

for the heterojunction between `Material1` and `Material2` and:

$$\Delta E_{c23} = \chi_2 - \chi_3 \quad \text{5-12}$$

and

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \quad \text{5-13}$$

for the heterojunction between `Material2` and `Material3`.

Using the ALIGN parameter on the MATERIAL statement

Notice that the reference material and the material with the smallest bandgap in this case, Material1, is not shared between the two larger bandgap materials, Material2, and Material3. This will be important in calculating the conduction band offsets for the heterojunction formed by Material2 and Material3 (the one where the reference material is not present).

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset for this heterojunction. Since the reference material is one of the materials of this heterojunction, we can proceed as before. Define the ALIGN parameter in the MATERIAL statement for Material2 using:

```
MATERIAL NAME=Material2 ALIGN=0.8
```

then

$$\Delta E_{c12} = (E_{g2} - E_{g1}) \cdot 0.80 \quad \text{5-14}$$

Internally, the affinity of Material2 is adjusted so that ΔE_{c12} equals this value.

Let's assign 70% of the bandgap difference between Material3 and Material2 to the conduction band offset for this heterojunction. Since the reference material is not one of the materials in this heterojunction, another procedure will be used. Since BLAZE always uses the bandgap of the reference material (the smallest bandgap material in overall structure) when calculating the conduction band offset using the ALIGN parameter on the MATERIAL statement, the actual value for the ALIGN parameter needs to be calculated as follows:

$$\texttt{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot \texttt{FRACTION} \quad \text{5-15}$$

where FRACTION is the desired fraction of the bandgap difference between Material3 and Material2 that will appear in the conduction band. ΔE_{g23} is the bandgap difference for the actual heterojunction and ΔE_{g31} is the bandgap difference using the reference material. Once calculated, you can use this value for the ALIGN parameter on the MATERIAL statement for Material3. FRACTION and ALIGN will only be equal when the reference material is one of the two materials in the heterojunction. For this example, let's assume that

$$\Delta E_{g32} = 0.2 \quad \text{5-16}$$

and

$$\Delta E_{g31} = 0.4 \quad \text{5-17}$$

then for a desired conduction band offset fraction of 0.70:

$$\texttt{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot \texttt{FRACTION} = (0.2 / 0.4) \times 0.70 = 0.35 \quad \text{5-18}$$

You can assign the proper value of the ALIGN parameter reflecting the desired conduction band offset as:

```
MATERIAL NAME=Material3 ALIGN=0.35
```

This assigns 70% of the bandgap difference between Material3 and Material2 as the conduction band offset. Internally, the affinity of Material3 is adjusted so that ΔE_{c23} equals this value.

Note: Calculating ALIGN in this manner is only necessary when the reference material is not in contact with the material where the ALIGN parameter will be specified.

These new values of electron affinity for Material2 (from the first heterojunction band offset calculation) and Material3 (from the second heterojunction band offset calculation) will override any electron affinity specification for these materials. This has an impact on any calculation where these materials's electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

Assigning the conduction band offsets for each heterojunction is accomplished by setting the electron affinities for Material2 and Material3 using the AFFINITY parameter on the MATERIAL statement. The electron affinity for Material2 is adjusted relative to Material1 and Material3 is adjusted relative to Material2 by the amount of the desired conduction band offset for each heterojunction. Since Material1 affinity is larger than that for Material2 and Material2 affinity is larger than that for Material3, the affinities for Material2 and Material3 are reduced to provide the desired conduction band offsets.

Let's assume an electron affinity for Material1 of 4eV (~ that of GaAs). Let's decide that between Material1 and Material2, the conduction band offset is 0.3eV and that between Material2 and Material 3, the conduction band offset is 0.2eV. Then, for Material2:

```
MATERIAL NAME=Material2 AFFINITY=3.7
```

and for Material3:

```
MATERIAL NAME=Material3 AFFINITY=3.5
```

This is the easiest method to define the conduction band offsets for multiple materials. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See "EXAMPLE 4" on page 5-10 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 4

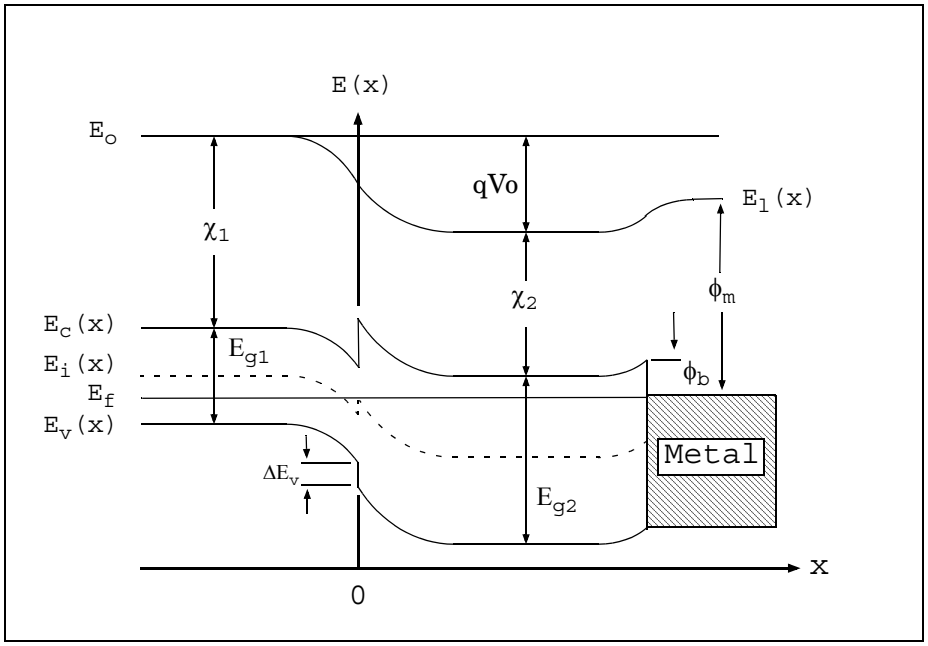


Figure 5-5: Schematic band diagram for an abrupt heterojunction

Figure 5-5 details a heterostructure device consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities $\chi_1$and χ_2 and a Schottky barrier. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

This example will first define the heterojunction band offsets and then the Schottky barrier height. Schottky contact barrier heights are calculated by BLAZE using the metal work function and the semiconductor electron affinity as:

$$\phi_b = \phi_m - \chi_s \quad \text{5-19}$$

where ϕ_b is the Schottky barrier height, ϕ_m is the work function of the metal, and χ_s is the semiconductor electron affinity. ϕ_m is set using the WORKFUN parameter in the CONTACT statement. Therefore, the semiconductor electron affinity as modified or defined during the heterojunction alignment process plays an important role in determining the value of the metal workfunction needed to provide the desired barrier height. Let's assume for this example that a Schottky barrier height of 0.2eV is desired and calculate the appropriate metal workfunction for each case.

Using the Affinity rule for the heterojunction

$$\Delta E_c = \chi_1 - \chi_2 \quad \text{5-20}$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \quad \text{5-21}$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface. ΔE_v is the amount of the valence band discontinuity.

Note: The Affinity Rule is used for a material if the ALIGN parameter is not specified on the MATERIAL statement for that material.

Let's use an electron affinity for Material1 of 4eV and for Material2 of 3.5eV. Since the affinity of the material on which the Schottky barrier is formed was not modified with this method of alignment, the metal work function needed to provide for a Schottky barrier height of 0.2eV is:

$$\phi_m = 3.5 + 0.2 = 3.7 \quad \text{5-22}$$

You can now assign this value to the WORKFUN parameter on the CONTACT statement as:

```
CONTACT NUMBER=1 WORKFUN=3.7
```

This produces a Schottky barrier height of 0.2eV between the metal and Material 2 using the ALIGN parameter on the MATERIAL statement:

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset, an electron affinity for Material1 of 4eV, and ΔE_g of 0.2eV. Then, define the ALIGN parameter on the MATERIAL statement for Material2 using:

```
MATERIAL NAME=Material2 ALIGN=0.80
```

Then:

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 = 0.2 \cdot 0.80 = 0.16 \quad \text{5-23}$$

Internally, the affinity of Material2 is reduced by 0.16eV so:

$$\chi_2 = \chi_1 - \Delta E_c = (4 - 0.16) = 3.84 \quad \text{5-24}$$

You can use this value of electron affinity to assign the proper value of WORKFUN on the CONTACT statement to provide for a Schottky barrier height of 0.2eV.

$$\phi_m = 3.84 + 0.2 = 4.04 \quad \text{5-25}$$

You can now assign this value to the WORKFUN parameter on the CONTACT statement as:

```
CONTACT NUMBER=1 WORKFUN=4.04
```

producing a Schottky barrier height of 0.2eV between the metal and Material2.

Manually Adjusting Material Affinity

Let's assign a conduction band offset between Material1 and Material2 of 0.15eV, an electron affinity for Material1 of 4eV, and the desired Schottky barrier height of 0.2eV. The affinity for Material2 is calculated from the affinity of Material1 and the desired conduction band offset as:

$$\chi_2 = \chi_1 - 0.15 = 4 - 0.15 = 3.85 \quad \text{5-26}$$

This is then used to assign the value of AFFINITY using:

```
MATERIAL NAME=Material2 AFFINITY=3.85
```

You can now use this value of electron affinity to calculate the metal workfunction necessary to produce a Schottky barrier height of 0.2eV as:

$$\phi_m = 3.85 + 0.2 = 4.05 \quad \text{5-27}$$

You can now assign this value to the WORKFUN parameter on the CONTACT statement as:

```
CONTACT NUMBER=1 WORKFUN=4.05
```

This produces a Schottky barrier height of 0.2eV between the metal and Material2.

5.1.3: The Drift Diffusion Transport Model

Drift-Diffusion with Position Dependent Band Structure

The current continuity equations for electrons and holes, and the Poisson Equation (see Chapter 3: "Physics", Equation 3-1) are the same as for the homogeneous case. Although the changing dielectric constant is taken into account. The current density expressions, however, must be modified to take into account the nonuniform band structure [101]. This procedure starts with the current density expressions:

$$\vec{J}n = -\mu_n n \nabla\phi_n \quad \text{5-28}$$

$$\vec{J}p = -\mu_p n \nabla\phi_p \quad \text{5-29}$$

where ϕ_n and ϕ_p are quasi-Fermi potentials.

$$\phi_n = \frac{1}{q}E_{FN} \quad \text{5-30}$$

$$\phi_p = \frac{1}{q}E_{FP} \quad \text{5-31}$$

The conduction and valence band edge energies can be written as:

$$E_c = q(\psi_0 - \psi) - \chi \quad \text{5-32}$$

$$E_v = q(\psi_0 - \psi) - \chi - E_g \quad \text{5-33}$$

where:

- ψ_0 is some reference potential.
- χ is the position-dependent electron affinity.
- E_g is the position-dependent bandgap.
- ψ_0 can be selected in the form:

$$\psi_0 = \frac{\chi_r}{q} + \frac{kT_L}{q}ln\frac{N_{cr}}{n_{ir}} = \frac{\chi_r + E_g}{q} - \frac{kT_L}{q}ln\frac{N_{vr}}{n_{ir}} \quad \text{5-34}$$

where n_{ir} is the intrinsic carrier concentration of the arbitrarily selected reference material, and r is the index that indicates all of the parameters are taken from reference material.

Fermi energies are expressed in the form:

$$E_{FN} = E_c + kT_L ln\frac{n}{N_c} - kT_L ln\gamma n \quad 5\text{-}35$$

$$E_{FP} = E_v + kT_L ln\frac{n}{N_v} - kT_L ln\gamma n \quad 5\text{-}36$$

The final terms in Equations 5-35 and 5-36 are due to the influence of Fermi-Dirac statistics. These final terms are defined as follows:

$$\gamma_n = \frac{F1/2(\eta_n)}{e^{\eta_n}},\ \eta_n = \frac{E_{FN} - E_c}{kT_L} = F^{-1}_{1/2}\left(\frac{n}{N_c}\right) \quad 5\text{-}37$$

$$\gamma_p = \frac{F1/2(\eta_p)}{e^{\eta_p}},\ \eta_p = \frac{E_v - E_{FP}}{kT_L} = F^{-1}_{1/2}\left(\frac{p}{N_v}\right) \quad 5\text{-}38$$

where N_c and N_v are position-dependent and γ_n=γ_p= *1* for Boltzmann statistics.

By combining Equations 5-28 to 5-38, you can obtain the following expressions for current densities.

$$\vec{J}n = kT_L\mu_n \nabla n - q\mu_n n \nabla\left(\psi + \frac{kT_L}{q} ln\gamma_n + \frac{\chi}{q} + \frac{kT_L}{q} ln\frac{N_c}{n_{ir}}\right) \quad 5\text{-}39$$

$$\vec{J}p = kT_L\mu_p \nabla n - q\mu_p p \nabla\left(\psi + \frac{kT_L}{q} ln\gamma_p + \frac{\chi + E_g}{q} + \frac{kT_L}{q} ln\frac{N_v}{n_{ir}}\right) \quad 5\text{-}40$$

5.1.4: The Thermionic Emission and Field Emission Transport Model

You can activate alternative current density expressions for electron and hole current [183,188], which take into account thermionic emission dominated current in abrupt heterojunctions. These equation applies only at the node points along the interface of the heterojunction and take the form:

$$\overrightarrow{J_n} = q\upsilon_n(1+\delta)\left(n^+ - n^- exp\left(\frac{-\Delta E_C}{kT_L}\right)\right) \quad 5\text{-}41$$

$$\overrightarrow{J_p} = (-q)\upsilon_p(1+\delta)\left(p^+ - p^- exp\left(\frac{-\Delta E_V}{kT_L}\right)\right) \quad 5\text{-}42$$

where J_n and J_p are the electron and hole current densities from the "-" region to the "+" region. ν_n and ν_p are the electron and hole thermal velocities. ΔE_c is conduction band energy change going from the “-” region to the “+” region. ΔE_v is the valence band energy change going from the “-” region to the “+” region. The δ parameter represents the contribution due to thermionic field emission (tunneling).

The thermal velocities ν_n and ν_p are given by:

$$\upsilon_n = \frac{A^*_n T_L^2}{qN_C} \quad 5\text{-}43$$

$$\upsilon_p = \frac{A_p^* T_L^2}{qN_V} \quad 5\text{-}44$$

where T_L is the lattice temperature, N_C is the conduction band density of states, N_V is the valence band density of states and A_n^* and A_p^* are the electron and hole Richardson constants.

The minimum valued Richardson constants from the "-" region or "+" region are used for the calculation of the thermal velocities [188]. You can specify the Richardson constants with the ARICHN and ARICHP parameters of the MATERIAL statement. If the Richardson constants aren't specified, the following expressions will be used:

$$A_n^* = \frac{4\pi q k^2 m_n^*}{h^3} \quad 5\text{-}45$$

$$A_p^* = \frac{4\pi q k^2 m_p^*}{h^3} \quad 5\text{-}46$$

where m_n* and m_p* are the electron and hole effective masses. The electron and hole effective masses can be specified with the M.VTHN and M.VTHP parameters of the MATERIAL statement. If the effective masses aren't specified, then they will be calculated from the conduction and valence band densities of states using Equations 3-31 and 3-32 in Chapter 3: "Physics".

The tunneling factor δ in Equation 5-41 is zero when the tunneling mechanism is neglected. When you account for tunneling by specifying the TUNNEL parameter in the INTERFACE statement, the tunneling factor, δ, is calculated by using the expression:

$$\delta = \frac{1}{kT} \int_{E_{min}}^{E_C^+} exp\left(\frac{E_C^+ - E_x}{kT}\right) exp\left(\frac{-4\pi}{h} \int_0^{X_E} \left[2m_n^*(E_C - E_x)\right]^{0.5} dx\right) dE_x \quad 5\text{-}47$$

where E_x is the energy component in the x direction and $E_{min} = max[E_c(0\text{-}),E_c(W)]$ as described in Figure 5-6.

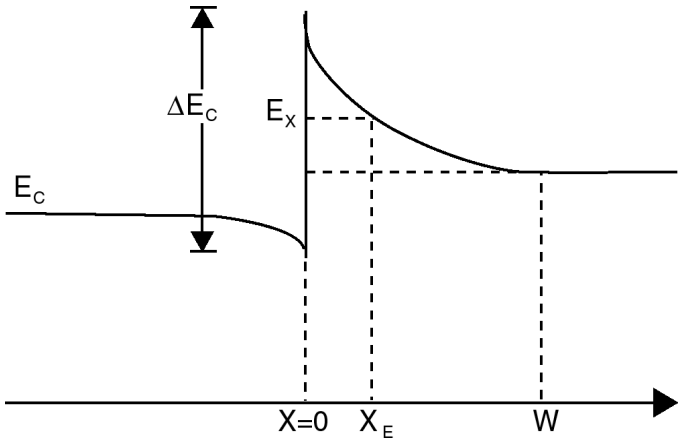


Figure 5-6: Band Diagram of p-n heterojunction

5.2: The Physical Models

Chapter 3: "Physics", Section 3.6: "Physical Models" already describes a comprehensive set of physical models for silicon. It is known, however, that for III-V, II-VI, and ternary compounds that special consideration are required such as for mole fraction dependence and material properties. The following sections describe the material dependent physical models that have been implemented into BLAZE to account for these effects.

First, a description of the common mobility model equations is given. These equations are applicable to all materials unless stated in the following material sections. Following this are sections describing the physical models on a material-per-material basis. For each material, there are descriptions of the models for bandgap narrowing, electron affinity, density of states, dielectric permittivity, and low field mobility.

5.2.1: Common Physical Models

Low Field Mobility Models

The default low field mobility models used for most materials in BLAZE are given by the following:

$$\mu_{n0}(T_L) = \texttt{MUN}\left(\frac{T_L}{300}\right)^{-\texttt{TMUN}} \quad \text{5-48}$$

$$\mu_{p0}(T_L) = \texttt{MUP}\left(\frac{T_L}{300}\right)^{-\texttt{TMUP}} \quad \text{5-49}$$

where T_L is the temperature in degrees Kelvin and the MUN, MUP, TMUN and TMUP parameters are user-definable as shown in Table 5-1.

Note: All the mobility models described in Chapter 3: "Physics", except for the TASCH model, can be used in BLAZE. But all default coefficients exist only for Silicon and therefore are not suitable for compound materials.

Table 5-1. User-Specifiable Parameters for Equations 5-48 and 5-49

Statement	Parameter	Units
MOBILITY	MUN	cm2/V·s
MOBILITY	TMUN	
MOBILITY	MUP	cm2/V·s
MOBILITY	TMUP	

Parallel Electric Field-Dependent Mobility Models

There are two types of electric field-dependent mobility models used in ATLAS/BLAZE. These models are called Standard Mobility Model and Negative Differential Mobility Model. Both of these models contain appropriate default values of parameters for different materials. You need to specify which type of mobility will be used for each material and which material parameters you want to alter.

The Standard Mobility Model that takes account of velocity saturation is defined according to:

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}E}{\mathrm{VSATN}}\right)^{\mathrm{BETAN}}}\right]^{1/\mathrm{BETAN}} \quad 5\text{-}50$$

$$\mu_p(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}E}{\mathrm{VSATP}}\right)^{\mathrm{BETAP}}}\right]^{1/\mathrm{BETAP}} \quad 5\text{-}51$$

where VSATN and VSATP are the saturation velocities for electrons and holes, BETAN and BETAP are constants given in Table 5-2, and $\mu_{n0,p0}$ are the electron and hole low field mobilities. This model is activated by specifying the FLDMOB and the EVSATMOD=0 parameter in the MODEL statement.

Table 5-2. User-Specifiable Parameters for Equations 5-50 and 5-51

Statement	Parameter	Units
MOBILITY	BETAN	
MOBILITY	BETAP	
MOBILITY	VSATN	cm/s
MOBILITY	VSATP	cm/s

The Negative Differential Mobility Model of Barnes et. al. [20] has been implemented to account for certain devices where the carrier drift velocity peaks at some electric field before reducing as the electric field increases. This model takes account of this through the carrier mobility with equations of the form:

$$\mu_n(E) = \frac{\mu_{n0}+\frac{\mathrm{VSATN}}{E}\left(\frac{E}{\mathrm{ECRITN}}\right)^{\mathrm{GAMMAN}}}{1+\left(\frac{E}{\mathrm{ECRITN}}\right)^{\mathrm{GAMMAN}}} \quad 5\text{-}52$$

$$\mu_p(E) = \frac{\mu_{p0}+\frac{\mathrm{VSATP}}{E}\left(\frac{E}{\mathrm{ECRITP}}\right)^{\mathrm{GAMMAP}}}{1+\left(\frac{E}{\mathrm{ECRITP}}\right)^{\mathrm{GAMMAP}}} \quad 5\text{-}53$$

where VSATN and VSATP are the electron and hole saturation velocities, E_0 is a constant, and $\mu_{n0,p0}$ are the low-field electron and hole mobilities. To activate this mobility model, specify EVSATMOD=1 in the MODEL statement.

Table 5-3. User-Specifiable Parameters for Equations 5-52 and 5-53			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ECRITN	4.0×10^3	V/cm
MOBILITY	ECRITP	4.0×10^3	V/cm
MOBILITY	GAMMAN	4.0	
MOBILITY	GAMMAP	1.0	

Note: The Negative Differential Mobility Model introduces an instability in the solution process and is not recommended for general use. Only activate this model in cases where the device operation directly depends on negative differential mobility (e.g., a Gunn diode).

For both the standard and negative differential models, an empirical temperature-dependent model for saturation velocity in GaAs [98] is implemented according to:

$$\text{VSATN} = \text{VSATN.T0} - \text{VSATN.T1}\ T_L \quad 5\text{-}54$$

$$\text{VSATP} = \text{VSATP.T0} - \text{VSATP.T1}\ T_L \quad 5\text{-}55$$

where VSATN and VSATP are expressed in cm/sec and T_L is the temperature in degrees Kelvin.

Alternatively, you can set the saturation velocities to constant values using the VSATN and VSATP parameters of the MATERIAL statement.

Table 5-4. User-Specifiable Parameters for Equations 5-54 and 5-55			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	VSATN.T0	1.13×10^7	cm/s
MOBILITY	VSATN.T1	1.2×10^6	cm/s·K
MOBILITY	VSATP.T0	1.13×10^7	cm/s
MOBILITY	VSATP.T1	1.2×10^4	cm/s·K

Velocity Saturation with Energy Balance Transport Model

When the Energy Balance Transport Model is activated, the mobility can be made a function of carrier energy. In Chapter 3: “Physics”, Section 3.6.1: “Mobility Modeling”, physical models for the dependence of carrier mobility on carrier energy were introduced. The same models are applicable for use within BLAZE with one additional model, which applies when the negative differential mobility model is used.

The carrier temperature dependence is activated when the `EVSATMOD=1` parameter is on the `MODEL` statement. This model can be derived in a similar fashion as in the case of `EVSATMOD=0` described in Chapter 3: “Physics”, “Parallel Electric Field-Dependent Mobility” on page 3-72. These expressions, however, require several piecewise approximations, which are too in-depth to show in this manual. To say that these piecewise expressions provide a continuous velocity saturation model for mobility versus carrier temperature are completely like the expressions for drift diffusion given in Equations 3-274 and 3-275 from Chapter 3: “Physics”.

5.2.2: Recombination and Generation Models

The recombination and generation models for compound semiconductors are the same as the models previously described in Chapter 3: “Physics”, Section 3.6.3: “Carrier Generation-Recombination Models”. The default parameters for different materials are automatically used, unless new values are specified. The default parameter values are listed in Appendix B: “Material Systems”.

5.3: Material Dependent Physical Models

5.3.1: Cubic III-V Semiconductors

The basic band parameters (*Eg*, χ, ε, *Nc*, and *Nv*) of most III-V cubic can be modeled using composition dependent interpolation schemes from the parameters of the constituent binary semiconductors. Table 5-5 lists the compound semiconductors you can model using this approach. The first column of this table gives the material name used on the REGION, DOPING, ELECTRODE, PRINT, MODELS, MATERIAL, INTERFACE, IMPACT, MOBILITY, TRAP, DEFECT, PROBE, LASER, and ODEFECTS statements. The second column of Table 5-5 shows the association of composition fractions with the constituent elements. The third column indicates whether this model is used by default (see Section 3.2.5: "Rules for Evaluation of Energy Bandgap"). If this model is not used, then there is another default model for that material described in the section listed in the last column of the table.

You can apply this model to those materials, which are not by default, by setting the CUBIC35 parameter of the MODELS statement. In specifying a material name, ATLAS will recognize alternate spellings switching the order of cations or anions as long as you remember to maintain the associations of elements with composition fractions as given in Table 5-5 (e.g., Al(x)Ga(1-x)As = Ga(1-x)Al(x)As).

Table 5-5. Cubic III-V Interpolation Model Applicability Chart

Atlas Name	Composition	Default	Section
GaAs		No	5.3.2
AlP		Yes	
AlAs		No	5.3.3
AlSb		Yes	
GaSb		Yes	
GaP		No	5.3.4
InP		No	5.3.4
InSb		Yes	
InAs		Yes	
AlGaAs	Al(x)Ga(1-x)As	No	5.3.3
InAlP	In(1-x)Al(x)P	Yes	
InGaAs	In(1-x)Ga(x)As	No	5.3.4
InGaP	In(1-x)Ga(x)P	No	5.3.4
GaSbP	GaSb(1-y)P(y)	Yes	
InAlAs	In(1-x)Al(x)As	Yes	
InAsP	InAs(1-y)P(y)	No	5.3.4
GaAsP	GaAs(1-y)P(y)	No	5.3.4
InGaSb	In(1-x)Ga(x)Sb	Yes	
InAlSb	In(1-x)Al(x)Sb	Yes	

Table 5-5. Cubic III-V Interpolation Model Applicability Chart			
AlGaSb	Al(x)Ga(1-x)Sb	Yes	
InAsSb	InAs(y)Sb(1-y)	Yes	
GaAsSb	GaAs(y)Sb(1-y)	Yes	
AlAsSb	AlAs(y)Sb(1-y)	Yes	
InPSb	InSb(1-y)P(y)	Yes	
AlPSb	AlP(y)Sb(1-y)	Yes	
AlPAs	AlP(y)As(1-y)	Yes	
AlGaP	Al(x)Ga(1-x)P	Yes	
InPAsSb	InP(x)As(y)Sb(1-x-y)	Yes	
InGaAsP	In(1-x)Ga(x)As(1-y)P(y)	No	5.3.4
AlGaAsP	Al(x)Ga(1-x)As(1-y)P(y)	Yes	
AlGaAsSb	Al(x)Ga(1-x)As(y)Sb(1-y)	Yes	
InAlGaAs	In(1-x-y)Al(x)Ga(y)As	Yes	
InAlGaP	In(1-x-y)Al(x)Ga(y)P	Yes	
InAlAsP	In(1-x)Al(x)As(1-y)P(y)	Yes	
InGaAsSb	In(1-x)Ga(x)As(y)Sb(1-y)	Yes	
InAlAsSb	In(1-x)Al(x)As(y)Sb(1-y)	Yes	

Energy Bandgap

The energy bandgaps of the binary compounds are calculated for transitions in each of the Γ, X and L directions using the universal formula for temperature dependent bandgap (see Equation 3-38). Table 5-5 shows the default parameters for Equation 3-38 for the binary compounds.

Table 5-6. Default Bandbap Parameters for Cubic III-V Binary Compounds [120]									
Binary	**$Eg^{\Gamma}(0)$ (eV)**	**α^{Γ} (meV/K)**	**β^{Γ} (K)**	**Eg^{X} (0) (eV)**	**α^{X} (meV/K)**	**β^{X} (K)**	**$Eg^{L}(0)$ (eV)**	**α^{L} (meV/K)**	**β^{L} (K)**
GaAs	1.519	0.5405	204.0	1.981	0.4600	204.0	1.815	0.6050	204
AlP	3.63	0.5771	372.0	2.52	0.3180	588.0	3.57	0.3180	588
AlAs	3.099	0.885	530.0	2.240	0.700	530.0	2.460	0.605	240
AlSb	2.386	0.42	140.0	1.696	0.39	140.0	2.329	0.58	140
GaSb	0.812	0.417	140.0	1.141	0.475	94.00	0.875	0.597	140
GaP	2.886	0.0	0.0	2.350	0.5771	372.0	2.720	0.5771	372
InP	1.4236	0.363	162.0	2.3840	0.0	0.0	2.0140	0.363	162
InSb	0.235	0.32	170.0	0.630	0.00	0.0	0.930	0.00	0.0
InAs	0.417	0.276	93.0	1.433	0.276	93.0	1.133	0.276	93

For the ternary materials the bandgap in each principal direction is approximated by the function [2]:

$$Tabc(x) = xTac + (1-x)Tbc - x(1-x)Cabc \quad 5\text{-}56$$

where *Tabc* is the ternary bandgap, *Tac* and *Tbc* are the binary bandgaps, *x* is the composition fraction and *Cabc* is the bowing factor as given for each of the ternary compounds in each of the principal directions in Table 5-6.

Table 5-7. Bowing Factors and Alignment for Cubic III-V Ternary Compounds [120]

Ternary	C(Eg^{Γ}) (eV)	C(Eg^{X}) (eV)	C(Eg^{L}) (eV)	dEc/dEg
AlGaAs	1.31*x-0.127	0.055	0.0	0.65
InAlP	-0.48	0.38	0.0	0.5
InGaAs	0.477	1.4	0.33	0.4
InGaP	0.65	0.2	1.3	0.4
GaSbP	2.558	2.7	2.7	0.5
GaSbAs	1.43	1.2	1.2	0.5
InAlAs	0.7	0.0	0.0	0.7
InAsP	0.1	0.27	0.27	0.4
GaAsP	0.19	0.24	0.16	0.4
InGaSb	0.415	0.330	0.4	0.5
InAlSb	0.43	0.0	0.0	0.5
AlGaSb	1.22*x-0.044	0.0	0.0	0.5
InAsSb	0.67	0.6	0.6	0.5
GaAsSb	1.43	1.2	1.2	0.5
AlAsSb	0.84	0.28	0.28	0.5
InPSb	1.9	1.9	1.9	0.5
AlPSb	3.56	2.7	2.7	0.5
AlPAs	0.22	0.22	0.22	0.5
AlGaP	0.0	0.13	0.0	0.5

For the quaternary compounds, the bandgap in each principal direction is approximated by the function[2]:

$$Qabcd(x,y) = \frac{x(1-x[yTabc(x) + (1-y)Tabd(x)] + y(1-y)[xTacd(y) + (1-x)Tbcd(y)])}{x(1-x) + y(1-y)} \quad 5\text{-}57$$

or for compounds with three anions [2]:

$$Qabcd(x,y) = \frac{xyTabc(u) + y(1-x-y)Tacd(v) + (1-x-y)xTabd(w)}{xy + y(1-x-y) + (1-x-y)x} \quad 5\text{-}58$$

where *Tabc*, *Tabd*, *Tacd*, and *Tbcd* are the ternary bandgaps, x and y are the composition fractions and u, v and w are given as follows:

$$u = (1-x+y)/2 \quad 5\text{-}59$$

$$v = (2-x-2y)/2 \quad 5\text{-}60$$

and

$$w = (2-2x-y)/2 \quad 5\text{-}61$$

For any case (binary, ternary or quaternary), the bandbap used in the drift-diffusion calculations is taken as the minimum of Eg^{Γ}, Eg^{χ} or Eg^{L}.

Electron Affinity

Table 5-8 shows the electron affinities for the binary compounds in the given direction in k space.

Table 5-8. Default Electron Affinities and Alignments for Cubic III-V Binary Compounds [120]

Binary	Affinity (eV)	Direction	dEc/dEg
GaAs	4.07	Γ	0.65
AlP	3.98	X	0.7
AlAs	3.85	X	0.65
AlSb	3.60	X	0.5
GaSb	4.06	Γ	0.5
GaP	4.0	X	0.4
InP	4.4	Γ	0.4
InSb	4.59	Γ	0.5
InAs	4.90	Γ	0.65

The affinities for the other directions in k space not given in Table 5-8 are calculated by the following:

$$\chi^{\Gamma} = \chi^{X} - (Eg^{X} - Eg^{\Gamma})(dEc/dEg) \quad 5\text{-}62$$

$$\chi^{\chi} = \chi^{\Gamma} - (Eg^{\Gamma} - Eg^{X})(dEc/dEg) \quad 5\text{-}63$$

$$\chi^{L} = \chi^{X/\Gamma} - (Eg^{X/\Gamma} - Eg^{L})(dEc/dEg) \quad 5\text{-}64$$

where X^{Γ}, X^{X} and X^{L} are the affinities in each of the principal directions, Eg^{Γ}, Eg^{X} and Eg^{L} are the energy bandgaps in each of the principal directions as calculated above and dEc/dEg is an alignment parameter describing the proportion of change in bandgap that can be associated with the conduction band edge. Table 5-8 lists the default values of dEc/dEg. You can define the value of dEc/dEg by using the ALIGN parameter of the MATERIAL statement.

For ternary compounds, the affinities are calculated from the binary values using Equation 5-56 except *Tabc* is the ternary affinity, *Tac* and *Tbc* are the binary affinities and *Cabc* is the bowing factor for affinity which is given by:

$$Cabc(\chi) = -Cabc(Eg)dEc/dEg \quad 5\text{-}65$$

where *Cabc(χ)* is the bowing factor for affinity, *Cabc(Eg)* is the bowing factor for bandgap as given in Table 5-7 and dEc/dEg is the alignment parameter as given in Table 5-7.

For quaternaries, the affinities are approximated by the Equations 5-57 and 5-58 with affinities replacing energy bandgaps.

Finally for all cases (binary, ternary and quaternary), the affinity used in the drift-diffusion calculation is chosen as the one in the same direction (Γ , X or L) as chosen for energy bandgap.

Density of States

For the III-V cubic semiconductor model, the density of states (*Nc* and *Nv*) are each calculated from the effective masses using Equations 3-31 and 3-32. For binary materials, the density of states effective masses are given in Table 5-10.

Table 5-9. Default Cubic III-V Effective Masses and Permittivities

Binary	m_c^{Γ} (×mo)	m_{ct}^{X} (×mo)	m_{cl}^{X} (×mo)	m_{ct}^{Γ} (×mo)	m_{cl}^{Γ} (×mo)	m_{hh} (×mo)	m_{lh} (×mo)	ε_r
GaAs	0.067	0.23	1.3	0.0754	1.9	0.49	0.16	12.91
AlP	0.22	0.155	2.68	0.0	0.0	0.63	0.20	9.8
AlAs	0.15	0.22	0.97	0.15	1.32	0.76	0.15	10.06
AlSb	0.14	0.123	1.357	0.23	1.64	0.94	0.14	12.04
GaSb	0.039	0.22	1.51	0.10	1.3	0.28	0.05	15.69
GaP	0.13	0.253	2.0	0.15	1.2	0.79	0.14	11.10
InP	0.0795	0.88	0.0	0.47	0.0	0.56	0.12	12.61
InSb	0.0135	0.0	0.0	0.25	0.0	0.43	0.015	17.70
InAs	0.026	0.16	1.13	0.05	0.64	0.57	0.025	15.15

For ternary materials, the effective masses are interpolated using the following expression[2]:

$$Mabc = xMac + (1-x)Mbc \quad 5\text{-}66$$

where *Mabc* is the ternary effective mass, *Mac* and *Mbc* are the binary effective masses and *x* is the composition fraction.

For quaternary compounds the effective masses are interpolated using the following expression[2]:

$$Mabcd = xyMac + x(1-y)Mad + (1-x)yMbc + (1-x)(1-y)Mbd \quad 5\text{-}67$$

or for compounds with 3 anions [2]:

$$Mabcd = xMab + yMab + (1 - x - y)Mad \quad \text{5-68}$$

where *Mabcd* is the quaternary effective mass, *Mac*, *Mad*, *Mbc* and *Mbd* are the binary effective masses and x and y are the composition fractions.

For the calculation of conduction band density of states, the conduction band density of states mass is calculated for the same direction (Γ , X or L) as is used for energy bandgap as follows:

$$m_c = m_c^{\Gamma} \quad \text{5-69}$$

$$m_c = \left[\left(m_{ct}^{X}\right)^2 \cdot \left(m_{cl}^{X}\right)\right]^{1/3} \quad \text{5-70}$$

or

$$m_c = \left[\left(m_{ct}^{L}\right)^2 \cdot \left(m_{cl}^{L}\right)\right]^{1/3} \quad \text{5-71}$$

For the calculation of valence band density of states, the valence band density of states mass is calculated as follows:

$$m_v = \left(m_{hh}^{3/2} + m_{lh}^{3/2}\right)^{2/3} \quad \text{5-72}$$

Static Permittivity

Table 5-9 shows the permittivities of the binary compounds. The ternary and quaternary permittivities are approximated by the interpolation formulas analogous to those used for effective mass (Equations 5-66, 5-67 and 5-68) replacing masses by permittivities.

5.3.2: Gallium Arsenide (GaAs) Physical Models

Bandgap Narrowing

Following Klausmeier-Brown [89], the bandgap narrowing effects are important only for p-type regions. By default, BLAZE uses the bandgap narrowing values shown in Table 5-10.

Table 5-10. Default Bandgap Narrowing Values

Concentration (cm^{-3})	Bandgap Narrowing (meV)
1.0×10^{18}	31.0
2.0×10^{18}	36.0
4.0×10^{18}	44.2
6.0×10^{18}	48.5
8.0×10^{18}	51.7
1.0×10^{19}	54.3

Table 5-10. Default Bandgap Narrowing Values	
Concentration (cm^{-3})	**Bandgap Narrowing (meV)**
2.0×10^{19}	61.1
4.0×10^{19}	64.4
6.0×10^{19}	61.9
8.0×10^{19}	56.9
1.0×10^{20}	53.2
2.0×10^{20}	18.0

Note: This table is only used for GaAs. No data is available at present for other materials. The C-INTERPRETER function for bandgap narrowing, however, allows a user-defined model for bandgap narrowing to be applied for these materials. See Appendix A: "C-Interpreter Functions" for more information about on these functions.

Low Field Mobility

You can make the mobility in GaAs concentration dependent by setting the CONMOB parameter of the MODEL statement. In this model, mobility is interpolated from the values in Table 5-11.

Table 5-11. Default Concentration Dependent Mobility for GaAs		
Concentration (cm^{-3})	**Mobility in GaAs (cm^2/v-s)**	
	Electrons	**Holes**
1.0×10^{14}	8000.0	390.0
2.0×10^{14}	7718.0	380.0
4.0×10^{14}	7445.0	375.0
6.0×10^{14}	7290.0	360.0
8.0×10^{14}	7182.0	350.0
1.0×10^{15}	7300.0	340.0
2.0×10^{15}	6847.0	335.0
4.0×10^{15}	6422.0	320.0
6.0×10^{15}	6185.0	315.0
8.0×10^{15}	6023.0	305.0
1.0×10^{16}	5900.0	302.0

Table 5-11. Default Concentration Dependent Mobility for GaAs		
Concentration (cm^{-3})	Mobility in GaAs (cm^2/v-s)	
	Electrons	Holes
2.0×10^{16}	5474.0	300.0
4.0×10^{16}	5079.0	285.0
6.0×10^{16}	4861.0	270.0
8.0×10^{16}	4712.0	245.0
1.0×10^{17}	4600.0	240.0
2.0×10^{17}	3874.0	210.0
4.0×10^{17}	3263.0	205.0
6.0×10^{17}	2950.0	200.0
8.0×10^{17}	2747.0	186.9
1.0×10^{18}	2600.0	170.0
2.0×10^{18}	2060.0	130.0
4.0×10^{18}	1632.0	90.0
6.0×10^{18}	1424.0	74.5
8.0×10^{18}	1293.0	66.6
1.0×10^{20}	1200.0	61.0

If MODEL ANALYTIC is specified, the program will use [89]:

$$\mu_{n,\,p} = 940 + \frac{8000 - 940}{1 + \left(\frac{Nt}{2.8 \times 10^{16}}\right)^{0.75}} \qquad 5\text{-}73$$

where Nt is the total impurity concentration.

5.3.3: Al(x)Ga(1-x)As System

The Al(x)Ga(1-x)As material system is commonly used for the fabrication of heterojunction devices. These materials are available in BLAZE by specifying the material name GaAs, AlGaAs, or AlAs. As a ternary material system, different material properties are obtained by adjusting the molar fraction of Aluminum and Gallium. This mole fraction is represented by the x as written in Al(x)Ga(1-x)As. GaAs material parameters are identical to the those of AlGaAs with the mole fraction x set equal to zero. AlAs material parameters are identical to the those of AlGaAs with mole the fraction x set equal to one. Fundamental in the proper simulation with the AlGaAs material system is the relationship between this mole fraction x, and the material parameters for that composition. In the following sections, the relationship between mole fraction and material parameters for the AlGaAs material

system will be described. You can specify the x.composition of a material in the REGION statement with the X.COMP parameter.

Bandgap

There are three primary conduction bands in the AlGaAs system, depending on mole fraction, that determine the bandgap. These are named Gamma, L, and X. The default bandgaps for each of these conduction band valleys are as follows:

$$E_{g\Gamma} = \mathrm{EG300} + \mathrm{x.comp} \cdot (1.155 + 0.37 \cdot \mathrm{X.COMP}) \quad \text{5-74}$$

$$E_{gL} = 1.734 + \mathrm{x.comp} \cdot (0.574 + 0.055 \cdot \mathrm{X.COMP}) \quad \text{5-75}$$

$$E_{gX} = 1.911 + \mathrm{x.comp} \cdot (0.005 + 0.245 \cdot \mathrm{X.COMP}) \quad \text{5-76}$$

The bandgap used for any given Al concentration is the minimum as calculated from these equations. EG300 is the bandgap at 300K and specified on the material statement. x.composition is the Aluminum mole fraction and can be user-defined in the REGION statement.

The temperature dependence of the bandgap is calculated according to:

$$E_g(T_L) = E_g(300) + \mathrm{EGALPHA} \cdot \left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}} \right] \quad \text{5-77}$$

The value of $E_g(300)$ is taken as the minimum of $E_{g\Gamma}$, E_{gx}, and E_{gL}. The default temperature dependent bandgap parameters for AlGaAs are listed in Table 5-12.

Table 5-12. Default Bandgap Parameters for Al(x)Ga(1-x)As

Statement	Parameter	Default	Units
MATERIAL	EG300	1.59	eV
MATERIAL	EGALPHA	5.405×10^{-4}	eV/K
MATERIAL	EGBETA	204	K

Electron Affinity

As indicated in the introduction, the semiconductor electron affinity χ is a key parameter for determining the alignment of heterojunctions. For AlGaAs, χ is a function of $E_{g\Gamma}$ and is given by:

$$\chi_{AlGaAs} = 4.07 - 0.85 \cdot (E_{g\Gamma}(\mathrm{X.COMP}) - E_{gGaAs}) \quad \text{5-78}$$

Density of States and Effective Mass

The valence and conduction band densities of states, N_C and N_V, are calculated from the effective masses according to the following equations:

$$N_c = 2\left(\frac{2\pi m_e^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad 5\text{-}79$$

$$N_v = 2\left(\frac{2\pi m_h^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad 5\text{-}80$$

For the AlGaAs system the conduction band and valence band effective masses, for electrons and holes, are given by [100]:

$$m_e = \begin{cases} 0.067 + 0.083x (0 < x < 0.45) \\ 0.85 - 0.14x (x > 0.45) \end{cases} \quad 5\text{-}81$$

$$m_{lh} = 0.087 + 0.063x \quad 5\text{-}82$$

$$m_{hh} = 0.62 + 0.14x \quad 5\text{-}83$$

$$m_h = (m_{lh}^{3/2} + m_{hh}^{3/2})^{2/3} \quad 5\text{-}84$$

Dielectric Permittivity

The default static dielectric constant for AlGaAs is given by:

$$\varepsilon_{AlGaAs} = 13.18 - 2.9 \cdot \mathrm{X.COMP} \quad 5\text{-}85$$

Low Field Mobility

The default low field electron mobility for AlGaAs is a function of the composition fraction, x, within the system. The following equations outline this relationship.

Table 5-13. Low Field Mobility Equations

AlGaAs Low Field Mobility	Mole Fraction Range
$\mu_n = 8000 - (1.818 \times 10^4 \cdot \mathrm{X.COMP})$	$(0 < \mathrm{X.COMP} < 0.429)$

Table 5-13. Low Field Mobility Equations	
AlGaAs Low Field Mobility	**Mole Fraction Range**
$\mu_n = 90 + 1.1435\times10^4 \cdot (\mathrm{X.COMP} - 0.46)^2$	$(0.429 < \mathrm{X.COMP} < 0.46)$
$\mu_n = 90 + 3.75\times10^4 \cdot (\mathrm{X.COMP} - 0.46)^2$	$(0.46 < \mathrm{X.COMP} < 0.5)$
$\mu_n = 200 - (2 / (\mathrm{X.COMP} - 0.46))$	$(0.5 < \mathrm{X.COMP} < 1.0)$

5.3.4: In(1-x)Ga(x)As(y)P(1-y) System

The In(1-x)Ga(x)As(y)P(1-y) material system is commonly used for the fabrication of heterojunction devices. These include laser diodes, photodiodes, Gunn diodes, and high speed heterostructure transistors. As a quaternary material, two different mole fraction parameters, x and y, are necessary to specify any particular combination. This produces a wide array of InGaAsP materials and characteristics. Of particular interest in this system are materials that are lattice matched to InP.

The default material characteristics in BLAZE for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP. The xcomposition and ycomposition are specified in the REGION statement with the X.COMP and Y.COMP parameters. The relationship between x and y that satisfy this condition is given by:

$$x = \frac{0.1896 \cdot \mathrm{Y.COMP}}{0.4176 - (0.0125 \cdot \mathrm{Y.COMP})} \qquad 0 < \mathrm{Y.COMP} < 1 \tag{5-86}$$

Many of the parameter models for the In(1-x)Ga(x)As(y)P(1-y) system are functions of the composition fraction Y.COMP only. The composition fraction, X.COMP, can be deduced from the preceding relationship. Again, the default material characteristics in BLAZE for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP.

Note: Don't use this material system to form GaAs by setting x=1 and y=1, specify GaAs as the material instead.

Bandgap

The default energy bandgap for the InP lattice matched In(1-x)Ga(x)As(y)P(1-y) system used in BLAZE is given by:

$$E_g(InGaAsP) = 1.35 + \mathrm{X.COMP} \cdot (0.642 + (0.758 \cdot \mathrm{X.COMP})) + (0.101 \cdot \mathrm{Y.COMP} - 1.101) \cdot \mathrm{Y.COMP} - (0.28 \cdot \mathrm{X.COMP} - 0.109 \cdot \mathrm{Y.COMP} + 0.159) \cdot \mathrm{X.COMP} \cdot \mathrm{Y.COMP} \tag{5-87}$$

Electron Affinity

The electron affinities for materials in the InP lattice matched InGaAsP system are derived from conduction band offsets and from the assumption that the affinity of InP is 4.4eV. The default conduction band edge offset between lattice matched InGaAsP and InP is then:

$$\Delta E_c = 0.268 \cdot \mathrm{Y.COMP} + 0.003 \cdot (\mathrm{Y.COMP})^2 \tag{5-88}$$

Density of States and Effective Mass

The density of states is defined, as before, as a function of the effective masses of electrons and holes according to Chapter 3: "Physics", Equation 3-31. For the InGaAsP system, the default conduction and valence band effective masses, for electrons and holes, are given by the following.

For the conduction band:

$$m_e^* = 0.08 - (0.116 \cdot \mathrm{Y.COMP}) + (0.026 \cdot \mathrm{X.COMP}) - 0.059 \cdot (\mathrm{X.COMP} \cdot \mathrm{Y.COMP}) + (0.064 - 0.02 \cdot \mathrm{Y.COMP}) \cdot (\mathrm{X.COMP})^2 + (0.06 + 0.032 \cdot \mathrm{X.COMP}) \cdot (\mathrm{Y.COMP})^2 \quad 5\text{-}89$$

For the valence band the hole effective mass is defined by:

$$m_h^* = \left(m_{lh}^{1.5} + m_{hh}^{1.5}\right)^{\frac{2}{3}} \quad 5\text{-}90$$

where the default light hole effective mass is given by:

$$m_{lh} = 0.120 - (0.116 \cdot \mathrm{Y.COMP}) + 0.03 \cdot (\mathrm{X.COMP})^2 \quad 5\text{-}91$$

and the default heavy hole effective mass is a constant and is given by:

$$m_{hh} = 0.46 \quad 5\text{-}92$$

Dielectric Permittivity

The default static dielectric constant for lattice matched InGaAsP to InP is given by

$$\varepsilon_{InGaAsP} = (14.6 \cdot (1 - \mathrm{X.COMP}) \cdot \mathrm{Y.COMP}) + 12.5 \cdot (1 - \mathrm{X.COMP}) \cdot (1 - \mathrm{Y.COMP}) + 13.18 \cdot \mathrm{X.COMP} \cdot \mathrm{Y.COMP} + 11.11 \cdot \mathrm{X.COMP} \cdot (1 - \mathrm{Y.COMP}) \quad 5\text{-}93$$

Low Field Mobility

The default low field mobility parameters for electrons and holes for lattice matched InGaAs are given by linear interpolations from the binary compounds GaAs and InP. The following formulas are used:

$$\mu_{n1} = 33000 + (8500 - 33000) \cdot \mathrm{X.COMP} \quad 5\text{-}94$$

$$\mu_{p1} = 460 + (400 - 460) \cdot \mathrm{X.COMP} \quad 5\text{-}95$$

$$\mu_{n2} = 4600 + (300 - 4600) \cdot \mathrm{X.COMP} \quad 5\text{-}96$$

$$\mu_{p2} = 150 + (100 - 150) \cdot \mathrm{X.COMP} \quad 5\text{-}97$$

$$\mu_{n0} = \mu_{n1} + (1 - \mathrm{Y.COMP})(\mu_{n2} - \mu_{n1}) \quad 5\text{-}98$$

$$\mu_{p0} = \mu_{p1} + (1 - \mathrm{Y.COMP})(\mu_{p2} - \mu_{p1}) \quad 5\text{-}99$$

5.3.5: The Si(1-x)Ge(x) System

Advances in the growth of Silicon and Si(1-x)Ge(x) alloys have allowed the potential for using bandgap engineering to construct heterojunction devices such as HBTs and HEMTs using these materials. BLAZE supports the SiGe material system by providing composition dependent material parameters. These parameters are accessed by specifying the material name SiGe.

The following sections describe the functional relationship between Ge mole fraction x, and the SiGe material characteristics necessary for device simulation.

Bandgap

Bandgap is one of the most fundamental parameters for any material. For SiGe, the dependence of the bandgap on the Ge mole fraction, *x.composition*, is divided into ranges as follows:

$$E_g = 1.08 + \mathrm{X.COMP} \cdot (0.945 - 1.08) / 0.245); \quad \text{5-100}$$

for $x \leq 0.245$

$$E_g = 0.945 + (\mathrm{X.COMP} - 0.245) \cdot (0.87 - 0.945) / (0.35 - 0.245); \quad \text{5-101}$$

for $0.245 < x \leq 0.35$

$$E_g = 0.87 + (\mathrm{X.COMP} - 0.35) \cdot (0.78 - 0.87) / (0.5 - 0.35); \quad \text{5-102}$$

for $0.35 < x \leq 0.5$

$$E_g = 0.78 + (\mathrm{X.COMP} - 0.5) \cdot (0.72 - 0.78) / (0.6 - 0.5); \quad \text{5-103}$$

for $0.5 < x \leq 0.6$

$$E_g = 0.72 + (\mathrm{X.COMP} - 0.6) \cdot (0.69 - 0.72) / (0.675 - 0.6); \quad \text{5-104}$$

for $0.6 < x \leq 0.675$

$$E_g = 0.69 + (\mathrm{X.COMP} - 0.675) \cdot (0.67 - 0.69) / (0.735 - 0.675); \quad \text{5-105}$$

for $0.675 < x \leq 0.735$

$$E_g = 0.67; \quad \text{5-106}$$

for $0.735 < x \leq 1$

The temperature dependence of the bandgap of SiGe is calculated the same as for Silicon except that EGALPHA and EGBETA are a function of Ge mole fraction x as follows:

$$E_g(T_L) = E_g + \mathrm{EGALPHA}\left[\frac{300^2}{300 + \mathrm{EGBETA}} - \frac{T_L^2}{T_L + \mathrm{EGBETA}}\right] \quad \text{5-107}$$

$$\mathrm{EGALPHA} = (4.73 + \mathrm{X.COMP} \cdot (4.77 - 4.73)) \times 10^4 \quad \text{5-108}$$

$$\mathrm{EGBETA} = 626 + \mathrm{X.COMP} \cdot (235 - 636) \quad \text{5-109}$$

where E_g is dependent upon the mole fraction as above.

Electron Affinity

The electron affinity χ of SiGe is taken to be constant (4.17) with respect to composition.

Density of States

The density of states for SiGe is defined differently compared to the previous materials by not being a function of the effective masses. Instead the density of states have been made to depend upon the Ge mole fraction, *x.composition*,according to:

$$N_c = 2.8\times10^{19} + \mathrm{X.COMP}\cdot(1.04\times10^{19} - 2.8\times10^{19}) \quad \text{5-110}$$

$$N_v = 1.04\times10^{19} + \mathrm{X.COMP}\cdot(6.0\times10^{18} - 1.04\times10^{19}) \quad \text{5-111}$$

Dielectric Function

The compositional dependence of the static dielectric constant of SiGe is given by

$$\varepsilon = 11.8 + 4.2\cdot\mathrm{X.COMP} \quad \text{5-112}$$

Low Field Mobility

No specific SiGe low field mobility models have been implemented into BLAZE.

Velocity Saturation

In SiGe, the temperature dependent velocity saturation, used in the field dependent mobility model is defined by the following equations.

$$\mathrm{VSATN} = 1.38{*}10^{7}\cdot\sqrt{\left(\tanh\left(\frac{175}{T_L}\right)\right)} \quad \text{5-113}$$

$$\mathrm{VSATP} = 9.05{*}10^{6}\cdot\sqrt{\left(\tanh\left(\frac{312}{T_L}\right)\right)} \quad \text{5-114}$$

Note: All other defaults used for SiGe are taken from Silicon

5.3.6: Silicon Carbide (SiC)

Silicon carbide materials are of interest for high power, high temperature applications. The main characteristics of silicon carbide are that they have a very wide bandgap, high thermal conductivity, high saturation velocity, and high breakdown strength. Silicon carbide is commercially available in three polytypes called 6H-SiC, 3H-SiC, and 4H-SiC. ATLAS supports all three these polytypes. The following paragraphs describe the material defaults for these materials.

Band Parameters for SiC

SiC band parameter equations are identical to those used for Silicon but with the values adjusted for 3C-SiC, 4H-SiC, and 6H-SiC. The physical band parameter values are shown in Tables B-15 and B-16 of Appendix B: “Material Systems”.

SiC Mobility Parameters

Isotropic Mobility

By default, mobility is assumed to be entirely isotropic in nature. That is, there is no directional component. The default low field mobilities of electrons and holes for 3C-SiC, 4H-SiC, and 6H-SiC are shown in Table 5-14.

Table 5-14. Silicon Carbide Low Field Mobility Defaults

Statement	Parameter	6H-SiC	3C-SiC	4H-SiC	Units
MOBILITY	MUN	330	1000	460	$cm^2/V \cdot s$
MOBILITY	MUP	60	50	124	$cm^2/V \cdot s$

Anisotropic Mobility

The mobility behavior within SiC is now known to be anisotropic in nature, which dramatically alters the electrical performance of a device. A anisotropic model has been implemented into ATLAS to correctly model this behavior. Following the ideas of Lindefelt [97] and Lades [92], the mobility within the drift diffusion equations has been made a tensor property. As a result the mobility has become:

$$\mu = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_2 \end{bmatrix} \quad \text{5-115}$$

where μ_1 represents the mobility defined in one plane and μ_2 the mobility defined in a second plane. In the case of SiC, μ_1 represents the mobility of plane <1100> while μ_2 represents the mobility of plane <1000>. These mobilities are defined for both holes and electrons.

Defining Anisotropic Mobility in ATLAS

To define a material with anisotropic mobility, specify two MOBILITY statements. In each statement, the N.ANGLE and P.ANGLE parameters are used to specify the direction where that particular mobility is to apply. The following example shows how this is done.

```
# FIRST DEFINE MOBILITY IN PLANE <1100>
#
MOBILITY MATERIAL=3C-SiC VSATN=2E7 VSATP=2E7 BETAN=2 BETAP=2 \
   MU1N.CAUG=10 MU2N.CAUG=410 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=20 MU2P.CAUG=95 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
#
# NOW DEFINE MOBILITY IN PLANE <1000>
#
```

```
MOBILITY MATERIAL=3C-SiC N.ANGLE=90.0 VSATN=2E7 VSATP=2E7 BETAN=2 BETAP=2 \
   MU1N.CAUG=5 MU2N.CAUG=80 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=2.5 MU2P.CAUG=20 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
```

Impact Ionization and Thermal Parameters

The equations governing these effects are identical to those for Silicon but with adjusted coefficients. See Appendix B: “Material Systems” for a list of all these parameters.

5.3.7: GaN, InN, AlN, AlGaN, and InGaN System

The following sections describe the relationship between mole fraction, X.COMP, and the material parameters of the Al/In/GaN system.

Bandgap

By default, the bandgap for the nitrides is calculated in a two step process. First, the bandgap(s) of the relevant binary compounds are computed as a function of temperature, T, using Equations 5-116 through 5-118 [169].

$$Eg(GaN) = 3.507 - \frac{0.909 \times 10^{-3} T^2}{T + 830.0} \quad \text{5-116}$$

$$Eg(InN) = 1.994 - \frac{0.245 \times 10^{-3} T^2}{T + 624.0} \quad \text{5-117}$$

$$Eg(AlN) = 6.23 - \frac{1.799 \times 10^{-3} T^2}{T + 1462.0} \quad \text{5-118}$$

Then, the dependence on composition fraction, x, is described by Equations 5-119 and 5-120 [122].

$$Eg(In_xGa_{1-x}N) = Eg(InN)x + Eg(GaN)(1-x) - 3.8x(1-x) \quad \text{5-119}$$

$$Eg(Al_xGa_{1-x}N) = Eg(AlN)x + Eg(GaN)(1-x) - 1.3x(1-x) \quad \text{5-120}$$

Electron Affinity

The electron affinity is calculated such that the band edge offset ratio is given by [122].

$$\frac{\Delta Ec}{\Delta Ev} = \frac{0.7}{0.3} \quad \text{5-121}$$

You can overide this ratio by specifying the ALIGN parameter of the MATERIAL statement.

Permittivity

The permittivity of the nitrides as a function of composition fraction, x, is given by linear interpolations of the values for the binary compounds as in Equations 5-122 and 5-123 [7].

$$\varepsilon(In_xGa_{1-x}N) = 19.6x + 10.4(1-x) \quad \text{5-122}$$

$$\varepsilon(Al_xGa_{1-x}N) = 10.1x + 10.4(1-x) \quad \text{5-123}$$

Density of States Masses

The nitride density of states masses as a function of composition fraction, x, is given by linear interpolations of the values for the binary compounds as in Equations 5-124 through 5-127 [169].

$$m_e(In_xGa_{1-x}N) = 0.12x + 0.2(1-x) \quad \text{5-124}$$

$$m_h(In_xGa_{1-x}N) = 0.17x + 1.0(1-x) \quad \text{5-125}$$

$$m_e(Al_xGa_{1-x}N) = 0.314x + 0.2(1-x) \quad \text{5-126}$$

$$m_h(Al_xGa_{1-x}N) = 0.417x + 1.0(1-x) \quad \text{5-127}$$

Low Field Mobility

The Albrecht Model

You can choose to model low field mobility following the work of Albrecht et.al [5] by specifying ALBRCT on the MODEL statement or ALBRCT.N or ALBRCT.P or both on the MOBILITY statement for separate control over electrons and holes. This model is described as follows:

$$\frac{1}{\mu(N,T)} = \frac{\text{AN.ALBRCT} \cdot N}{\text{N0N} \cdot \text{ALBRCT}}\left(\frac{T}{\text{T0N} \cdot \text{ALBRCT}}\right)^{-3/2} \ln\left[1 + 3\left(\frac{T}{\text{T0N} \cdot \text{ALBRCT}}\right)^{2}\left(\frac{N}{\text{N0N} \cdot \text{ALBRCT}}\right)^{-2/3}\right] + \text{BN.ALBRCT} \times \left(\frac{T}{\text{T0N} \cdot \text{ALBRCT}}\right)^{3/2} + \frac{\text{CN.ALBRCT}}{exp\,\text{T1N} \cdot \text{ALBRCT}/T - 1} \quad \text{5-128}$$

where $\mu(N,T)$ is the mobility as a function of doping and lattice temperature, N is the total doping concentration, and T is the lattice temperature. AN.ALBRCT, BN.ALBRCT, CN.ALBRCT, N0N.ALBRCT, T0N.ALBRCT and T1N.ALBRCT are user-specifiable parameters on the MOBILITY statement. You can use a similar expression for holes with the user-defined parameters AP.ALBRCT, BP.ALBRCT, CP.ALBRCT, N0P.ALBRCT, T0P.ALBRCT and T1P.ALBRCT.

Table 5-20 shows the default values for the parameters of the Albrecht Model.

Table 5-15. Default Parameter Values for the Albrecht Model

Parameter	Default	Parameter	Default	Units
AN.ALBRCT	2.61e-4	AP.ALBRCT	2.61e-4	V*s/(cm^2)
BN.ALBRCT	2.9e-4	BP.ALBRCT	2.9e-4	V*s/(cm^2)
CN.ALBRCT	170.0e-4	CP.ALBRCT	170.0e-4	V*s/(cm^2)

Table 5-15. Default Parameter Values for the Albrecht Model				
N0N.ALBRCT	1.0e17	N0P.ALBRCT	1.0e17	cm^{-3}
T0N.ALBRCT	300.0	T0P.ALBRCT	300.0	K
T1N.ALBRCT	1065.0	T1P.ALBRCT	1065.0	K

Farahmand Modified Caughey Thomas

You can use a composition and temperature dependent low field model by specifying the FMCT.N and FMCT.P in the MOBILITY statement. FMCT stands for Farahmand Modified Caughey Thomas. This model [53] was the result of fitting a Caughey Thomas like model to Monte Carlo data. The model is similar to the analytic model described by Equations 3-161 and 3-162 in Chapter 3: "Physics". This modified model is described by Equations 5-129 and 5-130 for electrons and holes.

$$\mu_n(T,N) = \mathrm{MU1N.FMCT}\left(\frac{T}{300}\right)^{\mathrm{BETAN.FMCT}} + \frac{(\mathrm{MU2N.FMCT}-\mathrm{MU1N.FMCT})\left(\frac{T}{300}\right)^{\mathrm{DELTAN.FMCT}}}{1+\left[\frac{N}{\mathrm{NCRITN.FMCT}\left(\frac{T}{300}\right)^{\mathrm{GAMMAN.FMCT}}}\right]^{\mathrm{ALPHAN.FMCT}(T/300)^{\mathrm{EPSP\cdot FMCT}}}} \quad \text{5-129}$$

$$\mu_p(T,N) = \mathrm{MU1P.FMCT}\left(\frac{T}{300}\right)^{\mathrm{BETAP.FMCT}} + \frac{(\mathrm{MU2P.FMCT}-\mathrm{MU1P.FMCT})\left(\frac{T}{300}\right)^{\mathrm{DELTAP.FMCT}}}{1+\left[\frac{N}{\mathrm{NCRITP.FMCT}\left(\frac{T}{300}\right)^{\mathrm{GAMMAP.FMCT}}}\right]^{\mathrm{ALPHAP.FMCT}(T/300)^{\mathrm{EPSP\cdot FMCT}}}} \quad \text{5-130}$$

In these equations, T is the lattice temperature and N is the total doping. Table 5-16 shows the user-definable parameters.

Table 5-16. User-specifiable parameters for the Faramand modified Caughey Thomas model for Nitrides.			
Parameter	**Statement**	**Type**	**Units**
MU1N.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU1P.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU2N.FMCT	MOBILITY	Real	$cm^2/(V*s)$
MU2P.FMCT	MOBILITY	Real	$cm^2/(V*s)$
ALPHAN.FMCT	MOBILITY	Real	
ALPHAP.FMCT	MOBILITY	Real	
BETAN.FMCT	MOBILITY	Real	
BETAP.FMCT	MOBILITY	Real	
GAMMAN.FMCT	MOBILITY	Real	
GAMMAP.FMCT	MOBILITY	Real	
DELTAN.FMCT	MOBILITY	Real	
DELTAP.FMCT	MOBILITY	Real	
EPSN.FMCT	MOBILITY	Real	
EPSP.FMCT	MOBILITY	Real	
NCRITN.FMCT	MOBILITY	Real	cm^{-3}
NCRITP.FMCT	MOBILITY	Real	cm^{-3}

Tables 5-17a and 5-16b show the default parameters as taken from the Monte Carlo fits for various nitride compositions.

Table 5-17a. Default Nitride Low Field Mobility Model Parameter Values [53]				
MATERIAL	MU1N.FMCT ($cm^2/V.s$)	MU2N.FMCT ($cm^2/V.s$)	ALPHAN.FMCT	BETAN.FMCT
InN	774	3138.4	0.68	-6.39
$In_{0.8}Ga_{0.2}N$	644.3	1252.7	0.82	-1.30
$In_{0.5}Ga_{0.5}N$	456.4	758.1	1.04	-1.74
$In_{0.2}Ga_{0.8}N$	386.4	684.2	1.37	-1.95
GaN	295.0	1460.7	0.66	-3.84
$Al_{0.2}Ga_{0.8}N$	132.0	306.1	0.29	-1.75

Table 5-17a. Default Nitride Low Field Mobility Model Parameter Values [53]				
MATERIAL	MU1N.FMCT ($cm^2/V.s$)	MU2N.FMCT ($cm^2/V.s$)	ALPHAN.FMCT	BETAN.FMCT
$Al_{0.5}Ga_{0.5}N$	41.7	208.3	0.12	-2.08
$Al_{0.8}Ga_{0.2}N$	47.8	199.6	0.17	-2.04
AlN	297.8	683.8	1.16	-1.82

Table 5-16b. Default Nitride Low Field Mobility Model Parameter Values [53]				
MATERIAL	GAMMAN.FMCT	DELTAN.FMCT	EPSN.FMCT	NLRITN.FMCT
InN	-1.81	8.05	-0.94	10^{17}
$In_{0.8}Ga_{0.2}N$	-1.30	4.84	-0.41	10^{17}
$In_{0.5}Ga_{0.5}N$	-1.74	2.21	-0.22	10^{17}
$In_{0.2}Ga_{0.8}N$	-1.95	2.12	-0.99	10^{17}
GaN	-3.84	3.02	0.81	10^{17}
$Al_{0.2}Ga_{0.8}N$	-1.75	6.02	1.44	10^{17}
$Al_{0.5}Ga_{0.5}N$	-2.08	10.45	2.00	10^{17}
$Al_{0.8}Ga_{0.2}N$	-2.04	20.65	0.01	10^{17}
AlN	-3.43	3.78	0.86	10^{17}

For composition fractions not listed in Tables 5-17a and 5-16b, the default parameters are linearly interpolated from the nearest composition fractions on the table. You can override these defaults by specifying any of the parameters listed in Table 5-16 on the MOBILITY statement. Currently, this model has only been calibrated for electrons. We only recommend that you use this model for holes when you define a set of real default parameters.

High Field Mobility

You can select nitride specific field dependent mobility model by specifying GANSAT.N and GANSAT.P on the MOBILITY statement. This model [53] is based on a fit to Monte Carlo data for bulk nitride, which is described in Equations 5-131 and 5-132.

$$\mu_n = \frac{\mu_{n0}(T,N) + \mathrm{VSATN}\dfrac{E^{\mathrm{N1N}\cdot\mathrm{GANSAT}} - 1}{\mathrm{ECN}\cdot\mathrm{GANSAT}^{\mathrm{N1N}\cdot\mathrm{GANSAT}}}}{1 + \mathrm{ANN}\cdot\mathrm{GANSAT}\left(\dfrac{E}{\mathrm{ECN}\cdot\mathrm{GANSAT}}\right)^{\mathrm{N2N}\cdot\mathrm{GANSAT}} + \left(\dfrac{E}{\mathrm{ECN}\cdot\mathrm{GANSAT}}\right)^{\mathrm{N1N}\cdot\mathrm{GANSAT}}} \quad \text{5-131}$$

$$\mu_p = \frac{\mu_{p0}(T,N) + \mathrm{VSATP}\dfrac{E^{\mathrm{N1P\cdot GANSAT}-1}}{\mathrm{ECP\cdot GANSAT}^{\mathrm{N1P\cdot GANSAT}}}}{1 + \mathrm{ANP\cdot GANSAT}\left(\dfrac{E}{\mathrm{ECP\cdot GANSAT}}\right)^{\mathrm{N2P\cdot GANSAT}} + \left(\dfrac{E}{\mathrm{ECP\cdot GANSAT}}\right)^{\mathrm{N1P\cdot GANSAT}}} \quad 5\text{-}132$$

In these equations, $\mu_0(T,N)$ is the low field mobility and E is the electric field. Table 5-17 lists the user-definable parameters.

Table 5-17. User Definable Low Field Nitride Mobility Model Parameters

Parameter	Statement	Type	Units
N1P.GANSAT	MOBILITY	Real	
N1P.GANSAT	MOBILITY	Real	
N2N.GANSAT	MOBILITY	Real	
N2P.GANSAT	MOBILITY	Real	
ANN.GANSAT	MOBILITY	Real	
ANP.GANSAT	MOBILITY	Real	
ECN.GANSAT	MOBILITY	Real	V/cm
ECP.GANSAT	MOBILITY	Real	V/cm

Table 5-18 shows the default parameters as taken from the Monte Carlo fits for various nitride compositions.

Table 5-18. Default Nitride Field Dependent Mobility Model Parameter Values [53]

MATERIAL	VSATN (10^7cm/s)	ECN.GANSAT	N1N.GANSAT	N2N.GANSAT	ANN.GANSAT
InN	1.3595	52.4242	3.8501	0.6078	2.2623
$In_{0.8}Ga_{0.2}N$	0.8714	103.4550	4.2379	1.1227	3.0295
$In_{0.5}Ga_{0.5}N$	0.7973	148.9098	4.0635	1.0849	3.0052
$In_{0.2}Ga_{0.8}N$	1.0428	207.5922	4.7193	1.0239	3.6204
GaN	1.9064	220.8936	7.2044	0.7857	6.1973
$Al_{0.2}Ga_{0.8}N$	1.1219	365.5529	5.3193	1.0396	3.2332
$Al_{0.5}Ga_{0.5}N$	1.1459	455.4437	5.0264	1.0016	2.6055
$Al_{0.8}Ga_{0.2}N$	1.5804	428.1290	7.8166	1.0196	2.4359
AlN	2.1670	447.0339	17.3681	0.8554	8.7253

For composition fractions not listed in Table 5-18, the default parameters are linearly interpolated from the nearest composition fractions on the table.

You can override these defaults by specifying any of the parameters listed in Table 5-17 in the MOBILITY statement. Currently, this model has only been calibrated for electrons. We only recommend that you use this model for holes when you define a set of real default parameters. Also note that these models exhibit negative differential mobility and may exhibit poor convergence. In such cases, you may do well to use the simpler model given in Equations 3-260 and 3-261 with a reasonable value of saturation velocity.

Impact Ionization Parameters

Table 5-19 shows the extracted default values for the Selberherr impact ionization model from [114] for GaN.

Table 5-19. Default Impact Ionization Parameters (IMPACT statement)

Parameter	Units	GaN	InN	AlN	InGaN	AlGaN
AN1	cm^{-1}	2.52e8	2.52e8	2.52e8	2.52e8	2.52e8
AN2	cm^{-1}	2.52e8	2.52e8	2.52e8	2.52e8	2.52e8
BN1	V/cm	3.41e7	3.41e7	3.41e7	3.41e7	3.41e7
BN2	V/cm	3.41e7	3.41e7	3.41e7	3.41e7	3.41e7
AP1	cm^{-1}	3.57e6	3.57e6	3.57e6	3.57e6	3.57e6
AP2	cm^{-1}	3.57e6	3.57e6	3.57e6	3.57e6	3.57e6
BP1	V/cm	1.96e7	1.96e7	1.96e7	1.96e7	1.96e7
BP2	V/cm	1.96e7	1.96e7	1.96e7	1.96e7	1.96e7
BETAN		1.0	1.0	1.0	1.0	1.0
BETAP		1.0	1.0	1.0	1.0	1.0
EGRAN	V/cm	0.0	0.0	0.0	0.0	0.0

Recombination Parameters

Table 5-20 shows the default values for radiative rates for the the binary wurtzite nitride compounds.

Table 5-20. Default Radiative Rates for Nitrides (MATERIAL COPT)

Material	COPT	Units	Reference
GaN	1.1e-8	cm^3/s	[111]
InN	2.0e-10	cm^3/s	[193]
AlN	0.4e-10	cm^3/s	[172]

Default Models for Heat Capacity and Themal Conductivity

By default, the heat capacity and thermal conductivities used in self-consistent heat flow simulations (GIGA and GIGA3D) for the GaN/AlGaN/InGaN system are given by the following equations [122].

$$C_L(T) = C_L(300K)\frac{20-(\Theta_D/T)^2}{20-(\Theta_D/300K)^2}P_L \quad 5\text{-}133$$

$$\kappa_L(T) = \kappa_L(300K)\left(\frac{T}{300K}\right)^{\delta_\kappa} \quad 5\text{-}134$$

Here, T is the local temperature and the other parameters are given in Table 5-21 [122].

Table 5-21. Default Heat Capacity and Thermal Conductivity Parameters for GaN

Parameter Unit	κ_L (W/Kcm)	C_L (Ws/gK)	ρ_L (g/cm^3)	Θ_D (K)	δ_κ
GaN	1.30	0.49	6.15	600	-0.28
AlN	2.85	0.6	3.23	1150	-1.64
InN	0.45	0.32	6.81	660	0.0

Values for Ternary composition are obtained by linear interpolation as a function of composition.

Adachi's Refractive Index Model [122]

For the III-V nitride compounds, Adachi's refractive index model is expressed by Equation 5-135.

$$n_r(\omega) = \sqrt{A\left(\frac{\hbar\omega}{E_g}\right)^{-2}\left\{2-\sqrt{1+\frac{h\omega}{E_g}}-\sqrt{1-\frac{h\omega}{E_g}}\right\}+B} \quad 5\text{-}135$$

where E_g is the bandgap, ω is the optical frequency and A and B are material composition dependent parameters. For $Al_xGa_{1-x}N$, the compositional dependence of the A and B parameters are given by the expressions in Equations 5-118 and 5-119.

$$A(x) = 9.827 - 8.216X - 31.59X^2 \quad 5\text{-}136$$

$$B(x) = 2.736 + 0.842X - 6.293X^2 \quad 5\text{-}137$$

For $In_xGa_{1-x}N$, the compositional dependence of the A and B parameters are given by the expressions in Equations 5-138 and 5-139.

$$A(x) = 9.827_{(1-X)} - 53.57X \quad 5\text{-}138$$

$$B(x) = 2.736(1-X) - 9.19X. \quad 5\text{-}139$$

5.3.8: The Hg(1-x)Cd(x)Te System

The following sections describe the relationship between mole fraction, X.COMP, and the material parameters of the of the Hg(1-x)Cd(x)Te system. This data has been taken from [176].

Bandgap

Equation 5-140 is used in the calculation of bandgap as a function of composition fraction.

$$Eg = -0.302 + 1.93*\mathrm{X.COMP} - 0.810*\mathrm{X.COMP}^2 + 0.832*\mathrm{X.COMP}^3 + 5.354 \times 10^{-4} * \left(1 - 2*\mathrm{X.COMP}\right) T_L \quad (5\text{-}140)$$

Here, T_L is lattice temperature.

Electron Affinity

Equation 5-141 is used in the calculation of electron affinity as a function of composition fraction.

$$\chi = 4.23 - 0.813*(EG300 - 0.0083) \quad (5\text{-}141)$$

Permittivity

Equation 5-142 is used in the calculation of relative permittivity as a function of composition fraction.

$$\varepsilon = 20.5 - 15.5*\mathrm{X.COMP} + 5.7*\mathrm{X.COMP}^2 \quad (5\text{-}142)$$

Density of States Masses

Equations 5-143 and 5-144 are used to calculate the density of states effective masses of electrons and holes.

$$\frac{me}{m_0} = \left[-0.6 + 6.333*\left(\frac{2}{EG300} + \frac{1}{EG300 + 1}\right)\right]^{-1} \quad (5\text{-}143)$$

$$\frac{me}{m_0} = 0.55 \quad (5\text{-}144)$$

Mobility

Equations 5-145 and 5-146 are used to calculate the electron and hole mobilities as functions of composition fraction.

$$\mu_e = 9 \times 10^8 \left(\frac{0.2}{\mathrm{X.COMP}}\right)^{7.5} T_L^{-2\left(\frac{0.2}{\mathrm{X.COMP}}\right)^{0.6}} \quad (5\text{-}145)$$

$$\mu_h = 0.01*\mu e \quad (5\text{-}146)$$

5.4: Simulating Heterojunction Devices with Blaze

5.4.1: Defining Material Regions with Positionally-Dependent Band Structure

Step Junctions

The easiest way to define a device with positionally dependent band structure is to specify two adjacent semiconductor regions with dissimilar bandgap. In this case, BLAZE would simulate an abrupt heterojunction between the two materials.

For example, you want to simulate an abrupt heterojunction parallel to the x-axis at a location of y=0.1 microns. For values of y greater than 0.1 specify, GaAs. For values of y less than 0.1, specify AlGaAs with a composition fraction of 0.3. The following statements would specify this situation.

```
REGION Y.MIN=0.1 MATERIAL=GaAs

REGION Y.MAx=0.1 MATERIAL=AlGaAs x.COMP=0.3
```

This fragment specifies that the two regions form an abrupt heterojunction at Y=0.1. The first region is composed of GaAs while the second is composed of AlGaAs.

These two material names are used by BLAZE to choose default material models for the two regions. For a complete list of the materials available in ATLAS/BLAZE, see Appendix B: "Material Systems". For the AlGaAs region a composition fraction of 0.3 is specified.

Graded Junctions

A grading can be applied to this heterojunction with a simple modification. For example:

```
REGION Y.MIN=0.1 MATERIAL=GaAs

REGION Y.MAx=0.1 MATERIAL=AlGaAs x.COMP=0.3 GRAD.34=0.01
```

specifies that the composition fraction of the AlGaAs region decreases from 0.3 at y=0.1 microns to 0.0 at y=0.11 microns. The GRAD parameter specifies the distance over which the mole fraction reduces to zero. The GRAD parameter is indexed such that GRAD.12 corresponds to the Y.MIN side of the region, GRAD.23 corresponds to the x.MAX side of the region, GRAD.34 corresponds to the Y.MAX side of the region, and GRAD.41 corresponds to the x.MIN side of the region. In most cases, the GRAD.n parameter acts to increase the size of the region. By default, the GRAD.n parameters are set to zero and all heterojunctions are abrupt. Note that the GRAD parameter acts just like the other region geometry parameters in that later defined regions overlapping the graded part of the region will overlap the grading. If in the previous example the grading had been applied to the GaAs region, it would be over lapped by the AlGaAs region. This would have produced an abrupt interface. A solution would be to limit Y.MAX in the AlGaAs region to 0.09. Make sure you specify regions in the proper order to avoid such problems.

Along similar lines, you can use the overlapping of regions to an advantage in forming graded heterojunctions between two materials in the same system with different non-zero composition fractions. For example:

```
REGION Y.MIN=0.1 MATERIAL=AlGaAs x.COMP=0.3 GRAD.12=0.02

REGION Y.MAx=0.11 MATERIAL=AlGaAs x.COMP=0.1
```

specifies a graded heterojunction with a composition of 0.3 at y = 0.1 falling to 0.1 at y = 0.11.

5.4.2: Defining Materials and Models

Materials

For example to set the bandgap for the material, InP, use the following syntax.

```
MATERIAL MATERIAL=InP EG300=1.35
```

Models

BLAZE has two ways of simulating the physical effects of variations in semiconductor composition. For relatively gradual variations in composition, the standard modifications to the drift-diffusion equations can be considered adequate for simulation purposes. For abrupt heterojunctions, it has been suggested that thermionic emission may be the dominant factor in the behavior of heterojunction behavior.

Individual material parameters and models can be defined for each material or region. These models are set in the MATERIAL, MODEL, and IMPACT statements.

This statement uses the MATERIAL parameter to select all regions composed of the material "InP". The bandgap in these regions will be set to 1.35. There are two ways to set the parameters of a particular region. The first is to use the region index. For example:

```
MODEL REGION=1 BGN
```

In this case, the band gap narrowing model is enabled in the region indexed number 1.

The second is to use the region name. For example:

```
IMPACT NAME=substrate SELB
```

This example turns on the Selberherr Impact Ionization Model in the region named substrate. You can then set the parameters for all regions and materials by omitting the MATERIAL, REGION, or NAME parameters, as in the following:

```
MODEL BGN
```

This statement sets the bandgap narrowing model for all regions and materials

Parser Functions

To use the C-INTERPRETER functions, you need to know the C programming language. See Appendix A: “C-Interpreter Functions” for a description of the parser functions.

To specify a completely arbitrary spatial variation of varying composition fraction, use a parser function. To define the parser function for composition fraction, write a C function describing the composition fraction as a function of position. A template for the function called COMPOSITION is provided with this release of ATLAS. Once you define the COMPOSITION function, store it in a file. To use the function for composition, set the F.COMPOSIT parameter to the file name of the function.

This page is intentionally left blank.

Chapter 6:
3D Device Simulator

6.1: 3D Device Simulation Programs

This chapter aims to highlight the extra information required for 3-D simulation as compared to 2-D. You should be familiar with the equivalent 2-D models before reading this chapter.

This chapter describes the set of ATLAS products that extends 2D simulation models and techniques and applies them to general non-planar 3D structures. The structural definition, models and material parameters settings and solution techniques are similar to 2D. You should be familiar with the simulation techniques described in Chapter 2: "Getting Started with ATLAS" and the equivalent 2D product chapters before reading the sections that follow. The products that form 3D device simulation in ATLAS are:

- DEVICE3D – silicon compound material and heterojunction simulation
- GIGA3D – non-isothermal simulation
- MIXEDMODE3D – mixed device-circuit simulation
- TFT3D – thin film transistor simulation
- QUANTUM3D – quantum effects simulation
- LUMINOUS3D – photodetection simulation

The 3D modules, THERMAL3D, are described in Chapter 17: "Thermal 3D: Thermal Packaging Simulator".

In a similar manner to the 2D products, GIGA3D, MIXEDMODE3D, LUMINOUS3D and QUANTUM3D should be combined with both DEVICE3D or BLAZE3D depending on the semiconductor materials used.

6.1.1: DEVICE3D

DEVICE3D provides semiconductor device in 3D. Its use is analogous to the 2-D simulations in S-PISCES and BLAZE. See Chapter 4: "S-Pisces: Silicon Based 2D Simulator" for more information on S-PISCES. See Chapter 5: "Blaze: Compound Material 2D Simulator" for more information on BLAZE.

6.1.2: GIGA3D

GIGA3D is an extension of DEVICE3D that accounts for lattice heat flow in 3D devices. GIGA3D has all the functionality of GIGA (see Chapter 7: "Giga: Self-Heating Simulator") with a few exceptions. One exception is that additional syntax has been added to account for the three dimensional nature of thermal contacts. You can specify the `Z.MIN` and `Z.MAX` parameters on the `THERMCONTACT` statement to describe the extent of the contact in the Z direction. Another exception is that there is no `BLOCK` method available in the 3D version.

6.1.3: TFT3D

TFT3D is an extension of DEVICE3D that allows you to simulate amorphous and polycrystalline semiconductor materials in three dimensions. TFT3D is completely analogous to the TFT simulator described in Chapter 14: "TFT: Thin-Film Transistor Simulator". The complete functionality of the TFT simulator is available in TFT3D for three dimensional devices.

6.1.4: MIXEDMODE3D

MIXEDMODE3D is an extension of DEVICE3D or BLAZE3D that allows you to simulate physical devices embedded in lumped element circuits (SPICE circuits). MIXEDMODE3D is completely analogous to the MIXEDMODE simulator, which is described in Chapter 12: "MixedMode: Mixed Circuit and Device Simulator". The complete functionality of the MIXEDMODE simulator is available in MIXEDMODE3D for three dimensional devices.

6.1.5: QUANTUM3D

QUANTUM3D is an extension of DEVICE3D or BLAZE3D, which allows you to simulate of the effects of quantum confinement using the Quantum Transport Model (Quantum Moments Model). QUANTUM3D is completely analogous to the QUANTUM model but applies to three dimensional devices. See Chapter 13: "Quantum: Quantum Effect Simulator" for more information on QUANTUM.

6.1.6: LUMINOUS3D

LUMINOUS3D is an extension of DEVICE3D that allows you to simulate photodetection in three dimensions. LUMINOUS3D is analogous to the LUMINOUS simulator, which is described in Chapter 10: "Luminous: Optoelectronic Simulator" with a few significant differences described in Section 6.3.11: "LUMINOUS3D Models".

6.2: 3D Structure Generation

All 3-D programs in ATLAS supports structures defined on 3D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension.

There are two methods for creating a 3D structure that can be used with ATLAS. One way is through the command syntax of ATLAS. Another way is through an interface to DEVEDIT3D.

A direct interface from ATHENA to 3D ATLAS impossible. But DEVEDIT3D provides the ability to read in 2D structures from ATHENA and extend them non-uniformly to create 3D structures for ATLAS.

ATLAS Syntax For 3D Structure Generation

Mesh generation

Chapter 2: "Getting Started with ATLAS", Section 2.6.3: "Using The Command Language To Define A Structure" covers the generation of 2D and 3D mesh structures using the ATLAS command language. The `Z.MESH` statement and the `NZ` and `THREE.D` parameters of the `MESH` statement are required to extend a 2D mesh into 3D.

Conventionally, slices are made perpendicular to the Z axis. The mesh is triangular in XY but rectangular in XZ or YZ planes.

Region, Electrode, and Doping definition

Chapter 2: "Getting Started with ATLAS", Section 2.6.3: "Using The Command Language To Define A Structure" also covers the definition of 2D regions, electrodes and doping profiles. To extend the regions into 3D, use the `Z.MIN` and `Z.MAX` parameters. For example:

```
REGION NUM=2 MATERIAL=Silicon X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1 Z.MIN=0
Z.MAX=1
ELECTRODE NAME=gate X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1 Z.MIN=0 Z.MAX=1
DOPING GAUSS N.TYPE CONC=1E20 JUNC=0.2 Z.MIN=0.0 Z.MAX=1.0
```

For 2D regions or electrodes defined with the command language, geometry is limited to rectangular shapes. Similarly, in 3D regions and electrodes are composed of rectangular parallelopipeds.

DevEdit3D Interface

DEVEDIT3D is a graphical tool that allows you to draw 3D device structures and create 3D meshes. It can also read 2D structures from ATHENA and extend them into 3D. These structures can be saved from DEVEDIT3D as structure files for ATLAS. Also, save a command file when using DEVEDIT3D. This file is used to recreate the 3D structure inside DEVEDIT3D, which is important, since DEVEDIT3D doesn't read in 3D structure files.

ATLAS can read structures generated by DEVEDIT3D using the command:

```
MESH INF=<filename>
```

The program is able to distinguish automatically between 2D and 3D meshes read in using this command.

Defining Devices with Circular Masks

DEVEDIT3D makes a triangular mesh in the XY plane and uses z-plane slices. This means, that normally the Y direction is vertically down into the substrate. But in the case of using circular masks, you need to rotate the device.

With defining devices using circular masks in DEVEDIT3D, the XY plane should be the surface of the device and the Z direction should be into the substrate.

6.3: Model And Material Parameter Selection in 3D

Models and material parameters are chosen in 3-D in common with other 2-D modules using the MODELS, IMPACT, MATERIAL, MOBILITY, INTERFACE, and CONTACT statements.

The following models are available in 3D device simulation programs. All of these models are doucmented in Chapter 3: "Physics" or in the 2D product chapters.

6.3.1: Mobility

- Table for 300K (CONMOB)
- Thomas (ANALYTIC)
- Arora's Model (ARORA)
- Klaassen's Model (KLAASSEN)
- Lombardi's Model (CVT)
- Yamaguchi Model (YAMA)
- Parallel Field Dependence (FLDMOB)
- Parallel Field Dependence with negative differential mobility (FLDMOB EVSATMOD=1)

6.3.2: Recombination

- Shockley Read Hall (SRH)
- Concentration dependent lifetime SRH (CONSRH)
- Klaassen's concentration dependent lifetime SRH (KLASRH)
- Auger (AUGER)
- Klaassen's concentration dependent Auger recombination model (KLAAUG)
- Optical Recombination (OPTR)
- Bulk and interface traps (TRAP, INTTRAP)
- Continuous defect states (DEFECT)

6.3.3: Generation

- Selberherr Impact Ionization (IMPACT SELB)
- Crowell Impact Ionization (IMPACT CROWELL)
- Hot Electron Injection (HEI)
- Fowler Nordheim Tunneling (FNORD)
- Single Event Upset (SINGLEEVENTUPSET)

6.3.4: Carrier Statistics

- Boltzmann (default)
- Fermi (FERMI)
- Band Gap Narrowing (BGN)
- Incomplete Ionization (INCOMPLETE)
- Quantum Mechanical Effects (QUANTUM)

6.3.5: Boundary Conditions

- Ohmic and Schottky
- Current Boundary Conditions
- Lumped Element Boundary Conditions
- Distributed Contact Resistance

6.3.6: Optical

• Photogeneration with Ray Tracing (LUMINOUS3D).

6.3.7: Single Event Upset Simulation

• Single Event Upset Simulation.

All these models, with the exception of SINGLEEVENTUPSET, are documented in the Chapter 3: "Physics" or in the 2D product chapters of this manual.

6.3.8: Boundary Conditions in 3D

External Passive Elements

You can attach external lumped resistors, capacitors and inductors to any contact. The syntax is the same as for the 2D products, which is:

```
CONTACT NAME=drain RES=1e3 CAP=1e-12 L=1e-6
```

You can also apply distributed resistances to contacts. The algorithm used for estimating contact area for 3D distributed contact resistance multiplies the contact perimeter in a given Z plane by the displacement in the Z direction. This algorithm will only work properly for planar contacts that do not vary in the Z direction. They may however abruptly terminate or start in the Z direction.

The units of lumped external passive elements are ohms for resistors, Farads for capacitors and Henrys for inductors. Distributed contact resistance is defined in ohms.cm^3.

Thermal Contacts for GIGA3D

Thermal contacts for non-isothermal simulation in GIGA3D are defined in an analogous manner to the 2D thermal contacts in GIGA. The Z.MIN and Z.MAX parameters are used to define the extent of the thermal contact in the z-plane. The units of the thermal resistance parameter ALPHA are scaled in 3D to W/(cm.K). For more information about GIGA3D, see Chapter 7: "Giga: Self-Heating Simulator".

6.3.9: TFT3D Models

Models for simulating thin-film transistors made from amorphous or polycrystalline semiconductors are supported in TFT3D. The definition of the continuous defect states in the bandgap is performed using the same parameters as in 2D simulations with TFT. The models for continuous defect (or trap) densities are documented in Chapter 14: "TFT: Thin-Film Transistor Simulator".

6.3.10: QUANTUM3D Models

Models for simulating quantum effects semiconductors are supported in QUANTUM3D. The definition of the quantum moments solver is the same as in 2D simulations with QUANTUM. The models for simulating quantum effects and the parameters to control the model are shown in Chapter 13: "Quantum: Quantum Effect Simulator".

6.3.11: LUMINOUS3D Models

Many of the models for simulating photodetection in LUMINOUS3D are similar to those for simulating photodetection in LUMINOUS. For more information about LUMINOUS, see Chapter 10: "Luminous: Optoelectronic Simulator". This section, however, shows several important differences between LUMINOUS and LUMINOUS3D.

Optical Source Specification

The specification of the optical source in 3D is similar to the specification of the source described in Chapter 10: "Luminous: Optoelectronic Simulator", Section 10.4.1: "Defining Optical Sources" with a few additions to account for the third dimension.

Figures 6-1 and 6-2 show the complete specification of the optical source in 3D includes the specification of the three coordinates of the source origin using the X.ORIGIN, Y.ORIGIN, and Z.ORIGIN parameters of the BEAM statement as well as two angles of rotation using the THETA and PHI parameters. The PHI parameter is analogous to the LUMINOUS parameter, ANGLE, and specifies the direction of propagation relative to the device x axis (see Figure 6-1). In face, PHI and ANGLE are synonymous. The THETA parameter specifies the rotated angle of propagation relative to the x-y plane (see Figure 6-2).

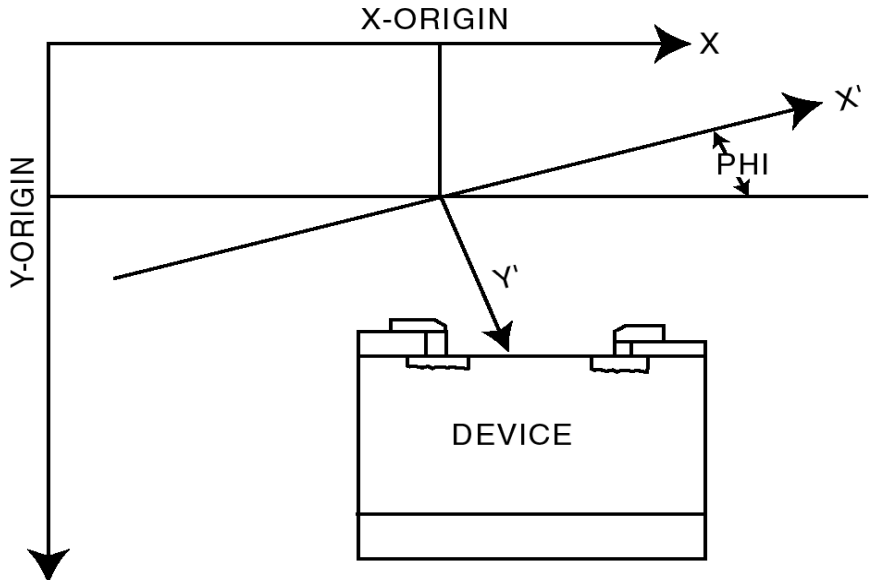


Figure 6-1: Source beam coordinate rotation around Z-axis

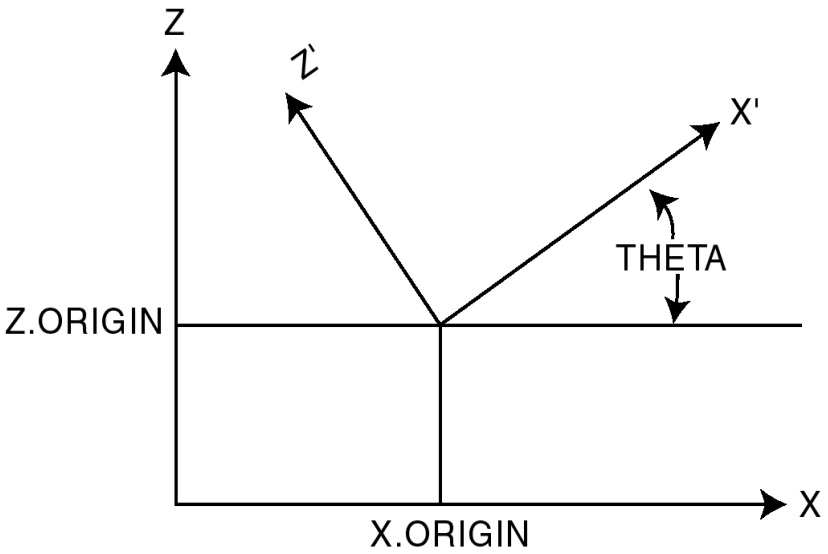


Figure 6-2: Source beam coordinate rotation around Y-axis

Ray tracing

The discrete sampling of the source beam into rays in LUMINOUS3D is unlike that done in LUMINOUS. In LUMINOUS, the source beam is automatically broken up into a set of rays that resolve the device topology and variations in the interior of the device. In LUMINOUS3D, this process is more complex and is a computational burden. As such in LUMINOUS3D, specify a discrete sampling of the source beam (see Figure 6-3).

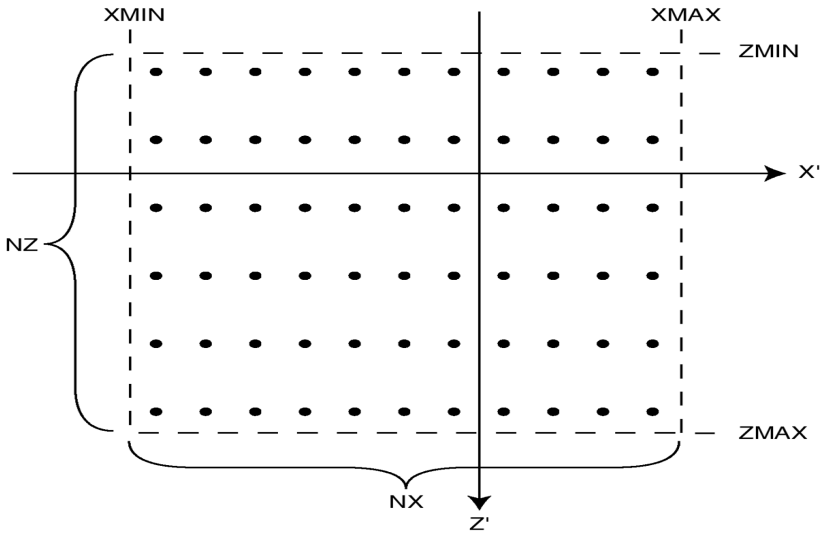


Figure 6-3: Source beam sampling

In Figure 6-3, the extent of the source sampling is specified by the XMIN, XMAX, ZMIN and ZMAX parameters of the BEAM statement. Even samples are taken along each of the beam front principal axes. The number of samples in the x and z directions are given by the NX and NZ parameters of the BEAM statement.

Lenslet specification

Another significant difference between LUMINOUS and LUMINOUS3D is that in 3-D you can specify a virtual lenslet (see Figure 6-4). Due to the restriction of prismatic elements in 3D, you can't accurately specify focusing elements using the device mesh.

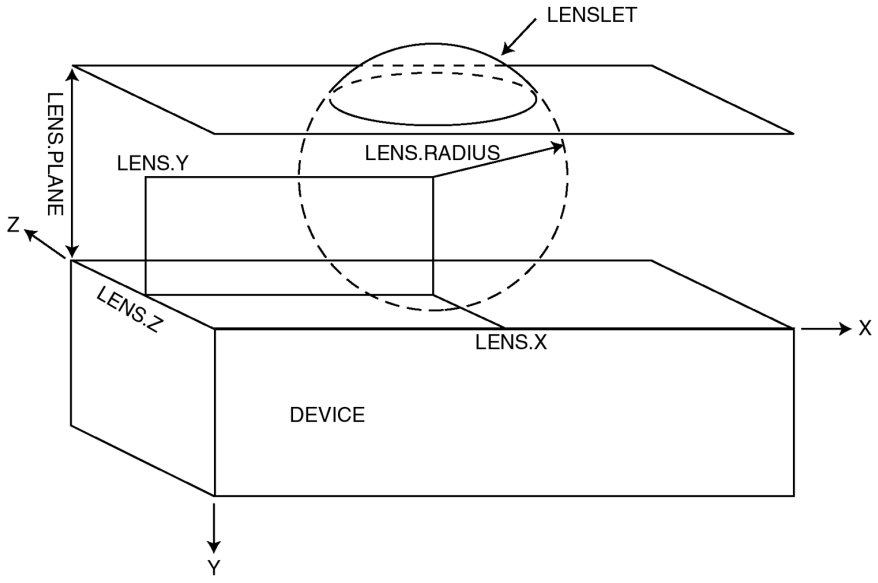


Figure 6-4: Luminous3D Lenslet Specification

As shown in Figure 6-4, the lenslet is represented by a spherical section atop a plane. The center of the sphere is specified by the LENS.X, LENS.Y and LENS.Z parameters of the BEAM statement. The radius of the sphere is specified by the LENS.RADIUS parameter of the BEAM statement. The location of the planar part surrounding the lenslet relative to the y=0 plane is specified by the LENS.PLANE parameter. The index of refraction of the lenslet as well as the volume between the lens plane and the device surface is specified by the LENS.INDEX parameter of the BEAM statement.

Note that the volume associated with the lenslet is not meshed or considered in the solution of the device equations. It is merely used in the ray tracing. Also, you should keep in mind that lenslet can only be specified in planes perpendicular to the y axis.

Other differences between Luminous and Luminous3D

LUMINOUS3D has several other differences with LUMINOUS. For example, in LUMINOUS the rays generated during ray tracing are stored in all subsequent structure files. In LUMINOUS3D, this isn't the case. To save the rays in LUMINOUS3D, set the RAYTRACE parameter of the BEAM statement to the name of a file where the results of the ray trace are to be stored.

Another difference is that LUMINOUS3D has an option to cause all metal regions to act as perfect reflectors. To enable this option, specify METAL.REFLECT in the BEAM statement.

LUMINOUS3D also allows you to simulate periodicity with respect to ray tracing by specifying PERIODIC on the BEAM statement. This causes rays exiting the sides of the device (perpendicular to the x-z plane) to re-enter the opposite side of the device.

6.4: Numerical Methods for 3D

6.4.1: DC Solutions

There are several differences between the default numerical methods applied in 2D ATLAS and those applied in 3D ATLAS. For example, with respect to non-linear iteration strategies, the current version of the 3D simulator does not support the BLOCK method. The NEWTON and GUMMEL iteration strategies are supported for 3D simulations, whereas NEWTON, GUMMEL and BLOCK are all supported for 2D simulations.

In solving the linear subproblem, the default approach in 3D is to use an iterative solver. This is believed to be the most efficient method for general 3D problems. In 2D, the direct solver is used by default. You may find it desirable to use direct methods in 3D problems due to improved convergence of computational efficiency. You can select the direct method by specifying DIRECT in the METHOD statement.

Also, in 3D there are two linear iterative solution methods available. The defaults are ILUCGS (incomplete lower-upper decomposition conjugate gradient system) and BICGST (bi-conjugate gradient stabilized). Historically, tests have shown that the current implementation of ILUCGS is slightly more stable than BICGST and is the default iterative solver in 3D. You can define the BICGST solver by specifying BICGST in the METHOD statement.

Note: Iterative solvers are recommended for large problems, typically greater than 5000 node points, due to lower solution times and memory usage.

6.4.2: Transient Solutions

In transient mode, a semi-implicit scheme is used in addition to the default TR-BDF algorithm [3]. This algorithm is recommended for complex simulations such as Single Event Upset. To select this method, use:

```
METHOD HALFIMPL
```

6.4.3: Obtaining Solutions In 3D

ATLAS3D programs can perform DC and transient analysis in an equivalent manner to 2D. The SOLVE statement is used to define the solution procedure. The syntax used is described in Chapter 2: “Getting Started with ATLAS”, Section 2.9: “Obtaining Solutions”.

6.4.4: Interpreting the Results From 3D

The log files produced by 3D ATLAS can be plotted in TONYPLOT exactly as those that result from S-PISCES or BLAZE. The only difference is the units of the currents produced. Log files from 3D simulations save current in Amperes, whereas the 2D simulations use Amperes/micron.

The solution files produced by 3D ATLAS should be plotted using TONYPLOT3D. These files cannot be read directly into the 2D TONYPLOT program. TONYPLOT3D contains features that allows you to make slices of the 3D structure, which can be plotted in 2D TONYPLOT. For more information about TONYPLOT3D, see the TONYPLOT3D USER’S MANUAL.

6.4.5: More Information

Many examples using 3D ATLAS have been installed on your distribution tape or CD. You can find more information about 3D ATLAS by reading the text associated with each example. You can also find find some more information at www.silvaco.com.

This page intentionally left blank.

Chapter 7:
Giga: Self-Heating Simulator

7.1: Overview

GIGA extends ATLAS to account for lattice heat flow and general thermal environments. GIGA implements Wachutka's thermodynamically rigorous model of lattice heating [170], which accounts for Joule heating, heating, and cooling due to carrier generation and recombination, and the Peltier and Thomson effects. GIGA accounts for the dependence of material and transport parameters on the lattice temperature. GIGA also supports the specification of general thermal environments using a combination of realistic heat-sink structures, thermal impedances, and specified ambient temperatures. GIGA works with both S-PISCES and BLAZE and with both the drift-diffusion and energy balance transport models. See Chapter 3: "Physics", the "Drift-Diffusion Transport Model" section on page 3-2 and the "Energy Balance Transport Model" section on page 3-4 for more information on these models.

Before continuing with this chapter, you should be familiar with ATLAS. If not, read Chapter 2: "Getting Started with ATLAS", along with Chapter 4: "S-Pisces: Silicon Based 2D Simulator" or Chapter 5: "Blaze: Compound Material 2D Simulator".

7.1.1: Applications

A major application of GIGA is the simulation of high-power structures including bipolar, MOS, IGBT, and thyristor devices. Another important application is the simulation of electrostatic discharge (ESD) protection devices. Thermal effects are also important in SOI device operation due to the low thermal conductivity of the buried oxide, and in devices fabricated in III-V material systems due to the relatively low thermal conductivity of these materials.

Recent studies have demonstrated that accounting self-consistently for lattice heating is necessary for accurate simulation of bipolar VLSI devices. This is due to the sensitive temperature dependence of the carrier injection process. Since bipolar devices are key components of CMOS technologies and many devices can be impacted by parasitic bipolar effects, the applications of GIGA are very general.

For other more information on GIGA's application, see Section 7.3: "Applications of GIGA".

7.1.2: Numerics

GIGA supplies numerical techniques that provide efficient and robust solution of the complicated systems of equations that result when lattice heating is accounted for. These numerical techniques include fully-coupled and block iteration methods. When GIGA is used with the energy balance equations, the result is a "six equation solver", which defines the state-of-the-art for general purpose device simulation.

For more information on numerical techniques, see Chapter 18: "Numerical Techniques".

7.2: Physical Models

7.2.1: The Lattice Heat Flow Equation

GIGA adds the heat flow equation to the primary equations that are solved by ATLAS. The heat flow equation has the form:

$$C\frac{\partial T_L}{\partial t} = \nabla(\kappa\nabla T_L) + H \qquad 7\text{-}1$$

where:

- C is the heat capacitance per unit volume.
- κ is the thermal conductivity.
- H is the heat generation
- T_L is the local lattice temperature.

The heat capacitance can be expressed as $C = \rho C_p$, where C_p is the specific heat and ρ is the density of the material.

Specifying the LAT.TEMP parameter in the MODELS statement includes the lattice heat flow equation in ATLAS simulations.

GIGA supports different combinations of models. For example, if the HCTE and LAT.TEMP parameters are specified in the MODELS statement and both particle continuity equations are solved, all six equations are solved. If HCTE.EL is specified instead of HCTE, only five equations are solved and the hole temperature T_p is set equal to lattice temperature T_L.

Built-in Models for Thermal Conductivity and Heat Capacity

The key material dependent parameters in Equation 7-1 are the thermal conductivity and the heat capacity. In general, both thermal conductivity and heat capacity are both composition and temperature dependent. In later sections, we will show how you have a great deal of flexibility in specifying these model dependencies.

For most materials, we provide reasonable default values for these material parameters, which you can examine by specifying PRINT on the MODELS statement. These models allow user specification but are not material composition dependent. As such, we have provided default models for some of the more popular materials that are both composition and temperature dependent. These models described here apply to the following materials: Si, Ge, GaAs, AlAs, InAs, InP, GaP, SiGe, AlGaAs, InGaAs, InAlAs, InAsP, GaAsP and InGaP.

The default built-in model derives temperature dependency in a manner very similar to the user specifiable models discussed later in this section.

The composition depencencies of the ternary compounds and the binary, SiGe, are derived from interpolation from the binary compounds (or the values for Si and Ge in the case of SiGe).

The basic model for thermal conductivity is given by:

$$\kappa(T_L) = K_{300} \cdot \left(\frac{T_L}{300}\right)^{\alpha} \qquad 7\text{-}2$$

where $\kappa(T_L)$ is the temperature dependent thermal conductivity, T_L is the lattice temperature and K_{300} and α are matrial dependent parameters.

Table 7-1 shows the default values for K_{300} and α for Si, Ge and the binary III-V compounds .

Table 7-1. Default Parameter Values for Thermal Conductivity

Material	K_{300}(W/Kcm)	α
Si	1.48	-1.65
Ge	0.60	-1.25
GaAs	0.46	-1.25
AlAs	0.80	-1.37
InAs	0.273	-1.1
InP	0.68	-1.4
GaP	0.77	-1.4

The parameters K_{300} and α are interpolated as a function of composition fraction x using Equations 7-3 and 7-4.

$$K_{300}^{AB} = \frac{1}{\left(\frac{1-x}{\frac{A}{K_{300}}} + \frac{x}{\frac{B}{K_{300}}} + \frac{(1-x)x}{C}\right)} \quad 7\text{-}3$$

$$\alpha^{AB} = (1-x)\alpha^{A} + \alpha^{B} \quad 7\text{-}4$$

The C parameter in Equation 7-3 is a bowing factor used to account for the non-linear aspects of the variation of thermal conductivity with composition. Table 7-2 shows the default values of the bowing factor, C, for the various ternary compounds and SiGe.

Table 7-2. Default Bowing Parameter Values for Thermal Conductivity

Material	C(W/K cm)
SiGe	0.028
AlGaAs	0.033
InGaAs	0.014
InAlAs	0.033
InAsP	0.033
GaAsP	0.014
InGaP	0.014

The temperature dependence of heat capacity can be expressed by:

$$C(T_L) = \rho\left[C_{300} + C_1\frac{\left(\frac{T_L}{300}\right)^{\beta} - 1}{\left(\frac{T_L}{300}\right)^{\beta} + \frac{C_1}{300}}\right] \quad 7\text{-}5$$

where $C(T_L)$ is the temperature dependent heat capacity, ρ is the mass density, C_{300}, C_1 and β are material dependent parameters. Table 7-3 shows the default values of ρ, C_{300}, C_1 and β for the binary compounds and Si and Ge.

Table 7-3. Default Parameter Values for Heat Capacity

Material	ρ_β (gm/cm)	C_{300} (J/K kg)	C_1 (J/K/kg)	β
Si	2.33	711	255	1.85
Ge	5.327	360	130	1.3
GaAs	5.32	322	50	1.6
AlAs	3.76	441	50	1.2
InAs	5.667	394	50	1.95
InP	4.81	410	50	2.05
GaP	4.138	519	50	2.6

For the ternary compounds and SiGe, the parameters ρ, C_{300}, C_1 and β are interpolated versus composition using a simple linear form as in Equation 7-6.

$$Pab = (1-x)Pa + xPb \quad 7\text{-}6$$

When the lattice heat flow equation is solved, the global device temperature parameter is the maximum temperature at any mesh point in the device.

Specifying Heat Sink Layers For Thermal Solutions

You can define regions to only include thermal calculations. These regions will typically consist of layers associated with heat sinks. They are defined using the REGION statement. Even though in reality the heat sink materials are typically metal conductors, it is easier to specify these layers with the material type, INSULATOR. This is because as insulators the program will only solve heat flow and not attempt to solve current continuity in these layers. The region number is subsequently used as an identifier when thermal conductivities and heat capacities are assigned to these regions.

The following statements specify two layers of a heat sink for inclusion in the thermal calculation.

```
REGION NUM=5 Y.MIN=0.5 Y.MAX=2.0 INSULATOR
REGION NUM=6 Y.MIN=2.0 Y.MAX=3.0 INSULATOR
```

Specifying Thermal Conductivity

The value of thermal conductivity, k, for each region should be specified in the MATERIAL statement. Because thermal conductivity is generally temperature dependent, the following four models are available:

$$k(T) = \texttt{TC.CONST} \qquad \left(W/cm \cdot K\right). \tag{7-7}$$

$$k(T) = (\texttt{TC.CONST})/(T/300)^{\texttt{TC.NPOW}} \qquad \left(W/cm \cdot K\right). \tag{7-8}$$

$$k(T) = 1/(\texttt{TC.A} + (\texttt{TC.B})*\mathrm{T} + (\texttt{TC.C})*\mathrm{T}^2) \qquad \left(W/cm \cdot K\right). \tag{7-9}$$

$$k(T) = (\texttt{TC.E})/(\mathrm{T} - \texttt{TC.}\mathbf{D}) \qquad \left(W/cm \cdot K\right). \tag{7-10}$$

The TC.CONST, TC.NPOW, TC.A, TC.B, TC.C, TC.D and TC.E parameters are all user-specifiable in the MATERIAL statement. The choice of models is also user-specified in the MATERIAL statement. To choose the model in Equation 7-7, specify the TCON.CONST parameter. To choose the model in Equation 7-8, specify the TCON.POWER parameter. To choose the model in Equation 7-9, specify the TCON.POLYN parameter. To choose the model in Equation 7-10, specify the TCON.RECIP parameter. The default model is the polynomial model in Equation 7-9.

The following statements would be used to specify the temperature dependent thermal conductivity of the regions previously defined.

```
MATERIAL REGION=5 TC.A=<n> TC.B=<n> TC.C=<n>

MATERIAL REGION=6 TC.A=<n> TC.B=<n> TC.C=<n>
```

C-Interpreter Defined Thermal Conductivity

You can use the C-INTERPRETER to define the thermal conductivity, TCOND, as a function of the lattice temperature, position, doping and fraction composition. This is defined using the syntax:

```
MATERIAL REGION=<n> F.TCOND=<filename>
```

where the <filename> parameter is an ASCII file containing the C-INTERPRETER function. For more information about the C-INTERPRETER, see Appendix A: “C-Interpreter Functions”.

Anisotropic Thermal Conductivity

The flux term in Equation 7-1 models the thermal conductivity κ as being isotropic by default. You can specify an anisotropic thermal conductivity κ_{aniso}. In GIGA2D, the anisotropic value is applied in the y-direction. In GIGA3D, it is applied in the z-direction by default. In this case, the thermal conductivity tensor is

$$\kappa = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_1 & 0 \\ 0 & 0 & \kappa_2 \end{bmatrix} \tag{7-11}$$

You can also change the anisotropic direction to be the y-direction instead of the z-direction by specifying the YDIR.ANISO parameter or by specifying ^ZDIR.ANISO.

The thermal conductivity is temperature dependent, and the anisotropic thermal conductivity must have a model selected for its dependence on temperature. To do this, select TANI.CONST, TANI.POWER, TANI.POLYNOM or TANI.RECIP. These are analagous to TCON.CONST, TCON.POWER, TCON.POLYNOM and TCON.RECIP respectively. You can specify different temperature dependency models for the $\kappa(T)$ and $\kappa_{aniso}(T)$. The parameters for the chosen model for $\kappa_{aniso}(T)$ are specified by using the relevant parameters from A.TC.CONST, A.TC.A, A.TC.B, A.TC.C, A.TC.D, A.TC.E and A.TC.NPOW. These are equivalent to the isotropic parameters but with a prefix of A. to distinguish them. No built-in models are available for the anisotropic component of thermal conductivity. To give the anisotropic component the same temperature dependence as for the built-in model (Equation 7-2), specify the TANI.POWER.

As in the case of anisotropic permittivity, the discretization of the flux term is modified. For the simple cases of a rectangular mesh and a diagonal thermal conductivity tensor, the discretization chooses the value of thermal conductivity appropriate to the direction.

If the coordinate system where the thermal conductivity tensor is diagonal and the ATLAS coordinate system are non-coincident, then the coordinate transformation can be specified as a set of vectors

```
X = (X.X, X.Y, X.Z)
Y = (Y.X, Y.Y, Y.Z)
Z = (Z.X, Z.Y, Z.Z)
```

where the vector components can be specified on the MATERIAL statement. The default is that

```
X = (1.0, 0.0, 0.0)
Y = (0.0, 1.0, 0.0)
Z = (0.0, 0.0, 1.0)
```

and you should specify all necessary components. You do not need to normalize the vectors to unit length. ATLAS will normalize them and transform the relative dielectric tensor according to the usual rules. If you specify any component of X, Y or Z, then a more complete form of discretization will be used.

This is similar to the complete discretization carried out for the case of a generally anisotropic dielectric permittivity described in Section 3.12: "Anisotropic Relative Dielectric Permittivity".

To enable the more complete form of discretization, use the TC.FULL.ANISO parameter on the MATERIAL statement.

Table 7-4 shows the parameters used in Anisotropic Thermal Conductivity.

Table 7-4. MATERIAL Statement Parameters

Parameter	Type	Default	Units
A.TC.CONST	Real	0.0	(W/cm/K)
A.TC.A	Real	0.0	cm K /W
A.TC.B	Real	0.0	cm /W
A.TC.C	Real	0.0	cm / W /K
A.TC.D	Real	0.0	K
A.TC.E	Real	0.0	W/cm
A.TC.NPOW	Real	0.0	-
TANI.CONST	Logical	False	
TANI.POWER	Logical	False	

Table 7-4. MATERIAL Statement Parameters			
Parameter	**Type**	**Default**	**Units**
TANI.POLYNOM	Logical	False	
TANI.RECIP	Logical	False	
TC.FULL.ANISO	Logical	False	
YDIR.ANISO	Logical	False	
ZDIR.ANISO	Logical	True	
X.X	Real	1.0	
X.Y	Real	0.0	
X.Z	Real	0.0	
Y.X	Real	0.0	
Y.Y	Real	1.0	
Y.Z	Real	0.0	
Z.X	Real	0.0	
Z.Y	Real	0.0	
Z.Z	Real	1.0	

Specifying Heat Capacity

For transient calculations, specify heat capacities for every region in the structure. These are also functions of the lattice temperature and are modeled as:

$$C = \mathrm{HC.A} + \mathrm{HC.B}\,T + \mathrm{HC.C}\,T^2 + \frac{\mathrm{HC.D}}{T^2} \quad (J/cm^3/K) \qquad 7\text{-}12$$

Default values of HC.A, HC.B, HC.C, and HC.D are provided for common materials. You can specify these values in the MATERIAL statement.

The following statements will be used to specify the temperature dependent heat capacities of the regions previously defined.

```
MATERIAL REGION=5 HC.A=<n> HC.B=<n> HC.C=<n> HC.D=<n>

MATERIAL REGION=6 HC.A=<>> HC.B=<n> HC.C=<n> HC.D=<n>
```

Table 7-5. User Specifiable Parameters for Equation 7-12		
Statement	**Parameter**	**Units**
MATERIAL	HC.A	$J/cm^3/K$

Table 7-5. User Specifiable Parameters for Equation 7-12		
Statement	**Parameter**	**Units**
MATERIAL	HC.B	$J/cm^3 \cdot K^2$
MATERIAL	HC.C	$J/cm^3 \cdot K^3$
MATERIAL	HC.D	JK/cm^3

C-Interpreter Defined Thermal Capacity

You can use the C-INTERPRETER to define the thermal capacity, TCAP, as a function of the lattice temperature, position, doping and fraction composition. This is defined using the syntax:

```
MATERIAL REGION=<n.region> F.TCAP=<filename>
```

where the <filename> parameter is an ASCII file containing the C-INTERPRETER function. For more information about the C-INTERPRETER, see Appendix A: "C-Interpreter Functions".

7.2.2: Non-Isothermal Models

Effective Density Of States

When lattice heating is specified with the drift-diffusion transport model, the effective density of states for electrons and holes are modeled as functions of the local lattice temperature as defined by Equations 3-31 and 3-32 in Chapter 3: "Physics".

When lattice heating is specified with the energy balance model, the effective densities of states are modeled as functions of the local carrier temperatures, T_n and T_p, as defined by Equations 3-120 and 3-121 in Chapter 3: "Physics".

Non-isothermal Current Densities

When GIGA is used, the electron and hole current densities are modified to account for spatially varying lattice temperatures:

$$\vec{J}n = -q\mu_n n(\nabla\phi_n + P_n \nabla T_L) \quad 7\text{-}13$$

$$\vec{J}p = -q\mu_p p(\nabla\phi_p + P_p \nabla T_L) \quad 7\text{-}14$$

where:

P_n and P_p are the absolute thermoelectric powers for electrons and holes. P_n and P_p are modeled as follows:

$$P_n = \frac{k}{q}\left(ln\frac{n}{N_c} - \left(\frac{5}{2} + \text{KSN}\right)\right) \quad 7\text{-}15$$

$$P_p = \frac{k}{q}\left(ln\frac{p}{N_v} - \left(\frac{5}{2} + \text{KSP}\right)\right) \quad 7\text{-}16$$

Table 7-6 shows the default values for Equations 7-15 and 7-16.

Table 7-6. User-Specifiable Parameters for Equations 7-15 and 7-16

Statement	Parameter	Default	Units
MODELS	KSN	-1	None
MODELS	KSP	-1	None

7.2.3: Heat Generation

When carrier transport is handled in the drift-diffusion approximation the heat generation term, H, used in Equation 7-1 has the form [144]:

$$H = q\frac{\left|\overrightarrow{J_n}\right|^2}{\mu_n n} + q\frac{\left|\overrightarrow{J_p}\right|^2}{\mu_p p} - qT_L\left(\overrightarrow{J_n}\nabla P_n\right) + qT_L\left(\overrightarrow{J_p}\nabla P_p\right) + \qquad 7\text{-}17$$

$$q(R-G)\left\{\left\{\left[T_L\left(\frac{\partial\phi_n}{\partial T}\right)_{n,\,p} - \phi_n\right]\right\}\left(-\left[T_L\left(\frac{\partial\phi_p}{\partial T}\right)_{n,\,p} - \phi_p\right]\right\}\right)$$

$$-qT_L\left[\left(\frac{\partial\phi_n}{\partial T}\right)_{n,\,p} + P_n\right]div\ \ J_n - qT_L\left[\left(\frac{\partial\phi_p}{\partial T}\right)_{n,\,p} + P_p\right]divJ_n$$

In the steady-state case, the current divergence can be replaced with the net recombination. Equation 7-17 then simplifies to:

$$H = \left[q\frac{\overrightarrow{\left|J_n\right|^2}}{\mu_n n} + q\frac{\overrightarrow{\left|J_p\right|^2}}{\mu_p p}\right] + q(R-G)[\phi_p - \phi_n + T_L(P_p - P_n)] - qT_L\left(\overrightarrow{J_n}\nabla P_n + \overrightarrow{J_p}\nabla P_p\right) \qquad 7\text{-}18$$

where:

$\left[q\frac{\left|\overrightarrow{J_n}\right|^2}{\mu_n n} + q\frac{\left|\overrightarrow{J_p}\right|^2}{\mu_p p}\right]$ is the Joule heating term,

$q(R-G)[\phi_p - \phi_n + T_L(P_p - P_n)]$ is the recombination and generation heating and cooling term,

$-qT\left(\overrightarrow{J_n}\nabla P_n + \overrightarrow{J_\rho}p\nabla P_p\right)$ accounts for the Peltier and Thomson effects.

A simple and intuitive form of H that has been widely used in the past is:

$$H = \left(\overrightarrow{J_n} + \overrightarrow{J_p}\right)\cdot\overrightarrow{E} \qquad 7\text{-}19$$

GIGA can use either Equations 7-18 or 7-19 for steady-state calculations. By default, Equation 7-19 is used. Equation 7-18 is used if the HEAT.FULL parameter is specified in the MODELS statement. To enable/disable the individual terms of Equation 7-18, use the JOULE.HEAT, GR.HEAT, and PT.HEAT parameters of the MODEL statement.

If the general expression shown in Equation 7-17 is used for the non-stationary case, the derivatives $\left(\frac{\partial \phi_n}{\partial T_L}\right)_{n,\,p}$ and $\left(\frac{\partial \phi_p}{\partial T_L}\right)_{n,\,p}$ are evaluated for the case of an idealized non-degenerate semiconductor and complete ionization.

The heat generation term, H, is always set equal to 0 in insulators.

When carrier transport is modeled in the energy balance approximation, the following expression is used for H:

$$H = W_n + W_p + E_g \quad U, \tag{7-20}$$

where U, W_n and W_p are defined by Equations 3-122 through 3-124 in Chapter 3: "Physics".

7.2.4: Thermal Boundary Conditions

At least one thermal boundary condition must be specified when the lattice heat flow equation is solved. The thermal boundary conditions used have the following general form:

$$\sigma\left(\vec{J}^{u}_{tot} \cdot \hat{s}\right) = \alpha(T_L - T_{ext}) \tag{7-21}$$

where σ is either 0 or 1, $\vec{J}^{u}_{tot}$ is the total energy flux and $\hat{s}$ is the unit external normal of the boundary.

The projection of the energy flux onto s is:

$$\left(\overrightarrow{J^{u}_{tot}} \cdot \hat{s}\right) = -\kappa\frac{\partial T_L}{\partial n} + (T_L P_n + \phi_n)\vec{J}n \cdot \hat{s} + (T_L P_p + \phi_p)\left(\vec{J}p \cdot \hat{s}\right) \tag{7-22}$$

When $\sigma = 0$, Equation 7-21 specifies a Dirichlet (fixed temperature) boundary condition:

$$T_L = \texttt{TEMPER}. \tag{7-23}$$

where you define TEMPER in the THERMCONTACT statement as shown in the next section. You can specify dirichlet boundary conditions for an external boundary (which may coincide with an electrode) or for an electrode inside the device.

When σ = 1, Equation 7-21 takes the form:

$$\left(\overrightarrow{J^{u}_{tot}} \cdot \hat{s}\right) = \frac{1}{R_{th}}(T_L - \texttt{TEMPER}) \tag{7-24}$$

where the thermal resistance, R_{th}, is given by:

$$R_{th} = \frac{1}{\texttt{ALPHA}} \tag{7-25}$$

and ALPHA is user-definable on the THERMCONTACT statement.

Specifying Thermal Boundary Conditions

Setting thermal boundary conditions is similar to setting electrical boundary conditions. The THERMCONTACT statement is used to specify the position of the thermal contact and any optional properties of the contact. You can place thermal contacts anywhere in the device (including sidewalls). Equation 7-24 is used if a value is specified for α. Otherwise, Equation 7-23 is used.

The following command specifies that thermal contact number 1 is located between x=0 μm and x=2 μm at y=0 μm, and that the temperature at the contact is 300K.

```
THERMCONTACT NUM=1 X.MIN=0 X.MAX=2 Y.MIN=0 Y.MAX=0 TEMP=300
```

You can use simpler statement if the coordinates of a thermal contact coincide with the coordinates of an electrical contact. In this case, it is permissible to specify the location of the thermal contact by referring to the electrode number of the electrical contact. For example, the statement:

```
THERMCONTACT NUM=1 ELEC.NUM=3 TEMP=400
```

specifies that thermal contact number 1 is located in the same position as electrode number 3 and that the contact temperature is 400K.

Specifying the BOUNDARY parameter gives you flexibility in applying thermal boundary conditions. This parameter is set by default and means that the thermal boundary condition will only be set on the outside surface of the thermal contact, where it forms the exterior of the device. If the parameter is cleared by specifying ^BOUNDARY in the THERMCONTACT statement, the boundary conditions will be applied to the interior of the thermal contact and to the part of the surface of the thermal contact that forms an interface with the interior of the device. In a 2D model, you may have a thermal contact that is internal to the device and in this case you would need to specify ^BOUNDARY. Otherwise, the contact will be ignored.

For example:

```
THERMCONTACT ELEC.NUM=1 ^BOUNDARY TEMPER=450 ALPHA=2.5
```

where the ELECTRODE extends into the device and will set the interior points of the thermalcontact to 450K and will apply the flux boundary condition on all faces of the thermal contact that interface with the interior of the device.

Table 7-7. User Specifiable Parameters for Equations 7-23 and 7-24

Statement	Parameter	Units	Default
THERMCONTACT	ALPHA	$W/(cm^2K)$	∞
THERMCONTACT	BOUNDARY	-	True
THERMCONTACT	TEMPER	K	300

Representing a thermal environment, in terms of thermal impedances, leads to efficient solutions. But, thermal impedance representations are typically only approximations. Detailed thermal modeling (e.g., the effect of heat sink design changes) typically requires using detailed modeling of thermal regions with specified external ambient temperatures.

Note: You can't alter the value of a thermal resistor within a sequence of SOLVE statements. Rerun the input file whenever a thermal resistor is changed.

7.2.5: Temperature Dependent Material Parameters

GIGA automatically uses the built-in temperature dependence of the physical models that are specified. When lattice heating is specified, temperature dependent values are evaluated locally at each point in the device. When lattice heating is not solved, the models only provide global temperature dependence. In other words, all points in the device are assumed to be at the specified temperature.

The non-isothermal energy balance model uses the same carrier temperature dependencies of the mobility and impact ionization models as in the pure energy balance case, but with coefficients that depend on the local lattice temperature. Impact ionization coefficients depend on lattice temperature. Almost all other models and coefficients depend on lattice temperature.

When lattice heating is used, there is no point in specifying models that do not include temperature dependence. For mobilities, don't specify CONMOB. Instead, specify ANALYTIC or ARORA. For impact ionization coefficients, specify the SELBERHERR model.

GIGA can account for the temperature dependence of the minority carrier lifetimes for electrons or holes or both. The LT.TAUN (electrons) and LT.TAUP (holes) parameters of the MATERIAL statement are used to select this model. This model is turned on whenever the value of LT.TAUN or LT.TAUP is greater than 0, which is the default.

The temperature dependence of electron and hole lifetimes in the SRH recombination model have the forms:

$$\tau_n = \mathrm{TAUN0}\left(\frac{T}{300}\right)^{\mathrm{LT\cdot TAUN}} \quad \text{7-26}$$

$$\tau_p = \mathrm{TAUP0}\left(\frac{T}{300}\right)^{\mathrm{LT\cdot TAUP}} \quad \text{7-27}$$

Table 7-8. User-Specifiable Parameters for Equations 7-26 and 7-27

Statement	Parameter	Default	Units
MATERIAL	TAUN0	1×10^{-7}	s
MATERIAL	TAUP0	1×10^{-7}	s
MATERIAL	LT.TAUN	0	
MATERIAL	LT.TAUP	0	

See Equations 3-285 and 3-286 in Chapter 3: "Physics" for information regarding concentration dependent lifetimes.

7.2.6: C-Interpreter Defined Peltier Coefficients

You can use the C-INTERPRETER to define the Peltier coefficients, KSN and KSP, as a function of the electron and hole carrier temperatures, T_n and T_p. This is defined using the syntax:

```
MODELS F.KSN=<filename> F.KSP=<filename>
```

where the <filename> parameter is an ASCII file containing the C-INTERPRETER function. See Appendix A: "C-Interpreter Functions" for more information regarding the C-INTERPRETER and its functions.

7.3: Applications of GIGA

7.3.1: Power Device Simulation Techniques

This section contains a series of techniques, which you may find useful when simulating typical power device structures. Not all of the features described below are specific to GIGA and are common to the ATLAS framework.

Floating Guard Rings

No special syntax is needed for the simulation of uncontacted doping areas used in floating guard rings. The program is able to simulate guard ring breakdown with the standard impact ionization models. In some extreme cases, convergence may be slow due to poor initial guesses. If convergence is slow, both GUMMEL and NEWTON should be used in the METHOD statement.

Floating Field Plates

You should use the ELECTRODE statement to specify the field plate regions as electrodes. If these plates do not contact any semiconductor, then you can set these electrodes to float in the same manner as EEPROM floating gates. The following statement line specifies that the field plate region PLATE1 is a floating field plate.

```
CONTACT NAME=PLATE1 FLOATING
```

If the plates do contact the semiconductor, then do not use this syntax. Instead, current boundary conditions are used at the electrode with zero current. See Chapter 2: “Getting Started with ATLAS”, the “Floating Contacts” section on page 2-28 for more information about floating electrodes.

External Inductors

Inductors are commonly used in the external circuits of power devices. You can use the CONTACT statement to set an inductor on any electrode. The following statement sets an inductance on the drain electrode of 3 μH/μm.

```
CONTACT NAME=DRAIN L=3E-3
```

The next statement is used to specify a non-ideal inductor with a resistance of 100 Ω/μm.

```
CONTACT NAME=DRAIN L=3E-3 R=100
```

7.3.2: More Information

Many examples using GIGA have been installed on your distribution tape or CD. These include power device examples but also SOI and III-V technologies. You can find more information about the use of GIGA by reading the text associated with each example.

This page is intentionally left blank.

Chapter 8:
Laser: Edge Emitting Simulator

8.1: Overview

LASER performs coupled electrical and optical simulation of semiconductor lasers. LASER works with BLAZE and allows you to:

- Solve the Helmholtz Equation (see Equation 8-1) to calculate the optical field and photon densities.
- Calculate the carrier recombination due to light emission (i.e., stimulated emission).
- Calculate optical gain, depending on the photon energy and the quasi-Fermi levels.
- Calculate laser light output power.
- Calculate the light intensity profile corresponding to the multiple transverse modes.
- Calculate light output and modal gain spectra for several longitudinal modes.

To perform a LASER simulation, you need to know ATLAS and BLAZE first. If you don't, read Chapter 2: "Getting Started" and Chapter 5: "BLAZE".

8.2: Physical Models

To simulate semiconductor lasers, the basic semiconductor equations (see Equations 3-1 to 3-4 in Chapter 3: "Physics") are solved self-consistently with an optical equation that determines the optical field intensity distribution. LASER uses the following coordinate system:

- The x axis is perpendicular to the laser cavity and goes along the surface (from left to right).
- The y axis is perpendicular to the laser cavity and goes down from the surface.
- The z axis goes along the laser cavity.

The x and y axes are the same as in other ATLAS products. The electrical and optical equations are solved in the x and y plane (i.e., perpendicular to the laser cavity).

8.2.1: Helmholtz Equation

LASER solves a two-dimensional Helmholtz equation to determine the transverse optical field profile $E_k(x,y)$ [180]:

$$\nabla_{xy}^2 E_k(x,y) + \left(\frac{\omega_m^2}{c^2} \varepsilon(x,y) - \beta_k^2 \right) E_k(x,y) = 0 \qquad 8\text{-}1$$

where:

$\nabla_{xy}^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$ is the two-dimensional Laplace operator, ω_m is the frequency corresponding to longitudinal mode m (ω_m corresponds to OMEGA on the LASER statement for single frequency simulations), c is the velocity of light in vacuum, and $\varepsilon(x,y)$ is the high frequency dielectric permittivity.

Equation 8-1 is a complex eigenvalue problem. LASER solves this equation to determine the set of complex eigenvalues (β_k) and corresponding eigenfunctions $E_k(x,y)$. LASER takes into account only the fundamental transverse mode solution. Therefore, the index (k) will be dropped from subsequent equations.

In principle, Equation 8-1 should be solved for each longitudinal mode that is taken into account. Since very few longitudinal modes are actually lasing, LASER solves Equation 8-1 only once for the longitudinal mode with the greatest power and subsequently assumes:

$$E_m(x,y) = E_o(x,y) \qquad 8\text{-}2$$

where $E_o(x,y)$ is the optical field corresponding to the most powerful longitudinal mode. This assumption is reasonable, since the shape of the solution is almost independent of the frequency within the range of interest.

For dielectric permittivity, LASER uses the following model [79]:

$$\varepsilon(r,z) = \varepsilon_0 + (-\mathrm{ALPHAR} + j)\frac{\sqrt{\varepsilon_0} g(x,y)}{k_\omega} - j\frac{\sqrt{\varepsilon_0}(\mathrm{ALPHAA} + \mathrm{FCN} \cdot n + \mathrm{FCP} \cdot p)}{k_\omega} \qquad 8\text{-}3$$

where:

- ε_0 is the bulk permittivity.
- ALPHAR is a line width broadening factor.
- $j = \sqrt{-1}$
- $k_\omega = \omega/c$
- $g\ (x,y)$ is the local optical gain.
- ALPHAA is the bulk absorption loss and is specified in the MATERIAL statement.
- ABSORPTION must be specified in the LASER statement to include absorption loss.
- FCN and FCP are the coefficients of the free-carrier loss and are set through the MATERIAL statement. FCARRIER must be specified on the LASER to include this loss mechanism.

Table 8-1: User-Specifiable Parameters for Equation 8-3

Statement	Parameter	Default	Units
MATERIAL	ALPHAR	4.0	
MATERIAL	ALPHAA	0.0	cm^{-1}
MATERIAL	EPSINF		
MATERIAL	FCN	3.0×10^{-18}	cm^2
MATERIAL	FCP	7.0×10^{-18}	cm^2
LASER	ABSORPTION	FALSE	
LASER	FCARRIER	FALSE	

Bulk Permittivity (ε_0)

The following rules apply to the specification of the bulk permittivity.

- If located in an MQW and EPSILON is defined in the MQW statement, that value is used.
- If EPSINF is defined in the MATERIAL statement, that value is used.
- If F.INDEX defined in the MATERIAL statement, then the bulk permittivity is calculated from the square of the index returned from the C-Interpreter function defined in the file specified by F.INDEX.
- If INDEX.FILE is defined in the MATERIAL statement, the bulk permittivity is calculated as the square of the index interpolated from the table specified in the file pointed to by INDEX.FILE.
- If the material is listed in Appendix B: “Material Systems”, Table B-30, the permittivity is calculated as the square of index interpolated from the built-in tables for that material.
- If PERMITTI is defined in the MATERIAL statement, that value is used. If it’s not defined, the default material permittivity is used.

8.2.2: Local Optical Gain

For a discussion of gain models, see Chapter 3: “Physics”, Sections 3.9.3: “The Standard Gain Model” through 3.9.5: “Tayamaya's Gain Model”.

8.2.3: Stimulated Emission

Carrier recombination due to stimulated light emission is modeled as follows:

$$R_{st}(x,y) = \sum_m \frac{c}{\mathrm{NEFF}} g(x,y)|E(x,y)|^2 \cdot S_m \quad 8\text{-}4$$

where R_{st} is the recombination rate due to stimulated light emission, NEFF is the group effective refractive index, and S is the linear photon density. The m subscript in this equation and all subsequent equations refer to a modal quantity. For example, S_m in Equation 8-4 is the photon linear density for mode m. The NEFF parameter is user-specifiable but has a default value of 3.57 (see Table 8-2).

Table 8-2: User-Specifiable Parameters for Equation 8-4

Statement	Parameter	Default	Units
LASER	NEFF	3.57	

8.2.4: Photon Rate Equations

Optical gain provides a link between optical and electrical models. The optical gain depends on the quasi-Fermi levels and in turn impacts dielectric permittivity (see Equation 8-3), and by the coupling between the stimulated carrier recombination rate (R_{st}) and the density of photons (S) as described by Equation 8-4.

To determine S_m, LASER solves the system of photon rate equations:

$$\frac{dS_m}{dt} = \left(\frac{c}{\mathrm{NEFF}} G_m - \frac{1}{\tau_{ph_m}} - \frac{c \ \mathrm{LOSSES}}{\mathrm{NEFF}} \right) S_m + R_{sp_m} \quad 8\text{-}5$$

where the modal gain G_m is given by:

$$G_m = \iint g_m(x,y) \cdot |E(x,y)|^2 \cdot dx \ dy \quad 8\text{-}6$$

and the modal spontaneous emission rate R_{sp_m} is given by:

$$R_{sp_m} = \iint (r_{sp}(x,y))_m \cdot dx \ dy \quad 8\text{-}7$$

LOSSES is the internal losses and can be specified in the MODEL statement (see Table 8-3). *E(x,y)* is the normalized optical field.

Table 8-3: User Specifiable Parameters for Equation 8-5

Statement	Parameter	Default	Units
LASER	LOSSES	0	cm^{-1}

The modal photon lifetime, τ_{ph_m}, in Equation 8-8 represents the losses in the laser. The losses per mode are given by [115,141]:

$$\frac{1}{\tau_{ph}} = \frac{c}{\mathrm{NEFF}}(\alpha_a + \alpha_{fc} + \alpha_{mir}) \quad 8\text{-}8$$

α_a is the bulk absorption loss, α_{fc} is the free-carrier loss, and α_{mir} is the mirror loss. The models for absorption and free carrier loss are switched on with the ABSORPTION and FCARRIER parameters on the LASER statement. The mirror loss is always on. These are defined as:

$$\alpha_a = \iint \mathrm{ALPHAA} \cdot |E(x,y)|^2 dx\, dy \quad 8\text{-}9$$

$$\alpha_{fc} = \iint (\mathrm{FCN}\, n + \mathrm{FCP}\, p) \cdot |E(x,y)|^2 \cdot dx\, dy \quad 8\text{-}10$$

The mirror loss, α_{mir}, can be defined in two ways. One way is to define the percentage reflectivity of both mirrors. The other way is to define the individual facet mirror reflectivities. If the former is chosen, then set the MIRROR.LOSS parameter on the LASER statement to the percentage reflectivity for the facet cavity mirrors. This assumes that both front and back mirrors are indentical and gives a mirror loss calculated by:

$$\alpha_{mir} = \frac{1}{2\ \mathrm{CAVITY.LENGTH}} ln\left(\frac{1}{\mathrm{MIRROR.LOSS}^2}\right) \quad 8\text{-}11$$

where CAVITY.LENGTH is set to the length of the laser cavity on the LASER statement.

If you choose the latter model for α_{mir}, set the RR and RF parameters on the LASER statement instead. These parameters are the rear and front facet reflectivities respectively. The mirror loss is then calcualated by:

$$\alpha_{mir} = \frac{1}{2 \cdot \mathrm{CAVITY.LENGTH}} ln \frac{1}{\mathrm{RF.RR}} \quad 8\text{-}12$$

The user-specified parameters for the loss models in Equations 8-8 to 8-12 are given in Tables 8-1, 8-2, 8-3, and 8-4.

Table 8-4: User-Specifiable Parameters for LASER Loss Models

Statement	Parameter	Default	Units
LASER	MIRROR.LOSS	90.0	%
LASER	CAVITY.LENGTH	100.0	μm

Note: Absorption loss and free carrier loss are switched off by default. These are switched on by specifying the ABSORPTION and FCARRIER parameters in the LASER statement. Mirror loss is switched on by default to 90%.

8.2.5: Spontaneous Recombination Model

For a discussion of spontaneous recombination models, see Chapter 3: "Physics", Sections 3.9.1: "The General Radiative Recombination Model" and 3.9.2: "The Default Radiative Recombination Model".

8.2.6: Optical Power

The optical power emitted from the front mirror is calculated by [115, 160]

$$P_f = \frac{h\omega S_m c}{2\mathrm{NEFF}} \frac{ln(1/(R_f R_r))}{1+\sqrt{R_f/R_r}(1-R_r)/(1-R_f)} \quad \text{8-13}$$

where S_m is the photon density of mode m, w is the frequency of emitted light, R_f and R_r are the front and rear mirror reflectivities, and NEFF is a user-defined in the LASER statement.

8.2.7: Gain Saturation

To simulate non-linear gain saturation in LASER, use the simple model described by [28]

$$g'(x,y) = \frac{g(x,y)}{1+I(x,y)/\mathrm{GAIN.SAT}} \quad \text{8-14}$$

where GAIN.SAT is a user-specifiable parameter on the MATERIAL statement, $g'(x,y)$ is the local gain and $I(x,y)$ is the local intensity given by

$$I(x,y) = \frac{\Sigma}{m} Sm\left|E_m(x,y)\right|^2 \quad \text{8-15}$$

To enable this model, specify GAIN_SAT on the LASER statement. Non-linear, absorption loss is similarly modeled using the following expression.

$$\alpha'(x,y) = \frac{\alpha(x,y)}{1+I(x,y)/\mathrm{ABSORPTION.SAT}} \quad \text{8-16}$$

where $\alpha'(x,y)$ is the local absorption and ABSORPTION.SAT is a user-definable parameter on the MATERIAL statement. To enable this model, specify ABSOR_SAT on the LASER statment.

8.3: Solution Techniques

LASER solves the electrical and optical equations self consistently. LASER solves the Helmholtz equation in a rectangular region, which is a subdomain of the general ATLAS simulation region. An eigenvalue solver provides a set of eigenfunctions and corresponding eigenvectors.

LASER uses boundary conditions in the form, $E(x,y)=0$, on the boundaries of the solution region. This region should be large enough to cover the entire active region of the laser diode with some inclusion of the passive regions. When the single frequency model is used, the lasing frequency used in Equation 8-1 is an external parameter that is fixed during the calculation and either model can be then used for optical gain.

The multiple longitudinal mode model requires you to use a physically-based optical gain model. Specify an initial estimate of the lasing frequency. This is adjusted during calculations. You can also specify a frequency range or photon energy range, which LASER will then calculate multiple longitudinal modes.

LASER may also simulate multiple transverse modes.

8.4: Specifying Laser Simulation Problems

The structure of the laser diode and the mesh used to simulate it are specified in the normal way using the capabilities provided by ATLAS and BLAZE. To enable LASER simulation, use the parameters and do the following:

1. Activate LASER by specifying a LASER statement.
2. Define a mesh for solution of the Helmholtz equation. The Helmholtz Equation is solved on a rectangular mesh. The rectangular mesh can be independent of the triangular mesh, which is used for device simulation. This rectangular mesh is specified by the LX.MESH and LY.MESH statements. The area specified by the laser mesh must be completely inside the ATLAS simulation domain and should completely cover the active lasing region.
3. For single-frequency calculations, you can select different gain models for different regions/ materials using the REGION or MATERIAL parameters of the MODELS statement.

Note: If multiple longitudinal modes are to be accounted for, then specify GAINMOD=1 for the active lasing region.

8.4.1: LASER Statement Parameters

You can simulate multiple transverse modes by setting the NMODE parameter in the LASER statement. Set it to the desired number of modes you want to simulate.

There are several kinds of parameters that appear in the LASER statement. There are parameters that deal with the numerical solution to the photon rate equation, parameters that deal with loss mechanisms, and parameters that deal with specifying energies to look for longitudinal solutions.

Specify laser physical parameters and models and do the following:

1. Specify the CAVITY.LENGTH parameter, which is the length of the laser cavity in the z direction.
2. Specify the PHOTON.ENERGY or OMEGA parameters for (initial) photon energy or laser frequency.
3. Specify laser loss mechanisms (MIRROR, FCARRIER, ABSORPTION parameters in the LASER statement) and any associated constants (FCN, FCP, ALPHAA in the MATERIAL statement).
4. Specify any additional laser losses (LOSSES parameter).

If you want to calculate the laser spectrum, specify the multiple longitudinal modes model and additional parameters. Then, do the following:

1. Specify the LMODES in the statement. This enables the Multiple Mode Model.
2. Specify the EINIT and EFINAL parameters. These parameters set the photon energy range within, which LASER will take into account multiple longitudinal modes. Make sure that initial photon energy is within this range and is specified for the active lasing region.
3. Specify the photon energy separation (ESEP parameter). If this isn't specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range. We recommend that you allow LASER to choose the photon energy separation.
4. Specify the spectrum file name (SPEC.NAME). LASER will produce a structure file containing spectrum data after calculation of each bias point. LASER will automatically append *_dcN.log* to the specified file name (where *N* is the number of the bias point) for steady-state solutions or *_trN.log* for a transient simulation. The first bias point where LASER is active will have *N=1*. This is often not the first bias point during simulation. You can examine these files using TONYPLOT. If you specify the SPECSAVE parameter, the spectrum files will only be saved on every las.specsave solution.

Note: The index in the spectrum file name will still increase by one each time. The spectrum data can be stored in a single file for the transient simulation, only if MULTISAVE - LASER statement is set to FALSE.

You can save near and far field patterns by specifying a value for the PATTERNS parameter in the SAVE statement. The value of the PATTERNS parameter is a character string representing the root name of a file for saving the near and far field patterns. The near field pattern is saved to a file with the string ".nfp" appended to the root. The far field pattern is saved to a file with the string ".ffp" appended to the root name. You can examine these files using TONYPLOT.

You can also use the numeric parameters. These parameters are TOLER, ITMAX, SIN, TAUSS, and MAXCH. The default values of these parameters have been selected to give a good exchange between accuracy, efficiency, and robustness for most applications. You can adjust the calculation by specifying different values. These parameters are discussed in the next section.

8.4.2: Numerical Parameters

The following numerical parameters control the performance of the LASER simulation. All of these parameters have reasonable default values, but you can specify them in the LASER statement.

- ITMAX sets maximum number of external LASER iterations during photon density calculation. The default value is 30.
- MAXCH is the maximum allowed relative change of the photon density between LASER iterations. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.
- SIN is the initial photon density used only with simple LASER models. LASER starts the iteration process for photon density calculation from this value. This parameter influences only the calculation time for the first bias point after the laser threshold is reached.
- TAUSS is the relaxation parameter for the solution of the photon rate equation(s). Larger values of TAUSS tends to reduce the calculation time required to achieve convergence but leads to less stability. In other words, increasing TAUSS reduces calculation time while decreasing TAUSS improve convergence.
- TOLER sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.
- TRAP specifies that a cutback scheme for the photon rate equation(s) should be used if convergence is not obtained. In this case, the value of TAUSS is reduced by multipling it by ATRAP and the solution for the rate equations is repeated. The value of MAXTRAP specifies the maximum number of cut-backs attempts before the simulation ends.

8.5: Semiconductor Laser Simulation Techniques

The most common technique for simulating laser diodes is to simulate forward characteristics with a gradually increasing bias. The forward voltage is usually specified, but current boundary conditions can be used, and external elements can be included.

To save computational time, we recommend that you don't enable LASER models at the start of the simulation. Ramp the device bias first. Then, enable LASER simulation by using the LASER parameter in an additional MODELS statement.

Generally, the single frequency LASER model, which doesn't take into account the longitudinal mode spectrum, is faster. This model provides accurate results. We recommend that you use it if the lasing spectrum isn't the subject of interest. If you use the multiple longitudinal mode model, computational time will longer depend on the number of longitudinal modes involved in the calculation.

Although ATLAS is a two-dimensional device simulator, the laser spectrum and all other laser results strongly depend on cavity length, which is effectively the device length of the z direction. For example, you can scale terminal currents to apply a different length in the third dimension by using simple multiplication. This isn't the case for LASER, since the cavity length determines energy spacing between longitudinal modes and influences all other laser characteristics.

Be sure to specify the CAVITY.LENGTH parameter in the MODELS statement when the longitudinal mode spectrum is to be calculated.

For symmetric devices, you can improve computational efficiency by simulating half of the device. To do this, define the device so that the center of the device is located at X=0 and only half of the device is defined at coordinate values at X>=0.

For device simulation, the default boundary condition is a mirror boundary, which means symmetry is implied.

For laser simulation, (see Helmholtz Equation in Section 8.2.1: "Helmholtz Equation") specify that the laser mesh starts at X=0. Also, specify the REFLECT parameter in the LASER statement. This tells the Helmholtz Equation that a mirror boundary condition is to be used at X=0.

8.5.1: Generation of Near-Field and Far-Field Patterns

The intensity profile of the laser at the cleaved surface can be examined in the standard structure file using TONYPLOT. To generate a structure file, use the OUTFILE parameter on either the SOLVE or SAVE statements. The intensity far from the laser surface, in the Fraunhofer region, is important for designing the optical coupling to the laser.

LASER can generate the far-field pattern if specify the PATTERNS parameter of the SAVE statement. The PATTERNS parameter specifies the file name prefix of two output files. The first of these files, with the *.nfp* suffix, contains the near-field pattern. This is essentially a copy of the near field pattern contained in the structure file output. The second of these files, with the *.ffp* suffix, contains the far-field pattern.

The far-field pattern is essentially a 2D fast Fourier transform (FFT) of the near-field pattern. The sampling for this transform is set by the minimum spacing in the laser mesh and the overall size of the laser mesh. The output samplings for the near-field and far-field patterns are controlled by the NEAR.NX, NEAR.NY, FAR.NX, and FAR.NY parameters of the LASER statement. These specify the numbers of samples in the X and Y directions in the near and far field. By default, the values are set to 100.

Chapter 9:
VCSEL Simulator

9.1: Overview

VCSEL (Vertical Cavity Surface Emitting Lasers) performs electrical, thermal, and optical simulation of vertical cavity surface emitting lasers using accurate, robust, and reliable fully numerical methods and non-uniform meshes. With VCSEL, you can simulate advanced structures containing multiple quantum wells and oxide apertures including the effects, such as gain guiding, lattice heating, current crowding, and spatial hole burning. VCSEL works with BLAZE and GIGA and allows you to do the following:

- Test whether the structure complies to general VCSEL device requirements by simulating cold cavity reflectivity experiments. Calculate the reflectivity of a VCSEL cavity as a function of the incident light wavelength in a user-specified range. Then, determine the resonant frequency of the cold cavity.
- Solve the Helmholtz equation (see Equation 9-1) in cylindrical coordinates to calculate optical intensities of the multiple transverse modes for index guiding and gain guiding structures.
- Calculate the optical gain in multiple quantum well systems depending on the photon energies, quasi-Fermi levels, temperature, and optical intensity distribution. Calculate the carrier recombination due to the spontaneous and stimulated emission.
- Solve the photon rate equations (see Section 9.2.4: “Photon Rate Equations”) for multiple transverse modes to calculate the photon density in each mode and the total photon density.
- Calculate the light output power and the wavelength for each transverse mode.
- Speed up solution of the Helmholtz equation by using perturbational treatment.
- Simulate more general cavities that support multiple longitudinal modes. This obtains modal gain spectra and determines the dominant mode with maximum gain.

To perform a VCSEL simulation, you need be familiar with ATLAS and BLAZE first. Please see Chapter 2: “Getting Started” and Chapter 5: “BLAZE” for more information.

Note: To avoid the confusion in terms, we refer to a VCSEL device as VCSEL and to a VCSEL simulator as VCSEL:ATLAS in this chapter.

9.2: Physical Models

When modeling VCSELs, it is essential to take into account interaction of optical, electrical, and thermal phenomena that occur during the VCSEL operation. Modeling only optical properties is already an involved task. Different methods developed recently to improve the accuracy of optical solution produce results that vary from method to method [22]. The variation is especially strong between vectorial models that tend to produce a spread in results much larger than simpler scalar models.

On the other hand, the complexity of the vectorial methods makes the self-consistent electro-thermo-optical simulation impractical from the point of view of the calculation time required. For VCSEL simulation in VCSEL:ATLAS, we adopted an approach that can account for mutual dependence of electrical, optical, and thermal phenomena.

Basic semiconductor equations (See Equations 3-1 to 3-4 in Chapter 3: "Physics") are solved self-consistently with the Helmholtz Equation, Lattice Heat Flow Equation (See Chapter 7: "Giga: Self-Heating Simulator", Equation 7-1), and the Photon Rate Equation (see Equation 9-12). Since we are using a scalar method, we neglect polarization effects and reduce Maxwell equations to scalar Helmholtz equation. The method used for the solution of Helmholtz equation is based on a well developed effective index model, which showed its validity for a much wider range of problems than originally expected [177]. This method fine tuned for simulation of various VCSEL structures is often referred to as Effective Frequency Method (EFM). EFM is a fast and flexible method that allows further development to include dispersion, diffraction losses, and graded interfaces.

VCSEL:ATLAS uses a cylindrical coordinate system to take advantage of the cylindrical symmetry of the VCSEL devices. Figure 9-1 shows the relationship between ATLAS's X-Y coordinate system and the r-θ-z cylindrical system used in VCSEL:ATLAS. Note that the X coordinate axis is equivalent to the radius r, and the Y coordinate axis is equivalent to the z axis in the cylindrical system.

9.2.1: Reflectivity Test Simulation

Reflectivity experiment is an important test conducted to evaluate the quality of a VCSEL device. In this experiment, the light is normally incident on a VCSEL device and its reflectivity is measured as a function of wavelength. Most manufactured VCSEL resonators are tested this way to ensure adequate optical performance. Numerical simulation of this experiment enables you to calibrate the material parameters used in the optical model and eliminate possible specification errors at the early stage of numerical analysis of a VCSEL device.

Another purpose of the numerical reflectivity test is to ensure that the analyzed device complies with general VCSEL requirements. These requirements include the following:

1. Presence of a high reflectivity band in the reflectivity spectrum of a VCSEL cavity.
2. The reflectivity over 99% in the high reflectivity band.
3. Presence of a drop in reflectivity within the high reflectivity band. The drop in reflectivity (increase in absorptance) occurs at the resonant wavelength of the cavity.

VCSEL:ATLAS calculates reflectivity and absorptance using transfer matrix approach (Chapter 10: "Luminous: Optoelectronic Simulator", Section 10.2.4: "Matrix Method"). In the simulation, the incident light propagates along the axis of symmetry of the device.

The `VCSEL.INCIDENCE` parameter specifies the position of the light origin with respect to the device (top of bottom). Specific output associated with the reflectivity test includes reflectivity and absorptance spectra plotted in the corresponding files, `reflectivity(absorption)_top.log` and `reflectivity(absorption)_bottom.log`.

To run the reflectivity test simulation, specify the `VCSEL.CHECK` parameter and other parameters in Table 9-1 on a `VCSEL` statement. Section 9.3.3: "Enabling VCSEL Solution" will discuss parameter specification in detail.

Table 9-1: User-Specifiable Parameters for the Reflectivity Test			
Statement	**Parameter**	**Default**	**Units**
VCSEL	VCSEL.CHECK	False	
VCSEL	VCSEL.INCIDENCE	1	
VCSEL	OMEGA		s^{-1}
VCSEL	EINIT		eV
VCSEL	EFINAL		eV
VCSEL	PHOTON.ENERGY		eV
VCSEL	INDEX.TOP	1.0	
VCSEL	INDEX.BOTTOM	1.0	
VCSEL	NSPEC	100	

If the structure does not meet requirements 1-3, VCSEL:ATLAS will close with an error. Otherwise, VCSEL:ATLAS uses Brent's minimization procedure to find the resonant frequency of the cold cavity. VCSEL:ATLAS uses the obtained value as a reference frequency in the solution of the Helmholtz equation.

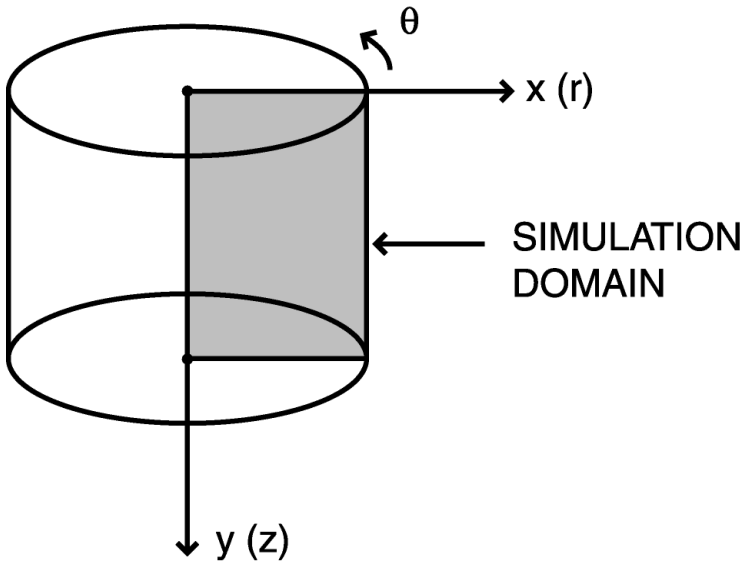


Figure 9-1: The relationship between rectangular and cylindrical coordinate systems used by VCSEL

9.2.2: Helmholtz Equation

VCSEL:ATLAS solves the Helmholtz Equation in cylindrical coordinates r, z, and ϕ using effective frequency method [177].

$$\nabla^2 E(r, z, \varphi) + \frac{\omega^2}{c^2}\varepsilon(r, z, \varphi, \omega)E(r, z, \varphi) = 0 \quad \text{9-1}$$

where ω is frequency, $\varepsilon(r, z, \varphi, \omega)$ is the complex dielectric permittivity, $E(r, z, \varphi)$ is the optical electric field, and c is the speed of light in vacuum.

Using the expansion around a reference frequency ω_0:

$$\frac{\omega^2}{c^2}\varepsilon(r, z, \varphi, \omega) \approx \frac{\omega_0^2}{c^2}\varepsilon(r, z, \varphi, \omega_0) + 2\frac{\omega_0}{c^2}\varepsilon(r, z, \varphi, \omega_0)(\omega - \omega_0) \quad \text{9-2}$$

we transform the Helmholtz equation into:

$$\left[\nabla^2 + k_0^2\varepsilon(r, z, \varphi)\right]E(r, z, \varphi) = \nu k_0^2\varepsilon(r, z, \varphi)E(r, z, \varphi) \quad \text{9-3}$$

where $k_0=\omega_0/c$ and ν is a dimensionless frequency parameter given by:

$$\nu = 2\frac{\omega_0 - \omega}{\omega_0} = 2\frac{\lambda - \lambda_0}{\lambda} - j\frac{2 \cdot Im(\omega)}{\omega_0} \quad \text{9-4}$$

We assume that the field is separable:

$$E(r, z, \varphi) = f(r, z, \varphi)\Phi(r, \varphi) \quad \text{9-5}$$

This assumption of separability is the main approximation of the method. The applicability of the method can be extended beyond this approximation by adding a non-separable component of the field on the right hand side of Equation 9-5. For dispersive materials, you can make another modification of the effective frequency method by taking into account material dispersion in Equation 9-2. For separable fields, we obtain the longitudinal wave equation for function f:

$$\left[\frac{\partial^2}{\partial z^2} + k_0^2\varepsilon(r, z, \varphi)\right]f(r, z, \varphi) = \nu_{eff}(r, \varphi)k_0^2\varepsilon(r, z, \varphi)f(r, z, \varphi) \quad \text{9-6}$$

at each lateral position (r, ϕ). Due to the cylindrical symmetry, we can ignore the azimuthal variations. In this case, we only need to solve Equation 9-6 for each cylindrically symmetric region characterized by a particular distribution of ε in the vertical direction z. Complex Newton method combined with the characteristic matrix approach (Chapter 10: "Luminous: Optoelectronic Simulator", Section 10.2.4: "Matrix Method") yields eigenvalues ν_{eff} and eigenfunctions $f(z)$ of the longitudinal wave equation.

The transverse wave equation takes the form:

$$\left[\frac{1}{r}\frac{d}{dr}\left(r\frac{d}{dr}\right) - \frac{l^2}{r^2} + \upsilon_{eff}(r)k_0^2\langle\varepsilon\rangle_r\right]\Phi_{lm}(r) = n_{lm}K_0^2\langle\varepsilon\rangle_r\Phi_{lm}(r) \quad \text{9-7}$$

where

$$\langle \varepsilon \rangle_r = \int_0^L \varepsilon(r,z) f^2(r,z) dz \quad \text{9-8}$$

VCSEL:ATLAS uses standard eigentechniques to solve an ordinary differential Equation 9-7. Solutions of this equation are LP_{lm} modes.

For each mode, the imaginary part of the eigenvalue determines the overall mode gain(loss) factor, which is negative below the lasing threshold. The lasing occurs when:

$$Im(n_{lm})\big|_{th} = 0 \quad \text{9-9}$$

The dielectric permittivity is given by:

$$\varepsilon(r,z) = \varepsilon_0 + (-\mathrm{ALPHAR} + j)\frac{\sqrt{\varepsilon_0} g(r,z)}{k_0} - j\frac{\sqrt{\varepsilon_0}(\mathrm{ALPHAA} + \mathrm{FCN} \cdot n + \mathrm{FCP} \cdot p)}{k_0} \quad \text{9-10}$$

where:

- ε_0 is the bulk permittivity.
- ALPHAR is a line width broadening factor.
- $k_0 = \omega_0 / c$
- $g\ (r, z)$ is the local optical gain.
- ALPHAA is the bulk absorption loss and is specified in the MATERIAL statement (specify ABSORPTION in the VCSEL statement to include absorption loss).
- FCN and FCP are the coefficients of the free-carrier loss and are set using the MATERIAL statement (specify FCARRIER in the VCSEL statement to include this loss mechanism).

Table 9-2: User-Specifiable Parameters for Equation 9-10

Statement	Parameter	Default	Units
MATERIAL	ALPHAR	4.0	
MATERIAL	ALPHAA	0.0	cm^{-1}
MATERIAL	EPSINF		
MATERIAL	FCN	3.0×10^{-18}	cm^2
MATERIAL	FCP	7.0×10^{-18}	cm^2
VCSEL	ABSORPTION	FALSE	
VCSEL	FCARRIER	FALSE	

Bulk Permittivity (ε_0)

The following rules apply to the specification of the bulk permittivity.

- If located in an MQW and EPSILON is defined in the MQW statement, that value is used.
- If EPSINF is defined in the MATERIAL statement, that value is used.
- If F.INDEX defined in the MATERIAL statement, then the bulk permittivity is calculated from the square of the index returned from the C-Interpreter function defined in the file specified by F.INDEX.
- If INDEX.FILE is defined in the MATERIAL statement, the bulk permittivity is calculated as the square of the index interpolated from the table specified in the file pointed to by INDEX.FILE.
- If the material is listed in Appendix B: "Material Systems", Table B-30, the permittivity is calculated as the square of index interpolated from the built-in tables for that material.
- If PERMITTI is defined in the MATERIAL statement, that value is used. If it's not defined, the default material permittivity is used.

9.2.3: Local Optical Gain

For a discussion of gain models, see Chapter 3: "Physics", Sections 3.9.3: "The Standard Gain Model" through 3.9.5: "Tayamaya's Gain Model".

Stimulated Emission

Carrier recombination due to stimulated light emission is modeled as follows:

$$R_{st}(r,z) = \sum_m \frac{c}{\mathrm{NEFF}} g_m(r,z)\left|E_m(r,z)\right|^2 \cdot S_m \qquad 9\text{-}11$$

where R_{st} is the recombination rate due to stimulated light emission, NEFF is the group effective refractive index, and S_m is the photon number. The m subscript in this equation and all subsequent equations refers to a modal quantity. For example, S_m in Equation 9-11 is the photon number for mode m. The NEFF parameter is user-defined but has a default value of 3.57 (see Table 9-3).

Table 9-3: User-Specifiable Parameters for Equation 9-11

Statement	Parameter	Default	Units
VCSEL	NEFF	3.57	

9.2.4: Photon Rate Equations

Optical gain provides the link between optical and electrical models. The optical gain depends on the quasi-Fermi levels and in turn impacts dielectric permittivity (see Equation 9-12), and by the coupling between the stimulated carrier recombination rate (R_{st}) and the density of photons (S) as described by Equation 9-11.

To determine S_m, VCSEL solves the system of photon rate equations:

$$\frac{dS_m}{dt} = \left(\frac{c}{\mathrm{NEFF}} G_m - \frac{1}{\tau_{ph_m}} - \frac{c}{\mathrm{NEFF}} \frac{\mathrm{LOSSES}}{}\right) S_m + R_{sp_m} \qquad 9\text{-}12$$

where the modal gain G_m is given by

$$G_m = \iiint g_m(r,z) \cdot \left|E_m(r,z)\right|^2 r\,d\theta\,dr\,dz \quad \text{9-13}$$

and the modal spontaneous emission rate R_{sp_m} is given by

$$R_{sp_m} = \iiint r_{sp}(r,z)_m r\,d\theta\,dr\,dz\,. \quad \text{9-14}$$

$E_m(r,z)$ is the normalized optical field.

Table 9-4: User Specifiable Parameters for Equation 9-12			
Statement	**Parameter**	**Default**	**Units**
VCSEL	LOSSES	0	cm^{-1}

The modal photon lifetime, τ_{ph_m}, in Equation 9-11 represents the losses in the laser. The losses per mode are given by

$$\frac{1}{\tau_{ph_m}} = \frac{c}{\mathrm{NEFF}}\left(\alpha_{a_m} + \alpha_{fc_m} + \alpha_{mir}\right) = \frac{c}{\mathrm{NEFF}} G_m - \omega_0 \cdot \nu_{lm} \quad \text{9-15}$$

α_a is the bulk absorption loss, α_{fc} is the free-carrier loss, and α_{mir} is the mirror loss. These are defined as:

$$\alpha_{a_m} = \iiint \mathrm{ALPHAA} \cdot \left|E_m(r,z)\right|^2 \times r\,d\theta\,dr\,dz \quad \text{9-16}$$

$$\alpha_{fc_m} = \iiint (\mathrm{FCN}\ n + \mathrm{FCP}\ p) \cdot \left|E_m(r,z)\right|^2 r\,d\theta\,dr\,dz \quad \text{9-17}$$

$$\alpha_{mir} = G_m - \alpha_{a_m} - \alpha_{fc_m} - \omega_0 \cdot \nu_{lm} \frac{c}{\mathrm{NEFF}} \quad \text{9-18}$$

Spontaneous Recombination Model

For a discussion of spontaneous recombination models, see Chapter 3: “Physics”, Sections 3.9.1: “The General Radiative Recombination Model” and 3.9.2: “The Default Radiative Recombination Model”.

Optical Power

The optical power emitted is given by Equation 9-19.

$$P_f = \frac{\hbar\omega S_m c}{\mathrm{NEFF}} \frac{\alpha_{mir}}{1 + \sqrt{R_f / R_r (1-R_r)/(1-R_f)}} \quad \text{9-19}$$

9.3: Simulating Vertical Cavity Surface Emitting Lasers

The key to simulating a Vertical Cavity Surface Emitting Laser is the input deck. The input deck describes the physical device structure, the physical models to be used, and the specific measurements to be performed. The deck itself can be roughly divided into three sections: structure specification, model definition, and obtaining solutions. The following sections will describe these concepts in detail.

9.3.1: Specifying the Device Structure

VCSEL devices are the most complicated devices to be addressed by device simulation. This is due to the many layers involved in making Distributed Bragg Reflectors (See "Specifying Distributed Bragg Reflectors" section on page 9-10) and the possible Multiple Quantum Well (See "Specifying Quantum Wells" section on page 9-11) active layers. This complexity is mitigated by the fact that these devices are mostly epitaxial. Also, many of the layers are periodic. Recognizing these simplifications, we've designed a custom syntax for defining VCSEL devices.

This syntax makes specification of periodic structures (e.g., super-lattices) simple, which completely avoids problems of aligning various layers to each other. This new syntax also simplifies meshing the structure.

Despite this simplification, specifying a VCSEL device can still be complex. We recommend that you look at the standard examples. See Chapter 2: "Getting Started with ATLAS", Section 2.4: "Accessing The Examples" for more information about these examples.

The order of statements is important when specifying the device structure. The following order of statements should be strictly followed.

1. Specify the `MESH` statement. This initializes the structure definition and allows specification of cylindrical symmetry.
2. Set up the mesh in the X direction by using the `X.MESH` statements. The mesh in the Y direction is usually specified along with the device layers in the `REGION` statements. You can, however, introduce the Y mesh lines by using the `Y.MESH` statements after the `X.MESH` statements.
3. Specify the device regions using `REGION` and `DBR` statements. These statements specify the layer thickness, composition, doping, order of placement, and meshing information in the Y direction. The order of `REGION` and `DBR` statements is up to you and is usually defined by the order the layers appear through the structure.
4. You can use quantum well models for the gain and spontaneous recombination and then specify the locations of quantum wells within the device, using the `MQW` statement. These statements specify the location, doping and composition of the wells, and certain model information about the wells.

   The wells are centered somewhere within the geometry of the previously specified (using `REGION` statements) regions, so that the well barrier layers take on the composition of the underlying region. The composition of the wells is specified in the `MQW` statement.

5. Specify the locations of the electrodes using `ELECTRODE` statements.
6. Use `DOPING` statements to specify additional doping or composition fraction information. This usually isn't necessary since most, if not all doping and composition information can be specified during the specification of regions.

Setting Up Cylindrical Symmetry

VCSELs use cylindrical symmetry to account for the 3D nature of VCSEL devices. To specify cylindrical symmetry in the MESH statement, use the CYLINDRICAL parameter. An example of specifying cylindrical symmetry is down below.

Example

```
MESH CYLINDRICAL
```

The axis of symmetry for cylindrical is the Y axis at X=0. When using cylindrical symmetry, no X coordinates should be specified with negative values since X represents radius.

The usual abbreviation rules for parameter names can be used. See Chapter 2: “Getting Started with ATLAS”, Section 2.5: “The ATLAS Syntax” for more information.

Specifying the Mesh

To specify the mesh in the X direction, use the X.MESH statement. Each X.MESH statement specifies the location of one mesh line in the X direction and the spacing between mesh lines at that location.

The location is specified by the LOCATION parameter in microns. The spacing is specified by the SPACING parameter, which is also in microns. The number of mesh lines specified is limited to built-in limits in the overall size of the mesh. There should be, however, at least two X.MESH statements. There shouldn’t be X.MESH specifications with negative values in the LOCATION parameter. Such specification would be ill-defined due to the cylindrical symmetry.

You should also specify mesh lines at any defining edges of regions, electrodes, or other geometrical aspects in the X direction Also for similar reasons, use the Y.MESH statements to specify mesh lines in the Y direction.

To have mesh lines inserted automatically at defined region edges, use the REGION and DBR statements. To enable this feature, specify AUTO in the MESH statements. See Chapter 19: “Statements”, Section 19.26: “MESH” for information about the MESH statement.

Specifying Regions

A region is a volume (in cylindrical coordinates a disk or annulus) that has a uniform material composition. The region is specified by a REGION statement. In the REGION statement, the material composition is specified by the MATERIAL parameter.

The MATERIAL parameter can take on values of any material name as described in Appendix B: “Material Systems”. For Ternary and Quaternary materials, you can also specify composition fractions X or Y or both, using the X.COMPOSE or Y.COMPOSE parameters or both. You can use both the DONORS and ACCEPTORS parameters to specify uniform densities of ionized donors or acceptors or both.

The thickness of the region in the Y direction in microns is specified by the THICKNESS parameter. You can also use the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters to specify the location and extent of the region. Except in special cases, we don’t recommend this because of the difficulties aligning the numerous layers involved.

Unless it’s specified, the region will encompass the entire range of mesh in the X direction as specified in the X.MESH statements. Mesh lines in the Y direction will then be added at the upper and lower edges of the region. The mesh spacing at the edges is specified in microns by the SY parameter.

The TOP/BOTTOM parameters specifies the relative ordering of regions in the Y direction. Specifying TOP indicates that the region starts at the top of all previously specified regions (i.e., at the minimum previously specified Y coordinate) and extends to the thickness in the negative Y direction. Remember for device simulation, the Y axis conventionally extends down into the wafer for increasing values of Y. If no previous regions are specified, TOP indicates the region starts at Y=0 and extends to the

thickness in the negative Y direction. BOTTOM indicates that the region starts at the bottom of all previously specified regions (i.e., at the maximum previously specified Y coordinate) and extends to the thickness in the positive Y direction. If no previous regions are specified, BOTTOM indicates the region starts at Y=0 and extends to the thickness in the positive Y direction.

Conventionally, we have found it convenient to start the active layers at the Y=0 location (as done in the standard examples). In ordering, REGION statements:

1. Start with the upper DBR by using the TOP parameter.
2. Use the TOP parameter on the top side contact layers.
3. Use the BOTTOM parameter on the active layers.
4. Use the BOTTOM parameter on the lower DBR.
5. Use the BOTTOM parameter on the substrate/lower contact regions.

Specifying Distributed Bragg Reflectors

A Distributed Bragg Reflector (DBR) is a periodic structure composed of regions of two alternating material compositions. A DBR can be specified by a series of REGION statements alternating between two different material composition. DBRs, however, typically consist of many of such layers and specifying them with REGION statements can be tedious, which could be prone to errors. You can use the DBR statement to simplify the specification instead. Generally, the DBR statement (alias SUPERLATTICE) can be used to specify any superlattice composed of layers of two alternating material compositions. For more information about this statement, see Chapter 19: "Statements", Section 19.6: "DBR".

The MAT1 and MAT2 parameters specify the material names of the two materials, which is used the same way the MATERIAL parameter of the REGION statement is used to specify a single material.

The X1.COMP, Y1.COMP, X2.COMP, and Y2.COMP parameters are used to specify the X and Y composition fractions of the two materials for Ternary and Quaternary materials.

The thicknesses of the two layers are specified by the THICK1 and THICK2 parameters of the DBR statement. The doping for the layers are specified by the NA1 and NA2 parameters for acceptors. The ND1 and ND2 parameters for donors.

The mesh spacing for the DBR can be specified using the SPA1 and SPA2 parameters, just like the SY parameter of the REGION statement. You can also use the N1 and N2 parameters to specify the integer number of the mesh divisions in the introduction of Y direction for each of the layers.

The total number of layers is specified by the integer HALF.CYCLE parameter. The HALF.CYCLE parameter specifies the total number of layers. Thus, an odd values specifies that one more layer of the material 1 is to be used than of material 2.

The TOP and BOTTOM parameters of the DBR statement are used just like the REGION statement, except the first material to be added to the top/bottom of the device is always of material 1.

DBR and REGION statements can be intermingled indiscriminately.

Specifying Oxide Apertures

Once you specify all the epitaxial layers of the VCSEL, you can then specify an oxide aperture by using a REGION statement with the MATERIAL parameter assigned to an insulator and specify the geometry by using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters.

Specifying Quantum Wells

The Multiple Quantum Well (MQW) models enable special models for gain and spontaneous recombination that account for quantum carrier confinement inside of the wells. These models are described in detail in Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.7: "Multiple Quantum Well Model".

The wells themselves are specified by using the MQW statement. The MQW statement should follow all the REGION and DBR statements and precede the ELECTRODE statements. For more information about this statement, see Chapter 19: "Statements", Section 19.30: "MQW".

The locations of the wells are specified by the XMIN, XMAX, YMIN, and YMAX parameters of the MQW statement. These parameters specify a bounding box in units of microns. XMIN and XMAX specify the extent of the wells in the X direction (i.e., the wells are horizontal and extend across the entire bounding box in the X direction). The YMIN and YMAX parameters act to define the center of the wells in the Y direction (i.e., the set of defined wells will be centered in the Y direction at the average coordinate of YMIN and YMAX. For example, if a single well is defined with YMIN=0 and YMAX=1.0, it will be centered at Y=0.5. If three wells were defined, the middle one would be centered at Y=0.5.

The WW parameter specifies the width of the wells in microns. The WB specifies the width of the barrier between wells in microns. The number of wells is specified by the NWELL parameter.

The material composition of the wells is specified by the MATERIAL, XCOMP, and YCOMP parameters. The MATERIAL parameter is assigned one of the valid material names described in Appendix B: "Material Systems". The XCOMP and YCOMP parameters are assigned to the X and Y composition fractions as necessary. The material composition of the barriers between wells is taken from whatever material regions that may have been defined coincident to the bounding box. This means, define the barrier material(s) the MQW with the REGION statements.

When MQW is specified, mesh lines will be added to the device mesh to resolve the locations of the edges of the wells and the bounding box. The SY parameter can also be specified to limit the maximum spacing between mesh lines within the bounding box.

When using MQW, set up a secondary mesh for the solution of the Schrodinger's Equation (See Chapter 13: "Quantum: Quantum Effect Simulator", Equation) in the wells to determine the bound state energies. This mesh is specified by the NX and NY parameters in the MQW statement. Typically, NY should be set to some large number so that there will be several samples per well. NX should be comparable to the number of grid lines in the device mesh over the same extent in the X direction.

Finally, there are several parameters that describe the details of the physical models. These parameters are described in detail in Chapter 13: "Quantum: Quantum Effect Simulator".

Specifying Electrodes

Device electrodes are specified using the ELECTRODE statement. Note that electrodes in device simulation are treated as boundary conditions. Therefore, there's no advantage to resolve the full outline of the electrode. Usually, the easiest way is to place electrodes at the top and bottom of the structure. To do this, specify the TOP or BOTTOM parameters of the ELECTRODE statement. The extent along the top or bottom of the device can be described by the X.MIN or X.MAX or both parameters. If you omit these parameters the electrode will extend all the way along the corresponding surface of the device.

Use the NUMBER parameter to assign electrodes to a number. Electrodes should be assigned consecutive numbers starting at 1. They can also be given a name by using the NAME parameter. They can also be addressed later by their name or number when modifying the electrode characteristics, using the CONTACT statement or assigning biasing information in the SOLVE statements.

9.3.2: Specifying VCSEL Physical Models and Material Parameters

Much of the simulation results depend on how the physical models are described. For more information about physical models, see Chapter 3: "Physics", Section 3.6: "Physical Models".

Selecting Device Models

For most laser devices including VCSELs, we advise that you at least enable predominant recombination mechanisms, since these usually define the threshold characteristics and compete with stimulated emission for carrier recombination. Thus, directly impact the efficiency of the device.

Typically, three bulk recombination mechanisms should be addressed: Schockley-Read-Hall (SRH), Auger, and Radiative Recombination. You can enable these mechanisms by specifying SRH, AUGER, and OPTR parameters on a MODEL statement.

The parameters of these models are specified on MATERIAL statements. These include TAUN and TAUP, which specify the SRH lifetimes. AUGN and AUGP, which specify the Auger coefficients. And COPT, which specifies the radiative recombination rate coefficient.

These models and associated parameters are discussed in more detail in Chapter 3: "Physics", Section 3.6.3: "Carrier Generation-Recombination Models".

Specifying Material Parameters

For LASER or VCSEL simulation there are several other material parameters that should be considered. First and foremost, specify the high frequency dielectric relative permittivity of each material region. This is specified by the EPSINF parameter in the MATERIAL statement.

The high frequency dielectric enters in the Helmholtz Equation and directly affects which laser mode may be present. In particular, try to consider the design of the DBRs carefully. Typically, the objective of the design of the DBRs is to choose the layer thicknesses so that they are one quarter wavelength thick relative to the local dielectric. During this design process, you need to also carefully consider the band-gap of the active region, since this affects the energy and wavelength of the maximal gain. A good design will try to operate near the peak gain, while satisfying the conditions for oscillation between the DBR mirrors.

Another important consideration for choosing material parameters is the selection of band edge parameters such as band gap and electron affinity. Although defaults do exist for many material systems it is almost always better for you to specify the best estimate. Also, consider the mobilities since they directly affect series resistance.

Chapter 3: "Physics", Section 3.6.1: "Mobility Modeling" has more information about these parameters. For more about MATERIAL statements, see Chapter 19: "Statements", Section 19.24: "MATERIAL".

9.3.3: Enabling VCSEL Solution

VCSEL Solution Mesh

The solution to Helmholtz equation is performed on a spatially discrete domain or mesh. Use of matrix method for the solution of the longitudinal wave equation allows the reduction in the number of mesh points in y (z in VCSEL) to the number of material boundaries. This significantly improves calculation time while maintaining the accuracy. The mesh points in the transverse direction x (r in VCSEL) coincide with the device mesh.

Specifying VCSEL Parameters

Specify the VCSEL statement to enable the VCSEL simulator. Once you enable it, the semiconductor device equations are solved self-consistently with the photon rate equations and the Helmholtz equation.

To enable reflectivity test simulation, do the following:

1. Specify the VCSEL.CHECK parameter in a VCSEL statement.
2. Specify the VCSEL.INCIDENCE parameter to monitor cold cavity reflectivity for light incident on a structure from either top or bottom of the structure.

- VCSEL.INCIDENCE = 1 - The light incident from the top.
- VCSEL.INCIDENCE = 0 - The light incident from the bottom.
- VCSEL.INCIDENCE = 2 or >2 - Both directions of light incidence are considered. The program compares cavity resonant frequencies obtained for each direction of incidence. If results do not agree within a certain tolerance, VCSEL:ATLAS will close with an error.

By default, light is incident from the top of the structure.

3. Specify INDEX.TOP for the refractive index of the medium above the structure.
4. Specify INDEX.BOTTOM for the refractive index of the medium below the structure. Default medium above and below the structure is air.
5. Specify the PHOTON.ENERGY or OMEGA parameters for (initial) photon energy or frequency.
6. Specify the EINIT and EFINAL paprameters. These parameters set the photon energy range in reflectivity test. If not specified, they take the following default values: EINIT=0.8 PHOTON.ENERGY and EFINAL=1.2 PHOTON.ENERGY.
7. Specify NSPEC for the number of sampling points between EINIT and EFINAL. The default value is NSPEC=100.

You can specify the following optional parameters in the VCSEL simulation:

- NMODE specifies the number of transverse modes of interest. Default value is NMODE=1.
- PROJ enables faster solution of the photon rate equations below threshold. You should disable this solution scheme when the bias reaches lasing threshold. Specify ^PROJ in a VCSEL statement to do that.
- PERTURB enables faster solution of the Helmholtz equation. When specified, the program solves the longitudinal wave equation only once. Perturbational approach is used to account for the refractive index changes [177].
- LOSSES specifies any additional laser losses.

9.3.4: Numerical Parameters

The following numerical parameters control the performance of the VCSEL simulation.

- TOLER sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.
- ITMAX sets maximum number of external VCSEL iterations during photon density calculation. The default value is 30.
- TAUSS is an iteration parameter used in the calculation of photon densities. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.
- MAXCH is the maximum allowed relative change of the photon density between VCSEL iterations. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.

9.3.5: Alternative VCSEL Simulator

If you need to model a vertical cavity laser that supports multiple longitudinal modes, use the LASER statement to enable VCSEL solution.

Specify the LASER statement to enable the laser simulator. Once you enable it, the semiconductor device equations are solved self-consistently with the laser rate equations. The major difference between this simulator and the one discussed above is in the approach to the solution of the Helmholtz equation.

The solver used in this case allows multiple longitudinal modes, but it is not as efficient in dealing with structural variations in the transverse direction.

There are several kinds of parameters that appear in the LASER statement. Most of the parameters available are discussed above with respect to VCSEL statement.

Specify laser physical parameters and models, and specify the DBR1.START, DBR1.FINAL, DBR2.START, and DBR2.FINAL parameters in the LASER statement.

The mirror loss is defined as:

$$\alpha_{mir} = \frac{1}{2L} ln\left(\frac{1}{R_f R_r}\right) \quad \text{9-20}$$

where L is the cavity length given as:

```
L=DBR2.START-DBR1.FINAL
```

R_f and R_r are the front and rear mirror reflectivities and are calculated from the solution of Helmholtz Equation in the range specified by DBR1.START, DBR2.START, and DBR2.FINAL.

LASER simulates single or multiple longitudinal modes as well as single or multiple transverse modes. Also with LASER, you can simulate index guiding only or gain guiding. Simulation of gain guiding takes more computation time but should be considered when spatially varying gain effects are anticipated.

If you to calculate the laser spectrum, specify the multiple longitudinal modes model and additional parameters. Do the following:

1. Specify the LMODES parameter in the statement. This enables the Multiple Mode Model.
2. Specify the EINIT and EFINAL parameters. These parameters set the photon energy range within, which LASER will take into account multiple longitudinal modes. Make sure initial photon energy is within this range and is specified for the active lasing region.
3. Specify the photon energy separation (ESEP parameter). If this isn't specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range. We recommend that you allow LASER to choose the photon energy separation.
4. Specify the spectrum file name (SPEC.SAVE). LASER will produce a structure file containing spectrum data after calculation of each bias point. LASER will automatically append _dcN.log to the specified file name (where N is the number of the bias point) for steady-state solutions or _trN.log for a transient simulation. The first bias point where LASER is active will have N=1. This is rarely the first bias point during simulation. Use TONYPLOT to examine these files. If you specify the SPECSAVE parameter, it will only save the spectrum files on every las.SPECSAVE solution.

The index in the spectrum file name will still increase by one each time. The spectrum data can be stored in a single file for the transient simulation, only if the MULTISAVE-LASER statement is set to FALSE. To enable multiple transverse mode solutions, specify the number of transverse modes by using the NMODE parameter from the LASER statement.

By default, VCSELs simulate using the Index Guiding Model. To simulate using the gain guiding, specify INDEX.MODEL=1.

You can save near and far field patterns by specifying a value for the PATTERNS parameter in the SAVE statement. The value of the PATTERNS parameter is a character string representing the root name of a file for saving the near and far field patterns. The near field pattern is saved to a file with the string .nfp appended to the root name. The far field pattern is saved to a file with the string .ffp appended to the root name. Use TONYPLOT to examine these files.

9.4: Semiconductor Laser Simulation Techniques

The most common technique for simulating laser diodes is to simulate forward characteristics with a gradually increasing bias. The forward voltage is usually specified. You can use, however, current boundary conditions and include external elements.

To save computational time, we recommend that you use the VCSEL statement. The effective frequency method employed for the solution of the Helmholtz equation in this case allows you to consider structures with oxide apertures or other structural variations in transverse direction. We also recommend that you test the structure using reflectivity experiment simulation prior to applying the bias.

If you use the multiple longitudinal mode model, computational time will be longer and strongly dependent on the number of longitudinal modes involved in the calculation.

Simulating multiple transverse modes takes more computation time and should only be preformed when the higher order modes are of interest.

Chapter 10:
Luminous: Optoelectronic Simulator

10.1: Overview

LUMINOUS is a general purpose ray trace and light absorption program integrated into the ATLAS framework, which is used to run with device simulation products. When used with S-PISCES or BLAZE, or device simulators, LUMINOUS will calculate optical intensity profiles within the semiconductor device, and converts these profiles into photogeneration rates. This unique coupling of tools allows you to simulate electronic responses to optical signals for a broad range of optical detectors. These devices include but are not limited to pn and pin photodiodes, avalanche photodiodes, Schottky photodetectors, MSMs, photoconductors, optical FETs, optical transistors, solar cells, and CCDs.

The following sections address various types of optoelectronic devices. Go to the sections that are most relevant to your application. We strongly recommend, however, that you read the other sections too.

Note: You should be familiar with ATLAS and either S-PISCES or BLAZE before you can use LUMINOUS. If not, read Chapter 2: "Getting Started with ATLAS" and either Chapter 4: "S-Pisces: Silicon Based 2D Simulator" or Chapter 5: "Blaze: Compound Material 2D Simulator".

10.2: Simulation Method

Optoelectronic device simulation is split into two distinct models that are calculated simultaneously at each DC bias point or transient timestep.

1.Optical ray trace using real component of refractive index to calculate the optical intensity at each grid point

2. Absorption or photogeneration model using the imaginary component of refractive index to calculate a new carrier concentration at each grid point.

This is followed by an electrical simulation using S-PISCES or BLAZE to calculate terminal currents.

Note: LUMINOUS assumes that the refractive indices, both real and imaginary, are constant within a particular region.

10.2.1: Ray Tracing in 2D

Defining The Incident Beam

An optical beam is modeled as a collimated source using the BEAM statement. The origin of the beam is defined by parameters X.ORIGIN and Y.ORIGIN (see Figure 10-1). The ANGLE parameter specifies the direction of propagation of the beam relative to the x-axis. ANGLE=90 is vertical illumination from the top. MIN.WINDOW/MAX.WINDOW parameters specify the illumination window. As shown in Figure 10-1, the Illumination Window is "clipped" against the device domain so that none of the beam bypasses the device.

The beam is automatically split into a series of rays so that the sum of the rays covers the entire width of the illumination window. When the beam is split, ATLAS automatically resolves discontinuities along the region boundaries of the device.

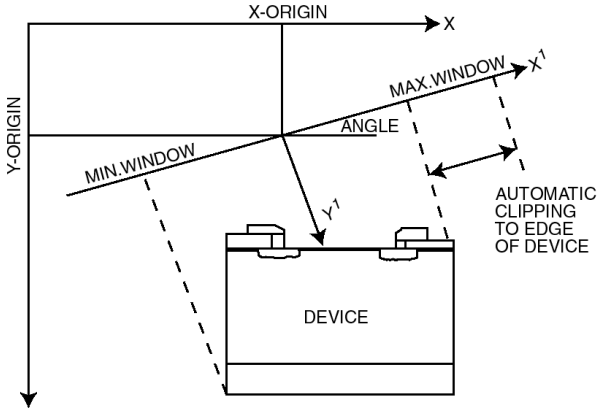


Figure 10-1: Optical Beam Geometry

Although the automatic algorithm is usually sufficient, you can also split the beam up into a number of rays using the RAYS parameter. Each ray will then have the same width at the beam origin and the sum of the rays will cover the illumination window. Even when you specify the RAYS parameter, ATLAS will automatically split the rays to resolve the device geometry.

Ray Splitting At Interfaces

Rays are also split at interfaces between regions into a transmitted ray and a reflected ray. Figure 10-2 illustrates the difference between rays that are split to resolve the geometry and transmitted/ reflected rays split at a region interfaces.

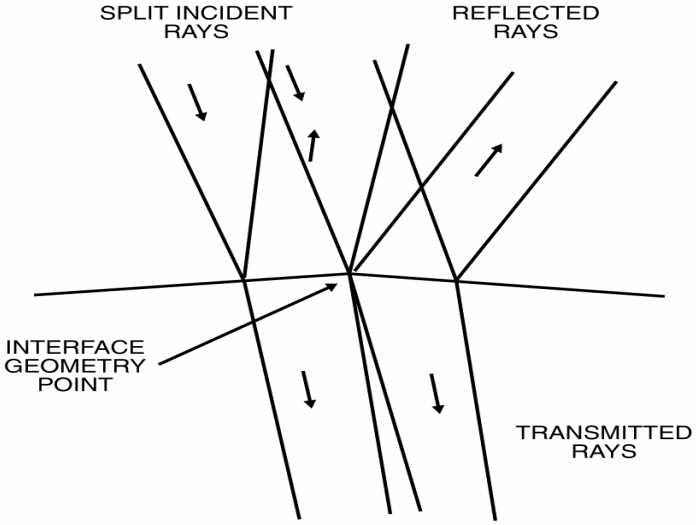


Figure 10-2: Reflected and Transmitted Rays

In Figure 10-2, the incident rays come in from the top left. They intersect a interface between two material regions with differing refractive indices. Within this interface lies a geometric point where the normal to the interface changes. This implies that the angles of reflection and transmission will be different for light incident to the left of the point from light incident on the right. Thus, the incident rays are split to resolve the interface point. The second level of splitting occurs at the interface itself. Here, the incident rays are split into reflected and transmitted rays.

10.2.2: Ray Tracing in 3D

In 3D, a simpler algorithm is used for ray tracing because the 2D algorithm takes up more computational time. Therefore, the algorithm doesn't automatically split rays to resolve topological features of the device.

Note: You must specify enough rays to resolve such features to the desired accuracy. But there's a trade off between the computation time and accuracy in specifying the number of rays.

The 3D source geometry is very similar to the geometry shown in Figure 10-1. In 3D, the source origin is specified by three coordinates: X.ORIGIN, Y.ORIGIN, and Z.ORIGIN. Unlike Figure 10-1, there are two angles that describe the direction of propagation. These are THETA, which describes the rotation of the X axis (THETA is an alias of ANGLE) and PHI, which describes the angle of rotation of Z axis.

In 3D, the clipping of the window of propagation is done in two directions. In the X direction, this is described by XMIN and XMAX. XMIN and XMAX are aliases of MIN.WINDOW and MAX.WINDOW. In the Z direction, the window is described by ZMIN and ZMAX.

The numbers of rays in the X and Z directions are described by NX and NZ. Ray samples are taken at regular intervals over the source window.

Figure 10-3 shows an example of a 3D ray trace, where TONYPLOT3D is used to visualize the rays.

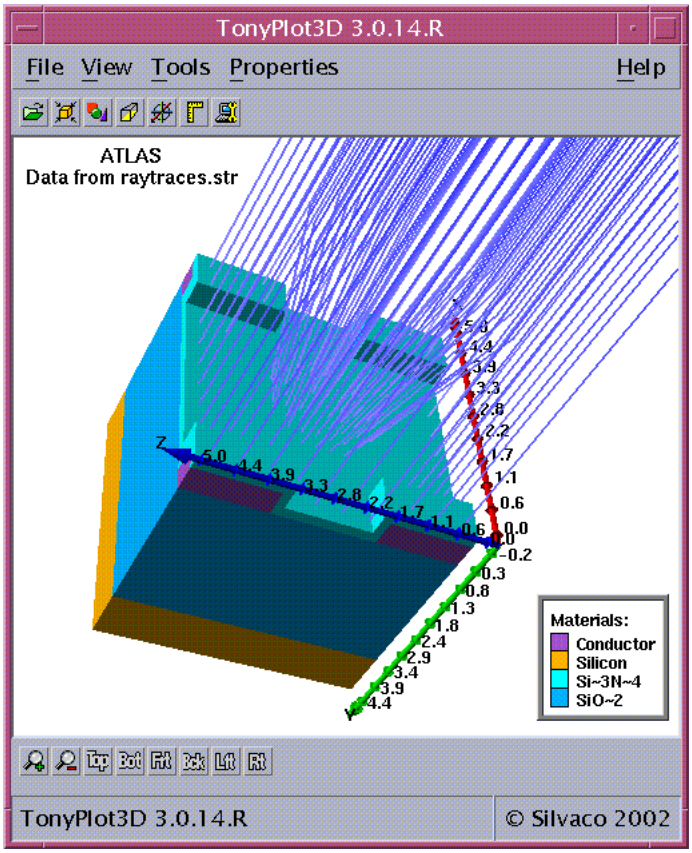


Figure 10-3: 3D Ray Trace

10.2.3: Reflection and Transmission

Figure 10-4 shows the relationship between the angles of incidence (θ_i), reflection (θ_r), and transmission (θ_t) at the interface between two media. These coefficients are calculated as a function of the refractive indices in the two media.

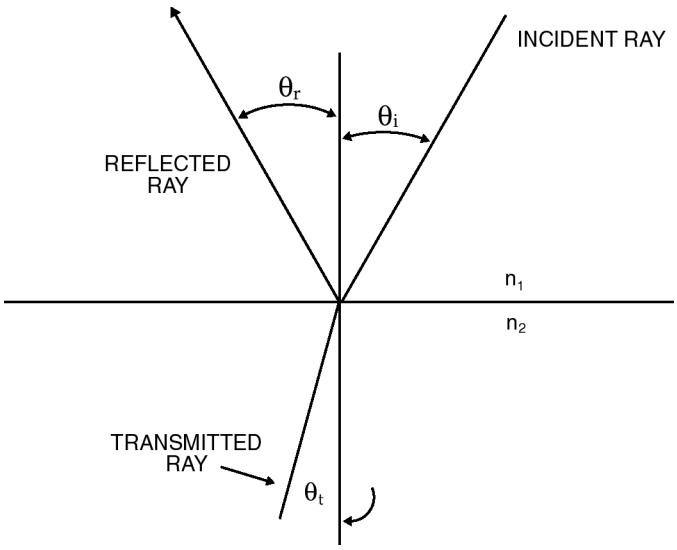


Figure 10-4: Angles of incidence, reflection and transmission

The reflection and transmission coefficients of the light for parallel and perpendicular polarization are calculated as shown in Equations 10-1 to 10-6.

$$E_r = \frac{n_2 \cos\theta_i - n_1 \cos\theta_t}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel polarization)} \tag{10-1}$$

$$E_t = \frac{2 n_1 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel polarization)} \tag{10-2}$$

$$E_r = \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular polarization)} \tag{10-3}$$

$$E_t = \frac{2 n_1 \cos\theta_i}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular polarization)} \tag{10-4}$$

$$R = \left(\frac{E_r}{E_i}\right)^2 \tag{10-5}$$

$$T = \left(\frac{E_t}{E_i}\right)^2 \frac{n_2}{n_1} \frac{\cos\theta_t}{\cos\theta_i} \tag{10-6}$$

where E_i is electric field of the incident wave, E_r is the field of the reflected wave, E_t is the field of the transmitted wave. R is the reflection coefficient, T is the transmission coefficient, n_1 is the refractive index on the incident side and n_2 is the refractive index on the transmission side.

The angles of reflection and transmission are given in Equations 10-7 and 10-8.

$$\theta_r = \theta_i \quad \text{10-7}$$

$$n_1 \sin\theta_i = n_2 \sin\theta_t \quad \text{10-8}$$

where θ_i is the angle of incidence, θ_t is the angle of transmission and θ_r is the angle of reflection.

Specifying Reflections

By default, no reflections are considered during the ray trace. The REFLECTS=<i> parameter is used to set an integer number of reflections to consider. Note that setting a very large number of reflections can lead to extremely long simulation times for the ray trace.

One convenient way to overcome the long CPU times is to use the MIN.POWER parameter. This terminates each ray when the optical power falls to the fraction of the original power defined by this parameter.

Front Reflection

By default, reflection and refraction at the first interface (the initial interface with the device) are ignored. The first reflection coefficient is zero and the transmission coefficient is one. The polarization and angle of the transmitted ray at the first interface is identical to the polarization and angle of the incident beam.

If the FRONT.REFL parameter of the BEAM statement is specified, the transmission coefficient is calculated using Equations 10-1 to 10-6. When the transmission coefficient is calculated, it's assumed that the material outside the device domain is a vacuum. The transmitted rays are attenuated by the transmission coefficient but the reflected ray is not traced.

Back Reflection

By default, the reflection at the back of the device are ignored. No reflected ray is traced once the back of the device is reached. If the BACK.REFL parameter is specified, the backside reflection coefficient is calculated (again assuming a vacuum outside the device) and the back-side reflected ray is traced.

Sidewall Reflection

By default, the reflection from the sides of the device are ignored. No reflected ray is traced back into the structure. As above, BACK.REFL is used to enable the sidewall reflections assuming a vacuum outside the device.

Discontinuous Regions

You can simulate devices electrically where a single region is defined as two or more separated areas. The ray tracing algorithm, however, doesn't support such structures. If a structure has two separate areas with the same region number, we recommend you using DEVEDIT to renumber the regions, perhaps even creating a new region number for each area.

Note: This limitation is only for two separated areas with the same region number and not for two regions with different region numbers of the same material. This latter case can be simulated.

Anti-Reflective Coatings

It's a popular strategy to place anti-reflective (AR) coatings on light detecting devices to improve device quantum efficiency. Such coatings rely on destructive interference of reflected waves to reduce overall reflection coefficient of light incident on the detecting device.

Simple AR coatings are typically composed of one layer of a transparent insulating material that is one quarter optical wavelength thick. Such coatings significantly reduce reflectivity of light at the design wavelength. Far from the wavelength in question, however, their performance is poor. Currently, more sophisticated multi-layer AR coatings are used for improved detection of broad-band light.

LUMINOUS enables you to model and design complicated AR coatings optimized for your specific application. AR coatings are simulated in LUMINOUS using an efficient transfer matrix method. This method enables you to deal with multi-layer AR coatings with virtually no overhead in computation time. Section 10.2.4: "Matrix Method" discusses matrix method and its application to wave propagation in a stratified medium.

Typically, AC coatings affect only optical properties of the device. Therefore, specifying the coating as a part of the device structure will unnecessarily increase the complexity of the electrical simulation. In ATLAS, coatings are associated with the interfaces of the device instead. This allows you to specify certain optical properties of any material boundary while not affecting electrical properties of the structure.

To define optical properties of a coating associated with that interface, use the `INTERFACE` statement and include the `OPTICAL` parameter. Using this statement, you can define an AR coating, a dielectric mirror, or an ideally reflecting surface. Note that the coating is absent in the structure defined for ATLAS.

The `AR.THICK` parameter defines the thickness of the coating layer. The `AR.INDEX` parameter defines the refractive index of the layer. For a single-layer coating or for the first layer of a multi-layer coating, specify the coordinates of the points defining the coated surface `P1.X`, `P1.Y`, `P2.X`, and `P2.Y`.

Note: These coordinates should define a line, not a rectangular box of thickness (AR.THICK).

Figure 10-5 shows an example of a single-layer coating. The following shows the syntax used for a single-layer coating.

```
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.0634 P1.X=0.0 P1.Y=0.0 P2.X=10.0
P2.Y=0.0
```

This defines a `63.4`μm layer of real refractive index `2.05` at `Y=0.0` μm in a structure.

For a single-layer coating or for the first layer of a multi-layer coating, you can also specify coordinates of the bounding box `XMIN`, `XMAX`, `YMIN`, `YMAX`. All interfaces within the bounding box obtain optical properties of the coating. If you don't set the bounding box coordinates, the default bounding box containing the whole device will be used.

The `REGION` parameter gives you additional flexibility in the description of an AR coating. `REGION` specifies the number of a region adjacent to the coating. `REGION=0` is the default value, corresponding to the ambient region outside the device boundaries.

When you define a bounding box for an interface, it could contain other interfaces, which you do not intend to be AR coated. When you use `REGION`, you can select only those interfaces adjacent to that region to have AR coating properties.

You can specify any number of coatings for each device. The `COATING` parameter in the `INTERFACE` statement refers to the number of the coating. If `COATING` parameter is not set, the first coating will be used. Coatings should be specified in order (i.e., `COATING=3` cannot be set before `COATING=2` is defined).

Different coatings should not overlap. If this occurs, the later coating will be used in the overlapping part.

Typically, the purpose of a single-layer AR coating is to minimize reflectivity of normally incident monochromatic light at the design wavelength λ.

Equations 10-9 and 10-10 define the parameters of choice for a single-layer AR coating between materials with refractive indexes n_1 and n_2:

$$\text{AR.INDEX} = \sqrt{n_1 n_2} \qquad 10\text{-}9$$

$$\text{AR.THICK} = \frac{\lambda}{4 \cdot \text{AR.INDEX}} \qquad 10\text{-}10$$

According to Equations 10-9 and 10-10, the coating in the example considered above has optimal properties for light at 520μm normally incident from air on a silicon detector.

You need several INTERFACE statements to specify a coating composed of multiple layers. Each statement defines only one layer of a coating. The syntax of the first INTERFACE statement for a multi-layer coating is identical to the specification of a single-layer coating. Each subsequent layer must have the number of the coating and the number of the layer specified by COATING and LAYER parameters. Specify the same COATING parameter for all INTERFACE statements that refer to the layers of the same AR coating. For example, the following statements specify a two-layer coating:

```
INTERFACE OPTICAL AR.INDEX=1.5 AR.THICK=0.06 P1.X=0.0 P1.Y=0.0 P2.X=10.0 \
                  P2.Y=0.0
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.06 COATING=1 LAYER=2
```

Here, COATING=1 and LAYER=2 show the second INTERFACE statement describing a second layer of a two-layer coating number one. Use the LAYER parameter to describe the order of layers in a coating. If LAYER is not specified, the first (top) layer will be used. Again, you cannot specify LAYER=3 before LAYER=2. Note that you only need to specify the coordinates of the interface for the first layer of each coating.

If a coating is made out of an absorbing material, you can use AR.ABSORB parameter to take into account absorption. You can also use totally reflective coatings. The coating will behave as an ideal reflector if you set the AR.INDEX parameter to a value > 1000.

Another way to specify the refractive index of the coating is to use the MATERIAL parameter on the INTERFACE statement. This enables you to apply any refractive index model supported by MATERIAL statement in ATLAS. This includes default wavelength dependent refractive index for supported materials, user-specified index in INDEX.FILE, C-interpreter function in F.INDEX, or the REAL.INDEX and IMAG.INDEX parameters on the corresponding MATERIAL statement. When you use the MATERIAL parameter on the INTERFACE statement, the AR.INDEX and AR.ABSORB parameters will be disabled.

An alternative way to set the properties of the top interface is to use C-INTERPRETER function with F.REFLECT specified in the BEAM statement. Using this function, you can specify the reflection coefficient, angle of transmission, and transmitted polarization as a function of position, wavelength, angle of incidence, and incident polarization.

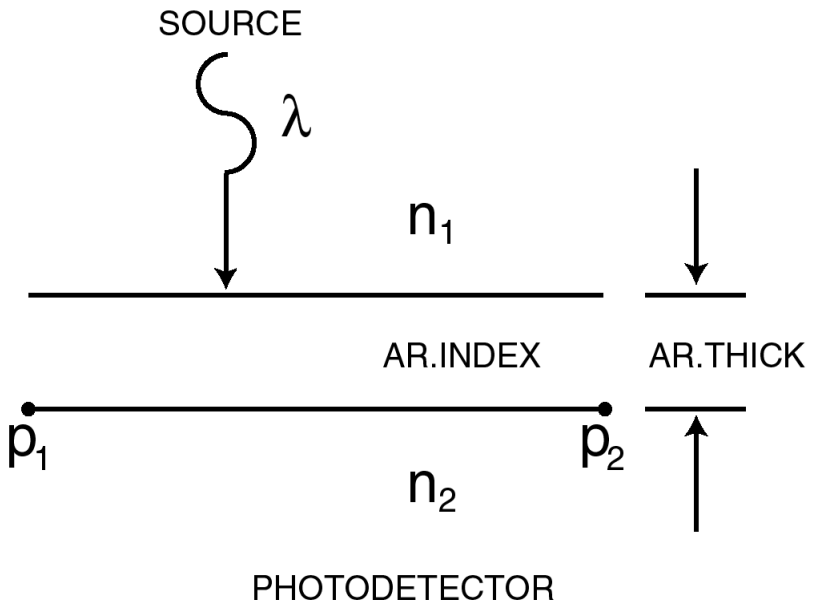


Figure 10-5: Single Layer AR Coating Under Normal Incidence

Anti-Reflective Coatings in LUMINOUS3D

LUMINOUS3D enables you to model and design multi-layer AR coatings optimized for your specific 3D application.

As in LUMINOUS, you can define properties of an AR coating using the INTERFACE statement with OPTICAL parameter in LUMINOUS3D. The AR.THICK, AR.INDEX, AR.ABSORB, LAYER, REGION, and COATING parameters retain their meaning and functionality in 3D. The only difference in description of an AR coating in 3D pertains to specification of the coating location. The coating location cannot be set by specifying point coordinates. For a single-layer 3D coating or for the first layer of a multi-layer 3D coating, you need to specify coordinates of the bounding box XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX.

The following example

```
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.0634 REGION=0
```

specifies that a 63.4μm layer of real refractive index 2.05 covers the entire outside boundary of a 3D device. All possible internal boundaries (heterojunctions) are not coated.

In a multi-layer coating, the first layer is assumed to be adjacent to the region specified by REGION.

```
INTERFACE OPTICAL AR.INDEX=1.5 AR.THICK=0.06 XMIN=0.0 XMAX=2.0 YMIN=0.0 \
          YMAX=1.0 ZMIN=0.0 ZMAX=6.0 REGION=2
INTERFACE OPTICAL AR.INDEX=2.05 AR.THICK=0.06 COATING=1 LAYER=2
```

In this example, the first coating layer with refractive index 1.5 is adjacent to the region number 2 on all boundaries within the specified bounding box. The second layer is next to the first layer at the same boundaries. Note that you only need to specify the bounding box and REGION only for the first layer of each coating.

10.2.4: Matrix Method

Matrix method presents a fast and accurate way to simulate electromagnetic wave propagation through a layered medium. Among a variety of approaches to matrix theory, the most commonly used are the characteristic matrix and the transfer matrix descriptions.

Using the matrix method is not limited to LUMINOUS only. It also finds its application in VCSEL and LED modules. ATLAS uses the characteristic matrix approach that relates total tangential components of the electric and magnetic fields at the multilayer boundaries. More often used transfer matrix, which relates tangential components of the electric field of left-travelling and right-travelling waves, has several disadvantages that limit its applicability.

The structure of a multilayer completely determines the characteristic matrix of this multilayer. The transfer matrix also contains information about the media on both sides of the multilayer. The characteristic matrix approach is more general and can be expanded to deal with graded interfaces, in which case, the transfer matrix method is inappropriate to use.

Characteristic Matrix

First, we describe the characteristic matrix of a single layer. As mentioned earlier, the matrix relates tangential components of the electric *E(z)* and magnetic *H(z)* fields at the layer boundaries *z=0* and *z=d*.

$$\begin{bmatrix} E(0) \\ H(0) \end{bmatrix} = M \begin{bmatrix} E(d) \\ H(d) \end{bmatrix} \quad \text{10-11}$$

The matrix itself is

$$M = \begin{bmatrix} \cos\varphi & j\sin\varphi / Y \\ jY\sin\varphi & \cos\varphi \end{bmatrix} \quad \text{10-12}$$

where

$$\varphi = \frac{2\pi}{\lambda} n d \cos\Theta \quad \text{10-13}$$

is the phase shift for the wave propagating through the layer, n is the complex refractive index, and Θ is the angle of wave propagation in the layer (Figure 10-6). Y is the optical admittance of the layer, which is for parallel (p) and perpendicular (s) polarizations, is given by:

$$Y^{(s)} = \sqrt{\frac{\varepsilon_0}{\mu_0}} n \cos\Theta \quad \text{10-14}$$

$$Y^{(p)} = \sqrt{\frac{\varepsilon_0}{\mu_0}} n / \cos\Theta \quad \text{10-15}$$

where ε_0 and μ_0 are permittivity and permeability of free space.

The characteristic matrix of a multilayer is a product of corresponding single layer matrices. If m is the number of layers, then the field at the first ($z=z_0$) and the last ($z=z_m$) boundaries are related as follows:

$$\begin{bmatrix} E(Z_0) \\ H(Z_0) \end{bmatrix} = M_1, M_2 \ldots M_m \begin{bmatrix} E(Z_m) \\ H(Z_m) \end{bmatrix}, M_i = \begin{bmatrix} \cos\varphi_i & j\sin\varphi_i / Y_i \\ jY_i\sin\varphi_i & \cos\varphi_i \end{bmatrix} \quad \text{10-16}$$

Reflectivity, Transmissivity, and Absorptance

The detailed derivation of amplitude reflection (r) and transmission (t) coefficients is given in [128]. The resulting expressions are shown in Equations 10-17 and 10-18:

$$r = \frac{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} - Y_s M_{22}}{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} + Y_s M_{22}} \quad 10\text{-}17$$

$$t = \frac{2Y_0}{Y_0 M_{11} + Y_0 Y_s M_{12} + M_{21} + Y_s M_{22}} \quad 10\text{-}18$$

where Y_0 and Y_s are the characteristic admittances of the media on both sides of the multilayer. M_{ij} are the elements of the characteristic matrix of the multilayer. Light is incident from the medium with admittance Y_0. The energy coefficients (reflectivity, transmissivity, and absorptance) are given by:

$$R = |r|^2 \quad 10\text{-}19$$

$$T = \frac{Re(Y_s)}{Re(Y_0)}|t|^2 \quad 10\text{-}20$$

$$A = (1-R)\left[1 - \frac{Re(Y_s)}{Re[(M_{11} + Y_s M_{12})(M_{21} + Y_s M_{22})^*]}\right] \quad 10\text{-}21$$

These expressions take into account imaginary part of refractive index. Therefore, they are more general than Fresnel formulae (Equations 10-1 through 10-6). LUMINOUS automatically uses these expressions when appropriate.

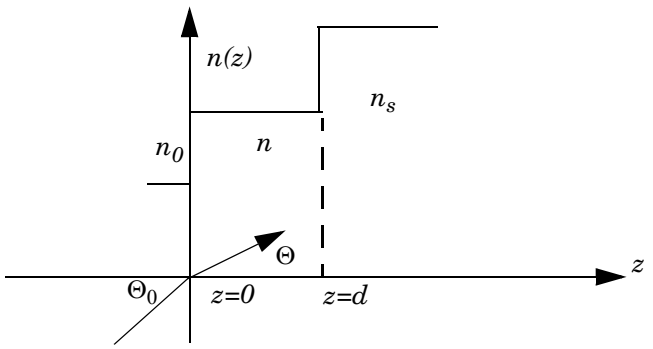


Figure 10-6: Refractive index profile of a single-layer coating

Transfer Matrix and Standing Wave Pattern

The characteristic matrix of a multilayer structure fully describes the optical properties of the multilayer. It is not completely, however, independent of external parameters. If the angle of incidence of light on a multilayer structure changes, the characteristic matrix changes too. Equations 10-13, 10-14, and 10-15 show the dependence of the single layer matrix elements on the angle of wave propagation in the layer. This implicit dependence on the angle of light incidence makes the characteristic matrix approach similar to the transfer matrix method for most practical purposes. In fact, when the multilayer structure does not contain any graded layers, the two methods are equivalent.

The transfer matrix directly relates the electric field amplitudes of transmitted and reflected waves to the amplitude of the incident wave. This property of the transfer matrix makes it the method of choice in many optics applications.

The transfer matrix elements are related to the characteristic martix elements by the following expressions:

$$Q_{11} = (M_{11} + M_{21}/Y_i + Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad 10\text{-}22$$

$$Q_{12} = (M_{11} + M_{21}/Y_i - Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad 10\text{-}23$$

$$Q_{21} = (M_{11} + M_{21}/Y_i + Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad 10\text{-}24$$

$$Q_{22} = (M_{11} + M_{21}/Y_i + Y_o M_{12} + Y_o/Y_i M_{22})/2 \quad 10\text{-}25$$

where Y_i and Y_o are the admittances of the input and output media defined according to Equations 10-14 and 10-15 for (s) and (p) polarization components respectively.

The amplitude reflection and transmission coefficients are:

$$r = Q_{21}/Q_{11} \quad 10\text{-}26$$

$$t = 1/Q_{11} \quad 10\text{-}27$$

In some devices that use multilayer structures, it is sufficient to know the transmittance and reflectance of the structure in order to fully characterize the device response to incident light. Other devices, however, such as thin film detectors, require the knowledge of light intensity distribution within the multilayer structure. Therefore in addition to the light transfer properties of such a structure, you need to be able to calculate the standing wave pattern formed by interference of traveling wave components in each layer. Standard intensity based ray-tracing calculations do not account for coherent effects and are unable to simulate standing wave patterns. Beam Propagation Method and full-wave Maxwell solvers are computationally expensive and are not useful for structures containing large number of layers. In this case, the numerical solution of the Helmholtz equation using the transfer matrix method is an attractive alternative.

Although this approach is inherently one-dimensional, it is known to produce highly accurate results for practical devices in two dimensions. Important requirements are:

- Lateral feature sizes in the device are much larger than typical layer thicknesses in a multilayer.
- Incident light is normal to the surface or is at small angles to the surface normal. When these requirements are met, the device can be devided into several lateral regions in which the structural variations occur in vertical direction z only.

For each lateral region, the one-dimentional Helmhotz equation is solved using the transfer matrix method.

First, we find reflected component of electrical field at *z=0*:

$$E_r = rE_i$$

where E_i and E_r are the amplitudes of incident and reflected waves respectively. The electric field and magnetic field at *z=0* are:

$$E(0) = E_i + E_r \quad \text{10-28}$$

$$H(0) = (E_r - E_i)Y_i \quad \text{10-29}$$

Then, the fields at layer interfaces can be found step by step using one layer characteristic matrices. The electric field at any point withis a certain layer is given by:

$$E_{(z)} = E(z_j)cos(\phi) - iH(z_j)sin(\phi)/Y(j) \quad \text{10-30}$$

where $E(z_j)$ and $H(z_j)$ are electric and magnetic fields at layer interface between layers *(j)* and *(j-1)*, *Y(j)* is the optical admittance of the layer *(j)* and phase is given by:

$$\phi = 2\pi n cos(\theta)(z - z_j)/\lambda \quad \text{10-31}$$

This way, intensity profile is generated throughout the device structure.

To enable transfer matrix method for calculation of intensity distribution and photogeneration profiles in thin film detectors, specify `TR.MATRIX` parameter on a `BEAM` statement.

10.3: Generation of Photocurrent

10.3.1: Light Absorption and Photogeneration

The cumulative effects of the reflection coefficients, transmission coefficients, and the integrated loss due to absorption over the ray path are saved for each ray. The generation associated with each grid point can be calculated by integration of the Generation Rate Formula (Equation 10-32) over the area of intersection between the ray and the polygon associated with the grid point.

$$G = \eta_0 \frac{P\lambda}{hc} \alpha e^{-\alpha y} \quad \text{10-32}$$

where:

- P is the ray intensity factor, which contains the cumulative effects of reflections, transmissions, and loss due to absorption over the ray path.
- η_o is the internal quantum efficiency, which represents the number of carrier pairs generated per photon observed.
- y is a relative distance for the ray in question.
- h is Planck's constant
- λ is the wavelength.
- c is the speed of light.
- α is the absorption coefficient given by Equation 10-33.

$$\alpha = \frac{4\pi}{\lambda} k \quad \text{10-33}$$

where k is the imaginary part of the optical index of refraction.

Photogeneration on a Non-uniform Mesh

The photogeneration algorithm used integrates the optical intensity around each node point. This is done to ensure that the total photogeneration rate isn't grid sensitive. A uniform photogeneration rate is defined as a constant value of (photogeneration rate at any node)*(element area around the node). In TONYPLOT, a uniform photogeneration rate may appear to vary across a non-uniform mesh density.

Photogeneration at Contacts

The photogeneration associated with nodes that are also defined as electrodes is a special case. The electrical boundary conditions require that the carrier concentration at electrode nodes equals the doping level. This means that photogeneration at nodes that are electrodes must be zero. But just setting these nodes to zero photogeneration will typically cause an apparent drop in quantum efficiency.

The photogeneration rate at the contact nodes is calculated as usual. This photogeneration rate, however, is applied to the neighboring node inside the semiconductor. This means for a uniform mesh and photogeneration rate, if the photogeneration rate is 1.0×10^{17} pairs/cm^{-3}s, then the nodes at the contacts will have zero photogeneration and the next node into the semiconductor will have 2.0×10^{17} pairs/cm^{-3}s.

User-Defined Arbitrary Photogeneration

An option exists for you to define the photogeneration rate. A C-INTERPRETER function written into a text file can be supplied to the program using the F.RADIATE parameter of the BEAM statement. For example, if a file, myoptics.c, was developed using the template C-INTERPRETER functions supplied, it can be referenced by using:

```
BEAM NUM=1 F.RADIATE=myoptics.c

.

.

SOLVE B1=1.0

SOLVE B1=2.0
```

The file, myoptics.c, returns a time and position dependent photogeneration rate to the program. This returned value is multiplied at every node point by the value of B1.

With this option, you override all other parameters of the BEAM statement and all the material refractive indices.

Another option is available if distribution of photo-injected carriers is uniform, linear, or has an exponential dependence on depth. Use the PHOTOGENERATE statement to specify the desired dependence on depth along a line segment.

The X.ORIGIN, Y.ORIGIN and X.END, Y.END parameters set the coordinates of the beginning and the end of the line segment. The default values correspond to the top left and bottom left corners of the device. Other parameters CONSTANT, EXPONENT, FACTOR, LINEAR, and RADIAL specify the photogeneration rate according to the following formula:

$$G(l, r) = (\mathrm{CONSTANT} + \mathrm{LINEAR} \cdot l + \mathrm{FACTOR} exp(-\mathrm{EXPONENT} \cdot l)) exp(-r^2 / \mathrm{RADIAL}^2) \qquad 10\text{-}34$$

where l is the distance along the line segment from the origin point and r is the radial distance from the line segment (Figure 10-7) .

PHOTOGENERATE statement ensures TMA compatibility with PHOTOGEN statement. See Chapter 19: “Statements”, Section 19.35: “PHOTOGENERATE” for the list of TMA compatible parameter names.

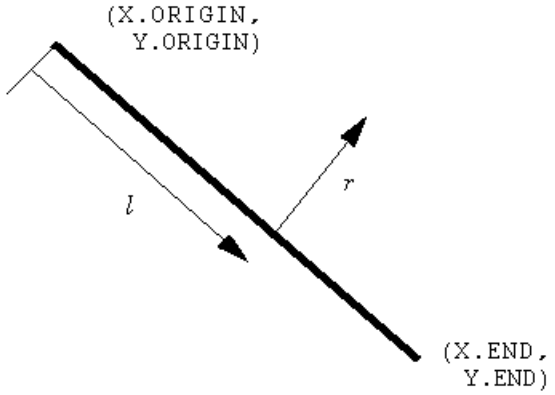


Figure 10-7: Specification of the Photogeneration Rate

10.3.2: Photocurrent and Quantum Efficiency

One of the important figures of merit of a photodetector is quantum efficiency. Here, quantum efficiency is defined as the ratio of the number of carriers detected at a given photodetector electrode divided by the number of incident photons on the detector. Ideally, this ratio should be 1.0 in detectors without any positive feedback mechanisms, such as avalanche gain. LUMINOUS doesn't directly calculate quantum efficiency. It does, however, calculate two useful quantities printed to the run-time output and saved to the log file. These quantities are source photocurrent and available photocurrent, which can be viewed in TONYPLOT from log files produced by LUMINOUS.

Definition of Source Photocurrent

The source photocurrent for a monochromatic source is given in Equation 10-35. Here, B_n is the intensity of the beam number n, which is user-definable in the SOLVE statement. λ is the source wavelength specified by the WAVELENGTH parameter in the BEAM statement. $\hbar$ is Planck's constant. c is the speed of light. W_t is the width of the beam including the effects of clipping (see Section 10.2.1: "Ray Tracing in 2D"). This can be considered as a measure of the rate of photons incident on the device expressed as a current density.

$$I_s = q\frac{B_n\lambda}{\hbar c}W_t \qquad 10\text{-}35$$

Definition of Available Photocurrent

The available photocurrent for a monochromatic source is given by Equation 10-36. Here, all the terms in front of the summation have the same definitions as for the source photocurrent. The sum is taken over the number of rays traced, N_R. W_R is the width associated with the ray. The integral is taken over the length, Y_i, associated with the ray. P_i accounts for the attenuation before the start of the ray due to non-unity transmission coefficients and absorption prior to the ray start. α_i is the absorption coefficient in the material that the ray is traversing.

The available photocurrent can be thought of as a measure of the photo absorption rate in the device expressed as a current density. This should be similar but somewhat less than the source photocurrent. The losses are due to reflection and transmission of light out of the device structure.

$$I_A = q\frac{B_n\lambda}{hc}\sum_{i=1}^{N_R} W_R\int_o^{Y_i} P_i\alpha_i e^{-\alpha_i y}\,dy \qquad 10\text{-}36$$

Depending how you define it, you can calcuate quantum efficiency by dividing the current from one of the device electrodes by either the source photocurrent or the available photocurrent.

10.4: Simulating Photodetectors

This section describes techniques simulate photodetectors. This section applies to the simulation of any of the following devices: p-n and p-i-n photodiodes, avalanche photodiodes, Schottky photodetectors, CCDs, MSMs, photoconductors, optical FETs and optically triggered power devices.

10.4.1: Defining Optical Sources

Identifying an Optical Beam

You can define up to ten optical sources. Optical sources are described by using the BEAM statement. All BEAM statements must appear somewhere after the MESH, REGION, DOPING, and ELECTRODE statements and before any SOLVE statement. The parameters in the BEAM statement describe a single optical source.

The NUM parameter is used to uniquely identify one optical source. Values between 1 and 10 are valid for the NUM parameter. Optical sources are subsequently referred to by the value of their NUM parameter. The power of the optical beam is set by using the B<**n**> parameter of the SOLVE statement, where **n** is the beam number defined by NUM.

Origin Plane of the Beam for 2D Optical Sources

To specify the origin of the optical source, use the X.ORIGIN and Y.ORIGIN parameters. These parameters describe the origin of the optical beam relative to the device coordinate system. Currently, it is required that the origin lie outside any device region. The ANGLE parameter specifies the angle of the direction of propagation of the beam with respect to the device coordinate system. ANGLE=90 specifies vertical (normal) illumination from above.

The width of the optical beam is specified using the MIN.WINDOW and MAX.WINDOW parameters. These parameters specify the limits of the source beam relative to the beam origin. If you omit either of the limits, that limit will be clipped to the edge of the device domain.

Note: It's extremely important that no section of the origin plane of the beam intersects or is inside the simulation grid. Otherwise, you'll get incorrect results. This is important to check in cases where the ANGLE isn't 90º or 270º.

Origin Plane of the Beam for 3D Optical Sources

To specify the origin of a 3D source, use the X.ORIGIN, Y.ORIGIN, and Z.ORIGIN parameters. These parameters describe the location of the origin relative to the device coordinate system. The THETA and PHI parameters describe the direction of propagation relative to the beam origin. The THETA parameter describes rotation of the X axis about the Z axis. The PHI parameter describes rotation of the Z axis about the Y axis. As with the 2D for normal illumination, specify a negative Y.ORIGIN and THETA=90.

Specify the window of illumination using the XMIN, XMAX, ZMIN, and ZMAX parameters. Then, specify the numbers of samples in the X and Z directions using the NX and NZ parameters. These values should be set to numbers large enough to resolve any salient features of the topology.

To save the rays into a file for visualization, specify the RAYTRACE=<filename> parameter. The components (u_x, u_y and u_z) of the direction vector representing the direction of the beam propagation are given by the following:

$$u_y = cos(\mathrm{PHI} - 90) \qquad 10\text{-}37$$

$$r = -sin(\mathrm{PHI} - 90) \qquad 10\text{-}38$$

$$u_x = r\cos(\texttt{THETA}) \quad \text{10-39}$$

$$u_z = r\sin(\texttt{THETA}) \quad \text{10-40}$$

Reflections

You can also specify whether to ignore the first reflection using the FRONT.REFL or the backside and sidewall reflection using the BACK.REFL. You should activate the backside reflections for devices, which use a back side reflector to improve collection efficiency. To set the number of reflections solved, use the REFLECTS parameter.

Typically, BACK.REFL should be used if the structure simulated is equivalent to the complete photodetector geometry as in a discrete device. If the simulation structure is a section of a larger substrate as in CCD simulation then don't use BACK.REFL.

Since the ray trace for arbitrary angles of incidence uses reflection and transmission coefficients, specify the polarization by using the POLARIZATION parameter.

In complex structures, you should limit the ray tracing to trace only those rays with significant optical power. The MIN.POWER parameter is used to terminate ray traces that drop below MIN.POWER* (optical source power).

Specifying Lenslets in 3D

In 3D, you can specify one of three lenslets to focus the light into the device. The lenslet has no direct effect on the device simulation mesh. It is only used in the ray trace. Lenslets are described by user defined parameters of the BEAM statement.

Spherical Lenslets

The spherical lenslet is described by the intersection of a plane (perpendicular to the Y axis and a sphere). The Y coordinate of the plane is given by LENS.PLANE. The origin of the sphere is described by LENS.X, LENS.Y, and LENS.Z. The sphere radius is described by LENS.RADIUS.

For ray tracing purposes, the space between the plane and the device surface is uniformly of a single index of refraction specified by LENS.INDEX. The spherical section projecting above the plane in the negative Y direction is also considered to consist of the same index of refraction. The plane and the spherical section form the lenslet.

Ellipsoidal Lenslets

To specify an ellipsoidal lenslet, define the axial half lengths using the X.SEMIAXIS, Y.SEMIAXIS and Z.SEMIAXIS parameters of the BEAM statement. The ellipsoidal lenslet center is specified by the LENS.X, LENS.Y and LENS.Z parameters. The LENS.PLANE parameter defines the location of the lens plane. The LENS.INDEX parameter gives the lens real index of refraction. Equation 10-41 gives the lenslet surface above the lens plane.

$$\frac{(x-\texttt{LENS.X})^2}{\texttt{X.SEMIAXIS}^2} + \frac{(y-\texttt{LENS.Y})^2}{\texttt{Y.SEMIAXIS}^2} + \frac{(z-\texttt{LENS.Z})^2}{\texttt{Z.SEMIAXIS}^2} \quad \text{10-41}$$

Composite Lenslets

You can also specify a composite lenslet. The composite lenslet is composed of a central planar section, quarter cylinder sections and eigth sphere sections, which in shown in Figure 10-8. The composite lenslet is defined by the LENS.XMIN, LENS.XMAX, LENS.ZMIN, LENS.ZMAX, LENS.WIDTH, LENS.HEIGHT, LENS.PLANE and LENS.INDEX parameters. Figure 10-8 shows the meanings of these parameters.

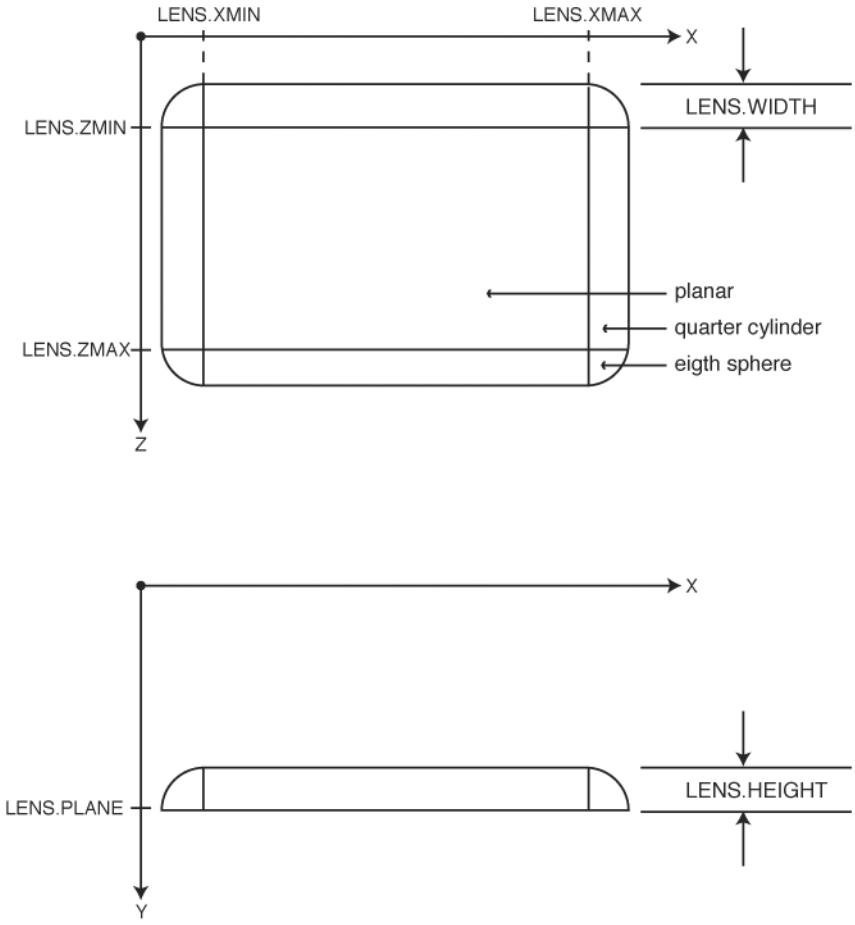


Figure 10-8: Composite Lenslet

Aspheric Lenslets

Aspheric lenslets are characterized in the normal directions x and z by the following two equations:

$$Y_x = \frac{X^2}{R_x + \left(R_x^2 - X^2\right)^{1/2}} + C_{4_x} X^4 + C_{6_x} X^6 + C_{8_x} X^8 + C_{10_x} X^{10} \quad \text{10-42}$$

$$Y_z = \frac{Z^2}{R_z + \left(R_Z^2 - Z^2\right)^{1/2}} + C_{4_z} Z^4 + C_{6_z} Z^6 + C_{8_z} Z^8 + C_{10_z} Z^{10} \quad \text{10-43}$$

To arrive at these equations, you must provide tablulated data in the form of Sag-h pairs describing the surface of the lenslet in the *x* and the *z* direction. You should specify the name of the file containing the data for the *x* direction in the `LENS.XSAGS` parameter in the `BEAM` statement. You should also specify the name of the file containing the data for the *z* direction in the `LENS.ZSAGS` parameter in the `BEAM` statement.

These files should first contain one line with the integer number of samples. The file should then contain a number (equal to the number of samples) of additional lines each containing a pair of real numbers representing the ordered pair of Sag-h data. These pairs must be arranged by increasing *h*. A sample at *h=0* is implicit and corresponds to *Sag=1*.

From this data, a Levenberg-Marquardt non-linear least squares algorithm is applied to obtain the constants *r*, c_4, c_6, c_8 and c_{10}. This algorithm is iterative. You can specify the number of iterations used in the SAG.ITS parameter on the BEAM statement with a default value of 10.

To have the results of the least squares fit written to files to be displayed in TONYPLOT, specify the file names using the OUT.XSAG and OUT.ZSAG parameters of the BEAM statement.

Other parameters on the BEAM statement describing the aspheric lenslet that should be specified include: LENS.X, LENS.Z, LENS.INDEX and LENS.PLANE.

Specifying Periodicity in the Ray Trace

By default, the unspecified boundary conditions for device simulation are mirror or periodic boundary conditions. For ray tracing, by default, the edges of the device are considered interfaces with a vacuum. Thus, at the edges of the device, by default, reflections are calculated. In some cases, it is convenient to consider the edges of the device to be periodic with respect to ray tracing. This is particularly true when the beam is not normal to the plane. You should specify PERIODIC in the BEAM statement to obtain such boundaries for ray tracing.

Defining Luminous Beam Intensity in 2D

When a beam is defined in LUMINOUS, it is, by default, assumed to have a uniform light intensity across the width of the beam. The intensity is defined in the SOLVE statement with the B<n> parameter and is in the units of W/cm^2.

The beam intensity distribution can also be defined with a Gaussian profile.

$$f(x) = \frac{1}{xsigma\sqrt{2\pi}} exp\left(-\frac{(x-mean)^2}{2(xsigma)^2}\right) \quad \text{10-44}$$

The location of the peak of the Gaussian distribution with respect to the beam coordinates is specified by the MEAN parameter of the BEAM statement. The standard deviation of the distribution is specified with the SIGMA parameter.

x is the cumulative distance over the beam width where the distribution will be formed. Define the RAYS parameter to get smooth distribution of intensity profile. This determines the increment of *x*. Therefore, the RAYS parameter should be set to a large number as the spacing between rays is always a constant (width of the beam /RAYS).

The resultant distribution has a unity peak value, which is then scaled by the B<n> value specified on the SOLVE statement. For example, the following BEAM statement will define a beam window 2 μm wide centered at X.ORIGIN and Y.ORIGIN, which has a Gaussian peak of 0.01 W/cm^2 and a standard deviation of 0.05 μm.

```
beam   num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=0.6 \
xmin=-1   xmax=1 GAUSSIAN   MEAN=0   SIGMA=0.05 RAYS=200 \
SOLVE B1=1E-2
```

Defining Luminous3D Beam Intensity

A beam defined in LUMINOUS3D is assumed by default to have a uniform light intensity across the width of the beam. The intensity is defined in the SOLVE statement with the B,n. parameter and is in the units of W/cm^2.

In LUMINOUS3D, you can also specify a circular or elliptical source. To specify an elliptical source, specify the appropriate origin location using X.ORIGIN, Y.ORIGIN, and Z.ORIGIN. This will specify

the center of the elliptical source. The major and minor axes of the ellipse are specified by the XRADIUS and ZRADIUS parameters. Remember, you must also specify the NX and NZ parameters. These will sample uniformly across the diameters of the ellipse.

You may also specify Gaussian profiles in LUMINOUS3D. To do this, specify the location of the beam origin. You can specify the Gaussian(s) in the X and Z direction. These can be done independently or together. These Gaussians are specified by the XMEAN, XSIGMA, ZMEAN, and ZSIGMA parameters of the BEAM statement.

Monochromatic or Multispectral Sources

The optical source can be either monochromatic or multispectral. For monochromatic sources, you can use the WAVELENGTH parameter to assign the optical wavelength. WAVELENGTH uses the units microns to be more consistent with the rest of ATLAS. Note this if you're accustomed to the optoelectronic engineering preference for nanometers.

For multispectral sources, spectral intensity is described in an external ASCII file indicated by the POWER.FILE parameter. This is a text file that contains a list of pairs defining wavelength and spectral intensity. The first line of the file gives the integer number of wavelength-intensity pairs in the file. An example of such a file is shown below.

```
4
0.4 0.5
0.5 1.0
0.6 1.2
0.7 1.1
```

This example specifies that there are four samples and that at a wavelength of 0.4 μm, the intensity is 0.5 Watts per square cm per μm of wavelength, and so on. With multispectral sources, specify a discretization of the interpolated information. Values must be specified for the WAVEL.START, WAVEL.END, and WAVEL.NUM parameters. These values specify the starting and ending wavelengths and the number of wavelengths to sample. LUMINOUS uses wavelengths at equal intervals over a specified range of wavelengths.

If you don't specify the values of WAVEL.START, WAVEL.END, and WAVEL.NUM, these parameters take on the corresponding values from the specified POWER.FILE. For the example file shown above, LUMINOUS uses the following default values of these parameters: WAVEL.START=0.4, WAVEL.END=0.7, and WAVEL.NUM=4.

LUMINOUS performs an independent ray trace at each of the sample wavelengths. For example:

```
WAVEL.START=0.4 WAVEL.END=0.6 WAVEL.NUM=2
```

causes ray traces at wavelengths of 0.45 and 0.55. LUMINOUS obtains the intensity associated with each sample by integrating the values of the spectral intensity file using a piece wise linear approximation. Each integral is performed over the range between successive midpoints. In the preceding example, the integration for the sample at 0.45 would be performed over the range of 0.4 to 0.5.

For a multispectral source, the generation rate (like Equation 10-32) is given by:

$$G = \eta_0 \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} \frac{P(\lambda)L\lambda}{hc}\alpha e^{-\alpha y} d\lambda \qquad \text{10-45}$$

where:

- η_0 is the internal quantum efficiency.

- $P(\lambda)$ is the power spectral density of the source.
- L is a factor representing the cumulative loss due to reflections, transmissions, and absorption over the ray path.
- λ is the wavelength.
- $\hbar$ is Planck's constant.
- c is the speed of light.
- α is the absorption coefficient given by Equation 10-33.
- y is the depth of the device, where x,y forms the two-dimensional mesh.

WAVEL.START and WAVEL.END are the spectral limits specified on the BEAM statement.

Note: The integral in Equation 10-45 may be inexact due to the discrete sampling in conjunction with wavelength dependence of the absorption coefficient. For constant absorption coefficient, the integral is exact of the number of discrete wavelength samples specified by WAVEL.NUM on the BEAM statement.

The source photocurrent (like Equation 10-35) is given by:

$$I_s = q\frac{B_n}{\hbar c}W_t \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} P(\lambda)\lambda d\lambda \qquad \text{10-46}$$

where $P(\lambda)$ is the power spectral density of the source. The other parameters have the same definition as in Equation 10-35.

The available photo current (like Equation 10-36) is given by:

$$I_A = q\frac{B_n}{\hbar c}\sum_{i=1}^{N_R} W_R \int_0^{X_i} \int_{\text{WAVEL.START}}^{\text{WAVEL.END}} P(\lambda)\alpha_i e^{-\alpha_i y} P_i \lambda dy d\lambda \qquad \text{10-47}$$

where $P(\lambda)$ is the power spectral density of the source. The other parameters have the same definition as in Equation 10-36.

LUMINOUS allows you to chose the units of spectral intensity in an input spectrum file. By default, the units of spectral intensity in the POWER.FILE are $W/(cm^2\ \mu m)$. In this case, LUMINOUS carries out the integration according to Equations 10-45, 10-46, 10-47. If the units of spectral intensity in the POWER.FILE are W/cm^2, then you need to use the flag ^INTEGRATE on the BEAM statement. In this case, each value $P(\lambda)$ in the second column of the file is a total intensity for a corresponding spectral interval. Integrals over spectrum in Equations 10-45, 10-46, 10-47 are reduced to summations of $P(\lambda)$ values. In both cases, LUMINOUS treats the subsequently specified value of B<n> on the SOLVE statement as a unitless multiplier or scale factor for intensity.

This is different from monochomatic case where this parameter has units of W/cm^2. To preserve consistency with monochromatic case, you can also do the same in the multispectral applications. If you specify the total beam intensity for multispectral sources using B<n> parameter on the SOLVE statement, you need to set NORMALIZE flag on the BEAM statements. This ensures that the intensity spectrum is normalized and B<n> has units of W/cm^2.

For either the monochromatic or multispectral sources, you can uniformly scale the wavelength(s) and intensities by using the WAVEL.SCALE parameter and the POWER.SCALE parameter respectively. These parameters are useful if the intensities or wavelengths are specified in units other than the default units.

User-Defined Beam

Standard beam input syntax in LUMINOUS and LUMINOUS3D allows specification of plane waves with Gaussian or flat-top (top-hat) irradiance profiles. Although many practical applications require only plane wave illumination, some optoelectronic devices rely on certain focusing or collimating optical systems that collect incident light. Some of these optical systems are implemented as lens arrays deposited on top of the device structure, which can be directly included in LUMINOUS3D. Still, a number of applications that require complex collimating or focusing optics cannot be approached in this way. To accomodate these applications, we enabled input of incident beams with complex wavefronts. Currently, the ray-tracing is not restricted to particular beam properties and allows specification of an arbitrary collection of rays as an input beam.

Theoretically, a number of ways to specify a beam of arbitrary shape exists. The one chosen in ATLAS aims to allow you an easy interface with optical system design software. Thus, collimators or focusing optics designed with conventional optical system tools can be represented in ATLAS by a ray bundle obtained on the output of the optics tool. ATLAS treats the ray bundle specified in a file as one beam. You need to set the input file name on the BEAM statement as a value of INPUTRAYS parameter.

The file INPUTRAYS must have the following format.

```
NUMBER OF RAYS
NUMBER OF CHANGING PARAMETERS
PARAMETER1 PARAMETER2 PARAMETER3 ...
RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
...              ...                  ...                   ...
RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
```

NUMBER OF RAYS stands for the total number of rays specified in the file. It should be the same as the last RAY NUMBER in the table.

NUMBER OF CHANGING PARAMETERS gives the total number of ray parameters given in the table. Most often, some parameters are the same for all the rays. For example, wavelength or ray power can be the same. In this case, there is no need to keep the table column for this constant parameter. Instead, you can set this parameter directly in the BEAM statement. NUMBER OF CHANGING PARAMETERS is equal to the number of columns in the table (not counting the first column of ray numbers).

PARAMETER1, PARAMETER2, PARAMETER3, and so on are numerical codes of parameters. Their order specifies the order of columns in the table. Numerical codes correspond to the following parameters.

- 1 - x coordinate of ray origin (in device coordinate system).
- 2 - y coordinate of ray origin (in device coordinate system).
- 3 - z coordinate of ray origin (in device coordinate system).
- 4 - ray angle phi in XY plane (0 - along X and 90 - along Y).
- 5 - ray angle theta in XZ plane (in 3D only, 0 - along X and 90 - along Z).
- 6 - ray polarization angle (relative to the plane of incidence, 0 - TM, and 90 - TE).
- 7 - ray relative power.
- 8 - ray wavelength.
- 9 - ray area (μm^2 in 3D) or thickness (μm in 2D).

The first column of the table where ray parameters are specified lists ray numbers (integer values) in ascending order. The first ray number should be 1. The last ray number should be equal to the NUMBER OF RAYS. Often, it is the case that ray parameters change continuously from ray to ray (at least in parts of the table). The INPUTRAYS file in ATLAS allows a shorthand input for these parts. You can omit ray numbers where possible to allow for automatic fitting of parameter values for these rays. Linear approximation is used for the omitted rays based on the values of parameters given in the table. This is better illustated in the following example.

INPUTRAYS File Example

```
21
3
1 4 7
1   0.0    60.0   0.2
11  300.0  90.0   1.0
21  600.0  120.0  0.2
```

Here, NUMBER OF RAYS=21, NUMBER OF PARAMETERS=3, and the order of parameter columns: x coordinate (1), angle phi (4), and relative ray power (7).

The rays from 2 to 10 are specified using linear interpolation with end values given by rays 1 and 11. The rays from 12 to 20 are also defined similarly. This shorthand input is equivalent to the following.

```
21
3
1 4 7
1   0.0    60.0   0.2
2   30.0   63.0   0.28
3   60.0   66.0   0.36
4   90.0   69.0   0.44
5   120.0  72.0   0.52
6   150.0  75.0   0.6
7   180.0  78.0   0.68
8   210.0  81.0   0.76
9   240.0  84.0   0.84
10  270.0  87.0   0.92
11  300.0  90.0   1.0
12  330.0  93.0   0.92
13  360.0  96.0   0.84
14  390.0  99.0   0.76
15  420.0  102.0  0.68
16  450.0  105.0  0.6
17  480.0  108.0  0.52
18  510.0  111.0  0.44
19  540.0  114.0  0.36
20  570.0  117.0  0.28
21  600.0  120.0  0.2
```

When you use the INPUTRAYS file to define the beam, some of the parameters on the BEAM statements are disabled because they become incompatible with the user-defined beam. The parameters that are ignored in this case are CIRCULAR, ELLIPTICAL, GAUSSIAN, all LENS.* parameters, MAX.WINDOW, MIN.WINDOW, MEAN, RAYS, SIGMA, X.CENTER, X.GAUSSIAN, X.MEAN, X.RADIUS, X.SIGMA, X.SEMIAXIS, XMAX, XMIN, Y.SEMIAXIS, Z.CENTER, Z.GAUSSIAN, Z.MEAN, Z.RADIUS, Z.SIGMA, Z.SEMIAXIS, ZMAX, and ZMIN.

The X.ORIGIN, Y.ORIGIN, Z.ORIGIN, PHI, THETA, POLARIZE, REL.POWER, WAVELENGTH, RAYAREA parameters in the BEAM statement specify values for all rays if the corresponding parameters are not defined explicitly in the INPUTRAYS file. The parameters Z.ORIGIN and THETA and the corresponding parameters in INPUTRAYS are for LUMINOUS3D applications only. They are ignored in LUMINOUS2D.

RAYAREA is a parameter that specifies area in μm^2 (thickness in 2D in µm) of each ray in the BEAM statement. This parameter is used when the specified INPUTRAYS file does not contain ray area information.

One difficulty that arises when dealing with arbitrary beams has to do with specification of beam intensity (W/cm^2) in the SOLVE statement. For a general ray bundle, illuminating the device the intensity distribution is not uniform. Thus, intensity cannot be specified by a single number. There are two ways to avoid this problem. Instead of setting intensity for a ray bundle obtained on the output of an optical system (collimator or focusing system), you can go back to the original beam that produced this ray bundle. According to the intensity law of geometrical optics [27], you can define ray area (thickness) and overall beam intensity for that input beam. Alternatively, if power (W) carried by each ray is known, you can treat beam intensity parameter B<n> in the SOLVE statement as a scaling factor. Then, set all ray power values using the ray relative power parameter in the INPUTRAYS file (or REL.POWER on the BEAM statement if it is the same for all rays). In this case, you can treat RAYAREA as another scaling parameter for unit conversions. In any case, make sure you define the ray area because it's important for obtaining correct values of photogeneration rates. If you don't specify ray area, ATLAS uses an estimate based of coordinate values of ray origins. You should not rely, however, on the accuracy of such an estimate.

You can use INPUTRAYS with POWER.FILE to define multi-spectral complex beams. In this case, ATLAS ignores ray wavelength specified in INPUTRAYS and uses the wavelength values from POWER.FILE for each ray. Ray power is then a product of relative ray power from INPUTRAYS and spectral power density defined in POWER.FILE.

10.4.2: Defining Optical Properties of Materials

For ray tracing, the complex index of refraction of the various material regions in the structure must be specified. For many of the more common semiconductors and insulators, there are built-in tables of index versus wavelength. You can specify the index for the materials lacking reasonable default complex index of refraction.

Note: You can add the INDEX.CHECK parameter to any SOLVE statement to print out the refractive indices being used for that bias step. The indices are only printed when you perform the ray trace at the first reference to a given beam on the SOLVE statement.

Setting Single Values For The Refractive Index

The REAL.INDEX and IMAG.INDEX parameters of the MATERIAL statement can be used to set the real and imaginary indices of a specified material, region or regions. For example, the statement:

```
MATERIAL MATERIAL=Air REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all material regions composed of "Air"

The statement:

```
MATERIAL REGION=1 REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for region number 1.

The statement:

```
MATERIAL REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all regions.

Setting A Wavelength Dependent Refractive Index

The preceding examples set the complex index of refraction for a material regardless of wavelength. This is probably adequate for monochromatic sources. For multispectral simulations, the index of refraction should be modeled as having a dependence on wavelength. There are two ways to do this.

ASCII File Input

The first way is to specify index versus wavelength in a file. This is a text file that contains ordered triplets of wavelength, real index, and imaginary index. The first entry in the table is the number of samples. If you set the INDEX.FILE parameter of the MATERIAL statement to the name of the index file, the linear interpolation from this table will complete to obtain the index of refraction as a function of wavelength.

A valid index file is shown below.

```
2
0.5 2.0 0.0
0.6 3.0 0.02
```

In this example, a real index of 2.5 and an imaginary index of 0.015 would be used for a wavelength of 0.55 microns.

C-Interpreter Function

The second way to specify index versus wavelength is by using the C-INTERPRETER function. The syntax is:

```
MATERIAL NAME=Silicon F.INDEX=myindex.c
```

The file myindex.c is an external file conforming to the template supplied with the program. It returns wavelength dependent real and imaginary indices. Instructions for using the C-INTERPRETER and finding the template functions are described in Appendix A: "C-Interpreter Functions".

10.4.3: Extracting Dark Characteristics

One of the first tasks in analyzing a new detector design is to examine dark current, device capacitance, and possibly other unilluminated characteristics. This can normally be done without using LUMINOUS. Extraction of the characteristics is adequately covered in the chapters on S-PISCES or BLAZE.

The extraction of reverse bias leakage currents for diodes presents some difficult numerical problems for device simulators. These problems are associated with limitations on numerical precision. ATLAS, as well as most other available device simulators, use double precision arithmetic to evaluate terminal currents. Double precision arithmetic provides roughly 16 decimal digits of precision. Internal scaling allows the measurement of currents down to a level of between about 10^{-12} A/micron to 10^{-16} A/micron. Unfortunately, photodiode leakage currents are often around or below this level. This means that the currents printed are subject to significant numerical noise and don't provide an accurate estimate of the device leakage currents. There are two ways to estimate reverse leakage current.

Integrated Recombination

From a theoretical standpoint, the reverse behavior of diodes can be dominated by one of two effects: diffusion currents in the neutral regions or recombination currents inside the depletion region [156]. ATLAS can provide insight into both of these contributing mechanisms. To estimate recombination current, use the MEASURE statement to calculate the integrated recombination rate. The following statement can be used:

```
MEASURE U.TOTAL
```

When this statement is executed, it prints out the total integrated recombination rate. Multiply this value by the electron charge (1.6023×10^{-19} coulombs) to obtain an estimate of the recombination current contribution to the reverse diode leakage current.

Extrapolation from High Temperatures

The diffusion current contribution can be estimated by taking advantage of the non-linear relationship between the diffusion current and temperature. Referring to the expression for the "Ideal Diode" current given by Equation 10-48, the dominant temperature dependency arises from the variation of the intrinsic concentration.

$$J = \left(\frac{qD_p p_{n0}}{L_p} + \frac{qD_n n_{p0}}{L_n}\right)\left(exp\left(\left(\frac{qV}{kT_L}\right) - 1\right)\right) \quad \text{10-48}$$

where n_{p0} and p_{n0} are thermal-equilibrium minority carrier densities on either side of the junction. This gives an exponential variation of the diffusion current with temperature given in Equation 10-49.

$$J \approx exp\left(\frac{-E_g}{kT_L}\right) \cdot \left(exp\left(\frac{qV}{kT_L}\right) - 1\right) \quad \text{10-49}$$

This relation can be used to estimate the diffusion current contribution at the operating temperature. The basic idea is to calculate the current at a high temperature where the problem of numerical precision does not arise. Then, scale the current to the operating temperature using Equation 10-49. For example, if the device is to operate at 300K, you can set the temperature to 450K using the TEMPERATURE parameter of the MODEL statement. Disable any temperature dependence of the energy gap by specifying the band gap using the EG300 parameter and setting EGALPHA and EGBETA parameters to zero, which are all on the MATERIAL statement. The following statement illustrates this approach as it might apply to a silicon diode:

```
MODEL TEMPERATURE=450
MATERIAL EG300=1.12 EGALPHA=0.0 EGBETA=0.0
```

ATLAS can then be used to obtain the reverse bias current at the elevated temperature. Equation 10-50 can be applied to obtain the depletion current contribution at the operating temperature:

$$J = J_e \cdot exp\left(\frac{E_g}{kT_e} - \frac{E_g}{kT_L}\right) \cdot \left(\frac{exp\left(\frac{qV}{kT_L}\right) - 1}{exp\left(\frac{qV}{kT_e}\right) - 1}\right) \quad \text{10-50}$$

where J_e is the current measured at the elevated temperature, E_g is the bandgap, T_e is the elevated temperature, T_L is the operating temperature, V is the operating bias voltage, and J is the current estimate at the operating temperature. Once you've obtained estimates of the recombination and diffusion contributions, you can obtain the total leakage current by summing the two contributions.

Numerical Solution Parameters

ATLAS uses a cut-off value of carrier concentration below, which solutions are not required to converge. This limit is set by the CLIM.DD. parameter. See Chapter 18: "Numerical Techniques" for more details on CLIM.DD. For photodetectors, you often need to reduce CLIM.DD to 10^5 in order to resolve carrier concentrations in depleted regions before illumination.

10.4.4: Extracting Detection Efficiency

One of the simpler tasks in characterizing a photodetector design is to measure DC detection efficiency. This will be done typically as a function of bias voltage, optical intensity, or wavelength. Each of these analyses can be performed using the SOLVE statement. The Bn parameter can be used to set the optical intensity of the optical sources described in the previous section. The following example illustrates obtaining a solution with a specified optical intensity:

```
SOLVE B1=1.0
```

This specifies that a solution is to be obtained for an optical intensity in the beam numbered "1" of 1.0 Watt/cm^2. If this were the first SOLVE statement specified, the ray trace in LUMINOUS would be initiated. At the start of the solution, the optical intensities of each optical source with a positive intensity is printed. In addition, the available photocurrent and source photocurrent are printed. See the prior section on Photocurrent for a definition of these two quantities.

Internal and External Quantum Efficiency

The available photocurrent divided by the source photocurrent is a measure of the external quantum efficiency of the detector. The calculated terminal current can be divided by the source or available photocurrents is used to evaluate the internal quantum efficiency of the device.

10.4.5: Obtaining Quantum Efficiency versus Bias

The intensities specified in the SOLVE statement apply until another SOLVE statement changes the intensity of the beam. Sequences of SOLVE statements can be used to vary the optical intensity at arbitrary intervals. The simple linear ramps of optical intensity can be abbreviated using the LIT.STEP and NSTEP parameters of the SOLVE statement. The LIT.STEP parameter specifies the size of the DC step and NSTEP specifies how many steps are desired.

Another option for analyzing DC quantum efficiency is to fix the optical intensity and vary bias voltages. The bias voltages can be varied in arbitrary discrete steps using several SOLVE statements, or in a linear ramp using individual SOLVE statements. This is useful for determining the optimum operating bias of devices such as avalanche detectors and photo transistors.

10.4.6: Obtaining Transient Response to Optical Sources

It is sometimes desirable to examine the time domain response of a detector to time-dependent (e.g., ramped or pulsed) optical sources. LUMINOUS provides this capability with the RAMP.LIT parameter. When the RAMP.LIT parameter is specified in a SOLVE statement, the optical intensity is changed linearly from the most recently set intensity to the intensity set in the B parameter. If a particular source intensity is not set using the corresponding B parameter, its intensity is not varied during the transient.

The period of the linear ramp is specified by the RAMPTIME parameter. The transient simulation stops after the time specified by the TSTOP parameter. If the time given by TSTOP is greater than that given by RAMPTIME, the source intensities remain constant until the time given by TSTOP. For transient ramps, the TSTEP parameter should also be set. TSTEP is typically set to allow several samples within the RAMPTIME. After the first time step, subsequent time step sizes will be chosen automatically based on estimates of truncation error. The following is an example of the specification of an optical impulse transient:

```
SOLVE B1=1.0 RAMPTIME=1E-9 TSTOP=1E-9 TSTEP=1E-11
```

```
SOLVE B1=0.0 RAMPTIME=1E-9 TSTOP=20E-9 TSTEP=1E-11
```

In this example, a triangular impulse in the intensity of the optical source is simulated. The peak intensity is 1.0 and the impulse width is 2 ns. The response of the device is simulated for an additional 18 ns. An initial time step of 10 ps is chosen for both parts of the impulse.

10.4.7: Obtaining Frequency Response to Optical Sources

You can also simulate small signal response to optical sources. To obtain a solution for small signal response, set the SS.PHOT parameter in the SOLVE statement. The BEAM parameter must also be assigned to the specific optical source index for small signal response of that source. A single source small signal frequency can be specified using the FREQUENCY parameter. You can vary the frequency within a single SOLVE statement by using the NFSTEP and FSTEP parameters. The NFSTEP indicates how many frequency steps are to be simulated, while the FSTEP indicates the step size. If the MULT.F parameter is specified, the start frequency is multiplied by the step size for the specified number of steps. If not, the step size will be added to the start frequency for the specified number of steps. For example:

```
SOLVE SS.PHOT BEAM=1 FREQUENCY=1e6 NFSTEP=5 FSTEP=10 MULT.F
```

will invoke solutions as optical frequencies at every decade from 1MHz to 100 GHz.

If the small signal parameters are specified in the same SOLVE statement as a DC bias ramp, the small signal response is extracted for each bias voltage for each small signal frequency. This is a useful strategy for analyzing the frequency response of the device as a function of bias voltage.

10.4.8: Obtaining Spatial Response

To obtain spatial response, move an optical spot along a line segment perpendicular to the direction of propagation of the source. Each incremental step is equal to the width of the spot. The total distance over which the source is scanned is defined by the MIN.WINDOW and MAX.WINDOW parameters of the BEAM statement. The number of steps is defined by the RAYS parameter of the BEAM statement. The spot width is defined by the ratio:

```
(MAX.WINDOW - MIN.WINDOW)/RAYS
```

The spot scan is started by the SCAN.SPOT parameter of the SOLVE statement. This parameter is set to the beam index of the optical source to be scanned. In other words, the beam defined by the BEAM statement whose NUMBER parameter is set to the beam index. During the spot scan, ATLAS obtains solutions and outputs, such as terminal currents, as well as the relative beam location at each incremental spot location. This information can be used by TONYPLOT to produce plots of photoresponse as a function of position.

10.4.9: Obtaining Spectral Response

The spectral response, defined as device current as a function of the wavelength of the optical source wavelength, can be obtained. The LAMBDA parameter of the SOLVE statement sets the source wavelength of the beam in microns. Since the wavelength can be set in each SOLVE statement, successive solutions can be obtained as a function of wavelength. Each time the LAMBDA parameter is specified, a new ray trace is run for that new wavelength and the electrical solution recalculated.

The following statements could be used to extract terminal currents at a series of discrete wavelengths:

```
SOLVE B1=1 LAMBDA=0.2
SOLVE B1=1 LAMBDA=0.3
SOLVE B1=1 LAMBDA=0.4
SOLVE B1=1 LAMBDA=0.5
SOLVE B1=1 LAMBDA=0.6
SOLVE B1=1 LAMBDA=0.7
SOLVE B1=1 LAMBDA=0.8
```

In this example, the spectral response is obtained for wavelengths from 0.2 microns to 0.8 microns. When using LAMBDA, the WAVELENGTH parameter of the BEAM statement will overridden. Make sure, however, to use a monochromatic beam and not a multi-spectral beam for this simulation.

Note: The units of LAMBDA and WAVELENGTH in the BEAM statement are in μm.

10.5: Simulating Solar Cells

10.5.1: Obtaining Open Circuit Voltage and Short Circuit Current

To obtain V_{OC} and I_{SS} for a solar cell, first define the illumination conditions. This should be done as discussed above in describing multi-spectral sources. The short circuit current is obtained by defining the contacts as voltage dependent contacts (default) and obtaining a solution with the device under zero bias with illumination. This can typically be done as a first step by the following input statement, for example:

```
SOLVE B1=1
```

In this statement, it's assumed the source, `B1`, has been already defined and the intensity is expected in the actual device. When you obtain a solution, the terminal currents represent the short circuit current.

The open circuit voltage is obtained by defining one or more of the contacts as current controlled. This is done using the `CONTACT` statement. For example:

```
CONTACT NAME=anode CURRENT
```

defines that electrode number 1 is a current controlled electrode. The open circuit voltage is then obtained by setting the current at this contact to zero and obtaining a solution. The following statement illustrates:

```
SOLVE I1=0.0 B1=1
```

Once the solution is obtained, the bias associated with `I1` is the open circuit voltage.

10.6: Beam Propagation Method in 2D

You can simulate most optoelectronic devices using ray tracing based on geometrical optic principles. But when diffraction or coherent effects are important, ray tracing methods are no longer sufficient. In this case, you need a method that takes into account the wave nature of light.

The method that does this is called the Beam Propagation Method (BPM). BPMs such as [166] rely on paraxial appoximation. The BPM in LUMINOUS, however, has been extended to solve a more general Helmholtz Wave Equation (Equation 10-51).

$$\nabla^2 E(\hat{r}, t) - \frac{n^2}{c^2} \frac{\partial^2 E(\hat{r}, t)}{\partial t^2} = 0 \qquad 10\text{-}51$$

Here, E is the electric field of an optical wave, n is the complex refractive index of the material, and c is the speed of light in vacuum.

The Helmholtz Wave Equation is consistent with the Rayleigh-Sommerfeld formulation of scalar diffraction theory [67]. It can also be derived from Maxwell's equations assuming homogeneous and isotropic media and neglecting non-linear effects.

These assumptions put certain restrictions on the applicability of BPM. For example, material regions with gradual change in refractive index or photodetector saturation modeling aren't supported. But the framework of the BPM library allows the non-linear effects (e.g., saturation) and treatment of inhomogeneous media (e.g., graded index regions) to be included in the future.

10.6.1: Using BPM

The BPM in LUMINOUS isn't designed to completely replace the fast and robust ray tracing algorithm. In fact, we recommend you start the analysis of your particular application using the ray tracing method (see Sections 10.2.1: "Ray Tracing in 2D" to 10.2.3: "Reflection and Transmission").

Only use BPM when you think light diffraction or coherent effects are important. Here are the following guidelines when to use it:

- The source is monochromatic (no coherent effects are observed for multispectral sources).
- The application is 2D (currently, a 3D BPM isn't implemented due to calculation time considerations).
- Spatial distribution of irradiance or photogenerated carrier density is important.
- The structure isn't periodic in x.

BPM should be useful for:

- Multilayer structures with layer thicknesses comparable to wavelength.
- Narrow incident beams (when transverse beam size is comparable to wavelength).

Currently, BPM doesn't support either user-specified reflection coefficient models (`F.REFLECT`) or photogeneration rate vs. position or time. The `SCAN.SPOT` parameter of `SOLVE` statement (see Chapter 19: "Statements", Section 19.44: "SOLVE") and anti-reflection coatings (see "Anti-Reflective Coatings" on page 10-7) are also not included.

If you need to use BPM, include the `BPM` parameter in the `BEAM` statement (see Chapter 19: "Statements", Section 19.2: "BEAM"). `BEAM` parameters related specifically to ray tracing (such as `RAYTRACE`, `RAYS`, `THINEST`, `MAX.WINDOW`, and `MIN.WINDOW`) are ignored when BPM is specified.

Unlike ray tracing, the incident beam is specified on the top of the device. The origin of the beam is then assumed to be at the `X.ORIGIN` in the `BEAM` statement, where `Y.ORIGIN` parameter is ignored. This is done to avoid unnecessary beam propagation in the material surrounding the device.

You can specify additional parameters (LONGIT.STEP and TRANSV.STEP) in the BEAM statement to set respective step sizes for the internal BPM grid. If you don't define these step sizes, the default values are used. The default step size in both directions is equal to λ/16.0, where λ is the wavelength of light in vacuum. Don't use step sizes larger than λ/2.0, especially if coherent effects are of interest. Also, we don't recommend extremely small step sizes because it can result in unnecessarily long computation times. The default values are adequate for most cases.

10.6.2: Light Propagation In A Multiple Region Device

Since Equation 10-51 doesn't describe light propagation through the region's boundary, each region has to be considered separately. Each region's boundary is composed of straight line segments (edges) that form a polygon. Light propagating from one of the segments could be reflected back into the region from other segments of the boundary, while transmitted light enters other regions. Reflected and transmitted optical fields are calculated using Fresnel formulae described in Equations 10-1 to 10-6 in Section 10.2.3: "Reflection and Transmission".

Once the field reflected from a certain boundary is known, it then propagates back into the region. The light then reaches all edges of the polygon. At this point, the field incident upon each boundary is saved (i.e., added to the total field incident on each boundary). Once all polygon edges are addressed like this, the algorithm then proceeds to the next region.

When the pass through the device is complete, one reflection of light on each boundary has been accounted. The REFLECTS parameter specifies the number of passes to be done.

Like in ray tracing, if the REFLECTS parameter is large (e.g., 10), the optical part of the simulation might take considerable amount of time. To limit the number of propagation routines, use the MIN.POWER parameter. But unlike in ray tracing, where the default value of this parameter is zero, the default is MIN.POWER=1e-6. This limit helps avoid unnecessary computations.

Since only propagation through convex polygons can be properly addressed, concave polygons are automatically split to form two convex regions. If necessary, air polygons are added at the top of the device structure to fill the empty regions, where the propagation of light also needs to be considered.

10.6.3: Fast Fourier Transform (FFT) Based Beam Propagation Algorithm

Field propagation from a boundary segment through a region is considered in the coordinate system of that segment. The X axis is along the boundary and Z axis is along the normal (Figure 10-9).

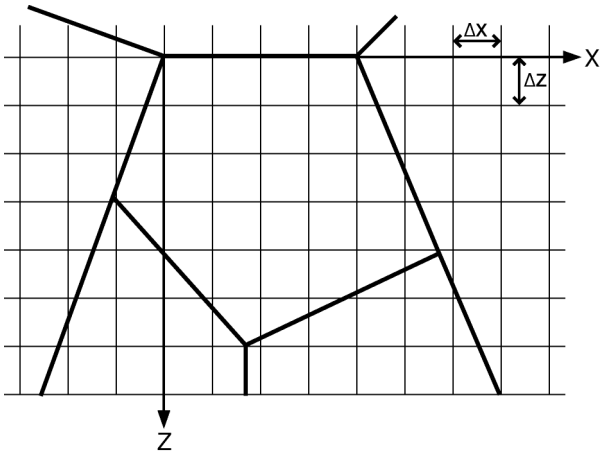


Figure 10-9: Internal Coordinate System

We assume the optical field and the device structure are uniform in the Y direction (perpendicular to the plane of the Figure 10-9). The rectangular grid covering the whole region is chosen according to the step sizes defined by LONGIT.STEP (ΔZ) and TRANSV.STEP (ΔX) parameters. The Helmholtz Equation in a 2D cartesian coordinate system (see Equation 10-52) describes field propagation through the region.

$$\frac{\partial^2 E(x,z)}{\partial z^2} + \frac{\partial^2 E(x,z)}{\partial x^2} + K^2 E(x,z) = 0 \qquad 10\text{-}52$$

Provided that the field on the input plane Z=0 is known, Equation 10-52 allows you to find the field on any subsequent plane Z=ΔZ. The numerical solution of Equation 10-52 is based on the transformation of input complex field into superposition of plane waves. This superposition represents the direction cosine spectrum of plane waves, which is obtained by the FFT of the original field (see Equation 10-53).

$$F(K_x, z=0) = \int_{-\infty}^{\infty} E(x, z=0)\exp(-iK_x X)dx \qquad 10\text{-}53$$

Each plane wave propagates at a certain angle to the Z axis. The phase accumulated by each plane wave component before reaching the plane Z=ΔZ is given by:

$$F(K_x, z=\Delta Z) = F(K_x, z=0)\exp\left(i\sqrt{K^2-K_x^2}\,\Delta Z\right) \qquad 10\text{-}54$$

where $K=2\pi n/\lambda$, n is the complex refractive index. We then can include diffraction and absorption upon propagation simultaneously. The direction cosine spectrum given by Equation 10-54 at Z=ΔZ is transformed back into the spatial domain by applying an inverse FFT.

$$E(x, z=\Delta Z) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(K_x, Z=\Delta Z)\exp(iK_x X)dK_x \qquad 10\text{-}55$$

The field distribution at Z=ΔZ is then found.

These steps are repeated until the field in the whole region is found. The field incident on each boundary segment and the field at each node within the region are obtained using bi-linear interpolation.

When the propagation through the same region is considered, the resulting field values at each node are added to the values saved upon the previous propagation. Since the phase information is retained in our propagation scheme, adding the field ensures the coherent effects are taken into account.

In ray tracing, the phase information is lost and irradiance values are added upon consecutive runs through the same region. This approach is only valid for incoherent sources.

The numerical scheme to solve Equation 10-52 requires the rectangular grid in the transverse direction (X) to cover an area significantly larger than the region of interest.

Zero-padding of the input field distribution is done automatically to specify the input field on the grid. Super-gaussian filters are used in the spatial domain to suppress possible reflections from numerical boundary.

Evanescent waves are included in the propagation scheme to avoid discontinuities in the spectral domain [67].

Chapter 11:
LED: Light Emitting Diode Simulator

11.1: Overview

LED is a simulator within the ATLAS framework that provides general capabilities for simulation of light emitting diode (LED) devices. When used with the BLAZE simulator, LED supports the simulation of DC and transient electrical and light emitting characteristics of LEDs. When used with the GIGA simulator, LED permits the self-consistent analysis of thermal effects on both electrical and optical emission characteristics. You can also use the LED simulator with the MIXEDMODE simulator to analyze the circuit level performance of LED devices.

The LED simulator supports simulation of both zincblende (e.g., AlGaAs/GaAs, InGaAsP/InP) as well as wurtzite (e.g., GaN/AlGaN/InGaN) material systems. It can also account polarization effects and the effects of strain on both emission spectra and piezoelectric polarization.

Extraction of electrical characteristics, such as DC, transient and small signal response, is augmented by the capability to extract light emitting characteristics, such as emission power versus current, emission spectra, emission wavelength, efficiency and output coupling (a function of emission angle).

The following sections describe how to specify LED devices, how to select and specify physical models including most importantly the optical models and how to extract the most pertinent device characteristics for device performance optimization.

11.2: Defining Light Emitting Devices (LEDs)

There are several ways to define an LED device structures in ATLAS/LED. First, you can define a structure using the ATHENA process simulator. ATHENA simulates the "construction" of devices by simulating the process flow and the physical process steps. Second, you can define the device structure using the DEVEDIT device builder tool. The DEVEDIT tool allows you to specify the device structure in a simple visual point and click environment. Finally, you can define device structures using ATLAS framework commands (statements). For LED devices, the ATLAS command language approach is probably the most convenient since the device structures are somewhat simple due to their epitaxial nature.

In the following sections, we will concentrate on definition of the device structure using the ATLAS command syntax. We recommend that you read Chapter 2: "Getting Started with ATLAS" where we describe general features of the ATLAS simulator, commands and structure definition before you proceed. Chapter 3: "Physics" also provides insight into the details of the electrical and optical physical models used by ATLAS/LED. Chapter 5: "Blaze: Compound Material 2D Simulator" describes some general physics of simulation of compound semiconductors and heterojunctions.

Within the ATLAS command syntax, there are two approaches for defining device structures. The general purpose structure definition approach (see Chapter 2: "Getting Started with ATLAS", Section 2.6.3: "Using The Command Language To Define A Structure") requires more user-specification and requires that you describe the entire mesh. The auto-mesh approach (see Chapter 2: "Getting Started with ATLAS", Section 2.6.4: "Automatic Meshing (Auto-meshing) Using The Command Language") is simpler to use, requires less user-specification and is particularly well-suited to specification of epitaxial (especially multiple-quantum well or super-lattice) devices. It is for these reasons that we recommend the auto-mesh approach when defining LED devices .

In the following sections, we will focus mainly on this approach. We will point out important differences with the other methods: ATHENA, DEVEDIT or manual-mesh as these differences may arise.

The first command in any ATLAS input deck is the MESH command. When loading a structure from ATHENA, DEVEDIT or a previous ATLAS simulation, you should assign the INFILE parameter of the MESH statement to the name of the file containing the device structure. For example:

```
MESH INFILE=led.in
```

would cause the simulator to load a device structure from the file named led.in in the current working directory. When defining the device structure in the ATLAS auto-meshing syntax, the first statement should contain the AUTO parameter as shown in the following statement:

```
MESH AUTO
```

This will invoke the auto meshing capability.

Note: The INFILE parameter is not specified since this would indicate that the device was both being loaded as well as constructed in the ATLAS synta.x Therefore, would produce an ambiguity or conflict.

When using auto-meshing, you must only specify the mesh in the X direction using X.MESH statements. The mesh spacing in the Y direction is specified in the REGION statements as described later. The following example shows how you would specify an X mesh 1 μm long with a uniform mesh spacing of 0.1 μm.

```
X.MESH LOCATION=0    SPACING=100

X.MESH LOCATION=1000 SPACING=100
```

Using auto-meshing, you should next specify the device regions using REGION statements. The REGION statement can be used to specify the material types of individual "regions", as characterized by a spatial extent in the X and Y directions, and their composition, doping, strain and certain other characteristics describing how the region is to be modeled.

In the following example, we will describe a simple GaN/InGaN diode structure suitable for LED device simulation.

```
REGION MATERIAL=GaN   THICKNESS=0.25  ACCEPTOR=1e19 BOTTOM NY=20

REGION MATERIAL=InGaN THICKNESS=0.003 DONOR=2e14  X.COMP=0.2 BOTTOM NY=20 LED

REGION MATERIAL=GaN   THICKNESS=0.25  DONOR=1e18 BOTTOM NY=20
```

In this example, we describe a three layer LED. The top layer is composed of GaN (MATERIAL=GaN), it has a thickness of 0.25 microns (THICKNESS=0.25) and has an acceptor concentration of 1e19 (ACCEPTOR=1e19). The middle layer is composed of InGaN (MATERIAL=InGaN), it has a thickness of 0.003 microns (THICKNESS=0.003) cm^{-3}, has an In composition fraction of 0.2 (X.COMP=0.2), and has a donor concentration of 2e14 (DONOR=2e14) cm^{-3}. The bottom layer is also composed of GaN (MATERIAL=GaN), it has a thickness of 0.25 microns (THICKNESS=0.25), and has a donor concentration of 1e18 (DONOR=1e18) cm^{-3}.

The BOTTOM parameter in each of the REGION statements specifies that each region resides directly below the preceeding region. NY specifies the minimum number of Y grid lines used to resolve the region. The LED parameter in the second REGION statement specifies that region will be treated as a light emitting region during the post-processing stages of the simulation, and certain light emitting characteristics of that region will be extracted.

The final step in defining the LED using auto-meshing is to specify the electrodes. To specify electrodes, use the ELECTRODE statement. The following example specifies two electrodes extending all the way across the top and the bottom of the device.

```
ELECTRODE NUMBER=1 NAME=anode TOP

ELECTRODE NUMBER=2 NAME=cathode BOTTOM
```

Here, the NUMBER parameter is used to give the electrodes numerical tags they can be referred to in subsequent operations. Similarly, the NAME parameter is used for future references (see Chapter 19: “Statements”, Section 19.44: “SOLVE”).

11.3: Specifying Light Emitting Diode Models

11.3.1: Specifying Polarization and Piezoelectric Effects

In some material systems (such as the GaN/InGaN system), the built-in fields due to polarization and strain (piezoelectric effect) can play significant roles in the LED emission characteristics. ATLAS introduces these polarization fields as sheet charges at the top and bottom edges of regions. To include the effects of polarization on a region, you should specify the POLARIZATION parameter on the corresponding REGION statement.

This will only include spontaneous polarization. If you want to also include piezoelectric polarization, you can also specify the CALC.STRAIN parameter. If you specify CALC.STRAIN, the simulator will automatically calculate strain from the lattice mismatch and will calculate the piezoelectric polarization and apply it to the region. You can also specify the value of the STRAIN parameter, which specifies the axial strain in the region.

With STRAIN and POLARIZATION set, the simulator will apply a piezoelectric polarization calculated using the strain value assigned by the STRAIN parameter.

11.3.2: Choosing Radiative Models

The next step in simulating LED devices is the selection of appropriate radiative models. For LED devices, the most relevant radiative model is the model for spontaneous (radiative) recombination. For radiative recombination, ATLAS/LED offers several options. Chapter 3: "Physics", Section 3.9: "Optoelectronic Models" discusses these options in more detail.

First, is the general radiative model (see Chapter 3: "Physics", Equation). To enable this equation, specify OPTR in a MODELS statement associated with the active region(s) of the LED device (i.e., those regions with the LED parameter specified in the REGION statement). For our example, the LED discussed is shown in the following statement:

```
MODELS MATERIAL=InGaN OPTR
```

will enable the standard radiative recombination model. The disadvantage of using the standard model is that it provides no information about the spectral content of the LED light emission. In fact to properly specify emission intensity, you must specify the emission wavelength in the SOLVE statement for extraction of LED luminous intensity. To specify the emission wavelength for the general radiative recombination model given by Equation 3-291 in Chapter 3: "Physics", assign the value of the wavelength to the L.WAVE parameter as shown in the following SOLVE statement, where L.WAVE is in units of microns.

```
SOLVE L.WAVE=0.51
```

The COPT parameter of the MATERIAL statement assignes the rate constant for the general radiative recombination model. For example:

```
MATERIAL COPT=1.25e-10
```

COPT assigned a value of 1.25×10^{-10} cm^3/s to the radiative rate constant. You can calibrate the value of this constant using the more physical radiative models discussed next. This calibration will be discussed at the end of this section.

Beyond the basic model just described, ATLAS/LED provides three more physical models for radiative rates. For zincblende materials, ATLAS provides a one band model [187] and a two band [96] model. For wurtzite materials, ATLAS/LED provides a three band model [38].

The one band zincblende model by *Yan et.al.* [187] accounts for optical transitions between a single valence band and the conduction band. To activate this model, specify YAN in the MODELS statement. See Chapter 3: "Physics", Section 3.9.7: "Yan's and Li's Models for Gain and Radiative Recombination in Zincblende Materials" for a discussion of the Yan model.

The two band zincblende model by *Li et.al*. [96] accounts for optical transitions between light and heavy hole valence bands and the conduction band. To activate this model, specify LI in the MODELS statement. See Chapter 3: “Physics”, Section 3.9.7: “Yan's and Li's Models for Gain and Radiative Recombination in Zincblende Materials” for a discussion of the Li model.

The three band wurtzite model by Chuang and Chang [38] is based on k.p modeling and accounts for optical transitions between the conduction band and heavy and light hole and the crystal field split-off valence bands. See Chapter 3: “Physics”, Section 3.9.8: “Chuang's Three Band Model for Gain and Radiative Recombination in Wurtzite Materials” for a discussion of the Chuang model. To activate this model, specify CHUANG in the MODELS statement. For the on-going example, we chose the following CHUANG model (since InGaN is a wurtzite material).

```
MODEL MATERIAL=InGaN CHUANG
```

You should note that any of the YAN, LI or CHUANG model can be used to analyze spectral characteristics, such as emission wavelength and emission spectra.

You can also choose to analyze LED efficiency degradation due to competing recombination mechanisms by enabling these other non-radiative mechanisms. Particularly, you can enable Schockley-Read-Hall or Auger mechanisms. To activate these models, specify SRH and AUGER in the MODELS statement as follows:

```
MODELS SRH AUGER
```

Make sure you do not enable two radiative mechanisms for the same region, such as OPTR and one or more of the YAN, LI or CHUANG models. If you do, you may over account for the same radiative recombination. For more information about recombination models, see Chapter 3: “Physics” Section 3.6.3: “Carrier Generation-Recombination Models”.

As mentioned before, you can calibrate the radiative rate constant (using COPT of the MATERIAL statement) of the basic radiative recombination model (enabled by the OPTR parameter of the MODELS statement) to the more physical models (YAN, LI or CHUANG). To do this, first prepare a simulation with the activated physical model. In this simulation, save a structure file around the device operating conditions. You can then extract the proper value of COPT to use with the OPTR model by dividing the radiative recombination rate at any point by the product of the electron and hole concentrations.

11.3.3: Using k.p Band Parameter Models in Drift Diffusion

For the Chuang model, the band calculations performed in Equations 3-477, 3-478, 3-479 and 3-480 can be used in the drift diffusion part of the simulation. To use this capability, specify the K.P parameter on the MODELS statement as follows:

```
MODELS CHUANG K.P
```

11.4: Data Extraction

11.4.1: Extracting Luminous Intensity

When the LED parameter is specified on any REGION statement, the luminous intensity emitted by the LED device is automatically calculated and written to the log file. The file that this information is output to is specified by the OUTFILE parameter of the LOG statement. The following example outputs the log file to the file led.log.

```
LOG OUTFILE=led.log
```

You can then load led.log into TONYPLOT for post-processing visualization, where you can plot characteristics, such as luminous intensity versus current (see Figure 11-1).

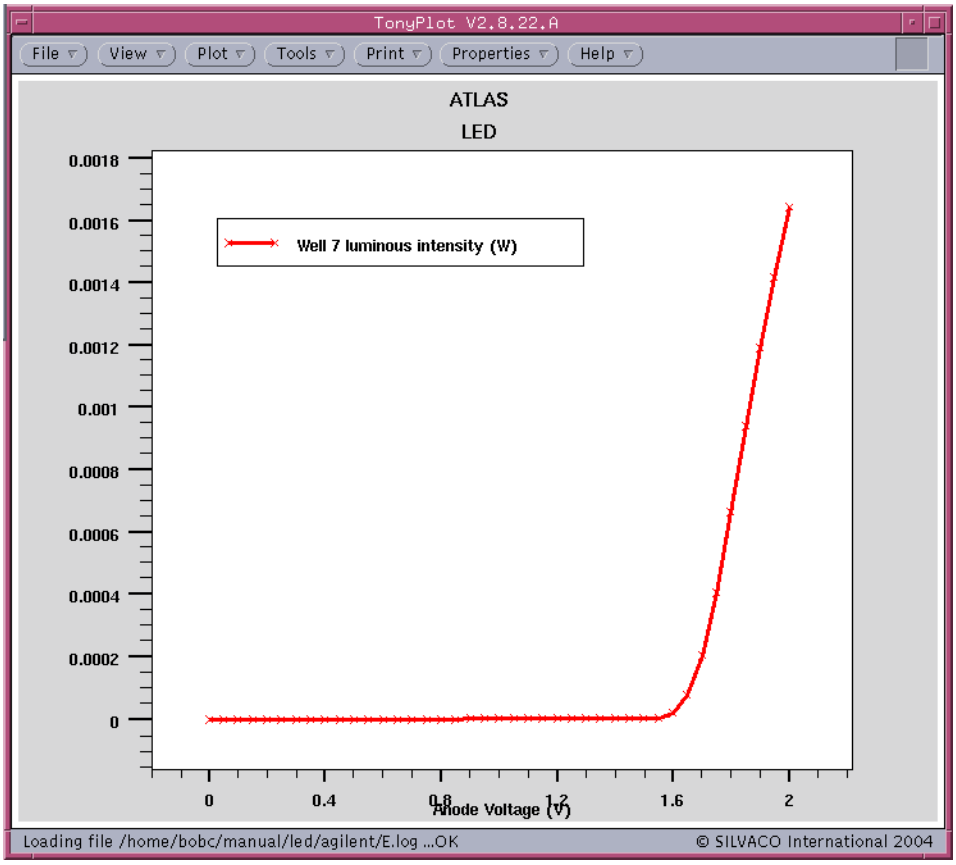


Figure 11-1: Example Luminous Intensity Plot

Note: Luminous intensity is calculated by integrating the luminous spectrum obtained by models containing spectral content (e.g., MODELS CHUANG SPONT). This does not include other non-spectral models (e.g., MODELS OPTR).

11.4.2: Extracting Emission Spectra

The LED simulator can save multiple spectrum files from the SOLVE statement. To save a spectrum file, specify the SPECTRUM parameter after each solution. SPECTRUM is assigned to the file name prefix. Each subsequent file will be written to a file named by SPECTRUM and appended with a version number starting at 0.

The LED simulator can output spectral intensity as a function of energy and wavelength integrated over all wells specified as LEDs. To output spectral intensity, specify the SPECTRUM parameter of the SAVE statement. SPECTRUM is assigned to the name of a file that the LED spectrum is written.

You must also specify either LMIN and LMAX or EMIN and EMAX. LMIN and LMAX specify the minimum and maximum values of wavelength in microns to be captured. EMIN and EMAX specify the minimum and maximum values of energy in eV to be captured. Specify either energy range or wavelength range but not both. To capture the number of descrete samples, specify the NSAMP parameter of the SAVE statement.

Figure 11-2 shows an example spectrum file plotted in TONYPLOT.

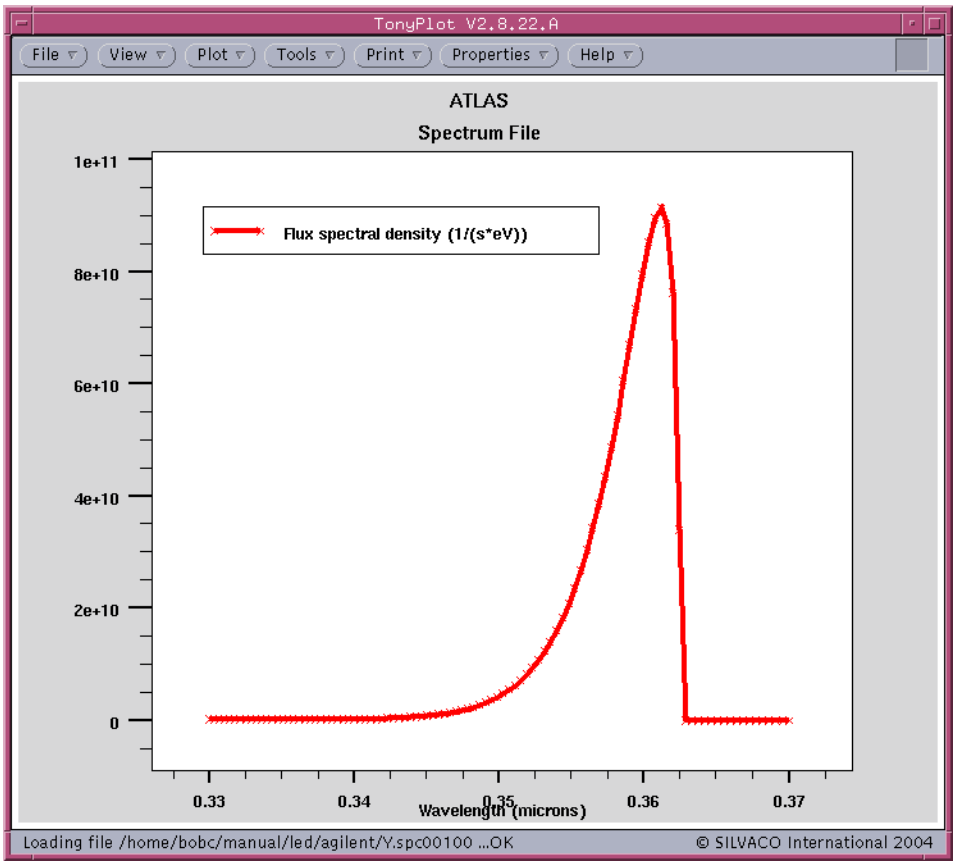


Figure 11-2: Plot of Spectrum File for LED

Note: Emission spectra can only be obtained using a spectral model (e.g., MODELS CHUANG SPONT).

11.4.3: Extracting Emission Wavelength

In the LED simulator, you can use the probe to measure wavelength for LED devices. This will extract the wavelength from the peak of the spectral response over a specified range. To use this capability, specify WAVELENGTH in the PROBE statement. Use either EMIN and EMAX or LMIN and LMAX in the PROBE statement to specify the search range. EMIN and EMAX specify the range in terms of energy in eV, while LMIN and LMAX specify the range in terms of wavelength in microns.

11.5: Reverse Ray-Tracing

Reverse ray-tracing is a technique that allows you to obtain optical output characteristics of an active optoelectronic device based on material properties and device geometry. Ray-tracing is commonly used for modeling of passive optoelectronic components, such as photo-detectors.

Rays originating at the external light source are traced into the device and are absorbed to form electron/hole pairs, which are subsequently detected (using the ray-tracing method in LUMINOUS is described in Chapter 10: "Luminous: Optoelectronic Simulator", Sections 10.2.1: "Ray Tracing in 2D" and 10.2.2: "Ray Tracing in 3D"). Unlike direct ray-tracing, the rays in the reverse method originate inside the active region and are traced until they exit the device. One or multiple origin points (user specified) for rays are considered, and interference effects for rays originating at the common sourse point can be taken into account. When you enable interference, you can analyze the spectral selectivity of the device structure by performing ray-tracing at multiple wavelengths.

You can assess light extraction from the device by using the SAVE statements (after biasing). Five different sets of parameters for reverse ray-tracing are used in the examples throughout the section to demonstrate the features available for modeling LEDs. The parameter sets are given in the order of increasing complexity of simulation. Computation times also vary. It takes approximately 200 times longer to run the last SAVE line compared to the first one.

Set the ANGPOWER parameter in the SAVE statement to start the reverse ray-tracing algorithm. The name of the output file containing the angular power density vs. output angle dependence is specified as a value for ANGPOWER. The information in the angpower file includes the angular power density for TE- and TM- polarized light and total angular power density and total flux angular density vs. output angle.

Note: In TonyPlot, polar charts show the y-axis directed upward (in the opposite direction to the internal coordinate system used in ATLAS). Therefore, the plots will appear to be flipped around x-axis (top of the structure is at the bottom of the chart).

The RAYPLOT parameter specifies the name of the output file containing the information on each ray exiting the device. This file is only created when single origin for all rays is assumed. The information includes ray output angle, relative ray power (TE-, TM-polarization, and total), and initial internal angle at the origin (only if INTERFERE parameter is not specified). 0° angle corresponds to the rays in the X axis direction. 90° angle corresponds to the rays in the Y axis direction.

To start ray-tracing from one point of origin, specify the origin's coordinates for rays X and Y and the wavelength L.WAVE. It is important to choose the origin in the active region of the device. Rays are not traced if the radiative recombination is zero at the ray origin. All remaining parameters outlined below are optional and their default values are assumed if the parameters are not specified.

REFLECTS specifies a number of reflections to be traced for each ray originating at the point (X,Y) (similar to REFLECTS parameter in the BEAM statement). The default value (REFLECTS=0) provides for a quick estimate of the coupling efficiency. To obtain a more accurate result, use REFLECTS>0, especially if you specify MIR.TOP or MIR.BOTTOM. The choice of this parameter is based on a compromise between calculation time and accuracy. The maximum allowed value is REFLECTS=10. Number of reflections set to 3 or 4 is often a good choice. Example 1 produces a simple ray-tracing analysis of an LED. Figure 11-3 shows the resulting angular distribution of the emitted light intensity.

Example 1

```
SAVE   ANGPOWER=OPTOEX18ANG_1.LOG   RAYPLOT=OPTOEX18RAY_1.LOG   X=2.0   Y=1.01
L.WAVE=0.9 REFLECTS=4
```

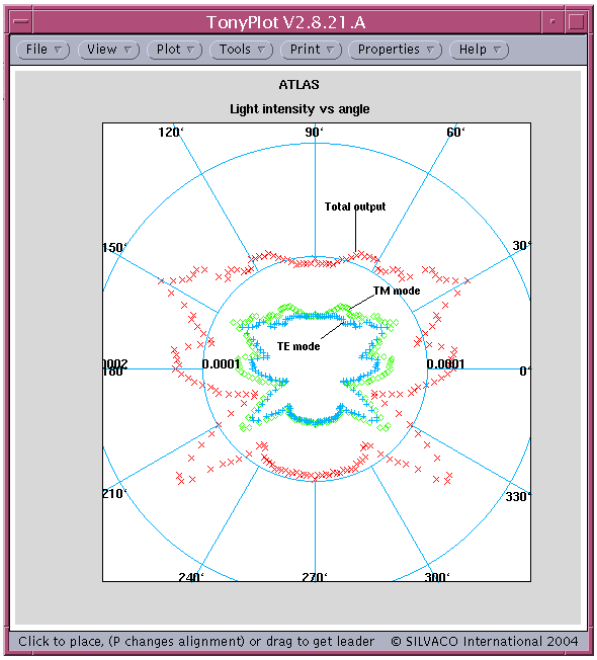


Figure 11-3: Emitted Light Intensity vs Angle for Example 1

Specify the DIPOLE parameter to turn on a particular angular distribution of the internal radiating field that corresponds to a preferred in-plane orientation of dipoles. This orientation is often found in polymer based OLEDs and results in higher optical output coupling. The default setting (DIPOLE=false) corresponds to isotropic distribution of emitting dipoles and spherically symmetric radiation pattern.

MIR.TOP specifies that the top surface of the device be treated as an ideal mirror.

MIR.BOTTOM specifies that the bottom surface of the device be treated as an ideal mirror.

SIDE specifies that the rays reaching the sides of the device terminate there and do not contribute to the total light output. This is often a good assumption for realistic LEDs as these rays tend to be either absorbed internally or blocked by the casing of the device.

TEMPER is the temperature (needed for using appropriate refractive indexes of the materials). The default setting of 300 K will be used if TEMPER is not specified.

POLAR specifies polarization of the emitted photons in degrees (linearly polarized light is assumed). Parallel (TM-mode, POLAR=0.0) and perpendicular (TE-mode, POLAR=90.0) polarizations result in significantly different output coupling values.

You should use the default value (POLAR=45.0) if there is no preferred direction of polarization of emitted photons (unpolarized light emission).

MIN.POWER specifies the minimum relative power of a ray (similar to MIN.POWER parameter in the BEAM statement). The ray is not traced after its power falls below MIN.POWER value. This is useful for limiting the number of rays traced. The default value is MIN.POWER=1e-4.

NUMRAYS specifies the number of rays starting from the origin. The default is 180. The acceptable range 36-3600.

Example 2 shows how some of the parameters described above are used for modeling of a realistic LED. The angular distribution of light power in Figure 11-4 exhibits almost a Lambertian pattern for this simple planar LED geometry. Note that optical coupling coefficient produced in this calculation reflects a 2D nature of the example (the light origin is not a point but rather an infinite line in z-direction).

Simultaneously, you can calculate the optical coupling efficiency for an axially symmetric 3D device. Normally, this is the value to be compared with experimental results. To do this, specify the COUPLING3D parameter (while SIDE is set). For this calculation, the light source is assumed to be a point located on the axis of symmetry.

Example 2

```
SAVE   ANGPOWER=OPTOEX18ANG_2.LOG   RAYPLOT=OPTOEX18RAY_2.LOG   X=2.0   Y=1.01
L.WAVE=0.9 SIDE MIR.BOTTOM NUMRAYS=360 REFLECTS=4
```

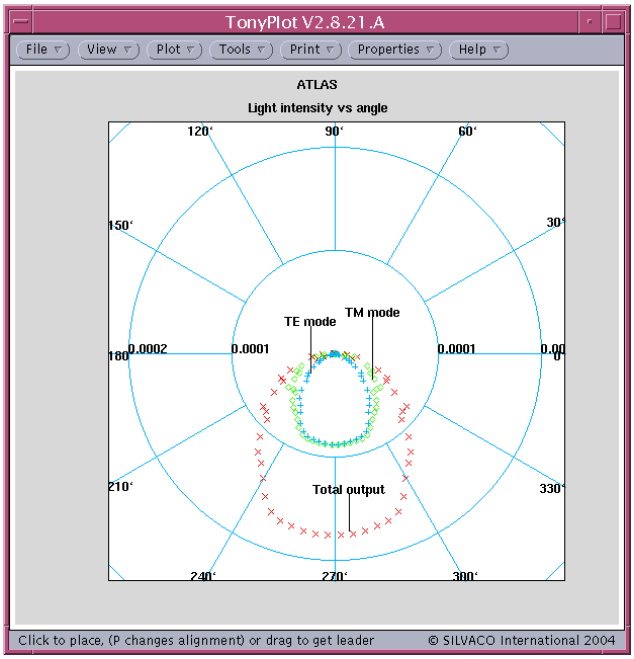


Figure 11-4: Emitted Light Intensity vs Angle for Example 2

The rays are assumed to be incoherent by default. This is a good approximation if the thickness of the active layer of the device is on the order of a wavelength/index. For layer thicknesses, smaller than the wavelength coherent effects might be important. When INTERFERE is set, the rays originating at the common source point are taken to be 100% coherent. In this case, the phase information upon propagation is preserved. Phase change upon reflection is also considered. Thus, interference of rays exiting the device at the same angle is taken into account. In this case, the internal angle information is not written to the output rayfile.

Example 3 takes into account interference. Figure 11-5 shows the result. You can take absorption of rays into account by setting the ABSORB parameter. The absorption is assumed to be specified for each material by the imaginary part of the refractive index. Carrier density dependent absorption and photon-recycling are not considered at this point.

Example 3

```
SAVE   ANGPOWER=OPTOEX18ANG_3.LOG   RAYPLOT=OPTOEX18RAY_3.LOG   X=2.0   Y=1.01
L.WAVE=0.9  INTERFERE SIDE MIR.BOTTOM REFLECTS=4
```

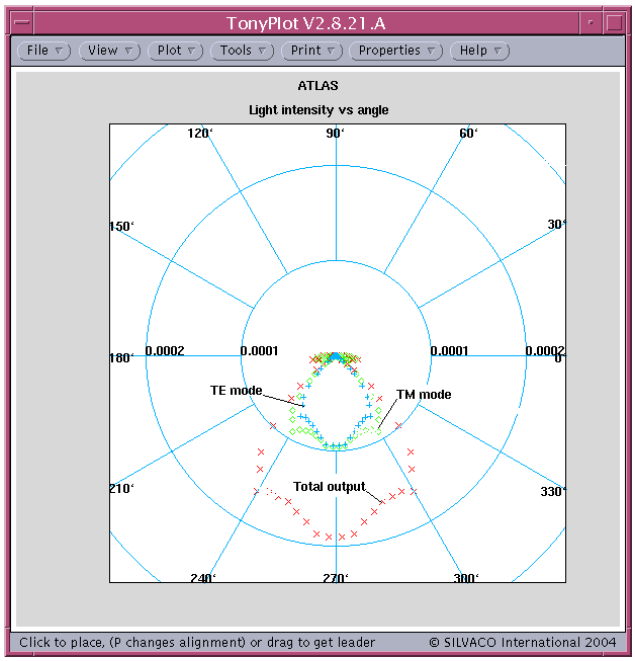


Figure 11-5: Emitted Light Intensity vs Angle for Example 3

Although ray-tracing from one point of origin can give a reasonable estimate of optical output coupling and angular distribution of light power, it is often desirable to consider multiple points within the active layer of the device to obtain more accurate results. To set multiple origin points, use the following parameters.

- XMIN, XMAX, YMIN, and YMAX define a rectangular area containing all origin points.
- XNUM and YNUM specify the number of points along x and y axes within the rectangular area. If XNUM=1, the x-coordinate of all origin points will be set to XMIN so that the points are chosen along the line X=XMIN. Similarly, you can choose points along the specific y-line.

Ray-tracing from multiple origins is realized by repeating single origin algorithm for each point and by adding up the normalized angular power density values thus obtained. The luminous power assigned to each source (origin) is proportional to the radiative recombination at that point. The luminous power of all sources adds up to the value obtained by integration of radiative recombination over the entire device. Rays originating at different source points are completely incoherent (even when INTERFERE is set), which is consistent with the spontaneous character of the radiation produced by an LED. The rayplot file is not written for multiple origins (even if RAYPLOT parameter is set). Example 4 shows how multiple origin simulation can be done. Figure 11-6 shows angular distribution of light obtained in this case.

Example 4

```
SAVE ANGPOWER=OPTOEX18ANG_4.LOG XMIN=1.0 XMAX=3.0 XNUM=3 YMIN=1.01 YMAX=1.09
YNUM=9 INTERFERE SIDE L.WAVE=0.9 MIR.BOTTOM REFLECTS=4
```

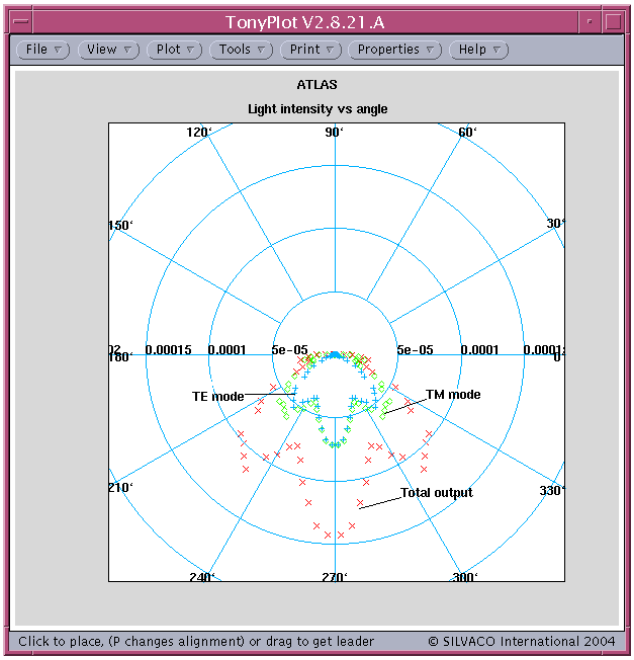


Figure 11-6: Emitted Light Intensity vs Angle for Example 4

You can also take spectral selectivity of optical output coupling for LEDs into account. Reverse ray-tracing at multiple wavelengths is considered if the SPECTRUM parameter specifies a filename for spectral selectivity output while the ANGPOWER parameter is set.

EMIN and EMAX, or LMIN and LMAX specify the energy or wavelength range respectively.

NSAMP specifies the number of spectral components to be considered. We suggest that you specify the SPECTRUM file only when INTERFERE is set. Otherwise, a warning will be issued during ray-tracing. When ANGPOWER and SPECTRUM are set in the SAVE statement, the resulting optical output coupling is averaged over the entire energy (wavelength) range from EMIN to EMAX (from LMIN to LMAX). The same applies to the quantities in the output angular distribution file. The spectrum file only shows how output coupling changes with wavelength. Currently, the shape of the gain curve is not taken into account (flat gain).

Example 5 shows how to use multiple spectral components. Figure 11-7 shows that the results obtained after averaging over spectral components and multiple origin points while taking interference into account are similar to the single point source model, where interference and multiple spectral content are ignored (Example 2). This demonstrates the applicability of a simpler model in this case.

Example 5

```
SAVE ANGPOWER=OPTOEX18ANG_5.LOG SPECTRUM=OPTOEX18SP_5.LOG XMIN=1.0 XMAX=3.0
XNUM=3 YMIN=1.01 YMAX=1.09 YNUM=5 INTERFERE SIDE LMIN=0.86 LMAX=0.94 NSAMP=15
MIR.BOTTOM REFLECTS=4
```

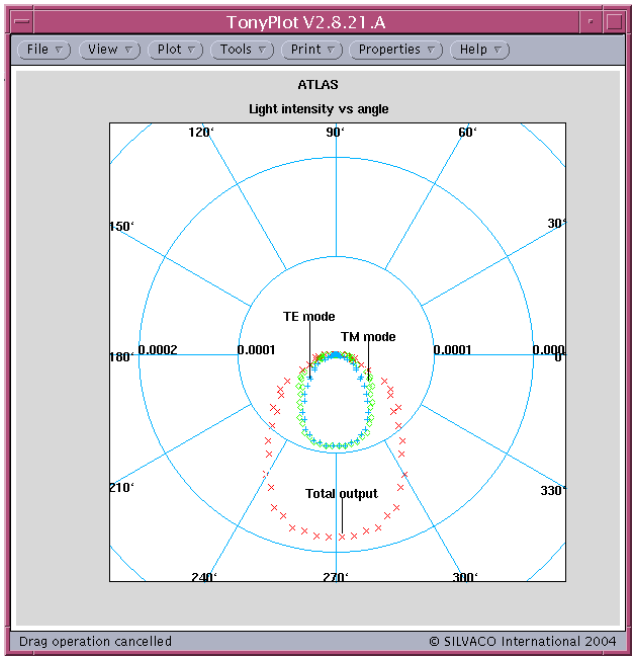


Figure 11-7: Emitted Light Intensity vs Angle for Parameter Set 5

Chapter 12:
MixedMode: Mixed Circuit and Device Simulator

12.1: Overview

MIXEDMODE is a circuit simulator that can include elements simulated using device simulation and compact circuit models. It combines different levels of abstraction to simulate relatively small circuits where compact models for single devices are unavailable or sufficiently accurate. MIXEDMODE also allows you to also do multi-device simulations. MIXEDMODE uses advanced numerical algorithms that are efficient and robust for DC, transient, small signal AC and small signal network analysis.

MIXEDMODE is typically used to simulate circuits that contain semiconductor devices for accurate compact models that don't exist or circuits where devices play a critical role must be modeled accurately. Applications of MIXEDMODE include: power circuits that may include diodes, power transistors, IGBTs, and GTOs, optoelectronic circuits, circuits subject to single event upset, thin film transistor circuits, high-frequency circuits, precision analog circuits, and high performance digital circuits.

MIXEDMODE circuits can include up to 200 nodes, 300 elements, and up to ten numerical simulated ATLAS devices. These limits are reasonable for most applications. But, they can be increased in custom versions on request to Silvaco. The circuit elements that are supported include dependent and independent voltage and current sources as well as resistors, capacitors, inductors, coupled inductors, MOSFETs, BJTs, diodes, and switches. Commonly used SPICE compact models are available. The SPICE input language is used for circuit specification.

This chapter describes circuit simulation capabilities rather than device simulation capabilities. The first part of the chapter contains introductory and background information. Then, describes presents and explains MIXEDMODE syntax. This is followed by some sample input decks. The final sections contain a statement reference and a detailed description of the provided electrical compact models for diodes, BJTs, and MOSFETs.

12.1.1: Background

Circuit simulators such as SPICE [10] solve systems of equations that describe the behavior of electrical circuits. The devices that are of interest to circuit designers are normally well characterized. Compact or circuit models are analytic formulae that approximate measured terminal characteristics. Advanced compact models provide high accuracy with minimum computational complexity. Device modeling, device characterization and parameter extraction are concerned with the development and use of accurate and efficient compact models.

Physically based device simulation solves systems of equations that describe the physics of device operation. This approach provides predictive capabilities and information about the conditions inside a device. It can, however, require significant amounts of CPU time. Information is usually transferred from device simulation to circuit simulation as follows: electrical characteristics are calculated using a physically-based device simulator. These calculated electrical characteristics are then used as input by a device modeling and parameter extraction package such as UTMOST [162]. The extracted parameters are used to characterize a compact model used by the circuit simulator.

This approach is adequate for many purposes but has limitations. It requires that satisfactory compact models already exist. The use of compact models always introduces some error. Models that are adequate for digital circuit simulation may be inadequate for other applications. Applications and devices for which compact modeling is not always satisfactory include: precision low power, high power, high frequency circuit simulation, SOI, IGBT, GTO, TFT, and optoelectronic devices.

12.1.2: Advantages of MixedMode Simulation

The limitations of compact models can be overcome by using physically-based device simulation to predict the behavior of some of the devices contained in a circuit. The rest of the circuit is modeled using conventional circuit simulation techniques. This approach is referred to as mixed-mode simulation, since some circuit elements are described by compact models, and some by physically-based numerical models.

MIXEDMODE simulation provides several worthwhile advantages. No compact model need be specified for a numerical physically-based device. The approximation errors introduced by compact models can be avoided particularly for large signal transient performance. You can also examine the internal device conditions within a numerical physically-based device at any point during the circuit simulation. But the cost is increased CPU time over SPICE as CPU time is comparable to a device simulation excluding the external circuit nodes. MIXEDMODE simulation normally uses numerical simulated devices typically only for critical devices. Non-critical devices are modeled using compact models.

12.2: Using MixedMode

Input file specification for MIXEDMODE is different in many respects to the rest of ATLAS. But if you're familiar with SPICE and ATLAS syntax, you should have little difficulty understanding how these two syntax styles are joined in MIXEDMODE.

Each input file is split in two parts. The first part is SPICE-like and describes the circuit netlist and analysis. The second part is ATLAS-like and describes the device simulation model parameters. These two sections of an input file are separated as described in the next section.

The circuit description includes the circuit topology (called the netlist) and the electrical models and parameters of the circuit components. The simulation conditions specify the types of analysis to be performed. These items are described using syntax based on SPICE.

The ATLAS device descriptions provide information about device geometry, doping distribution, and meshes. Device descriptions can be prepared using the built-in ATLAS syntax, the ATHENA process simulator, or the DEVEDIT structure specification and meshing tool. You can also read-in previously calculated device solutions. The device data is read-in from standard structure format files.

When a simulation has finished, the following information is available:

- I-V data (voltages in all circuit nodes and currents in all circuit branches).
- Internal distributions of solution variables (such as electron, hole, and potential distributions) within the numerical devices.

The results of previous runs of MIXEDMODE can be used as initial guesses for future simulations. This is particularly helpful when multiple simulations must be performed from the same starting point.

The accessing and running of examples for ATLAS are documented in the DECKBUILD USER'S MANUAL. We recommend that you run at least one MIXEDMODE example provided on the distribution tape before trying their own simulations.

12.2.1: General Syntax Rules

The SPICE-like part of any MIXEDMODE input file starts with the `.BEGIN` statement. The SPICE-like part of the input file ends with `.END`. All parameters related to the device simulation models appear after the `.END` statement. To start MIXEDMODE3D, specify the `3D` parameter in the `BEGIN` statement.

The first non-comment statement after initializing ATLAS (`go atlas`) has to be `.BEGIN`. The order of the following netlist and control statements is arbitrary. The last SPICE-like statement has to be `.END`.

Unlike the rest of ATLAS, for SPICE-like statements the `exact` command has to be used, unique abbreviations are not accepted. Statements are not case sensitive.

There has to be at least one numerical ATLAS device ("A" device) within the netlist.

Comment characters are `#` and `$`, but not `*`.

All ATLAS statements specifying the parameters for the numerical device simulation have to be specified after `.END`

After all ATLAS statements, the simulation has to be explicitly terminated (`quit`, `go <simulator>`).

These rules do not apply to the `SET` statement for parameterization of the input file, since it is interpreted by DECKBUILD only.

`EXTRACT` statements are also an exception similar to `SET`. Since MIXEDMODE input files are parsed completely before execution (see Section 12.2.4: "Recommendations" for more information), extractions can only be done after completion of the simulation. To extract results from a MIXEDMODE simulation, `EXTRACT` should be specified after re-initialization of ATLAS (`go atlas`).

12.2.2: Circuit and Analysis Specification

The SPICE-like MIXEDMODE statements can be divided into three categories:

- Element Statements: These statements define the circuit netlist.
- Simulation Control Statements: These statements specify the analysis to be performed.
- Special Statements: These statements typically related to numerics and output (first character being a dot ".").

The specification of the circuit and analysis part has to be bracketed by a .BEGIN and an .END statements. In other words, all MIXEDMODE statements before .BEGIN or after .END will be ignored or regarded as an error. The order is between .BEGIN and .END is arbitrary.

Netlist Statements

Each device in the circuit is described by an element statement. The element statement contains the element name, the circuit nodes to which the element is connected and the values of the element parameters. The first letter of an element name specifies the type of element to be simulated. For example, a resistor name must begin with the letter, R, and can contain one or more characters. This means that R1, RSE, ROUT, and R3AC2ZY are all valid resistor names. Some elements, such as diodes and transistors, must always refer to a model. A set of elements can refer to the same model. For some elements, such as resistors and capacitors, model referencing is optional. Each element type has its own set of parameters. For example, a resistor statement can specify a resistance value after the optional model name. The bipolar transistor statement (Q) can specify an area parameter. All parameters have corresponding default values. Independent voltage and current sources have different specifications for transient, DC, and AC phases of simulation. Transient specifications use the keywords: EXP, PULSE, GAUSS, SFFM, SIN, and TABLE. AC parameters start with the keyword, AC.

Elements to be simulated numerically are defined as "A" devices (ATLAS devices). At least one ATLAS device in a circuit is mandatory.

MIXEDMODE supports the use of the following circuit elements:

- Numerically simulated ATLAS devices ("A" devices)
- User-defined two-terminal elements ("B" devices)
- Capacitors ("C" devices)
- Diodes ("D" devices)
- Voltage controlled voltage source ("E" devices)
- Current controlled current source ("F" devices)
- Voltage controlled current source ("G" devices)
- Current controlled voltage source ("H" devices)
- Independent current sources ("I" devices, may be time dependent)
- JFETs ("J" devices)
- Coupled (mutual) inductors ("K" devices)
- Inductors ("L" devices)
- MOSFETs ("M" devices)
- Optical sources ("O" devices)
- Bipolar junction transistors ("Q" devices)
- Resistors ("R" devices, may be time dependent)
- Lossless transmission lines ("T" devices)
- Independent voltage sources ("V" devices, may be time dependent)
- MESFETs ("Z" devices)

The physical models for linear elements (resistors, capacitors, sources, and so on) are described Section 12.4.1: “Circuit Element Statements”. The models for diodes: BJTs, JFETs, MESFETs and MOSFETs are described in Section 12.4.2: “Control and Analysis Statements”. You can find more extensive documentation, however, in the SMARTSPICE/UTMOST MODELING MANUALS Volumes 1, 2, and 3.

A node is a point in the circuit where two or more elements are connected. A node can be described either in terms of a node name or as a node number. The node names and numbers are arbitrary with the exception of the ground node. The ground node is set either by specifying “0” as the node number or using the name GND. All voltages at the nodes are calculated with respect to the voltage at the ground node.

Example

The netlist for the circuit shown in Figure 12-1 is represented by the following MIXEDMODE input deck fragment.

```
# independent voltage source, 0.1V, connected to node 0 (GND) and 1:
V0    1           0           0.1
# 1kOhm resistor, connected to node 1 and 2
R1    1           2           1K
# ATLAS device, connected to node 2 (anode) and 0 (cathode),
# current scaled by 5e7, mesh from file dio.str
ADIO  2=anode     0=cathode  WIDTH=5e7   INFILE=dio.str
```

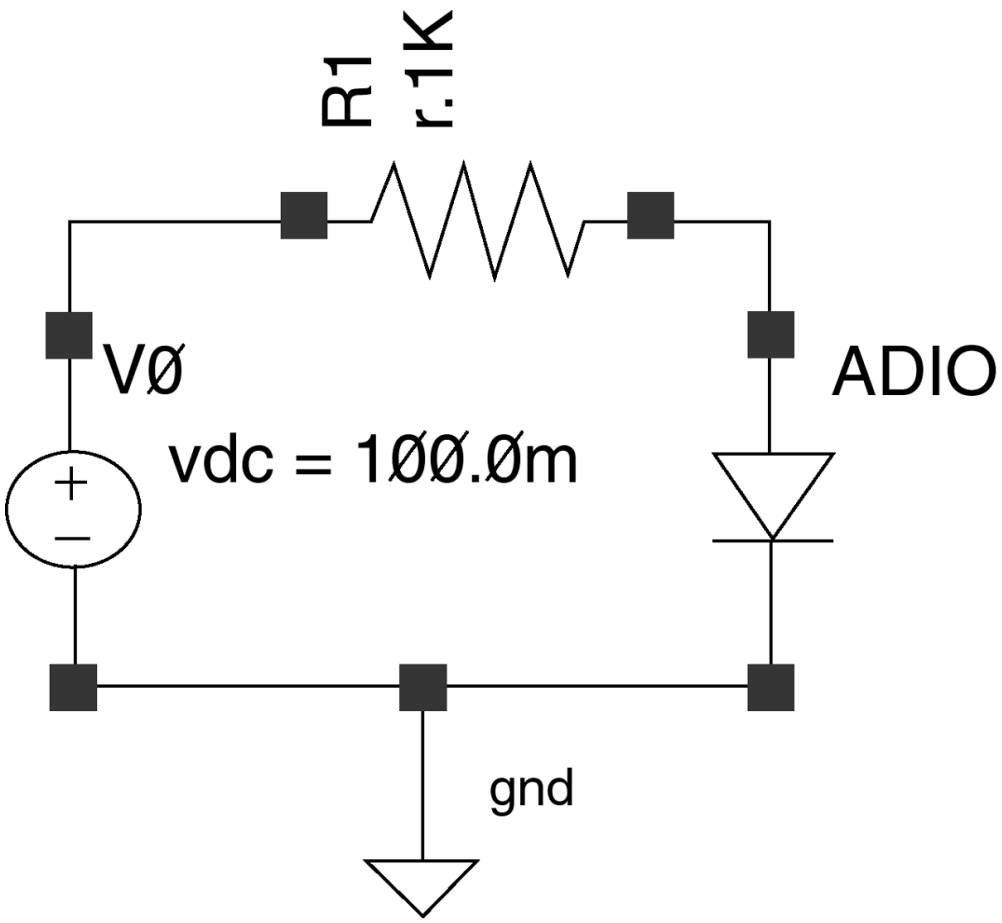


Figure 12-1: Schematic of Primitive Example Circuit

Control Statements

Control statements are used to specify the analysis to be performed in MIXEDMODE. These take the place of the SOLVE statements in a regular ATLAS input file. At least one of these statements must appear in each MIXEDMODE input file. These statements are:

- **.DC:** steady state analysis including loops
- **.TRAN:** transient analysis
- **.AC:** small signal AC analysis
- **.NET:** small signal parameter extraction (e.g., s-parameters)

If you wish to perform a DC analysis, you should have an associated DC source to relate the DC analysis to. Likewise, for an AC analysis you should have an associated AC source that you relate the AC analysis to. For a transient analysis, you should have an associated source with transient properties that you relate the transient analysis to. Failing to correctly run an analysis on the same sort of source could yield simulation problems, such an running an AC analysis on an independent voltage source where no AC magnitude for the source has been specified.

For more information about these statements, see Section 12.4.2: "Control and Analysis Statements".

Special statements

Other statements beginning with a dot " . " specify special parameters for the circuit simulation. These include numerical options, file input and output, and device parameter output. These statements are listed below.

- compact device models (.MODEL)
- the output files (.LOG, .SAVE)
- initial conditions settings (.NODESET, .IC)
- initial conditions from a file (.LOAD)
- numerics (.NUMERIC, .OPTIONS)
- device parameter output (.PRINT)
- miscellaneous (.OPTIONS)

Full descriptions of each statement and associated parameters are found in Section 12.4.2: "Control and Analysis Statements".

12.2.3: Device Simulation Syntax

The second part of a MIXEDMODE command file (after .END) is used to define physical models, material parameters, and numerical methods for ATLAS devices referenced in the "A"-element statements. The following statements may appear in this part of the command file: BEAM, CONTACT, DEFECT, IMPACT, INTERFACE, INTTRAP, MATERIAL, MOBILITY, METHOD, MODELS, OUTPUT, PROBE, TRAP, and THERMCONTACT.

You always need to include an indicator to the circuit element name in each device simulation statement even if there is only one A-device in the circuit. All statements specifying the device properties and models are just supplemented by the DEVICE=name parameter, where name is the circuit element in the netlist, and name will always begin with the letter "A". This makes it possible to define different material properties and model settings for different devices within the circuit.

We recommended that you specify the REGION parameter referring to only one region in IMPACT, MATERIAL, and MODELS statements. If the device consists of more than one region, several statements with the same device parameters and different region parameters are recommended.

For example to specify the bipolar set of models to a device the syntax used might be:

```
MODEL DEVICE=AGTO REGION=2 BIPOLAR PRINT
```

12.2.4: Recommendations

Input Parsing

In regular ATLAS (non-MIXEDMODE) simulations, the input is interpreted line by line and each statement is executed immediately. This is very useful and nicely supported by DECKBUILD for the interactive development of the input. Circuit simulations, however, require the complete input before any simulation can be performed. Consequently, the following occur:

- The complete input is read and parsed before any simulation is initiated.
- An explicit termination of a simulation is required (quit).
- All post processing (extraction and plotting) has to be done after re-initializing ATLAS again.

No simulation is started until either a QUIT statement or a GO statement is seen in the input file. Post-processing can be done by restarting ATLAS.

Scale and Suffixes

In the MIXEDMODE part of the input, numerical values of parameters are represented in standard floating-point notation. The scale suffix may be followed by a unit suffix (e.g., A for Ampere, V for Volt, and so on). Using a unit suffix can increase the clarity of a command file. The unit suffix is ignored by the program. The scale suffixes are shown in Table 12-1.

Table 12-1. Scale Suffixes

Factor	Name	Suffix
10^{-15}	femto-	F
10^{-12}	pico-	P
10^{-9}	nano-	N
10^{-6}	micro-	U
10^{-3}	milli-	M
10^{3}	kilo-	K
10^{6}	mega-	MG
10^{9}	giga-	G
10^{12}	tera-	T

Numerics

MIXEDMODE solves circuit and device equations simultaneously using fully coupled algorithms. This provides better convergence and requires less CPU time than alternative approaches. The number of circuit variables is often small in comparison with the number of device variables. In this case, the CPU time required for simulation performed using MIXEDMODE does not increase drastically compared to the sum of the simulation times required for the individual numerical physically based devices. MIXEDMODE uses the Newton algorithm for each bias point during steady-state analysis and for each time step during transient analysis. Different variants of the Newton algorithm are used depending on the circumstances [106, 107]. The full Newton method [.OPTIONS FULLN] and a modified two-level Newton method [.OPTIONS M2LN] are available for steady-state simulation. The full Newton method provides rapid convergence when a good initial guess is available. The modified two-level Newton algorithm is less sensitive to the initial guess. For transient simulation, a good initial guess always exists. The full Newton method therefore works very well. Therefore, it is always used for transient simulation.

When using MIXEDMODE3D, it is recommended that you specify the DIRECT or GMRES solver in the ATLAS part of the MIXEDMODE input deck on the METHOD statement. Also, use the NOPROJ parameter in the .OPTIONS statement in the MIXEDMODE of the input deck.

Multi-Device Structure Representation

If more than one ATLAS device is defined in a MIXEDMODE simulation, the structures are merged together internally. The output solution file is a single file which contains both structures. The first structure referenced will be on top, all other structures will be attached below.

Example

A diode and a bipolar transistor are specified as numerical devices with the following element statements:

```
ABJT 1=BASE 2=EMITTER 4=COLLECTOR WIDTH=1E4 INFILE=bjt.str

ADIO 3=ANODE 4=CATHODE WIDTH=1.5E5 INFILE=dio.str
```

After outputting the solution with:

```
.SAVE MASTER=mas
```

The solution file for the first DC-point, `mas_dc_1`, contains both structures with the second ATLAS device (diode) shifted downwards (see Figure 12-2).

This coordinate shift has to be accounted for eventually when extracting position dependent solution quantities or when defining spatially dependent properties with the C-INTERPRETER (See Appendix A: "C-Interpreter Functions" for more information).

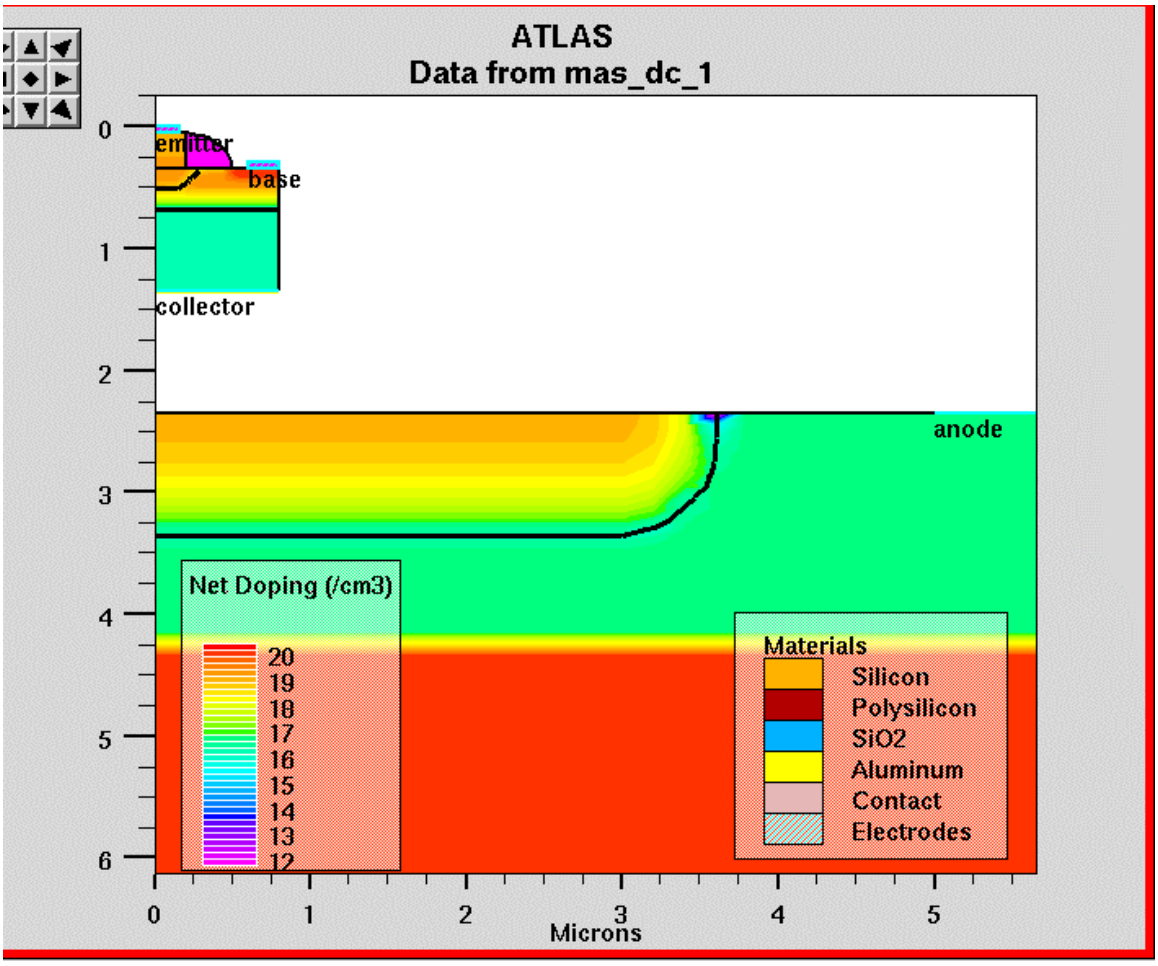


Figure 12-2: Display of a MixedMode solution with two Numerical Devices

Extraction of Results

By default, EXTRACT reads its data from the currently opened log-file when executed along with ATLAS. Since the extraction of MIXEDMODE log-files require a re-initialization of ATLAS (see "Input Parsing" section on page 12-7), EXTRACT has to be initialized explicitly with the correct name of the MIXEDMODE log file. To extract voltages at specific nodes, use the syntax vcct.node."circuit node". To extract circuit elements, use icct.node."circuit element".

Example

Specify the log file:

```
.LOG OUTFILE=hallo
```

Subsequent extraction from the transient log-file is done with:

```
go atlas
extract init inf="hallo_tr.log"
extract name="t0" x.val from curve(time,icct.node."Adio_anode") \
        where y.val=0
```

It extracts the time, t0, when the transient of the current from the anode electrode of the adio device in the circuit crosses zero. For more details for the EXTRACT syntax, see the DECKBUILD USER'S MANUAL.

Using MixedMode inside the VWF Automation Tools

Like all other Silvaco products, MIXEDMODE is fully integrated into the VWF framework and can be used for automated experiments. There are, however, some factors to take into account.

One factor is that the **auto-interface** feature doesn't work with MIXEDMODE. All structures have to be explicitly saved in unique files previous to the MIXEDMODE runs and referred to in the A- element statements. Another factor is that splits within MIXEDMODE runs are impossible. To overcome this problem, use the SET statement to define a variable in a process simulator or in the "dummy" internal run. This variable is used to parameterize the input file.

Example

Capacitance as an independent split variable in a VWF experiment.

```
go internal
# define the independent split variable in a re-entrant simulator:
set cap=5e-9

go atlas

.BEGIN
# use the variable as parameter in MixedMode:
C1    2     3     $cap
```

Another factor is that the automation tools only store files opened by the normal ATLAS LOG statement in the VWF database but ignore those defined by .LOG. To overcome this, re-initialize ATLAS and open the relevant log file of the previous MIXEDMODE run with .log as the append option (so that the file is not reset).

Example

MIXEDMODE log-file definition.

```
.LOG OUTFILE=hallo
```

Re-opening the second DC-log file and the transient log file to get them stored in the VWF database.

```
go atlas

log outfile=hallo_dc_2.log append
log outfile=hallo_tr.log append
```

Initial Settings

Initial convergence is critically dependent on the initial settings of the node voltages (i.e, .IC and .NODESET). There should not be any problem starting from the zero bias case. Just like starting from a preceding MIXEDMODE solution is simple, since the complete solution of the circuit and the ATLAS devices is directly available (i.e, .LOAD and .SAVE). But when loading solutions for the numerical devices from ATLAS, using .OPTIONS LOADSOLUTIONS, sometimes precise matching of the initial circuit condition is required. In this case, it is practical to extract the relevant properties in the preceding ATLAS run and use them to parameterize the MIXEDMODE input.

In the following example, the voltages and current of an ATLAS solution is extracted, and the results are used for the initial definition of the circuit.

The end of the first part is the stand-alone ATLAS simulation:

```
# extract the final voltage drop on the anode:
extract name="Von" max(vint."anode")
# extract the gate current:
extract name="I_gate" y.val from curve(vint."anode",i."gate") \
where x.val = $"Von"
extract name="V_gate" y.val from curve(vint."anode",vint."gate") \
where x.val = $"Von"

# now the MIXEDMODE part

go atlas
.BEGIN
# define the gate current source, use extracted value as parameter
I1 0 7 $"I_gate"
#
# use extracted gate bias and other expressions to calculate
# the node settings:
set Rgl = 10.5
set v7= $V_gate + $I_gate * $Rgl
.NODESET V(1)=2000 V(2)=$"Von" V(3)=$"V_gate" V(4)=$"V_gate" V(5)=-25 \
V(6)=-15 V(7)=$"v7"
```

12.3: A Sample Command File

A sample MIXEDMODE command file is shown below. This file is used to simulate the reverse recovery of a power diode. Several MIXEDMODE examples are provided with the product which can be accessed using DECKBUILD.

```
1. go atlas
2. .BEGIN
3. V1 1 0 1000.
4. R1 1 2 1m
5. L1 2 3 2nH
6. R2 4 0 1MG EXP 1MG 1E-3 0. 20NS 10 200
7. IL 0 4 300
8. ADIODE 3=cathode 4=anode WIDTH=5.E7 INFILE=pd.str
9. .NUMERIC LTE=0.3 TOLTR=1.E-5 VCHANGE=10.
10. .OPTIONS PRINT RELPOT WRITE=10
11. $
12. .LOAD INFILE=pdsave
13. .LOG OUTFILE=pd
14. .SAVE MASTER=pd
15. $
16. .TRAN 0.1NS 2US
17. $
18. .END
19. $
20. MODELS   DEVICE=ADIODE REG=1 CONMOB FLDMOB CONSRH AUGER BGN
21. MATERIAL DEVICE=ADIODE REG=1 TAUN0=5E-6 TAUP=2E-6
22. IMPACT   DEVICE=ADIODE REG=1 SELB
23. $
24. METHOD   CLIM.DD=1.E8 DVMAX=1.E6
25. $
26. go atlas
27. tonyplot pd_tr.log
```

Description

Line 1: All ATLAS input files should begin with `go atlas`

Line 2: The `.BEGIN` and `.END` statements indicate the beginning and end of the circuit simulation syntax. These commands are similar to those used in SPICE.

Lines 3-7: Circuit components, topology, and analysis are defined within. Generally, the circuit component definition consists of three parts: the type of component, the lead or terminal mode assignments, and the component value or model name. For example, if the first component definition in this simulation is a DC voltage source, then `V1` defines the component as voltage source number one, `1` and `0` are the two circuit modes for this component, and `1000` indicates that the voltage source value is 1000 volts. The remaining circuit components are resistors (`R1`, `R2`) inductor (`L1`) and independent current source (`IL`).

The reverse recovery of the diode is simulated by dropping the value of output resistor `R2` over a small increment of time. The `R2` statement contains additional syntax to perform this task. Here, the resistor is treated as a source whose resistance decreases exponentially from 1 mOhm to 1 mOhm over the specified time step. This action essentially shorts out the parallel current source `IL`, which is also connected to the base of the diode.

Line 8: The ADIODE statement specifies a device to be analyzed by ATLAS. The A part of the ADIODE command specifies that this is a device statement. The DIODE portion simply defines the device name. The option INFILE= indicates which device structure file is to be used.

Lines 9-10: These set numerical options for the circuit simulation. WRITE=10 specifies that every tenth timestep will be saved into the solution file specified on the .SAVE statement.

Line 12: Specifies a file generated by a previous MIXEDMODE simulation to be used as an initial guess to the voltage.

Line 13-14: Specifies the output log and solution filenames. These names are root names and extensions will be added automatically by the program.

Line 16: Indicates the type of analysis required. In this case, it is a transient simulation lasting 2 microseconds with an initial timestep of 0.1 nanoseconds.

Line 18: Indicates the end of the circuit description. All following statements will be related to the ATLAS device.

Lines 20-22: To completely specify the simulation, the physical models used by ATLAS must be identified. Note that DEVICE=ADIODE must be specified for each line. The MODEL statement is used to turn on the appropriate transport models. This set includes:

- conmob: the concentration dependent mobility mode,
- fldmob: the lateral electric field-dependent mobility model,
- consrh: Shockley-Read-Hall recombination using concentration dependent lifetimes,
- auger: recombination accounting for high level injection effects,
- bgn: band gap narrowing.

The MATERIAL statement is used to override default material parameters. In this case, the carrier recombination fixed lifetimes are set. Finally, the Selberherr impact ionization model is enabled using the IMPACT statement with the SELB option.

Line 24: The METHOD statement specifies numerical options for the device simulation. The METHOD statement must come after all other device simulation statements.

Line 26: The command GO ATLAS or a QUIT statement is needed to initiate simulation. Since a plot of the final log file is desired, the GO ATLAS option is used to restart ATLAS after the end of the MIXEDMODE simulation.

Line 27: The TONYPLOT command is used to plot the resulting log file.

12.4: MixedMode Syntax

This section is split into two parts. The first part describes circuit element statements, which describe the netlist. The second part describe the control and analysis statements.

12.4.1: Circuit Element Statements

A – ATLAS device to be simulated using device simulation

Syntax

```
Axxx n1=name1 n2=name2 [n3=name3 ...] infile=filename [width=val]
```

Description

This statement defines a device to be represented by a numerical ATLAS model. The device description with all necessary information (geometry, mesh, doping, models, electrode names, and so on) must be available in a standard structure file prior to starting a MIXEDMODE simulation.

Axxx: Name of the element. It must begin with `A`.

n1: Circuit node to which the ATLAS device electrode with the name, `name1`, is connected. The ATLAS device must have at least two electrodes. The maximum number of electrodes allowed in ATLAS is 55. This means that up to 25 ATLAS devices can be specified. For example, 25 devices with 2 electrodes each, or 10 devices with 5 electrodes each can be specified. The ATLAS device models should be used sparingly, because it can be very time consuming. Use circuit models for less important circuit components to conserve CPU time.

infile: Name of standard structure format file with device geometry, mesh, doping, electrodes names, and so on. The number of electrodes and their names should match those mentioned in this statement. Optionally, this file can contain a solution, which MIXEDMODE will use as an initial guess (see the `.OPTIONS` statement for more details).

width: Device width. This is an optional parameter (`default=1`). All currents through ATLAS device terminals calculated using the 2-D ATLAS model will be multiplied by this parameter to account for the third dimension of the device. `width` can still be used as a multiplier to the ATLAS current if a 3D ATLAS structure is used in MIXEDMODE3D.

Example

```
ABJT1 3=EMITTER 4=BASE 6=COLLECTOR INFILE=BJT1.STR WIDTH=10
```

Note: Optional parameters for a statement are shown with square brackets (e.g., [n3=name3]).

B – User-defined two-terminal element

Syntax

```
Bxxx n+ n- INFILE=file_name FUNCTION=function_name
```

Description

Bxxx: User-defined two terminal element name. It must begin with B.

n+, n-: Positive and negative terminal nodes.

INFILE: Name of the text file (*file_name*) that contains C source code for a user-defined function that describes element behavior. This file can contain more than one function description.

function_name: Name of the function (“*function_name*”) from the file.

Example

```
B1 2 3 infile=ud.c function=rc
```

Note: For more information on User-defined Two-Terminal Elements, see the “User-Defined Model” section on page 12-40.

C – Capacitor

Syntax

```
Cxxx n+ n- value
```

Description

Cxxx: Name of a capacitor element. It must begin with “C”.

n+, n-: Positive and negative terminal nodes.

value: Capacitance in farads.

Example

```
Cload 3 0 1pF
```

D – Diode

Syntax

```
Dxxx n+ n- mname [area] [L=val] [W=val] [PJ=val] [WP=val] [LP=val]
[WM=val]
[LM=val] [OFF] [IC=val] [M=val] [TEMP=val] [DTEMP=val]
```

Description

Dxxx: Name of the diode element. It must begin with D.

n+, n-: Positive (anode) and negative (cathode) terminal nodes.

mname: Diode model name. It must refer to a diode model.

area: Area factor. The default is 1.0.

L: Length of the diode in meters. Used for LEVEL 3 diode model only.

W: Width of the diode in meters. Used for LEVEL 3 diode model only.

PJ: Periphery of the diode junction. Calculated from W and L if they are specified (in the LEVEL 3 diode model). The ISW and CJSW model parameters are affected by the value of PJ.

WP: Width of the polysilicon capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.

LP: Length of the polysilicon capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.

WM: Width of the metal capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.

LM: Length of the metal capacitor in meters. Used for LEVEL 3 diode model only. Default is 0m.

OFF: Sets ON/OFF startup condition for DC analysis. Default is ON.

IC: Initial voltage across the diode.

M: Multiplier used to describe multiple parallel diodes.

TEMP: Device operating temperature (^{o}C).

DTEMP: Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
D1 2 3 dmodel1
Dclmp 3 7 Diol 3.0 IC=0.3
```

Note: See the SMARTSPICE/UTMOST MODELING MANUAL VOLUME 2 for a complete description of the diode models.

E – Linear voltage controlled source

Syntax

```
Exxx n+ n- nc+ nc- gain
```

Description

Exxx: Name of the linear voltage controlled voltage source. It must begin with E.

n+, n-: Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.

nc+, nc-: Positive and negative controlling node numbers.

gain: Voltage gain.

The linear voltage-controlled voltage source is characterized by Equation 12-1.

$$v(v+,n) = gain*v(nc+,nc) \qquad 12\text{-}1$$

Example

```
ER 4 5 6 7 55
```

F – Linear current controlled current source

Syntax

```
Fxxx n+ n- vcontrolname gain
```

Description

Fxxx: Name of the linear current controlled current source. It must begin with F.

n+, n-: Positive and negative terminal nodes. A positive current flows from the node, n+, through the source to the node, n-.

vcontrolname: Name of the voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of **vcontrolname**.

gain: Current gain.

The linear current-controlled current source is characterized by Equation 12-2.

$$v(n+,n-) = gain*i(vcontrolname) \quad \text{12-2}$$

Example

```
F12 4 5 VIN 0.1
```

G – Linear voltage controlled current source

Syntax

```
Gxxx n+ n- nc+ nc- transconductance
```

Description

Gxxx: Name of the linear voltage controlled current source. It must begin with `G`.

n+, n-: Positive and negative terminal nodes. A positive current flows from the node, `n+`, through the source to the node, `n-`.

nc+, nc-: Positive and negative controlling node numbers.

transconductance: Transconductance (in 1/Ohms).

The linear voltage controlled current source is characterized by Equation 12-3.

$$i(n+,n-) = transconductance*v(nc+,nc-) \quad \text{12-3}$$

Example

```
G2 4 5 6 7 5.5
```

H – Linear current controlled voltage source

Syntax

```
Hxxx n+ n- vcontrolname transresistance
```

Description

Hxxx: Name of the linear current controlled voltage source. Must begin from `H`.

n+, n-: Positive and negative terminal nodes. A positive current flows from the node, `n+`, through the source to the node, `n-`.

vcontrolname: Name of voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of **vcontrolname**.

transresistance: transresistance (in Ohms).

The linear current controlled voltage source is characterized by Equation 12-4.

$$v(n+,n-) = transresistance*i(vcontrolname) \quad \text{12-4}$$

Example

```
H12 4 5 V1 0.1K
```

I – Independent current source

Syntax

```
Ixxx n+ n- value [AC acmag] [transient_parameters]
```

Description

Ixxx: Name of the independent current source. It must begin with `I`.

n+, n-: Positive and negative terminal nodes.

value: DC value of the source (ampers).

AC: Keyword for the AC source value.

acmag: AC magnitude.

transient_parameters: The transient parameters are described in Section 12.4.3: "Transient Parameters".

Example

```
I1 2 8 0. PULSE 0 200 0 20ns 20ns 100ns 10 100
I2 1 5 1u AC 2u
```

J – Junction Field-Effect Transistor (JFET)

Syntax

```
Jxxx nd ng ns [nb] mname [area] [M=val] [L=val] [W=val] [OFF]
[IC=vds,vgs]
[TEMP=val] [DTEMP=val]
```

Description

Jxxx: Name of the JFET element. It must begin with `J`.

nd, ng, ns, nb: Drain, gate, source and bulk terminal nodes. The bulk node doesn't need to be specified. If the bulk node is not specified, then the bulk is connected to the source node

mname: Model name. It must refer to a JFET model.

area: Area factor. The default is 1.0.

M: Multiplier used to describe multiple parallel JFETs.

L: Length of the gate in meters.

W: Width of the gate in meters.

OFF: Sets ON/OFF startup condition for DC analysis. Default is ON.

IC: Initial condition specification for *vds* and *vgs*.

TEMP: Device operating temperature (^{o}C).

DTEMP: Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
J44 1 4 6 jmodel
```

Note: See SMARTSPICE/UTMOST MODELING MANUAL, VOLUME 2 for a complete description of the JFET models.

K – Coupling between two inductors

Syntax

```
Kxxx Lyyy Lzzz kval
```

Description

This is not a real circuit element. This statement defines only the coupling between two inductors.

Kxxx: Name. This parameter is not important and is used only to distinguish the statement. It must begin with K.

Lyyy: First inductor element name. It must begin with an L and match one of the inductor names from the circuit.

Lzzz: Second inductor element name. It must begin with an L and match one of the inductor names from the circuit.

kval: Coefficient of mutual coupling, which must be in the range 0 < kval < 1. The mutual inductance M will be determined from Equation 12-5.

$$M = kval*L1*L2 \qquad 12\text{-}5$$

Example

```
K1 L22 LLOAD 0.99
```

L – Inductor

Syntax

```
Lxxx n+ n- value
```

Description

Lxxx: Name of the inductor. It must begin with L.

n+, n-: Positive and negative terminal nodes.

value: Inductance in henries.

Example

```
L2 2 3 2.5nH
```

M – MOSFET

Syntax

```
Mxxx nd ng ns [nb] mname [L=val] [W=val] [AD=val] [AS=val]
[PD=val] [PS=val] [NRD=val] [NRS=val] [OFF] [IC=vds,vgs,vbs] [M=val]
[TEMP=val] [DTEMP=val] [GEO=val] [DELVTO=val]
```

Description

Mxxx: MOSFET element name. It must begin with `M`.

nd, ng, ns, nb: Drain, gate, source, and bulk terminal nodes. The bulk terminal node name is optional. If it is unspecified, ground is used.

mname: Model name. It must refer to a MOSFET model.

L=val: Channel length in meters.

W=val: Channel width in meters.

AD: Drain diffusion junction area (meters^2). This default is 0.

AS: Source diffusion junction area (meters^2). This default is 0.

PD: Drain diffusion junction perimeter (meters). This default is 0.

PD: Source diffusion junction perimeter (meters). This default is 0.

NRD: The Number of squares of drain diffusion for resistance calculations. The default is 0.

NRS: The Number of squares of source diffusion for resistance calculations. The default is 0.

OFF: Sets ON/OFF startup condition for DC analysis. Default is ON.

IC: Initial voltage condition specification for *vds*, *vgs*, and *vbs*.

M: Multiplier used to describe multiple parallel MOSFETs. The default is 1.

TEMP: Device operating temperature (^{o}C).

DTEMP: Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

DELVTO: Threshold-voltage shift. When specified on the device line, the value overrides the value of the model parameter, `DELVTO`. If not specified, the value of the model parameter is used.

Example

```
M1 2 4 8 9 mod1
Mout2 19 20 21 0 nmos L=5u W=2u TEMP=50
M22 3 5 7 8 mosmod1 L=10u W=5u AD=150p AS=150p PD=50u
PS=50u NRD=10 NRS=20
```

Note: See SMARTSPICE/UTMOST MODELING MANUAL, VOLUME 1 for a complete description of the MOSFET models.

O – Optical source

Syntax

```
Oxxx beam value [transient_parameters]
```

Description

Oxxx: Name of an independent optical source. It must begin with `O`.

beam: Beam number. The beam with this number should be described in the ATLAS section of the command file. See Chapter 10: "Luminous: Optoelectronic Simulator" for a complete description of optoelectronic simulation.

value: DC optical intensity value (W/cm2).

transient_parameters: The transient parameters are described in Section 12.4.3: "Transient Parameters".

Note: The treatment of optical sources is fully similar to the treatment of independent voltage/current sources. In other words you can use, `.DC` statements to simulated DC light responses of the circuit and transient parameters in order to describe the transient behavior of the optical sources.

Example

```
O1 1 0.001 pulse 0.001 0.002 0 2ns 2ns 100ns 10 100
```

Q – Bipolar junction transistor

Syntax

```
Qxxx nc nb ne [ns] mname [area] [OFF] [IC=vbe,vce] [M=val] [TEMP=val]
[DTEMP=val]
```

or

```
Qxxx nc nb ne [ns] mname [area=val] [areab=val] [areac=val] [OFF]
[IC=vbe,vce] [M=val] [TEMP=val] [DTEMP=val]
```

Description

Qxxx: Name of a bipolar junction transistor. It must begin with `Q`.

nc, nb, ne, ns: Collector, base, emitter, and substrate nodes. The substrate terminal node name is optional. If it is unspecified, ground is used.

mname: Model name. It must refer to a BJT model.

area: Emitter area factor. The default value is 1.0.

areab: Base area factor. The default is `area`.

areac: Collector area factor. The default is `area`.

OFF: Sets ON/OFF startup condition for DC analysis. Default is ON.

IC: Initial voltage condition specification for *vbe*, *vce*.

M: Multiplier used to describe multiple parallel BJTs. The default is 1.

TEMP: Device operating temperature (^{o}C).

DTEMP: Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
Q1 2 3 9 npnmod 1.5 IC=0.6,5.0
Q9 10 11 12 20 mod22 OFF TEMP=50
```

Note: See SMARTSPICE MODELING MANUAL, VOLUME 2 for a complete description of the BJT models.

R – Resistor

Syntax

```
Rxxx n+ n- value [transient_parameters]
```

Description

Rxxx: Name of the resistor element. It must begin with "R".

n+, n-: Positive and negative terminal nodes.

value: Resistance in ohms.

transient_parameters: The transient parameters are described in Section 12.4.3: "Transient Parameters".

Note: Unlike the traditional SPICE program, transient parameters are acceptable for resistor elements. This allows simulation of different kinds of time-dependent resistors and switches in a simple way.

Example

```
R12 4 5 100k
```

T – Lossless transmission line

Syntax

```
Txxx n1 n2 n3 n4 Z0= val TD=val
```

Description

Txxx: Name of the transmission line element. It must begin with T.

n1, n2: Nodes at port 1.

n3, n4: Nodes at port 2.

Z0: Characteristic impedance.

TD: Transmission delay.

Example

```
T1 1 0 2 0 Z0=50 TD=10ns
```

V – Independent voltage source

Syntax

```
Vxxx n+ n- value [AC acmag] [transient_parameters]
```

Description

Vxxx: Name of the independent voltage source. It must begin with V.

n+, n-: Positive and negative terminal nodes.

value: DC value of the source in units of volts.

AC: Keyword for the AC source value.

acmag: AC magnitude.

transient_parameters: The transient parameters are described in Section 12.4.3: “Transient Parameters”.

Example

```
VCC 5 0 10.5
VIN 2 4 5.5 AC 1
```

X - Subcircuit Call

Syntax

```
Xxxx n1 <n2 n3 ...> subcktname <<PARAMS:> parname=val ...>
```

Xxxx: Subcircuit names must begin with X.

n1, **n2**, **...**: Node names for external reference.

subcktname: Subcircuit definition.

PARAMS: The optional parameter preceding the list of parameters and their values.

parname: The parameter name whose value is set to val, which will be local to the subcircuit. valis either a numerical value, an expression enclosed in single quotes containing previously defined parameters, or a string, which defines a model’s name, that can be transmitted in subcircuit. If the same parameter is assigned a value on more than one statement in the input deck (.PARAM, .SUBCKT, and X statements), then will be one of the following:

- a value assigned in a global .PARAM statement (outside subcircuits) is used if it exists.
- a value assigned in a corresponding X statement is used if it exists.
- a value assigned in a corresponding .SUBCKT statement is used if it exists.
- a value assigned in a local .PARAM statement (inside the subcircuit) is used if it exists.

This statement creates instance Xxxx of subcircuit subcktname. There must be the same number of nodes n1 . . . as in the subcircuit definition (see .SUBCKT statement). These nodes replace the formal node names in the .SUBCKT statement.

Subcircuits can be nested. The number of nesting levels is not limited but nesting must not be recursive. When the circuit is loaded, all subcircuit device names are expanded to the form devtype.subcktname.devname, where devtype is the first letter of devname. All local node names in the subcircuit are expanded to the form subcktname.nodename.

For nested subcircuits, expanded names have multiple dot-separated qualifiers. For example, if a circuit has a call to subcircuit `subcktname1` and the `subcktname1` definition contains a call to subcircuit `subcircuit2`, then expanded names will have the following format:

```
devtype.subcktname1.subcktname2.devname

subcktname1.subcktname2.nodename

subcktname1.subcktname2.modelname
```

Z – MESFET

Syntax

```
Zxxx nd ng ns [nb] mname [area] [M=val] [L=val] [W=val] [OFF]
[IC=vds,vgs]

[TEMP=val] [DTEMP=val]
```

Description

Jxxx: Name of the MESFET element. It must begin with `Z`.

nd, ng, ns, nb: Drain, gate, source and bulk terminal nodes. The bulk node doesn't need to be specified. If the bulk node it not specified, then the bulk is connected to the source node.

mname: Model name. It must refer to a MESFET model.

area: Area factor. The default is 1.0.

M: Multiplier used to describe multiple parallel MESFETs.

L: Length of the gate in meters.

W: Width of the gate in meters.

OFF: Sets ON/OFF startup condition for DC analysis. Default is ON.

IC: Initial condition specification for *vds* and *vgs*.

TEMP: Device operating temperature (^{o}C).

DTEMP: Difference (in ^{o}C) between the device operating temperature and the circuit temperature. Default value is 0.

Example

```
Z44 1 4 6 jmodel
```

Note: See SMARTSPICE MODELING MANUAL, VOLUME 2 for a complete description of the MESFET models.

12.4.2: Control and Analysis Statements

.AC

.AC performs an AC linear small-signal analysis on the circuit. MIXEDMODE first creates a linearized small-signal model at the operating point of the circuit and the computes the frequency response over a user-specified range of frequencies.

Syntax

```
.AC DEC|OCT|LIN numpfstart fstop/ sweep source_name start stop step
```

Description

DEC: Sweep frequency by decades.

OCT: Sweep frequency by octaves.

LIN: Linear frequency sweep. This is default.

nump: Total number of points per decade or per octave, or the total number of points of the linear sweep.

fstart: Starting frequency (Hz).

fstop: Final frequency (Hz).

Several .AC statements can be specified in the same command file. In this case, they will be executed sequentially. Before executing the first .AC statement, the program will execute all .DC statements (if any), regardless of the order of the .AC and .DC statements in the command file.

source_name: Name of the independent voltage, current, or optical source to be swept.

start: Starting value of the sweep.

step: Increment value of the sweep.

stop: Final value of the sweep.

sweep: Performs a DC sweep at every frequency point.

Example

```
.AC DEC 3 1.e3 1.e12
.AC LIN 20 1.e5 2.e6
```

.BEGIN

.BEGIN indicates the start of the circuit part of a MIXEDMODE command file. The synonym for this parameter is START.

```
.BEGIN 3D
```

Description

3D: Specifies that MIXEDMODE3D will be used instead of MIXEDMODE.

.DC

.DC causes a DC transfer curve to be computed for the circuit with all capacitors opened and all inductors shorted.

Syntax

```
.DC DEC|OCT|LIN source_name start stop numbers_steps
source_name2 DEC|OCT|LIN start2 stop2 number_steps2
```

Description

DEC: Sweep DC bias (voltage or current) by decades.

OCT: Sweep DC bias by octaves.

LIN: Linear DC bias sweep. This is the default.

source_name: Name of the independent voltage, current, or optical source to be swept.

start: Starting value of the sweep argument.

stop: Final value of the sweep argument.

number_steps: Number of steps of the inner sweep

source_name2: Name of the secondary sweep source.

start2: start value of the secondary sweep source

stop2: Final value of the secondary sweep source

number_steps2: Number of steps of the secondary sweep.

Several .DC statements can be specified in a command file. In this case, they will be executed sequentially. Before executing the first .DC statement, the program will simulate the circuit with the independent source values given in the description of those sources.

The .DC statement is also often used to increment the values of independent voltage and current sources in a circuit to avoid convergence problems.

Examples

```
.DC VIN 0. 5. 0.25
.DC IE 50 500 50
```

.END

.END indicates the end of the circuit part of a MIXEDMODE command file. The synonym for this parameter is FINISH.

.IC

.IC sets specified node voltages during the steady-state simulation.

Syntax

```
.IC [V(I)=val_I...]
```

Description

This statement forces the specified node voltages to be set to specified values during the steady-state simulation. These voltages are release when the transient simulation begins.

Example

```
.IC V(1)=10
.IC V(node1)=-0.5
```

.LOAD

.LOAD loads a solution file.

Syntax

```
.LOAD INFILE=filename
```

Description

INFILE: Name of a file (*filename*) to be loaded as an initial guess for further simulation. This file must have been saved during a previous run of MIXEDMODE using the.SAVE statement.

Example

```
.LOAD INFILE=pdsave
```

Note: This statement is not used to load SSF format solution files from ATLAS (see.OPTIONS LOADSOLUTIONS)

.LOG

.LOG specifies the filename for the circuit voltages and currents that will be saved.

Syntax

```
.LOG OUTFILE=filename
```

Description

OUTFILE: Name of a file (*filename*) for the circuit voltages and currents to be saved in standard structure format files.

These files will have the following names:

For steady-state analysis:

```
"filename"_dc_1.log
"filename"_dc_2.log
"filename"_dc_3.log
..
```

(new file will be created for each .DC statement).

For AC analysis:

```
"filename"_ac_1.log
..
For network parameter extraction:
"filename"_net_1.log
..
```

For transient analysis:

```
"filename"_tr.log
..
```

To plot results of and entire steady-state analysis simultaneously, load all files related to steady-state analysis into TONYPLOT.

Example

```
.LOG OUTFILE=pd
```

.MODEL

.MODEL specifies the circuit element model to be used for diodes, BJTs, or MOSFETs, and the numerical values of parameters associated with the model.

Syntax

```
.MODEL name type <parameters>
```

Description

name: This is the model name. Circuit element definition statements refer to this name to link elements to models.

type: This is the model type. This type must be consistent with the type of the circuit elements that uses the model. The type can be one of the following:

- **D** - Diode model
- **NMOS** - n-channel MOSFET model.
- **PMOS** - p-channel MOSFET model.
- **NPN** - npn BJT model
- **PNP** - pnp BJT model
- **NJF** - n-channel JFET/MESFET model
- **PJF** - p-channel JFET/MESFET model

parameters: Model parameters. The parameters are described in the SMARTSPICE/UTMOST MODELING MANUALS VOLUMES 1,2 and 3.

Example

```
.MODEL MODBJT NPN IS=1.E-17 BF=100 CJE=1F TF=5PS \
CJC=0.3F RB=100 RBM=20
```

.NET

.NET specifies that a network parameter extraction is to be performed.

Syntax

```
.NET INPORT OUTPORT DEC|OCT|LIN nump fstart fstop [Z0] [INDIN] [RSIN]
[INDOUT]] [RSOUT] [CIN] [COUT]
```

Description

INPORT: Input port description. It should be in one of the following formats:

V(n+,n-): two nodes (positive (n+) and negative (n-)).

Vxxxx: where Vxxxx is the name of an existing voltage source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node.

Ixxxx: where Ixxxx is the name of an existing current source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node.

Note: If the nodes specified as the input port are the same nodes as an existing current or voltage source, then the name of the source must be specified as inport. Also, remove all AC parameters from voltage or current sources before using the .NET statement.

OUTPORT: Output port description. It should be in one of the following formats.

V(n+,n-): two nodes (positive (n+) and negative (n-)).

Vxxxx: where Vxxxx is the name of an existing voltage source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node.

Ixxxx: where Ixxxx is the name of an existing current source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node.

Note: If the nodes specified as the output port are the same nodes as an existing current or voltage source, then the name of the source must be specified as outport.

DEC: Sweep frequency by decades.

OCT: Sweep frequency by octaves.

LIN: Linear frequency sweep. This is default.

nump: Total number of points per decade or per octave, or the total number of points of the linear sweep.

fstart: Starting frequency (Hz).

fstop: Final frequency (Hz).

Additional optional parameters may also be specified on the .NET statement.

Z0: Matching Impedance (default = 50 Ohms).

INDIN: Inductance through which the DC voltage source is connected to the input source (only if INPORT is given as Vxxxx).

RSIN: Series resistance of INDIN.

INDOUT: Inductance through which the DC voltage source is connected to the output source (only if OUTPORT is given as Vxxxx).

RSOUT: Series resistance of INDOUT.

CIN: Capacitance through which the S-parameter test circuit is connected to the input port.

COUT: Capacitance through which the S-parameter test circuit is connected to the output port.

Note: The S-parameters will be automatically saved to the log file. The Z, Y, H, ABCD, and gain small-signal parameters can also be written to the log file. These are selected through the .OPTIONS statement. You can also view the default values if PRINT is specified in the .OPTIONS statement

---

Examples

```
.NET V1 V2 DEC 10 1e6 1e10
.NET I1 V2 DEC 10 1e6 1e10 Z0=75 RSOUT=100
.NET V(1,0) V(2,3) DEC 10 1e6 1e10
```

.NODESET

.NODESET sets initial values for circuit node voltages.

Syntax

```
.NODESET [V(I)=VAL_I ...]
```

Description

This statement specifies the initial values for circuit node voltages. If a node voltage is not specified, the program will try to find a solution using zero as an initial guess for this node. This statement can significantly reduce the CPU time needed to calculate the initial condition.

Example

```
.NODESET V(1)=50 V(2)=49.4 V(3)=10 V(5)=-1.5
.NODESET V(in1)=0 V(2)=2 V(out1)=-1
```

.NUMERIC

.NUMERIC specifies special numeric parameters for the circuit analysis.

Syntax

```
.NUMERIC   [parameters]
```

Parameter	Type	Default	Units
IMAXDC	Integer	25	
IMAXTR	Integer	15	
DTMIN	Real	$1*10^{-12}$	s
LTE	Real	0.1	
TOLDC	Real	$1*10^{-4}$	

Parameter	Type	Default	Units
TOLTR	Real	$1*10^{-4}$	
VCHANGE	Real	$5*10^{7}$	V
VMAX	Real	$5*10^{7}$	V
VMIN	Real	$-5*10^{7}$	V

Description

IMAXDC: Maximum number of mixed circuit-device iterations to be performed during steady-state analysis.

IMAXTR: Maximum number of mixed circuit-device iterations to be performed during transient analysis.

DTMIN: Minimum time step value for transient analysis.

LTE: Local truncation error for transient analysis.

TOLDC: Relative accuracy to be achieved during steady-state analysis for the calculation of voltages in circuit nodes.

TOLTR: Relative accuracy to be achieved during transient analysis for the calculation of voltages in circuit nodes.

VCHANGE: Maximum allowable change in circuit node voltages between two mixed circuit-device iterations. This parameter can be useful for reaching steady-state convergence with a bad initial guess.

VMAX: Maximum value for circuit node voltages.

VMIN: Minimum value for circuit node voltages.

Example

```
.NUMERIC LTE=0.05 TOLDC=1.*10-8 DTMIN=1ns
```

.OPTIONS

.OPTIONS specifies various circuit simulation options.

Syntax

```
.OPTIONS [parameters]
```

Parameter	Type	Default	Units
ABCD.PARAM	Logical	False	
CNODE	Real	$1*10^{-16}$	F
CYLINDR	Logical	False	
DC.WRITE	Real	1	
DT.NOCHECK	Logical	False	
FULLN	Logical	True	

Parameter	Type	Default	Units
GAINS	Logical	False	
H.PARAM	Logical	False	
LOADSOLUTIONS	Logical	False	
M2LN	Logical	False	
M2LN.TR	Logical	False	
NOPROJ	Logical	False	
NOSHIFT	Logical	False	
PRINT	Logical	False	
RELPOT	Logical	False	
RV	Real	$1*10^{-4}$	W
TEMP	Real	300	K
TNOM	Real	300	K
WRITE	Integer	1	
Y.PARAM	Logical	False	
Z.PARAM	Logical	False	
ZIP.SOLVER	Logical	False	

Description

ABCD.PARAM: ABCD parameters will be written to the log file. This is used in conjunction with the .NET statement.

CNODE: A very small capacitance, which for algorithmic reasons automatically connected from each circuit node to ground. This value can be set to 0.

CYLINDR: Cylindrical coordinate system for all ATLAS devices.

DC.WRITE specifies how often a structure file (as specified by the MASTER parameter on the .SAVE statement) will be written during .DC solutions. If this parameter is set to 0, then no .DC structure files will be written.

DT.NOCHECK specifies that the transient pulse and table definitions will not be taken into account when calculating the time step.

FULLN: Full Newton solution method is used during steady-state simulation.

GAIN: Stability factor (K), unilateral power gain (GU), maximum unilateral transducer power gain (GTUmax) and $|H21|^2$ are written to the LOG file. This is used in conjunction with the .NET statement.

H.PARAM: H-parameters will be written to the LOG file. This is used in conjunction with the .NET statement.

LOADSOLUTIONS: Solutions, structures, doping distributions, and meshes, are to be loaded from standard structure files. The solutions are used as the initial guess or initial conditions for subsequent MIXEDMODE simulation. To use this feature, you must:

- Calculate a solution for each ATLAS device and save each solution in a separate standard structure format file using SAVE or SOLVE.... MASTER.
- For each ATLAS device, use the A statement in the MIXEDMODE command file to specify the associated standard structure format file
- Set node voltages to appropriate values with the .NODESET statement
- Specify LOADSOLUTIONS in the .OPTIONS statement

Note: If using this feature, specify solutions for all ATLAS devices.

The .NODESET statement must always be used when LOADSOLUTIONS is used. The .NODESET statement is used to make the initial circuit voltages match those device solutions that were obtained. You may also need to specify the NOSHIFT parameter of the OPTIONS statement. By default, MIXEDMODE shifts device terminal voltages with respect to the voltage on the first terminal that is specified in the A statement. You must either prepare initial solutions with this terminal grounded, or specify NOSHIFT in the OPTIONS statement.

M2LN: Uses the modified two-level Newton solution method during steady-state simulation. The full NEWTON method provides faster solution than the modified two-level Newton method when a good initial guess is available. The modified two-level method is more reliable when the initial guess is far from the solution. The default is the full NEWTON method.

M2LN.TR: Uses the modified two-level Newton solution method during transient simulations.

NOPROJ: Disables the initial guess project method for the ATLAS iterations. MIXEDMODE attempts to extrapolate the values of the ATLAS device variables (such as potential and carrier concentration) for the each iteration. Specifying NOPROJ disables the extrapolation and the previous values of potential and carrier concentration are used instead.

NOSHIFT: Disables the shift of voltages for ATLAS device models. MIXEDMODE normally shifts the voltages on ATLAS device terminals to be referenced to the voltage on the first terminal. From the physical point of view, the state of the p-n diode is the same for voltages of 0V and 0.5V on the diode terminals with 1000V and 1000.5V, but the first situation is better for numerical simulation.

PRINT: Enables printing of circuit nodes voltages after the calculation for each bias point (DC analysis) or time step (transient the analysis).

RELPOT: Enables the use of relative convergence criteria for potential for ATLAS models. By default, ALTAS models use absolute convergence criteria for potential. When bias voltages are large (a common situation for power devices), then absolute convergence criteria are not appropriate and this parameter should be specified.

RV: Defines the ohmic resistance that MIXEDMODE associates with all voltage sources and all inductances. This value should never be set to 0. The default value is small enough to avoid errors due to the influence of the internal resistance. Usually, extremely small values of this parameters can cause convergence problems. It is usually acceptable to decrease this parameter to the range of $1*10^{-6}$-$1*10^{-7}$. This parameter should not be varied unless there is a compelling reason to do so.

TEMP: Device temperature to be use during the simulation.

TNOM: Circuit temperature to be use during the simulation.

WRITE: How often the solution is to be saved in standard structure files during the simulation. For example, write=3 specifies that the solution will be saved at every third timestep. Specifying this parameter can help avoid disk overflow.

Y.PARAM: Y-parameters should be written to the log file. This is used in conjunction with the .NET statement.

Z.PARAM: Z-parameters should be written to the log file. This is used in conjunction with the .NET statement.

ZIP.SOLVER: Specifies that the ZIP libary version of BICGST iternative solver will be used to calculate the device conduction matrix instead of the CGS or BICGST solver. If you specify the DIRECT parameter in the METHOD statement, then the direct solver will be used for both the device conduction matrix calculation and the ATLAS solution.

Example

```
.OPTIONS TNOM=293 FULLN
```

.PRINT

.PRINT specifies which device output parameters will be printed to the log files.

Syntax

```
.PRINT [antype]  parmeter(device_name) [parmeter2(device_name) ...]
```

Description

antype: Type of analysis for which the outputs are desired. If antype is unspecified, the outputs of all simulation types will be printed. antype must be one of the following keywords:

- **AC**: AC analysis outputs
- **DC**: DC analysis outputs
- **NET**: Network analysis outputs
- **TRAN**: Transient analysis outputs
- **parmeter:** Output variables or expressions to be printed.
- **device_name:** The device name.

Example

```
.PRINT ic(q1) ib(q1) is(q1)

.PRINT AC ic(q1) ib(q2)

.PRINT DC ic(q1) ib(q2) i(d1)

.PRINT TRAN cd(m1) cg(m1) cs(m1) cb(m1)
```

.PARAM

.PARAM specifies a parameter definition.

Syntax

```
.PARAM parname1=val1 [parname2=val2 ...]
```

The .PARAM statement defines one or more parameter labels by assigning the names parname1, parname2 to constant values val1, val2. Parameter labels must begin with alphabetic character(s) or the underscore character(s).

The .PARAM statement can be globally and locally (i.e., inside a subcircuit definition) specified. Once defined, a global parameter label or an expression containing global parameter labels can be used in the input deck when a numerical value is expected. A local parameter label can be used only inside the subcircuit.

.SAVE

.SAVE saves simulation results into files for visualization or for future use as an initial guess.

Syntax

```
.SAVE OUTFILE=name [MASTER=mname][TSAVE=timepts]
```

Description

MASTER:

Example

```
.SAVE MASTER=my_filename.str
```

This will cause MIXEDMODE to output standard structure files for visualization in TONYPLOT. These files, with the base name, mname, will be written after the calculation of each bias point during DC simulation, and after of each time step during transient simulation. For example, if you wish to save a structure file for each DC bias solution, you would write:

```
.save master=my_structure.str
```

MIXEDMODE will then output structure files at each DC bias point. MIXEDMODE will also automatically append a number to the end of the file name indicating which bias point the file is for.

OUTFILE:

Example

```
.save outfile=my_filename.str
```

Specifies that after the simulation is finished the solution is to be written to a file called my_filename.str. In other words, only one structure file will output after the simulation has finished. Even if you have several .dc solutions, MIXEDMODE will only output one structure file if for example:

```
.save outfile=filename.str
```

has been used. The ATLAS model solutions will be written to the file (name) and the circuit solution will be written to the file, name.cir. These files can be used later for loading solutions to be used as an initial guess (see .LOAD statement).

TSAVE: Specifies the transient time points at which a MASTER file will be written. The time points should be specified as a comma seperated list.

The program will automatically add the following suffixes to mname:

- During DC simulation: _dc_number, where number is the number of the DC point.
- During transient simulation: _tr_number, where number is the number of the time step.

Example

```
.SAVE OUTFILE=pdsave MASTER=pd
```

.SUBCKT

.SUBCKT specifies a subcircuit defintion.

Syntax

```
.SUBCKT subcktname <n1 n2 ...>
+ <OPTIONAL: optn1=defval1 <optn2=defval2 ...>>
+ <<PARAMS|PARAM:> parname1=val1 <parname2=val2 ...>>
```

subcktname: The subcircuit name. It is used by an X element statement to reference the subcircuit.

n1 ...: The optional list of external nodes. Ground nodes (zero) are not allowed.

PARAMS|PARAM: The parameter preceding the list of parameters and their values.

parname1, parname2, ...: The name of a parameter with value set to val1, val2, . . . that will be local to the subcircuit. val is a numerical value. If the same parameter is assigned, a value on more than one statement in the input deck (.PARAM, .SUBCKT, and X statements) will be one of the following:

- a value assigned in a global .PARAM statement (outside subcircuits) is used if it exists.
- a value assigned in a corresponding X statement is used if it exists.
- a value assigned in a corresponding .SUBCKT statement is used if it exists.
- a value assigned in a local .PARAM statement (inside the subcircuit) is used if it exists.

A subcircuit definition is initiated by a .SUBCKT statement. The group of element statements, which immediately follow the .SUBCKT statement define the subcircuit. The last statement in a subcircuit definition is the .ENDS statement. Control statements cannot appear within a subcircuit definition; however, subcircuit definitions may contain anything else, including other subcircuit definitions, device models, and subcircuit calls.

All internal nodes (if not included on the .SUBCKT statement) are local with the exception of the global node 0.

.TRAN

.TRAN specifies that a transient analysis is to be performed.

Syntax

```
.TRAN tstep tstop
```

Description

tstep: Time interval in seconds.

tstop: Final time value for which the simulation is to be performed.

Transient analysis is performed only after the execution of all .DC statements. If no DC statements are used, the transient starts after the calculation of the initial circuit state with the values of the independent sources given in the descriptions of those sources.

Multiple .TRAN statements are supported. The TSTOP parameter is not reset between each .TRAN statement.

Example

```
.TRAN 1ns 100ns
```

12.4.3: Transient Parameters

MIXEDMODE allows you to specify transient parameters for voltage sources (Vxxx), current sources (Ixxx), and resistors (Rxxx). These parameters describe the time development behavior of the source.

EXP

EXP is used to define an exponential waveform. The waveform is specified as follows:

```
EXP i1 i2 td1 tau1 td2 tau2
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td1** is the rise delay time.
- **td2** is the fall delay time.
- **tau1** is the rise time constant.
- **tau2** is the fall time constant.

The transient behavior is shown in Table 12-2.

Table 12-2. Transient behavior for EXP

Time	Value
0 < t < td1	i1
td1 ≤ t < td2	i1+(i2-i1).(1-exp[-(t-td1)/tau1])
td2 ≤ t	i1+(i2-i1).(1-exp[-(t-td1)/tau1]) +(i1-i2).(1-exp[-(t-td2)/tau2])

GAUSS

GAUSS is used to define a Gaussian waveform. The waveform is specified as follows:

```
GAUSS i1 i2 td1 tau1 td2 tau2
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td1** is the rise delay time.
- **td2** is the fall delay time.
- **tau1** is the rise time constant.
- **tau2** is the fall time constant.

The transient behavior is shown in Table 12-3.

Table 12-3. The Transient behavior for GAUSS	
Time	**Value**
0 < t < td1	i1
td1 ≤ t < td2	i1+(i2-i1).(1-exp[((-t-td1)/tau1)^2]
td2 ≤ t	i1+(i2-i1).(1-exp[((-t-td1)/tau1)^2] +(i1-i2).(1-exp[((-t-td2)/tau2)^2]

PULSE

PULSE is used to define a pulse waveform. The waveform is specified as follows:

```
PULSE i1 i2 td tr tf pw per
```

where:

- **i1** is the initial value.
- **i2** is the pulsed value.
- **td** is the delay time before the pulse is started.
- **tr** is the rise time of the pulse.
- **tf** is the fall time of the pulse.
- **pw** is the pulse length per period
- **per** is the period.

The transient behavior is described in Table 12-4. Intermediate points are found by linear interpolation.

Table 12-4. Transient behavior for PULSE	
Time	**Value**
0	i1
td	i1
td + tr	i2
td + tr + pw	i2
td + tr + pw + tf	i1
td + per	i1
td + per + tr	i2 (second period)

PWL

PWL is used to define a piece-wise linear waveform. The waveform is specified as a list of time and value point with the time values in ascending order. The waveform is specified as follows:

```
PWL t1, v1 [t2,v2 ....]
```

where:

- **t1** is the first time value.
- **v1** is the bias value at `time=t1`.
- **t2** is the second time value (`t2>t1`).
- **v2** is the bias value at `time=t2`.

SFFM

SFFM is used to define a modulated sinusoidal waveform. The waveform is specified as follows:

```
SFFM io ia fc mdi fs
```

where:

- **io** is the DC offset.
- **ia** is the amplitude.
- **fc** is the carrier frequency.
- **mdi** is the modulation index.
- **fs** is the signal frequency.

The transient behavior will be:

$$value(t)=io + ia \cdot \sin[\pi \cdot fc \cdot t + mdi \cdot \sin(2\pi \cdot fs \cdot t)]$$

SIN

SIN is used to define a sinusoidal waveform. The waveform is specified as follows:

```
SIN io ia freq td theta
```

where:

- **io** is the offset.
- **ia** is the amplitude.
- **freq** is the frequency.
- **td** is the delay.
- **theta** is the damping factor.

The transient behavior is shown in Table 12-5.

Table 12-5. Transient Behavior for SIN

Time	Value
$t < td$	$value(t) = io$
$t \geq td$	$value(t) = i0+ia.\exp[(-t-td)/THETA] .\sin[2\pi.freq.(t-td)]$

TABLE

TABLE is used to define a waveform using a table of values. This parameter is used as follows:

```
TABLE infile=<table_file_name>
```

where table_file_name is an ASCII text file that contains the tabulated time-dependence of a variable in the following format:

```
t1 v1
t2 v2
t3 v3
...
tN vN
end
```

Each line contains two numbers. The first number is the time in seconds. The second number is the time-dependent variable) voltage in volts, the current in amps, or the resistance in ohms). Up to 1000 lines can be used. Input is terminated by the word end.

If during the simulation the transient time becomes larger than the last value in the table, then the last value will be used for the remainder of the simulation.

12.4.4: User-Defined Two-Terminal Elements

MIXEDMODE users who are familiar with C-INTERPRETER can define their own two-terminal elements using the B statement and a function written in C that defines the behavior of the element.

User-Defined Model

A user-defined model is specified by defining the following:

- The dependencies of the device terminal current on the terminal voltages
- The derivatives of the terminal current with respect to the terminal voltages
- The device current (I) is described by Equation 12-6.

$$I = F1(U,t) + F2(U,t) \cdot \left(\frac{dU}{dt}\right) \qquad 12\text{-}6$$

where:

U is the device voltage, t is time, and $F1$ and $F2$ are functions that determine the behavior of the device. The first term on the right hand side describes the "DC" current and the second term describes "capacitive" current.

The following four functions are user-specified:

- **F1(U,t):** the DC current.
- **F2(U,t):** the capacitance.
- **dF1(U,t)/dU:** the DC differential conductance.
- **Q:** the charge associated with F2(U,t).

To define the element, prepare a text file that contains an appropriate function written in C. A template for this user-defined function is shown below:

```
int udef(double v, double temp, double ktq, double time, double *curr,
 double *didv, double *cap, double *charge)
{
```

```
/* user-supplied code here  */
return(0);
}
```

Input Parameters

Four input parameters are supplied to the function and can be used in the user-defined code. The input parameters are:

- **v:** the voltage across the element (V)
- **temp:** the temperature (K)
- **ktq:** the thermal voltage kT/q (V)
- **time:** transient time (sec); a value of 0 is supplied during DC calculations

Output Parameters

The four output parameters that must be returned by the function are:

- **curr:** the value of F1 (Amps)
- **didv:** the value of dF1(v, time)/dU (A/V)
- **cap:** the value of F2(v, time)
- **charge:** the value of the charge (Q)

Example

Consider an element that consists of a resistor R and a capacitor C connected in parallel. The equation for the total current through this combination is:

$$I(U,t) = \frac{U}{R} + C \cdot \left(\frac{dU}{dt}\right) \quad \text{12-7}$$

The quantities that must be user-defined are:

$$F1(U,t) = \frac{U}{R} \quad \text{12-8}$$

$$F2(U,t) = C \quad \text{12-9}$$

$$\frac{dF(U,t)}{dU} = \frac{1}{R} \quad \text{12-10}$$

$$Q = C \cdot U \quad \text{12-11}$$

When R=2kΩ and C=100pF, a user-defined function could have the following form:

```
intrc(double v, double temp, double ktq, double time, double *curr, double
*didv, double *cap, double *charge)
          {
       *curr = v/2000.0;
       *didv = 1.0/2000.0
       *cap = 1.0e-10;
       *charge=1.0e-10*v;
```

```
        return(0);                      /* 0 - ok */
    }
```

Chapter 13:
Quantum: Quantum Effect Simulator

13.1: Quantum Effects Modeling

There are several quantum effects models that work with ATLAS to simulate various effects of quantum mechanical confinement. These models are generally independent and should not be used simultaneously in any given simulation. These models are described in the following sections.

In Section 13.2: “Self-Consistent Coupled Schrodinger Poisson Model”, we discuss the self-consistent Schrodinger-Poisson model. This model self-consistently solves Poisson's equation (for potential) and Schrodinger's equation (for bound state energies and carrier wavefunctions). This model should not be used to look at carrier transport problems.

In Section 13.3: “Density Gradient (Quantum Moments Model)”, we describe the density gradient model. This model is based on the moments of the Wigner Function Equations of Motion and calculates a quantum correction to the carrier temperatures in the transport equations. This model can accurately reproduce the carrier concentration predicted by the Schrodinger-Poisson model but can also predict transport properties. This model cannot, however, predict the bound state energies or wave functions given by the Schrodinger-Poisson model.

In Section 13.4: “Bohm Quantum Potential (BQP)”, we describe the Bohm quantum potential model. This model can be used to address a similar set or problems as the density gradient model. The Bohm quantum potential model has two advantages over the density gradient model: 1) better convergence and 2) better calibration to the Schrodinger-Poisson model.

In Section 13.5: “Quantum Correction Models”, we describe two quantum correction models. These models by Van Dort and Hansch are both phenomenological models that give corrections for the effects of quantum confinement in the inversion layer under the gate of MOSFET devices.

In Section 13.6: “General Quantum Well Model”, we describe a general quantum well model used to predict the gain and spontaneous recombination in quantum well light emitting devices. This model is based on Schrodinger-Poisson modelling and is used to calculate the bound state energies and wave functions. The bound state energies are used to predict the gain and spontaneous recombination rates and the wave functions can be used to predict the overlap integral.

In Section 13.7: “Multiple Quantum Well Model”, we describe a similar quantum well model to that described in Section 13.6: “General Quantum Well Model”. This model is an older approach and is superseded by the newer General Quantum Well Model. The description in Section 13.7: “Multiple Quantum Well Model” in only included for backward compatibility. We do not recommended that you use this method.

13.2: Self-Consistent Coupled Schrodinger Poisson Model

To model the effects of quantum confinement, ATLAS also allows the solution of Schrodinger's Equation along with the fundamental device equations. The solution of Schrodinger's Equation gives a quantized description of the density of states in the presence of quantum mechanical confining potential variations.

The self-consistent coupled Schrodinger-Poisson model is enabled for electrons by setting the SCHRO parameter of the MODEL statement. For holes, the Schrodinger-Poisson model is enabled by setting the P.SCHRO parameter in the MODEL statement.

When the quantum confinement is in one dimension (along y-axis), the calculation of the quantized density of states relies upon a solution of a 1D Schrodinger equation solved for eigen state energies $E_{i\nu}(x)$ and wavefunctions $\Psi_{i\nu}(x,y)$ at each slice perpendicular to x-axis and for each electron valley (or hole band) ν.

$$-\frac{\hbar^2}{2}\frac{\partial}{\partial y}\left(\frac{1}{m_y^{\nu}(x,y)}\frac{\partial \Psi_{i\nu}}{\partial y}\right)+E_C(x,y)\,\Psi_{i\nu}=E_{i\nu}\,\Psi_{i\nu} \qquad 13\text{-}1$$

Here, $m_y^{\nu}(x,y)$ is a spatially dependent effective mass in y-direction for the ν-th valley and $E_C(x,y)$ is a conduction band edge. The equation for holes is obtained by substituting hole effective masses instead of electron ones and valence band edge $-E_V(x,y)$ instead of $E_C(x,y)$.

When the confinement is in two dimensions (x-y plane), specify the option SCHRO.2DXY on the MODEL statement and a 2D Schrodinger equation will be solved.

$$-\frac{\hbar^2}{2}\left[\frac{\partial}{\partial x}\left(\frac{1}{m_x^{\nu}(x,y)}\frac{\partial \Psi_{i\nu}}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{m_y^{\nu}(x,y)}\frac{\partial \Psi_{i\nu}}{\partial y}\right)\right]+E_C(x,y)\,\Psi_{i\nu}=E_{i\nu}\,\Psi_{i\nu} \qquad 13\text{-}2$$

In ATLAS3D, a 2D Schrodinger equation is solved in y-z slices, which are perpendicular to x-axis. The equation is analogous to Equation 13-2.

The electron valleys denoted in TONYPLOT as **Longitudinal**, **Transverse1** and **Transverse2** correspond to the effective mass along the y-axis being respectively ML, MT1, and MT2 on the MATERIAL statement. Table 13-1 shows the effective mass for each valley along all the axes.

Table 13-1. Effective Mass

Valley	mx	my	mz
Longitudinal	MT1	ML	MT2
Transverse1	MT2	MT1	ML
Transverse2	ML	MT2	MT1

In the case where the mass is isotropic (i.e., is the same in all directions), only one solution to Schrodinger's equation is obtained with the appropriate mass. ATLAS will automatically determine the appropriate number of valleys based on the material in question. You can, however, limit the number of directions in k space by using the NUM.DIRECT. To choose the number of directions in k space, specify the NUM.DIRECT parameter on the MODELS statement. If you set NUM.DIRECT to 1, you will obtain the solution for isotropic effective mass (MC on the MATERIAL statement). If you set NUM.DIRECT to 3, you will obtain a solution for a single lateral effective mass and two transverse

masses (ML, MT1 and MT2 on the MATERIAL statement). In a special case of a 1D confinement and equivalent transverse masses, you can set NUM.DIRECT to 2 and only two solutions will be obtained (ML and MT1) with appropriate degeneracy factors.

To specify how many valence bands to consider, specify the NUM.BAND parameter. If you set NUM.BAND to 1, you will obtain a Schrodinger solution for only one valence band (MV on the MATERIAL statement). If you set NUM.BAND to 3, you will cause solutions for heavy holes, light holes and holes in the split off band (MHH, MLH, and MSO on the MATERIAL statement).

Using Fermi-Dirac statistics, the discrete nature of the quantized density of states reduces the integral over energy to a sum over bound state energies. The expression for the electron concentration then becomes

$$n(x,y) = 2\frac{k_B T}{\pi h^2}\sum_{\nu}\sqrt{m_x^{\nu}(x,y)m_z^{\nu}(x,y)}\sum_{i=0}^{\infty}\left|\Psi_{i\nu}(x,y)\right|^2 ln\left[1+exp\left(-\frac{E_{i\nu}-E_F}{k_B T}\right)\right] \quad \text{13-3}$$

for 1D confinement and

$$n(x,y) = 2\frac{\sqrt{2k_B T}}{h}\sum_{\nu}\sqrt{m_z^{\nu}(x,y)}\sum_{i=0}^{\infty}\left|\Psi_{i\nu}(x,y)\right|^2 F_{-1/2}\left(-\frac{E_{i\nu}-E_F}{k_B T}\right) \quad \text{13-4}$$

for a 2D confinement case, where $F_{-1/2}$ is the Fermi-Dirac integral of order *-1/2*.

With this parameter set, ATLAS solves the one-dimensional Schrodinger's Equation along a series of slices in the y direction relative to the device.

The location of the slices in the y direction is developed in two ways. For rectangular ATLAS-defined meshes, the slices will automatically be taken along the existing mesh lines in the ATLAS mesh. If the mesh is non-rectangular or not an ATLAS defined mesh or both, you must specify a rectangular mesh. To do this, specify the locations of individual mesh lines and their local spacings using the SX.MESH and SY.MESH statements like that to the specification of a device mesh using the X.MESH and Y.MESH or a laser mesh using the LX.MESH and LY.MESH statements.

If you want, you may specify a mesh for Schrodinger's Equation to solve on an ATLAS rectangular mesh using the SX.MESH and SY.MESH statements. But since, by default, the solver uses the ATLAS mesh, you must specify NEW.SCHRODINGER in the MODEL statement to use the separate mesh.

Once the solution of Schrodinger's Equation is taken, carrier concentrations calculated from Equation 13-3 are substituted into the charge part of Poisson's Equation. The potential derived from solution of Poisson's Equation is substituted back into Schrodinger's Equation. This solution process (alternating between Schrodinger's and Poisson's equations) continues until convergence and a self-consistent solution of Schrodinger's and Poisson's equations is reached.

Since the wavefunctions diminish rapidly from the confining potential barriers in the Schrodinger solutions, the carrier concentrations become small and noisy. You can refine these carrier concentrations by setting a minimum carrier concentration using the QMINCONC parameter on the MODELS statement. This parameter sets the minimum carrier concentration passed along to the Poisson solver and the output to the structure files. The transition between the Schrodinger solution and the minimum concentration is refined between 10×QMINCONC and QMINCONC so that it is continuous in the first derivative.

Use the SAVE statement or the OUTFILE parameter on the SOLVE statement to write the solutions of the self-consistent system into a structure file. These structure files will then contain the self-consistent potential and electron or hole concentrations.

The Eigen energies and functions can also be written to the structure file by specifying the EIGENS parameter on the OUTPUT statement. This parameter specifies the number of Eigen energies/wave functions to be written.

The number of Eigen values solved is limited to a number of 2 less than the total number of grid points in the Y direction. Note that the self-consistent solution of Schrodinger's Equation with the Poisson's Equation doesn't allow solutions for the electron and hole continuity equations in the current ATLAS version. Non-self-consistent solutions, however, can be obtained by setting the POST.SCHRO parameter in the MODELS statement. These non-self-consistent solutions are obtained by solving Schrodinger's Equation only after convergence is obtained. That way, you can obtain Schrodinger solutions with the electron and hole continuity equations.

Similar results are saved to the structure file and the meaning of the EIGENS parameter in the OUTPUT statement is the same. In obtaining self-consistent solutions for the Schrodinger's Equation, an assumption is made about the location of the electron or hole quasi-fermi level. You can set two flags in the MODELS statement to vary the interpretation of these results. These flags are the FIXED.FERMI and CALC.FERMI parameters, which have interpretations in Table 13-2.

Table 13-2. Interpretations of optional parameters during post-processed Schrodinger solution

FIXED.FERMI	CALC.FERMI	Quasi-Fermi Level Calculation Method
FALSE	FALSE	Quasi-Fermi level is calculated from the local electron density using Equation 3-9.
FALSE	TRUE	Quasi-Fermi level varies with Y position and is calculated to match the local classical and quantum-mechanical charge concentration.
TRUE	FALSE	Quasi-Fermi level is uniformly zero.
TRUE	TRUE	Quasi-Fermi level is uniform across Y slice and is calculated to match the classical and quantum-mechanical sheet charge.

If you want a carrier concentration under non-equilibrium, specify CARRIERS=2 or CARRIERS=1 HOLES or CARRIERS=1 ELECTRONS on the METHOD statement. In this case, a classical quasi Fermi level will be computed by solving drift-diffusion equations. It is then used to find quantum electron density self-consistently with Poisson equation. Flags CALC.FERMI and FIXED.FERMI will be ignored.

13.3: Density Gradient (Quantum Moments Model)

The quantum models apply to several different types of problems. These problems are HEMT channel confinement simulation, thin gate oxide MOS capacitors and transistors, and other problems such as small geometry MESFETs and heterojunction diodes.

The effects due to confinement of carriers associated with variations of local potential on the scale of the electron wave functions (i.e., quantum effects) can be modeled in ATLAS using a density gradient. This model is based on the moments of the Wigner Function Equations-of-Motion [194, 195, 185, 178], which consists of quantum correction to the carrier temperatures in the carrier current and energy flux equations (see Chapter 3: "Physics", Equations 3-20 to 3-23).

In the density gradient model, the expression for electron current given in Chapter 3: "Physics", Equation 3-11 is replaced by the expression given in Equation 13-5.

$$\vec{J}_n = qD_n \nabla n - qn\mu_n \nabla(\psi - \Lambda) - \mu_n n(kT_L \nabla(ln n_{ie})) \quad 13\text{-}5$$

Here, Λ is a quantum correction potential.

Equation 13-6 is the density gradient model for holes.

$$\vec{J}_p = -qD_p \nabla p - qp\mu_p \nabla(\psi - \Lambda) + \mu_p p(kT_L \nabla(ln n_{ie})) \quad 13\text{-}6$$

The quantum potential can be calculated using either of the two expressions given in Equations 13-7 and 13-8.

$$\Lambda = -\frac{\gamma\hbar^2}{12m}\left[\nabla^2 log n + \frac{1}{2}(\nabla log n)^2\right] \quad 13\text{-}7$$

$$\Lambda = -\frac{\gamma\hbar^2}{6m}\frac{\nabla^2\sqrt{n}}{\sqrt{n}} \quad 13\text{-}8$$

where γ is a fit factor, m is the carrier effective mass, and n represents electron or hole concentration as appropriate. To choose between these expressions, specify DGLOG for Equation 13-7 or DGROOT for Equation 13-8 in the MODELS statement. By default, DGLOG is used. The fit factors for electrons and holes can be specified independently using the DGN.GAMMA or the DGP.GAMMA parameters of the MODELS statement respectively.

These expressions and particularly their derivatives can be seen to be sensitive to low carrier concentrations. You can specify a minimum concentration to use in these calculations using the QMINCONC parameter in the MODELS statement. This parameter specifies the minimum concentration used in the calculations of Equations 13-7 and 13-8 in cm^{-3}.

You can activate the model for electrons and holes independently. To activate the quantum model with the Quantum Moments Equation for electrons, use the quantum switch, MODELS QUANTUM, in the MODELS statement. To activate the Quantum Moments Equation for holes, use the quantum switch, MODELS P.QUANTUM.

Specify T.QUANTUM in the OUTPUT statement to write the quantum temperatures in the standard structure file. Once written you can examine the quantum temperature distribution using TONYPLOT.

Since the distributions of carriers given by the density gradient can vary greatly from the distributions predicted by the standard drift-diffusion or energy-balance models, the standard initial guess strategies (e.g., INIT) aren't usually suitable for obtaining solutions for quantum moments. Until more

suitable initial guess strategies can be devised, we've included a damping factor to gradually apply to the density gradient.

This damping factor is specified by the QFACTOR parameter in the SOLVE statement. The QFACTOR is implemented as a pre-factor to the expression for the quantum correction potential, Λ, in Equations 13-7 and 13-8. As such, a value of QFACTOR of 0.0 implies that the density gradient is either turned off or not applied. A value of QFACTOR of 1.0 implies that the Quantum Moment Model is turned on and applied.

You can vary the value of QFACTOR between 0.0 and 1.0 to overcome the problems of initial guess. During this ramping of QFACTOR, PREVIOUS should be used as an initial guess. Also, while varying the QFACTOR, trapping is available if a given solution doesn't converge. In such cases, the QFACTOR is reduced and solutions proceed until convergence is obtained or the maximum number of trap steps is exceeded.

The following is a typical fragment from an input file that ramps the QFACTOR parameter from 0 to 1:

```
MODEL NUMCARR=1 QUANTUM FLDMOB CONMOB SRH PRINT

OUTPUT CON.BAND VAL.BAND BAND.PARAM T.QUANTUM

SOLVE INIT

SOLVE QFACTOR=0.0

SOLVE QFACTOR=0.0001

SOLVE QFACTOR=0.001

SOLVE QFACTOR=0.01

SOLVE QFACTOR=0.1

SOLVE QFACTOR=1.0

LOG OUTF=test.log

SOLVE VDRAIN=0.01
```

13.4: Bohm Quantum Potential (BQP)

This model was developed for SILVACO by the University of Pisa and has been implemented into ATLAS with the collaboration of the University of Pisa. This is an alternative to the Density Gradient method and can be applied to a similar range of problems. There are two advantages to using Bohm Quantum Potential (BQP) over the density gradient method. First, it has better convergence properties in many situations. Second, you can calibrate it against results from the Schrodinger-Poisson equation under conditions of negligible current flow.

The model introduces a position dependent Quantum Potential, Q, which is added to the Potential Energy of a given carrier type. This quantum potential is derived using the Bohm interpretation of quantum mechanics [73] and takes the following form

$$Q = \frac{-h^2}{2}\frac{\gamma \underline{\nabla}(M^{-1}\underline{\nabla}(n^{\alpha}))}{n^{\alpha}} \qquad 13\text{-}9$$

where γ and α are two adjustable parameters, M^{-1} is the inverse effective mass tensor and n is the electron (or hole) density. This result is similar to the expression for the quantum potential in the density gradient model with $\alpha = 0.5$, but there are some differences about how they are implemented. Q is added to the continuity equations the same way Λ is for the density gradient method as shown in Equations 13-5 and 13-6.

The Bohm Quantum Potential (BQP) method can also be used for the Energy balance and hydrodynamic models, where the semi-classical potential is modified by the quantum potential the same way as for the continuity equations.

The iterative scheme used to solve the non-linear BQP equation along with a set of semi-classical equations is as follows. After an initial semi-classical solution has been obtained, the BQP equation is solved on its own Gummel iteration to give Q at every node in the device. The semi-classical potential is modified by the value of Q at every node and the set of semi-classical equations is then solved to convergence as usual (using a Newton or Block iterative scheme). Then, the BQP equation is solved to convergence again and the process is repeated until self-consistency is achieved between the solution of the BQP equation and the set of semi-classical equations. The set of semi-classical equations solved can be any of the combinations usually permitted by ATLAS.

To use the BQP model for electrons (or holes), specify `BQP.N` (`BQP.P`) in the `MODELS` statement. You can also set the parameter values (α and γ) and the direction of the quantization (confinement). Tables 13-3 and 13-4 show the parameters to use for the `MODEL` statement.

Table 13-3. MODEL Statement Parameters for Electrons

Parameter	Type	Default	Units
BQP.N	Logical	false	
BQP.NGAMMA	Real	1.2	--
BQP.NALPHA	Real	0.5	--
BQP.QDIR	Integer	2	--

Table 13-4. MODEL Statement Parameters for Holes			
Parameter	**Type**	**Default**	**Units**
BQP.P	Logical	false	
BQP.PGAMMA	Real	1.0	--
BQP.PALPHA	Real	0.5	--
BQP.QDIR	Integer	2	--

BQP.N switches on the model for electrons. BQP.P switches it on for holes. The BQP.NGAMMA and BQP.PGAMMA allow you to set the γ parameter for electrons and holes respectively. BQP.NALPHA and BQP.PALPHA allow you to set the α parameter for electrons and holes respectively.

You can specify BQP.N and BQP.P separately. But if quantum effects for electrons and holes occur in the same device, then you can specify them together in the same MODELS statement.

The BQP.QDIR parameter specifies the principal quantization direction.

BQP.QDIR = 1 means the x-direction

BQP.QDIR = 2 means the y-direction

BQP.QDIR = 3 means the z-direction

For example, if a MOSFET channel is in the x-y plane, the direction of quantization (quantum confinement) is the z-direction you would set BQP.QDIR=3. For semiconductors with spherical bands, BQP.QDIR will have no effect.

13.4.1: Calibration against Schrodinger-Poisson Model

You can obtain close agreement between BQP and the results of Schrodinger-Poisson (S-P) calculations for any given class of device. ATLAS has a Schrodinger-Poisson model that can model spatial confinement in only one direction. Therefore, calibration is currently restricted to this case. To obtain comparisons with S-P results, we recommend to use either the new quasistatic capacitance-voltage profile feature or compare charge-voltage curves. This will ensure similar charge control properties between the two models.

The first part of the calibration is to choose a suitable biasing for the device. There should be negligible current flow and quantum confinement effects that manifest at the chosen biases. The second part of the calibration is to set the appropriate BQP parameters in the MATERIAL or MODELS statements, and to set CARRIERS=0 in the METHOD statement.

This will cause the BQP equation to be coupled with Poisson's equation using the charge density terms.

$$n = N_c exp\left(-\frac{(E_c + qQ)}{kT}\right) \quad \text{13-10}$$

$$p = N_v exp\left(-\frac{(qQ - E_v)}{kT}\right) \quad \text{13-11}$$

This gives the same results as solving the current continuity equations with the constraint of zero current density.

The third part of calibration is to choose the quantity to compare with S-P results.

For example, for a MOSFET holding the drain and source voltages at the same bias and ramping the gate bias will give us a bias dependent capacitance with negligible current flow. So for an NMOS, you may have the statement

```
SOLVE VGATE=0.0 NAME=GATE VSTEP=0.01 VFINAL=2.0 QSCV
```

to give us the quasi-static C-V curve as it is biased into inversion. It is best to use a fine voltage step with QSCV to give good resolution. The process can be repeated by setting the S-P model in the MODELS statement instead of BQP to obtain the same set of curves for the S-P model.

The BQP model is then rerun with different sets of parameters until an acceptable agreement with the curves produced by the S-P model is achieved.

13.4.2: Post Calibration runs

After obtaining the parameters for BQP, either the Drift-Diffusion or energy balance (hydrodynamic) equations can be solved as usual. For cases where Lattice Heating is important then LAT.TEMP can be enabled at the same time.

The iteration scheme uses a modified version of BLOCK. Set BLOCK in the METHOD statement (although NEWTON and GUMMEL are ignored, BLOCK is always used if the BQP model is set). If an Energy Balance model is chosen (HCTE.EL or HCTE.HO on the MODELS statement), then an alternative iteration scheme will become available by specifying BQP.ALTEB in the METHOD statement. This method is slower and is only available in ATLAS2D but may have better convergence properties.

By using BQP.NOFERMI, the BQP equation will only use its Boltzmann statistics form. Without this parameter, the statistics used are those specified for the other equations. With fermi statistics, the convergence can be poor for very high carrier densities, and this parameter can circumvent some convergence properties.

Re-calibrate the BQP parameters if you set BQP.NOFERMI.

Table 13-5. METHOD Statement Parameter

Parameter	Type	Default	Units
BQP.ALTEB	Logical	false	
BQP.NOFERMI	Logical	false	

To speed up convergence, specify the NOCURRENT parameter on the first SOLVE statement after SOLVE INIT. It should prevent the need to use the QFACTOR parameter as was necessary for the Density Gradient method. QSCV enables the quasistatic capacitance calculation and output.

Table 13-6. SOLVE Statement Parameters

Parameter	Type	Default	Units
NOCURRENT	Logical	false	
QSCV	Logical	false	
QFACTOR	Real	1.0	

Use the parameters in Table 13-7 to control solution convergence behavior.

Table 13-7. METHOD Statement Parameters		
Parameter	**Type**	**Default**
BQPX.TOL	Real	2.5e-7
BQPR.TOL	Real	1.0E-26 (2d)1.0E-18 (3d)
NBLOCKIT	Integer	15
ITLIMIT	Integer	25
GUMITS	Integer	100

The solution comprises of OUTER iterations (which end when self-consistency occurs) and within these are solutions of groups of equations or single equations. The BQP equation is implemented so that both LHS and RHS convergence are required. You cannot override this default behavior. Tests carried out have shown it gives the best performance. ITLIMIT controls how many iterations can occur for the BQP equation in SOLVE INIT. This is 10 * ITLIMIT. For CARRIERS=0 solutions, the maximum number of iterations for the individual equation solvers is GUMITS. The maximum number of OUTER iterations is ITLIMIT. NBLOCKIT controls the number of OUTER cycles in the Drift-Diffusion and Energy-Balance cases. The maximum number of Drift-Diffusion solution and Carrier-Energy solution iterations is ITLIMIT, and the maximum number of BQP solutions is GUMITS.

Note: The Bohm Quantum potential is stored in ATLAS as an energy and so the convergence criteria are actually of order KT less than those for scaled electrostatic potential. In ATLAS 3D, the BQP equation had an extra scaling factor applied and so the RHS norms are larger than in 2D. LHS norms, however, are the same.

The Bohm Quantum Potential is automatically calculated in insulators. If you specify the SEMICONDUCTOR parameter in the MATERIAL statement for the insulator, the drift-diffusion equations will be solved in the insulator and will include the BQP corrections. It is also recommended to set the densities of states of the insulator to be the same as the semiconductor to which they interface. For example for Silicon/SiO2 system, set

```
MATERIAL REGION=N NC300=2.8E19 NV300=1.04E19
```

where N is region number of SiO2.

You can also set the BQP parameters on a material by material basis, which may be necessary for heterojunction based devices. The BQP model is applied globally and cannot be set on a region by region basis.

Table 13-8 shows the material parameters that have a direct impact on the BQP equation.

Table 13-8. MATERIAL Statement		
Parameter	**Type**	**Default**
SEMICONDUCTOR	Logical	false
EG300	Real	
AFFINITY	Real	

NC300	Real	
NV300	Real	
BQP.NALPHA	Real	0.5
BQP.NGAMMA	Real	1.2
BQP.PALPHA	Real	0.5
BQP.PGAMMA	Real	1.0

There is one more pertinent parameter, BQP.NEUMANN, which can be set on the MODELS statement.

BQP.NEUMANN sets the normal gradient of classical+quantum potential explicitly to zero on non-contact boundaries. If cleared using MODELS ^BQP.NEUMANN, the Neumann conditions will then be applied implicitly. Clearing this flag may give a slight speed increase and slight differences in solution.

Table 13-9. OUTPUT Statement Parameters

Parameter	Type	Default
P.QUANTUM	Logical	False

Setting the P.Quantum parameter in the OUTPUT statement will cause the Bohm Quantum Potential to output to a structure file. The Electron Quantum Potential and Hole Quantum Potential will both output.

13.5: Quantum Correction Models

In deep submicron MOS devices, quantum effects in the channel can have significant effects on the device characteristics. These effects are directly due to the increased doping levels and thinner gate oxide. In the channel of such devices, the potential well formed during inversion where the effects of quantum confinement must be considered. The direct effect of such quantum confinement is that the peak of the carrier concentration is shifted away from the interface and the thinner gate oxide can cause a marked difference in gate capacitance. Consideration of the effects may be essential for accurate prediction of the device turn-on voltage.

To fully treat quantum effects, solving the Schrodinger's Equation, isn't always desirable or necessary.

13.5.1: Hansch's Model

The quantum mechanical correction given by Hansch [65] is suitable for accurate simulation of the effects of quantum mechanical confinement near the gate oxide interface in MOSFETs. To enable the model, specify the HANSCHQM parameter of the MODEL statement. This correction model is a modification of the density of states as a function of depth below the Si/SiO_2 interface, which is given by Equation 13-12.

$$N_C^* = N_C\left[1 - exp-\left(\frac{z}{\mathrm{LAMBDA}}\right)^2\right]. \qquad 13\text{-}12$$

Here, N_C is the standard density of states, z is the depth below the interface. LAMBDA is a user-definable parameter in the MODEL statement.

Table 13-10. User-definable parameter for Equation 13-12

Statement	Parameter	Default	Units
MODELS	LAMBDA	1.0×10^{-3}	µm

13.5.2: Van Dort's Model

In the Van Dort's Model [164], the effects of the quantum confinement are modeled by broadening the bandgap near the surface, which makes a function of a perpendicular electric field and distance from the surface. In Van Dort's Model, the change in bandgap is given by the expression in Equation 13-13.

$$\Delta Eg = (\mathrm{B.DORT})\ \beta\frac{13}{9}\left(\frac{\varepsilon_{si}}{4_q k_B T_L}\right)^{1/3}(E_\perp)^{2/3} g(y)\ . \qquad 13\text{-}13$$

Here, B.DORT is a user-definable parameter in the MODEL statement, β is equal to 6.1×10^{-8} eV cm, $E_\perp$ is the perpendicular electric field, and g(y) is a function to restrict the application of the model to the channel region, given by Equation 13-14.

$$g(y) = \frac{2e^{-(Y/(\mathrm{D.DORT}))^2}}{1 + e^{-2(Y/(\mathrm{D.DORT}))^2}}. \qquad 13\text{-}14$$

The Van Dort's Model for N channel devices is enabled by specifying N.DORT in the MODELS statement. For P channel devices, the Van Dort's Model is enabled by specifying P.DORT in the MODELS statement.

Table 13-11. User-definable parameters for Equation 13-14			
Statement	**Parameter**	**Default**	**Units**
MODELS	D.DORT	2.5×10^{-6}	cm
MODELS	B.DORT	1.0	cm

13.6: General Quantum Well Model

The general quantum well model is used to predict bound state energies in a region for subsequent use in modelling optoelectronic gain, radiative recombination and absorption. Figure 13-1 shows the simulation flow for such modelling.

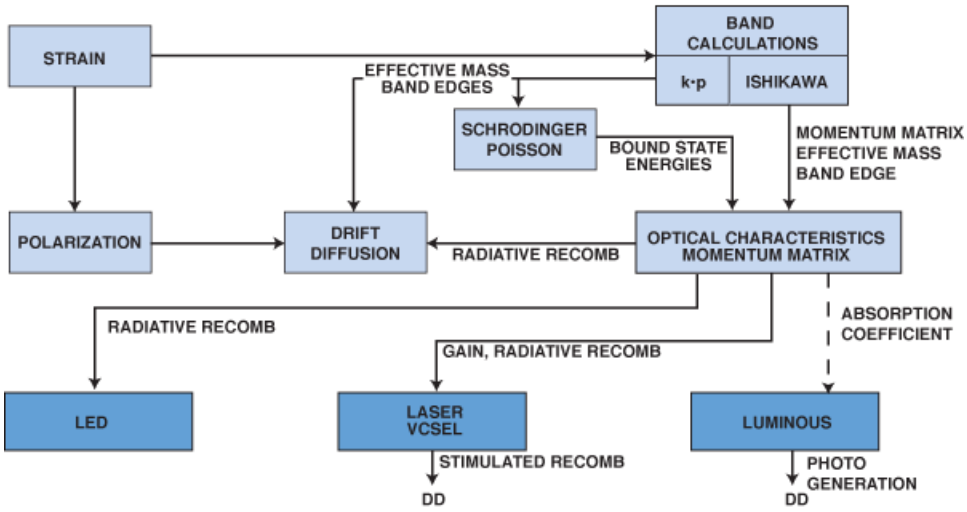


Figure 13-1: Simulation Flow For Physically Based Optoelectronic Models

The extraction of bound state energies is done by solving the Schrodinger equation (see Equation) along descrete slices in the Y direction using the interpolated values of the local potential and effective masses. You can enable the quantum well model by specifying QWELL in the REGION statement. You should also specify WELL.NX and WELL.NY in the REGION statement. The WELL.NX parameter specifies the integer number of slices along which Schrodinger solutions are taken. The WELL.NY parameter describes the number of samples along each slice. The sampling along both X and Y is uniform and bound state energy values are interpolated back onto the device mesh for the optical modelling. Generally, you should select WELL.NX and WELL.NY to adequately resolve the geometry but there is a tradeoff between computational accuracy and speed.

The WELL.CNBS and WELL.VNBS parameters of the REGION statement specify the maximum number of bound state energies to resolve in the conduction and valence band. When calculating the bound state energies, the actual number of bound states may be less than or equal to the value specified by these parameters. The actual number of bound states calculated also must be less than the number of samples specified by the WELL.NY parameter minus 2.

You can also ignore the effects of electric field by specifying the WELL.FIELD parameter of the REGION statement. This parameter enables the calculation of the dependence of bound state energy on bias conditions. By default, WELL.FIELD is true. If disabled by specifying WELL.FIELD, the bound states are only calculated once at zero bias and used for all subsequent solutions regardless of bias conditions.

You can also specify regions for quantum well simulation in the SUPERLATTICE (DBR) statement. To enable the model, specify the QWELL1 and QWELL2 parameters. These enable quantum modelling for the first and second superlattice cycles respectively. The sampling of the quantum wells are controlled by the WELL1.NX, WELL2.NX, WELL1.NY and WELL2.NY parameters. The numbers of bound states are controlled by the WELL1.CNBS, WELL2.CNBS, WELL1.VNBS and WELL2.VNBS parameters. The field effects are controlled by the WELL1.FIELD and WELL2.FIELD parameters.

In ATLAS3D, you can take a confinement in the z-direction into account. In this case, a 1D Schrodinger Equation is solved along the z-axis at each mesh node in the x-y plane.

13.7: Multiple Quantum Well Model

Optical gain and spontaneous recombination models implemented in ATLAS are used to account for the effects of quantum mechanical confinement of carriers and strain effects in Multiple Quantum Wells (MQW). To enable these models, use the MQW statement (see Chapter 19: "Statements", Section 19.30: "MQW").

The MQW statement has parameters describing the locations and compositions of the wells, the effects of strain on the band edges, parameters relating to the gain and recombination models, and parameters relating to how the models interact with other electrical and optical simulation models.

Note: The method for defining quantum wells using the MQW statement described in this section has been superceeded by the method described in Section 13.7: "Multiple Quantum Well Model". Therefore, we highly recommended that you use to the newer syntax.

13.7.1: Specifying MQW Location and Composition

In order to model the MQWs, specify the locations of the wells. The location of the wells is specified by the XMIN, XMAX, YMIN, and YMAX parameters. These parameters specify the location of a bounding box in units of microns (See Figure 13-2). XMIN and XMAX specify the extent of the wells in the X direction (i.e., the wells are horizontal and extend across the entire bounding box in the X direction). The YMIN and YMAX parameters define the center of the wells in the Y direction (i.e., the set of defined wells will be centered in the Y direction at the average coordinate of YMIN and YMAX).

The WW parameter specifies the width of each well in microns. The WB parameter specifies the width of the barrier between the wells in microns. The NWELL parameter specifies the number of wells.

Figure 13-2 gives illustrations of the how quantum wells are located for 1, 2, 3, and 4 quantum wells.

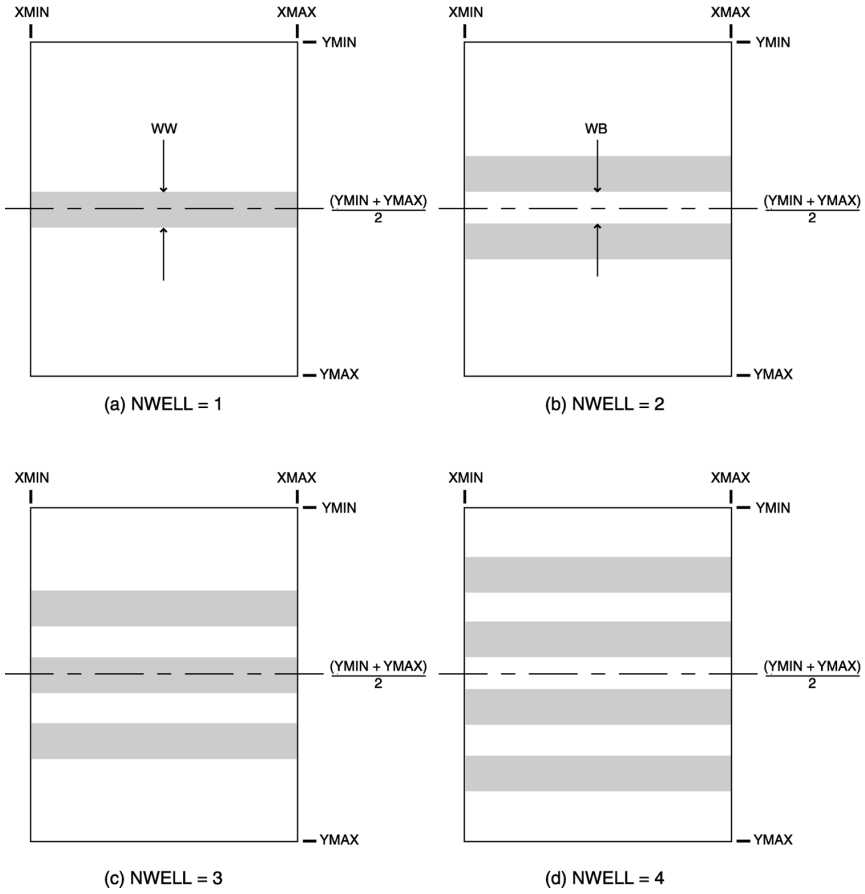


Figure 13-2: Examples of Specifying MQW Locations

The MATERIAL, XCOMP, YCOMP, DONORS, and ACCEPTORS parameters in the MQW statement define the composition of the wells. The MATERIAL parameter is assigned to the name of one of the allowed ATLAS materials discussed in Appendix B: "Material Systems", Section B.3: "ATLAS Materials". For Ternary and Quaternary materials, the XCOMP and YCOMP parameters can be used to specify the X and Y composition fractions. The DONORS and ACCEPTORS parameters specify uniform densities of ionized donors and acceptors in the wells.

The STRAIN, ASTR, BSTR, CSTR, DSTR, ESTR, FSTR, and STABLE parameters specify the stain effects. See Chapter 19: "Statements", Section 19.30: "MQW" for more information on these effects.

Chapter 14:
TFT: Thin-Film Transistor Simulator

14.1: Polycrystalline and Amorphous Semiconductor Models

TFT is an ATLAS module that simulates disordered material systems. TFT doesn't contain material models so you need to combine either S-PISCES or BLAZE with TFT to simulate these material systems.

TFT enables you to define an energy distribution of defect states in the bandgap of semiconductor materials. This is necessary for the accurate treatment of the electrical properties of such materials as polysilicon and amorphous silicon.

The syntax used by TFT is a part of the ATLAS syntax so there's no need for you to learn how to use a new simulator to run TFT simulations.

Before continuing with this chapter, you should be familiar with ATLAS physics. If not, read Chapters 2: "Getting Started with ATLAS" and 3: "Physics" before proceeding further. For more information on S-PISCES and BLAZE, see Chapters 4: "S-Pisces: Silicon Based 2D Simulator" and 5: "Blaze: Compound Material 2D Simulator".

14.2: Simulating TFT Devices

This section is intended to illustrate the basic building blocks for a thin-film transistor simulation.

To use TFT, specify the addition of defect states into the bandgap of a previously defined crystalline material. Throughout this section the example of polysilicon is used. Amorphous silicon and other disordered materials are handled in a similar manner.

14.2.1: Defining The Materials

Use the ATLAS command syntax to define a simple polysilicon thin-film transistor structure. The MESH, X.MESH, and Y.MESH statements can be used to construct a mesh as described in Chapter 2: "Getting Started with ATLAS", Section 2.6: "Defining A Structure". For more information on MESH statements, Chapter 19: "Statements," Sections 19.26: "MESH" and 19.56: "X.MESH, Y.MESH, Z.MESH".

When defining the material regions, use the following syntax.

```
REGION Y.MIN=-0.05 Y.MAX=0   OXIDE

REGION Y.MIN=0     Y.MAX=0.2 SILICON

REGION Y.MIN=0.2   Y.MAX=2   OXIDE
```

Note that the region is defined as silicon. You can also define the material as polysilicon. But note that it is the defect distribution rather than this initial material definition that will determine the electrical characteristics.

14.2.2: Defining The Defect States

Disordered materials contain a large number of defect states within the band gap of the material. To accurately model devices made of polycrystalline or amorphous materials, use a continuous density of states. The DEFECT statement is used to specify the density of defect states (DOS) as a combination of exponentially decaying band tail states and Gaussian distributions of mid-gap states [82, 63]. In addition, you may need to model the grain-grain boundary interface as a thermionic field emission boundary [84].

Note: The INTDEFECTS, INTERFACE S.S and TRAP.COULOMBIC models are avaiable for grain-grain boundary interface modeling. These models have been developed in collaboration with Epson and Cambridge University. For more information, see Chapter 19: "Statements,", Sections 19.17: "INTDEFECTS", 19.18: "INTERFACE", and 19.29: "MODELS".

14.2.3: Density of States Model

It is assumed that the total density of states (DOS) and $g(E)$, is composed of four bands: two tail bands (a donor-like valence band and an acceptor-like conduction band) and two deep level bands (one acceptor-like and the other donor-like) which are modeled using a Gaussian distribution.

$$g(E) = g_{TA}(E) + g_{TD}(E) + g_{GA}(E) + g_{GD}(E) \quad \text{14-1}$$

Here, E is the trap energy, E_C is the conduction band energy, E_V is the valence band energy and the subscripts (T, G,A, D) stand for tail, Gaussian (deep level), acceptor and donor states respectively.

$$g_{TA}(E) = \text{NTA}\ exp\left[\frac{E - E_c}{\text{WTA}}\right] \quad \text{14-2}$$

$$g_{TD}(E) = \texttt{NTD}\ exp\left[\frac{E_v - E}{\texttt{WTD}}\right] \quad 14\text{-}3$$

$$g_{GA}(E) = \texttt{NGA}\ exp\left[-\left[\frac{\texttt{EGA} - E}{\texttt{WGA}}\right]^2\right] \quad 14\text{-}4$$

$$g_{GD}(E) = \texttt{NGD}\ exp\left[-\left[\frac{E - \texttt{EGD}}{\texttt{WGD}}\right]^2\right] \quad 14\text{-}5$$

For an exponential tail distribution, the DOS is described by its conduction and valence band edge intercept densities (NTA and NTD), and by its characteristic decay energy (WTA and WTD).

For Gaussian distributions, the DOS is described by its total density of states (NGA and NGD), its characteristic decay energy (WGA and WGD), and its peak energy/peak distribution (EGA and EGD). Table 14-1 shows the user-specifiable parameters for the density of defect states.

Table 14-1. User-Specifiable Parameters for Equations 14-2 to 14-5

Statement	Parameter	Default	Units
DEFECT	NTA	1.12×10^{21}	cm^{-3}
DEFECT	NTD	4.0×10^{20}	cm^{-3}
DEFECT	NGA	5.0×10^{17}	cm^{-3}
DEFECT	NGD	1.5×10^{18}	cm^{-3}
DEFECT	EGA	0.4	eV
DEFECT	EGD	0.4	eV
DEFECT	WTA	0.025	eV
DEFECT	WTD	0.05	eV
DEFECT	WGA	0.1	eV
DEFECT	WGD	0.1	eV

14.2.4: Trapped Carrier Density

The ionized densities of acceptor and donor like states (n_T and p_T respectively) are given by:

$$p_T = p_{TA} + p_{GA} \quad 14\text{-}6$$

$$n_T = n_{TD} + n_{GD} \quad 14\text{-}7$$

where n_{TA}, n_{GA}, p_{TD} and p_{GD} are given below.

$$p_{TA} = \int_{E_V}^{E_C} g_{TA}(E) \cdot f_{t_{TA}}(E, n, p)\, dE \qquad 14\text{-}8$$

$$p_{GA} = \int_{E_V}^{E_C} g_{GA}(E) \cdot f_{t_{GA}}(E, n, p)\, dE \qquad 14\text{-}9$$

$$n_{TD} = \int_{E_V}^{E_C} g_{TD}(E) \cdot f_{t_{TD}}(E, n, p)\, dE\ . \qquad 14\text{-}10$$

$$n_{GD} = \int_{E_V}^{E_C} g_{GD}(E) \cdot f_{t_{GD}}(E, n, p)\, dE \qquad 14\text{-}11$$

$f_{t_{TA}}(E, n, p)$ and $f_{t_{GA}}(E, n, p)$ are the ionization probabilities for the tail and Gaussian acceptor DOS, while $f_{t_{TD}}(E, n, p)$ and $f_{t_{GD}}(E, n, p)$ are the ionization probabilities for the donors.

In the steady-state case, the probability of occupation of a trap level at energy E for the tail and Gaussian acceptor and donor states are given by Equations 14-12 through 14-15.

$$f_{t_{TA}}(E, n, p) = \frac{v_n\ \mathtt{SIGTAE}\ n + v_p\ \mathtt{SIGTAH}\ n_i\ exp\left[\frac{E_i - E}{kT}\right]}{v_n\ \mathtt{SIGTAE}\left(n + n_i\ exp\left[\frac{E - E_i}{kT}\right]\right) + v_p\ \mathtt{SIGTAH}\ \left(p + n_i\ exp\left[\frac{E_i - E}{kT}\right]\right)} \qquad 14\text{-}12$$

$$f_{t_{GA}}(E, n, p) = \frac{v_n\ \mathtt{SIGGAE}\ n + v_p\ \mathtt{SIGGAH}\ n_i\ exp\left[\frac{E_i - E}{kT}\right]}{v_n\ \mathtt{SIGGAE}\left(n + n_i\ exp\left[\frac{E - E_i}{kT}\right]\right) + v_p\ \mathtt{SIGGAH}\left(p + n_i\ exp\left[\frac{E_i - E}{kT}\right]\right)} \qquad 14\text{-}13$$

$$f_{t_{TD}}(E, n, p) = \frac{v_p\ \mathtt{SIGTDH}\ p + v_n\ \mathtt{SIGTDE}\ n_i\ exp\left[\frac{E - E_\zeta}{kT}\right]}{v_n\ \mathtt{SIGTDE}\left(n + n_i\ exp\left[\frac{E - E_i}{kT}\right]\right) + v_p\ \mathtt{SIGTDH}\left(p + n_i\ exp\left[\frac{E_i - E}{kT}\right]\right)} \qquad 14\text{-}14$$

$$f_{t_{GD}}(E,n,p) = \frac{v_p \ \texttt{SIGGDH}\ p + v_n \ \texttt{SIGGDE}\ n_i \ exp\left[\frac{E-E_i}{kT}\right]}{v_n \ \texttt{SIGGDE}\left(n + n_i \ exp\left[\frac{E-E_i}{kT}\right]\right) + v_p \ \texttt{SIGGDH}\left(p + n_i \ exp\left[\frac{E_i-E}{kT}\right]\right)} \quad \text{14-15}$$

where v_n is the electron thermal velocity and v_p is the hole thermal velocity, n_i is the intrinsic carrier concentration. SIGTAE and SIGGAE are the electron capture cross-section for the acceptor tail and Gaussian states respectively. SIGTAH and SIGGAH are the hole capture cross-sections for the acceptor tail and Gaussian states respectively and SIGTDE, SIGGDE, SIGGDH, and SIGGDH are the equivalents for donors states.

Table 14-2. User-Specifiable Parameters for Equations 14-12 to 14-15

Statement	Parameter	Default	Units
DEFECT	SIGTAE	1.0×10^{-16}	cm^2
DEFECT	SIGTDE	1.0×10^{-14}	cm^2
DEFECT	SIGGAE	1.0×10^{-16}	cm^2
DEFECT	SIGGDE	1.0×10^{-14}	cm^2
DEFECT	SIGTAH	1.0×10^{-14}	cm^2
DEFECT	SIGTDH	1.0×10^{-16}	cm^2
DEFECT	SIGGAH	1.0×10^{-14}	cm^2
DEFECT	SIGGDH	1.0×10^{-16}	cm^2

14.2.5: Steady-State Trap Recombination

For steady-state conditions, the net recombination/generation rate is identical for electrons (R_n) and holes (Rp) (i.e., instantaneous equilibrium). Using Equations 14-12 through 14-15 to give the values of f_t and following the derivation by Shockley and Read [149] and Hall [64], the Shockley-Read-Hall recombination/generation rate due to the defect states is given by:

$$R_{n,p} = \int_{E_V}^{E_C} \left(\frac{v_n\, v_p\, \mathtt{SIGTAE}\, \mathtt{SIGTAH} \left(n\,p - n_i^2\right) g_{TA}(E)}{v_n\, \mathtt{SIGTAE} \left(n + n_i\, exp\left[\frac{E - E_i}{k\,T}\right]\right) + v_p\, \mathtt{SIGTAH} \left(p + n_i\, exp\left[\frac{E_i - E}{k\,T}\right]\right)} \right. \quad \text{14-16}$$

$$+ \frac{v_n\, v_p\, \mathtt{SIGTGAE}\, \mathtt{SIGGAH} \left(n\,p - n_i^2\right) g_{GA}(E)}{v_n\, \mathtt{SIGGAE} \left(n + n_i\, exp\left[\frac{E - E_i}{k\,T}\right]\right) + v_p\, \mathtt{SIGGAH} \left(p + n_i\, exp\left[\frac{E_i - E}{k\,T}\right]\right)}$$

$$+ \frac{v_n\, v_p\, \mathtt{SIGTDE}\, \mathtt{SIGTDH} \left(n\,p - n_i^2\right) g_{TD}(E)}{v_n\, \mathtt{SIGTDE} \left(n + n_i\, exp\left[\frac{E - E_i}{k\,T}\right]\right) + v_p\, \mathtt{SIGTDH} \left(p + n_i\, exp\left[\frac{E_i - E}{k\,T}\right]\right)}$$

$$\left. + \frac{v_n\, v_p\, \mathtt{SIGGDE}\, \mathtt{SIGGDH} \left(n\,p - n_i^2\right) g_{GD}(E)}{v_n\, \mathtt{SIGGDE} \left(n + n_i\, exp\left[\frac{E - E_i}{k\,T}\right]\right) + v_p\, \mathtt{SIGGDH} \left(p + n_i\, exp\left[\frac{E_i - E}{k\,T}\right]\right)} \right) dE$$

14.2.6: Transient Traps

For the transient case, time is required for carriers to be emitted or captured and therefore instantaneous equilibrium cannot be assumed. This means that Equation 14-16 is no longer valid for transient simulations. Instead, the total recombination/generation rate for electrons (which is equal to electron recombination rate minus the generation rate for electrons) is calculated using the transient probabilities of occupation for acceptors (f_{tTA} and f_{tGA}). These are calculated by solving additional rate equations (Equations 14-17 and 14-18).

$$\frac{d}{dt}(p_{TA}) = \int_{E_V}^{E_C} g_{TA}(E) \left[v_n\ \mathtt{SIGTAE} \left(n \left(1 - f_{t_{TA}}(E)\right) - f_{t_{TA}}(E)\, n_i\, exp\left[\frac{E - E_i}{k\,T}\right]\right)\right. \quad \text{14-17}$$

$$\left. - v_p\ \mathtt{SIGTAH} \left(p\, f_{t_{TA}}(E) - \left(1 - f_{t_{TA}}(E)\right)\, n_i\, exp\left[\frac{E_i - E}{k\,T}\right]\right) \right] dE$$

$$\frac{d}{dt}(p_{GA}) = \int_{E_V}^{E_C} g_{GA}(E)\left[v_n \ \texttt{SIGGAE}\left(n\left(1-f_{t_{GA}}(E)\right)-f_{t_{GA}}(E)\ n_i\ exp\left[\frac{E-E_i}{k\ T}\right]\right)\right. \quad \text{14-18}$$

$$\left. - v_p \ \texttt{SIGGAH}\left(p\, f_{t_{GA}}(E) - \left(1-f_{t_{GA}}(E)\right)\ n_i\ exp\left[\frac{E_i-E}{k\ T}\right]\right)\right] dE$$

The total hole recombination/generation rate can also be determined from the transient values of f_{tTD} and f_{tGD} (see Equations 14-19 and 14-20).

$$\frac{d}{dt}(n_{TD}) = \int_{E_V}^{E_C} g_{TD}(E)\left[v_n \ \texttt{SIGTDH}\left(p\left(1-f_{t_{TD}}(E)\right)-f_{t_{TD}}(E)\ n_i\ exp\left[\frac{E_i-E}{k\ T}\right]\right)\right. \quad \text{14-19}$$

$$\left. - v_n \ \texttt{SIGTDE}\left(n\, f_{t_{TD}}(E) - \left(1-f_{t_{TD}}(E)\right)\ n_i\ exp\left[\frac{E-E_i}{k\ T}\right]\right)\right] dE$$

$$\frac{d}{dt}(n_{GD}) = \int_{E_V}^{E_C} g_{GD}(E)\left[v_p \ \texttt{SIGGDH}\left(p\left(1-f_{t_{GD}}(E)\right)-f_{t_{GD}}(E)\ n_i\ exp\left[\frac{E_i-E}{k\ T}\right]\right)\right. \quad \text{14-20}$$

$$\left. - v_n \ \texttt{SIGGDE}\left(n\, f_{t_{GD}}(E) - \left(1-f_{t_{GD}}(E)\right)\ n_i\ exp\left[\frac{E-E_i}{k\ T}\right]\right)\right] dE$$

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. Specifying FAST on the DEFECTS statement will neglect the trap rate equation from the simulation.

Trap-Assisted Tunneling

Trap-Assisted Tunneling models the trap-to-band phonon-assisted tunneling effects for Dirac wells. At high electric fields, tunneling of electrons from the valence band to the conduction band through trap or defect states can have an important effect on the current.

Trap-assisted tunneling is modeled by including field-effect enhancement terms [70] (Γ_n^{DIRAC} and Γ_p^{DIRAC}) in the trap lifetimes in capture cross-sections. This model is enabled by specifying TRAP.TUNNEL in the MODELS statement.

The electron capture cross-section (SIGTAE, SIGGAE, and SIGTDE, and SIGGDE) are modified by including the electron field-effect term (Γ_n^{DIRAC}). For example, the electron capture cross-section for acceptor tail states (SIGTAE) becomes:

The field-effect enhancement term for electrons is given by:

$$\texttt{SIGTAE} \times \left(1 + \Gamma_n^{DIRAC} \right) \tag{14-21}$$

SIGGAE, SIGTDE, and SIGGDE are also modified this way.

$$I_n^{DIRAC} = \frac{\Delta En}{kT_L} \int_0^1 exp\left(\frac{\Delta En}{KT_L} u - k_n u^{3/2} \right) du \tag{14-22}$$

While the field-effect enhancement term for hole is:

$$I_p^{DIRAC} = \frac{\Delta Ep}{kT_L} \int_0^1 exp\left(\frac{\Delta En}{KT_L} u - k_p u^{3/2} \right) du \tag{14-23}$$

The hole capture cross-sections (SIGTAH, SIGGAH, SIGTDH, and SIGGDH) are modified by including the hole field effect term (Γ_n^{DIRAC}). For example, SIGTAH now becomes:

$$\texttt{SIGTAH} \times 1 + \Gamma_n^{DIRAC} \tag{14-24}$$

SIGGAH, SIGTDH, and SIGGDH are also modified this way.

where u is the integration variable, ΔE_n is the energy range where tunneling can occur for electrons, ΔE_p is the tunneling energy range for holes, and k_n and k_p are given by:

$$k_n = \frac{4}{3} \frac{\sqrt{2 m_0 \texttt{MASS.TUNNEL} \Delta En^3}}{3q\hbar|E|} \tag{14-25}$$

$$k_p = \frac{4}{3} \frac{\sqrt{2 m_0 \texttt{MASS.TUNNEL} \Delta Ep^3}}{3q\hbar|E|} \tag{14-26}$$

$\hbar$ is the reduced Planck's constant, m_0 is the rest mass of an electron, and MASS.TUNNEL is the relative effective mass (You can specify MASS.TUNNEL by setting the MASS.TUNNEL parameter in the MODELS statement).

Poole-Frenkel Barrier Lowering

The Poole Frenkel barrier lowering effect enhances the emission rate for trap-to-band phonon-assisted tunneling and pure thermal emissions at low electric fields. The Poole Frenkel effect occurs when the Coulombic potential barrier is lowered due to the electric field and only occurs in traps with a Coulombic potential (such as traps that are neutral when filled).

The Poole-Frenkel effect is modeled by including field-effect enhancement terms for Coulombic wells (Γ_n^{COUL} and Γ_p^{COUL}) and thermal emission (χ_F) [100] in the capture cross-sections. The model also includes the trap-assisted tunneling effects in the Dirac well. To enable this model, specify TRAP.COULOMBIC in the MODELS statement.

In the electron recombination/generation term (R_N), the Coulombic barrier lowering term (cF), and the electron Coulombic field-effect enhancement term (Γ_n^{COUL}) are applied to the electron capture cross-sections.

For example, SIGTAE now becomes:

$$\texttt{SIGTAE} \times \left(\chi_F + \Gamma_n^{COUL} \right) \quad \text{14-27}$$

SIGGAE, SIGTDE, and SIGGDE are modified in the same manner.

The hole capture cross-section are modified by including the Dirac field-effect enhancement term for holes (Γ_n^{DIRAC}).

For example, SIGTAH now becomes:

$$\texttt{SIGTAE} \times \left(\chi_F + \Gamma_n^{COUL} \right) \quad \text{14-28}$$

SIGGAH, SIGTDH, and SIGGDH are modified in the same manner.

In the hole recombination/generation term (R_p), the Coulombic terms are applied to the hole capture cross-sections and the Dirac terms applied to the electron capture cross-sections.

The Poole-Frenkel thermal emission enhancement factor, χ_F, is defined as:

$$\chi_F = exp\left(\frac{\Delta E_{fp}}{kT_L}\right) \quad \text{14-29}$$

ΔE_{fp} is the barrier lowering term for a Coulombic well (see Equation 14-30).

$$\Delta E_{fp} = \sqrt{\frac{q|E|}{\pi\varepsilon}} \quad \text{14-30}$$

The Coulombic field-enhancement terms, Γ_n^{COUL} and Γ_p^{COUL}, are defined as:

$$\Gamma_n^{COUL} = \frac{\Delta En}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} exp\left(\frac{\Delta En}{KT_L}u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_n}\right)^{5/3}\right]\right)du \quad \text{14-31}$$

$$\Gamma_p^{COUL} = \frac{\Delta E_p}{kT_L} \int_{\frac{\Delta E_{fp}}{\Delta E_n}}^{1} exp\left(\frac{\Delta E_p}{KT_L}u - k_p u^{3/2}\left[1 - \left(\frac{\Delta E_{fp}}{u\Delta E_p}\right)^{5/3}\right]\right)du \quad \text{14-32}$$

14.2.7: Continuous Defects

If CONTINUOUS is specified in the DEFECTS statement, then the integral equations for the charge and recombination are evaluated using a numerical integral scheme [144]. In this case, the NUMA and NUMD (DEFECTS statement) parameters correspond to the number of acceptor and donor energy level intervals used in the integral.

14.2.8: Discrete Defects

If CONTINUOUS is not specified on the DEFECTS statement, the equation is modeled with discrete energy levels. The integrals terms in Equations 14-8 to 14-11 are replaced by summations, which run over the number of discrete energy levels (NUMA and NUMD). The acceptor and donor density of states terms are integrated separately. For example, the equation for the electron trap concentration (Equation 14-6) is replaced by:

$$n_T = \sum_{i=0}^{\mathrm{NUMA}} \left(f_{t_{TA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{TA}(E)\, dE + f_{t_{GA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{GA}(E)\, dE \right) \quad \text{14-33}$$

Table 14-3. Additional Parameters for the DEFECTS Statement

Statement	Parameter	Default	Units
DEFECT	FAST	FALSE	
DEFECT	CONTINUOUS	FALSE	
DEFECT	NUMA	12	
DEFECT	NUMD	12	

Syntax for a typical defect states definition is given below.

```
DEFECTS NTA=1.12E21 NTD=4.E20 WTA=0.025 WTD=0.05 \
     NGA=5.E17 NGD=1.5E18 EGA=0.4 EGD=0.4 \
     GA=0.1 WGD=0.1 SIGTAE=1.E-16 \
     SIGTAH=1.E-14 SIGTDE=1.E-14 \
     SIGTDH=1.E-16 SIGGAE=1.E-16 SIGGAH=1.E-14 \
     SIGGDE=1.E-14 SIGGDH=1.E-16
```

Figure 14-1 shows how the syntax is used to define the peak density of states and distribution widths for the two tail states and two Gaussian distributions.

14.2.9: Plotting The Density Of States Versus Energy

The DFILE and AFILE parameters were used to allow you to specify output file names for capture of defect densities as a function of energy for donor states and acceptor states, respectively. For example, if you want to look at the donor and acceptor defect distributions, specify the following line:

```
DEFECTS DFILE=donors AFILE=acceptors
```

The files, *donors* and *acceptors*, could then be loaded into TONYPLOT to look at the distributions of donor and acceptor defects as a function of energy.

14.2.10: Using the C-Interpreter to define DEFECTS

The C-INTERPRETER can be used to define the defect states in the bandgap. The F.TFTDON and F.TFTACC parameters of the DEFECT statement indicate the filenames containing the C functions. For more information on using the C-INTERPRETER, See Appendix A: "C-Interpreter Functions".

```
DEFECTS F.TFTDON=mydefects.c F.TFTACC=mydefects.c
```

The file, *mydefects.c*, will contain C functions for donor and acceptor defect densities as a function of energy. These user defined defects are added to the existing defect distribution. If you want to use only your own function, set the gaussian and tail functions to zero. The following example defect states are defined in the file, *tft.lib*. These are added to a zero background set using the tail and gaussian state syntax. The resultant distribution of defects versus energy can be plotted in the files, *don.dat* and *acc.dat*.

```
DEFECTS F.TFTDON=tft.lib F.TFTACC=tft.lib DFILE=don.dat AFILE=acc.dat \
     NTA=0 NTD=0 WTA=1.0 WTD=1.0 \
     NGA=0 NGD=0 EGA=0.6 EGD=0.6 WGA=1 WGD=1 \
     SIGTAE=1.E-16 SIGTAH=1.E-14 SIGTDE=1.E-14 SIGTDH=1.E-16
     SIGGAE=1.E-16 SIGGAH=1.E-14 SIGGDE=1.E-14 SIGGDH=1.E-16
```

14.2.11: Setting Mobility and Other Models

TFT uses adaptations of the standard models of S-PISCES or BLAZE. The example below shows how to select the models and material parameters for polysilicon.

```
MATERIAL MUN=300 MUP=30

MODELS SRH
```

Typical mobility values for amorphous silicon can be set by:

```
MATERIAL MUN=20 MUP=1.5
```

Other models are also available in TFT. These include impact ionization and tunneling. Set them by using:

```
MODELS BBT.STD IMPACT
```

Note: Don't use concentration dependent mobility models (CONMOB, ANALYTIC, ARORA, KLA) because they overwrite the low field mobilities set in the MATERIAL statement.

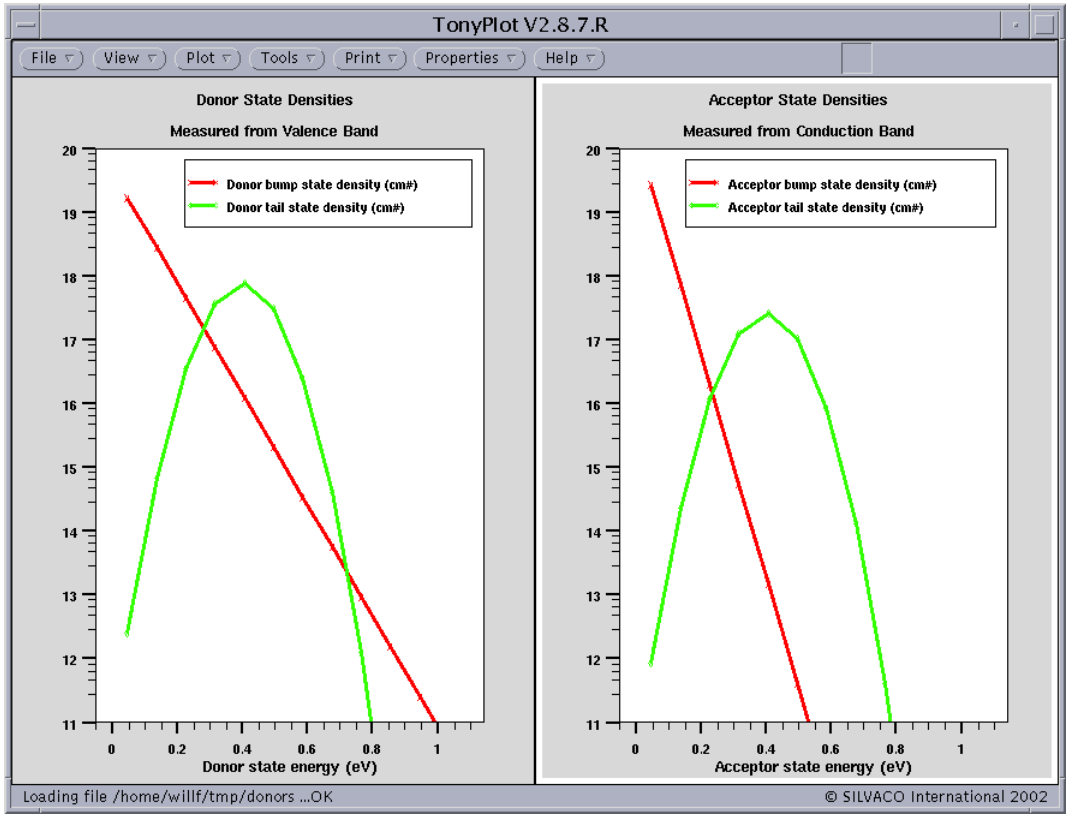


Figure 14-1: Syntax Guide to Define Two Tail States and Two Gaussian Distributions. NGA and NDG are the integrated values of the Gaussian distributions. Gaussians are entered on energies EGA and EGD respectively.

Chapter 15:
OTFT/OLED: Organic and Polymer Materials Simulators

15.1: Overview

OTFT allows ATLAS to simulate the characteristics of organic materials based devices. OTFT works in conjunction with the BLAZE module to simulate the DC and transient electrical and optical characteristics of these devices. OLED allows ATLAS to simulate the singlet and triplet excition densities. These modules can also be used with the MIXEDMODE simulator to analyze circuit level behavior.

You should be familiar with ATLAS physics before performing an organic material simulation. If not, read Chapter 2: “Getting Started with ATLAS” and 5: “Blaze: Compound Material 2D Simulator” in the ATLAS USER’S MANUAL.

OTFT and OLED supports the simulation of the following organic materials: Tetracene, Pentacene, PPV, TPD and Alq3. User-defined materials can also be used with this module (see the REGION and MATERIAL definitions in Chapter 18: "Statements").

The OTFT module allows you to do the following:

- Define an energy distribution of defect states based on a characteristic temperature and density.
- Calculate the Poole-Frenkel mobility.
- Calculate the bimolecular Langevin recombination rate.

The OLED module allows you to calculate the density of singlet and triplet excitons.

15.1.1: Introduction

The numerical framework for simulation of organic materials is the drift-diffusion model. Organic materials have electronic traps due to their weak molecular bonds and structural disorder. Traps introduce energy levels inside the energy gap between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). In the following sections, we describe the models that are particular to the simulation of polymer and organic devices.

15.2: Organic Materials Physical Models

This section describes the model definition required for the simulation of organic devices.

15.2.1: Organic Defects Model

Density of States

The ODEFECTS statement is used to specify the energy distribution of defect states. The total density of states (DOS) is composed of an acceptor-like conduction band DOS ($g_A(E)$) and a donor-like valence band DOS ($g_D(E)$). There are two defect distribution models available. The first model defines the defects based on a characteristic temperature and density [145].

$$g_A(E) = \frac{\mathrm{HA}}{k\mathrm{TCA}} exp\left(\frac{E-E_c}{k\mathrm{TCA}}\right) \quad 15\text{-}1$$

$$g_D(E) = \frac{\mathrm{HD}}{k\mathrm{TCD}} exp\left(\frac{E_v-E}{k\mathrm{TCD}}\right) \quad 15\text{-}2$$

HA is the acceptor traps density of states, HD is the donor traps density of states, TCA is the acceptor traps characteristic temperature, TCD is the donor traps characteristic temperature, and E is the energy.

Table 15-1. User-Specifiable Paramters for Equations 15-1 and 15-2

Statement	Parameter	Default	Units
ODEFECTS	HA	0	cm^{-3}
ODEFECTS	HD	0	cm^{-3}
ODEFECTS	TCA	0	K
ODEFECTS	TCD	0	K

The trapped state concentrations for electrons (n_A) and holes (p_D) are given by:

$$n_A = \int_{Ev}^{Ec} g_A(E) \cdot f_{tA}(E, n, p)dE \quad 15\text{-}3$$

$$p_D = \int_{Ev}^{Ec} g_D(E) \cdot f_{tD}(E, n, p)dE \quad 15\text{-}4$$

where $f_{tA}(E,n,p)$ is the acceptor probability of occupation and $f_{tD}(E,n,p)$ is the donor probability of occupation.

The second model is the effective transport hopping energy model and defines the defects using a double peak Gaussian distribution [13],[14].

$$g_A(E) = \frac{\texttt{NIA}}{\sqrt{2\pi}\ \texttt{SIGMAIA}}\ exp\left(\frac{(E-E_c)^2}{2\texttt{SIGMAIA}^2}\right) + \frac{\texttt{NA}}{\sqrt{2\pi}\ \texttt{SIGMAA}}\ exp\left(\frac{(E-E_c+\texttt{EA})^2}{2\texttt{SIGMAA}^2}\right) \quad 15\text{-}5$$

$$g_D(E) = \frac{\texttt{NID}}{\sqrt{2\pi}\ \texttt{SIGMAID}}\ exp\left(\frac{(E_v-E)^2}{2\texttt{SIGMAID}^2}\right) + \frac{\texttt{ND}}{\sqrt{2\pi}\ \texttt{SIGMAD}}\ exp\left(\frac{(E_v-E+\texttt{ED})^2}{2\texttt{SIGMAD}^2}\right) \quad 15\text{-}6$$

NIA is the total intrinsic density for acceptor-like traps. NID is the total intrinsic density for donor-like traps. SIGMAIA is the Gaussian width for the intrinsic acceptor-like traps. SIGMAID is the Gaussian width for the intrinsic donor-like traps. NA is the total doping density for acceptor-like traps. ND is the total doping density for donor-like traps. SIGMAA is the Gaussian width for the doping acceptor-like traps. SIGMAD specifies the Gaussian width for the doping donor-like traps. EA is the energy shift between the intrinsic and doping states for acceptor-like traps. ED is the energy shift between the intrinsic and doping states for donor-like traps.

The trapped state concentrations are given by:

$$n_A = \int_{E_v}^{E_{tr_n}} gA(E)F_{tA}(E,n,p)dE \quad 15\text{-}7$$

$$p_D = \int_{E_{tr_p}}^{E_c} gD(E)F_{tD}(E,n,p)dE \quad 15\text{-}8$$

where E_{tr_n} is the effective transport energy for electrons and E_{tr_p} is the effective transport energy for holes. You must specify the HOPPING parameter on the ODEFECTS statement to use the effective transport hopping energy model. The effective transport energy for electrons and holes is calculate by solving Equations 15-5 and 15-6 along with Equations 15-7 and 15-8.

$$\int_{-\infty}^{E_{tr_n}} gA(E)\left(E_{tr_n}-E\right)^3 dE = \frac{6\texttt{HOPN.BETA}}{\pi}(\texttt{HOPN.GAMMA}kT)^3 \quad 15\text{-}9$$

$$\int_{-\infty}^{E_{tr_p}} gD(E)\left(E_{tr_p}-E\right)^3 dE = \frac{6\texttt{HOPP.BETA}}{\pi}(\texttt{HOPP.GAMMA}kT)^3 \quad 15\text{-}10$$

where HOPN.BETA is the electron percolation constant, HOPP.BETA is the hole percolation constant, HOPN.GAMMA is 1/carrier localization radius for electrons and HOPP.GAMMA is 1/carrier localization radius for holes.

Table 15-2. User-Specifiable Parameters for Equations 15-5-15-10

Statement	Parameter	Value	Units
ODEFECTS	EA	0.0	eV
ODEFECTS	ED	0.0	eV

Table 15-2. User-Specifiable Parameters for Equations 15-5-15-10			
MATERIAL	HOPN.BETA	1.5	
MATERIAL	HOPN.GAMMA	5.0e7	cm^{-1}
MATERIAL	HOPP.BETA	1.5	
MATERIAL	HOPP.GAMMA	5.0e7	cm^{-1}
ODEFECTS	HOPPING	False	
ODEFECTS	NA	0.0	cm^{-3}
ODEFECTS	ND	0.0	cm^{-3}
ODEFECTS	NIA	0.0	cm^{-3}
ODEFECTS	NID	0.0	cm^{-3}
ODEFECTS	SIGMAA	0.0	eV
ODEFECTS	SIGMAD	0.0	eV
ODEFECTS	SIGMAIA	0.0	eV
ODEFECTS	SIGMAID	0.0	eV

In steady-state, the probabilities of occupation are given by:

$$f_{tA} = \frac{v_n \mathrm{SIGAE} n + v_p \mathrm{SIGAH} n_i exp\left[\frac{E_i - E}{kT}\right]}{v_n \mathrm{SIGAE}\left(n + n_i exp\left[\frac{E - E_i}{kT}\right]\right) + v_p \mathrm{SIGAH}\left(p + n_i exp\left[\frac{E_i - E}{kT}\right]\right)} \quad 15\text{-}11$$

$$f_{tD} = \frac{v_n \mathrm{SIGDE} n + v_p \mathrm{SIGDH} n_i exp\left[\frac{E_i - E}{kT}\right]}{v_n \mathrm{SIGDE}\left(n + n_i exp\left[\frac{E - E_i}{kT}\right]\right) + v_p \mathrm{SIGDH}\left(p + n_i exp\left[\frac{E_i - E}{kT}\right]\right)} \quad 15\text{-}12$$

where v_n is the electron thermal velocity, v_p is the hole thermal velocity, n_i is the intrinsic carrier concentration, SIGAE and SIGAH are the acceptor electron and hole capture cross-sections respectively and SIGDE and SIGDH are the donor electron and hole capture cross-sections respectively.

Table 15-3. User-Specifiable Paramters for Equations 15-11 and 15-12			
Statement	**Parameter**	**Default**	**Units**
ODEFECTS	SIGAE	0	cm^{-3}

Table 15-3. User-Specifiable Paramters for Equations 15-11 and 15-12			
ODEFECTS	SIGAH	0	cm^{-3}
ODEFECTS	SIGDE	0	K
ODEFECTS	SIGDH	0	K

Steady-State and Transient Recombination

The steady-state net recombination/generation rate is identical for electrons (R_n) and holes (R_p) as equilibrum is assumed to be instantaneous. The Schockley-Read-Hall recombination/generation rate due to the defect states is given by:

$$R_{n,p} = \int_{Ev}^{Ec} \left[\frac{v_n v_p \mathrm{SIGAE\ SIGAH}(np - n_i^2) g_A(E)}{v_n \mathrm{SIGAE}\left(n + n_i exp\left[\frac{E - E_i}{kT}\right]\right) + v_p \mathrm{SIGAH}\left(p + n_i exp\left[\frac{E_i - E}{kT}\right]\right)} \right. \quad \text{15-13}$$

$$\left. + \frac{v_n v_p \mathrm{SIGDE\ SIGDH}(np - n_i^2) g_D(E)}{v_n \mathrm{SIGDE}\left(n + n_i exp\left[\frac{E - E_i}{kT}\right]\right) + v_p \mathrm{SIGDH}\left(p + n_i exp\left[\frac{E_i - E}{kT}\right]\right)} \right] dE$$

In transient simulations, the instantaneous equilibrium can no longer be assumed. Instead, the recombination/generation rate is calculated using the transient acceptor and donor probabilities of occupation. These are calculated by solving additional rate equations:

$$\frac{d}{dt}(n_A) = \int_{Ev}^{Ec} g_A(E) \left[v_n \mathrm{SIGAE}\left(n(1 - f_{tA}(E)) - f_{tA}(E) n_i exp\left[\frac{E - E_i}{kT}\right]\right) \right. \quad \text{15-14}$$

$$\left. - v_p \mathrm{SIGAH}\left(p f_{tA}(E) - (1 - f_{tA}(E)) n_i exp\left[\frac{E - E_i}{kT}\right]\right) \right] dE$$

$$\frac{d}{dt}(p_D) = \int_{Ev}^{Ec} g_D(E) \left[v_n \mathrm{SIGDE}\left(n(1 - f_{tD}(E)) - f_{tD}(E) n_i exp\left[\frac{E - E_i}{kT}\right]\right) \right. \quad \text{15-15}$$

$$\left. - v_p \mathrm{SIGDH}\left(p f_{tD}(E) - (1 - f_{tD}(E)) n_i exp\left[\frac{E - E_i}{kT}\right]\right) \right] dE$$

Additional Parameters

If the parameter FAST is specified in the ODEFECTS statement, then the traps are assumed to reach equilibrum instantaneously. Therefore, Equations 15-14 and 15-15 will not be solved. In this case, the steady-state probabilities of a occupation will be used in transient simulations.

You can use the C-Interpreter to define the defect states in the bandgap. The F.OTFTDON and F.OTFTACC parameters of the ODEFECTS statement indicate the filenames containing the C functions. For more information on using the C-Interpreter, see Appendix A: “C-Interpreter Functions”.

```
ODEFECTS F.OTFTDON=mydefects.c F.OTFTACC=mydefects.c
```

The file, *mydefects.c*, will contain C functions for donor and acceptor defect densities as a function of energy. The following example defect states are defined in the file, *otft.lib*. The resultant distribution of defects versus energy can be plotted in the files, *odon.dat* and *oacc.dat*.

```
ODEFECTS F.OTFTDON=otft.lib F.OTFTACC=otft.lib DFILE=odon.dat AFILE=oacc.dat \
        SIGAE=1.E-16 SIGAH=1.E-14 SIGDE=1.E-14 SIGDH=1.E-16
```

If you specify the CONTINUOUS parameter in the ODEFECTS statement, then a numerical integral will be used to calculate the charge and recombination integrals. In this case, you can use the parameters NUMA and NUMD to specify the number of acceptor and donor energy levels intervals used in the integral. If CONTINUOUS is not specified, then the integrals will be modelled with discrete energy levels. The integral terms are replaced by summations, which run over the number of discrete energy levels (NUMA for acceptors and NUMD for donors).

Table 15-4. Additional Parameters for the ODEFECTS statement

Statement	Parameter	Default	Units
ODEFECTS	AFILE		
ODEFECTS	CONTINUOUS	False	
ODEFECTS	DFILE		
ODEFECTS	F.OTFTACC		
ODEFECTS	F.OTFTDON		
ODEFECTS	FAST	False	
ODEFECTS	NUMA	12	
ODEFECTS	NUMD	12	

15.2.2: Hopping Mobility Model

If the effective transport hopping energy model is specified (HOPPING on the ODEFECTS statement), then the field-independent hopping mobility can be used (HOPMOB on the MODELS statement). This is given by [14]:

$$\mu_{n0} = \frac{q\,\mathrm{HOPN.V0}}{kT}\left[\int_{-\infty}^{E_{tr_n}} g_A(E)dE\right]^{-2/3} exp\left[-2\left(\frac{3\,\mathrm{HOPN.BETA}}{4\pi}\right)^{1/3}\mathrm{HOPN.GAMMA}\left[\int_{-\infty}^{E_{tr_n}} g_A(E)dE\right]^{-1/3}\right] \quad \text{15-16}$$

$$\mu_{p0} = \frac{q\,\mathrm{HOPP.V0}}{kT}\left[\int_{-\infty}^{E_{tr_p}} g_D(E)dE\right]^{-2/3} exp\left[-2\left(\frac{3\,\mathrm{HOPP.BETA}}{4\pi}\right)^{1/3}\mathrm{HOPP.GAMMA}\left[\int_{-\infty}^{E_{tr_p}} g_D(E)dE\right]^{-1/3}\right] \quad \text{15-17}$$

where HOPN.V0 is the attempt-to-jump frequency for electrons and HOPP.V0 is the attempt-to-jump frequency for holes.

Table 15-5. User-Specifiable Parameters for Equations 15-16 and 15-17			
Statement	**Parameter**	**Value**	**Units**
MODEL	HOPMOB	False	
MOBILITY	HOPMOB.N	False	
MOBILITY	HOPMOB.P	False	
MATERIAL	HOPN.V0	1.0e11	Hz
MATERIAL	HOPP.V0	1.0e11	Hz

15.2.3: Poole-Frenkel Mobility Model

The Poole-Frenkel mobility model is given by [58][72]:

$$\mu_n(E) = \mu_{n0} exp\left(-\frac{\texttt{DELTAEN.PFMOB}}{kT_{neff}} + \left(\frac{\texttt{BETAN.PFMOB}}{kT_{neff}} - \texttt{GAMMAN.PFMOB}\right)\sqrt{|E|}\right) \quad \text{15-18}$$

$$\mu_p(E) = \mu_{p0} exp\left(-\frac{\texttt{DELTAEP.PFMOB}}{kTpeff} + \left(\frac{\texttt{BETAP.PFMOB}}{kTpeff} - \texttt{GAMMAP.PFMOB}\right)\sqrt{|E|}\right) \quad \text{15-19}$$

where $\mu_n(E)$ is the field dependent mobility, μ_{n0} is the zero field mobility, and E is the electric field. DELTAEN.PFMOB and DELTAEP.PFMOB are the activation energy at zero electric field for electrons and holes respectively. BETAN.PFMOB is the electron Poole-Frenkel factor, and BETAP.PFMOB is the hole Poole-Frenkel factor.

T_{neff} is the effective temperature for electrons and is defined as:

$$T_{neff} = \frac{\texttt{T0N} \cdot \texttt{PFMOB}\ T}{\texttt{T0N} \cdot \texttt{PFMOB} - T} \quad \text{15-20}$$

T_{peff} is the effective temperature for holes and is defined as:

$$T_{peff} = \frac{\texttt{T0P} \cdot \texttt{PFMOB}\ T}{\texttt{T0P} \cdot \texttt{PFMOB} - T} \quad \text{15-21}$$

If you specify the BETANAUTO.PFMOB parameter on the MOBILITY statement, then BETAN.PFMOB in Equation 15-18 will be calculated from the permittivity:

$$\texttt{BETAN.PFMOB} = q\sqrt{\frac{q}{\pi\varepsilon\varepsilon 0}} \quad \text{15-22}$$

Similarly, if you specify BETAPAUTO.PFMOB, then BETAP.PFMOB in Equation 15-19 will be calculated from the permittivity.

If you use E0N.PFMOB, E0P.PFMOB, E0NAUTO.PFMOB or E0PAUTO.PFMOB instead of BETAN.PFMOB, BETAP.PFMOB, BETANAUTO.PFMOB and BETAPAUTO.PFMOB, you will then use a simplified form of the Poole-Frenkel mobility [72][134]:

$$\mu_n(E) = \mu_{n0} exp\left(-\frac{\texttt{DELTAEN.PFMOB}}{kT_{neff}} + \sqrt{\frac{|E|}{\texttt{E0N.PFMOB}}}\right) \quad \text{15-23}$$

$$\mu_p(E) = \mu_{p0} exp\left(-\frac{\texttt{DELTAEP.PFMOB}}{kT_{peff}} + \sqrt{\frac{|E|}{\texttt{E0P.PFMOB}}}\right) \quad \text{15-24}$$

where *E0* is the characteristic field. You can use the parameters E0NAUTO.PFMOB and E0PAUTO.PFMOB in the MOBILITY statement to calculate the electron and hole characteristic field values from the permittivity:

$$E0 = \left(\frac{kT}{q}\right)^2 \left(\frac{\pi\varepsilon 0}{q}\right) \quad \text{15-25}$$

If you specify the PFMOB parameter on the MODELS statement, the Poole-Frenkel mobility model will be used for both electrons and holes. You can specify this model individually for electrons and holes by using the PFMOB.N and PFMOD.P parameters in the MOBILITY statement.

Table 15-6. User-Specifiable Parameters for Poole-Frenkel Mobility Model

Statement	Parameter	Default	Units
MOBILITY	BETAN.PFMOB	0	$eV(cm/V)^{1/2}$
MOBILITY	BETAP.PFMOB	0	$eV(cm/V)^{1/2}$
MOBILITY	BETANAUTO.PFMOB	False	
MOBILITY	BETAPAUTO.PFMOB	False	
MOBILITY	DELTAEN.PFMOB	0	eV
MOBILITY	DELTAEP.PFMOB	0	eV
MOBILITY	EON.PFMOB	1956	V/cm
MOBILITY	EOP.PFMOB	1956	V/cm
MOBILITY	EONAUTO.PFMOB	False	
MOBILITY	EOPAUTO.PFMOB	False	
MOBILITY	GAMMAN.PFMOB	0	$(cm/V)^{1/2}$
MOBILITY	GAMMAP.PFMOB	0	$(cm/V)^{1/2}$
MOBILITY	PFMOB	False	
MOBILITY	PFMOB.N	False	
MOBILITY	PFMOB.P	False	
MOBILITY	TON.PFMOB	0	K
MOBILITY	TOP.PFMOB	0	K

15.2.4: Bimolecular Langevin Recombination Model

To enable this model, specify the parameter LANGEVIN in the MODELS statement. The Langevin recombination is given by [24][140]:

$$R_{n,p} = \frac{qnP}{\varepsilon\varepsilon 0}(\mu_n(E) + \mu_p(E)) \quad \text{15-26}$$

15.3: Singlet and Triplet Excitons

The distribution of singlet or triplet excitons can be used to infer the radiative rates for luminescense or phosphorescence in organic material LEDs. In ATLAS, you can self-consistently solve the singlet and triplet exciton continuity equations along with the electron and hole drift diffusion equations. The singlet exciton continuity equation is given by [134] and [168]:

$$\frac{dS(x,y,t)}{dt} = \texttt{RST.EXCITON} r(x,y,t) n(x,y,t) p(x,y,t) \quad \text{15-27}$$

$$+ \frac{\texttt{RST.EXCITON}}{1 + \texttt{RST.EXCITON}} \texttt{KTT.EXCITON}\ T^2(x,y,t) + \ -\texttt{KISC.EXCITON}\ S(x,y,t)$$

$$-\texttt{KST}\cdot\texttt{EXCITON}\ S(x,y,t)\ \ T(x,y,t)$$

$$-\texttt{KSP}\cdot\texttt{EXCITON}\ S(x,y,t)(n(x,y,t) + p(x,y,t))$$

$$-\frac{\texttt{KSS}\cdot\texttt{EXCITON}}{2}\ S^2(x,y,t)\ -\texttt{KNRS.EXCITON}\ S(x,y,t)$$

$$-\frac{S(x,y,t)}{\texttt{TAUS}\cdot\texttt{EXCITON}} + \nabla(D_s \nabla S(x,y,t))$$

while the triplet exciton continuity equation is given by:

$$\frac{dT(x,y,t)}{dt} = (1 - \texttt{RST.EXCITON}) r(x,y,t) n(x,y,t) p(x,y,t)$$

$$+\texttt{KISC.EXCITON}\ S(x,y,t) - \texttt{KTP.EXCITON}\ T(x,y,t)(n(x,y,t) + p(x,y,t))$$

$$-\texttt{KTT.EXCITON}\ T^2(x,y,t) - \texttt{KNRT}\cdot\texttt{EXCITON}\ T(x,y,t) \quad \text{15-28}$$

$$-\frac{\texttt{T}}{\texttt{TAUT}\cdot\texttt{EXCITON}} + \nabla(D_T \nabla T(x,y,t))$$

where:

- *S(x,y,t)* is the spatial and temporal singlet exciton density,
- *T(x,y,t)* is the spatial and temporal triplet exciton density,
- RST.EXCITON is the fraction of singlets formed,
- *r(x,y,t)* is the Langevin recombination rate,
- *n(x,y,t)* is the electron concentration,
- *p(x,y,t)* is the hole concentration,
- TAUS.EXCITON is the singlet radiative decay lifetime,
- TAUT.EXCITON is the triplet radiative decay lifetime,
- KISC.EXCITON is the intersystem crossing constant,
- KST.EXCITON is the singlet-triplet constant,
- KSP.EXCITON is the singlet-polaron constant,
- KTP.EXCITON is the triplet-polaron constant,
- KSS.EXCITON is the singlet-singlet constant,
- KTT.EXCITON is the triplet-triplet constant,
- KNRS.EXCITON is the singlet non-radiative decay constant
- KNRT.EXCITON is the triplet non-radiative decay constant.

The singlet diffusion constant, D_S, can be expressed as:

$$D_S = \frac{\text{LDS.EXCITON}^2}{\text{TAUS.EXCITON}} \quad \text{15-29}$$

where LDS.EXCITON is the singlet diffusion length.

The triplet diffusion constant, D_T, can be expressed as:

$$D_T = \frac{\text{LDT.EXCITON}^2}{\text{TAUT.EXCITON}} \quad \text{15-30}$$

where LDT.EXCITON is the triplet diffusion length.

Table 15-7. User-Specifiable Parameters for Equations 15-27, 15-28, 15-29, and 15-30

Statement	Parameter	Default	Units
MATERIAL	KISC.EXCITON	0	s^{-1}
MATERIAL	KNRS.EXCITON	0	s^{-1}
MATERIAL	KNRT.EXCITON	0	s^{-1}
MATERIAL	KSP.EXCITON	0	$cm^3\ s^{-1}$
MATERIAL	KST.EXCITON	0	$cm^3\ s^{-1}$
MATERIAL	KTP.EXCITON	0	$cm^3\ s^{-1}$
MATERIAL	KSS.EXCITON	0	$cm^3\ s^{-1}$
MATERIAL	KTT.EXCITON	0	$cm^3\ s^{-1}$
MATERIAL	LDS.EXCITON	0.01	microns
MATERIAL	LDT.EXCITON	0.0632	microns
MATERIAL	RST.EXCITON	0.25	
MATERIAL	TAUS.EXCITON	1×10^{-9}	s
MATERIAL	TAUT.EXCITON	1×10^{-4}	s

Table 15-8. User-Specifiable Parameters for Tetracene

Statement	Parameter	Default	Units
MATERIAL	KISC.EXCITON	0	s^{-1}
MATERIAL	KNRS.EXCITON	0	s^{-1}
MATERIAL	KNRT.EXCITON	0	s^{-1}
MATERIAL	KSP.EXCITON	4×10^{-8}	$cm^3\ s^{-1}$

Table 15-8. User-Specifiable Parameters for Tetracene			
MATERIAL	KST.EXCITON	2×10^{-7}	$cm^3\ s^{-1}$
MATERIAL	KTP.EXCITON	2×10^{-9}	$cm^3\ s^{-1}$
MATERIAL	KSS.EXCITON	5×10^{-8}	$cm^3\ s^{-1}$
MATERIAL	KTT.EXCITON	1×10^{-9}	$cm^3\ s^{-1}$
MATERIAL	LDS.EXCITON	0.01	microns
MATERIAL	LDT.EXCITON	0.0632	microns
MATERIAL	RST.EXCITON	0.25	
MATERIAL	TAUS.EXCITON	1×10^{-7}	
MATERIAL	TAUT.EXCITON	1×10^{-4}	

To enable the self-constant simulation of both singlet and triplet excitons, specify EXCITONS in the MODEL statement. Specifying SINGLET instead of EXCITONS will enable the solution of the singlet exciton equations, while specifying TRIPLET will enable the triplet exciton equation.

Table 15-9. Exciton Model Flags		
Statement	**Parameter**	**Default**
MODELS	EXCITONS	False
MODELS	SINGLET	False
MODELS	TRIPLET	False

When solving for excitons, you can examine the density distribution of the singlet and triplet excitons in TONYPLOT in files saved in the standard structure file format. You can also specify EXCITON in the PROBE statement to save the singlet exciton density to log files. For further analysis of LED devices, see Chapter 11: "LED: Light Emitting Diode Simulator".

Chapter 16:
NOISE: Electronic Noise Simulator

16.1: Introduction

NOISE is an ATLAS based product that simulates the small-signal noise generated by devices. Electronic noise results in an unavoidable degradation of a circuit's performance. It is important to understand the properties of noise to minimize its effect.

You should already be familiar with ATLAS before using NOISE. If not, see Chapter 2: "Getting Started with ATLAS".

16.2: Simulating Noise in ATLAS

ATLAS models the noise of a device by calculating the statistical behavior of equivalent random voltage sources at its ports. You can use the results of this calculation in a circuit simulator (e.g., SMARTSPICE) to minimize the disruption caused by the (inevitable) presence of noise. The following shows the minimum needed to simulate noise in ATLAS.

1.Add the `noise` parameter to a `log` command. For example:

```
log outfile=noise.log inport=gate outport=drain noise
```

2. Use the `noise` parameter on the subsequent `solve` command. For example:

```
solve noise frequency=100.0
```

NOISE is an extension of the AC analysis of a device (see Chapter 2: "Getting Started with ATLAS", Section 2.9.3: "Small-Signal AC Solutions" for more information on AC analysis). You can perform a noise simulation on any device (one-port or two-port) where small-signal AC analysis can be performed.

A direct AC analysis is automatically performed on the device before the noise simulation (i.e., you don't need to perform the analysis).

For more information about the `LOG` and `SOLVE` commands, see Chapter 19: "Statements", Sections 19.22: "LOG" and 19.44: "SOLVE".

16.3: Circuit Level Description of Noise

Noise is the name given to the random fluctuations in the currents and voltages of real devices.

The simple Drude model of an n-type resistor gives

$$\langle J \rangle = \langle \sigma \rangle \cdot E = q \cdot \langle n \rangle \cdot \langle v \rangle \quad \text{16-1}$$

where $\langle J \rangle$ is the mean current density, q is the electronic charge, $\langle n \rangle$ is the mean electron density, and $\langle v \rangle$ is the mean electron drift velocity. Time dependent fluctuations in the instantaneous electron density and drift velocity will cause fluctuations in the resultant current density.

The current at a contact of the resistor would be

$$i(t) = \langle i \rangle + \delta i(t) \quad \text{16-2}$$

where the $\delta i(t)$ term is the noise generated by the resistor.

A noisy device can be represented by external current sources added to the terminals of an ideal (noiseless) device (see Figure 16-1).

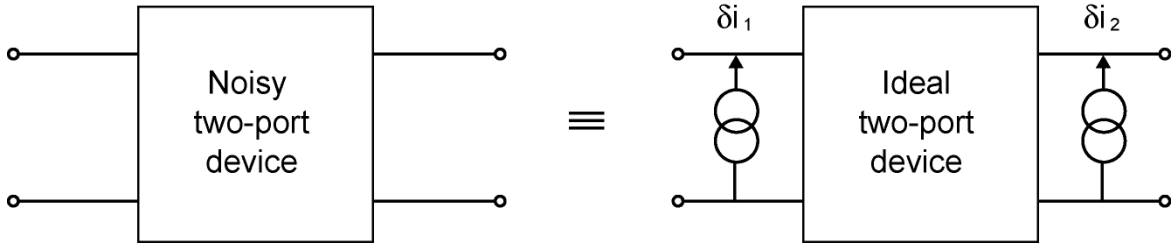


Figure 16-1: A real (noisy) device is modelled as an ideal (noiseless) device with random current sources attached to the ports.

These current sources are small and random. The description of noise is the statistical properties of these current sources. Normally, you describe the behavior of the frequency domain representation of the current sources.

The means of the current sources are zero. For example:

$$\langle \delta i_1(\omega) \rangle = 0 \ \ and \ \ \langle \delta i_2(\omega) \rangle = 0 \quad \text{16-3}$$

The noise is described by the auto-correlation:

$$\langle \delta i_1^2(\omega) \rangle \ \ and \ \ \langle \delta i_2^2(\omega) \rangle \quad \text{16-4}$$

and the cross-correlation:

$$\langle \delta i_1(\omega) \cdot \delta i_2(\omega)^* \rangle \ \ and \ \ \langle \delta i_2(\omega) \cdot \delta i_1(\omega)^* \rangle \quad \text{16-5}$$

It may be more convenient to describe the noise in terms of the behavior of voltage sources instead of current sources. The noise current can be easily translated into a noise voltage. For example:

$$\begin{pmatrix} \delta v_1(\omega) \\ \delta v_2(\omega) \end{pmatrix} = \begin{pmatrix} Z_{11}(\omega) & Z_{12}(\omega) \\ Z_{21}(\omega) & Z_{22}(\omega) \end{pmatrix} \cdot \begin{pmatrix} \delta i_1(\omega) \\ \delta i_2(\omega) \end{pmatrix} \quad \text{16-6}$$

Figure 16-2 shows that both representations are the same.

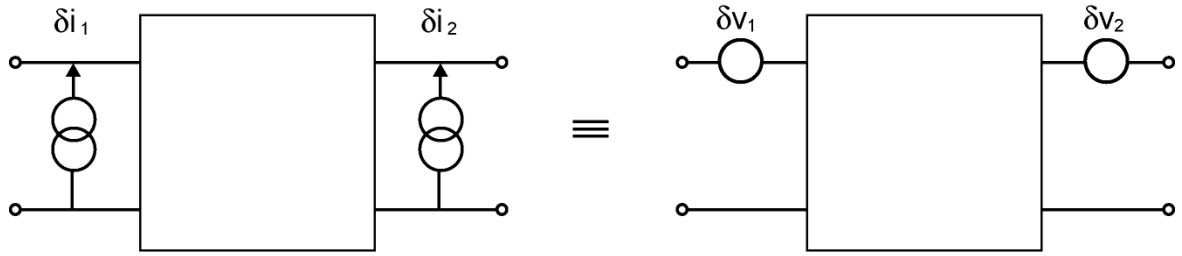


Figure 16-2: The noise can be modelled with random current sources or random voltage sources.

16.3.1: Noise "Figures of Merit" for a Two-Port Device

Figure 16-3 shows a simple two-port circuit.

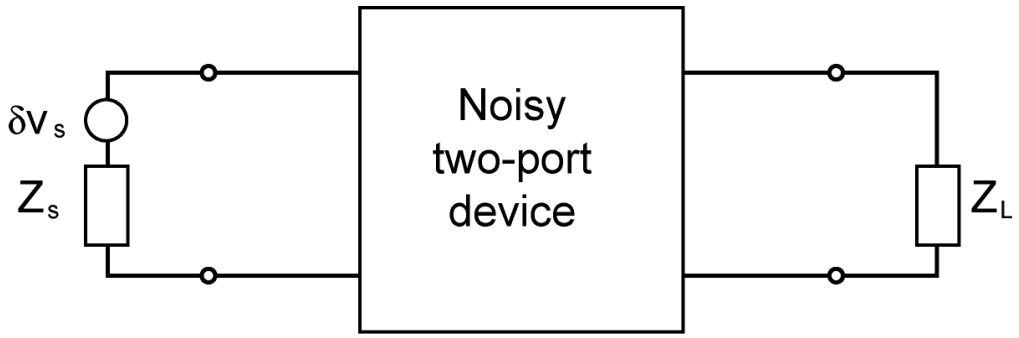


Figure 16-3: A block diagram of a simple circuit. We are interested in how much noise reaches the load.

The noise figure, *F*, is a measure of the increased amount of noise power delivered to the load because of the device's noise.

$$F = \frac{P_{S,D}}{P_S} \quad \text{16-7}$$

Here:

- P_S is the noise power that would be delivered to the load if the device was noiseless (i.e., only the noise from the source is transferred to the load).
- $P_{S,D}$ is the noise power that is delivered to the load from both the source and the device.

The standard definition of noise figure assumes the source is generating thermal noise at the temperature of the device:

$$\langle v_s^2 \rangle = 4kTR_s \quad \text{16-8}$$

Here, we have for the two-port circuit:

$$F = F_{min} + \frac{g_n}{R_s}\left|Z_S - Z_0\right|^2 \qquad 16\text{-}9$$

where the traditional two-port figures of merit for noise are as follows:

- the minimum noise figure, F_{min} ;
- the best source impedance, Z_0 ;
- the noise conductance, g_n.

The best source impedance gives the minimum noise figure (i.e., F=F_{min} when Z_S=Z_0). The noise conductance is a measure of how much the noise figure increases as the source impedance moves away from Z_0.

16.4: Noise Calculation

ATLAS does the following to calculate the noise of a device.

1.Takes a small volume of a device and calculates the random current flucluations in that volume.

2. Uses the impedence field to calculate the resultant voltage on a contact.

3. Repeats the above steps until the noise from each part of the device has been calculated, which is then the total noise of a device.

16.4.1: The Impedance Field

The impedance field is a transfer function relating current (injected at some point in the device) to the resultant contact voltage.

Suppose a current is injected into a point ***r*** of a device (see Figure 16-4).

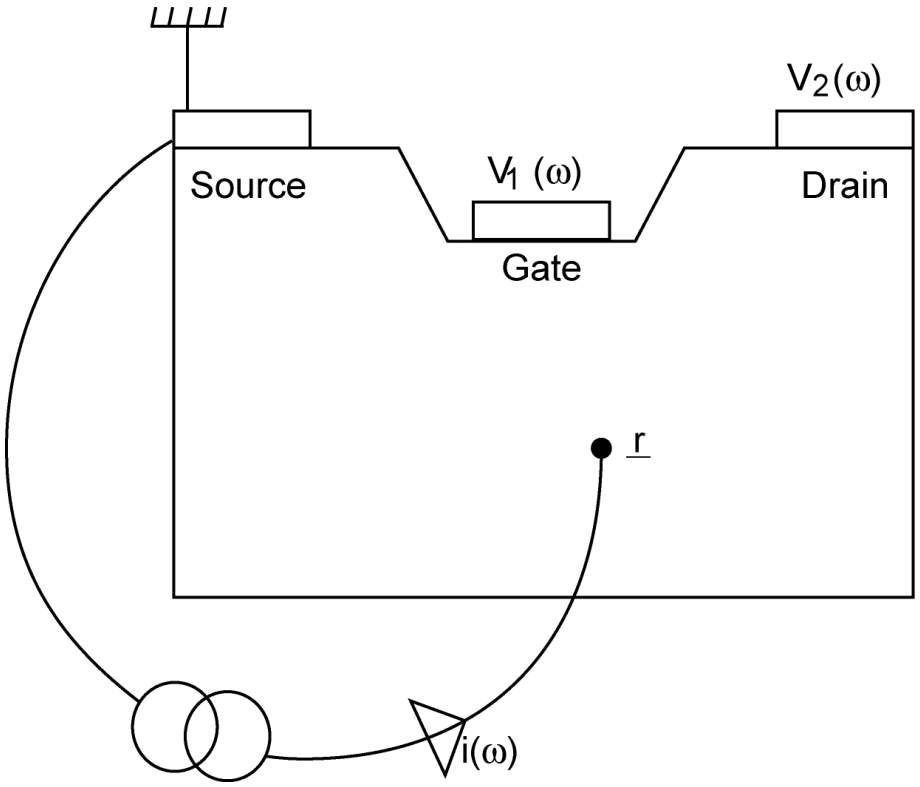


Figure 16-4: We calculate the impedance field at point *r* by injecting a unit current at the point getting the resultant voltages V1 and V2. (This is impossible to do in practice but is easy for ATLAS to simulate.)

This will generate a voltage on the open circuit contacts. The device is assumed to be linear so that the voltage will be at the same frequency as the input current. Here, we get

$$v_1(\omega) = Z_1(\mathbf{r};\omega) \cdot i(\omega) \qquad 16\text{-}10$$

and

$$v_2(\omega) = Z_2(\mathbf{r};\omega) \cdot i(\omega) \qquad 16\text{-}11$$

The $Zi(r;\omega)$ parameters, when taken over the whole device, are called the impedance fields.

There are separate impedance fields for when the injected current is electrons or holes. In other words, a two-port device has four associated impedance fields: $Z^n_1(\boldsymbol{r};\omega)$, $Z^n_2(\boldsymbol{r};\omega)$, $Z^p_1(\boldsymbol{r};\omega)$, and $Z^p_2(\boldsymbol{r};\omega)$.

16.4.2: Microscopic Noise Source

We assume the statistical behavior of the noise from one point in the device is totally uncorrelated with the statistical behavior of the noise at all other points. So we describe the correlation of a parameter, x, as:

$$\langle \delta x \cdot \delta x' \rangle(\boldsymbol{r}, \boldsymbol{r}';\omega) = \boldsymbol{k}_{\delta x,\ \delta x}(\boldsymbol{r};\omega) \cdot \delta(\boldsymbol{r} - \boldsymbol{r}') \qquad 16\text{-}12$$

where the term $k_{\delta x,\ \delta x}(\boldsymbol{r};\omega)$ is called the microscopic noise source.

The theory of Generation-Recombination noise gives the auto-correlation of the current:

$$k_{\delta i,\ \delta i}(\boldsymbol{r};\omega) \qquad 16\text{-}13$$

The theory of Diffusion noise gives the auto-correlation of the current density:

$$k_{\delta \boldsymbol{J},\ \delta \boldsymbol{J}}(\boldsymbol{r};\omega) \qquad 16\text{-}14$$

16.4.3: Local Noise Source

The local noise source is the effect that a microscopic noise source has on the noise behavior of the device.

For $k_{\delta i,\ \delta i}(\boldsymbol{r};\omega)$ we have

$$\langle \delta v_\alpha, \delta v_\beta^{\ *} \rangle(\omega) = \int_\Omega Z_\alpha(\boldsymbol{r};\omega) k_{\delta i,\ \delta i}(\boldsymbol{r};\omega) \cdot Z_\beta^{\ *}(\boldsymbol{r};\omega)\, dV\, d\boldsymbol{r} \qquad 16\text{-}15$$

where the integral is over the device volume. The integrand function is called the local noise source.

For $k_{\delta J,\ \delta J}(r;\omega)$ we have

$$\langle \delta v_\alpha, \delta v_\beta^{\ *} \rangle(\omega) = \int_\Omega \boldsymbol{Z}_\alpha(\boldsymbol{r};\omega) k_{\delta \boldsymbol{J},\ \delta \boldsymbol{J}}(\boldsymbol{r};\omega) \cdot \boldsymbol{Z}_\beta^{\ *}(\boldsymbol{r};\omega) \cdot d\boldsymbol{r} \qquad 16\text{-}16$$

Again, the integral is over the device volume and the integrand function is called the local noise source.

The vector impedance field is defined as

$$Z(r;\omega) = \nabla Z(r;\omega) \qquad 16\text{-}17$$

16.5: ATLAS Models

ATLAS has models for three types of microscopic noise source: Diffusion Noise, Generation-Recombination Noise, and Flicker Noise.

If noise is being calculated then diffusion noise is on by default. Generation-recombination noise is on if any Generation-Recombination (GR) or Impact Ionization (II) models are defined. Flicker noise is off by default.

The default noise models for diffusion and GR/II have no user-definable parameters. They get all the needed information from the simulation.

16.5.1: Diffusion Noise

Diffusion noise is caused by variations in the velocity of the carriers.

Einstein Diffusion Equation

There are no independent user-definable parameters for this model. You can set the mobility model in a MATERIAL statement (See Chapter 19: "Statements", Section 19.24: "MATERIAL") but that will affect the whole simulation not just the noise calculation. It is on by default and can only be turned off by choosing a different model for diffusion noise.

The equation for the microscopic noise source is

$$k_{\delta \boldsymbol{J},\ \delta \boldsymbol{J}}(\boldsymbol{r};\omega) = 4q^2 n(\boldsymbol{r}) D(\omega) \qquad \text{16-18}$$

where:

- q is the fundamental electronic charge.
- $n(\boldsymbol{r})$ is the carrier concentration.
- $D(\omega)$ is the diffusion coefficient.

The diffusion coefficient is given by Einstein's relationship, which is:

$$D(\omega) = \frac{kT}{q}\mu \qquad \text{16-19}$$

where:

- k is Boltzmann's constant.
- T is the temperature.
- μ is the carrier mobility.

C-Interpreter

You can define a C-Interpreter function to calculate the microscopic noise source for diffusion noise. If a C-Interpreter function is defined, then the default model is turned off.

For a C-Interpreter function to calculate the electron microscopic noise source, use the MATERIAL statement:

```
material F.MNSNDIFF=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for electron diffusion noise.
 * Statement: MATERIAL
 * Parameter: F.MNSNDIFF
 * Arguments:
```

```
 * xcomp      composition fraction x
 * ycomp      composition fraction y
 * temp       lattice temperature (K)
 * n          electron concentration (cm^-3)
 * mun        electron mobility (cm^2/V s)
 * e          electric field (V/cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnsndiff(double xcomp, double ycomp, double temp, double n,
double mun, double e, double *mns);
```

For a C-Interpreter function to calculate the hole microscopic noise source, use the MATERIAL statement:

```
material F.MNSPDIFF=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for hole diffusion noise.
 * Statement: MATERIAL
 * Parameter: F.MNSPDIFF
 * Arguments:
 * xcomp      composition fraction x
 * ycomp      composition fraction y
 * temp       lattice temperature (K)
 * p          hole concentration (cm^-3)
 * mup        hole mobility (cm^2 / V s)
 * e          electric field (V / cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnspdiff(double xcomp, double ycomp, double temp, double p,
double mup, double e, double *mns);
```

16.5.2: Generation-Recombination Noise

Generation-Recombination noise is caused by variations in the number of the carriers.

There are no user-definable parameters for GR noise. You can, however, set the parameters for the various GR models in the MATERIAL statement (see Chapter 19: “Statements”, Section 19.24: “MATERIAL”) and the various II models in the IMPACT statement (Chapter 19: “Statements”, Section 19.16: “IMPACT”). Setting these parameters will affect the whole simulation, not just the noise calculation.

There are two families of GR in ATLAS: direct and trap assisted. Direct GR is when the electron travels directly from the conduction band to the valence band (or vice versa). Trap assisted GR is when the electron travels from a band to a trap level. If more than one trap level is defined, then they are treated as independent trap assisted GR sites (i.e., electrons are not assumed to move between different trap levels).

Generation-Recombination is defined by four parameters:

- G_n
- G_p
- R_n
- R_P

G is the generation rate, R is the recombination rate, n is for electrons, and p is for holes. The generation rate of electrons is the number of electrons that appears (per unit time, per unit volume) in the conduction band. The recombination rate is the number of electrons that disappears from the conduction band.

Direct GR

Since there are no trap levels, all electrons leaving the conduction band end up in the valence band. All electrons leaving the valence band end up in the conduction band.

Therefore:

$$G_n = G_p \ and \ R_n = R_p \tag{16-20}$$

The GR parameters are calculated with

$$\begin{aligned} G &= C \cdot n_i^2 \\ R &= C \cdot np \end{aligned} \tag{16-21}$$

For optical GR (`OPTR` in the `MODELS` statement), the C parameter is `COPT` in the `MATERIALS` statement. For Auger GR (`AUGER` in the `MODELS` statement), the C parameter is a function of the carrier density.

The microscopic noise sources are

$$k_{\delta n,\, \delta n} = k_{\delta p,\, \delta p} = k_{\delta n,\, \delta p} = 2(G + R) \tag{16-22}$$

Trap Assisted GR

There are assumed to be no direct transitions in trap assisted GR. Therefore, an electron leaving the conduction band ends up at the trap level. An electron entering the conduction band comes from the trap level. Unlike direct GR, the following four parameters are independent.

$$\begin{aligned} G_n &= \frac{n_1}{\tau_n} \frac{n_t}{N_t} \\ G_p &= \frac{p_1}{\tau_p}\left(1 - \frac{n_t}{N_t}\right) \\ R_n &= \frac{n}{\tau_n}\left(1 - \frac{n_t}{N_t}\right) \\ R_p &= \frac{p}{\tau_p} \frac{n}{N_t} \end{aligned} \tag{16-23}$$

The ratio n_t/N_t is the fraction of ionized traps. The concentration n_1 is the electron concentration if the Fermi level were at the trap level. p_1 is the hole concentration if the Fermi level were at the trap level.

You can define the lifetimes τ_n and τ_p using the `TAUN0` and `TAUP0` parameters from the `MATERIALS` statement.

The microscopic noise sources are

$$\begin{aligned} k_{\delta n,\, \delta n} &= 2(G_n + R_p) \\ k_{\delta p,\, \delta p} &= 2(G_n + R_p) \end{aligned} \tag{16-24}$$

There are no direct transitions between the conduction and valence band. Therefore:

$$k_{\delta n,\ \delta p} = 0 \qquad 16\text{-}25$$

Impact Ionization

Impact ionization is a pure generation term. For each electron created in the conduction band, a corresponding hole is also created in the valence band. This is similar to direct GR without a recombination term.

The microscopic noise sources are

$$k_{\delta n,\ \delta n} = k_{\delta p,\ \delta p} = k_{\delta p} = 2G \qquad 16\text{-}26$$

16.5.3: Flicker Noise

Flicker noise is observed experimentally, however, a complete theory about what causes flicker noise isn't known [26].

Hooge

The Hooge model, which is shown below, is a phenomenological microscopic noise source.

$$K_{\delta Jn,\ \delta Jn}(\boldsymbol{r};\omega) = \frac{\alpha_{Hn}}{f}\frac{\left|J_n(\boldsymbol{r})\right|^2}{n(\boldsymbol{r})} \quad and \quad K_{\delta Jp\ ,\ \delta Jp}(\boldsymbol{r};\omega) = \frac{\alpha_{Hp}}{f}\frac{\left|J_p(\boldsymbol{r})\right|^2}{p(\boldsymbol{r})} \qquad 16\text{-}27$$

In this model:

- α_{Hn} is the electron Hooge constant.
- α_{Hp} is the hole Hooge constant.
- f is the frequency.
- $J_n(\boldsymbol{r})$ is the electron current density.
- $J_p(\boldsymbol{r})$ is the hole current density.
- $n(\boldsymbol{r})$ is the electron concentration.
- $p(r)$ is the hole concentration.

The Hooge constants are defined in the MATERIAL statement:

```
material HOOGEN=<value> HOOGEP=<value>
```

The default values for these parameters is zero (which means the flicker noise is turned off).

C-Interpreter

You can define a C-Interpreter function to calculate the microscopic noise source for flicker noise. If a C-Interpreter function is defined, then the default model is turned off.

For a C-Interpreter function to calculate the electron microscopic noise source, use the MATERIAL statement:

```
material F.MNSNFLICKER=<filename>
```

The header for this C-Interpreter function is:

```
/*
 * Microscopic Noise Source for electron flicker (1/f) noise.
 * Statement: MATERIAL
 * Parameter: F.MNSNFLICKER
```

```
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * temp       lattice temperature (K)
 * n          electron concentration (cm^-3)
 * jn         electron current density (A / cm^2)
 * f          frequency (Hz)
 * e          electric field (V / cm)
 * *mns       microscopic noise source (s A^2 / cm)
 */
int mnsnflicker(double x, double y, double temp, double n,
double jn, double f, double e, double *mns);
```

For a C-Interpreter function to calculate the hole microscopic noise source, use the MATERIAL statement:

```
material F.MNSPFLICKER=<filename>
```

The header for this C-Interpreter function is

```
/*
 * Microscopic Noise Source for hole flicker (1/f) noise.
 * Statement: MATERIAL
 * Parameter: F.MNSPFLICKER
 * Arguments:
 * x          location x (microns)
 * y          location y (microns)
 * temp       lattice temperature (K)
 * p          hole concentration (cm^-3)
 * jp         hole current density (A / cm^2)
 * f          frequency (Hz)
 * e          electric field (V/cm)
 * *mns       microscopic noise source (s A^2/cm)
 */
int mnspflicker(double x, double y, double temp, double p,
double jp, double f, double e, double *mns);
```

16.6: Output

The noise simulation results can output to the log file. The intermediate values in the calculation can output to the structure file.

16.6.1: Log Files

There are five groups of data that can output to a log file. They are as follows:

- The two-port figures of merit (FOM): F_{min}, Z_{opt}, g_n.
- The correlation of the total noise voltage sources.
- The correlation of the total noise current sources.
- The correlation of the individual noise voltage sources (GR, II, electron and hole diffusion; electron and hole flicker).
- The correlation of the individual noise current sources (GR, II, electron and hole diffusion; electron and hole flicker).

Table 16-1 shows the logical parameters in the LOG command (see Chapter 19: "Statements", Section 19.22: "LOG" for more information) that cause these groups to output. The columns in this table correspond to the groups above.

Table 16-1. Logical Parameters in the LOG command

	two-port FOM [a]	Total V correlation	Total I correlation	Individual V correlation	Individual I correlation
NOISE	X	_ [b]	-	-	-
NOISE.V	X	X	-	-	-
NOISE.I	X	-	X	-	-
NOISE.IV	X	X	X	-	-
NOISE.VI	X	X	X	-	-
NOISE.V.ALL	X	X	-	X	-
NOISE.I.ALL	X	-	X	-	X
NOISE.ALL	X	X	X	X	X

Note a: If the device is a one-port, then this data will not be output: the two-port figures of merit have no meaning for a one-port device.

Note b: If the device is a one-port and NOISE is the only noise output in the LOG command, then the total noise voltage sources will output.

16.6.2: Structure Files

There are five groups of data that can output to a structure file. They are as follows:

- Total Local Noise Source (LNS)
- Impedance fields
- Short-circuit Green's function
- Individual Microscopic Noise Sources (MNS)
- Individual local noise sources

Table 16-2 shows the logical parameters from the OUTPUT command (see Chapter 19: "Statements", Section 19.34: "OUTPUT" for more information) that cause these groups to output. The columns in this table correspond to the groups above.

Table 16-2. Logical Parameters in the OUTPUT command

	Total LNS	Impedance Fields	Short-circuit Green's function	Individual MNS	Individual LNS
NOISE	X	-	-	-	-
NOISE.IMP	-	X	-	-	-
NOISE.ALL	X	X	X	X	X

Chapter 17:
Thermal 3D: Thermal Packaging Simulator

17.1: Overview

THERMAL3D solves the steady-state heat equation to find the equilibrium temperature distribution in planar and non-planar three-dimensional structures. Specify the heat sinks and sources and choose from several temperature-dependent models for thermal conductivity within each region. A typical application for THERMAL3D is a package simulation for a power circuit or III-V ICs.

If you're unfamiliar with THERMAL3D, see also Chapter 6: "3D Device Simulator". If you're unfamiliar with ATLAS, see Chapter 2: "Getting Started with ATLAS".

17.1.1: 3D Structure Generation

THERMAL3D supports structure defined on 3-D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension. There are two methods for creating a 3-D structure that can be used with THERMAL3D. One method is through the command syntax of ATLAS. The other method is through an interface to DEVEDIT. See Chapter 6: "3D Device Simulator", Section 6.2: "3D Structure Generation" for more information on both methods.

Defining Heat Sources

Heat sources are identified with regions in the 3D structure. Regions are defined in the manner documented in Chapter 2: "Getting Started with ATLAS", the "Specifying Regions And Materials" section on page 2-12 and Chapter 6: "3D Device Simulator", the "ATLAS Syntax For 3D Structure Generation" section on page 6-3. A region has a unique number, which is used to identify the region on the MATERIAL statement. The POWER parameter of the MATERIAL statement is used to set the power of the heat source in watts.

```
MATERIAL REGION=2 POWER=0.35
```

Set the POWER parameter in the SOLVE statement if you want to step the power through a range of values. See Section 17.4: "Obtaining Solutions In THERMAL3D" for the proper syntax.

Defining Heat Sinks

Heat sinks are identified as electrodes in the 3-D structure. Heat sink areas should be defined as electrodes in the manner documented in Chapter 2: "Getting Started with ATLAS", the "Specifying Electrodes" section on page 2-13 and Chapter 6: "3D Device Simulator", the "ATLAS Syntax For 3D Structure Generation" section on page 6-3.

Each electrode (heat sink) has a unique number, which is used to set the temperature on the heat sink during simulation. For more information on setting the temperature of the heat sink, see Section 17.4: "Obtaining Solutions In THERMAL3D".

17.2: Model and Material Parameter Selection

17.2.1: Thermal Simulation Model

To obtain steady-state solutions for the temperature distribution, THERMAL3D solves Poisson's equation for temperature:

$$\nabla(k(T)\nabla T) = q \qquad 17\text{-}1$$

where T represents the steady state temperature, k the temperature-dependent thermal conductivity, and q the power generation per unit volume in the medium (heat sources).

The prescribed temperatures at the heat sinks form boundary conditions for Equation 17-1.

The solution of the heat equation by THERMAL3D is invoked by the following syntax:

```
MODELS THERMAL
```

17.2.2: Setting Thermal Conductivity

The value of thermal conductivity k in units of W/cm/K for each region should be specified in the MATERIAL statement because thermal conductivity is generally temperature dependent. The following four models available for thermal conductivity are:

$$k(T) = \mathrm{TC.CONST} \qquad 17\text{-}2$$

$$k(T) = \left(\mathrm{TCON.CONST}\right)/(T/300)^{\mathrm{TC.NPOW}} \qquad 17\text{-}3$$

$$k(T) = 1/\left(\mathrm{TC.A} + \mathrm{TC.B} \times T + \mathrm{TC.C} \times T^2\right) \qquad 17\text{-}4$$

$$k(T) = \mathrm{TC.E}/\left(T - \mathrm{TC.D}\right) \qquad 17\text{-}5$$

These models are specified on a material by material basis as follows.

To choose the model in Equation 17-2, specify the TCON.CONST (default) parameter in the MATERIAL statement. To choose the model in Equation 17-3, specify the TCON.POWER parameter in the MATERIAL statement. To choose the model in Equation 17-4, specify the TCON.POLY parameter in the MATERIAL statement. To choose the model in Equation 17-5, specify the TCON.RECIP parameter in the MATERIAL statement.

Table 17-1. User-Specifiable Parameters For Equations 17-2 to 17-5

Parameter	Statement	Units
TC.CONST	MATERIAL	W/cm/K
TC.NPOW	MATERIAL	cmk/W
TC.A	MATERIAL	
TC.B	MATERIAL	cm/W

Table 17-1. User-Specifiable Parameters For Equations 17-2 to 17-5		
Parameter	**Statement**	**Units**
TC.C	MATERIAL	cm/W/k
TC.D	MATERIAL	K
TC.E	MATERIAL	W/cm

Example

```
MATERIAL REGION=2 TCON.POWER TC.CO=1.0 TC.NPOW=1.2
```

means that the temperature-dependent thermal conductivity for region #2 is expressed as:

$$K(T) = \frac{1}{(T/300)^{1.2}} \qquad 17\text{-}6$$

17.2.3: Suggested Parameters For Thermal Conductivity

There are no preset material defaults in THERMAL3D. The following values are recommended for thermal conductivity of GaAs and Si.

For GaAs, use model B with:

```
MATERIAL TCON.POWER TC.CO=0.44 TC.POW=1.25
```

For Si, use model B, C, or D with:

```
(model B)MATERIAL TCON.POWER TC.CO=1.55 TC.POW=-1.33
(model C)MATERIAL TCON.POLYN TC.A=0.03 TC.B=1.56e-3 TC.C=1.65e-6
(model D)MATERIAL TCON.RECIP TC.E=320 TC.D=80
```

17.3: Numerical Methods

No special numerical methods are required for thermal simulation. A METHOD statement with no parameters is assumed by default.

17.4: Obtaining Solutions In THERMAL3D

The SOLVE statement in used in THERMAL3D for heat-flow solutions much the same as it is used in other ATLAS simulations involving electrical biases. The temperature in kelvin on each heat sink is used to prescribe the boundary condition temperatures. For example:

```
SOLVE T1=300 T2=500
```

sets the temperature at 300 K and 500 K on heat sink #1 and #2 respectively.

Multiple SOLVE statements are allowed in THERMAL3D. This is useful for obtaining solutions for several combinations of heat sinks and thermal power sources. A range of solutions can also be obtained by stepping the value of a heat sink or power source. For example:

```
SOLVE T1=300 POWER2=0.35 POWER3=0.4 NSTEPS=5 STEPREGION=3 \
POWERFINAL=0.8 OUTFILE=thermal_out0
```

increments the thermal power source in region #3 from 0.4 watts to 0.8 watts in 5 steps and:

```
SOLVE T1=300 POWER2=0.35 NSTEPS=3 ELECTRODE=1 \
TEMPFINAL=600 OUTFILE=thermal_out0
```

increments the temperature on electrode #1 from 300 K to 600 K in 3 steps. Thermal power and temperature can be simultaneously sweep.

If more than one region power is specified during a sweep, the region to be stepped must be specified by STEPREGION=# as shown in the first example above. If the STEPREGION parameter is not specified, the smallest numerical value of POWER# is stepped.

During temperature and power sweeps, the output filename is modified according to the following rule. The rightmost character is incremented using the sequence: 0-9, A-Z, a-z. When a character is incremented beyond z, the character is set to 0> and the character to the left (smaller than ‘z’) is incremented.

Complete syntax information for the SOLVE statement can be found in Chapter 19: “Statements”, Section 19.44: “SOLVE”.

The SOLVE statement is used in THERMAL3D for heat flow solutions much as it is used in other ATLAS simulations for electrical biases. The temperature in Kelvin on each heat sink in the device must be set in the SOLVE statement. The Tn parameter, where n is the number of the heat sink, is used to set the temperature. For example:

```
SOLVE T1=300 T2=500
```

This sets 300K on heat sink #1 and 500K on heat sink #2. Only one SOLVE statement is allowed in any THERMAL3D input file.

17.5: Interpreting The Results From THERMAL3D

The output of thermal simulation consists of the minimum and maximum calculated temperature of each region and its location. The three-dimensional temperature distribution can be saved in an output structure file by setting the OUTFILE parameter in the SOLVE statement and visualized using TONYPLOT3D.

17.6: More Information

Many examples using THERMAL3D have been installed on your distribution tape or CDROM. More information about the use of THERMAL3D can be found by reading the text associated with each example.

You can also find some information at www.silvaco.com.

This page is intentionally left blank.

Chapter 18: Numerical Techniques

18.1: Overview

This chapter describes the overall process of obtaining a numerical solution, the subtasks involved, and the options and defaults available in ATLAS.

You don't need to master this material in order to use ATLAS. Chapter 2: "Getting Started with ATLAS", Section 2.8: "Choosing Numerical Methods" presents the information about numerical techniques that is needed. This chapter provides additional information that will mainly be of interest to advanced users.

18.2: Numerical Solution Procedures

Semiconductor device operation is modeled in ATLAS by a set of anywhere from one to six coupled, non-linear, partial differential equations (PDEs). ATLAS produces numerical solutions of these equations by calculating the values of unknowns on a mesh of points within the device. An internal discretization procedure converts the original, continuous model to a discrete non-linear algebraic system that has approximately the same behavior. The set of PDEs, the mesh and the discretization procedure determine the non-linear algebraic problem that must be solved.

The non-linear algebraic system is solved using an iterative procedure that refines successive estimates of the solution. Iteration continues until the corrections are small enough to satisfy convergence criteria, or until it is clear that the procedure is not going to converge. The non-linear iteration procedure starts from an initial guess. The corrections are calculated by solving linearized versions of the problem. The linear subproblems are solved by using direct techniques or iteratively.

Different solution procedures exhibit different behavior with respect to convergence, accuracy, efficiency, and robustness. The two main aspects of convergence are whether a solution is obtained and how rapidly it is approached. Accuracy is how closely the computed solution approximates the true solution. Efficiency is the time required to produce a solution. Robustness is the ability to converge for a wide range of structures, using meshes and initial guess strategies that are not optimum.

When solving general systems of non-linear equations, there are no guarantees that any particular method will always work. It is also the case that different methods can work better for different problems. Fortunately, there is now a lot of practical experience concerning the numerical techniques that are effective for device simulation. This practical experience has been captured in ATLAS in the form of default methods and parameters that work well in almost all circumstances. This chapter provides advanced information of interest to users who want to change the defaults.

18.3: Meshes

The specification of meshes involves a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine mesh that can resolve all significant features of the solution. Numerical efficiency requires a coarse mesh that minimizes the total number of grid points. This trade-off between accuracy and numerical efficiency is frequently a source of problems for beginners. Fortunately, enough experience to define reasonable meshes is soon acquired.

ATLAS uses triangular meshes. Some triangulations yield much better results than others. Mesh generation is still an inexact science. Guidelines and heuristics, however, for defining satisfactory meshes exist. Good triangulations have the following features:

- They contain enough points to provide the required accuracy.
- They do not contain too many unnecessary points that impair efficiency.
- They avoid, or at least minimize, the number of obtuse triangles. Obtuse triangles tend to impair accuracy, convergence, and robustness.
- They avoid, or at least minimize, the number of long, thin triangles. These triangles also tend to impair accuracy, convergence, and robustness.
- They allow the average size of triangles to change smoothly in transition from a region where very small triangles must be used to a region where the use of much larger triangles is acceptable.

The error associated with a mesh can be investigated systematically by repeating a calculation using a sequence of finer meshes. This is very time consuming and is hardly ever done. The typical approach is to adequately resolve structural features, including doping, with an initial or base mesh, and then add nodes as required to resolve significant features of the solution. The insertion of additional nodes (regridding) is normally done by the program using user-specified criteria.

The initial mesh used by ATLAS can be specified in several ways. It can be inherited from ATHENA. It can be constructed using DEVEDIT. It can be specified using the ATLAS command language. Meshes can be refined using ATLAS commands, or using DEVEDIT. The remainder of this section will focus on the capabilities available using ATLAS commands. The capabilities provided by DEVEDIT are documented in the DEVEDIT USER'S MANUAL.

There are limits on the maximum number of nodes that can be specified. Two-dimensional ATLAS simulations may have up to 20,000 nodes. Three-dimensional simulations may have up to 200,000 nodes, 400,000 elements, with no more than 20,000 in a single plane and a maximum of 200 planes in the z direction. Most devices can be adequately simulated in two dimensions using meshes that contain from several hundred to around 3000 nodes.

18.3.1: Mesh Regridding

The `REGRID` statement supports refinement of regions of the mesh according to specified criteria. Refinement can occur when a specified solution variable exceeds some value, or when the change in that variable across a triangle exceeds a value. The variable can be any of the key quantities in a problem, such as potential, carrier concentration, doping concentration, or electric field.

The regrid algorithm searches the initial grid for triangles that meet the criterion specified for refinement. Each triangle that is identified is divided into four congruent subtriangles. Grid quantities (doping, potential, carrier concentrations, and so forth) are interpolated onto the new nodes, using linear or logarithmic interpolation, as appropriate for that quantity. The initial grid is referred to as being "on level 0" and the new triangles are referred to as "on level 1". After all level 0 triangles have been examined, the same procedure is applied to level 1 triangles, and any subtriangles of level 1 become "level 2" triangles. The grid is checked for consistency at each level and is updated to avoid abrupt changes of size from one triangle to the next. The regrid process continues until no more triangles meet the refinement criteria, or until a specified maximum level of refinement is reached. Grids used in practice are often coarser than is required to meet desirable refinement criteria. Therefore, the maximum level is the key factor in determining the size of the grid after refinement.

The MAX.LEVEL parameter of the REGRID statement is used to limit the amount of refinement at each step. By default, ATLAS sets the maximum level equal to one more than the highest level in the existing mesh. To update a coarse region without regridding the finer regions, after a mesh has already been refined several times, set the maximum level below the level of the finer regions in the existing grid.

If several levels of regrid are performed in immediate succession, interpolated data is used to make the refinement decisions at higher levels. Since semiconductor problems are non-linear, this interpolation may not produce satisfactory results. It is often a good idea to calculate a new solution between regrid operations. In other words, to regrid only one level at a time and obtain a new solution after each regrid operation.

Two popular choices of quantities to be used for mesh refinement are potential and doping. Ideally, variations of electrostatic and quasi-Fermi potentials across an element would be limited to no more than kT/q, and variations in doping would be no more than a factor of 5 or so. In practice, the refinement criteria are often significantly coarser: around 10-20 kT/q for potential, and two to three orders of magnitude for doping. In high level injection situations, it is a good idea to regrid when the value of minority carrier concentration exceeds the local doping concentration.

18.3.2: Mesh Smoothing

Although every step of grid generation can introduce obtuse triangles, two steps in particular can cause problems. The first is that distorting a rectangular mesh introduces a very large number of obtuse elements. The second is that when regridding a rectangular grid that contains triangles with an aspect ratio of 4:1 or greater, very obtuse triangles are created in the transition region between high and low grid density. The REGRID statement allows several procedures to be used when dealing with poorly shaped elements such as obtuse triangles.

The two techniques are node smoothing and triangle smoothing. With node smoothing, several iterative passes are carried out during, which each node is moved to a position and improves the angles of the triangles surrounding it. Node smoothing should only be used for grids that are already irregular. If node smoothing is used for nearly rectangular grids, it may significantly degrade the quality of the mesh.

With triangle smoothing (which is also referred to as diagonal flipping), each adjoining pair of triangles is examined. If appropriate, the diagonal of the quadrilateral is flipped to stabilize the discretization. The diagonal is never flipped when two elements are composed of different materials. When elements are of the same material but have different region numbers, you can specify whether or not to flip the diagonals.

Triangle smoothing is desirable in almost all cases, and should be performed on both the initial grid and on subsequent regrids. The only exception to this rule arises from an undesirable interaction of three elements: regrid, high aspect ratio triangles, and smoothing. This situation frequently occurs in gate oxide regions that involve long, thin triangles. In these cases, smoothing may produce large triangles surrounded by many smaller triangles, giving the appearance of a hole in the mesh. To overcome this, use the smoothing command SMOOTH=4 to limit the formation of the large triangles.

18.4: Discretization

18.4.1: The Discretization Process

The discretization process yields relationships between the variables defined at mesh points [125]. In order to be useful, a discretization must be consistent (i.e., it must limit to the underlying equation in the limit that the mesh size tends to zero). All of the discretizations used in ATLAS have this property. Different discretizations can have different properties with respect to accuracy. The most important measure of accuracy is the order of the scheme (i.e., how errors scaled as the difference between mesh points tends to zero). Discretization schemes used in device simulation are often second order. In other words, as the mesh becomes very fine the discretization error varies as the square of the separation between mesh points.

The discretizations implemented in ATLAS uses the Box Integration Method [167] to approximate differential operators on a general triangular grid. Each equation is integrated over a small polygon which encloses each node. The set of all polygons completely covers the solution domain. The integration procedure equates fluxes into a polygon with sources and sinks inside the polygon, which means that quantities that are conserved physically are also conserved by the numerical solution.

The fluxes must be discretized carefully for the carrier continuity and energy balance equations. Otherwise, nonphysical oscillations and negative carrier concentrations and temperatures may arise. Scharfetter and Gummel [136] introduced approximations for current densities that overcome this problem. Generalizations of this approach are used in ATLAS for the discretization of current densities and energy fluxes.

18.5: Non-Linear Iteration

The non-linear solution method, and associated parameters such as iteration and convergence criteria, are specified in the METHOD statement. Non-linear iteration solution methods are specified in the METHOD statement using the NEWTON, GUMMEL, or BLOCK parameters. Combinations of these parameters may also be specified. In order to understand the effect of these parameters, it is helpful to briefly review how numerical solutions are obtained.

The non-linear algebraic system that results from discretization on a mesh is solved iteratively starting from an initial guess [132,181,8,31,29]. Linearized subproblems are set up and solved. These provide corrections that are used to update the current estimate of the solution. Different sequences of linear subproblems correspond to different non-linear iteration strategies. Iteration continues until convergence criteria are met, in which case the solution is accepted. Iternation also continues until a preset maximum allowable number of iterations is reached, in which case a different technique is tried or the solution procedure is abandoned. When a solution fails to converge, a user normally tries a different grid, a different initial guess strategy, or a different non-linear iteration technique.

18.5.1: Newton Iteration

Each iteration of the Newton method solves a linearized version of the entire non-linear algebraic system. The size of the problem is relatively large, and each iteration takes a relatively long time. The iteration, however, will normally converge quickly (in about three to eight iterations) as long as the initial guess is sufficiently close to the final solution. Strategies that use automatic bias step reduction in the event of non-convergence loosen the requirement of a good initial guess. Newton's method is the default for drift-diffusion calculations in ATLAS. There are several calculations for which ATLAS requires that Newton's method is used. These are DC calculations that involve lumped elements, transient calculations, curve tracing, and when frequency-domain small-signal analysis is performed.

The Newton-Richardson Method is a variant of the Newton iteration that calculates a new version of the coefficient matrix only when slowing convergence demonstrates that this is necessary. An automated Newton-Richardson method is available in ATLAS, and improves performance significantly on most problems. To enable the automated Newton-Richardson Method, specify the AUTONR parameter of the METHOD statement.

If convergence is obtained only after many Newton iterations, the problem is almost certainly poorly defined. The grid may be very poor (i.e., it contains many obtuse or high aspect ratio triangles), a depletion region may have extended into a region defined as an ohmic contact, or the initial guess may be very poor.

18.5.2: Gummel Iteration

Each iteration of Gummel's method solves a sequence of relatively small linear subproblems. The subproblems are obtained by linearizing one equation of the set with respect to its primary solution variable, while holding other variables at their most recently computed values. Solving this linear subsystem provides corrections for one solution variable. One step of Gummel iteration is completed when the procedure has been performed for each independent variable. Gummel iteration typically converges relatively slowly, but the method will often tolerate relatively poor initial guesses. The Gummel algorithm cannot be used with lumped elements or current boundary conditions

Two variants of Gummel's method can improve its performance slightly. These both limit the size of the potential correction that is applied during each Gummel loop. The first method, called damping, truncates corrections that exceed a maximum allowable magnitude. It is used to overcome numerical ringing in the calculated potential when bias steps are large (greater than 1V for room temperature calculations). The maximum allowable magnitude of the potential correction must be carefully specified: too small a value slows convergence, while too large a value can lead to overflow. The DVLIMIT parameter of the METHOD statement is used to specify the maximum allowable magnitude of the potential correction. By default, the value of this parameter is 0.1 V. Thus, by default Gummel

iterations are damped. To specify undamped Gummel iterations, specify DVLIMIT to be negative or zero.

The second method limits the number of linearized Poisson solutions per Gummel iteration, usually to one. This leads to under-relaxation of the potential update. This "single-Poisson" solution mode extends the usefulness of Gummel's method to higher currents. It can be useful for performing low current bipolar simulations, and simulating MOS transistors in the saturation region. It is invoked by specifying the SINGLEPOISSON parameter of the METHOD statement.

18.5.3: Block Iteration

ATLAS offers several block iteration schemes that are very useful when lattice heating or energy balance equations are included. Block iterations involve solving subgroups of equations in various sequences. The subgroups of equations used in ATLAS have been established as a result of numerical experiments that established, which combinations are most effective in practice.

In non-isothermal drift-diffusion simulation, specifying the BLOCK method means that Newton's method is used to update potential and carrier concentrations, after which the heat flow equation is solved in a decoupled step.

When the carrier temperature equations are solved for a constant lattice temperature, the BLOCK iteration algorithm uses Newton's method to update potential and concentrations. The carrier temperature equation is solved simultaneously with the appropriate continuity equation to update the carrier temperature and again carrier concentration.

When both the heat flow equation and the carrier temperature equations are included, the BLOCK scheme proceeds as described previously for the carrier temperature case. It then performs one decoupled solution for lattice temperature as a third step of each iteration.

18.5.4: Combining The Iteration Methods

You can start with the GUMMEL scheme. Then switch to BLOCK or NEWTON if convergence has not occurred within a certain number of iterations. One circumstance where this can be very helpful is that Gummel iteration can refine initial guess to a point from which Newton iteration can converge. The number of initial GUMMEL iterations is limited by GUM.INIT.

You may want to use BLOCK iteration and switch to NEWTON if there is no convergence. This is the recommended strategy for calculations that include lattice heating or energy balance. The number of initial BLOCK iterations is limited by NBLOCKIT.

Any combination of the parameters: GUMMEL, BLOCK, and NEWTON can be specified on the METHOD statement. ATLAS will start with GUMMEL if it is specified. If convergence is not achieved within the specified number of iterations, it will then switch to BLOCK if BLOCK is specified. If convergence is still not achieved the program will then switch to NEWTON.

18.5.5: Solving Linear Subproblems

The linear subproblems generated by non-linear iteration can be solved by direct or iterative methods. Direct methods produce solutions in a predictable number of arithmetic operations. Solutions are affected by roundoff error but are otherwise exact. Iterative methods obtain solutions by making a series of corrections to an initial guess. Iteration proceeds until the calculated corrections satisfy specified convergence criteria. The results are not exact to within roundoff error, and convergence is not guaranteed for general problems. Iterative methods, however, can be more efficient than direct methods for large linear systems, and generally require less memory.

There are mathematical proofs regarding the types of linear system for which iterative techniques converge. All common iterative schemes converge for linear systems that are symmetric positive definite (SPD). The linearized form of Poisson's equation is of this type. The continuity equations are not SPD. But they can be reliably solved by modern, advanced iterative methods.

The size and structure of the coefficient matrix of the linear system plays an important role in the choice of direct or iterative methods. The overall problem size is determined by the number of variables per node (m) and the number of nodes (n). The number of unknowns is m×n. The linear sub problems associated with Newton iteration have a coefficient matrix with $(m \times n)^2$ elements. Each linearized subproblem, which is used in Gummel iteration has a coefficient matrix with n^2 elements.

For practical 2-D device simulation problems, the number of elements in the coefficient matrix is typically between 10^5 and 10^8. Fortunately, the matrices are sparse (i.e., most of the entries are zero, and you do not to store them). The sparsity arises because the variables at each node are coupled to only a few neighboring nodes. Special direct techniques are available for solving sparse matrices.

Direct techniques are preferred for relatively small problems. They provide reliable solutions that are exact to within roundoff error in a time that is predictable. Iterative techniques are preferred for very large problems because they are faster and require less memory. Direct techniques for solving sparse matrices have a competitive performance for the problem sizes typically encountered in 2-D device simulation, and are used by ATLAS to solve most of the linear subproblems that arise.

18.5.6: Convergence Criteria for Non-linear Iterations

After a few non-linear iterations, the errors will generally decrease at a characteristic rate as the iteration proceeds. Non-linear iteration techniques typically converge at a rate that is either linear or quadratic. The error decreases linearly when Gummel iteration is used (i.e., it is reduced by about the same factor at each iteration). For Newton iteration, the convergence is quadratic. In other words, small errors less than one are approximately squared at each iteration. The non-linear iteration is terminated when the errors are acceptably small. The conditions required for termination are called convergence criteria. Much effort has gone into developing reliable default convergence criteria for ATLAS. The default parameters work well for nearly all situations, and most will never need to change them.

The main technique in ATLAS is Gaussian Elimination (or LU Decomposition) with a minimum degree of reordering applied to the nodes [50].

18.5.7: Error Measures

A single positive number that characterizes error is obtained by taking a norm of the errors associated with each unknown. The quantity that ATLAS tries to reduce to zero is the difference between the left and right hand sides of the equation. It is natural to use this quantity as the measure of the error. The associated error norm is called the right hand side (RHS) norm. The units of the RHS norm are C/μm for the Poisson equation, and A/μm for the continuity equations.

Carrier Concentrations and CLIM.DD (CLIMIT)

Another measure of error is provided by the size of the calculated corrections for each unknown. Since the updates are the unknown "xs" at each step, this is called the X norm. Potential updates are measured in units of kT/q. Updates to carrier concentrations are measured relative to the previous value at the point. This relative error (ε_C) is defined as:

$$\varepsilon_C = \max_m \frac{C_m^{K+1} - C_m^K}{\max\left(C_0, C_m^K\right)} \qquad 18\text{-}1$$

where $C = n$ or p, for electrons and holes respectively. m is the node identifier. *C0* is a characteristic concentration. K is the iteration number. C_0 is specified as or as `CLIM.DD` or as *CLIM·c** and

$$\text{CLIM.DD} = \text{CLIMT} \cdot c^* \quad \text{18-2}$$

where:

$$c^* = 4\sqrt{N_c N_v} \quad \text{18-3}$$

CLIM.DD (or CLIMIT) is specified in the METHOD statement.

It is difficult to specify a reasonable default value for the CLIM.DD parameter in all situations. In many difficult cases, the round-off numerical errors do not allow for the resolution of very low concentrations. The default value of CLIMIT is set at 10^4 (the corresponding default value for CLIM.DD in Silicon is $4.5 \cdot 10^{13} cm^{-3}$). In simulation of breakdown, a lower value of CLIM.DD (~$10^8 cm^{-3}$ for Silicon diodes) should be specified. Otherwise, a "false" solution may be obtained.

Discussion of CLIM.EB

To estimate errors in the lattice temperature equation and the energy balance equations, corresponding RHS norms and X norms are calculated in ATLAS. Updates to temperature are measured relative to some characteristic value of temperature. The CLIM.EB parameter can be viewed as a regularization parameter for the case of very small electron or hole densities in the energy balance equations. The CLIM.EB parameter specifies the minimum value of concentration for which the relaxation term in the energy balance equation will be properly resolved. The temperatures for points where the concentration is much less than CLIM.EB are equal to the lattice temperature. The units of CLIM.EB are cm^{-3} and the default is 0.0.

18.5.8: Terminal Current Criteria

Another qualification for convergence is derived from the relative changes in terminal currents and the satisfaction of total current conservation. This qualification can be expressed as the simultaneous satisfaction of the following conditions:

$$\left| I_i^{K+1} - I_i^K \right| \leq E_1 \left| I_i^{K+1} \right| + E_2 \quad \text{18-4}$$

and

$$\left| \Sigma I_i^{K+1} \right| < 0.01 \, maxi \left(\left| I_i^{K+1} \right|, E_2 \right)$$

$$I = 1, nc \quad \text{18-5}$$

where

- I_i is the current through contact i
- K is the iteration number
- E_1 and E_2 are specified tolerances
- nc is the number of contacts

18.5.9: Convergence Criteria

A summary of the termination criteria that is enough for most purposes will now be given. Detailed reference information is provided in the next section.

The non-linear iteration is terminated when one of the following four criteria is satisfied.

1. The X norm for every equation falls below a specified tolerance. The specified tolerances for X norms are:

   - P_{tol}^{x} for potential equation.
   - C_{tol}^{x} for concentration equations.
   - TL_{tol}^{x} for lattice temperature equation.
   - TC_{tol}^{x} for carrier temperature equation.

2. The RHS for all equations and X norms for energy balance falls below a specified tolerance. The specific tolerances are:

   - P_{tol}^{r} for potential equations.
   - C_{tol}^{r} for concentration equations.
   - TL_{tol}^{r} for lattice temperature equations.
   - TC_{tol}^{r} for carrier temperature equations.

3. For every equation either the X norm or the RHS norm falls below a specified tolerance. In this case, both the XNORM and RHSNORM parameters must be specified true.
4. If either 1 or 2 or 3 criterion is fulfilled for weaker values of tolerances. In other words, for specified tolerances multiplied by the W parameter and current criteria 18-4 and 18-5 are satisfied.

To exclude the X-norm criterion, specify ^XNORM in the METHOD statement. To exclude RHS-norm criterion, specify ^RHSNORM. To exclude the current criterion, make E1 and E2 very small. You can change all the above mentioned tolerances simultaneously by specifying the relaxation factor TOL.RELAX in the METHOD statement.

Table 18-1. User-Specifiable Parameters for Convergence Criteria

Symbol	Statement	Parameter	Default
P_{tol}^{x}	METHOD	PX.TOL	10^{-5}
C_{tol}^{x}	METHOD	CX.TOL	10^{-5}
TL_{tol}^{x}	METHOD	TLX.TOL	10^{-5}
TC_{tol}^{x}	METHOD	TCX.TOL	10^{-5}

Table 18-1. User-Specifiable Parameters for Convergence Criteria			
Symbol	**Statement**	**Parameter**	**Default**
P_{tol}^{r}	METHOD	PR.TOL	5.0×10^{-26}
C_{tol}^{r}	METHOD	CR.TOL	5.0×10^{-18}
TL_{tol}^{r}	METHOD	TLR.TOL	100
TC_{tol}^{r}	METHOD	TCR.TOL	100
E_1	METHOD	IX.TOL	2.0×10^{-5}
E_2	METHOD	IR.TOL	5.0×10^{-15}
W	METHOD	WEAK	200
TOL.RELAX	METHOD	TOL.RELAX	1
XNORM	METHOD	XNORM	TRUE
RHSNORM	METHOD	RHSNORM	TRUE
CLIMIT	METHOD	CLIMIT	10^4
CLIM.DD	METHOD	C_0	4.5×10^{13}
CLIM.EB	METHOD	CLIM.EB	0

18.5.10: Detailed Convergence Criteria

Only in very difficult situations is more detailed information concerning error estimation and the specification of convergence criteria needed. The material is organized by algorithm.

Convergence Criteria For Gummel's Algorithms

Relative update errors are defined as follows.

For potential:

$$\varepsilon_v = \frac{\max\limits_m \left|\varphi_m^{K+1} - \varphi_m^{K}\right|}{max\left(1,\ \varphi_{mmax}^{K+1}\right)}, \qquad 18\text{-}6$$

where *mmax* is the node where $\left|\varphi_m^{K+1} - \varphi_m^{K}\right|$ has its maximum value.

For electrons:

$$\varepsilon_n = \frac{\max_m \left| n_m^{K+1} - n_m^K \right|}{max\left(C_0, n_m^K\right)} \quad 18\text{-}7$$

For holes:

$$\varepsilon_p = \frac{\max_m \left| p_m^{K+1} - p_m^K \right|}{max\left(C_0, P_m^K\right)} \quad 18\text{-}8$$

For lattice temperature:

$$\varepsilon_{T_L} = \frac{(T_L)_{nmax}^{K+1} - (T_L)_{nmax}^K}{(T_L)_{nmax}^{K+1}} \quad 18\text{-}9$$

where *nmax* is the node where $(T_L)_i^{K+1}$ has its maximum value.

For carrier temperature:

$$\varepsilon_{T_c} = max\left(\varepsilon_{T_n}, \varepsilon_{T_p}\right) \quad 18\text{-}10$$

where

$$\varepsilon_{T_n} = \frac{\max_m \left| (T_n)_m^{K+1} - (T_n)_m^K \right|}{\max_m (T_n)_m^{K+1}} \quad 18\text{-}11$$

$$\varepsilon_{T_p} = \frac{\max_m \left| (T_p)_m^{K+1} - (T_p)_m^K \right|}{\max_m (T_p)_m^{K+1}} \quad 18\text{-}12$$

For drift diffusion, iterations are terminated if the following criteria are satisfied:

$$\varepsilon_v =\leq P_{tol}^x \quad 18\text{-}13$$

$$\varepsilon_n =\leq c_{tol}^x \quad 18\text{-}14$$

$$\varepsilon_v =\leq C_{tol}^x \quad 18\text{-}15$$

In non-isothermal drift diffusion, iterations are terminated if Equations 18-10 to 18-12 are satisfied, the current convergence criteria in Equations 18-4 and 18-5 are met and:

$$\varepsilon_{T_L} < TL_{tol}^{x} \quad \text{18-16}$$

In Gummel's method with energy balance equations, NITGUMM iterations, which only the non-linear Poisson equation is solved, will always be done. Gummel's method with energy balance equations is terminated if Equation 18-16 is fulfilled, the carrier temperature convergence criteria

$$\varepsilon_{T_c} \leq TC_{tol}^{x} \quad \text{18-17}$$

is achieved, and one of the following conditions is valid:

- NITGUMM < K+1 ≤ NITGUMM+NT1 and Equation 18-13 is fulfilled,
- NT1 + NITGUMM < K+1 ≤ NITGUMM + NT3 and

$$\varepsilon_n \leq P_{tol}^{x} \cdot w \quad m \quad \text{18-18}$$

where w = 10

- NT1 + NITGUMM < K + 1 ≤ NITGUMM + NT3, the current convergence criteria (Equations 18-4 and 18-5) are satisfied, and inequality (Equation 18-18) is valid for w = 100,
- NT1 + NITGUMM < K+1 and (Equation 18-18) is valid for w = 100,
- NT1 + NITGUMM < K+1 the current convergence criteria (Equations 18-4 and 18-5) are satisfied, and (Equation 18-18) is valid for w = 500.

The default values of the iteration parameters are NITGUMM = 5, NT0 = 4, NT1 = 10, and NT3 = 100

Convergence Criteria For Newton's Algorithm

Relative update errors are defined as follows.

For potential

$$\varepsilon_v = \max_m \left| \varphi_m^{K+1} - \varphi_m^{K} \right| \quad \text{18-19}$$

For electron concentration:

$$\varepsilon_n = \frac{\max_m \left| n_m^{K+1} - n_m^{K} \right|}{\max\left(C_0, n_m^{K} \right)} \quad \text{18-20}$$

For hole concentration:

$$\varepsilon_p = \frac{\max_m \left| p_m^{K+1} - p_m^{K} \right|}{\max\left(C_0, p_m^{K} \right)} \quad \text{18-21}$$

For lattice temperature and carrier temperature:

$$\varepsilon_{T_L} = \frac{\max_m \left| (T_L)_m^{K+1} - (T_L)_m^K \right|}{T_{scale}}, \quad 18\text{-}22$$

$$\varepsilon_{T_n} = \frac{\max_m \left| (T_n)_m^{K+1} - (T_n)_m^K \right|}{T_{scale}}, \quad 18\text{-}23$$

$$\varepsilon_{T_p} = \frac{\max_m \left| (T_p)_m^{K+1} - (T_p)_m^K \right|}{T_{scale}}, \quad 18\text{-}24$$

where the scaling temperature, *Tscale*, is by default equal to 300K.

To define the RHS norms used in ATLAS, first represent the non-linear equations obtained after discretization at every node as

$$(\ )\ \overrightarrow{(\chi)} = 0 \quad 18\text{-}25$$

where α can be ψ, n, p, T_L, T_n and T_p for the potential equation, electron continuity equation, hole continuity equation, lattice temperature equation, electron temperature equation and hole temperature equation, respectively. $\vec{\chi}$ represents the vector of unknowns.

The RHS norms in ATLAS is then defined as follows:

For the potential equation:

$$\varepsilon_v^g = \max_m |F\varphi| \quad 18\text{-}26$$

For the electron and continuity equations:

$$\varepsilon_n^g = \max_m \left| F_n \right| \cdot C_T \quad 18\text{-}27$$

$$\varepsilon_p^g = \max_m \left| F_p \right| \cdot C_T \quad 18\text{-}28$$

where $CT = 10\text{-}4\ 4\sqrt{N_c N_v}\quad kT$

For the lattice temperature and carrier temperature equations:

$$\varepsilon_{T_L}^g = \max_m \left| F_{T_L} \right| \quad 18\text{-}29$$

$$\varepsilon_{T_n}^g = \max_m \left| F_{T_n} \right| \cdot \quad 18\text{-}30$$

$$\varepsilon_{T_p}^g = \max_m \left| F_{T_p} \right| \quad 18\text{-}31$$

Newton iterations are terminated if one of the following criteria is satisfied.

The XNORM parameter is true and:

$$\varepsilon_v \le P_{tol}^x \cdot w, \quad \text{18-32}$$

$$\varepsilon_n \le c_{tol}^x \cdot w, \quad \text{18-33}$$

$$\varepsilon_p \le P_{tol}^x \cdot w, \quad \text{18-34}$$

$$\varepsilon_{T_L} \le TL_{tol}^x \cdot w, \quad \text{18-35}$$

$$\varepsilon_{T_n} \le TC_{tol}^x \cdot w, \quad \text{18-36}$$

$$\varepsilon_{T_p} \le TL_{tol}^x \cdot w, \quad \text{18-37}$$

where $w = 1$.

The RHSNORM parameter is true, conditions for Equations 18-36 and 18-37 are satisfied for $w = 1$ and:

$$\varepsilon_v^g \le P_{tol}^r \cdot w_1, \quad \text{18-38}$$

$$\varepsilon_n^g \le C_{tol}^r \cdot w_1, \quad \text{18-39}$$

$$\varepsilon_p^g \le P_{tol}^r \cdot w_1, \quad \text{18-40}$$

$$\varepsilon_{T_L}^g \le TL_{tol}^r \cdot w_1, \quad \text{18-41}$$

$$\varepsilon_{T_n}^g \le TC_{tol}^r \cdot w_1, \quad \text{18-42}$$

$$\varepsilon_{T_p}^g \le TL_{tol}^r \cdot w_1, \quad \text{18-43}$$

Both the XNORM and RHSNORM parameters are true. The convergence criteria for Equations 18-36 and 18-37 for carrier temperature are fulfilled for the inequalities (Equations 18-32 and 18-38), (Equations 18-33 and 18-39), (Equations 18-34 and 18-40), (Equations 18-35 and 18-41), and one of the conditions for every pair.

If the current convergence criteria are satisfied, then condition a) or condition b) or condition c) is fulfilled for `w = w1 = WEAK`.

Convergence Criteria For Block Iteration

For the potential equation and the continuity equations, Equations 18-19 to 18-21 and Equations 18-26 to 18-28 define the X norms and RHS norms for the Newton method. With *s* as an index that denotes the number of block iterations, update errors between successive pairs of block iterations for lattice and carrier temperature are defined by the following expressions.

For lattice temperature:

$$\varepsilon_{T_L}^{B} = \frac{\left|(T_L)_{mmax}^{s+1} - (T_L)_{mmax}^{s}\right|}{(T_L)_{mmax}^{s+1}} \quad 18\text{-}44$$

where *mmax* is the number of the node where $(TL)_m$ has its maximum value.

For electron temperature:

$$\varepsilon_{T_n}^{B} = \frac{\left|(T_n)_{mmax}^{s+1} - (T_n)_{mmax}^{s}\right|}{(T_n)_{mmax}^{s+1}} \quad 18\text{-}45$$

For hole temperature:

$$\varepsilon_{T_p}^{B} = \frac{\left|(T_p)_{mmax}^{s+1} - (T_p)_{mmax}^{s}\right|}{(T_p)_{mmax}^{s+1}} \quad 18\text{-}46$$

Block iterations are terminated if:

$$\varepsilon_{T_L}^{B} \le TL_{tol}^{x} \quad 18\text{-}47$$

$$\varepsilon_{T_n}^{B} \le TC_{tol}^{x} \quad 18\text{-}48$$

$$\varepsilon_{T_p}^{B} \le TL_{tol}^{x} \quad 18\text{-}49$$

and one of the following criteria is fulfilled:

- the XNORM parameter is true and Equations 18-32, 18-33, and 18-34 are valid for $w = 1$.
- the RHSNORM parameter is true and Equations 18-38, 18-39, and 18-40 are valid for $w_1 = 1$.
- both XNORM and RHSNORM are true and for the pairs of inequalities (Equations 18-32 and 18-38), (Equations 18-33 and 18-39), (Equations 18-34 and 18-40) one of the conditions is satisfied for each pair.
- the current convergence criteria are satisfied and condition 1, condition 2, or condition 3 is fulfilled for `w = w1 = WEAK`.

18.6: Initial Guess Strategies

Non-linear iteration starts from an initial guess. The quality of the initial guess (i.e., how close it is to the final solution) affects how quickly the solution is obtained and whether convergence is achieved. ATLAS users aren't required to specify an initial guess strategy. If no strategy is defined, ATLAS follows certain rules that implement a sensible, although not necessarily optimum strategy.

There is some interaction between the choice of non-linear iteration scheme and the initial guess strategy. Decoupled iteration usually converges linearly, although perhaps slowly, even from a relatively poor initial guess. Newton iteration converges much faster for a good initial guess, but fails to converge if started from a poor initial guess.

One very simple initial guess strategy is to use the most recent solution as the initial guess. Of course, there is no previous solution for the first calculation in a series of bias points. In this case, an initial solution is obtained for equilibrium conditions. There is no need to solve the current continuity equations at equilibrium, and a solution of Poisson's equation is quickly obtained.

You can also modify the initial guess in a way that makes some allowance for the new bias conditions. Typical strategies include:

- Using two previous solutions and interpolation to project a new solution at each mesh point.
- Solving a form of current continuity equation with carrier concentrations held constant. This strategy yields an improved estimate of new potential distribution.
- Modifying the majority carrier quasi-Fermi levels by the same amount as the bias changes.

You can use the parameters on the SOLVE statement to specify an initial guess strategy. Six initial guess strategies are available.

- INITIAL starts from space charge neutrality throughout the device. This choice is normally used to calculate a solution with zero applied bias.
- PREVIOUS uses the currently loaded solution as the initial guess at the next bias point. The solution is modified by setting a different applied bias at the contacts.
- PROJECTION takes two previous solutions whose bias conditions differ at one contact and extrapolates a solution for a new applied bias at that contact. This method is often used when performing a voltage ramp.
- LOCAL sets the applied bias to the specified values, and changes the majority carrier quasi-Fermi levels in heavily doped regions to be equal to the bias applied to that region. This choice is effective with Gummel iteration, particularly in reverse bias. It is less effective with Newton iteration.
- MLOCAL starts from the currently loaded solution and solves a form of the total current continuity equation that provides an improved estimate of the new potential distribution. All other quantities remain unchanged. MLOCAL is more effective than LOCAL because it provides a smooth potential distribution in the vicinity of p-n junctions. It is usually more effective than PREVIOUS because MLOCAL provides a better estimate of potential. This is especially true for highly doped contact regions and resistor-like structures.
- NOCURRENT assumes there is negligible current flow in the device and solves the non-linear Poisson equation to arrive at the initial guess. If the assumption of negligible current density is indeed correct, then the initial guess should be the solution to the current continuity equations too. You can use NOCURRENT to avoid ramping a bias to attain the solution at high bias.

When a regrid is performed, the solution is interpolated from the original grid onto a finer grid. This provides an initial guess that can be used to start the solution of the same bias point on the new grid. Although the initial guess is an interpolation of an exact solution, this type of guess does not provide particularly fast convergence.

18.6.1: Recommendations And Defaults

It is not normally required to specify any initial guess parameter.

The INITIAL parameter is normally used only to obtain a thermal equilibrium solution. The PREVIOUS, PROJECT, LOCAL, and MLOCAL parameters are used for other bias points. PREVIOUS and PROJECTION are the most frequently used. PROJECTION is normally preferred to PREVIOUS when it is available (i.e., when there are two previous solutions differing in the bias applied to the appropriate terminal). PREVIOUS is required for transient simulations, and for simulations that use current boundary conditions. LOCAL and MLOCAL tend to work well for reverse-biased devices, and are especially efficient when trying to increase very large voltage increments. By default, PROJECTION is used whenever two appropriate solutions are available. Otherwise, the PREVIOUS guess is used, unless there is no previous solution, in which case INITIAL is used.

18.7: The DC Curve-Tracer Algorithm

Tracing I-V curves for complicated device phenomena, such as breakdown or latchup, can be very difficult using conventional methods. ATLAS includes a special purpose DC curve-tracing algorithm that overcomes these problems. This algorithm is based on a dynamic load-line technique that adapts the boundary conditions for each step. The approach implemented in ATLAS for curve tracing is based on the work describe in [60].

The key idea is that bias conditions can evolve smoothly between the limits of pure voltage control and pure current control. This is achieved using external resistors that adapt dynamically to the shape of the I-V curves to ensure that at each point the load line is perpendicular to the local tangent of the trace. With this value for the external resistor, the solution is projected to the next operating point by stepping the external voltage. Once the solution has converged, a new external resistance is calculated based on the new tangent information and the process repeats itself.

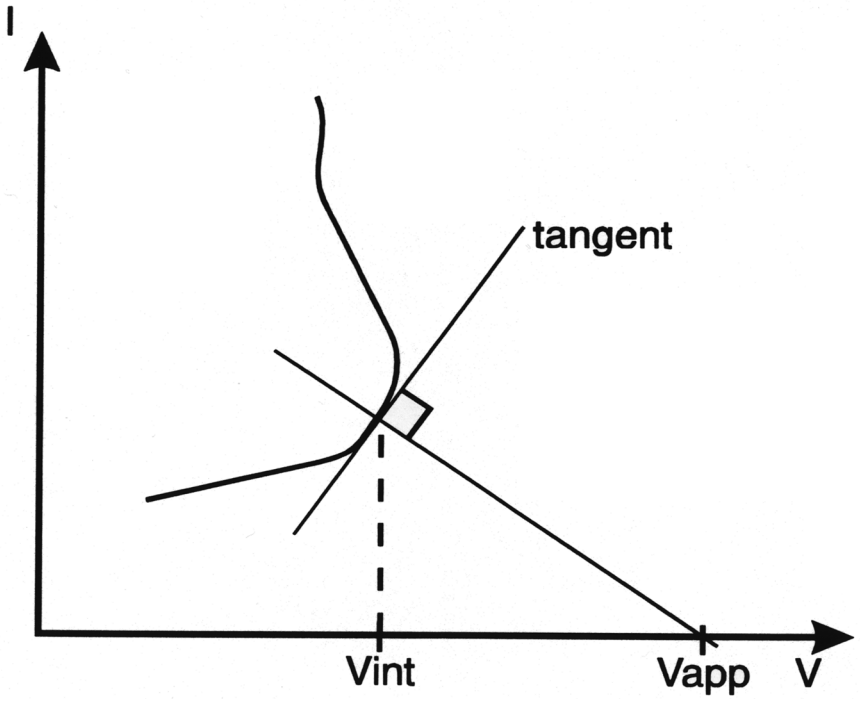


Figure 18-1: Load the algorithm used in the Curve Tracer

The curve tracing capability is activated by specifying the CURVETRACE parameter in the SOLVE statement. Prior to this, the CURVETRACE statement is used to set the parameters for the tracing algorithm. These parameters are the name CONTR.NAME of the ramped electrode (which will be referred to as a control electrode), the initial voltage increment STEP.INIT, the upper bound of the tracing curve, and additional parameters if they differ from the default values. The Upper bound parameter, END.VAL, is used to stop tracing. If the VOLT.CONT parameter is specified, END.VAL is a voltage. If the CURR.CONT parameter is specified, END.VAL is a current.

The applied voltage at each step is altered in accordance with the slope of the I-V curve. The resistor between the applied voltage and the semiconductor is also changed dynamically to ensure the voltage at the semiconductor (VINT) is smoothly varied along the I-V curve. If STEP.CONT is specified, the number of operational points on a trace will not exceed specified parameter STEPS.

18.8: Transient Simulation

When transient simulation is performed, the carrier continuity equations are integrated in the time domain. Time integration schemes differ in their accuracy, in the number of previous time levels they employ, and in their stability properties.

Accuracy is usually referred to as being "nth order", where n is usually an integer between 1 and 4. In the limit of a small timestep, the magnitude of the local truncation error (LTE) introduced by the time integration scheme is proportional to the nth power of the timestep. Schemes that require the storage of solutions at timesteps previous to the most recent one are unattractive due to storage requirements. Single step integration schemes that use the solution at only one previous time level have a maximum order of 2.

The continuity equations are "stiff" (i.e., they are impacted by phenomena with very short timescales). Stiffness imposes stringent requirements on the stability of time integration schemes. Two forms of stability, A-stability and L-stability, are important. A-stability guarantees that errors introduced at one time step will not increase at the next timestep. L-stability guarantees that errors will decay even for large time step values. A-stability is a requirement for any practical scheme. L-stability is extremely desirable to avoid non-physical "ringing".

Most device simulation codes use a simple first-order (implicit) backward difference formula for time integration [57,45,93]. This scheme, which is known as BDF1, is both A-stable and L-stable. Unfortunately, the scheme is inaccurate for typical timesteps of interest. Second order accuracy is obtained using the trapezoidal rule (TR) for time integration. This scheme is A-stable, but it is not L-stable. This means that solutions exhibit non-physical "ringing", unless very small timesteps (much smaller than those dictated by LTE considerations) are used. The BDF2 scheme is second order, and is both A-stable and L-stable. The scheme, however, uses solutions from two previous time levels and is less accurate than TR.

For drift-diffusion calculations, ATLAS uses a composite TR-BDF2 scheme that was developed by Bank et. al.[18]. This method is one-step, second order, and both A-stable and L-stable. An estimate of the LTE is obtained at each timestep. This estimate is also used to automatically adapt the timestep.

Different schemes are used for transient solutions that include lattice heating or energy balance. If lattice heating is included, the block iterative procedure is organized at each time step in the same way as for the steady state case. If energy balance is selected, the absolutely stable half-implicit scheme [124] is used. Automatic timestep selection with local error control is implemented in this case. You can specify tolerance using the `TOL.TIME` parameter in the `METHOD` statement.

Note: You normally specify only the initial timestep with the `TSTEP` parameter of the `SOLVE` statement. After this, time steps are derived from the LTE and will typically increase.

18.9: Small Signal and Large Signal Analysis

There are several ways to predict the small-signal and large-signal high-frequency properties of semiconductor devices (review of these different techniques was presented by Laux [94]). Frequency domain perturbation analysis can be used to determine the small-signal characteristics. Fourier analysis of transient responses (FATR), however, can be used for both small-signal and large signal response.

You can also use charge partitioning models [76]. These methods, however, have been dropped because they can only be applied to insulated contacts, where there is only displacement current and are strictly quasi-static (low frequency).

18.9.1: Frequency Domain Perturbation Analysis

Frequency-domain perturbation analysis of a DC solution can be used to calculate small-signal characteristics at any user-specified frequency [94]. This scheme works for both 2D and 3D. The calculation proceeds in the following manner:

1. Variables are represented as the sum of the known DC component and a small unknown sinusoidal AC component.
2. All equations are expanded.
3. Differentiation in time becomes multiplication by the value of ω ($\omega = 2\pi$ frequency).
4. Products of AC quantities are neglected since they are small with respect to other quantities
5. The DC solution is subtracted.

What remains is a complex linear system whose unknowns are the AC components of the solution. Solving this linear system with appropriate boundary conditions yields small-signal characteristics.

The coefficient matrix of the complex linear system is simply the Jacobian associated with the DC operating point with some terms on the leading diagonal supplemented by $j\omega$. The Jacobian is available 'for free' if the DC solution was calculated using Newton iteration. This is a very attractive feature of the Newton method. The resulting linear complex system can be solved by an iterative or a direct method. A form of successive over relaxation (SOR) works well for frequencies significantly below the cutoff frequency but fails at higher frequencies. More sophisticated iterative techniques can be used at higher frequencies. It is simpler and more reliable, however, to switch to direct sparse matrix methods if SOR fails to converge.

Frequency domain perturbation analysis is extremely attractive when the full Newton method is used to calculate a DC solution. The method works for all frequencies. It requires a predictable amount of computation, which is quite low with respect to other calculations, since only a single linear system is solved.

Frequency domain perturbation analysis is invoked in ATLAS by specifying the appropriate parameters in the SOLVE statement. The AC.ANALYSIS parameter specifies that this analysis is to be performed. TERMINAL specifies the contact whose terminal voltage is to be perturbed. A full characterization requires that all but one of the device terminals is perturbed. FREQUENCY, FSTEP, and NFSTEPS determine the frequencies for which solutions are obtained. FREQUENCY specifies the initial frequency. FSTEP and NFSTEPS define a loop on frequencies. If MULT.FREQ is specified, the frequency is multiplied by FSTEP at each increment. This is useful for characterizing the small-signal response over several decades of frequency. The solution method can be specified using the SOR, DIRECT, and AUTO parameters. AUTO starts out using SOR but switches to DIRECT if convergence problems are encountered.

18.9.2: Fourier Analysis Of Transient Responses

FATR is a post-processing step that must be performed on a LOG file, which contains transient data. The FOURIER statement performs a Fast Fourier Transform (FFT) on the time domain data transforming it into the frequency domain. Table 18-2 shows the syntax of the FOURIER statement. For more information about the FOURIER statement, see Chapter 19: "Statements", Section 19.14: "FOURIER".

Table 18-2. User-Specifiable Parameters for Fourier Statement

Statement	Parameter	Default
FOURIER	INFILE	
FOURIER	OUTFILE	
FOURIER	T.START	
FOURIER	T.STOP	
FOURIER	FUNDAMENTAL	
FOURIER	MAX.HARMONIC	
FOURIER	NUM.SAMPLES	64
FOURIER	INTERPOLATE	FALSE
FOURIER	COMPLEX.VALUES	FALSE

The explanation for the FOURIER parameters are as follows:

- INFILE – input log file. This should contain data from a transient simulation.
- OUTFILE – file output file for the Fourier transform.

The following parameters are optional:

- T.START – start of time data to be used for the FFT. The default value is the first time point in the input log file.
- T.STOP – end of time data to be used for the FFT. The default value is the last time point in the input log file.
- FUNDAMENTAL – fundamental frequency. If this is not specified, then the fundamental frequency is set to 1/(T.STOP - T.START). If the fundamental frequency is specified then T.STOP is set to T.START + 1/FUNDAMENTAL.
- MAX.HARMONIC – maximum harmonic frequency that the FFT should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.
- NUM.SAMPLES – number of samples. This should be an integer power of 2 (i.e., 2^n) where n is a positive integer. The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case, the number of samples is set to the nearest integer power of 2, which will generate this frequency.
- INTERPOLATE – performs linear interpolation on input data with non-uniform timesteps. This interpolates the data on to uniform timesteps. Interpolation of data can introduce addition (inaccurate) harmonic values into the FFT, which would not occur if uniform time is taken. INTERPOLATE must be used if the log file contains non-uniform time steps.
- COMPLEX.VALUES – prints the real and imaginary components to file as well as the magnitude and phase.

Note: FFT works best with uniform time steps. Therefore, set DT.MIN and DT.MAX on the METHOD statement to the same value. The time step should be set to: time step = 1/(number of samples * fundamental frequency).

The FFT can then calculate harmonic frequencies up to:

```
(number of samples/2 - 1)*fundamental frequency.
```

You can obtain small-signal data at high frequencies by calculating the terminal current responses to terminal voltage perturbations. Then, Fourier analysis will be performed on currents and voltages. Their ratio at each frequency provides admittance data for that frequency. The voltage perturbations are normally selected to have an analytic form with a known Fourier transform. Care must be taken to self-consistently account for geometric capacitances when step function voltage perturbations are used. The advantages of this technique are that it can be used whenever transient calculations are possible. Each transient solution gives information over a broad range of frequencies. The main disadvantage is that transients become very long when low frequency effects are investigated.

18.9.3: Overall Recommendations

Use frequency-domain perturbation analysis when it is available. The method works for arbitrary frequencies and does not require transient calculations. Use Fourier analysis of transient responses for high frequencies when it is unavailable but transient calculations are possible.

18.10: Differences Between 2D and 3D Numerics

With respect to numerical techniques, there are several differences between 2D and 3D simulations.

First, with respect to the nonlinear iteration strategies, all three strategies, NEWTON, GUMMEL and BLOCK are supported in 2D simulation, whereas, only NEWTON and GUMMEL are supported for 3D simulations. Implementation of the BLOCK strategy is expected in a future release.

Second, solution of the linear subproblem is handled differently for 2D and 3D simulations. As previously noted, the computational burden of solving the linear subproblem increases with the size of the solution domain. For smaller problems, direct methods are quicker while for larger problems iterative methods are preferred. It turns out that the point at which the iterative methods become less burdensome roughly coincides with the transition between 2D and 3D domains. As such, the default method for 2D simulations is a direct solver. For 3D simulations, the default method is an iterative solver. By default, ILUCGS is applied to 3D simulations. ILUCGS is an acronym for incomplete lower upper decomposition conjugate gradient squared. Two alternative iterative solvers are also available for 3D simulations. BICGST (BICGST on the METHOD statement) is an acronym for biconjugate gradient squared stabilized. GMRES (GMRES on the METHOD statement) is an acronym for generalized minimum residual. Direct methods can be used for 3D simulation by specifying DIRECT on the METHOD statement. Practical experience shows that for 3D simulations either of the iterative methods are faster than the direct method. But in some cases, the accuracy produced by the iterative methods can prevent convergence in the nonlinear outer loop. For 2D simulations, only direct methods are supported.

Chapter 19: Statements

19.1: Input Language

This chapter contains a complete description (in alphabetical order) of every statement and parameter used by any of the ATLAS products, except for MIXEDMODE. MIXEDMODE specific parameters are described in Chapter 12: "MixedMode: Mixed Circuit and Device Simulator". Descriptions for the statements in this chapter are as follows:

- The statement name.
- The product for which the statement is applicable.
- The syntax of the statement.
- A list of all statement parameters, their type, default value, and units.
- A description of each parameter.
- An example of the correct usage of each statement.

Note: An error message will be generated if you attempt to specify a statement for a simulator that you haven't purchased. For example, the BEAM statement can only be used if you have purchased LUMINOUS or LUMINOUS 3D.

19.1.1: Syntax Rules

An input deck line is referred to as a statement (or statement line). Since statements and parameters are not case sensitive, they can be entered using either uppercase or lowercase letters.

A statement is specified in the general format:

```
<STATEMENT> <PARAMETER>=<VALUE>
```

where STATEMENT is the statement name, PARAMETER is the parameter name, and VALUE is the parameter value. The space character is used to separate the parameters in a statement.

The words and numbers, which follow a statement, are parameters of that statement. A word is an alphanumeric string, which is terminated either by a space or by a carriage return. A number is a numeric or alphanumeric string which is terminated either by a space or by a carriage return. Numerical values may range from 10^{-38} to 10^{38}. A number may contain the symbols + (positive), - (negative), and/or E (decimal notation). For example:

```
+10 -1.234E5 .003 -.12E+10
```

Four types of parameters are used by the ATLAS products. These are: real, integer, logical, and character. Table 19-1 explains these parameter types.

Table 19-1. Types of Parameters

Parameter	Description	Value Required	Example
Character	Any character string	Yes	INFILE=NMOS.DOP
Integer	Any whole number	Yes	REGION=2
Logical	A true or false condition	No	GAUSSIAN
Real	Any real number	Yes	X.MIN=0.52

Any parameter that doesn't have a logical value must be specified in the form: PARAM=VAL, where PARAM is the name of the parameter, and VAL is the value of the parameter. Logical parameters must be separated from other parameters or commands by a space.

For example, in the statement:

```
DOPING UNIFORM CONCENTRATION=1E16 P.TYPE
```

the UNIFORM and P.TYPE parameters have logical values and the CONCENTRATION parameter has a value of $1x10^{16}$ (real).

Logical parameters can be turned off (switched from true to false) by placing a caret (^) in front of the logical parameter. For example, in the statement:

```
DOPING UNIFORM CONCENTRATION=1E16 ^P.TYPE
```

the P.TYPE parameter has been set to false.

Mnemonics

It is not always necessary to input the entire statement or parameter name. ATLAS only requires that you input enough letters to distinguish that command or parameter from other commands or parameters. For example, DOP may be used to abbreviate the DOPING command. Excessive truncation is not recommended, since future ATLAS syntax might make short abbreviations become ambiguous.

Continuation Lines

Since it may be necessary for a statement line to contain more than 256 characters, ATLAS allows you to specify continuation lines. To continue a line, put a backslash (\) character at the end of the line that is to be continued. When ATLAS encounters the backslash, it will interpret the next line to be a continuation of the current line. The PISCES-II continuation of using a (+) at the start of the subsequent line is not supported in ATLAS.

Comments

Comments are indicated either by COMMENT command, or by a pound sign (#). All characters on a line, which follow a comment indicator (COMMENT or #) will not be analyzed by ATLAS.

Synonyms

Some parameters have synonyms. These are parameters that have a different name but the same functionality. A parameter's synonym is listed in the parameter descriptions of the statements.

Pseudonyms

Throughout the statement descriptions, pseudonyms are used either to indicate a group of parameters or to indicate the value of a particular parameter. A < symbol indicates the start of a pseudonym(s). A > symbol indicates the end of a pseudonym(s). Pseudonyms will be separated from one another by a space character ( ). For example, LOC might indicate a group of location parameters, and FILENAME might indicate the name of a file that must be specified.

Symbols

The following symbols are used in the statement descriptions:

- < Indicates the start of a list of pseudonyms.
- > Indicates the end of a list of pseudonyms.
- | Separates parameters or pseudonyms which are mutually exclusive. Only one of these parameters may be used in a statement.
- [Indicates the start of an optional command, parameter, or pseudonym.

-] Indicates the end of an optional command, parameter, or pseudonym.

Expressions

ATLAS does not support arithmetic expressions in the syntax. You can, however, evaluate and use expressions by using the SET or EXTRACT statements.

19.2: BEAM

BEAM specifies an optical input signal in the form of a collimated beam of light. This statement is used with LUMINOUS or LUMINOUS 3D.

Syntax

```
BEAM <parameters>
```

Parameter	Type	Default	Units
AMBIENT.INDEX	Real	1.0	
ANGLE	Real	0.0	Degrees
BACK.REFL	Logical	False	
BPM	Logical	False	
CIRCULAR	Logical	False	
DEVICE	Character		
ELLIPTICAL	Logical	False	
F.RADIATE	Character		
F3.RADIATE	Character		
F.REFLECT	Character		
FRONT.REFL	Logical	False	
GAUSSIAN	Logical	False	
INPUTRAYS	Character		
INTEGRATE	Logical	True	
ITERATION	Integer	20	
LENS.HEIGHT	Real	0.0	microns
LENS.INDEX	Real	1.0	
LENS.PLANE	Real	0.0	μm
LENS.RADIUS	Real	0.0	μm
LENS.WIDTH	Real		microns
LENS.X	Real	0.0	μm
LENS.XMAX	Real	0.0	microns
LENS.XMIN	Real	0.0	microns
LENS.XSAGS	Character		
LENS.Y	Real	0.0	μm
LENS.Z	Real	0.0	μm

Parameter	Type	Default	Units
LENS.ZMAX	Real	0.0	microns
LENS.ZMIN	Real	0.0	microns
LENS.ZSAGS	Character		
LONGIT.STEP	Real	0.0	
MAX.WINDOW	Real	1.0×10^{20}	μm
MEAN	Character	0.0	μm
METAL.REFLECT	Logical	False	
MIN.POWER	Real	0.0	W/cm^2
MIN.WINDOW	Real	-1.0×10^{20}	μm
NORMALIZE	Logical	False	
NSAMP	Integer	25	
NUMBER	Integer	1	
NX	Integer	10	
NZ	Integer	10	
OUT.SAG	Character		
OUT.XSAG	Character		
OUT.ZSAG	Character		
PERIODIC	Logical	False	
PHI	Real	0.0	Degrees
POLARIZE	Real	0.0	Degrees
POWER.FILE	Character		
POWER.SCAL	Real	1.0	
QUANTUM.EFF	Real	1.0	
RAY.CHECK	Logical	False	
RAYAREA	Real	0.0	μm(2D) μm^2(3D)
RAYS	Integer	1	
RAYTRACE	Character		
REFLECTS	Integer	1	
REL.POWER	Real	1.0	
SAG.ITS	Integer	10	
SIGMA	Real	0.0	μm

Parameter	Type	Default	Units
STRUCTURE	Character		
THETA	Real	0.0	Degrees
THINEST	Real	0.0	μm
TRANSV.STEP	Real	0.0	
USER.SPECTRUM	Logical	False	
VERBOSE	Logical	False	
WAVELENGTH	Real	0.623	μm
WAVEL.END	Real	0.0	μm
WAVEL.NUM	Integer	1	
WAVEL.SCAL	Real	1.0	
WAVEL.START	Real	0.0	μm
X1	Real	0.0	μm
X2	Real	0.0	μm
X.CENTER	Real	0.0	μm
X.GAUSSIAN	Logical	False	
X.MEAN	Real	0.0	μm
X.ORIGIN	Real	0.0	μm
X.RADIUS	Real	0.0	μm
X.SEMIAXIS	Real	0.0	microns
X.SIGMA	Real	0.0	μm
XCENTER	Real	0.0	μm
XGAUSSIAN	Logical	False	
XMAX	Real	-1.0×10^{20}	μm
XMEAN	Real	0.0	μm
XMIN	Real	1.0×10^{20}	μm
XRADIUS	Real	0.0	μm
XSIGMA	Real	0.0	μm
Y.ORIGIN	Real	0.0	μm
Y.SEMIAXIS	Real	0.0	microns
Z1	Real	0.0	μm
Z2	Real	0.0	μm
Z.CENTER	Real	0.0	μm

Parameter	Type	Default	Units
Z.GAUSSIAN	Logical	False	
Z.MEAN	Real	0.0	μm
Z.ORIGIN	Real	0.0	μm
Z.RADIUS	Real	0.0	μm
Z.SEMIAXIS	Real	0.0	microns
Z.SIGMA	Real	0.0	μm
ZCENTER	Real	0.0	μm
ZGAUSSIAN	Logical	False	
ZMEAN	Real	0.0	μm
ZMAX	Real	-1.0×10^{20}	μm
ZMIN	Real	1.0×10^{20}	μm
ZRADIUS	Real	0.0	μm
ZSIGMA	Real	0.0	μm

Description

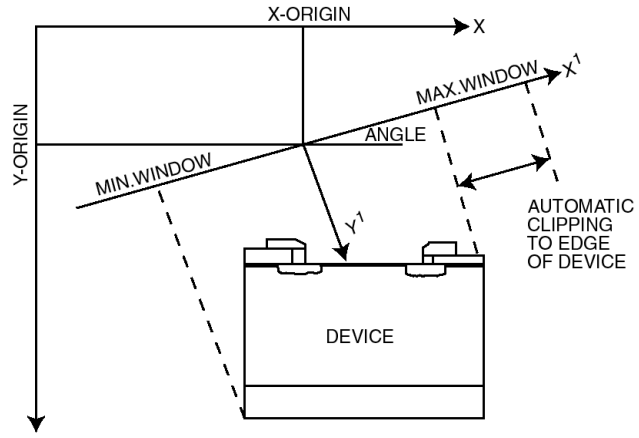


Figure 19-1: LUMINOUS Optical Source Coordinate System

AMBIENT.INDEX specifies the index of refraction of domain outside of all meshed regions.

ANGLE is the angle of propogation of the optical beam (see Figure 19-1 and Chapter 6: "3D Device Simulator", Figure 6-1). ANGLE=90 is vertical illumination from the top of the device. The synonym for this parameter is PHI.

BACK.REFL specifies that back side reflections are to be taken into account. When BACK.REFL is specified, the area outside the device domain is assumed to be a vacuum (i.e., $n = 1.0$, $k = 0.0$).

BPM (LUMINOUS2D) use the Beam Propagation Method instead of Ray-Tracing for analysis of light propagation in the device. Use BPM when diffraction of light or coherent effects are important (see Chapter 10: "Luminous: Optoelectronic Simulator", Section 10.6: "Beam Propagation Method in 2D").

BEAM parameters related specifically to ray tracing (such as RAYTRACE, RAYS, THINEST, MAX.WINDOW, MIN.WINDOW) are ignored when BPM is specified.

DEVICE specifies the name of a device in MIXEDMODE to identify to which device the beam is directed. The synonym for this parameter is STRUCTURE.

CIRCULAR is the synonym for ELLIPTICAL.

ELLIPTICAL specifies that an elliptical source is to be used in LUMINOUS 3D. When the ELLIPTICAL parameter is specified, the X.CENTER, Z.CENTER, X.RADIUS, and Z.RADIUS should also be specified. The synonym for this parameter is CIRCULAR.

F.REFLECT specifies the name of a file containing a C-INTERPRETER function for specifying reflection coefficient models as a function of wavelength, position, and angle incidence (LUMINOUS only).

F.RADIATE specifies the name of a file containing a C-INTERPRETER function for specifying generation rate as a function of position and optionally time. This function can be used to simulate single event upset (LUMINOUS2D only).

F3.RADIATE is the same as the F.RADIATE parameter but is applied in 3-D. This is typically used for single event or photogeneration simulations with LUMINOUS 3D.

FRONT.REFL specifies that front side reflections are to be taken into account. When FRONT.REFL is specified, the area outside the device domain is assumed to be a vacuum (i.e.,. n=1.0, k= 0.0).

GAUSSIAN is the synonym for X.GAUSSIAN.

INPUTRAYS specifies the filename of a file containing user-defined beam information. The file INPUTRAYS must have the following format.

```
NUMBER OF RAYS


NUMBER OF CHANGING PARAMETERS


PARAMETER1 PARAMETER2 PARAMETER3 ...


RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
...             ...                   ...                    ...
RAY NUMBER     PARAMETER1 VALUE     PARAMETER2 VALUE      PARAMETER3 VALUE ...
```

NUMBER OF RAYS stands for the total number of rays specified in the file. It should be the same as the last RAY NUMBER in the table.

NUMBER OF CHANGING PARAMETERS gives the total number of ray parameters given in the table. Most often, some parameters are the same for all the rays. For example, wavelength or ray power could be the same. In this case, there is no need to keep the table column for this constant parameter. Instead, you can set this parameter directly in the BEAM statement. NUMBER OF CHANGING PARAMETERS is equal to the number of columns in the table (not counting the first column of ray numbers).

PARAMETER1, PARAMETER2, PARAMETER3, and so on are numerical codes of parameters. Their order specifies the order of columns in the table. Numerical codes correspond to the following parameters:

- 1 - x coordinate of ray origin (in device coordinate system).
- 2 - y coordinate of ray origin (in device coordinate system).
- 3 - z coordinate of ray origin (in device coordinate system).

- 4 - ray angle phi in XY plane (0 - along X, 90 - along Y).
- 5 - ray angle theta in XZ plane (in 3D only, 0 - along X, and 90 - along Z).
- 6 - ray polarization angle (relative to the plane of incidence, 0 - TM, and 90 - TE).
- 7 - ray relative power.
- 8 - ray wavelength.
- 9 - ray area (μm^2 in 3D) or thickness (μm in 2D).

The first column of the table where ray parameters are specified lists ray numbers (integer values) in ascending order. Other colums list corresponding parameter values for each ray. For more details, see Chapter 10: "Luminous: Optoelectronic Simulator", the "User-Defined Beam" Section on page 10-23.

INTEGRATE specifies whether the user-specified spectrum (contained in the file identified by the POWER.FILE parameter) should be numerically integrated and averaged over the wavelength sampling or if the samples should be interpolated directly from the table.

LENS.HEIGHT specifies an attribute of the composite lenslet (see Chapter 10: "Luminous: Optoelectronic Simulator", Figure 10-9).

LENS.INDEX specifies the index of refraction of a lenslet (LUMINOUS 3D only).

LENS.PLANE specifies the minimum y-cordinate of the lenslet sphere (LUMINOUS 3D only).

LENS.RADIUS specifies the radius of the spere defining a lenslet (LUMINOUS 3D only).

LENS.WIDTH specifies an attribute of the composite lenslet (see Chapter 10: "Luminous: Optoelectronic Simulator", Figure 10-9).

LENS.X specifies the x coordinate of the center of the spere defining a lenslet (LUMINOUS 3D only).

LENS.Y specifies the y coordinate of the center of the spere defining a lenslet (LUMINOUS 3D only).

LENS.Z specifies the z coordinate of the center of the spere defining a lenslet (LUMINOUS 3D only).

LENS.XMIN, **LENS.XMAX**, **LENS.ZMIN**, **LENS.ZMAX** specify attributes of the composite lenslet (see Chapter 10: "Luminous: Optoelectronic Simulator", Figure 10-9).

LENS.XSAGS, **LENS.ZSAGS** specify the file names of aspheric lenslet Sag-h data files. These files must be in the following format.

```
number_of_pairs
Sag_1 h_1
Sag_2 h_2
  .    .
  .    .
  .    .
Sag_n h_n
```

Here, n is the number of pairs specified in the first line of the file. The pairs must be arranged in order of increasing h. The data modeling algorithm assumes that there is a sample with Sag=0 at h=0. Therefore, such a sample must not be included in the file.

LONGIT.STEP sets the mesh size in the longitudinal direction when using BPM. The default value is WAVELENGTH/16.0.

MAX.WINDOW specifies the maximum x-value of the illumination window relative to the coordinate system of the optical beam (see Figure 19-1). The illumination window is always clipped to the device domain. The synonym for this parameter is XMAX.

MEAN is the synonym for X.MEAN.

METAL.REFLECT sepcifies that all metals are to be treated as perfect reflectors.

MIN.POWER specifies the minimum intensity relative to the source that a given ray will be traced. This is useful for limiting the numbers of rays traced.

MIN.WINDOW specifies the minimum x-value of the illumination window relative to the coordinate system of the optical beam. The synonym for this parameter is XMIN.

NORMALIZE indicates that the POWER.FILE spectral intensity needs to be normalized (integral over spectrum equals 1). In this case, beam intensity is set by the B<n> parameter on the SOLVE statement.

NUMBER specifies the beam number (from 1 to 10). This number is used by the SOLVE statement to specify the relative intensity of different beams. You may specify beam numbers in any order that you desire.

NX specifies the number of rays traced along the source beam's x axis for LUMINOUS 3D (see Chapter 6: "3D Device Simulation", Figure 6-3).

NZ specifies the number of rays traced along the source beam's z axis for LUMINOUS 3D (see Chapter 6: "3D Device Simulator", Figure 6-3).

OUT.SAG specifies the name of a file used to output an aspheric lenslet profile for subsequent display in TONYPLOT. OUT.SAG allows a cutline along an arbitrary direction specified by two points in the X-Z plane using the X1, Z1, X2, and Z2 parameters. NSAMP specifies the number of samples used to represent the aspheric lenslet profile output using OUT.SAG.

OUT.XSAG, **OUT.ZSAG** specify the file names for output log files that can be displayed using TONYPLOT. These files contain the data input from the LENS.XSAGS and LENS.ZSAGS files and the results of the data modeling.

PERIODIC specifies that for ray tracing, the structure is to be treated as periodic in the x and z directions. Rays exiting the sides of the device are wrapped around to the other side of the device.

PHI is the synonym for ANGLE.

POLARIZE specifies the polarization of the optical beam at the origin. The polarization angle is the angle between the E vector and the incidence plane.

POWER.SCALE specifies a scale factor. This factor is multiplied by each of the relative powers in the spectrum file when multi-spectral simulations are performed. The POWER.SCALE parameter can be used to perform unit conversions.

POWER.FILE specifies the filename of a spectrum file. The spectrum file must be in the following format.

```
number of pairs
wavelength_1  power_1
wavelength_2  power_2
    . . .       . . .
    . . .       . . .
wavelength_n  power_n
```

Note: The power file must contain at least two power/wavelengths pairs.

QUANTUM.EFF is a quantum efficiency factor which specifies the number of carrier pairs generated per photon absorbed.

RAYTRACE specifies the name of a file where the results of a ray trace are saved. The ray trace may be viewed using TONYPLOT3D (LUMINOUS 3D only).

RAY.CHECK enables diagnostic printing during ray tracing.

RAYAREA is a parameter that specifies area in μm^2 (thickness in 2D in um) of each ray in the BEAM statement. This parameter is used when the specified INPUTRAYS file does not contain ray area information.

RAYS specifies the number of rays you want to split the optical beam. LUMINOUS will automatically split the beam into enough rays to resolve the geometry. Use of the number parameter will cause further splitting of the optical beam (LUMINOUS only).

REFLECTS specifies the number of reflections that will be traced. When the value of the REFLECTS parameter is increased, the total number of rays traced increases non-linearly. We recommend that this parameter be used wisely. For example, a single ray incident on three material layers will produce 4 rays if REFLECTS=0 is specified, 10 rays if REFLECTS=1 is specified, and 24 rays if REFLECTS=2 is specified.

REL.POWER specifies the relative power in the beam when mono-spectral simulations are performed. This factor is multiplied by the power parameters specified in the SOLVE statement to give the total optical power in the beam.

SPG.ITS specifies the number of interations used to model the aspherical lenslet Sag-h data using non-linear least squares fitting.

SIGMA is the synonym for X.SIGMA.

STRUCTURE is a synonym for DEVICE.

THETA specifies the angle of rotation for the source beam direction of propogation relative to the x-y plane (see Chapter 6: "3D Device Simulator", Figure 6-2) (LUMINOUS 3D only).

THINEST specifies the width of the thinnest ray to be traced (LUMINOUS only).

TRANSV.STEP sets the mesh size in the transverse direction when using BPM. The default value is WAVELENGTH/16.0.

USER.SPECTRUM specifies that sampling in wavelength used for simulation is the same as that contained in the POWER.FILE.

VERBOSE enables a higher level of diagnostic run-time printing.

WAVELENGTH specifies the optical wavelength of the source beam (in the vacuum) for mono-spectral simulations.

WAVEL.END specifies the maximum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.

WAVEL.NUM specifies the number of wavelengths which will be used when multi-spectral simulations are performed. Spectral illumination is selected when the WAVEL.NUM parameter is greater than 1. Once multi-spectral simulations are selected, you must specify the POWER.FILE, WAVEL.START, and WAVEL.END parameters.

WAVEL.SCAL specifies the scale factor for the wavelengths which are used in multi-spectral simulations. Each of the wavelengths in the spectrum file is multiplied by this scale factor.

WAVEL.START specifies the minimum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.

X1, **Z1**, **X2**, and **Z2** specify the coordinates of two points in the X-Z plane for extraction of an aspheric lenslet profile along the cutline defined by the two points. The profile is output to the file specified by the OUT.SAG parameter for subsequent display in TONYPLOT.

X.CENTER specifies the X coordinate of the center of an elliptical source related to the beam coordinate system (which is centered at the beam origin) for LUMINOUS 3D. Note that to enable elliptical soruces, the ELLIPTICAL parameter should also be specified. The synonym for this parameter is XCENTER.

X.GAUSSIAN specifies that a LUMINOUS 3D source is to have a Gaussian intensity profile in the X direction related to the beam coordinate system (which is centered at the beam origin). When X.GAUSSIAN is specified, X.MEAN and X.SIGMA should also be specified. The synonyms for this parameter are XGAUSSIAN and GAUSSIAN.

X.MEAN specifies the location of the mean value of a Gaussian source in the X direction related to the beam coordinate system (i.e., from the X.CENTER position). When X.MEAN is specified, X.GAUSSIAN and X.SIGMA should also be specified. The synonyms for this parameter are XMEAN and MEAN.

X.ORIGIN specifies the x coordinate of the optical beam origin (see Figure 19-1 and Chapter 6: "3D Device Simulator", Figure 6-1). The beam must originate outside all device regions.

X.RADIUS specifies the X axis radius of an elliptical source related to the beam coordinate system (which is centered at the beam origin) for LUMINOUS 3D. Note that to enable elliptical sources, the ELLIPTICAL should also be specified. The synonym for this parameter is XRADIUS.

X.SEMIAXIS, **Y.SEMIAXIS**, **Z.SEMIAXIS** specify attributes of the elliptical lense (see Chapter 10: "Luminous: Optoelectronic Simulator", Equation 10-27).

X.SIGMA specifies the standard deviation in the X direction of the beam coordinate system for a Gaussian source in LUMINOUS 3D. When X.SIGMA is specified, X.GAUSSIAN and X.MEAN should also be specified. The synonym for this parameter is XSIGMA.

XMAX specifies the maximum x coordinate in the source beam coordinate system for ray tracing in LUMINOUS 3D (see Chapter 6: "3D Device Simulator", Figure 6-3).

XMIN specifies the minimum x coordinate in the source beam coordinate system for ray tracing in LUMINOUS 3D (see Chapter 6: "3D Device Simulator", Figure 6-3).

Y.ORIGIN specifies the y coordinate of the optical beam origin (see Figure 19-1 and Chapter 6: "3D Device Simulator", Figure 6-1). The beam must originate outside all device regions.

Z.CENTER specifies the Z coordinate of the center of an elliptical source related to the beam coordinate system (which is centered at the beam origin). Note that to enable elliptical sources, the ELLIPTICAL parameter should also be specified. The synonym for this parameter is ZCENTER.

Z.GAUSSIAN specifies that a LUMINOUS 3D source is to have a Gaussian intensity profile in the Z direction related to the beam coordinae system (which is centered at the beam origin). When Z.GAUSSIAN is specified, Z.MEAN and Z.SIGMA should also be specified. The synonym for this parameter is ZGAUSSIAN.

Z.MEAN specifies the location of the mean value of a Gaussian soruce in the Z direction related to the beam coordinate system. When Z.MEAN is specified, Z.GAUSSIAN and Z.SIGMA should also be specified. The synonym for this parameter is ZMEAN.

Z.RADIUS specifies the X axis radius of an elliptical source related to the beam coordinate system. When Z.MEAN is specified, Z.GAUSSIAN and Z.SIGMA should also be specified. The synonym for this parameter is ZRADIUS.

Z.SIGMA specifies the standard deviation in the Z direction of the beam coordinate system for a Gaussian source in LUMINOUS 3D. When Z.SIGMA is specified, Z.GAUSSIAN and Z.MEAN should also be specified. The synonym for this parameter is ZSIGMA.

ZMAX specifies the maximum z coordinate in the source beam coordinate system for ray tracing in LUMINOUS 3D (see Chapter 6: "3D Device Simulator", Figure 6-1).

ZMIN specifies the minimum z coordinate in the source beam coordinate system for ray tracing in LUMINOUS 3D (see Chapter 6: "3D Device Simulator", Figure 6-1).

Z.ORIGIN specifies the z coordinate of the optical beam origin (see Chapter 6: "3D Device Simulator", Figure 6-1). The beam must originate outside all device regions (LUMINOUS 3D only).

Monochromatic Beam Example

This beam has a monochromatic spectrum with a wavelength of 0.6 μm. The beam originates at x=0.5 and y=-2.0. It has a 90 degree propogation angle and a beam width of 0.2 μm which is centered at the beam origin.During the ray trace calculation the rays will be terminated when the power level along the ray falls to 5% of the original power.

```
BEAM NUM=1 WAVELENGTH=0.6 X=0.5 Y=-2.0 ANG=90.0 MIN=-0.1 MAX=0.1 \
          MIN.POWER=0.05
```

Multi-spectral Beam Example

A multi-spectral beam (at a 45 degree angle) which originates at x=0.0 and y=-1.0. The multi-spectral source is imported from the spectrum file, *source.spc*. The spectrum is discretized into our wavelengths between 0.4 μm and 0.6 μm.

```
BEAM NUM=2 X=0.0 Y=-1.0 ANG=45.0 \
     POWER.FILE=SOURCE.SPC WAVEL.START=0.4 \
     WAVEL.END=0.6 WAVEL.NUM=4
```

LUMINOUS3D Lens Example

```
BEAM NUM=1 X.ORIGIN=2.5 Y.ORIGIN=-1.0 Z.ORIGIN = 2.5 ANG=90.0 WAVEL=0.6 \
          NX=10 NZ=10 LENS.X=2.5 LENS.Y=-0.5 LENS.Z=2.5 \
          LENS.INDEX=2.03 LENS.RADIUS=0.25 LENS.PLANE=-0.5
```

Gaussian Intensity Profile Example

The following beam statement will define a beam window 2 μm wide that is centred at (X.ORIGIN, Y.ORIGIN) with a Gaussian peak of 0.01 W/cm^2 with a standard deviation of 0.05 μm.

```
beam   num=1 x.origin=5.0 y.origin=-1.0 angle=90.0 wavelength=0.6  \
       xmin=-1   xmax=1 GAUSSIAN   MEAN=0   SIGMA=0.05 RAYS=200
SOLVE B1=1E-2
```

Note: It is recommended that you use the RAYS parameter to define a large number of rays across the beam to ensure that the Gaussian profile is adequately reproduced. The rays are evenly spaced across the beam so it is necessary to use a large number of them.

19.3: COMMENT,

COMMENT allows comments to be placed in an ATLAS input file. ATLAS will print and display comment lines.

Syntax

```
COMMENT [<string>]
      # [<string>]
```

string is any alphabetic, numeric, or alphanumeric sequence of characters. The synonym for this parameter for #.

Example

```
COMMENT  ATLAS is a copyright of Silvaco International
#        ATLAS is a copyright of Silvaco International
```

Note: The $ was allowed as a comment character in previous versions of ATLAS. This should be avoided and replaced by the # or COMMENT statement.

19.4: CONTACT

CONTACT specifies the physical attributes of an electrode.

Note: If the CONTACT statement is not used for a given electrode, the electrode is assumed to be charge-neutral (Ohmic).

Syntax

```
CONTACT NUMBER=<n>|NAME=<ename>|ALL [<wfp>] [<bc>] [<lcr>] [<link>]
```

Parameter	Type	Default	Units
ALL	Logical	False	
ALPHA	Real	0	cm
ALUMINUM	Logical	False	
BARRIER	Logical	False	
BETA	Real	1.0	
CAPACITANCE	Real	0	F/μm
COMMON	Character		
CON.RESIST	Real	0	$\Omega \cdot cm^2$
CURRENT	Logical	False	
DEVICE	Character		
E.TUNNEL	Logical	False	
F.WORKF	Character		
ELE.CAP	Integer		
EXCLUDE_NEAR	Logical	False	
EXT.ALPHA	Real	0	$W/(cm^2K)$
EXT.TEMP	Real	300	K
F.ETUNNEL	Character		
FACTOR	Real	0	
FLOATING	Logical	False	
FG.CAP	Real	0.0	F/μm
GAMMA	Real	1.0	
INDUCTANCE	Real	0	H·μm
ME.TUNNEL	Real	1.0	
MO.DISILICIDE	Logical	False	

Parameter	Type	Default	Units
MOLYBDENUM	Logical	False	
MULT	Logical	False	
NAME	Character		
NEUTRAL	Logical	True	
N.POLYSILICON	Logical	False	
NUMBER	Integer		
P.POLYSILICON	Logical	False	
PARABOLIC	Logical	False	
PSURF.REC	Logical	False	
REFLECT	Logical	False	
RESISTANCE	Real	0	Ω.μm
SHORT	Logical	False	
STRUCTURE	Character		
SURF.REC	Logical	False	
QTUNN.CMASS	Real	1.0	
QTUNN.VMASS	Real	1.0	
THERMIONIC	Logical	False	
TU.DISILICIDE	Logical	False	
TUNGSTEN	Logical	False	
VSURFN	Real	see description	cm/s
VSURFP	Real	see description	cm/s
WORKFUN	Real	0	V

Description

NAME specifies the name of a previously defined electrode. See Section 19.10: "ELECTRODE" for more information.

NUMBER specifies the contact number to be defined. It must be the number of a previously defined electrode. It is recommended that electrode names be used rather than numbers.

ALL defines the same properties for all electrodes.

DEVICE specifies which device the CONTACT statement applies to in MIXEDMODE. The synonym for this parameter is STRUCTURE.

STRUCTURE is a synonym for DEVICE.

wfp is one of the work function parameters described below. It is permitted to either specify the name of a material or a work function value (WORKFUN parameter).

bc is one or more of the boundary condition parameters.

lcr is one or more of the external parasitic element parameters.

link is one or more of a set of parameters that allow you to associate two or more electrodes electrically.

Workfunction Parameters

ALUMINUM specifies aluminum as the contact material for the electrode. This sets the workfunction to 4.10V. Note that this parameter should not be set if an Ohmic contact is required.

F.WORKF specifies the name of a file containing a C-INTERPRETER function describing the workfunction as a function of the electric field.

MOLYBDENUM specifies molybdenum as the contact material for the electrode. This sets the work function of the electrode to 4.53V.

MO.DISILICIDE specifies molybdenum disilicide as the contact material for the electrode. This sets the work function of the electrode to 4.80V.

NEUTRAL specifies that the electrode is Ohmic. This is the default characteristic of an electrode.

N.POLYSILICON specifies n+ doped polysilicon as the contact material for the electrode. This sets the work function to 4.17V.

NSURF.REC enables finite surface recombination velocity for electrons.

P.POLYSILICON specifies p+ polysilicon as the contact material for the electrode. This sets the work function to 4.17V + Eg(Si).

PARABOLIC enables the parabolic Schottky field emission model as given in Equation 3-147.

PSURF.REC enables finite surface recombination velocity for holes.

THERMIONIC enables the thermionic emission in the Schottky models.

TU.DISILICIDE specifies tungsten disilicide as the contact material for the electrode. This sets the work function to 4.80V.

TUNGSTEN specifies tungsten as the contact material for the electrode. This sets the work function to 4.63V.

WORKFUN specifies the work function of the electrode in V. This parameter must be specified in the form `WORKFUN=n` where `n` is a real number. This specification is absolute workfunction and not workfunction difference to the semiconductor.

Note: If no `WORKFUN` or material type parameter is specified, the electrode is assumed to be an Ohmic contact

---

Boundary Conditions

CURRENT specifies current boundary conditions. If specified, `CAPACITANCE`, `CON.RESIST`, `INDUCTANCE`, or `RESISTANCE` may not be specified.

FLOATING specifies a charge boundary condition. This parameter is used to specify the floating gate in EPROM devices. This parameter can only be used for insulated contacts. If specified, `CAPACITANCE`, `CON.RESIST`, `INDUCTANCE`, or `RESISTANCE` may not be specified, but special syntax exists for adding capacitances to a floating contact. See `EL<n>.CAP` and `FG<n>.CAP`. The synonym for this parameter is `CHARGE`.

ALPHA specifies the linear, dipole lowering coefficient. This parameter has no effect unless the `BARRIER` parameter has been specified (See Chapter 3: “Physics”, Equation 3-143).

BARRIER turns on the barrier lowering mechanism for Schottky contacts.

BETA specifies a prefactor in the barrier lowering (See Chapter 3: "Physics", Equation 3-143).

E.TUNNEL specifies that the Schottky tunneling model for electrons will be used. `E.TUNNEL` will also enable the `SURF.REC` boundary condition which models the thermionic emission in a Schottky contact.

EXCLUDE_NEAR specifies that the contact should be excluded from the algorithm that finds the nearest electrode to a given point in the direct quantum tunneling model (See Chapter 3: "Physics", Section 3.6.6: "Gate Current Models").

EXT.ALPHA specifies the inverse value of the thermal resistance that can be applied to a contact when performing energy balance simulations. Basically, the thermal resistance allows the carrier energy boundary condition at a contact to be a value other than the ambient temperature.

EXT.TEMP specifies the external temperature, or carrier energy, of an electrode when performing energy balance simulations.

F.ETUNNEL specifies the name of a file containing a C-INTERPRETER function that specifies electron tunneling at a Schottky contact.

GAMMA specifies a exponent in the linear barrier lowering term (See Chapter 3: "Physics", Equation 3-143).

ME.TUNNEL specifies the relative effective mass for use in the electron tunneling mode (see `E.TUNNEL`).

REFLECT specifies that for Energy Balance and Hydrodynamic simulation using a Neumann (reflective) boundary condition for carrier temperature equations. By default, contacts are handled as Dirichlet boundaries with carrier temperature equal to lattice temperature.

SURF.REC specifies that finite surface recombination velocities are used at the respective contact. This parameter must be specified in the form `SURF.REC [VSURFN=<n>] [VSURFP=<p>]`, where **n** and **p** are real numbers.

VSURFN specifies the actual surface recombination velocities for electrons (V_{sn}). If this parameter is not specified, its default value is calculated by Equation 19-1.

$$V_{sn} = \frac{\mathtt{ARICHN} \cdot T_L^{\,2}}{qN_C} \qquad \text{19-1}$$

where `ARICHN` is the effective Richardson constant for electrons. This constant accounts for quantum mechanical reflections and tunneling.

VSURFP specifies the actual surface recombination velocities for holes (V_{sp}). If this parameter is not specified, its default value is calculated by Equation 19-2.

$$V_{sp} = \frac{\mathtt{ARICHP} T_L^{\,2}}{qN_V} \qquad \text{19-2}$$

where `ARICHP` is the effective Richardson constants for holes. This constant accounts for quantum mechanical reflections and tunneling.

QTUNN.CMASS specifies the electron effective mass to use in the contact for a direct quantum tunneling model (See Chapter 3: "Physics", Section 3.6.6: "Gate Current Models").

QTUNN.VMASS specifies the hole effective mass to use in the contact for a direct quantum tunneling model (See Chapter 3: "Physics", Section 3.6.6: "Gate Current Models").

Contact Parasitics

RESISTANCE specifies a lumped resistance value. You may not specify both RESISTANCE and CON.RESIST.

INDUCTANCE specifies an external inductance which is related to the specified electrode. A synonym is L.

CAPACITANCE specifies a lumped capacitance value to be attached to the contact.

CON.RESIST specifies a distributed contact resistance. You cannot specify both CON.RESIST and RESIST.

Note: There are restrictions on the allowed numerical methods and types of analysis possible when any form of parasitic element is attached to a contact. See the Getting Started section for details.

Electrode Linking Parameters

These parameters allow one electrode to be biased as a function of another electrode. This allows separate regions of the same physical contact to be linked together. For example, in the statement:

```
CONTACT NAME=MYDRAIN COMMON=DRAIN FACTOR=4.6
```

The electrode, MYDRAIN, is linked to the electrode, DRAIN. The bias on MYDRAIN will always be equal to the bias on the drain plus 4.6. If the optional MULT parameter had been specified, the bias on MYDRAIN would be equal to the bias on the drain multiplied by 4.6.

COMMON specifies the electrode name to which the contact refered to by NAME is linked. Although the electrodes are linked, separate currents will be saved for both electrodes unless SHORT is also specified. The electrode refered to in NAME should not appear on any SOLVE statments since its bias is now determined as a function of the electrode refered to by COMMON.

SHORT specifies that the electrode referred to by NAME is shorted to the electrode specified by the COMMON parameter. This implies that the two electrodes will be treated as one and only one value will be written to log files and in the run time output.

FACTOR specifies the constant offset voltage (or current) between the electrodes refered by by NAME and COMMON. By default, FACTOR is added the defined voltage.

MULT specifies that FACTOR is a multiplier.

Floating Gate Capacitance Parameters

In some cases, you may want to simulate floating gate structures, in 2D, which have control gates that are longer in the unsimulated dimension than the floating gate. In these cases, specify the following parameters to account for addition capacitance between the floating gate and the control gate and other electrodes. Up to four extra capacitances are allowed so in the following **<n>** is an integer number between 1 and 4.

EL<n>.CAP specifies the name of the electrode to which the extra capacitance is linked.

FG<n>.CAP specifies the additional capacitance per unit length to be added between the floating gate and electrode specified in EL<n>.CAP.

Schottky Barrier and Surface Recombination Example

This example defines all electrodes except number 2 (aluminum) to be neutral. Electrode number 2 also includes finite surface recombination velocities and barrier lowering. A definition in the second statement overrides that definition in the first statement.

```
CONTACT ALL NEUTRAL
CONTACT NUMBER=2 ALUMINUM SURF.REC BARRIER
```

Parasitic Resistance Example

This example attaches a lumped resistor with a value of $10^5 \Omega\mu m$ to the substrate. A distributed contact resistance of $10^{-6} \Omega \cdot cm^2$ is included on the drain.

```
CONTACT NAME=substrate RESISTANCE=1E5
CONTACT NAME=drain CON.RESIST=1E-6
```

Floating Gate Example

This syntax defines a floating contact with a workfunction equal to 4.17eV. An extra 1fF/um capacitance is added between this electrode and the electrode named `cgate`.

```
CONTACT NAME=fgate FLOATING N.POLY EL1.CAP=cgate FG1.CAP=1e-15
```

Note: The `MODELS PRINT` command can be used to echo the `back contact` workfunction and parasitic elements settings to the run-time output.

19.5: CURVETRACE

CURVETRACE sets the parameters for the automatic curve tracing routine.

Syntax

```
CURVETRACE <params>
```

Parameter	Type	Default	Units
ANGLE1	Real	5	
ANGLE2	Real	10	
ANGLE3	Real	15	
BEG.VAL	Real	0.0	V
CONTROLRES	Real	1.0×1.0^{10}	Ω
CONTR.ELEC	Integer		
CONTR.NAME	Character		
CURR.CONT	Logical		
END.VAL	Real		
MAXDV1	Real	0.1	V
MAXDV2	Real	1.0	V
MINCUR	Real		
MINDL	Real	0.1	
NEXTST.RATIO	Real	2.0	
STEP.CONT	Logical		
STEP.INIT	Real	0.1	V
STEPS	Real	0.1	
TURNINGPOINT	Logical	False	
VOLT.CONT	Logical	False	

Note: The CURVETRACE function of ATLAS creates several temporary files in the local directory where it is run. Two simultaneous ATLAS runs using curvetrace in the same directory will interfere with each other.

Description

ANGLE1, ANGLE2, and **ANGLE3** are critical angles (in degrees) affecting the smoothness and step size of the trace. If the difference in slopes of the last two solution points is less than ANGLE1, the step size will be increased for the next projected solution. If the difference lies between ANGLE1 and ANGLE2, the step size remains the same. If the difference is greater than ANGLE2, the step size is reduced. ANGLE3 is the maximum difference allowed, unless overridden by the MINDL parameter. ANGLE2 should always be greater than ANGLE1 and less than ANGLE3.

BEG.VAL is the value of the voltage at the starting point of the curve trace for the controlling electrode.

CONTR.ELEC is the number of the electrode that is designated as a control electrode.

CONTR.NAME is the name of the control electrode.

CURR.CONT denotes that a maximum current on the control electrode, specified by END.VAL is used as the upper bound on the trace.

CONTROLRES specifies the control resistance value to be set at the control electrode after the current at the control electrode becomes greater than the value specified by the MINCUR parameter. Using a large value of the control resistance allows you to set the value of the current (instead of the internal voltage) on the control electrode. A large value of the control resistance is suitable for electron devices where the shape of the characteristic curve is vertical or exhibits the presence of turning points (where it is required to switch the electrode control from voltage to current).

END.VAL is used to stop tracing if the voltage or current of control electrode equals or exceeds END.VAL.

MAXDV1 specifies maximum value of the voltage bias step at the beginning of the CURVETRACER algorithm. This limitation is used until the current at the control electrode is greater than the value specified by the MINCUR parameter.

MAXDV2 specifies maximum value of the voltage bias step to be used after the current at the control electrode becomes greater than the value specified by the MINCUR parameter.

MAXDV1 and **MAXDV2** control (together with NEXTST.RATIO) the maximum voltage step at each point of the IV curve. Using large values for them means that few points are used to trace the curve. This results in a faster simulation but can affect the stability of the tracing algorithm and decrease the accuracy of the IV curve.

MINCUR may be used to set a small current value in order to switch from internal control electrode bias ramping to external ramping with load resistor. This parameter is recommended for small current breakdown simulation.

MINDL is the minimum normalized step size allowed in the trace. Usually, you don't need to adjust this parameter. Increasing MINDL will reduce the smoothness of the trace by overriding the angle criteria, resulting in more aggressive projection and fewer simulation points. Reducing MINDL will enhance the smoothness and increase the number of points in the trace.

NEXTST.RATIO specifies which factor to use to increase the voltage step on the smooth parts of the I-V curve.

STEPS is the number of operational points on a trace if STEP.CONT was specified.

STEP.CONT specifies that the trace will proceed for a certain number of simulation points.

STEP.INIT specifies initial voltage step size.

Note: To set a sweep of increasingly negative voltage in CURVETRACE, you only need to set STEP.INIT to be negative. Since all parameters are multiplier of STEP.INIT, the whole voltage sweep will be negative.

TURNINGPOINT specifies that binary output solution files will be saved whenever the slope of the IV curve changes sign (i.e., there is a turning poin)t. The name of the output file is *soln.num*, where *num* is the number of the current solution.

VOLT.CONT denotes that a maximum voltage on the control electrode, specified by `END.VAL` is used as the upper bound on the trace.

Diode Breakdown Example

To trace a diode breakdown curve using current value as a termination criteria, the following statement may be used:

```
CURVETRACE CURR.CONT END.VAL=0.01 CONTR.NAME=anode \
MINCUR=5E-12 NEXTST.RATIO=1.1  STEP.INIT=0.1
SOLVE CURVETRACE
```

19.6: DBR

The DBR statement is used to define Distributed Bragg Reflectors (DBR). DBRs are periodic structures composed of alternating layers of two different materials. The alias for this parameter is SUPERLATTICE.

Syntax

```
DBR <parameters>
```

Parameter	Type	Default	Units
BOTTOM	Logical	False	
CALC1.STRAIN	Logical	False	
CALC2.STRAIN	Logical	False	
HALF.CYCLES	Real	2	
LAYERS	Real	2	
LED1	Logical	False	
LED2	Logical	False	
MAT1	Character		
MAT2	Character		
N1	Real	0	
N2	Real	0	
NA1	Real	0.0	cm^{-3}
NA2	Real	0.0	cm^{-3}
NAME1	Character		
NAME2	Character		
ND1	Real	0.0	cm^{-3}
ND2	Real	0.0	cm^{-3}
NU12	Integer	5	
NU21	Integer	5	
POLARIZ1	Logical	False	
POLARIZ2	Logical	False	
POLAR1.CHARG	Real	0	cm^{-2}
POLAR2.CHARG	Real	0	cm^{-2}
POLAR1.SCALE	Real	1.0	
POLAR2.SCALE	Real	1.0	

QWELL1	Logical	False	
QWELL2	Logical	False	
SPA1	Real	0.0	microns
SPA2	Real	0.0	microns
STRAIN1	Real	0.0	
STRAIN2	Real	0.0	
TH12	Real		
TH21	Real		
THICK1	Real	0.0	microns
THICK2	Real	0.0	microns
TOP	Logical	False	
WELL1.CNBS	Integer	1	
WELL2.CNBS	Integer	1	
WELL1.FIELD	Logical	True	
WELL2.FIELD	Logical	True	
WELL1.GAIN	Real	1.0	
WELL2.GAIN	Real	1.0	
WELL1.NX	Integer	10	
WELL2.NX	Integer	10	
WELL1.NY	Integer	10	
WELL2.NY	Integer	10	
WELL1.VNBS	Integer	1	
WELL2.VNBS	Integer	1	
Y.START	Real	0.0	microns
Y.FINISH	Real	0.0	microns
Y.FINAL	Real	0.0	microns
Y.END	Real	0.0	microns
Y1.COMP	Real	0.0	
Y2.COMP	Real	0.0	
X1.COMP	Real	0.0	
X2.COMP	Real	0.0	

Description

The DBR statement is a short cut for specifying a set of REGION statements, which specifies a stack of alternating layers of two materials. The material compositions of the layers are specified by the MAT1, MAT2, NA1, NA2, ND1, ND2, STRAIN1, STRAIN2, X1.COMP, X2.COMP, Y1.COMP, and Y2.COMP parameters. The number of layers are specified by the HALF.CYCLES parameter. The locations of the layers are specified by the Y.START, Y.FINISH or the TOP/BOTTOM parameters. The meshing of the layers is specified by the N1, N2 or the SPA1, SPA2 parameters.

For more information about DBR, see Chapter 9: "VCSEL Simulator", the "Specifying Distributed Bragg Reflectors" Section on page 9-10.

BOTTOM specifies that the DBR is to be added at the bottom (starting at the maximum previously specified Y coordinate and extending in the positive Y direction) of the structure.

CALC1.STRAIN, **CALC2.STRAIN** specifies that the strain in the region/cycle is calculated from the lattice mismatch with adjacent regions.

HALF.CYCLES specifies the total number of layers. Note that if HALF.CYCLES is odd, there will be more layers of one material than the other. Its alias is LAYERS.

Note: Don't confuse HALD.CYCLES with fractions of the optical emisson wavelength. HALF.CYCLES relates to layers.

LED1, **LED2** specifies that the region/cycle is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 11: "LED: Light Emitting Diode Simulator".

MAT1 specifies the material name for layers of material 1.

MAT2 specifies the material name for layers of material 2.

N1 specifies the integer number of mesh lines per layer of material 1. Note that if N1 is specified, then SPA1 shouldn't be specified.

N2 specifies the integer number of mesh lines per layer of material 2. Note that if N2 is specified, then SPA2 shouldn't be specified.

NAME1, **NAME2** specify the names of the regions composed of material 1 and material 2. You can conveniently use these names to modify material parameters and models on the MATERIAL, MODEL, IMPACT, and MOBILITY statements.

NA1 specifies the ionized acceptor concentration in layers of material 1.

NA2 specifies the ionized acceptor concentration in layers of material 2.

ND1 specifies the ionized donor concentration in layers of material 1.

ND2 specifies the ionized donor concentration in layers of material 2.

NU12 is the number of grid lines in the graded region between the 1st and 2nd half cycle in the direction specified by the BOTTOM or TOP parameters.

NU21 is the number of grid lines in the graded region between the 2nd and 1st half cycle in the direction specified by the BOTTOM or TOP parameters.

POLARIZ1, **POLARIZ2** enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

POLAR1.CHARG, **POLAR2.CHARG** specify polarization charge densities to replace those calculated using Equations 3-295 and 3-296.

POLAR1.SCALE, **POLAR2.SCALE** specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the DBR statement. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

QWELL1, **QWELL2** specifies that the region/cycle is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.

SPA1 specifies the spacing between mesh lines in layers of material 1. Note that this should be smaller than or equal to THICK1.

SPA2 specifies the spacing between mesh lines in layers of material 2. Note that this should be smaller than or equal to THICK2.

STRAIN1 specifies the strain in the layer of material 1.

STRAIN2 specifies the strain in the layer of material 2.

TOP specifies that the DBR is to be added at the top (starting at the minimum previously specified Y coordinate and extending in the negative Y direction) of the structure.

TH12 is the thickness of grading region between 1st half cycle and 2nd half cycle going in the direction specified by the BOTTOM or TOP parameters.

TH21 is the thickness of grading region between 2nd half cycle and 1st half cycle going in the direction specified by the BOTTOM or TOP parameters.

THICK1 specifies the thickness of each layer of material 1.

THICK2 specifies the thickness of each layer of material 2.

WELL1.CNBS, **WELL2.CNBS**, **WELL1.VNBS**, **WELL2.VNBS** specify the number of bound states retained for calculation of radiative recombination or gain if the region/cycle is treated as a quantum well as specified by the QWELL1, QWELL2 parameters.

WELL1.FIELD, **WELL2.FIELD**, when enabled, specify that the calculations of bound state energies should include the effects of the local field.

WELL1.GAIN, **WELL2.GAIN** specify a constant scale factor multiplied by the calculated gain to give the net gain used for certain optoelectronic calculations.

WELL1.NX, **WELL2.NX**, **WELL1.NY**, **WELL2.NY** specify the number of slices (WELL#.NX) and the number of samples per slice (WELL#.NY) are used in the solution of Schrodinger's equation to obtain the local bound state energies for calculation of radiative recombination or gain or both for certain optoelectronic models.

Y.START specifies the starting location of a DBR that begins with a layer of material 1 and alternates materials in an increasing Y direction.

Y.FINISH specifies the starting location of a DBR that begins with a layer of material 1 and alternates materials in an decreasing Y direction.

Y.FINAL and **Y.END** are aliases for Y.FINISH.

Y1.COMP specifies the composition fraction y in layers of material 1.

Y2.COMP specifies the composition fraction y in layers of material 2.

X1.COMP specifies the composition fraction x in layers of material 1.

X2.COMP specifies the composition fraction x in layers of material 2.

Note: DBR statements can be intermixed with X.MESH, Y.MESH, and REGION statements.

19.7: DEFECTS

DEFECTS activates the band gap defect model and sets the parameter values. This model can be used when thin-film transistor simulations are performed using the TFT product.

Syntax

```
DEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	False	
DEVICE	Character		
DFILE	Character		
EGA	Real	0.4	eV
EGD	Real	0.4	eV
F.TFTACC	Character		
F.TFTDON	Character		
FILE.INT	Logical	False	
INT_LIM1	Real	0	eV
INT_LIM2	Real	0	eV
MATERIAL	Character		
NA.MIN	Real	1.0×10^{13}	cm^{-3}
ND.MIN	Real	1.0×10^{13}	cm^{-3}
NGA	Real	5.0×10^{17}	cm^{-3}
NGD	Real	1.5×10^{18}	cm^{-3}
NTA	Real	1.12×10^{21}	cm^{-3}/eV
NTD	Real	4.0×10^{20}	cm^{-3}/eV
NUMBER	Real	All	
NUMA	Real	12	
NUMD	Real	12	
SIGGAE	Real	1.0×10^{-16}	cm^2
SIGGAH	Real	1.0×10^{-14}	cm^2
SIGGDE	Real	1.0×10^{-14}	cm^2
SIGGDH	Real	1.0×10^{-16}	cm^2

Parameter	Type	Default	Units
SIGTAE	Real	1.0×10^{-16}	cm^2
SIGTAH	Real	1.0×10^{-14}	cm^2
SIGTDE	Real	1.0×10^{-14}	cm^2
SIGTDH	Real	1.0×10^{-16}	cm^2
STRUCTURE	Character		
TFILE	Character		
WGA	Real	0.1	eV
WGD	Real	0.1	eV
WTA	Real	0.025	eV
WTD	Real	0.05	ev
X.MIN	Real		µm
X.MAX	Real		µm
Y.MIN	Real		µm
Y.MAX	Real		µm
Z.MIN	Real		µm
Z.MAX	Real		µm

Description

The DEFECTS statement is used to describe the density of defect states in the band gap. You can specify up to four distributions, two for donor-like states and two for acceptor-like states. Each type of state may contain one exponential (tail) distribution and one Gaussian distribution.

AFILE specifies the file name where the acceptor state density distribution, as a function of energy, will be stored. You can examine this file by using TONYPLOT.

CONTINUOUS specifies that the continuous defect integral model will be used.

DEVICE specifies which device the statement applies in mixed mode simulation. The synonym for this parameter is STRUCTURE.

DFILE specifies the file name where the donor state density distribution, as a function of energy, will be stored. You can examine this file by using TONYPLOT.

EGA specifies the energy that corresponds to the Gaussian distribution peak for acceptor-like states. This energy is measured from the conduction band edge.

EGD specifies the energy that corresponds to the Gaussian distribution peak for donor-like states. This energy is measured from the valence band edge.

F.TFTACC specifies the name of a file containing a C-INTERPRETER function, describing the distribution of acceptor state densities as a function of energy.

F.TFTDON specifies the name of a file containing a C-INTERPRETER function, describing the distribution of donor state densities as a function of energy.

FILE.INT specifies the integrated density of states with respect to energy will be stored in the AFILE, DFILE, and TFILE files for discrete defects.

INT_LIM1 specifies the lower limit for the numerical integration of the CONTINUOUS method.

INT_LIM2 specifies the upper limit for the numerical integration of the CONTINUOUS method.

MATERIAL specifies which material from the table in Appendix B: "Material Systems" will apply to the DEFECT statements. If a material is specified, then the regions defined as being composed of that material will be affected.

NA.MIN specifies the minimum value of the acceptor trap DOS that will be used in the discrete DEFECTS model.

ND.MIN specifies the minimum value of the donor trap DOS that will be used in the discrete DEFECTS model.

NGA specifies the total density of acceptor-like states in a Gaussian distribution.

NGD specifies the total density of donor-like states in a Gaussian distribution.

NTA specifies the density of acceptor-like states in the tail distribution at the conduction band edge.

NTD specifies the density of donor-like states in the tail distribution at the valence band edge.

NUMBER or **REGION** specifies the region index to which the DEFECT statement applies.

NUMA specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.

NUMD specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.

SIGGAE specifies the capture cross-section for electrons in a Gaussian distribution of acceptor-like states.

SIGGAH specifies the capture cross-section for holes in a Gaussian distribution of acceptor-like states.

SIGGDE specifies the capture cross-section for electrons in a Gaussian distribution of donor-like states.

SIGGDH specifies the capture cross-section for holes in a Gaussian distribution of donor-like states.

SIGTAE specifies the capture cross-section for electrons in a tail distribution of acceptor-like states.

SIGTAH specifies the capture cross-section for holes in a tail distribution of acceptor-like states.

SIGTDE specifies the capture cross-section for electrons in a tail distribution of donor-like states.

SIGTDH specifies the capture cross-section for holes in a tail distribution of donor-like states.

STRUCTURE is a synonym for DEVICE.

TFILE specifies the file name where the acceptor and donor state density distributions, such as function of energy, referenced from eV will be stored. You can examine this file by using TONYPLOT.

WGA specifies the characteristic decay energy for a Gaussian distribution of acceptor-like states.

WGD specifies the characteristic decay energy for a Gaussian distribution of donor-like states.

WTA specifies the characteristic decay energy for the tail distribution of acceptor-like states.

WTD specifies the characteristic decay energy for the tail distribution of donor-like states.

X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN**, and **Z.MAX** specify the bounding box for the DEFECT statement.

TFT Example

The following statement lines specify distributed defect states which would typically be used for polysilicon.

```
DEFECTS NTA=1.E21 NTD=1.E21 WTA=0.033 WTD=0.049 \
        NGA=1.5E15 NGD=1.5E15 EGA=0.62 EGD=0.78 \
        WGA=0.15 WGD=0.15 SIGTAE=1.E-17 \
        SIGTAH=1.E-15 SIGTDE=1.E-15 SIGTDH=1.E-17 \
        SIGGAE=2.E-16 SIGGAH=2.E-15 SIGGDE=2.E-15 \
        SIGGDH=2.E-16
```

19.8: DEGRADATION

DEGRADATION specifies parameters for MOS device degradation modeling.

Syntax

```
DEGRADATION <params>
```

Parameter	Type	Default	Units
F.NTA	Char		
F.NTD	Char		
F.NTD	Char		
F.SIGMAE	Char		
F.SIGMAH	Char		
NTA	Real	1.0×10^{11}	cm^{-2}
NTD	Real	1.0×10^{10}	cm^{-2}
SIGMAE	Real	1.0×10^{17}	cm^{-2}
SIGMAH	Real	1.0×10^{17}	cm^{-2}

Description

NTA specifies the uniform acceptor-like trap density on the interface.

NTD specifies the uniform donor-like trap density on the interface.

SIGMAE specifies the acceptor-like trap capture cross section.

SIGMAH specifies the donor-like trap capture cross section.

F.NTA specifies the file name for a C-INTERPRETER function that specifies arbitrary density distribution of the acceptor-like traps on the interface.

F.NTD specifies the file name for a C-INTERPRETER function that specifies arbitrary density distribution of the donor-like traps on the interface.

F.SIGMAE specifies the name of a file containing a C-INTERPRETER function specifying the distribution of acceptor trap cross-sections.

F.SIGMAH specifies the name of a file containing a C-INTERPRETER function specifying the distribution of donor trap cross-sections.

MOS Interface State Example

```
DEGRADATION NTA=1.E-12 SIGMAE=5.E-18
```

This syntax defines a density of acceptor states uniformly distributed along the silicon-oxide interface. The trapping cross section is also defined. Traps will be filled by gate current in transient mode simulations leading to a shift in device parameters.

19.9: DOPING

DOPING specifies doping profiles either analytically or from an input file. The alias for this parameter is PROFILE.

Syntax

```
DOPING <prof>[<psp>][<bound>][<loc>][<sprea>>][OUTFILE= <fn>][<trps>]
```

Parameter	Type	Default	Units
1D.PROC	Logical	False	
1D.PROC	Logical	False	
2D.ASCII	Logical	False	
ACTIVE	Logical	True	
ANTIMONY	Logical	False	
ARSENIC	Logical	False	
ASCII	Logical	False	
ATHENA	Logical	False	
ATHENA.1D	Logical	False	
BACKDOPE	Real	none	cm^{-3}
BORON	Logical	False	
CHARACTERISTIC	Real		μm
CHEMICAL	Logical	False	
CONCENTRATION	Real	0	cm^{-3}
DEGEN	Real		
DEVICE	Char		1
DIRECTION	Character	y	
DOP.SEED	Integer	-10	
DOP.SIGMA	Real	0.0	cm^{-3}
DOP.OFFSET	Real	0	cm^{-3}
DOSE	Real		cm^{-2}
ERFC.LATERAL	Logical	False	
E.LEVEL	Real		eV
F.COMPOSIT	Character		
F.DOPING	Character		
F3.DOPING	Character		

Parameter	Type	Default	Units
FILE	Character		
GAUSSIAN	Logical	False	
IN.FILE	Character		
INDIUM	Logical	False	
INFILE	Character		
INT.LIN	Logical	False	
INT.LOG	Logical	True	
INT.OPTM	Logical	False	
JUNCTION	Real		μm
LAT.CHAR	Real		μm
MASTER	Logical	False	
MATERIAL	Character		
N.COLUMN	Integer		
N.TYPE	Logical	False	
N-TYPE	Logical	False	
N.OFFSET	Real	0.0	cm^{-3}
N.PEAK	Real	0	cm^{-3}
NAME	Character		
NET	Logical	False	
NOROLLOFF	Logical	False	
NOXROLLOFF	Logical	False	
NOYROLLOFF	Logical	False	
OUTFILE	Character		
PHOSPORUS	Logical	False	
P.COLUMN	Integer		
P.TYPE	Logical	False	
P-TYPE	Logical	False	
RATIO.LATERAL	Real	0.7	
RESISTI	Real		
OX.CHARGE	Logical	False	
REGION	Integer	All	

Parameter	Type	Default	Units
SIGN	Real		cm^2
SIGP	Real		cm^2
SLICE.LAT	Real		μm
START	Real	0	
STRUCTURE	Character		
SUPREM3	Logical	False	
TMA.SUPREM3	Logical	False	
TAUN	Real		
TAUP	Real		
TRAP	Logical	False	
UNIFORM	Logical	False	
WIDTH	Real	width of structure	μm
X1	Real	0	μm
X2	Real	0	μm
XY	Logical	True	
XZ	Logical	False	
X.CHAR	Real		μm
X.COLUMN	Integer		
X.COMP	Logical	False	
X.DIR	Logical	False	
X.ERFC	Logical	False	
X.LEFT	Real	left of structure	μm
X.MAX	Real		μm
X.MIN	Real		μm
X.RIGHT	Real	right of structure	μm
XY.RATIO	Real		0.7
Y.BOTTOM	Real	bottom of structure	μm
Y.CHAR	Real		μm
Y.COLUMN	Integer		
Y.JUNCTI	Real		μm
Y1	Real	0	μm
Y2	Real	0	μm

Parameter	Type	Default	Units
YX	Logical	True	
YZ	Logical	False	
Y.COMP	Logical	False	
Y.DIR	Logical	False	
Y.MAX	Real		μm
Y.MIN	Real		μm
Y.TOP	Real	top of structure	
Z1	Real	0	μm
Z2	Real	0	μm
ZY	Logical	False	
ZX	Logical	False	
Z.BACK	Real		
Z.DIR	Logical	False	
Z.FRONT	Real		
Z.MIN	Real		μm
Z.MAX	Real		μm
ZLAT.CHAR	Real		
ZSLICE.LAT	Real		
ZRATIO.LAT	Real	0.7	

Description

The DOPING statement is used to define doping profiles in the device structure. Typically a sequence of DOPING statements is given each building on the others.

OUTFILE specifies the name of an output file for use with REGRID. The first DOPING statement should use this parameter to specify a filename. All doping information from the first DOPING statement and all subsequent DOPING statements in the input file are saved to this file. The REGRID statement can read this file and interpolate doping on the new grid.

Note: The file from OUTFILE cannot be used in TONYPLOT or in the MESH statement.The SAVE command should be used after all of the DOPING commands required to save a file for plotting the doping profile.

Statement Applicability Parameters

DEVICE specifies which device the statement applies to in the MIXEDMODE simulation. The synonym for this parameter is STRUCTURE.

MATERIAL restricts the applicability of the statement to regions of the specified material.

NAME restricts the applicability of the statement to regions with the specified name.

REGION restricts the applicability of the statement to regions with the specified region number.

STRUCTURE is a synonym for DEVICE.

Note: If you don't specify the DEVICE, MATERIAL, NAME and REGION parameters, the DOPING statement will apply to all regions.

Analytical Profile Types

These parameters specify how ATLAS will generate a doping profile from analytical functions.

DOP.SIGMA specifies the variance for random gaussian dopant distribution.

DOP.SEED specifies a seed value for random gaussian dopant distribution.

GAUSSIAN specifies the use of a gaussian analytical function to generate the doping profile. If GAUSSIAN is specified, the following parameters must also be specified:

- Polarity parameters N.TYPE or P.TYPE
- One of the following groups of profile specifications:
  - Group 1:CONCENTRATION and JUNCTION
  - Group 2:DOSE and CHARACTERISTIC
  - Group 3:CONCENTRATION and CHARACTERISTIC

UNIFORM specifies the use of uniform (constant) analytical functions to generate the doping profile. If Uniform is specified, the N.TYPE, P.TYPE, and CONCENTRATION parameters must be specified. Doping is introduced into a box defined by the boundary parameters (see the "Boundary Conditions" Section on page 19-17). The box by default includes the entire region.

F.DOPING specifies the name of a file containing a C-INTERPRETER function specifying the spatial distribution of dopants.

F3.DOPING specifies the name of a file containing a C-INTERPRETER function specifying the spatial distribution of dopants for a 3D device.

File Import Profile Types

These parameters specify how ATLAS will generate a doping profile from a file. Files can be user-defined or from process simulation.

1D.PROC specifies is an alias for TMA.SUPREM3.

2D.ASCII specifies that a 2D doping profile, which is defined on a rectangular Cartessian grid, should be loaded from a file specified by INFILE. 2D.ASCII must be specified along with either the N.TYPE, P.TYPE or NET parameters. This first column of the file should contain the x coordinates. The second column should contain the y coordinates. The third column should contain the doping data.

ASCII has two separate meanings. The first meaning is that it specifies the file type as ASCII when it's combined with other format parameters. The second meaning is when this parameter is used alone, it specifies ASCII data files containing concentration versus depth information. The alias for this parameter is 1D.PROC.

In the second meaning, this parameter must be written in the form:

```
ASCII INFILE=<filename>
```

where filename is the name of the ASCII input file. The data file must be in the following format:

```
depth    concentration
depth    concentration
```

```
depth    concentration
          ...
```

where depth is specified in μm and concentration is specified in cm^{-3}. An input file name, a dopant type, and boundary parameters must be specified. Positive concentrations are assumed to be n-type and negative concentrations are assumed to be p-type unless the N.TYPE or P.TYPE parameters are used.

ATHENA.1D specifies that the doping file is a ATHENA 1D export file. This parameter acts in a similar way to the SSUPREM3 parameter.

ATHENA reads 2D doping information from ATHENA standard structure file (SSF) or PISCES-II format files. The PISCES-II format is an obsolete file format. Doping information obtained from this file will be added to each point of the current ATLAS mesh. If points in the ATLAS mesh do not coincide with points in the ATHENA mesh, doping for ATLAS mesh points will be interpolated from ATHENA doping information. If this profile type is used, the INFILE parameter must also be specified.

Note: The X.STRETCH function available in previous versions of ATLAS has been replaced by similar more powerful functions in DEVEDIT. This feature should no longer be used in ATLAS.

FILE is an alias for INFILE.

IN.FILE is a synonym for INFILE. The alias for this parameter is FILE.

INFILE specifies the name of the appropriate input file. The synonym for this parameter is IN.FILE.

MASTER specifies that the INFILE is written in the Silvaco standard structure file (SSF) format. This file format is the default output format of ATHENA and SSUPREM3. This parameter is typically combined with the SSUPREM3, ATHENA 1D, or ATHENA parameters. If neither of these are used the default is SUPREM3.

N.COLUMN specifies which column of an 2D ASCII table corresponds to the net donor concentration when 2D.ASCII is specified.

P.COLUMN specifies which column of an 2D ASCII table corresponds to the net acceptor concentration when 2D.ASCII is specified.

SUPREM3 specifies the INFILE was produced by SSUPREM3 in standard structure file (SSF) format or binary or an ASCII export format. If this profile type is used, an input file name, a dopant, and boundary parameters must be specified. When SSUPREM3 produces an output file, the doping profiles are stored by dopant. Therefore, a dopant parameter should be specified in order to import the correct doping profile into ATLAS. If a specific dopant is not specified the total donors and acceptor concentrations are loaded.

TMA.SUPREM3 specifies that the file specified by INFILE is in TMA SUPREM3 binary format. The alias for this parameter is 1D.PROC.

Note: Files containing 1D doping profiles can be loaded into BLAZE, BLAZE3D, DEVICE3D, or S-PISCES. Files containing 2D doping profiles can only be loaded into S-PISCES.

X.COLUMN specifies which column of an 2D ASCII table corresponds to the x coordinate value when 2D.ASCII is specified.

Y.COLUMN specifies which column of an 2D ASCII table corresponds to the y coordinate value when 2D.ASCII is specified.

Dopant Type Specification Parameters

These parameters give information about the dopant species or type to be used in the specified profile. Different profile types require different profile specifications.

ACTIVE specifies that for the dopant specified the active concentration as opposed to the chemical concentration is added. This is true by default. Files from ATHENA or SSUPREM3 contain both active and chemical concentrations for each dopant.

ANTIMONY specifies that antimony dopant information be extracted from an imported file.

ARSENIC specifies that arsenic dopant information be extracted from an imported file.

BORON specifies that boron dopant information be extracted from an imported file.

CHEMICAL specifies that the chemical concentration (as opposed to the active concentration) will be read from the imported file. This is generally not advisable.

DOP.OFFSET subtracts a background doping value from the ATHENA or SSUPREM3 doping. The alias for this parameter is N.OFFSET.

E.LEVEL sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge. For donors, it is relative to the valence band edge.

INDIUM specifies that indium dopant information be extracted from an imported file.

NET specifies that net doping information be extracted from an imported file. This is usually not advisable. It is better to use several DOPING statements to extract data dopant by dopant from a file.

N.OFFSET is an alias for DOP.OFFSET.

N.TYPE, **N-TYPE**, **DONOR** specifies an n-type or donor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.

P.TYPE, **P-TYPE**, **ACCEPTOR** specifies a p-type or acceptor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.

PHOSPHORUS specifies that phosphorus dopant information be extracted from an imported file.

TRAP specifies that the dopant concentration is to be treated as a trap state density.

OX.CHARGE specifies a fixed oxide charge profile. Oxide charge can only be placed in any insulator region. The N.TYPE/P.TYPE parameters are not used hence a negative concentration implies a negative charge.

X.COMP specifies a profile of composition fraction **x** as defined in Appendix B: “Material Systems”. This profile can be used to change the composition fraction of cations in ternary and quaternary materials over a spatial distribution.

Y.COMP specifies a profile of composition fraction **y** as defined in Appendix B: “Material Systems”. This profile can be used to change the composition fraction of anions in ternary and quaternary materials over a spatial distribution.

RESISTI can be used to specify resistivity, which is converted to carrier concentration in silicon (i.e., this parameter replaces the CONC parameter). This conversion uses tables of resistivity versus concentration for donors and acceptors. These tables currently make no distinction between specific changes of the same type. The Arora mobility model is the basis of these tables.

Vertical Distribution Parameters

CHARACTERISTIC specifies the principal characteristic length of the implant. For Gaussians, the characteristic length is equal to the square root of two times the standard deviation. If this parameter is left unspecified, the principal characteristic can be computed from the values of the

- Polarity Parameters

- Boundary Parameters
- Concentration and Junction parameters

The alias for this parameter is Y.CHAR.

CONCENTRATION specifies the peak concentration when a Gaussian profile is used. If this parameter is not specified, peak concentration may be computed from the values of the polarity, boundary, DOSE, or RESISTI, CHARACTERISTIC concentrations. When a uniform profile is specified, the CONCENTRATION parameter sets the value of the uniform doping level. Concentrations must be positive. The alias for this parameter is N.PEAK.

DOSE specifies the total dose for a Gaussian profile.

JUNCTION specifies the location of a p-n junction within the silicon region of a Gaussian profile. When JUNCTION is specified, the characteristic length is computed by examining the doping at a point halfway between the end of the constant box and the given depth. The JUNCTION location is evaluated considering all previous DOPING statements only. This means that in some cases the order of DOPING statements is important.

N.PEAK is an alias for CONCENTRATION.

PEAK specifies the depth location of the peak doping in a Gaussian profile.

Y.CHAR is an alias for CHARACTERISTIC.

Y.JUNCTI is an alias for JUNCTION.

Location Parameters

DIRECTION specifies the axis along which a one-dimensional profile is directed in a two-dimensional device (x or y). DIR=y will typically be used for implanted profiles.

REGION specifies the region number where doping is to be added.

START specifies the depth in the y direction where the profile should start.

Lateral Extent Parameters

These parameters must be specified when a 1-D doping profile type is used (MASTER, GAUSSIAN, or ASCII). These boundary parameters set the doping boundaries before applying lateral spreading. This is equivalent to setting implant mask edges.

WIDTH specifies the extent of the profile in the x-direction. Specifying WIDTH is equivalent to specifying X.MAX such that X.MAX=X.MIN+WIDTH.

X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN and **Z.MAX** specify the x, y and z bounds of a rectangular shaped region or box in the device. The dopant profile within this box will be constant with a density equal to the value specified by the CONC parameter. Outside this box the profile decreases from the peak, CONC, with distance, from the box along the principal axes. The relationship between the concentration, outside the box, to distance will depend upon the profile type as specified by the GAUSSIAN, MASTER, ATHENA, ATLAS, and UNIFORM parameters.

X.LEFT, X.MIN specifies the left boundary of a vertical 1-D profile.

X.RIGHT, X.MAX specifies the right boundary of a vertical 1-D profile.

Y.BOTTOM, Y.MAX specifies the bottom boundary of a horizontal 1-D profile.

Y.TOP, Y.MIN specifies the top boundary of a horizontal 1-D profile.

Z.BACK, **Z.MIN** specifies the back boundary of a z directed 1-D or 2-D profile.

Z.FRONT, Z.MAX specifies the front boundary of a z directed 1-D or 2-D profile.

Lateral Distribution Parameters

These parameters specify how a vertical 1-D profile is extended outside the box defined by the lateral extent parameters.

BACKDOPE specifies the value to which the doping profile specified from the 1D profile will roll-off to outside its lateral and vertical extents. If this value is not specified, then the last doping value in the ASCII file is used as the background doping level, regardless of windowing in the y-direction. If NOXROLLOFF is used, then BACKDOPE will be ignored for the x-direction. If NOYROLLOFF is used, then BACKDOPE will be ignored for the y-direction. If NOROLLOFF is used, then BACKDOPE will be completely ignored.

ERFC.**LATERAL** specifies an error function used for lateral spreading. If two-dimensional spreading parameters are used in conjunction with a Gaussian profile, the lateral impurity profile may be transformed into an error function. The alias for this parameter is X.ERFC.

LAT.**CHAR** specifies the characteristic length of the lateral profile. If this parameter is not specified, the characteristic length is defined by:

$$CL = RL \times OCL \qquad 19\text{-}3$$

where:

- CL is the lateral characteristic length in the x direction.
- RL is the value of RATIO.LATERAL.
- OCL is the characteristic length of the original profile in the y direction.

The alias for this parameter is X.CHAR.

NOXROLLOFF causes the doping level to abruptly change to zero outside the x-limits.

NOYROLLOFF causes the doping level to abruptly change to zero outside the y-limits.

NOROLLOFF is the same as setting both NOXROLLOFF and NOYROLLOFF.

RATIO.**LATERAL** is the ratio of characteristic lengths in the x and y directions.

SLICE.**LAT** specifies the point at which the doping is examined to compute the characteristic length of a Gaussian profile after JUNCTION has been specified. The default for this parameter is a point halfway between the end of the constant box and the given depth.

X.CHAR is an alias for LAT.CHAR.

X.ERFC is an alias for ERFC.LAT.

XY.RATIO is an alias for RATIO.LATERAL.

ZLAT.**CHAR** specifies the characteristic length of the lateral profile in the z direction. See also LAT.CHAR.

ZRATIO.**LATERAL** is used analogously to RATIO.LATERAL but applies to lateral spreading in the z direction. See also LAT.CHAR.

ZSLICE.**LAT** is similar to SLICE.LAT but applies to profiles in the z direction.

Trap Parameters

E.**LEVEL** sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge, for donors it is relative to the valence band edge.

DEGEN.**FAC** specifies the degeneracy factor of the trap level used to calculate the density.

SIGN specifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level.

Note: See Section 19.52: "TRAP" for more information on each of these parameters

Angled Distribution Parameters

X1,**Y1**,**X2**,**Y2** specify the X and Y coordinates of the ends of the line segment describing the location of the specified angled profile.

X.DIR and **Y.DIR** specify the direction in which the angled profile is extended.

In DEVICE3D, (see Chapter 6: "3D Device Simulator" for more information about this simulator), dopants can be added along angled segments in the X-Y plane. The start and ending coordinates of the line segment are defined by the X1, Y1, X2, and Y2 parameters. You can then specify whether a 2D profile is extended from the line segment in either the X or the Y direction. To specify it, set the X.DIR or Y.DIR parameter. You can then specify a 2D doping profile in the same DOPING statement. The profile can be an analytic, SUPREM, ASCII, or SUPREM4.

The dopants are placed relative to the defined line segment, according to the setting of X.DIR or Y.DIR. If X.DIR is specified, then the effective Y coordinate of the profile is the device Z coordinate and the effective X coordinate of the profile is the distance in the X direction from the center of the line segment. No dopants are added if the device Y coordinate is outside of the Y coordinates of the line segment. If Y.DIR is specified, then the effective Y coordinate of the profile is the device Z coordinate and the effective X coordinate of the profile is the distance in the Y direction from the of the line segment. No dopants are added if the device X coordinate is outside the X coordinates of the line segment.

Analytical Doping Definition Example

This example describes a 1.0μm n-channel MOSFET using Gaussian source and drain profiles. The lateral extent of the source is given by X.RIGHT=2. This corresponds to the mask edge for the implant. Sub-diffusion is determined by an error function based on the RATIO.LAT and JUNCTION parameters. For both source and drain, the n+ doping is added to the uniform p-well concentration to ensure a junction depth of 0.3μm.

```
DOP UNIF CONC=1E16 P.TYPE

DOP GAUSS CONC=9E19 N.TYPE X.RIGHT=2 JUNC=0.3 RATIO.LAT=0.6 ERFC.LAT

DOP GAUSS CONC=9E19 N.TYPE X.LEFT=3  JUNC=0.3 RATIO.LAT=0.6 ERFC.LAT
```

1D ATHENA or SSUPREM3 Interface Example

This example reads a 1D ATHENA bipolar profile and adds it to a uniform substrate concentration. The base and emitter doping are loaded from the same file by specifying the impurity required for each area (boron in the base and arsenic in the emitter).

The DOPOFF parameter is used to subtract the substrate arsenic dopant out of the 1-D profile that is loaded since this dopant was already specified in the substrate doping line.

Versions of SSUPREM3 later than 5.0 use standard structure files as default when saving data. These can be loaded in ATLAS with the syntax below by replacing ATHENA.1D with SSUPREM3.

```
#   SUBSTRATE

DOP REGION=1 UNIF CONC=1E16 N.TYPE

#   BASE

DOP REGION=1 MASTER ATHENA.1D BORON RATIO.LAT=0.7 INF=bipolar.exp
```

```
#    EMITTER
DOP REGION=1 MASTER ATHENA.1D ARSENIC RATIO.LAT=0.6 \
    INF=bipolar.exp X.LEFT=12.0 X.RIGHT=13.0 DOPOFF=1e16
```

Athena Doping Interface Example

This example demonstrates how to use an SSF format ATHENA file to interpolate doping onto a ATLAS grid and save the doping information for subsequent regrid operations.This is an alternative to the ATHENA/ATLAS interface, which is described in Chapter 2: “Getting Started with ATLAS”, Section 2.6.1: “Interface From ATHENA”.

```
DOPING ATHENA MASTER INFILE=NMOS.DOP OUTFILE=NMOS.DOP
REGRID DOPING ABS LOG RATIO=4 OUTFILE=NMESH1.STR DOPFILE=NMOS.DOP
```

3D Doping Definition Example

The following example illustrates the formation of a Gaussian highly doped n-type area in a three-dimensional structure.

```
DOPING GAUSS N.TYPE CONC=1e20 PEAK=0.0 CHAR=0.2 X.LEFT=0.5 \
X.RIGHT=1.0 Z.LEFT=0.5 Z.RIGHT=1.0
```

3D Doping From ASCII 1D File

You can read a 1D doping profile from an ASCII file and apply it to the entire or part of a 3-D device. The X.DIR, Y.DIR, or Z.DIR parameters specify the direction of the profile as applied in the 3-D device. The starting position of the doping profile will be set by the relevant X.MIN, Y.MIN, or Z.MIN parameter. The ending position of the doping profile will be the minimum of the relevant X.MIN, Y.MIN, or Z.MIN plus the spatial extent of the doping profile in the ASCII file, or the X.MAX, Y.MAX, Z.MAX parameter or a physical boundary of the device. You can specify a general parallelogram in the plane perpendicular to the direction of doping direction. To specify it, use the appropriate combination of X1, X2, Y1, Y2, Z1, Z2, X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN, Z.MAX. For example, we consider the Y.DIR parameter specified. We can then specify a parallelogram in the X-Z plane as shown in Figure 19-2.

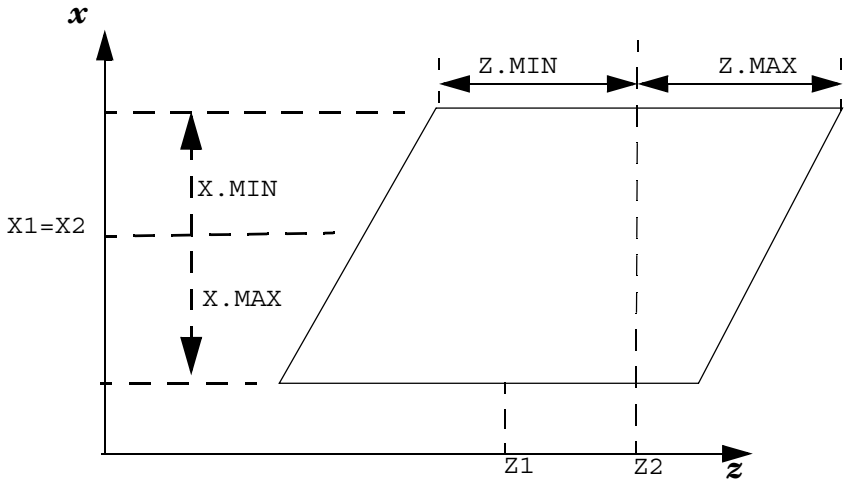


Figure 19-2: Parallelogram Geometry Example

In the figure, two sides are parallel to the Z-axis. Z1 and Z2 specify the Z-coordinate of the midpoints of these two sides. Z.MIN and Z.MAX together specify the lengths of the sides, and Z.MIN will be negative and equal to -Z.MAX. (In fact, the positions are evaluated as Z1+Z.MIN and Z1+Z.MAX for one side and Z2+Z.MIN and Z2+Z.MAX for the other. Therefore, having Z1 and Z2 and so on refer to the side midpoint is just a convenient convention.)

You must specify X1 and X2 as having the same value, and specify X.MIN and X.MAX the side lengths the same way as for Z.MIN and Z.MAX.

Setting Z1 and Z2 to the same value will result in a rectangle. By then changing X1 and X2 so they are not equal will result in a parallelogram with sides parallel to the x-axis. If a different direction of doping variation is defined, then a parallelogram can be set up in the plane perpendicular to it in analogously to the above example.

Outside the specified parallelogram the doping level will be either the value of the last point in the ASCII file, or the value specified by the BACKDOPE parameter. A smooth transition of the doping from the value inside the parallelogram to the value outside can be applied elsewhere using the LAT.CHAR or RATIO.LAT parameters.

X1, **X2**, **Y1**, **Y2**, **Z1**, **Z2**, by convention, specify the midpoints of the sides making up the parallelogram.

X.MIN, **X.MAX** determine the X-coordinate of the start and end points of the placement of the doping profile read in from an ASCII file (if you specify X.DIR). Otherwise, they specify the lateral extent of the parallelogram sides defining the location of the doping in a plane perpendicular to the specified direction of the doping variation. In this case, they are added to X1 and to X2 to determine the coordinates of the ends of sides.

Y.MIN, **Y.MAX**, **Z.MIN**, **Z.MAX** are the same as X.MIN and X.MAX except they apply to the y and z-directions respectively.

Example:

```
doping ascii inf=ydop.dat y.dir n.type x1=6.0 x2=4.0 x.min=-2 x.max=2
z.min=3.0 z.max=7.0 y.min=2.0 y.max=10.0 lat.char=0.005 backdope=0.0
```

This will apply the doping profile specified in ydop.dat as n-type doping in the y-direction between the positions y=2 microns and y=10 microns (unless the length range in ydop.dat is less than this distance). In the x-z plane, it will be applied to a parallelogram with sides parallel to the x-direction of length 4 microns. The z-coordinates of these sides are 3.0 microns and 7.0 microns. The doping will transition from the value inside the parallelogram to 0.0 outside on a length scale of 0.005 microns.

Using 2D ATHENA Master Files for Doping in ATLAS

In a 2D ATHENA master file, the doping profile is defined over a box in two dimensions. This doping is put over 2D-sections normal to the plane of a parallelogram to be defined in the doping statement. This parallelogram must lie in one of the xy, yz, and xz planes and have two edges parallel to one of the coordinate axes.

As the mesh in the ATHENA master file doesn't necessarily coincide with the three-dimensional mesh in ATLAS, an interpolation routine is required to import the doping in ATLAS. You can then choose between a linear and a logarithmic interpolation algorithm.

Here's a list of the parameters that are used for this kind of doping.

INT.LIN specifies that a linear interpolation is to be used to import doping from ATHENA into ATLAS.

INT.LOG specifies that the ATHENA doping is imported in ATLAS by using a logarithmic interpolation algorithm.

INT.OPTM enables an optimized interpolation routine, which attempts to reduce CPU time due to the doping interpolation.

LAT.CHAR defines the characteristic length, CL, of the lateral spreading (in the y>y2 and y<y1 planes). If RATIO.LAT is used instead, then the characteristic length is assumed to come from this parameter by the height of the parallelogram (i.e., CL=RL X OCL, OCL being the height of the parallelogram (y2-y1 in this case). If both LAT.CHAR and RATIO.LAT aren't specified, then no lateral spreading is done.

XY specifies that the parallelogram containing doping in ATLAS3D is in the xy plane.

YX is a synonym for XY.

YZ specifies that a parallelogram containing doping in ATLAS3D is in the yz plane.

ZY is a synonym for YZ.

XZ specifies that a parallelogram containing doping in ATLAS3D is in the xz plane.

ZX is a synonym for ZX.

X1, **X2**, **Y1**, **Y2**, **Z1**, **Z2** specify an average segment in one of the three coordinate planes defining the orientation of the parallelogram.

X.DIR, Y.DIR, **Z.DIR** specify that a parallelogram containing doping in ATLAS3D has two edges in the x direction, y direction, and z direction.

X.MIN, **X.MAX** define the minimum and maximum lateral extent of a parallelogram along the x direction lying in the xy or zx plane, starting from its average segment (they must be specified with X1 and X2). If a parallelogram is defined in the yz plane, X.MIN can then be used to specify the initial coordinate to start the doping along the x direction.

Y.MIN, **Y.MAX** define the minimum and maximum lateral extent of a parallelogram along the y direction lying in the xy or yz plane, starting from its average segment (they must be specified with Y1 and Y2). If a parallelogram is defined in the zx plane, Y.MIN can then be used to specify the initial coordinate to start the doping along the y direction.

Z.MIN, **Z.MAX** define the minimum and maximum lateral extent of a parallelogram along the z direction lying in the yz or zx plane, starting from its average segment (they must be specified with Z1 and Z2). If a parallelogram is defined in the xy plane, Z.MIN can then be used to specify the initial coordinate to start the doping along the z direction.

The following three cases, which correspond to the parallelograms in the xy, yz and zx planes, describe how the doping from the ATHENA master file is put into the ATLAS structure .

First Case: Parallelogram In The XY Plane

SUB-CASE A: Parallelogram along the x direction.

Sections of the ATLAS structure in zx planes are considered, which intersect for y1<y<y2, the segment (x_a, x_b) inside the parallelogram. Each of these sections defines a box where the 2D doping is imported from ATHENA. Particularly, all the coordinates in ATHENA are scaled and translated so that the edge (x_c, x_d) of the ATHENA box containing the doping coincide with the segment (x_a, x_b) of the ATLAS box. This is done for all the sections inside the parallelogram. See Figure 19-3.

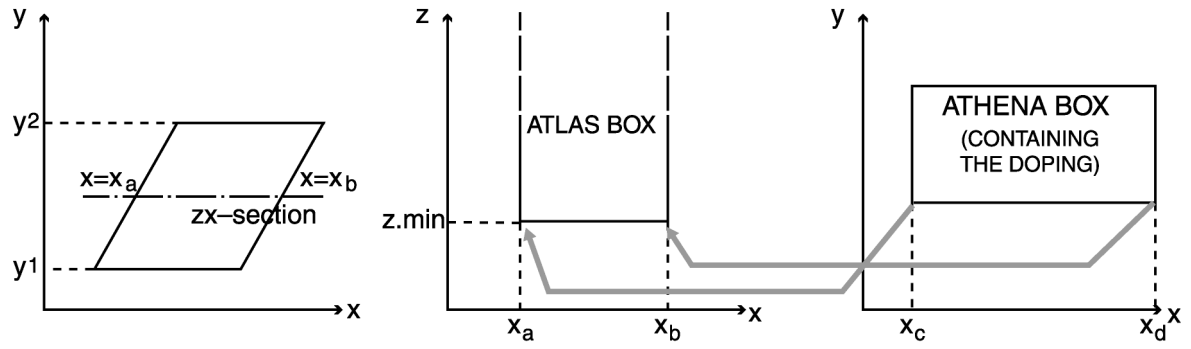


Figure 19-3: Parallelogram in the XY plane in the X direction and doping from the ATHENA2D master file

Examples:

```
doping athena master inf=athena.str boron xy \
x1=2.0 x2=3.0 y1=1.5 y2=4.0 x.dir \
x.min=-2.0 x.max=1.0 z.min=0.6 lat.char=0.05 int.log int.optm
```

For the z coordinate, `z.min` is used to specify the minimum value where to start putting the doping into the zx-sections (which properly defines the segment (x_a, x_b) in this plane).

In addition, a lateral spreading can be partially accomplished by extending the parallelogram into the planes, y>y2 and y<y1, where the doping is spread out according to a Gaussian law.

SUB-CASE B: Parallelogram along the Y direction

Sections of the ATLAS structure are considered in the yz planes. A transformation of coordinates in ATHENA is accomplished in order to place the ATHENA segment (y_c, y_d) into the ATLAS one (y_a, y_b). See Figure 19-4.

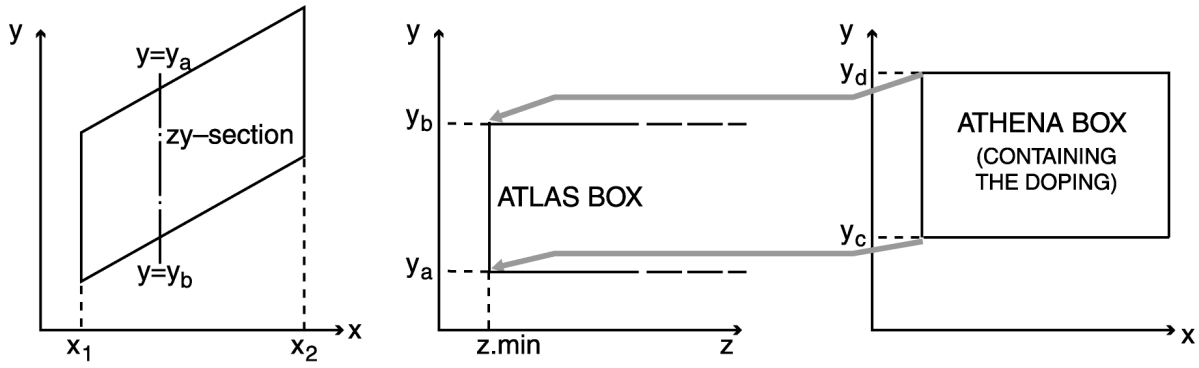


Figure 19-4: Parallelogram in the XY plane in the Y direction and doping from the ATHENA2D master file

Examples:

```
doping athena master inf=athena.str boron xy \
x1=2.0 x2=3.0 y1=1.5 y2=4.0 y.dir \
y.min=-2.0 y.max=1.0 z.min=0.6 lat.char=0.05 int.lin int.optm
```

Second Case: YZ Plane

The yz plane containing the parallelogram is defined by the YZ parameter (or ZY) in the DOPING statement.

If parallelograms along the z direction are defined (specify Z.DIR in doping statement), the doping from ATHENA is put over sections in the zx planes. If the parallelograms along the y direction are defined (using Y.DIR in DOPING statement), the doping from ATHENA is put over sections in the xy planes.

The minimum x coordinate in ATLAS to start adding doping can be specified by X.MIN in DOPING statement.

Examples:

```
doping athena master inf=athena.str boron zy z.dir \
z1=0.3 z2=0.6 y1=0.35 y2=0.25 z.min=-0.2 z.max=0.6 \
x.min=0.2 ratio.lat=0.05


doping athena master inf=athena.str boron zy y.dir \
z1=0.3 z2=0.6 y1=0.35 y2=0.25 y.min=-0.2 y.max=0.6 \
x.min=0.2 ratio.lat=0.05
```

Third Case: ZX Plane

The zx plane containing the parallelogram is defined by the ZX parameter (or XZ) in DOPING statement. If parallelograms along the z direction are defined (using Z.DIR in the DOPING statement), the doping from ATHENA is placed over sections in the yz planes.

If the parallelograms along the x direction are defined (using X.DIR in DOPING statement), the doping from ATHENA is placed over the sections in the xy planes.

The minimum y coordinate in ATLAS to start adding doping can be specified by Y.MIN in doping statement.

Examples:

```
doping athena master inf=athena.str boron int.lin \
xz z.dir z1=0.45 z2=0.55 x1=0.2 x2=0.7 x.min=-0.4 \
x.max=0.2 y.min=0.0 lat.char=0.02
doping athena master inf=athena.str boron int.lin \
xz x.dir z1=0.4 z2=0.8 x1=0.2 x2=0.7 z.min=-0.4 \
z.max=0.2 y.min=0.5 ratio.lat=0.5
```

19.10: ELECTRODE

ELECTRODE specifies the locations and names of electrodes in a previously defined mesh.

Syntax

```
ELECTRODE NAME=<en> [NUMBER=<n>] [SUBSTRATE] <pos> <reg>
```

Parameter	Type	Default	Units
BOTTOM	Logical	False	
IX.HIGH	Integer	right side of structure	
IX.LOW	Integer	left side of structure	
IX.MAX	Integer	right side of structure	
IX.MIN	Integer	left side of structure	
IY.MAX	Integer	bottom side of structure	
IY.MIN	Integer	top side of structure	
IY.HIGH	Integer	bottom of structure	
IY.LOW	Integer	top of structure	
IZ.MAX	Integer		
IZ.MIN	Integer		
IZ.HIGH	Integer		
IZ.LOW	Integer		
LEFT	Logical	False	
LENGTH	Real	length of structure	μm
MATERIAL	Character	Contact	
NAME	Character		
NUMBER	Integer	defined #(electrodes)+ 1	
RIGHT	Logical	False	
SUBSTRATE	Logical	False	
THERMAL	Logical	False	
TOP	Logical	False	
X.MAX	Real	right side of structure	μm
X.MIN	Real	left side of structure	μm
Y.MAX	Real	Y.MIN	μm
Y.MIN	Real	top of the structure	μm
Z.MIN	Real		μm
Z.MAX	Real		μm

Description

MATERIAL specifies a material for the electrode (see Appendix B: “Material Systems”, Table B-1). This material will be displayed in TONYPLOT. The electrode material can also be used to define the electrode thermal characteristics (thermal conductivity) and optical characteristics (complex index of refraction). Setting the material here does not apply any electrical property such as workfunction to the terminal. All electrical properties of electrodes are set on the CONTACT statement.

NAME specifies an electrode name. The electrode name can be referenced by other ATLAS statements to modify characteristics of the specified electrode. For reference by the CONTACT or THERMALCONTACT statements any valid character string can be used and properly cross-referenced. But when setting voltages, currents and charge from the SOLVE statement certain electrode names are recognized in a simplified syntax. By prepending the electrode name with "V" for voltage, "I" for current and "Q" for charge, you can directly and conveniently set the electrode bias, current or charge respectively. For example:

```
SOLVE VGATE=1.0
```

can be used to assign 1 volt bias to the electrode named "GATE". In such a manner, the following list of names can be used to set voltage, current or charge.

- GATE
- FGATE
- CGATE
- NGATE
- PGATE
- VGG

The following list of names can be used to assign only voltage or current.

- DRAIN
- SOURCE
- BULK
- SUBSTRATE
- EMITTER
- COLLECTOR
- BASE
- ANODE
- CATHODE
- WELL
- NWELL
- PWELL
- CHANNEL
- GROUND
- NSOURCE
- PSOURCE
- NDRAIN
- PDRAIN
- VDD
- VSS
- VEE
- VBB

- VCC

NUMBER specifies an electrode number from 1 to 50. Electrode numbers may be specified in any order. If NUMBER is not specified, electrodes will be automatically numbered in sequential order. This parameter cannot re-number electrodes already defined in ATLAS or other programs.

pos is one of the position parameters described below.

reg is a set of the region parameters described on the next page.

SUBSTRATE places the specified electrode at the bottom of the device and names the electrode, substrate.

THERMAL specifies that the electrode is treated as a boundary condition for heatflow simulation using GIGA.

Position Parameters

BOTTOM or **SUBSTRATE** specifies that the electrode is positioned along the bottom of the device.

LEFT specifies that the electrode starts at the left-hand edge of the device. The electrode will be positioned from left to right along the top of the device.

RIGHT specifies that the electrode starts at the right-hand edge of the device. The electrode will be positioned from right to left along the top of the device.

TOP specifies that the electrode is positioned along the top of the device.

Region Parameters

Device coordinates may be used to add regions to both rectangular and irregular meshes. In either case, boundaries must be specified with the X.MAX**,** X.MIN, Y.MAX, and Y.MIN parameters.

LENGTH specifies the length of the electrode in the x direction. It is not necessary to specify X.MIN, X.MAX, and LENGTH. If two of these parameters are specified, the value of the third parameter will be calculated.

X.**MAX** specifies the maximum x-boundary of the electrode.

X.**MIN** specifies the minimum x-boundary of the electrode.

Y.**MAX** specifies the maximum y-boundary of the electrode.

Y.**MIN** specifies the minimum y-boundary of the electrode.

Z.**MIN** specifies the minimum z-boundary of the electrode.

Z.**MAX** specifies the maximum z-boundary of the electrode.

Note: If an electrode has been shortened to fit the current mesh, a warning message will be generated by ATLAS. Electrode placement can only occur at previously defined mesh nodes.

Grid Indices

As an alternative to the region parameters, you can use grid indices to define a region only when the mesh is rectangular although these parameters are not recommended. To define a region with a rectangular mesh:

1. Use the X.MESH and Y.MESH statements to specify grid indices.

2. Use the IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters to specify x and y values.

IX.**HIGH** specifies the maximum x-value of the grid index. The alias for this parameter is IX.MAX.

IX.**LOW** specifies the minimum x-value of the grid index. The alias for this parameter is IX.MIN

IY.**HIGH** specifies the maximum y-value of the grid index. The alias for this parameter is IY.MAX.

IY.**LOW** specifies the minimum y-value of the grid index. The alias for this parameter is IX.MIN.

Nodes, which have x and y grid indices, between IX.LOW and IX.HIGH and between IY.LOW and IY.HIGH are designated electrode nodes. Normally, horizontal planar electrodes will be used. In this case, IY.LOW equals IY.HIGH.

IZ.**HIGH** specifies the maximum z-value of the grid index. The alias for this parameter is IZ.MAX.

IZ.**LOW** specifies the minimum z-value of the grid index. The alias for this parameter is IX.MIN.

IX.MAX, **IX.MIN**, **IY.MIN**, **IZ.MAX**, **IZ.MIN** are aliases for IX.HIGH, IX.LOW, IY.HIGH, IY.LOW, IZ.HIGH, and IZ.LOW.

MOS Electrode Definition Example

This example defines electrodes for a typical MOS structure.

```
ELEC X.MIN=0.5 LENGTH=0.25 NAME=gate

ELEC LENGTH=0.25 Y.MIN=0 LEFT NAME=source

ELEC LENGTH=0.25 Y.MIN=0 RIGHT NAME=drain

ELEC SUBSTRATE
```

3D Electrode Definition Example

The following example illustrates electrode definition for a 3-D structure.

```
ELECTRODE NAME=ANODE X.MIN=0.5 X.MAX=1.0 \

          Z.MIN=0.5 Z.MAX=1.0
```

Note: In ATLAS, it is preferred to refer to ELECTRODES by name rather than number. Some functions, however, may require the electrode number. The syntax, MODELS PRINT, can be used to echo electrode numbers to the run-time output.

19.11: ELIMINATE

ELIMINATE terminates mesh points along lines in a rectangular grid defined within ATLAS in order to reduce the local mesh density.

Syntax

```
ELIMINATE X.DIRECTION|Y.DIRECTION [<boundary>]
```

Parameter	Type	Default	Units
COLUMNS	Logical	False	
IX.LOW	Integer		
IX.HIGH	Integer		
IY.LOW	Integer		
IY.HIGH	Integer		
ROWS	Logical	False	
X.MIN	Real		μm
X.MAX	Real		μm
Y.MIN	Real		μm
Y.MAX	Real		μm

Description

The ELIMINATE statement is used to remove points along every other line within the chosen range. Successive eliminations of the same range remove points along every fourth line. For horizontal elimination, the vertical bounds should be decreased by one at each re-elimination of the same region. For vertical elimination, the horizontal bounds should be decreased by one at each re-elimination of the same region.

ROWS or **X.DIR** eliminates points along horizontal lines.

COLUMNS or **Y.DIR** eliminates points along vertical lines.

Boundary Parameters

X.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** specify the location of the boundaries of an area in coordinates, where the elimination is applied.

The following are provided for backward compatibility only. Their use is not recommended.

IX.**HIGH** specifies the mesh line number high boundary in the x direction.

IX.**LOW** specifies the mesh line number low boundary in the x direction.

IY.**HIGH** specifies the mesh line number high boundary in the y direction.

IY.**LOW** specifies the mesh line number low boundary in the y direction.

Substrate Mesh Reduction Example

This example removes vertical points between the depth of 10μm and 20μm.

```
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=8
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=7
```

Note: In some cases, applications of the ELIMINATE statement can cause internal inconsistencies in the mesh. When this occurs, an error message will appear, warning you that there are triangles that are not associated with any region.

Note: The ELIMINATE statement only works on meshes defined using ATLAS syntax. You can eliminate mesh points on arbitrary meshes in DEVEDIT

19.12: EXTRACT

EXTRACT statements are used to measure parameters from both log and solution files.

Note: These commands are executed by DECKBUILD. This statement is documented in the DECKBUILD USER'S MANUAL.

Terminal Current Extraction Example

By default, EXTRACT works on the currently open log file. For example, to extract peak drain current from a run immediately after solution, type:

```
LOG OUTF=myfile.log
SOLVE .........
EXTRACT NAME="peak Id" max(i."drain")
```

Extraction Example from Previously Generated Results

To extract the same data from a previously run simulation, use the INIT parameter.

```
EXTRACT INIT INFILE="myfile.log"
EXTRACT NAME="peak Id" max(i."drain")
```

Solution Quantities Extraction Example

To use EXTRACT with solution files, use the INIT parameter. To find the integrated number of electrons in a 1D slice at X=1.0, use:

```
SAVE OUTF=mysolve.str    or    SOLVE ...... MASTER OUTF=mysolve.str
EXTRACT INIT INFILE="mysolve.str"
EXTRACT NAME="integrated e-" area from curve(depth,n.conc \
        material="Silicon" mat.occno=1 x.val=1.0)
```

Note: EXTRACT commands are generally case sensitive.

19.13: EYE.DIAGRAM

EYE.DIAGRAM specifies that an eye diagram should be generated from the specified log file. An eye diagram is created by dividing up transient data in periods of fixed size and then overlaying it.

Syntax

```
EYE.DIAGRAM INFILE OUTFILE PERIOD + [OPTIONAL PARAMETERS]
```

Parameter	Type	Default	Units
INFILE	Character		
OUTFILE	Character		
PERIOD	Real		s
T.START	Real		s
T.STOP	Real		s

Description

INFILE specifies the input log file. This should contain data from a transient simulation.

OUTFILE specifies the file output file for the eye diagram.

PERIOD specifies the window period.

T.START specifies the inital time value to be used. The default valueis the first time point in the input log file.

T.STOP specifies the final time value to be used. The default value is the last time point in the input log file.

Examples

```
EYE.DIAGRAM INFILE=laser.log OUTFILE=eye.log PERIOD=2e-10 T.START=1.5e-9
```

In this example, laser.log is a log file from the transient simulation of a buried heterostructure laser when it is biased using a pseudo-random bit sequence. The data values from time=1.5ns onwards are used to create the eye diagram with a data period of 0.2ns. The results are then saved to the log file, eye.log. Figure 19-5 shows the eye diagram in TONYPLOT.

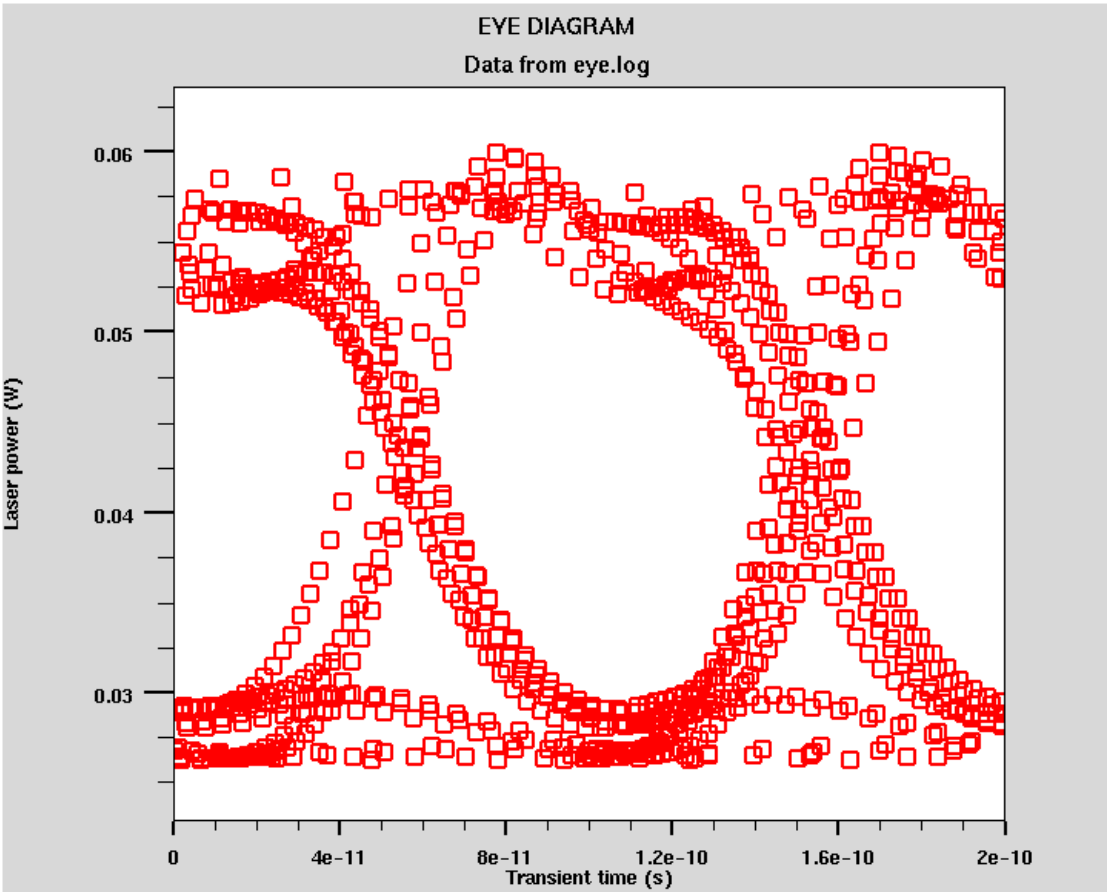


Figure 19-5: Eye diagram for BH Laser diode with a psuedo-random bit sequence input

19.14: FOURIER

FOURIER enables you to do Fourier transformations.

Syntax

```
FOURIER INFILE OUTFILE + [ OPTIONAL PARAMETERS ]
```

Parameter	Type	Default	Units
COMPLEX.VALUES	Logical	False	
FUNDAMENTAL	Real		Hz
INFILE	Character		
INTERPOLATE	Logical	False	
MAX.HARMONIC	Real		Hz
NO.DC	Logical	False	
NUM.SAMPLES	Integer	64	
OUTFILE	Character		
T.START	Real		s
T.STOP	Real		s

Description

The FOURIER statement performs a Fast Fourier Transform on log file data.

COMPLEX.VALUES specifies that the real and imaginary components are saved to file as well as the magnitude and phase values The synonym for this parameter is REAL.VALUES.

FUNDAMENTAL specifies the fundamental frequency. If the fundamental frequency is specified, T.STOP is set to T.START + 1/FUNDAMENTAL. If this is not specified, the fundamental frequency is set to (T.STOP-T.START)/NUM.SAMPLES.

INFILE specifies the input log file. This should contain data from a transient simulation.

INTERPOLATE specifies that the input data should be linearly interpolated such that data at uniform time steps are created. Interpolation of data can introduce addition (inaccurate) harmonic values into the Fourier transform. INTERPOLATE must be used if the log file contains non-uniform time steps.

MAX.HARMONIC specifies the maximum harmonic frequency that the Fourier transform should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.

NO.DC specifies that the DC component will not be saved to the log file.

NUM.SAMPLES specifies the number of discrete samples. This should be an integer power of 2 (i.e., 2^n where n is a positive integer). The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case the number of samples is set to the nearest integer power of 2 which will generate this frequency.

OUTFILE specifies the output file for the Fourier transform data.

T.START specifies the start of time data to be used for the Fourier transform. The default value is the first time point in the input log file.

T.STOP specifies the end of time data to be used for the Fourier transform. The default value is the last time point in the input log file.

Example 1

In this example, the transient data previously written to log file, *hemt1.log*, is transformed from the time domain to the frequency domain. The fundamental frequency is set to 0.5 GHz, and harmonic frequencies up to 16 GHz are calculated. Since the data in *hemt1.log* has non-uniform time steps, the INTERPOLATE flag must be enabled. The complex values as well as the magnitude and phase values are stored in *fftout1.log*.

```
FOURIER INFILE=hemt1.log FUNDAMENTAL=5e8 MAX.HARMONIC=1.6E10 \
        OUTFILE=fftout1.log INTERPOLATE COMPLEX.VALUES
```

Example 2

In this example, the log file values between 31.25 ps and 2ns are transformed into the frequency domain. The fundamental frequency is automatically determined from the time period set by T.START and T.STOP. The data values from this time period are interpolated into 64 samples, giving a maximum harmonic frequency of 15.5 GHz. The magnitude and phase values are then stored in *fftout2.log*.

```
FOURIER INFILE=hemt1.log T.START=3.125e-11 T.STOP=2e-9 NUM.SAMPLES=64 \
        OUTFILE=fftout2.log INTERPOLATE
```

19.15: GO

GO quits and restarts ATLAS and also defines certain global parameters for ATLAS execution

Note: This command is executed by DECKBUILD. This statement is documented in the DECKBUILD USER'S MANUAL.

Example starting a given ATLAS Version

To start a given version of ATLAS the syntax is set by the simflags argument. To start version 4.3.0.R, type:

```
go atlas simflags="-V 4.3.0.R"
```

Parallel ATLAS Example

To define the number of processors to be used in parallel ATLAS, use the P flag. For example, to start parallel ATLAS use four processors. For example:

```
go atlas simflags="-V 4.3.2.C -P 4"
```

19.16: IMPACT

IMPACT specifies and set parameters for impact ionization models.

Syntax

```
IMPACT <model>
```

Parameter	Type	Default	Units
A.NT	Real	0.588	
A.PT	Real	0.588	
AN1	Real	7.03×10^{5}	cm^{-1}
AN2	Real	7.03×10^{5}	cm^{-1}
AN0.VALD	Real	4.3383	
AN1.VALD	Real	-2.42e-12	
AN2.VALD	Real	4.1233	
AP0.VALD	Real	2.376	
AP1	Real	6.71×10^{5}	cm^{-1}
AP2	Real	1.682×10^{6}	cm^{-1}
ANGLE	Real	0.0	Degrees
AP1.VALD	Real	0.01033	
AP2.VALD	Real	1.0	
B.NT	Real	0.248	
B.PT	Real	0.248	
BETAN	Real	1.0	
BETAP	Real	1.0	
BN1	Real	1.231×10^{6}	V/cm
BN2	Real	1.231×10^{6}	V/cm
BN1.VALD	Real	0.0	
BN0.VALD	Real	0.235	
BP1	Real	1.231×10^{6}	V/cm
BP2	Real	2.036×10^{6}	V/cm
BP0.VALD	Real	0.17714	
BP1.VALD	Real	-0.002178	
C0	Real	2.5×10^{-10}	

Parameter	Type	Default	Units
CHIA	Real	3.0×10^{5}	
CHIB	Real	5.0×10^{4}	
CHI.HOLES	Real	4.6×10^{4}	
CN2	Real	0.0	$cm^{-1}K^{-1}$
CN0.VALD	Real	1.6831e4	
CN1.VALD	Real	4.3796	
CN2.VALD	Real	1.0	
CN3.VALD	Real	0.13005	
CP	Real	0.0	
CP2	Real	0.0	$cm^{-1}K^{-1}$
CP0.VALD	Real	0.0	
CP1.VALD	Real	0.00947	
CP2.VALD	Real	2.4929	
CP3.VALD	Real	0.0	
CROWELL	Logical	False	
CSUB.N	Real	2.0×10^{14}	
CSUB.P	Real	4.0×10^{12}	
DN0.VALD	Real	1.233735e6	
DN1.VALD	Real	1.2039e3	
DN2.VALD	Real	0.56703	
DP0.VALD	Real	1.4043e6	
DP1.VALD	Real	2.9744e3	
DP2.VALD	Real	1.4829	
DEVICE	Character		
DIRECTED	Logical	False	
E.DIR	Logical	True	
E.SIDE	Logical	False	
E.VECTOR	Logical	False	
ECN.II	Real	1.231e6	
ECP.II	Real	2.036e6	
EGRAN	Real	4.0×10^{5}	V/cm

Parameter	Type	Default	Units
ENERGY	Real	0.0025	
ETH.N	Real	1.8	eV
ETH.P	Real	3.5	eV
EXN.II	Real	1.0	
EXP.II	Real	1.0	
F.EDIIN	Character		
F.EDIIP	Character		
GRANT	Logical	False	
ICRIT	Real	4.0×10^{-3}	A/cm^2
INFINITY	Real	0.001	
LAMDAE	Real	6.2×10^{-7}	cm
LAMDAH	Real	3.8×10^{-7}	cm
LAN300	Real	6.2×10^{-7}	cm
LAP300	Real	3.8×10^{-7}	cm
LENGTH.REL	Logical	False	
LREL.EL	Real	3.35×10^{-2}	µm
LREL.HO	Real	2.68×10^{-2}	µm
M.ANT	Real	1.0	
M.APT	Real	1.0	
M.BNT	Real	1.0	
M.BPT	Real	1.0	
MATERIAL	Character		
NAME	Character		
NEW	Logical	False	
N.ION.1	Real	0.0	$cm^{-1}K^{-1}$
N.ION.2	Real	0.0	$cm^{-1}K^{-2}$
N.IONIZA	Real	7.03×10^{5}	cm^{-1}
OLD	Logical	False	
OP.PH.EN	Real	0.063	eV
OPPHE	Real	0.063	eV

Parameter	Type	Default	Units
P.ION.1	Real	0.0	$cm^{-1}K^{-1}$
P.ION.2	Real	0.0	$cm^{-1}K^{-2}$
P.IONIZA	Real	1.682×10^{6}	cm^{-1}
REGION	Integer		
SELB	Logical	False	
STRUCTURE	Character		
TAUSN	Real	0.4×10^{-12}	s
TAUSP	Real	0.4×10^{-12}	s
TOYABE	Logical	False	
VAL.AN0	Real	4.3383	
VAL.AN1	Real	-2.42e-12	
VAL.AN2	Real	4.1233	
VAL.AP0	Real	2.376	
VAL.AP1	Real	0.01033	
VAL.AP2	Real	1.0	
VAL.BN0	Real	0.235	
VAL.BN1	Real	0.0	
VAL.CN0	Real	1.6831e4	
VAL.CN1	Real	4.3796	
VAL.CN2	Real	1.0	
VAL.CN3	Real	0.13005	
VAL.DN0	Real	1.233735e6	
VAL.DN1	Real	1.2039e3	
VAL.DN2	Real	0.56703	
VAL.BP0	Real	0.17714	
VAL.BP1	Real	-0.002178	
VAL.CP0	Real	0.0	
VAL.CP1	Real	0.00947	
VAL.CP2	Real	2.4924	

Parameter	Type	Default	Units
VAL.CP3	Real	0.0	
VAL.DP0	Real	1.4043e6	
VAL.DP1	Real	2.9744e3	
VAL.DP2	Real	1.4829	
VALDINOCI	Logical	False	
ZAPPA	Logical	False	
ZAP.AN	Real	1.9	eV
ZAP.BN	Real	41.7	angstroms
ZAP.CN	Real	46.0	meV
ZAP.DN	Real	46.0	meV
ZAP.EN	Real	46.0	meV
ZAP.AP	Real	1.4	eV
ZAP.BP	Real	41.3	angstroms
ZAP.CP	Real	36.0	meV
ZAP.DP	Real	36.0	meV
ZAP.EP	Real	36.0	meV

Description

The impact ionization model for continuity equations allows the accurate prediction of avalanche breakdown for many devices. Since impact ionization is a two-carrier process, the following statement must be specified after setting impact ionization models.

```
METHOD CARRIERS=2
```

Model Selection Flags

CROWELL specifies the Crowell and Sze formulae [42].

GRANT selects Grant's impact ionization model [48] (see Chapter 3: "Physics", Equations 3-325 through 3-341).

SELB selects the impact ionization model described by Selberherr [142].

N.CONCANNON, P,CONCANNON set the Concannon substrate current model.

TOYABE is a synonym for SELB.

ZAPPA enables Zappa's model for ionization rates in InP.

Note: If no model selection flag is set, the model parameters from Grant [48] are used. See Chapter 3: "Physics", "Grant's Impact Ionization Model" Section on page 3-93.

Electric Field Model Selection Flags

The default electric field to use for ionization rate calculations is calculated as the modulus of the electric field over a triangle. In ATLAS3D, this is combined with the z-component of the field at each corner of the prism to give an overall field modulus at each node. This corresponds to Equation 3-298 in Chapter 3: "Physics" for a triangle.

ANGLE specifies the direction along which field magnitude will be calculated for use in calculating ionization rate when you specify DIRECTED.

DIRECTED specifies that the field for ionization rate calculations will be calculated along a user-specified direction. That direction is specified by the ANGLE parameter.

E.DIR specifies that the impact ionization rate will be calculated as a function of the electric field in the direction of the current (see Equation 3-259 in Chapter 3: "Physics"). Its alias is NEW.

E.SIDE specifies that the impact ionization rate will be calculated as a function of the electric field along the side of the triangle (see Equation 3-258 in Chapter 3: "Physics"). This model is only supported in ATLAS2D. Its alias is OLD.

E.VECTOR specifies that the vector electric field will be used in the calculated of the impact ionization rate (see Equation 3-257 in Chapter 3: "Physics").

OLD is the model that corresponds to Equation 3-299 in Chapter 3: "Physics" and is only supported in ATLAS2D.

NEW is the model that corresponds to Equation 3-300 in Chapter 3: "Physics", where the component of the field in the direction of the current is used in the ionisation rate calculation. It is implemented in both ATLAS2D and ATLAS3D. If the dot product of E and J is negative, then the field component is taken as 0. Consequently, impact ionisation may only occur when a current is dominated by the drift term.

Model Localization Parameters

DEVICE specifies the device in MIXEDMODE simulation to which the statement should apply. The synonym for this parameter is STRUCTURE.

MATERIAL specifies what material from Appendix B: "Material Systems", Table B-1 the statement should apply. If a material is specified then all regions defined as being composed of that material will be affected.

NAME specifies what region that the IMPACT statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.

REGION specifies that index of the region to which the impact parameters apply.

STRUCTURE is a synonym for DEVICE.

Valdinoci Model Parameters

AN0.VALD is an alias for VAL.AN0.

AN1.VALD is an alias for VAL.AN1.

AN2.VALD is an alias for VAL.AN2.

AP0.VALD is an alias for VAL.AP0.

AP1.VALD is an alias for VAL.AP1.

AP2.VALD is an alias for VAL.AP2.

BN0.VALD is an alias for VAL.BN0.

BN1.VALD is an alias for VAL.BN1.

BP0.VALD is an alias for VAL.BP0.

BP1.VALD is an alias for VAL.BP1.

CN0.VALD is an alias for VAL.CN0.

CN1.VALD is an alias for VAL.CN1.

CN2.VALD is an alias for VAL.CN2.

CN3.VALD is an alias for VAL.CN3.

CP0.VALD is an alias for VAL.CP0.

CP1.VALD is an alias for VAL.CP1.

CP2.VALD is an alias for VAL.CP2.

CP3.VALD is an alias for VAL.CP3.

DN0.VALD is an alias for VAL.DN0.

DN1.VALD is an alias for VAL.DN1.

DN2.VALD is an alias for VAL.DN2.

DP0.VALD is an alias for VAL.DP0.

DP1.VALD is an alias for VAL.DP1.

DP2.VALD is an alias for VAL.DP2.

VAL.AN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-326. The alias for this parameter is AN0.VALD.

VAL.AN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-326. The alias for this parameter is AN1.VALD.

VAL.AN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-326. The alias for this parameter is AN2.VALD.

VAL.BN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-327. The alias for this parameter is BN0.VALD.

VAL.BN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-327. The alias for this parameter is BN1.VALD.

VAL.CN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-328. The alias for this parameter is CN0.VALD.

VAL.CN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-328. The alias for this parameter is CN1.VALD.

VAL.CN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-328. The alias for this parameter is CN2.VALD.

VAL.CN3 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-328. The alias for this parameter is CN3.VALD.

VAL.DN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-329. The alias for this parameter is DN0.VALD.

VAL.DN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-329. The alias for this parameter is DN1.VALD.

VAL.DN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-329. The alias for this parameter is DN2.VALD.

VAL.AP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-330.

VAL.AP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-330. The alias for this parameter is AP1.VALD.

VAL.AP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-330. The alias for this parameter is AP2.VALD.

VAL.BP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-331. The alias for this parameter is BP0.VALD.

VAL.BP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-331. The alias for this parameter is BP1.VALD.

VAL.CP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-332. The alias for this parameter is CP0.VALD.

VAL.CP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-332. The alias for this parameter is CP1.VALD.

VAL.CP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-332. The alias for this parameter is CP2.VALD.

VAL.CP3 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-332. The alias for this parameter is CP3.VALD.

VAL.DP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is DP0.VALD.

VAL.DP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-333. The alias for this parameter is DP1.VALD.

VAL.DP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-333. The alias for this parameter is DP2.VALD.

Zappa Model Parameters

ZAP.AN specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-334.

ZAP.BN specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-335.

ZAP.CN specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-335.

ZAP.DN specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-336.

ZAP.EN specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-336.

ZAP.AP specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-337.

ZAP.BP specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-338.

ZAP.CP specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-338.

ZAP.DP specifies the value of a temperature dependent ionization parameter in Chapter 3: “Physics”, Equation 3-339.

ZAP.EP specifies the value of a temperature dependent ionization parameter in Chapter 3: "Physics", Equation 3-339.

Crowell Model Parameters

LAMBDAE specifies the mean free path for electrons. The alias for this parameter is LAN300.

LAMDAH specifies the mean free path for holes . The alias for this parameter is LAP300.

LAN300 is an alias for LAMBDAE.

LAP300 is an alias for LAMBDAH.

OP.PH.EN is an alias for OPPHE.

OPPHE specifies the optical phonon energy. The alias for this parameter is OP.PH.EN.

Selberrherr Model Parameters

AN1, **AN2**, **BN1**, **BN2**, **EGRAN** specify the basic set of parameters for Selberherr's impact ionization model. Index 1 (AP1, BP1, AN1, and BN1) corresponds to field values less than EGRAN, and index 2 (AP2, BP2, AN2, and BN2) corresponds to field values greater than EGRAN. The aliases for these parameters are **N.IONIZA**, **P.IONIZA**, **ECN.II**, and **ECP.II**.

BETAN for electrons and **BETAP** for holes correspond to coefficients for the power of ECRIT/E. The aliases for these parameters are **EXN.II** and **EXP.II**.

EXN.II is an alias for BETAN.

EXP.II is an alias for BETAP.

Temperature Dependence Parameters

A.NT specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-311.

A.**PT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-312.

B.**NT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-313.

B.**PT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-314.

M.**ANT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-311.

M.**APT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-312.

M.**BNT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-313.

M.**BPT** specifies the value of the temperature-dependent parameter in Chapter 3: "Physics", Equation 3-314.

CN2, **CP2**, **DN2** and **DP2** are specifiable coefficients in the temperature dependent models described in Chapter 3: "Physics", Equations 3-312 and 3-313. The aliases for these parameter are N.ION.1, P.ION.1, N.ION.2 and P.ION.2.

N.ION.1, **P.ION.1**, **N.ION.2** and **P.ION.2** are the aliases for CN2, CP2, DN2 and DP2.

Parameters for use with Energy Balance

F.EDIIN specifies the name of the file containing a C-INTERPRETER function describing the values of the parameters in Chapter 3: "Physics", Equation 3-309 as a function of electron temperature.

F.EDIIP specifies the name of the file containing a C-INTERPRETER function describing the values of the parameters in Chapter 3: "Physics", Equation 3-310 as a function of hole temperature.

LENGTH.REL specifies the use of energy relaxation length for the impact ionization model with the energy balance model. If LENGTH.REL is specified, TAUS and TAUSP cannot be specified and have any affect.

LREL.EL specifies an energy relaxation length for electrons if LENGTH.REL is specified.

LREL.HO specifies an energy relaxation length for holes if LENGTH.REL is specified.

TAUSN specifies the relaxation time for electrons in the temperature dependent impact model.

TAUSP specifies the relaxation time for holes in the temperature dependent impact model.

Note: When energy balance simulations are run, the Toyabe impact ionization model is used. This model is used regardless of the SELB or CROWELL settings. See Chapter 3: "Physics", "Toyabe Impact Ionization Model" Section on page 3-97 for more information about this model.

Concannon Model Parameters

CSUB.N is an empirical tuning factor used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-365) for electrons.

CSUB.P is an empirical tuning factor used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-366) for holes.

ETH.N specifies the ionization threshold energy for electrons used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-365).

ETH.P specifies the ionization threshold energy for holes used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-366).

C0 specifies the electron distribution weight factor used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-369).

CHIA specifies the electron distribution function constant used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-369).

CHIB specifies the electron distribution function constant used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-369).

CHI.HOLES specifies the hole distribution function constant used in Concannon's Substrate Current Model (see Chapter 3: "Physics", Equation 3-370).

ENERGY.STEP specifies the energy step for numeric integration used in Concannon's Substrate Current Model

INFINITY specifies the limit for the highest energy in numeric integration used in Concannon's Substrate Current Model.

Selberrherr Model Example

This example shows an IMPACT statement which specifies all parameters used by the model selected by the SELB parameter. In this case, only parameters for holes are field dependent. The AP1 and BP1 parameters correspond to parameters at field values more than EGRAN. The AP2 and BP2 parameters correspond to field values less than EGRAN. Coefficients for electrons should be repeated.

```
IMPACT SELB AN1=7.03E5 AN2=7.03E5 BN1=1.231E6 \
       BN2=1.231E6 AP1=6.71E5 AP2=1.58E6 BP1=1.693E6 \
       BP2=2.036E6 BETAN=1 BETAP=1 EGRAN=4.0E5
```

19.17: INTDEFECTS

INTDEFECTS activates the band gap interface defect model and sets the parameter values. This model can be used when thin-film transistor simulations are performed using the TFT product.

Syntax

```
INTDEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	True	
DEVICE	Character		
DFILE	Characater		
EGA	Real	0.4	eV
EGD	Real	0.4	eV
F.TFTACC	Character		
F.TFTDON	Character		
NGA	Real	5.0×10^{17}	cm^{-2}
NGD	Real	1.5×10^{18}	cm^{-2}
NTA	Real	1.12×10^{21}	cm^{-2}/eV
NTD	Real	4.0×10^{20}	cm^{-2}/eV
NUMBER	Real	All	
NUMA	Real	12	
NUMD	Real	12	
SIGGAE	Real	1.0×10^{-16}	cm^2
SIGGAH	Real	1.0×10^{-14}	cm^2
SIGGDE	Real	1.0×10^{-14}	cm^2
SIGGDH	Real	1.0×10^{-16}	cm^2
SIGTAE	Real	1.0×10^{-16}	cm^2
SIGTAH	Real	1.0×10^{-14}	cm^2
SIGTDE	Real	1.0×10^{-14}	cm^2
SOGTDH	Real	1.0×10^{-16}	cm^2
STRUCTURE	Character		
WGA	Real	0.1	eV

Parameter	Type	Default	Units
WGD	Real	0.1	eV
WTA	Real	0.025	eV
WTD	Real	0.05	ev
X.MIN	Real	left of structure	μm
X.MAX	Real	right of structure	μm
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MIN	Real	back of structure	μm
Z.MAX	Real	front of structure	μm

Description

The INTDEFECTS statement is used to describe the density of defect states in the band gap at semiconductor interfaces. You can specify up to four distributions, two for donor-like states and two for acceptor-like states. Each type of state may contain one exponential (tail) distribution and one Gaussian distribution.

AFILE specifies the file name where the acceptor state density distribution, as a function of energy, will be stored. You can examine this file by using TONYPLOT.

CONTINUOUS specifies that the continuous defect integral model will be used.

DEVICE specifies which device the statement applies in mixed mode simulation. The synonym for this parameter is STRUCTURE.

DFILE specifies the file name where the donor state density distribution, as a function of energy, will be stored. You can examine this file by using TONYPLOT.

EGA specifies the energy that corresponds to the Gaussian distribution peak for acceptor-like states. This energy is measured from the conduction band edge.

EGD specifies the energy that corresponds to the Gaussian distribution peak for donor-like states. This energy is measured from the valence band edge.

F.TFTACC specifies the name of a file containing a C-INTERPRETER function, discribing the distribution of acceptor state densities as a function of energy.

F.TFTDON specifies the name of a file containing a C-INTERPRETER function, describing the distribution of donor state densities as a function of energy.

NGA specifies the total density of acceptor-like states in a Gaussian distribution.

NGD specifies the total density of donor-like states in a Gaussian distribution.

NTA specifies the density of acceptor-like states in the tail distribution at the conduction band edge.

NTD specifies the density of donor-like states in the tail distribution at the valence band edge.

NUMBER or **REGION** specifies the region index to which the DEFECT statement applies.

NUMA specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.

NUMD specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.

SIGGAE specifies the capture cross-section for electrons in a Gaussian distribution of acceptor-like states.

SIGGAH specifies the capture cross-section for holes in a Gaussian distribution of acceptor-like states.

SIGGDE specifies the capture cross-section for electrons in a Gaussian distribution of donor-like states.

SIGGDH specifies the capture cross-section for holes in a Gaussian distribution of donor-like states.

SIGTAE specifies the capture cross-section for electrons in a tail distribution of acceptor-like states.

SIGTAH specifies the capture cross-section for holes in a tail distribution of acceptor-like states.

SIGTDE specifies the capture cross-section for electrons in a tail distribution of donor-like states.

SIGTDH specifies the capture cross-section for holes in a tail distribution of donor-like states.

STRUCTURE is a synonym for `DEVICE`.

WGA specifies the characteristic decay energy for a Gaussian distribution of acceptor-like states.

WGD specifies the characteristic decay energy for a Gaussian distribution of donor-like states.

WTA specifies the characteristic decay energy for the tail distribution of acceptor-like states.

WTD specifies the characteristic decay energy for the tail distribution of donor-like states.

X.MIN specifies the left boundary of a box, where an interface must exist, where defects are to be applied.

X.MAX specifies the right boundary of a box, where an interface must exist, where defects are to be applied.

Y.MIN specifies the top boundary of a box, where an interface must exist, where defects are to be applied.

Y.MAX specifies the bottom boundary of a box, where an interface must exist, where defects are to be applied.

Z.MIN specifies the back boundary of a box, where an interface must exist, where defects are to be applied.

Z.MAX specifies the front boundary of a box, where an interface must exist, where defects are to be applied.

19.18: INTERFACE

INTERFACE specifies interface parameters at semiconductor/insulator boundaries. All parameters apply only at the boundary nodes except where stated.

Syntax

```
INTERFACE [<params>]
```

Parameter	Type	Default	Units
AR.ABSORB	Real	0.0	cm^{-1}
AR.INDEX	Real	1	none
AR.THICK	Real	0	μm
CHARGE	Real	0.0	cm^{-2}
COATING	Integer	1	
DEVICE	Char		
DD.TUNNEL	Real	0.1	
DY.TUNNEL	Real	0.1	
F.QF	Char		
INT.RESIST	Real	0.0	Ωcm^2
LAYER	Integer	1	
MATERIAL	Char		
OPTICAL	Logical	False	
QF	Real	0.0	cm^{-2}
REGION	Integer		
S.C	Logical	False	
S.N	Real	0.0	cm/s
S.P	Real	0.0	cm/s
S.I.	Logical	True	
S.S.	Logical	False	
S.X.	Logical	False	
STRUCTURE	Character		
THERMIONIC	Logical	False	
TUNNEL	Logical	False	
X.MAX	Real	right hand side of structure	μm
X.MIN	Real	left hand side of structure	μm

Parameter	Type	Default	Units
Y.MAX	Real	bottom of structure	μm
Y.MIN	Real	top of structure	μm
Z.MIN	Real		μm
Z.MAX	Real		μm
P1.X	Real		μm
P2.X	Real		μm
P1.Y	Real		μm
P2.Y	Real		μm

Description

The INTERFACE statement consists of a set of boundary condition parameters for the interface and a set of parameter to localize the effect of these parameters.

Boundary Condition Parameters

AR.ABSORB specifies the absorption coefficient of a anti-refractive coating layer. Default value is 0.0 cm^{-1}.

AR.INDEX specifies the real component refractive index for the anti-reflective coating model in LUMINOUS. See Chapter 10: “Luminous: Optoelectronic Simulator”, “Anti-Reflective Coatings” Section on page 10-7 for more information about the anti-reflective coating model.

AR.THICK specifies the thickness of an anti-reflective coating layer for the reflection model in LUMINOUS. This layer should generally not exist in the device mesh structure. See Chapter 10: “Luminous: Optoelectronic Simulator”, “Anti-Reflective Coatings” Section on page 10-7 for more information about the anti-reflective coating model.

CHARGE specifies interface charge density (cm^{-2}) applied between two materials. The additional parameters (S.I, S.S, and S.X) allow this value to be applied at semiconductor-insulator, semiconductor-semiconductor, and semiconductor-domain edges respectively. A value of 1e10 cm^{-2} represents 1e10 electronic charges per cm^{-2} at the interface. A positive value will introduce a positive charge value and a negative value will introduce a negative charge value.

COATING specifies the number of a multilayer anti-reflective coating (ARC) referred to in the INTERFACE statement. If the COATING parameter is not set, the first coating is assumed. You must specify coatings in order (i.e., you must define COATING=2 before you define COATING=3). Different coatings should not overlap. If this occurs, the later coating is assumed in the overlapping part. For more information about ARC, see Chapter 10: “Luminous: Optoelectronic Simulator”, “Anti-Reflective Coatings” Section on page 10-7.

DD.TUNNEL controls the numerical integration in Equation 5-47. The integral in considered converged when the argument of the outer integral is less than the value of DD.TUNNEL.

Note: The numerical integral starts at the peak of the barrier where the argument should be equal to 1.0. Reducing this value increases simulation time but may improve accuracy.

DY.TUNNEL controls the step size in the numerical integration in Equation 5-47. The value of DY.TUNNEL specifies the size of the energy step, dEx. The energy step is chosen as the product of the

value for DY.TUNNEL and the energy change over the first triangle adjacent to the edge at the location in question. Reducing this value increases simulation time but may improve accuracy.

F.QF specifies the name of a file containing a C-INTERPRETER function describing the density of the interface fixed charge as a function of position.

LAYER describes the order of layers in a coating. If LAYER is not specified, the first (top) layer is assumed. You must specify the layers in order (i.e., you must define LAYER=2 before you define LAYER=3). Note that you only need to specify the coordinates of the interface for the first layer of each coating.

INT.RESIST specifies the value of a non-zero interface resistance between a conductor region and a semiconductor region.

MATERIAL specifies material name for the layer of a coating defined by the statement. Setting the material of a coating in this way enables you to apply any refractive index model supported by MATERIAL statement in ATLAS. See Chapter 10: "Luminous: Optoelectronic Simulator", "Anti-Reflective Coatings" Section on page 10-7 for more information about anti-reflective coatings.

OPTICAL specifies the optical properties of the interface defined in the statement are modeled using Transfer Matrix Method. See Chapter 10: "Luminous: Optoelectronic Simulator", Section 10.2.4: "Matrix Method" for more information.

QF specifies fixed oxide charge density (cm^{-2}) applied at a semiconductor to insulator interface. A value of 1e10 cm^{-2} represents 1e10 electronic charges per cm^{-2} at the interface. A positive value will introduce a positive charge value and a negative value will introduce a negative charge value.

S.C specifies that the application of interface models specified in this INTERFACE statement should include semiconductor-conductor interfaces.

S.I specifies that the application of interface models specified in this INTERFACE statement should include semiconductor-insulator interfaces.

S.N specifies the electron surface recombination velocity.

S.P specifies the hole surface recombination velocity.

S.S specifies that the application of interface models specified in this INTERFACE statement should include semiconductor-semiconductor interfaces.

S.X specifies that the application of interface models specified in this INTERFACE statement should include any interfaces including interfaces with the outside domain.

THERMIONIC specifies that carrier transport across an interface is modeled by thermionic emission. This model will only be applied at semiconductor/semiconductor boundaries. See Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.1.4: "The Thermionic Emission and Field Emission Transport Model" for more information on thermionic emission.

TUNNEL specifies that the carrier transport across the interface will account for thermionic field emission. When TUNNEL is specified, THERMIONIC should also be specified. See Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.1.4: "The Thermionic Emission and Field Emission Transport Model" for more information on thermionic emission.

Note: To use the INTERFACE statement to specify thermionic or tunneling interfaces, place the INTERFACE statement immediately after all MESH, REGION, ELECTRODE and DOPING statements and before any CONTACT, MODEL, MATERIAL, MOBILITY or IMPACT statements.

Position Parameters

X.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** define a bounding box. Any semiconductor/insulator interfaces found within this region are charged. If there is only one interface in a device, a non-planar surface may be defined using a box which contains the whole device.

X.**MIN** specifies the left x coordinate of the bounding box.

X.**MAX** specifies the right x coordinate of the bounding box.

Y.**MIN** specifies the bottom y coordinate of the bounding box.

Y.**MAX** specifies the top y coordinate of the bounding box.

Z.**MIN** specifies the front z coordinate of the bounding box. It is used in 3-D modules only.

Z.**MAX** specifies the back z coordinate of the bounding box. It is used in 3-D modules only.

P1.X, **P1**.**Y**, **P2.X**, and **P2.Y** define a bounding box. Within this box must lie the interface that is to be represented as the anti-reflective coating. See Chapter 10: "Luminous: Optoelectronic Simulator", the "Anti-Reflective Coatings" Section on page 10-7 for more information about the anti-reflective coating model.

Note: For anti-reflective coatings in Luminous3D, use `X.MIN`, `X.MAX`, `Y.MIN`, `Y.MAX`, `Z.MIN`, and `Z.MAX` to define the bounding box.

P1.X specifies the left x coordinate of the bounding box.

P2.X specifies the right x coordinate of the bounding box.

P1.Y specifies the bottom y coordinate of the bounding box.

P2.Y specifies the top y coordinate of the bounding box.

DEVICE specifies which device in a MIXEDMODE simulation applies to the statement. The synonym for this parameter is `STRUCTURE`.

REGION specifies which region number applies to the statement.

STRUCTURE is a synonym for `DEVICE`.

MOS Example

This example defines an interface with both fixed charge and recombination velocities.

```
INTERFACE X.MIN=-4  X.MAX=4 Y.MIN=-0.5 Y.MAX=4 \
          QF=1E10 S.N=1E4 S.P=1E4
```

SOI Example

To define different fixed charge on the front and back interfaces of an SOI transistor, you need two `INTERFACE` statements.

In the syntax below, the first statement will apply 5.10^{10} cm^{-2} charge to any silicon/oxide interface above Y=0.01μm. The second statement applied a higher charge to any interface below Y=0.01μm. Note that charges are only applied at the material interfaces so the y coordinate needs only to be somewhere within the silicon film.

```
INTERFACE Y.MAX-0.01 QF=5e10

INTERFACE Y.MIN=0.01 QF=2e11
```

Interface Charge for III-V Devices

By default, the INTERFACE statement is applied to semiconductor-insulator interfaces. Interface charge can, however, be added at the interfaces between two semiconductor regions or at the edges of semiconductor regions.

The CHARGE parameter defines the interface charge value in cm^{-2}. The S.I, S.S, and S.X parameters control whether the charge is placed between semiconductor-insulator regions, semiconductor-semiconductor regions, or at the semiconductor domain edges. You can control the location of the added charge by using the position parameters.

19.19: INTTRAP

INTTRAP activates interface defect traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
INTTRAP <type> E.LEVEL=<r> DENSITY=<r> <capture parameters>
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
DEGEN.FAC	Real	1	
DENSITY	Real		cm^{-2}
DEVICE	Char		
DONOR	Logical	False	
E.LEVEL	Real		eV
F.DENSITY	Char		
REGION	Integer		
SIGN	Real		cm2
SIGP	Real		cm2
STRUCTURE	Character		
TAUN	Real		s
TAUP	Real		s
X.MIN	Real	left of structure	μm
X.MAX	Real	right of structure	μm
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MIN	Real	back of structure	μm
Z.MAX	Real	front of structure	μm

Description

DEVICE specifies which device the statement applies to in MIXEDMODE simulation. The synonym for this parameter is STRUCTURE.

DONOR specifies a donor-type trap level.

ACCEPTOR specifies an acceptor-type trap level.

DEGEN.FAC specifies the degeneracy factor of the trap level used to calculate the density.

DENSITY sets the maximum density of states of the trap level.

E.LEVEL sets the energy of the discrete trap level. It is equal to the energy distance between conductance band and trap level for acceptor trap, and to energy distance between trap level and valence band for donor trap.

F.DENSITY specifies the name of a file containing a C-INTERPRETER function describing the density of donor/acceptor interface traps as a function of position.

STRUCTURE is a synonym for DEVICE.

X.MIN specifies the left boundary of a box, where an interface must exist and where traps are to be applied.

X.MAX specifies the right boundary of a box, where an interface must exist and where traps are to be applied.

Y.MIN specifies the top boundary of a box, where an interface must exist and where traps are to be applied.

Y.MAX specifies the bottom boundary of a box, where an interface must exist and where traps are to be applied.

Z.MIN specifies the back boundary of a box, where an interface must exist and where traps are to be applied.

Z.MAX specifies the front boundary of a box, where an interface must exist and where traps are to be applied.

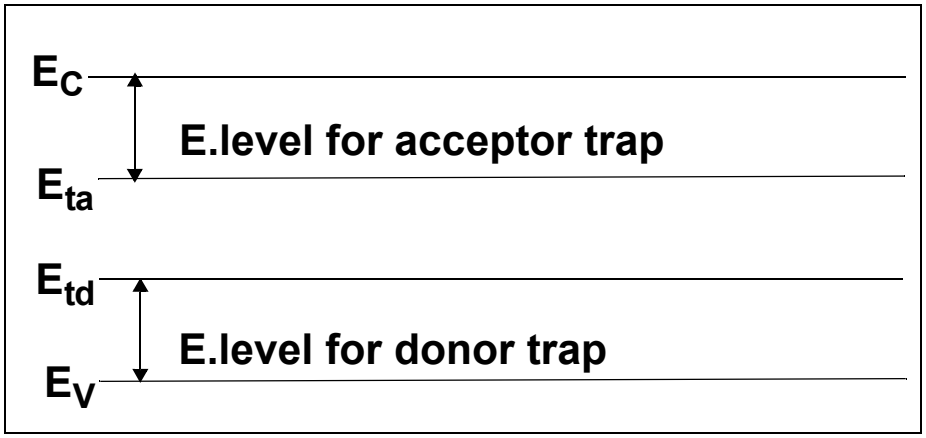


Figure 19-6: Acceptor and Donor Interface Trap Energy Levels

Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN specifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level.

Multiple Interface Trap States Example

The following example sets three discrete interface trap levels within the silicon bandgap. These trap levels will capture carriers, which slows down the switching speed of any device. In this example, the capture cross sections are used to define the properties of each trap.

```
inttrap e.level=0.49 acceptor density=2.e10 degen=12 \
sign=2.84e-15 sigp=2.84e-14
inttrap e.level=0.41 acceptor density=1.e10 degen=12 \
sign=7.24e-16 sigp=7.24e-15
inttrap e.level=0.32 donor density=1.e10 degen=1 \
sign=1.00e-16 sigp=1.00e-17
```

Note: For semiconductor bulk trap levels, see Section 19.52: “TRAP”.

19.20: LASER

LASER defines physical models and model parameters for laser and Vertical Cavity Surface-Emitting Lasers (VCSEL) simulation. For more information about VCSEL, see Chapter 9: "VCSEL Simulator".

Syntax

```
LASER<parameters>
```

Parameter	Type	Default	Units
ABSORPTION	Logical	False	
ATRAP	Real	0.5	
CAVITY.LENGTH	Real	100.0	cm
COUPLED	Logical	False	
DBR1.START	Real	0.0	microns
DBR1.FINAL	Real	0.0	microns
DBR2.START	Real	0.0	microns
DBR3.FINAL	Real	0.0	microns
DEVICE	Character		
EFINAL	Real	0.0	eV
EINIT	Real	0.0	eV
ESEP	Real	eV	
ETRANS	Logical	False	
F.MIRROR	Character		
FAR.NX	Integer	100	
FAR.NY	Integer	100	
FCARRIER	Logical	False	
INDEX.BOTTOM	Real	1.0	
INDEX.MODEL	Integer	0	
INDEX.TOP	Real	1.0	
ITMAX	Integer	30	
LMODES	Logical	False	
LOSSES	Real	0	
MATERIAL	Character		
MAXCH	Real	2.5	
MAXTRAPS	Integer	2	

MIRROR	Real	90.0	%
MULTISAVE	Logical	True	
NAME	Character		
NEAR.NX	Integer	100	
NEAR.NY	Integer	100	
NEFF	Real	3.57	
NMODE	Integer	1	
NX	Integer		
NSPEC	Integer	100	
NY	Integer		
OMEGA	Real	2.16×10^{15}	Hz
PHOTON.ENERGY	Real	0	eV
PROJ	Logical	False	
REFLECT	Logical	False	
RF	Real	0.0	%
RF.FILE	Character		
REGION	Integer		
RR	Real	0.0	%
RR.FILE	Character		
SIN	Real	100000	cm^2
SPEC.NAME	Character		
SPECSAVE	Integer	1	
SPONTANEOUS	Integer	1	
STRUCTURE	Character		
TAUSS	Real	0.05	
TIMERATE	Logical	True	
TOLER	Real	0.01	
TRANS_ENERGY	Real	eV	
TRAP	Logical	False	
VCSEL.CHECK	Integer	1	
VCSEL.INCIDENCE	Integer	False	
XMIN	Real	min. X	microns

XMAX	Real	max. X	microns
YMIN	Real	min. Y	microns
YMAX	Real	max. Y	microns

Description

Localization Parameters

DEVICE specifies which device the LASER statement should apply to in MIXEDMODE simulation. The synonym for this parameter is STRUCTURE.

MATERIAL specifies which material from the table in Appendix B: "Material Systems" that the LASER statement should apply. If a material is specified, then all regions defined as being composed of that material will be affected.

NAME specifies which region the LASER statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.

REGION specifies the region number to which these parameters apply. If there is more than one semiconductor region, specification of different parameters for each region is allowed. If LASER is not specified, all regions in the structure are changed.

STRUCTURE is a synonym for DEVICE.

Laser Model Parameters

ABSORPTION enables the absorption loss model in LASER.

ATRAP specifies the photon rate equation cutback ratio (see Chapter 8: "Laser: Edge Emitting Simulator", Section 8.4.2: "Numerical Parameters").

DBR1.START, **DBR1.FINAL**, **DBR2.START**, and **DBR2.FINAL** parameters specify the locations of the front and rear DBR mirrors for a VCSEL device (See Figure 19-7). The reflectivity of front and rear DBR mirrors will be calculated from the optical intensity profile.

CAVITY.LENGTH specifies the cavity length in the longitudinal direction (in μm).

COUPLED specifies that the solution process for the photon rate equation is fully coupled to the Jacobian of the drift-diffusion equations.

EINIT, **EFINAL** specify the lower and upper photon energies. LASER will calculate multiple longitudinal photon rates within this range. Using wide ranges can slow down simulation.

ESEP specifies the photon energy separation. If this isn't specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range.

ETRANS enables the selection of a specific transverse mode. The selected transverse mode will be the mode with the closest energy to TRANS_ENERGY.

F.MIRROR specifies the name of a file containing a C-Interpreter function that defines the front and rear mirror reflectivities as a function of wavelength.

FAR.NX, **FAR.NY** describe the number of samples to output for the far-field pattern in the X and Y directions. For more information about the far-field pattern, see Chapter 8: "Laser: Edge Emitting Simulator", Section 8.5.1: "Generation of Near-Field and Far-Field Patterns".

FCARRIER enables the free carrier loss model in LASER.

INDEX.BOTTOM specifies the refractive index of the medium below the structure. The default value is 1.0.

INDEX.MODEL specifies whether the simple refractive index model (INDEX.MODEL=0) or the more complex gain dependent refractive index (INDEX.MODEL=1) is used.

INDEX.TOP specifies the refractive index of the medium above the structure. The default value is 1.0.

Note: When using INDEX.MODEL=1, a complex value Eigenvalue solver is used. This requires a refined x direction LASER mesh and a refined y direction LASER mesh to ensure the accuracy of the solution. When the bulk refractive index model is used, only a refined y direction LASER mesh is required for the Eigenvalue solver.

ITMAX specifies the maximum number of iterations allowed for LASER simulation at each bias point.

LMODES specifies that multiple longitudinal models are to be accounted for during laser simulation.

LOSSES specifies the total losses in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-5.

MAXCH specifies the maximum allowed relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.

MAXTRAPS specifies the maximum number of cutbacks for the photon rate equation (see Chapter 8: “Laser: Edge Emitting Simulator”, Section 8.4.2: “Numerical Parameters”).

MIRROR specifies the percentage facet reflectivity for the mirror loss in LASER. 100% reflectivity is equivalent to no mirror loss. Both facets are assumed to have this value of reflectivity.

MULTISAVE specifies the whether to save the transient laser spectrum as one file or multiple files.

NEAR.NX, **NEAR.NY** describe the number of samples to output for the near-field pattern in the X and Y directions. For more information about the near-field pattern, see Chapter 8: “Laser: Edge Emitting Simulator”, Section 8.5.1: “Generation of Near-Field and Far-Field Patterns”.

NEFF specifies the effective refractive index in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-4.

NMODE is the same as the LAS.NMODE parameter in the MODELS statement.

NSPEC specifies the number of sampling points between EINIT and EFINAL in the reflectivity test. The default value is NSPEC=100.

NX defines the number of mesh divisions for the laser mesh in the X direction.

NY defines the number of mesh divisions for the laser mesh in the Y direction.

PHOTON.**ENERGY** specifies the energy of photons to be used in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-1. If Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-2 is used for simulation, this parameter will only specify an initial estimate of the photon energy. If that’s the case, use the LAS.OMEGA parameter instead to specify the lasing frequency.

PROJ is the same as the LAS.PROJ parameter in the METHOD statement.

OMEGA specifies the lasing frequency to be used in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-1. If model 2 is used for simulation, then this parameter will estimate the lasing frequency. If that’s the case, use the PHOTON.ENERGY parameter to specify photon energy instead.

REFLECT specifies that only half of the Laser structure is to be simulated. The axis of symmetry in this case is at x=0. Specify the laser mesh so that the minimum x coordinate is zero.

RF and **RR** specify the front and rear facet mirror reflectivities of Fabry-Perot type lasers. If these parameters aren’t specified, then they will be calculated from the average mirror loss parameter (MIRROR.LOSS parameter in the LASER or MODELS statements). If MIRROR.LOSS is used, then RF and RR shouldn’t be specified.

RF.FILE and **RR.FILE** specify the front and rear facet mirror reflectivity files. If RF.FILE is specified, the rear reflectivity will be read from the specified file. A linear interpolation algorithm will calculate value of the reflectivity from the tabulated values with the bounds given by the first and last wavelength value. The wavelenght values in the file must be ascending order. The file format is as follows:

```
number of samples
wavelength (in microns)                          reflectivity (in %)
```

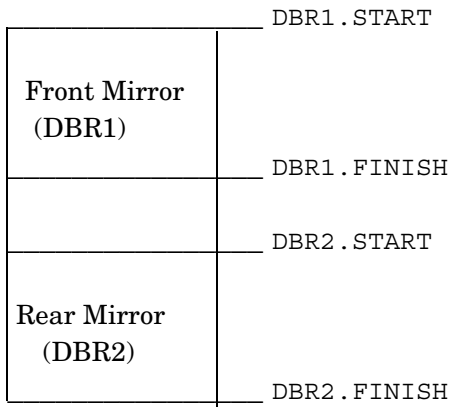


Figure 19-7: DBR Location Parameters

The cavity length used in the mirror loss calculation is given by DBR2.START - DBR1.FINISH. If both DBR mirrors are not defined, then the mirror loss will set to 0.

SIN specifies an initial photon density in the fundamental lasing mode. This value provides an initial guess for subsequent iterations. This parameter is used only when the single frequency model has been selected.

SPEC.NAME specifies the name of a spectrum file, which LASER will produce for each bias point, if the LMODES parameter has been specified.

SPECSAVE the spectrum file will be saved after every LAS.SPECSAVE laser solution step.

SPONTANEOUS enables the Spontaneous Recombination Model (see Chapter 3: "Physics", Sections 3.9.1: "The General Radiative Recombination Model" and 3.9.2: "The Default Radiative Recombination Model").

TAUSS specifies the relaxation parameter to be used for the photon rate equation. See Chapter 8: "Laser: Edge Emitting Simulator", Section 8.4.2: "Numerical Parameters" for more information.

TIMERATE specifies that the time dependent photon rate equation will be used in a transient laser simulation.

TOLER specifies the desired accuracy in photon areas.

TRANS_ENERGY specifies the energy for selective a single transverse mode (see also ETRANS).

TRAP specifies the photon rate equation cutback scheme (see Chapter 8: "Laser: Edge Emitting Simulator", Section 8.4.2: "Numerical Parameters").

VCSEL.CHECK enables reflectivity test simulation of the VCSEL structure.

VCSEL.INCIDENCE specifies the direction of light incident on the structure. VCSEL.INCIDENCE=1 is the light incident from the top. VCSEL.INCIDENCE=0 is the light incident from the bottom. VCSEL.INCIDENCE=2 or >2 means both directions of light incidence are considered. By default, light is incident from the top of the structure.

XMIN defines the minimum X coordinate for the laser mesh. See also LX.MESH.

XMAX defines the maximum X coordinate for the laser mesh. See also LX.MESH.

YMIN defines the minimum Y coordinate for the laser mesh. See also LY.MESH.

YMAX defines the maximum Y coordinate for the laser mesh. See also LY.MESH.

19.21: LOAD

LOAD loads previous solutions from files as initial guesses to other bias points.

Syntax

```
LOAD [ASCII|MASTER] [NO.CHECK] <files>
```

Parameter	Type	Default	Units
ASCII	Logical	False	
IN.FILE	Character		
INFILE	Character		
IN1FILE	Character		
IN2FILE	Character		
LAS.SPECTRUM	Logical	False	
MASTER	Logical	False	
NO.CHECK	Logical	False	
TWOD	Logical	False	

Description

ASCII specifies that any original PISCES format files read or written by this statement will be in an ASCII rather than in a binary format.

LAS.SPECTRUM loads the spectrum file (specified by the IN.FILE parameter) into LASER.

Note: The number of longitudinal and transverse modes in the spectrum file must be the same as created by the LASER statement.

MASTER specifies that any files read by this statement will be in a standard structure file rather than the original PISCES format. If you are using TONYPLOT to plot simulation results, this parameter should be specified.

NO.CHECK prevents checking material parameter differences between loaded binary files and the values set in the current input file.

TWOD allows loading of a 2-D solution into a 3-D structure. Note that the values from the 2-D solution are loaded uniformly in the Z direction.

File Parameters

The LOAD statement requires that one of the following file parameter syntax be used.

```
LOAD INFILE=<filename>
```

or

```
LOAD IN1FILE=<filename> IN2FILE=<filename>
```

IN.FILE is a synonym for INFILE.

INFILE specifies a single input filename for solution data. This parameter should be used when you wish to load only one solution which is the most common case. The synonym for this parameter is IN.FILE.

IN1FILE specifies a filename for present solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the SOLVE statement).

IN2FILE specifies an input filename for previous solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the SOLVE statement). The solution specified by this parameter is the first to be overwritten when new solutions are obtained.

Simple Save and Load Examples

This example saves and loads the master format solution file, SOL.STR.

```
              SAVE OUTF=SOL.STR.

....

  LOAD INFILE=SOL.STR MASTER
```

As before but using the SOLVE syntax.

```
SOLVE OUTF=SOL.STR MASTER

  ..

  LOAD INF=SOL.STR MASTER
```

When the save and load operations are not done within the same ATLAS run see the note below.

Binary Format Example

Saving and loading using the binary format. This is quicker but these files cannot be plotted in TONYPLOT.

```
SOLVE OUTF=SOLVE_TMP

 ..

LOAD INF=SOLVE_TMP
```

Note: The function to calculate the difference between two files is now inside TONYPLOT. It has been discontinued from the LOAD statement

---

Note: The LOAD statement loads only the saved solution quantities into ATLAS. The mesh, electrodes, doping, regions, contact settings, material parameters, models, and numerical methods must all be specified in advance of any LOAD statement. See the Chapter 2: “Getting Started with ATLAS”, “Re-initializing ATLAS at a Given Bias Point” Section on page 2-55.

19.22: LOG

LOG allows all terminal characteristics of a run to be saved to a file. Any DC, transient, or AC data generated by SOLVE statements after the LOG statement is saved. Any parameters specified by the PROBE statement are also stored in the logfile. If a log file is already open, the open log file is closed and a new log file is opened.

Syntax

```
LOG [OUTFILE=<filename>] [MASTER] [acparams]
```

Parameter	Type	Default	Units
ABCD.PARAM	Logical	False	
APPEND	Logical	False	
CLOSE	Logical	False	
GAINS	Logical	False	
H.PARAM	Logical	False	
IMPEDANCE	Real	50	ohms
INPORT	Character		
IN2PORT	Character		
J.DISP	Logical	False	
J.ELECTRON	Logical	False	
J.HOLE	Logical	False	
LCOMMON	Real	0	H
LGROUND	Real	0	H
LIN	Real	0	H
LOUT	Real	0	H
MASTER	Logical	True	
NOISE	Logical	False	
NOISE.ALL	Logical	False	
NOISE.I	Logical	False	
NOISE.IV	Logical	False	
NOISE.I.ALL	Logical	False	
NOISE.V	Logical	False	
NOISE.VI	Logical	False	
NOISE.V.ALL	Logical	False	
OFF	Logical	False	

Parameter	Type	Default	Units
OLD	Logical	False	
OUTPORT	Character		
OUT2PORT	Character		
OUT.FILE	Character		
OUTFILE	Character		
RCOMMON	Real	0	ohms
RGROUND	Real	0	ohms
RIN	Real	0	ohms
ROUT	Real	0	ohms
S.PARAM	Logical	False	
SIM.TIME	Logical	False	s
WIDTH	Real	1	microns
Y.PARAM	Logical	False	
Z.PARAM	Logical	False	

File Output Parameters

APPEND specifies that the output I-V information should be appended to an existing log file. Make sure that the existing log files contain the same type of data (e.g., DC, AC, transient) as the subsequent SOLVE statements.

CLOSE is an alias for OFF.

J.DISP specifies that displacement currents are written to the log file.

J.ELECTRON specifies that electron currents are to be written into the log file.

J.HOLE specifies that hole currents are to be written into the log file.

MASTER specifies that AC data and I-V information will be saved in a standard structure file format. This is the default format.

OFF specifies that any currently open log file will be closed and log file output is discontinued. The alias for this parameter is CLOSE.

OLD specifies that AC data and IV information will be saved in the original PISCES-II file format. A synonym for this parameter is PISCES.

OUT.FILE is a synonym for OUTFILE.

OUTFILE specifies the log file that will be used to store DC, AC, or transient I-V information. The synonym for this parameter is OUT.FILE.

Note: The older ACFILE syntax is not supported and should not be used. AC results are stored in the file specified by OUTFILE as long as the first SOLVE statement after the LOG statement contains AC analysis.

RF Analysis Parameters

If S.PARAM, H.PARAM, Z.PARAM, GAINS, or ABCD.PARAM is specified, the capacitance and conductance data will be converted into the requested set of AC parameters.

S.**PARAM** elects s parameter analysis. For S-parameter analysis, you can also choose to set any of the parasitic element parameters.

H.**PARAM** selects h parameter analysis.

Y.**PARAM** selects Y parameter analysis

Z.**PARAM** selects z parameter analysis.

ABCD.**PARAM** selects ABCD parameter analysis.

GAINS selects the calculation of several types of gains used in RF analysis [21]. These are the stability factor, unilateral power gain (GUmax), maximum unilateral transducer power gain (GTmax), maximum available power gain (Gma), and the maximum stable power gain (Gms). The magnitude of H21 is also calculated.

IMPEDANCE specifies the matching impedance for s-parameter calculation.

INPORT specifies the electrode name for the primary input port used when performing any AC parameter calculations.

IN2PORT specifies the electrode name of the secondary input port.

OUTPORT specifies the electrode name for the output ports used when performing any AC parameter calculations.

OUT2PORT specifies the electrode n of the secondary output port

NOISE Parameters

To perform noise analysis on a one-port device, define the INPORT. To perform noise analysis on a two-port device, define the INPORT and the OUTPORT.

NOISE selects F_{min}, Z_O, and g_n (the two-port noise figures of merit, 2pNFOM) for output.

NOISE.ALL selects all noise results for output. These are 2pNFOM, total and individual noise voltage sources, and total and individual noise current sources.

NOISE.I selects the 2pNFOM and the correlation of the total noise current sources.

NOISE.I.ALL selects the 2pNFOM, the correlation of the total noise current sources, and the correlation of the noise current sources from the individual noise mechanisms (GR, II, electron and hole diffusion; electron and hole flicker).

NOISE.V selects the 2pNFOM and the correlation of the total noise voltage sources.

NOISE.VI or **NOISE.IV** selects the 2pNFOM and the correlations of both the total noise voltage sources and the total noise current sources.

NOISE.V.ALL selects the 2pNFOM, the correlation of the total noise voltage sources, and the correlation of the noise voltage sources from the individual noise mechanisms (GR, II, electron and hole diffusion; electron and hole flicker).

Note: The 2pNFOM have no meaning for a one-port device and therefore don't output. If NOISE is the only parameter defined for a one-port device, then the correlation of the total noise voltage sources outputs (rather than outputting nothing).

Parasitic Element Parameters

For RF parameter extraction, you can also set any of the parasitic element parameters. By setting the parasitic element parameters, you can apply lumped parasitic resistances or inductances to the terminal of the two-port device during the RF parameter extraction. These parameters will not affect the capacitance or conductance matrices calculated by ATLAS.

RIN specifies the lumped parasitic resistance on the input to the two port device for s-parameter extraction. The value of RIN is in Ohms.

ROUT specifies the lumped parasitic resistance on the output to the two port device for s-parameter extraction. The value of ROUT is in Ohms.

RGROUND specifies the lumped parasitic resistance on the ground or common side of the two port device for s-parameter extraction. The value of RGROUND is in Ohms. RCOMMON is an alias for RGROUND.

LIN specifies the lumped parasitic inductance on the input to the two port device for s-parameter extraction. The value of LIN is in Henrys.

LOUT specifies the lumped parasitic inductance on the output to the two port device for s-parameter extraction. The value of LOUT is in Henrys.

LGROUND specifies the lumped parasitic inductance on the ground or common side of the two port device for s-parameter extraction. The value of LGROUND is in henries. LCOMMON is an alias for LGROUND.

SIM.TIME saves the time taken for a bias point into the log file and measured in seconds. Note that if multiple jobs are using the same CPU that this method may not be a true reflection of processor speed.

WIDTH specified an output width (in z direction) to apply during the s-parameter calculation. Note that this parameter affects *only* the derived RF parameters and not currents, capacitances or conductances. The WIDTH parameter of the MESH statement can be used to scale these. Using both these WIDTH parameters will lead to a multiplication of the two widths for the RF parameters.

Simple Logfile Definition Example

This example saves all I-V data in file, *myfile.log*.

```
LOG OUTF=myfile.log
```

Results should be plotted using TONYPLOT.

RF Analysis Example

To generate s-parameters, assume the input is gate/source and the output is drain/source. A width of 100 microns is also defined along with 100ohm resistance on the input. For example:

```
LOG OUTF=mysparams.log S.PARAM INPORT=gate OUTPORT=drain \
    IN2PORT=source OUT2PORT=source WIDTH=100 RIN=100
```

Transient or AC Logfile Example

The contents of LOG files varies for different types of simulations (e.g., DC, transient, AC). The content is set by the first SOLVE statement after the LOG statement. Therefore, the following syntax is required.

```
SOLVE VDRAIN=0.5

   LOG OUTF=myfile.log

   SOLVE VDRAIN=3.0 RAMPTIME=1e-9 DT=1e-11 TSTOP=1e-7
```

Correct transient parameters would have not been stored, if the LOG statement had been placed before the first SOLVE statement, which is DC.

19.23: LX.MESH, LY.MESH

L<n>.MESH specifies the location of grid lines along the <n>-axis in a rectangular mesh used in LASER simulation. The syntax is equivalent for x and y directions.

Syntax

```
LX.MESH NODE=<n> LOCATION=<n>

LX.MESH SPACING=<n> LOCATION=<n>

LY.MESH NODE=<n> LOCATION=<n>

LY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE specifies the mesh line index. These mesh lines are assigned consecutively.

LOCATION specifies the location of the grid line.

SPACING specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

LASER Mesh Example

This syntax defines a mesh of 33x33 covering the area bounded by (0.3,0.0) to (2.4,1.0).

```
LX.M n=1 l=0.3

LX.M n=33 l=2.4

LY.M n=1 l=0.0

LY.M n=33 l=1.0
```

Note: The mesh defined in these statements for the Laser Helmholtz Solver is entirely separate from the electrical device simulation mesh defined on the MESH statement.

Setting Locally Fine Grids Example

This example shows how to space grid lines closely around a topological feature such as a junction at y=0.85 microns.

```
LY.MESH  LOC=0.0  SPAC=0.2

LY.MESH  LOC=0.85 SPAC=0.01

LY.MESH  LOC=2    SPAC=0.35
```

19.24: MATERIAL

MATERIAL associates physical parameters with materials in the mesh. The parameter default values for standard semiconductors are shown in Appendix B: “Material Systems”.

Syntax

```
MATERIAL <localization>  <material_definition>
```

Parameter	Type	Default	Units
A.DEFPOT	Real	2.1	eV
A.PASSLER	Real	3.18E-4	eV/K
A1	Real	0.0	
A2	Real	0.0	
A3	Real	0.0	
A4	Real	0.0	
A5	Real	0.0	
A6	Real	0.0	
AADACHI	Real	see Appendix B	
ABSORPTION.SAT	Real	1.0E7	
AC	Real	0.0	eV
AFFINITY	Real	see Appendix B	eV
ALATTICE	Real	0.0	A
ALIGN	Real	use AFFINITY	
ALPHAA	Real	0.0	
ALPHAR	Real	4.0	
AN	Real	1.0	
AN2	Real	7.03×10^{5}	cm^{-1}
AN0.VALD	Real	4.3383	
AN1.VALD	Real	-2.42e-12	
AN2.VALD	Real	4.1233	
AP0.VALD	Real	2.376	
AP1.VALD	Real	0.01033	
AP2.VALD	Real	1.0	
AP	Real	1.0	
AP2	Real	1.682×10^{6}	cm^{-1}

Parameter	Type	Default	Units
ARICHN	Real	see Appendix B	$A/cm^2/K^2$
ARICHP	Real	see Appendix B	$A/cm^2/K^2$
ASTR	Real	0.0	
ASYMMETRY	Real	0.5	
A.TC.A	Real	0.0	cm K /W
A.TC.B	Real	0.0	cm /W
A.TC.C	Real	0.0	cm / W /K
A.TC.CONST	Real	0.0	(W/cm/K)
A.TC.D	Real	0.0	K
A.TC.E	Real	0.0	W/cm
A.TC.NPOW	Real	0.0	-
AUGN	Real	see Appendix B	cm^6/s
AUGP	Real	see Appendix B	cm^6/s
AUGKN	Num	0.0	cm^3
AUGKP	Num	0.0	cm^3
AV	Real	0.0	eV
B.DEFPOT	Real	-2.33	eV
BADACHI	Real	see Appendix B	
BBB	Real	0.0	eV
BB.A	Real	4.0E14	$eV^{-2}*s^{-1}*cm^{-1}$
BB.B	Real	1.97E7	V/cm
BB.GAMMA	Real	2.5	
BETAN	Real	1.0	
BETAP	Real	1.0	
BGN.C	Real	0.5	
BGN.N	Real	1.0×10^{17}	cm^{-3}
BGN.E	Real	9.0×10^{-3}	V
BN	Real	1.0	
BN2	Real	1.231×10^{6}	V/cm
BN1.VALD	Real	0.0	

Parameter	Type	Default	Units
BN0.VALD	Real	0.235	
BP	Real	1.0	
BP2	Real	2.036×10^{6}	V/cm
BP0.VALD	Real	0.17714	
BP1.VALD	Real	-0.002178	
BQP.NGAMMA	Real	1.2	
BQP.NALPHA	Real	0.5	
BQP.PGAMMA	Real	1.0	
BQP.PALPHA	Real	0.5	
BSTR	Real	0.0	
C.DEFPOT	Real	-4.75	eV
C.DIRECT	Real	0.0	cm^3/s
C11	Real	0.0	GPa
C12	Real	0.0	GPa
C13	Real	0.0	GPa
C33	Real	0.0	GPa
CON.BGN	Real	0.5	
CN	Real	0.0	
CN2	Real	0.0	$cm^{-1}K^{-1}$
CN0.VALD	Real	1.6831e4	
CN1.VALD	Real	4.3796	
CN2.VALD	Real	1.0	
CN3.VALD	Real	0.13005	
COPT	Real	0.0	cm^3/s
CP	Real	0.0	
CP2	Real	0.0	$cm^{-1}K^{-1}$
CP0.VALD	Real	0.0	
CP1.VALD	Real	0.00947	
CP2.VALD	Real	2.4929	
CP3.VALD	Real	0.0	
CSTR	Real	0.0	

Parameter	Type	Default	Units
D.DEFPOT	Real	1.1	eV
D1	Real	0.0	eV
D2	Real	0.0	eV
D3	Real	0.0	eV
D4	Real	0.0	eV
DADACHI	Real	see Appendix B	eV
DEGENERACY	Integer	2	
DELTA1	Real	0.0	eV
DELTA2	Real	0.0	eV
DELTA3	Real	0.0	eV
DEVICE	Character		
DGN.GAMMA	Real	3.6	
DGP.GAMMA	Real	3.6	
DINDEXDT	Real	0.0	1/K
D.TUNNEL	Real	10^{-6}	cm
DIST.SBT	Real	10^{-6}	cm
DN2	Real	0.0	$cm^{-1}K^{-1}$
DN0.VALD	Real	1.233735e6	
DN1.VALD	Real	1.2039e3	
DN2.VALD	Real	0.56703	
DP2	Real	0.0	$cm^{-1}K^{-1}$
DP0.VALD	Real	1.4043e6	
DP1.VALD	Real	2.9744e3	
DP2.VALD	Real	1.4829	
DRHODT	Real	0.0	$\mu\Omega$·cm/k
DSTR	Real	0.0	
E.FULL.ANISO	Logical	False	E.FULL.ANISO
E31	Real	0.0	C/m2
E33	Real	0.0	C/m2
EAB	Real	0.045	eV
ECN.II	Real	1.231×10^{6}	

Parameter	Type	Default	Units
ECP.II	Real	2.036×10^6	
EDB	Real	0.044	eV
EG300	Real	see Appendix B	eV
EG1300	Real	0.0	eV
EG2300	Real	0.0	eV
EG12BOW	Real	0.0	eV
EG1ALPH	Real	0.0	eV/K
EG2ALPH	Real	0.0	eV/K
EG1BETA	Real	0.0	K
EG2BETA	Real	0.0	K
EGALPHA	Real	see Appendix B	eV/K
EGBETA	Real	see Appendix B	K
EMISSION.FACTOR	Real	27.1728E-5	
EN	Real	0.0	
EP	Real	0.0	
EPS11	Real	0.0	
EPS12	Real	0.0	
EPS13	Real	0.0	
EPS22	Real	0.0	
EPS23	Real	0.0	
EPS33	Real	0.0	
ESTR	Real	0.0	
ETRAP	Real	0.0	eV
EPSINF	Real		eV/K
EXN.II	Real	1.0	
EXP.II	Real	1.0	
F.ALPHAA	Character		
F.BANDCOMP	Character		
F.BGN	Character		
F.CBDOSFN	Character		
F.CONMUN	Character		
F.CONMUP	Character		

Parameter	Type	Default	Units
F.COPT	Character		
F.EPSILON	Character		
F.ENMUN	Character		
F.ENMUP	Character		
F.FERRO	Character		
F.GAUN	Character		
F.GAUP	Character		
F.INDEX	Character		
F.MNSNDIFF	Character		
F.MNSPDIFF	Character		
F.MNSNFLICKER	Character		
F.MNSPFLICKER	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.RECOMB	Character		
F.TAUN	Character		
F.TAUP	Character		
F.TAURN	Character		
F.TAURP	Character		
F.TCAP	Character		
F.TCOND	Character		
F.VBDOSFN	Character		
F.VSATN	Character		
F.VSATP	Character		
FCN	Real	3.0×10^{-18}	cm^{-2}
FCP	Real	7.0×10^{-18}	cm^{-2}
FERRO.EC	Real	0.0	V/cm
FERRO.EPSF	Real	1.0	
FERRO.PS	Real	0.0	C/sqcm
FERRO.PR	Real	0.0	C/sqcm
FSTR	Real	0.0	
GAIN.SAT	Real	1.0E7	

Parameter	Type	Default	Units
GAIN0	Real	2000.0	cm^{-1}
GAIN00	Real	-200.0	cm^{-1}
GAIN1N	Real	0	cm^2
GAIN1P	Real	0	cm^2
GAIN1MIN	Real	3.0×10^{-16}	cm^2
GAIN2NP	Real	0	cm^5
GAMMA	Real		
GCB	Real	2.0	
GN1	Real	3.0×10^{-16}	cm^{-2}
GN2	Real	4.0×10^{-15}	cm^{-2}
G.SURF	Real	1.0	cm^2
GVB	Real	4.0	
HC.A	Real	See Appendix B	J/Kcm^3
HC.B	Real	See Appendix B	J/K^2cm^3
HC.C	Real	See Appendix B	J/K^3cm^3
HC.D	Real	See Appendix B	JK/cm^3
HOOGEN	Real	0.0	
HOOGEP	Real	0.0	
HOPN.BETA	Real	1.5	
HOPN.GAMMA	Real	5×10^{-7}	cm^{-1}
HOPN.V0	Real	1×10^{11}	Hz
HOPP.BETA	Real	1.5	
HOPP.GAMMA	Real	1×10^{11}	cm^{-1}
HOPP.V0	Real	1×10^{11}	Hz
IG.ELINR	Real	6.16×10^{-6}	cm
IG.HLINR	Real	6.16×10^{-6}	cm
IG.ELINF	Real	9.2×10^{-7}	cm
IG.HLINF	Real	9.2×10^{-7}	cm
IMAG.INDEX	Real	See Appendix B	

Parameter	Type	Default	Units
INDEX.FILE	Character		
KAUGCN	Real	1.83×10^{-31}	cm^6/s
KAUGCP	Real	2.78×10^{-31}	cm^6/s
KAUGDN	Real	1.18	
KAUGDP	Real	0.72	
KISC.EXCITON	Real	0	s^{-1}
KNRS.EXCITON	Real	0	s^{-1}
KNRT.EXCITON	Real	0	s^{-1}
KSP.EXCITON	Real	0	$cm^{-3}s^{-1}$
KSRHCN	Real	3.0×10^{-13}	cm^3
KSRHCP	Real	11.76×10^{-13}	cm^3
KSRHGN	Real	1.77	
KSRHGP	Real	0.57	
KSRHTN	Real	2.5×10^{-3}	s
KSRHTP	Real	2.5×10^{-3}	s
KSS.EXCITON	Real	0	$cm^{-3}s^{-1}$
KST.EXCITON	Real	0	$cm^{-3}s^{-1}$
KTP.EXCITON	Real	0	$cm^{-3}s^{-1}$
KTT.EXCITON	Real	0	$cm^{-3}s^{-1}$
L1SELL	Real	Appendix B	µm
L2SELL	Real	Appendix B	µm
LAMDAE	Real	6.2×10^{-7}	cm
LAMDAH	Real	3.8×10^{-7}	cm
LAMHN	Real	9.2×10^{-7}	cm
LAMHP	Real	9.2×10^{-7}	cm
LAMRN	Real	6.16×10^{-6}	cm
LAMRP	Real	6.16×10^{-6}	cm
LAN300	Real	6.2×10^{-7}	cm
LAP300	Real	3.8×10^{-7}	cm

Parameter	Type	Default	Units
LDS.EXCITON	Real	0.01	µm
LDT.EXCITON	Real	0.632	µm
LT.TAUN	Real	0.0	
LT.TAUP	Real	0.0	
LUTT1	Real	0.0	
LUTT2	Real	0.0	
LUTT3	Real	0.0	
M.DSN	Real	See Appendix B	
M.DSP	Real	See Appendix B	
M.VTHN	Real		
M.VTHP	Real		
MATERIAL	Character		
MC	Real	0.0	
ME.TUNNEL	Real		
MH.TUNNEL	Real		
ME.SBT	Real		
MH.SBT	Real		
MHH	Real	0.49	
MINIMA	Real	6	
ML	Real	0.916	
MLH	Real	0.16	
MSO	Real	0.23	
MSTAR	Real	0.0	
MTT	Real	0.0	
MT1	Real	0.191	
MT2	Real	0.191	
MUN	Real	See Appendix B	
MUP	Real	See Appendix B	
MV	Real	0.0	
MZZ	Real	0.0	
N.ION.1	Real	0.0	$cm^{-1}K^{-1}$
N.ION.2	Real	0.0	$cm^{-1}K^{-2}$

Parameter	Type	Default	Units
N.IONIZA	Real	7.03×10^{5}	cm^{-1}
N0.BGN	Real	1.0×10^{17}	cm^{-3}
NAME	Character		
NC.F	Real	1.5	
NC300	Real	See Appendix B	cm^{-3}
NDX.ADACHI	Logical	False	
NDX.SELLMEIER	Logical	False	
NI.MIN	Real	0.0	cm^{-3}
NSRHN	Real	See Appendix B	cm^{-3}
NSRHP	Real	See Appendix B	cm^{-3}
NTRANSPARENT	Real	2.0×10^{18}	cm^{-3}
NUE.EXTR	Real		
NUH.EXTR	Real		
NV.F	Real	1.5	
NV300	Real	See Appendix B	cm^{-3}
OP.PH.EN	Real	0.063	eV
OPPHE	Real	0.063	eV
OXCH.ONLY	Logical	False	
P.PASSLER	Real	2.33	
PERM.ANISO	Real	-999.0	
PERMITTIVITY	Real	See Appendix B	
POWER	Real		
PSP	Real	0.0	C/m2
P.ION.1	Real	0.0	$cm^{-1}K^{-1}$
P.ION.2	Real	0.0	$cm^{-1}K^{-1}$
P.IONIZA	Real	1.682×10^{6}	cm^{-1}
REAL.INDEX	Real	See Appendix B	
REGION		All regions	
RESISTIVITY	Real		$\mu\Omega\cdot cm$
RST.EXCITON	Real	0.25	

Parameter	Type	Default	Units
S0SELL	Real	See Appendix B	
S1SELL	Real	See Appendix B	
S2SELL	Real	See Appendix B	
SEMICONDUC	Logical	False	
STABLE	Integer		
STRUCTURE	Character		
T.PASSLER	Real	406.0	
TANI.CONST	Logical	False	
TANI.POWER	Logical	False	
TANI.POLYNOM	Logical	False	
TANI.RECIP	Logical	false	
TAUMOB.EL	Real	0.4×10^{-12}	s
TAUMOB.HO	Real	0.4×10^{-12}	s
TAUN0	Real	See Appendix B	s
TAUP0	Real	See Appendix B	s
TAUREL.EL	Real	0.4×10^{-12}	s
TAUREL.HO	Real	0.4×10^{-12}	s
TAUS.EXCITON	Real	1×10^{-9}	s
TAUT.EXCITON	Real	1×10^{-9}	s
TC.A	Real	See Appendix B	$\frac{cm \cdot K}{W}$
TC.B	Real	See Appendix B	$\frac{cm}{W}$
TC.C	Real	See Appendix B	$\frac{cm}{W \cdot K}$
TC.CONST	Real		W/cm·K
TC.NPOW	Real		
TC.D	Real		K
TC.E	Real		W/cm
TC.FULL.ANISO	Logical	False	
TCON.CONST	Logical		
TCON.POWER	Logical		

Parameter	Type	Default	Units
TCON.POLYNOM	Logical		
TCON.RECIPRO	Logical		
TE.MODES	Logical	False	
TMUN	Real		
TMUP	Real		
TRE.T1	Real		s
TRE.T2	Real		s
TRE.T3	Real		s
TRE.W1	Real		eV
TRE.W2	Real		eV
TRE.W3	Real		eV
TRH.T1	Real		s
TRH.T2	Real		s
TRH.T3	Real		s
TRH.W1	Real		eV
TRH.W2	Real		eV
TRH.W3	Real		eV
U.DEFPOT	Real	10.5	eV
UBGN.B	Real	3.1×10^{12}	
UBGN.C	Real	-3.9×10^{-5}	
USER.DEFAULT	Character		
USER.GROUP	Character	SEMICONDUCTOR	
V0.BGN	Real	9.0×10^{-3}	eV
VAL.AN0	Real	4.3383	
VAL.AN1	Real	-2.42e-12	
VAL.AN2	Real	4.1233	
VAL.AP0	Real	2.376	
VAL.AP1	Real	0.01033	
VAL.AP2	Real	1.0	
VAL.BN0	Real	0.235	
VAL.BN1	Real	0.0	

Parameter	Type	Default	Units
VAL.CN0	Real	1.6831e4	
VAL.CN1	Real	4.3796	
VAL.CN2	Real	1.0	
VAL.CN3	Real	0.13005	
VAL.DN0	Real	1.233735e6	
VAL.DN1	Real	1.2039e3	
VAL.DN2	Real	0.56703	
VAL.BP0	Real	0.17714	
VAL.BP1	Real	-0.002178	
VAL.CP0	Real	0.0	
VAL.CP1	Real	0.00947	
VAL.CP2	Real	2.4924	
VAL.CP3	Real	0.0	
VAL.DP0	Real	1.4043e6	
VAL.DP1	Real	2.9744e3	
VAL.DP2	Real	1.4829	
VSAT	Real		cm/s
VSATN	Real		cm/s
VSATP	Real		cm/s
WELL.DELTA	Real	0.341	eV
WELL.EPS	Real	0	eV
WELL.GAMMA0	Real	2E-3	eV
WELL.TAUIN	Real	3.3E-13	s
X.X	Real	1.0	
X.Y	Real	0.0	
X.Z	Real	0.0	
YDIR.ANISO	Logical	False	
Y.X	Real	0.0	
Y.Y	Real	1.0	
Y.Z	Real	0.0	

Parameter	Type	Default	Units
YDIR.ANISO	Logical	False	
ZDIR.ANISO	Logical	True	
Z.X	Real	0.0	
Z.Y	Real	0.0	
Z.Z	Real	1.0	

Description

The MATERIAL statement is used set basic material parameters related to band structure and parameters for certain mobility, recombination or carrier statistics models. Parameters for temperature dependence are noted in a separate section below.

Localization of Material Parameters

DEVICE specifies which device the MATERIAL statement should apply to in MIXEDMODE simulation. The synonym for this parameter is STRUCTURE.

MATERIAL specifies which material from the table in Appendix B: "Material Systems" that the MATERIAL statement should apply. If a material is specified, then all regions defined as being composed of that material will be affected.

Note: You can specify the following logical parameters to indicate the material instead of assigning the MATERIAL parameter: SILICON, GAAS, POLYSILI, GERMAINU, SIC, SEMICOND, SIGE, ALGAAS, A-SILICO, DIAMOND, HGCDTE, INAS, INGAAS, INP, S.OXIDE, ZNSE, ZNTE, ALINAS, GAASP, INGAP and MINASP.

NAME specifies which region the MATERIAL statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.

REGION specifies the region number to which these parameters apply. If there is more than one semiconductor region, specification of different parameters for each region is allowed. If REGION is not specified, all regions in the structure are changed.

STRUCTURE is a synonym for DEVICE.

NDX.ADACHI enables Adachi's refractive index model. See Chapter 3: "Physics", Equation 3-506.

NDX.SELLMEIER enables Sellmeier's refractive index model. See Chapter 3: "Physics", Equation 3-505.

AADACHI, **BADACHI** and **DADACHI** are the parameters for Adachi's refractive index model. See Chapter 3: "Physics", Equation 3-506.

S0SELL, **S1SELL**, **S2SELL**, **L1SELL** and **L2SELL** are parameters for Sellmeier's refractive index model. See Chapter 3: "Physics", Equation 3-505.

Band Structure Parameters

A.DEFPOT, **B.DEFPOT**, **C.DEFPOT**, **D.DEFPOT**, and **U.DEFPOT** specify the deformation potentials used in calculating the effects of strains on the bandgap of silicon (see Section 3.6.11).

AFFINITY specifies the electron affinity.

ALIGN specifies the fraction of the bandgap difference that is applied to the conduction band edge, relative to the minimum bandgap material in the device. Note that specifying this parameter overrides any electron affinity specification. See the Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.1.2: "Alignment" for information on setting the band alignment.

ARICHN specifies the effective Richardson constant for electrons.

ARICHP specifies the effective Richardson constant for holes.

D.TUNNEL specifies the maximum tunneling distance for the universal Schottky tunneling model (see Chapter 3: "Physics", Section 3.5.2: "Schottky Contacts"). The alias for this parameter is `DIST.SBT`.

DIST.SBT is an alias for `D.TUNNEL`.

EG300 specifies energy gap at 300K (see Chapter 3: "Physics", Equation 3-38). All semiconductor materials in ATLAS must have a defined `EG300`.

EG1300, **EG2300**, **EG12BOW**, **EG1ALPH**, **EG2ALPH**, **EG1BETA**, and **EG2BETA** specify parameters of the General Ternary bandgap model described in Chapter 3: "Physics", Equations 3-42 through 3-44.

EPS11, **EPS12**, **EPS13**, **EPS22**, **EPS23**, **EPS33** specify the strain tensor used in the calculation of the strain effects on bandgap of silicon (see Section 3.6.11).

F.BANDCOMP specifies the name of a file containing a C-INTERPRETER function that defines temperature and composition dependent band parameter models.

F.CBDOSFN specifies the C-Interpreter file for the Conduction band effective density of states as a function of Lattice/Electron temperature. This function is called `cbdosfn`.

F.TCAP specifies the name of a file containing a C-INTERPRETER function that defines the lattice thermal capacity as a function of the lattice temperature, position, doping and fraction composition.

F.TCOND specifies the name of a file containing a C-INTERPRETER function that defines the lattice thermal conductivity as a function of the lattice temperature, position, doping and fraction composition.

F.EPSILON specifies the name of a file containing a C-INTERPRETER function that defines temperature and composition dependent static dielectric constant models.

F.FERRO specifies the name of a file containing a C-INTERPRETEr function that defines dielectric permittivity as a function of electric field and position (x,y). You need a license to use this parameter.

F.VBDOSFN specifies the C-Interpreter file for the Valence band effective density of states as a function of Lattice/Hole temperature. This function is called `vbdosfn`.

M.DSN specifies the electron density of states effective mass. When specified, it is in Equation 3-31 in Chapter 3: "Physics" to calculate electron density of states.

M.DSP specifies the hole density of states effective mass. When specified it is in Equation 3-32 in Chapter 3: "Physics" to calculate hole density of states.

M.VTHN specifies the electron effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-45).

M.VTHP specifies the hole effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-46).

ME.SBT and **MH.SBT** are aliases for ME.TUNNEL and MH.TUNNEL.

ME.TUNNEL and **MH.TUNNEL** specify the electron and hole effective masses for tunneling used in the universal Schottky tunneling model (see Chapter 3: "Physics", Section 3.5.2: "Schottky Contacts"). The aliases for these parameters are ME.SBT and MH.SBT.

NC.F specifies the conduction band density of states temperature (see Chapter 3: "Physics", Equation 3-31).

NV.F specifies the valence band density of states temperature (see Chapter 3: "Physics", Equation 3-32).

NC300 specifies the conduction band density at 300K. (see Chapter 3: "Physics", Equation 3-31).

NI.MIN specifies the minimum allowable value of the intrinsic carrier density.

NV300 specifies valence band density at 300K (see Chapter 3: "Physics", Equation 3-32).

PERMITTIVITY specifies dielectric permittivity of the material. All materials in an ATLAS structure must have a defined permittivity.

BQP Parameters

BQP.NGAMMA and **BQP.PGAMMA** parameters allow you to set the γ parameter of the BQP model for electrons and holes respectively.

BQP.NALPHA and **BQP.PALPHA** parameters allow you to set the α parameter of the BQP model for electrons and holes respectively.

Mobility Model Parameters

F.CONMUN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent electron mobility models.

F.CONMUP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent hole mobility models.

F.ENMUN specifies a C-Interpreter file for the electron mobility as a function of Electron Temperature and perpendicular field as well as other choice variables. The function itself is named endepmun and only applies to ATLAS2D.

F.ENMUP specifies a C-Interpreter file for the hole mobility as a function of Hole Temperature and perpendicular field as well as other choice variables. The function itself is named endepmup and only applies to ATLAS2D.

F.MUNSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent electron mobility model for velocity saturation.

F.MUPSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent hole mobility model for velocity saturation.

F.VSATN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent electron saturation velocity models.

F.VSATP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent hole saturation velocity models.

GSURF specifies a factor by which mobility is reduced at the semiconductor surface.This is a simple but not accurate alternative to the transverse field dependent or surface mobility models set on the MODELS statement.

MUN specifies low-field electron mobility. This parameter is only used if no concentration dependent mobility model is specified.

MUP specifies low-field hole mobility.This parameter is only used if no concentration dependent mobility model is specified.

VSATURATION specifies the saturation velocity for the electric field dependent mobility.

VSATN specifies the saturation velocity for electrons.

VSATP specifies the saturation velocity for holes.

Recombination Model Parameters

AN, **AP**, **BN**, **BP**, **CN**, **EN**, and **EP** parameters of the concentration dependent lifetime model (see Chapter 3: "Physics", Equations 3-283 and 3-284).

AUGN specifies the Auger coefficient, *cn* (see Chapter 3: "Physics", Equation 3-294).

AUGP Specifies the Auger coefficient, *cp* (see Chapter 3: "Physics", Equation 3-294).

AUGKN parameter of the narrow band-gap electron Auger recombination coefficient model.

AUGKP parameter of the narrow band-gap electron Auger recombination coefficient model.

BB.A, **BB.B** and **BB.GAMMA** specify the band-to-band tunneling parameters (see Chapter 3: "Physics", Equation 3-368).

C.DIRECT is an alias for `COPT`.

COPT specifies the optical recombination rate for the material. This parameter has no meaning unless `MODELS OPTR` has been specified (see Chapter 3: "Physics", Equation 3-286). The alias for this parameter is `C.DIRECT`.

ETRAP specifies the trap energy for SRH recombination

F.COPT specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the radiative recombination rate.

F.GAUN specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the electron Auger coefficient.

F.GAUP specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the hole Auger coefficient.

F.RECOMB specifies the name of a file containing a C-Interpreter function for the specification of temperature, composition, electron and hole concentration dependent recombination rate models.

F.TAUN specifies the name of a file containing a C-INTERPRETER function for the specification of position dependent electron lifetime models.

F.TAUP specifies the name of a file containing a C-INTERPRETER function for the specification of position dependent hole lifetime models.

F.TAURN specifies the name of a file containing a C-INTERPRETER function specifying the electron relaxation time as a function of electron energy.

F.TAURP specifies the name of a file containing a C-INTERPRETER function specifying the hole relaxation time as a function of hole energy.

Impact Ionization Parameters

AN0.VALD is an alias for `VAL.AN0`.

AN1.VALD is an alias for `VAL.AN1`.

AN2.VALD is an alias for `VAL.AN2`.

AP0.VALD is an alias for `VAL.AP0`.

AP1.VALD is an alias for `VAL.AP1`.

AP2.VALD is an alias for VAL.AP2.

BETAN for electrons and **BETAP** for holes correspond to coefficients for the power of ECRIT/E. The aliases for these parameters are **EXN.II** and **EXP.II**.

BN0.VALD is an alias for VAL.BN0.

BN1.VALD is an alias for VAL.BN1.

BP0.VALD is an alias for VAL.BP0.

BP1.VALD is an alias for VAL.BP1.

CN2, **CP2**, **DN2** and **DP2** are specifiable coefficients in the temperature dependent models described in Chapter 3: "Physics", Equations 3-312 and 3-313. The aliases for these parameter are N.ION.1, P.ION.1, N.ION.2 and P.ION.2.

CN0.VALD is an alias for VAL.CN0.

CN1.VALD is an alias for VAL.CN1.

CN2.VALD is an alias for VAL.CN2.

CN3.VALD is an alias for VAL.CN3.

CP0.VALD is an alias for VAL.CP0.

CP1.VALD is an alias for VAL.CP1.

CP2.VALD is an alias for VAL.CP2.

CP3.VALD is an alias for VAL.CP3.

DN0.VALD is an alias for VAL.DN0.

DN1.VALD is an alias for VAL.DN1.

DN2.VALD is an alias for VAL.DN2.

DP0.VALD is an alias for VAL.DP0.

DP1.VALD is an alias for VAL.DP1.

DP2.VALD is an alias for VAL.DP2.

ECN.II is an alias for BN2.

ECP.II is an alias for BP2.

EXN.II is an alias for BETAN.

EXP.II is an alias for BETAP.

LAMBDAE specifies the mean free path for electrons. The alias for this parameter is LAN300.

LAMDAH specifies the mean free path for holes . The alias for this parameter is LAP300.

LAN300 is an alias for LAMBDAE.

LAP300 is an alias for LAMBDAH.

N.ION.1, **P.ION.1**, **N.ION.2** and **P.ION.2** are the aliases for CN2, CP2, DN2 and DP2.

N.IONIZA is an alias for AN2.

OP.PH.EN is an alias for OPPHE.

OPPHE specifies the optical phonon energy. The alias for this parameter is OP.PH.EN.

P.IONIZA is an alias for AP2.

VAL.AN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is AN0.VALD.

VAL.AN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is AN1.VALD.

VAL.AN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is AN2.VALD.

VAL.BN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is BN0.VALD.

VAL.BN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is BN1.VALD.

VAL.CN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is CN0.VALD.

VAL.CN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is CN1.VALD.

VAL.CN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is CN2.VALD.

VAL.CN3 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is CN3.VALD.

VAL.DN0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is DN0.VALD.

VAL.DN1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is DN1.VALD.

VAL.DN2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-270. The alias for this parameter is DN2.VALD.

VAL.AP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271.

VAL.AP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is AP1.VALD.

VAL.AP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is AP2.VALD.

VAL.BP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is BP0.VALD.

VAL.BP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is BP1.VALD.

VAL.CP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is CP0.VALD.

VAL.CP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is CP1.VALD.

VAL.CP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is CP2.VALD.

VAL.CP3 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is CP3.VALD.

VAL.DP0 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: “Physics”, Equation 3-271. The alias for this parameter is DP0.VALD.

VAL.DP1 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-271. The alias for this parameter is DP1.VALD.

VAL.DP2 specifies the value of a temperature dependent impact ionization parameter in Chapter 3: "Physics", Equation 3-271. The alias for this parameter is DP2.VALD.

Klaassen Model parameters

KSRHTN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHTP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHCN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHCP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHGN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHGP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KAUGCN coefficient for Klaassen's concentration dependent Auger model.

KAUGCP coefficient for Klaassen's concentration dependent Auger model.

KAUGDN coefficient for Klaassen's concentration dependent Auger model.

KAUGDP coefficient for Klaassen's concentration dependent Auger model.

NSRHN specifies the SRH concentration parameter for electrons (see Chapter 3: "Physics", Equation 3-283).

NSRHP specifies the SRH concentration parameter for holes (see Chapter 3: "Physics", Equation 3-284).

TAUN0 specifies SRH lifetime for electrons (see Chapter 3: "Physics", Equation).

TAUP0 specifies SRH lifetime for holes (see Chapter 3: "Physics", Equation 3-284).

See also Chapter 3: "Physics", Equations 3-285 and 3-286 and Equations 3-295 and 3-297 for more information about Klaassen models.

Carrier Statistics Model Parameters

ASYMMETRY specifies the relative degree where band gap narrowing applies to the conduction band versus the valence band. The value of the ASYMMETRY parameter is multiplied by the total change in band gap due to band gap narrowing and that product is applied to the conduction band edge. For example, if the ASYMMETRY parameter has a value of 1.0, then the change in band gap due to band gap narrowing is applied only to the conduction band edge and the valence band edge remains unaffected.

BGN.E, **BGN.N** and **BGN.C** specify the parameters of the band gap narrowing model given in Chapter 3: "Physics", Equation 3-46. The aliases for these parameters are V0.BGN, N0.BGN, and C0N.BGN.

CON.BGN is an alias for BGN.C.

EAB specifies acceptor energy level (see Chapter 3: "Physics", Equation 3-53).

EDB specifies donor energy level (see Chapter 3: "Physics", Equation 3-52).

F.BGN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent bandgap narrowing models.

GCB specifies the conduction-band degeneracy factor (see Chapter 3: "Physics", Equation 3-52).

GVB specifies the valence-band degeneracy factor (see Chapter 3: "Physics", Equation 3-53).

N0.BGN is an alias for BGN.N.

V0.BGN is an alias for BGN.E.

Energy Balance Parameters

TAUMOB.**EL** specifies the relaxation time for electrons in the temperature dependent mobility model (see Chapter 3: "Physics", Equation 3-129).

TAUMOB.**HO** specifies the relaxation time for holes in the temperature dependent mobility model (Chapter 3: "Physics", Equation 3-130).

TAUREL.**EL** specifies the relaxation time for electrons in the energy balance model (see Chapter 3: "Physics", Equation 3-123).

TAUREL.**HO** specifies the relaxation time for holes in the energy balance model (see Chapter 3: "Physics", Equation 3-124).

TRE.**T1**, **TRE**.**T2**, **TRE**.**T3**, **TRE**.**W1**, **TRE**.**W2**, **TRE**.**W3**, **TRH**.**T1**, **TRH**.**T2**, **TRH**.**T3**, **TRH**.**W1**, **TRH**.**W2**, and **TRH**.**W3** are used in the temperature dependent energy relaxation time model based on table data from Laux-Fischetti Monte-Carlo simulation (see Chapter 3: "Physics", Table 3-12).

Hot Carrier Injection Parameters

IG.ELINR specifies the electron mean free path between redirecting collisions. The alias for this parameter is LAMRN.

IG.HLINR specifies the hole mean free path between redirecting collisions. The alias for this parameter is LAMRP.

IG.ELINF specifies the electron mean free path length for scattering by optical phonons. The alias for this parameter is LAMHN.

IG.HLINF specifies the hole mean free path length for scattering by optical phonons. The alias for this parameter is LAMHP.

LAMRN is an alias for IG.ELINR.

LAMRP is an alias for IG.HLINR.

LAMHN is an alias for IG.ELINF.

LAMHP is an alias for IG.HLINF.

Lattice Temperature Dependence Parameters

A.PASSLER specifies a parameter of the Passler temperature dependent bandgap model given in Chapter 3: "Physics", Equation 3-39.

EGALPHA specifies the alpha coefficient for temperature dependence of bandgap (see Chapter 3: "Physics", Equation 3-38).

EGBETA specifies the beta coefficient for temperature dependence of bandgap (see Chapter 3: "Physics", Equation 3-38).

HC.**A**, **HC**.**B**, **HC**.**C**, and **HC**.**D** specify the values of the four coefficient of the heat capacity equation (see Chapter 7: "Giga: Self-Heating Simulator", Equation 7-12).

LT.**TAUN** specifies the temperature dependence for electron lifetimes (see Chapter 7: "Giga: Self-Heating Simulator", Equation 7-26).

LT.**TAUP** specifies the temperature dependence for hole lifetimes (see Chapter 7: "Giga: Self-Heating Simulator", Equation 7-27).

P.PASSLER specifies a parameter of the Passler temperature dependent bandgap model given in Chapter 3: "Physics", Equation 3-39.

POWER specifies the value of thermal power generated in a power source associated with a region in THERMAL3D (see Chapter 17: "Thermal 3D: Thermal Packaging Simulator").

T.PASSLER specifies a parameter of the Passler temperature dependent bandgap model given in Chapter 3: "Physics", Equation 3-44.

TC.**A**, **TC**.**B**, and **TC**.**C** specify the three thermal conductivity coefficients (see Chapter 7: "Giga: Self-Heating Simulator", Equation 7-9).

TC.**C0** specifies the equilibrium value of thermal conductivity, *k(T0)*, in Chapter 17: "Thermal 3D: Thermal Packaging Simulator", Equation 17-2. The synonym for this parameter is `TC.CONST`.

TC.**D** specifies the value of the parameter D in Chapter 17: "Thermal 3D: Thermal Packaging Simulator", Equation 17-5.

TC.**E** specifies the value of the parameter E in Chapter 17: "Thermal 3D: Thermal Packaging Simulator", Equation 17-5.

TCON.**CONST** specifies that thermal conductivity should be modeled as constant with respect to temperature. The value of the thermal conductivity is given by the value of the `TC.C0` parameter.

TCON.**POWER** specifies that the temperature dependence of thermal conductivity should be modeled using Equation 17-3 in Chapter 17: "Thermal 3D: Thermal Packaging Simulator".

TCON.**POLY** specifies that the temperature dependence of thermal conductivity should be modeled using equation Equation 17-4 in Chapter 17: "Thermal 3D: Thermal Packaging Simulator".

TCON.**RECIP** specifies that the temperature dependence of thermal conductivity should be modeled using equation Equation 17-5 in Chapter 17: "Thermal 3D: Thermal Packaging Simulator".

TC.**NPOW** specifies the value of the coefficient of temperature dependence of thermal conductivity, n, in equation Equation 17-3 in Chapter 17: "Thermal 3D: Thermal Packaging Simulator".

TMUN and **TMUP** specify the lattice temperature coefficients for the temperature dependence of electron lattice mobility, and of hole lattice mobility respectively.

UBGN.B and **UBGN.C** define the numerical values of the fitting parameters for Equation 3-50.

Oxide Material Parameters

SEMICONDUC specifies that an oxide region is to be treated as a semiconductor.

OXCH.ONLY specifies that electron and hole concentrations are omitted from Poisson's Equation in oxides.

Photogeneration Parameters

F.**INDEX** specifies the name of a file containing a C-INTERPRETER function for the specification of wavelength dependent complex index of refraction models.

IMAG.**INDEX** specifies the imaginary portion of the refractive index of the semiconductor (see Chapter 10: "Luminous: Optoelectronic Simulator", Equation 10-33). Wavelength dependent defaults exist for certain materials as documented in Appendix B: "Material Systems".

INDEX.**FILE** specifies the filename from which refractive indices for a material are read. This parameter is useful to load wavelength dependent refractive indices for use with both mono and multi-spectral light sources. The format of this file is:

```
<n>
wavelength(1) real index(1) imaginary index(1)
wavelength(2) real index(2) imaginary index(2)
...
```

```
wavelength(n) real index(n) imaginary index(n)
```

where n is the number of lines to be read.

Note: The index file must be ordered by increasing wavelength.

REAL.INDEX specifies the real portion of the refractive index of the semiconductor. Wavelength dependent defaults exist for certain materials as documented in Appendix B: “Material Systems”.

LASER Parameters

ABSORPTION.SAT specifies the absorption saturation intensity used in the non-linear absorption loss model given in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-16.

ALPHAA specifies the bulk absorption coefficient in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-2.

ALPHAR specifies the line width broadening factor in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-2.

EMISSION.FACTOR specifies a scale factor accounting the proportion of light directionally coupled into the lasing mode.

EPSINF specifies the high frequency relative dielectric permittivity ($\varepsilon\infty$) (see Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-2). If this parameter is not specified, it will be set equal to the static dielectric permittivity of the material.

F.ALPHAA specifies the name of a file containing a C-INTERPRETER function for the bulk absorption coefficient.

FCN is the free-carrier loss model parameter in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-3.

FCP is the free-carrier loss model parameter in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-3.

GAIN.SAT specifies the non-linear gain saturation factor used in Chapter 8: “Laser: Edge Emitting Simulator”, Equation 8-14.

GAIN0 specifies the parameter in Chapter 3: “Physics”, Sections 3.9.2: “The Default Radiative Recombination Model” and 3.9.3: “The Standard Gain Model”.

GAIN00 specifies the parameter in Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”.

GAIN1N specifies the parameter in Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”.

GAIN1P specifies the parameter in Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”.

GAIN1MIN specifies the parameter in Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”.

GAIN2NP specifies the parameter in Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”.

GAMMA specifies the parameter in Chapter 3: “Physics”, Sections 3.9.2: “The Default Radiative Recombination Model” and 3.9.3: “The Standard Gain Model”. If this parameter is not specified, it will be calculated using Equation 3-459.

GN1 specifies the parameter in Chapter 3: “Physics”, Section 3.9.5: “Tayamaya's Gain Model”.

GN2 specifies the parameter in Chapter 3: “Physics”, Section 3.9.5: “Tayamaya's Gain Model”.

NTRANSPARENT specifies the parameter in Chapter 3: “Physics”, Section 3.9.5: “Tayamaya's Gain Model”.

NOISE Parameters

F.MNSNDIFF specifies the name of a file containing a C-Interpreter function for the microscopic noise source for electron diffusion noise.

F.MNSPDIFF specifies the name of a file containing a C-Interpreter function for the microscopic noise source for hole diffusion noise.

F.MNSNFLICKER specifies the name of a file containing a C-Interpreter function for the microscopic noise source for electron flicker noise.

F.MNSPFLICKER specifies the name of a file containing a C-Interpreter function for the microscopic noise source for hole flicker noise.

HOOGEN specifies the Hooge constant for electron flicker noise.

HOOGEP specifies the Hooge constant for hole flicker noise.

Organic Transport Parameters

HOPN.BETA specifies the percolation constant for electrons.

HOPN.GAMMA specifies 1/carrier localization radius for electrons.

HOPN.V0 specifies the attempt-to-jump frequency for electrons.

HOPP.BETA specifies the percolation constant for holes.

HOPP.GAMMA specifies 1/carrier localization radius for holes.

HOPP.V0 specifies the attempt-to-jump frequency for holes.

Exciton Material Parameters

KISC.EXCITON specifies the exciton intersystem crossing constant.

KNRS.EXCITON specifies the singlet non-radiative decay constant.

KNRT.EXCITON specifies the triplet non-radiative decay constant.

KSP.EXCITON specifies the singlet-polaron constant.

KST.EXCITON specifies the singlet-triplet constant.

KTP.EXCITON specifies the triplet-polaron constant.

KSS.EXCITON specifies the singlet-singlet constant.

KTT.EXCITON specifies the triplet-triplet constant.

LDS.EXCITON specifies the singlet diffusion length. The synonym for this parameter is `LD.EXCITON`.

LDT.EXCITON specifies the triplet diffusion length.

RST.EXCITON specifies the fraction of singlets formed.

TAUS.EXCITON specifies the singlet radiative decay lifetime. The synonym for this parameter is `TAU.EXCITON`.

TAUT.EXCITON specifies the triplet radiative decay lifetime.

Miscellaneous Material Parameters

A1, **A2**, **A3**, **A4**, **A5** and **A6** specify valence band effective mass parameters.

AC, **AV** and **BBB** specify cubic deformation potentials.

ALATTICE specifies the in plane lattice constant.

ASTR, **BSTR**, **CSTR**, **DSTR**, **ESTR** and **FSTR** are user-definable parameters of the equations for heavy and light hole masses in Ishikawa's strain model (see Chapter 3: “Physics”, Section 3.9.10: “Ishikawa's Strain Effects Model”).

C11, **C12**, **C13** and **C33** specify the elastic stiffness coeficients.

DEGENERACY specifies the spin degeneracy.

DELTA1, **DELTA2** and **DELTA3** specify the valence band energy splits.

DGN.GAMMA and **DGP.GAMMA** specify the electron and hole tuning parameters for density gradient modeling.

DINDEXDT specifies the temperature coefficient of the index of refraction.

DRHODT specifies the temperature coefficient of conductor resistivity.

D1, **D2**, **D3**, and **D4** specify shear deformation potentials.

E31, **E33** and **PSP** specify piezo-electric constants.

LUTT1, **LUTT2** and **LUTT3** specify the luttinger parameters.

M.VTHN and **M.VTHP** specify the electron and hole effective masses used for calculation of thermal velocity.

MC specifies the conduction band effective mass.

MINIMA specifies the number of equavilent minima in the conduction band energy.

MHH specifies the heavy hole effecitive mass.

ML specifies the conduction band longitudinal effective mass.

MLH specifies the light hole effective mass.

MSO specifies the split off (crystal lattice) effective mass.

MSTAR specifies the conduction band effective mass dispersion.

MTT specifies the transverse effective mass (wurtzite).

MT1 and **MT2** specify the transverse effective masses (zincblende).

MV specifies the valence band effective mass.

MZZ specifies the effective mass along the crystal axis (wurtzite).

PSP specifies the spontanteous polarization.

RESISTIVITY specifies the resistivity of conductor regions.

STABLE specifies the selected strain table for the Ishikawa strain effects model. See Chapter 3: “Physics”, Section 3.9.10: “Ishikawa's Strain Effects Model”.

TE.MODES specifies whether TE or TM modes will be used for calculation of assymetry factors for the LI model (see Chapter 13: “Quantum: Quantum Effect Simulator”, Section 13.7: “Multiple Quantum Well Model”).

USER.DEFAULT specifies which material the user-defined material should use for its default parameters.

USER.GROUP specifies the material group for the user-defined material. `USER.GROUP` can be either `SEMICONDUCTOR`, `INSULATOR`, or `CONDUCTOR`.

WELL.DELTA specifies the spin orbital splitting energy in a quantum well.

WELL.EPS specifies the high frequency permittivity used in calculating gain and radiative recombination in a quantum well.

WELL.GAMMA0 specifies the Lorentzian gain broadening factor from Chapter 3: "Physics", Section 3.9.9: "Lorentzian Gain Broadening".

WELL.TAUIN specifies the Lorentzian gain broadening factor from Chapter 3: "Physics", Section 3.9.9: "Lorentzian Gain Broadening".

Conductor Parameters

DRHODT specifies the temperature coefficient of resistivity in μΩ·cm/K.

RESISTIVITY specifies the material resisitivity in 0Ωcm.

Material Coefficient Definition Examples

Numbered region

This example specifies SRH lifetimes and concentration independent low-field mobilities for region number 2. All other parameters use default values and parameters in other regions are unaffected.

```
MATERIAL TAUN0=5.0E-6 TAUP0=5.0E-6 MUN=3000 MUP=500 REGION=2
```

All regions

This example defines carrier lifetimes and the refractive index for all semiconductor regions.

```
MATERIAL TAUP0=2.E-6 TAUN0=2.E-6 REAL.INDEX=3.7 \
         IMAG.INDEX=1.0E-2
```

Named Material

This shows the definition of bandgap for all InGaAs regions in the structure:

```
MATERIAL MATERIAL=InGaAs EG300=2.8
```

All materials are divided into three classes: semiconductors, insulators and conductors. See Appendix B: "Material Systems" for more information about the parameters required for each material class.

Note: You can use the `MODEL PRINT` command to echo back default material parameters or `MATERIAL` parameter settings to the run-time output.

19.25: MEASURE

MEASURE extracts selected electrical data from the solution.

Note: This statement is almost obsolete. Its functions have been replaced by the EXTRACT , OUTPUT , or PROBE statement.

Syntax

```
MEASURE <dt> [<boundary>] [OUTFILE=<filename>]
```

Parameter	Type	Default	Units
CONTACT	Integer		
E.CRIT	Real1E-8		
ELECTRON	Logical	False	
HOLE	Logical	False	
IONIZINT	Logical	False	
LRATIO	Real	1.0	
METAL.CH	Logical	False	
N.CURRENT	Logical	False	
N.LAYER	Real	15	
N.LINES	Integer	50	
N.RESIST	Logical	False	
NET.CARR	Logical	False	
NET.CHAR	Logical	False	
OUTFILE	Character		
P.CURRENT	Logical	False	
P.RESIST	Logical	False	
REGIONS	Integer	All regions	
SUBSTR	Character		
U.AUGER	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
U.TOTAL	Logical	False	
X.MIN	Real	Left of device	μm

Parameter	Type	Default	Units
X.MAX	Real	Right of device	μm
Y.MIN	Real	Top of device	μm
Y.MAX	Real	Bottom of device	μm

Description

dt is used to specify the type of information to be measured.

boundary specifies which nodes will be measured.

OUTFILE specifies a filename where simulation results and bias information will be written.

Data Type Parameters

Net carrier concentration, charge concentration, electron concentration, or hole concentration can be integrated over a section of a device. The charge on part of an electrode can be calculated, just like the current through that part. This is useful for capacitance studies in conjunction with the difference mode of the LOAD statement. The resistance of a structure cross-section, such as a diffused line, can also be calculated.

E.CRIT specifies the critical electric field used to calculate integration integrals.

ELECTRON extracts integrated electron concentration.

HOLE extracts integrated hole concentration.

IONIZINT enable the calculation of ionization integrals. Other integral ionization parameters will be ignored unless IONIZINT is specified.

LRATIO specifies the ratio between electric field lines used in ionization integral calculation. The value of this parameter should be set from 0.5 to 1.5.

METAL.CH extracts integrated charge on a contact.

N.CURRENT extracts n-current through an electrode.

N.LINES specifies the number of ionization integrals.

N.RESIST extracts n-resistance of a cross-section.

NET.CARR extracts integrated carrier concentration.

NET.CHAR extracts integrated net charge.

NLAYERS controls the distance from the contact where electric field lines start.

P.CURRENT extracts p-current through an electrode.

P.RESIST extracts p-resistance of a cross-section.

SUBSTR selects the substrate electrode for electric field lines. You don't need to specify this parameter if a substrate electrode has been defined in the ELECTRODE statement.

U.AUGER specifies that the integrated Auger recombination rate is to be extracted.

U.RADIATIVE specifies that the integrated radiative recombination rate is to be extracted.

U.SRH specifies that the integrated SRH recombination rate is to be extracted.

U.TOTAL specifies that the integrated total recombination rate is to be extracted.

Boundary Parameters

Boundary parameters: **X**.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** define a bounding box. Only nodes falling within this bounding box are included in the integration. The default bounds are the boundaries of the entire device.

CONTACT specifies the contact number. For electrode quantities (current and metal charge), a contact must be selected. Only nodes falling within the bounds and belonging to the contact are included in the integration. When IONIZINT is specified, this is the electrode used to start electric field lines.

REGIONS specifies a particular set of regions. If REGIONS is specified, only nodes within the specified bounds that are part of a particular set of regions will be integrated.

X.**MAX** specifies the x coordinate of the right edge of the bounding box.

X.**MIN** specifies the x coordinate of the left edge of the bounding box.

Y.**MAX** specifies the y coordinate of the top of the bounding box.

Y.**MIN** specifies the y coordinate of the bottom of the bounding box.

Resistance Example

This example extracts the resistance of a p-type line diffused into a lightly doped n-substrate. Since the p-conductivity of the substrate is negligible, the integration bounds can include the whole device.

```
MEASURE P.RESIST
```

Gate Charge Example

In this example, the charge on the lower surface of a gate electrode is integrated. There is 0.05 μm of gate oxide on the surface, which is located at y=0.

```
MEASURE METAL.CH CONT=1 X.MIN=-2.0 X.MAX=2.0 \
Y.MAX=-0.0499 Y.MIN=-0.0501
```

Ionization Integral Example

This example shows how to extract the maximum ionization integral in the device.

```
 MEASURE IONIZINT CONTACT=3 SUSTR=4 N.LINES=200 \
NLAYERS=15 LRATIO=1.1
```

This syntax was the original implementation of ionization integrals in early ATLAS versions. It has been superseded. See Sections 19.44: “SOLVE” and 19.34: “OUTPUT” for the recommended approach to extract ionization integrals.

19.26: MESH

MESH generates a mesh or reads a previously generated mesh.

Syntax

```
MESH <prev>|<new> [<output>]
```

Parameter	Type	Default	Units
ATHENA	Logical	False	
AUTO	Logical	False	
CONDUCTOR	Logical	False	
CYLINDRICAL	Logical	False	
DIAG.FLIP	Logical	True	
ELEC.BOT	Logical	False	
FLIP.Y	Logical	False	
IN.FILE	Character		
INFILE	Character		
MASTER.IN	Logical	True	
MASTER.OUT	Logical	True	
NX	Integer		
NY	Integer		
NZ	Integer		
OUT.FILE	Character		
OUTFILE	Character		
PISCES.IN	Logical	False	
SCALE	Integer	1	
SCALE.X	Integer	1	
SCALE.Y	Integer	1	
SCALE.Z	Integer	1	
SMOOTH.KEY	Integer		
SPACE.MULT	Real	1.0	
THREE.D	Logical	False	
TIF	Logical	False	
VERT.FLIP	Logical	False	
WIDTH	Real	1.0	µm

Description

prev is a set of parameters that allows you to read a previously generated mesh type.

new is a set of parameters that allow you to initiate the generation of a rectangular mesh.

output is a set of the parameters for saving the mesh.

Mesh File Parameters

CONDUCTOR interprets metal regions loaded in with the INFILE parameter as conductors .

CYLINDRICAL specifies that the mesh being read in contains cylindrical symmetry. Since this information is not saved in the mesh file, the CYLINDRICAL parameter must be specified each time a structure with cylindrical symmetry is loaded.

FLIP.Y reverses the sign of the y coordinate.

IN.FILE is a synonym for INFILE.

INFILE specifies the name of a previously generated mesh that has been saved to disk. The synonym for this parameter is IN.FILE.

MASTER.IN specifies a filename to read mesh and doping information in the Silvaco Standard Structure File (SSF) Format. This parameter is used to read ATHENA or DEVEDIT structure files. Typically, these files contain all REGION, ELECTRODE, and DOPING information. Although ATLAS allows you to modify the structure using these statements, this parameter is true by default and is the only file format supported by Silvaco.

PISCES.IN indicates that the mesh file is in the old PISCES-II format. This is not recommended or supported by Silvaco.

SCALE specifies a scale factor by which all x, y and z coordinates are multiplied.

SCALE.X specifies a scale factor by which all x coordinates are multiplied.

SCALE.Y specifies a scale factor by which all x and y coordinates are multiplied.

SCALE.Z specifies a scale factor by which all z coordinates are multiplied.

SPACE.MULT is a scale factor that is applied to all specified grid spacings. This parameter can be used to produce a coarse mesh and thereby reduce the simulation time.

ATHENA reads mesh and doping data generated by the ATHENA PISCES-II format file. This parameter and file format is obsolete.

Mesh Parameters

AUTO specifies that mesh lines will be generated automatically from REGION statements. See Chapter 9: “VCSEL Simulator”, “Specifying the Mesh” Section on page 9-9 for more information on how to specify mesh lines.

CYLINDRICAL specifies that the mesh contains cylindrical symmetry. Since this information is not saved in the mesh file, the CYLINDRICAL parameter must be specified each time a grid with cylindrical symmetry is loaded. Structures defined as cylindrical will be rotated 360 degrees about the y-axis. The coordinate, x=0, must be on the leftmost side of the structure.

DIAG.FLIP flips the diagonals in a square mesh about the center of the grid. If the parameter is negated, using DIAG.FLIP is specified, all diagonals will be in the same direction.

NX specifies the number of nodes in the x direction.

NY specifies the number of nodes in the y direction.

NZ specifies the number of nodes in the z direction, used in DEVICE3D only.

RECTANGULAR initiates the generation of a rectangular mesh.

THREE.D starts ATLAS3D.

TIF specifies that structure file is in TIF format.

VERT.FLIP flips the direction of mesh triangle diagonals. In other words, the diagonals of the mesh are mirrored in the vertical direction. This also works in the XY plane for 3D structures generated within ATLAS.

WIDTH specifies a scale factor to represent the un-simulated dimension for 2-D simulations. This scale factor is applied to all run time and log file outputs.

Output Parameters

OUT.FILE is a synonym for OUTFILE.

OUTFILE specifies the output filename to which the mesh is written. The synonym for this parameter is OUT.FILE.

MASTER.OUT specifies the format of the output file. This parameter is true by default so the output file will conform to the Silvaco Standard Structure File Format and can be plotted in TONYPLOT.

SMOOTH.KEY specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows:

- Triangle smoothing. All region boundaries remain fixed.
- Triangle smoothing. Only material boundaries are maintained.
- Node averaging.
- Improved triangle smoothing method. This method uses diagonal flipping to reduce the number of obtuse triangles.
- Triangle smoothing by flipping diagonals according to electric field.

Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSUPREM4. Option 4 is similar to Option 1 but Option 4 usually creates less obtuse triangles.

Mesh Definition Example

This example initiates a rectangular mesh and stores the mesh in file, MESH1.STR.

```
MESH RECTANGULAR NX=40 NY=17 OUTF=MESH1.STR
```

ATHENA Interface Example

This syntax reads in a mesh from ATHENA or DEVEDIT:

```
MESH INFILE=NMOS.STR
```

When the **auto-** interface feature is used in DECKBUILD, the program will automatically insert the MESH statement to load the result of previous programs into ATLAS.

Note: See Chapter 2: "Getting Started with ATLAS", Sections 2.6.1: "Interface From ATHENA" and 2.6.2: "Interface From DevEdit", or the on-line examples for details of the interfaces from ATHENA or DEVEDIT to ATLAS.

19.27: METHOD

METHOD sets the numerical methods to be used to solve the equations and parameters associated with the these algorithms.

Syntax

```
METHOD <gp> <mdp>
```

Parameter	Type	Default	Units
2NDORDER	Logical	True	
ACONTINU	Real	0.5	
ALT.SCHRO	Logical	False	
ATRAP	Real	0.5	
AUTONR	Logical	False	
BICGST	Logical	False	
BLOCK	Logical	False	
BLOCK.TRAN	Logical	False	
BQP.ALTEB	logical	false	
BQP.NOFERMI	logical	false	
BQPX.TOL	REAL	2.5E-7	
BQPR.TOL	REAL	1.0E-26 (2D) 1.0E-18 (3D)	
CARRIERS	Real	2	
C.ITLIMIT	Integer	500	
C.STABIL	Real	$1.0x10^{-10}$	
C.RESID	Real	$1.0x10^{-8}$	
CLIM.DD	Real	4.5×10^{13}	cm^{-3}
CLIM.EB	Real	0	cm^{-3}
CLIMIT	Real	10000	
CONT.RHS	Logical	True	
CONTINV	Logical	True	
CR.TOLER	Real	5.0×10^{-18}	
CUR.PROJ	Logical	False	
CX.TOLER	Real	1.0×10^{-5}	
DIRECT	Logical	False	

Parameter	Type	Default	Units
DG.INIT	logical	false	
DG.N.INIT	logical	false	
DG.P.INIT	logical	false	
DT.MAX	Real	•	
DT.MIN	Real	1.0×10^{-25}	s
DVLIMIT	Real	0.1	
DVMAX	Real	2.5	V
EF.TOL	Real	10^{-12}	
ELECTRONS	Logical	True	
ETR.TOLE	Real	100	
ETX.TOLE	Real		
EXTRAPOLATE	Logical	False	
FIX.QF	Logical	False	
GCARR.ITLIMIT	Real	1	
GCON.ITLIMIT	Real	0	
GMRES	Logical	False	
GUM.INIT	Integer	15	
GUMITS	Integer	100	
GUMMEL	Logical	False	
GUMMEL.NEWTON	Logical	False	
HALFIMPLICIT	Logical	False	
HCIR.TOL	Real	$5.0x10^{-11}$	
HCIX.TOL	Real	$5.0x10^{-4}$	
HOLES	Logical	True	
ICCG	Logical	False	
IR.TOL	Real	5.0×10^{-15}	
ITLIMIT	Integer	25	
IX.TOL	Real	2.0×10^{-5}	
LAS.PROJ	Logical	False	
LTR.TOLE	Real	100	
LTX.TOLE	Real		

Parameter	Type	Default	Units
LU1CRI	Real	3.0×10^{-3}	
LU2CRI	Real	3.0×10^{-2}	
L2NORM	Logical	True	
MAXTRAPS	Integer	4	
MAX.TEMP	Real	2000	K
MEINR	Logical	False	
MIN.TEMP	Real	120	K
NBLOCKIT	Integer	15	
NEG.CONC	Logical	False	
NEW.EI	Logical	False	
NEWTON	Logical	True	
NITGUMM	Integer	5	
NT0	Integer	4	
NT1	Integer	10	
NT2	Integer	25	
NT3	Integer	100	
NRCRITER	Real	0.1	
NO.POISSON	Logical	False	
PR.TOLER	Real	1.0×10^{-26}	
PX.TOLER	Real	1.0×10^{-5}	
RHSNORM	Logical	False	
RXNORM	Logical	True	
QUASI	Logical	True	
SEMIIPLICIT	Logical	False	
SINGLEPOISSON	Logical	False	
STACK	Integer	4	
TAUTO	Logical	True	
TCR.TOL	Real	100	
TCX.TOL	Real		
TLR.TOL	Real	100	
TLX.TOL	Real		

Parameter	Type	Default	Units
TMIN.FACT	Real	0.4	
TOL.LTEMP	Real	0.001	
TOL.RELAX	Real	1	
TOL.TIME	Real	5.0×10^{-3}	
TRAP	Logical	True	
TSETP.INCR	Real	2.0	
TTNMA	Real	60000.0	K
TTPMA	Real	60000.0	K
V.TOL	Real	1.0×10^{-6}	
VSATMOD.INC	Real	0.01	
XNORM	Logical	False	
WEAK	Real	200	
ZIP.BICGST	Logical	False	

Description

The METHOD statement is used to set the numerical methods for subsequent solutions. All structure and model defintions should precede the METHOD statement and all biasing conditions should follow it. Parameters in the METHOD statement are used to set the solution technique, specify options for each technique and tolerances for convergence.

Solution Method Parameters

ALT.SCHRO specifies that the alternative Schrodinger solver should be used.

BLOCK specifies that the block Newton solution method will be used as a possible solution method in subsequent solve statements until otherwise specified. The Block method only has meaning when either lattice heating or energy balance is included in the simulation. For isothermal drift diffusion simulations, BLOCK is equivalent to NEWTON.

BLOCK.TRAN specifies that the block Newton solution method will be used in subsequent transient solutions.

BICGST switches from the default ILUCGS iterative solver to the BICGST iterative solver for 3-D simulations.

DIRECT specifies that a direct linear solver should be used to solve the linear problem during 3-D simulation. By default the ILUCGS iterative solver is used for 3-D problems.

GMRES switches from the default LUCGS iterative solver to the GMRES iterative solver for 3-D simulations.

GUMMEL specifies the Gummel method will be used as a solution method in subsequent SOLVE statements until otherwise specified. If other methods (BLOCK or NEWTON) are specified in the same METHOD statement, each solution method will be applied in succession until convergence is obtained. The order that the solution methods will be applied is GUMMEL then BLOCK then NEWTON. If no solution methods are specified NEWTON is applied by default.

GUMMEL.NEWTON specifies that a NEWTON solution will always be performed after a GUMMEL solution, even if GUMMEL has converged for all equations.

HALFIMPLICIT specifies that a semi-implicit scheme will be used for transient solutions in 3-D. In most cases this method is significantly less time consuming than the default TR-BDG method.

MEINR specifies the Meinerzhagens Method, whereby carrier temperature equations will be coupled with the associated carrier continuity equation and used during GUMMEL iterations.

NEWTON specifies that Newton's method will be used as the solution method in subsequent SOLVE statements unless specified otherwise. Certain models and boundary conditions settings require that the Newton's method is used. If no solution methods are specified, NEWTON is applied by default.

NEW.EI enables an alternative eigenvalue solver that obtains eigenvectors using the reverse iteration method.

ZIP.BIGGST specifies that the ZIP library version of the BICGST interative solver will be used for 3D solution.

Note: Details on the different solution methods can be found in Chapter 18: "Numerical Techniques".

Equation Solver Parameters

CARRIERS specifies the number of carrier continuity equations that will be solved Valid values are 0, 1 and 2. CARRIERS=0 implies that only Poisson's Equation will be solved. CARRIERS=1 implies that only one carrier solution will be obtained. When this is specified, also specify either HOLES or ELECTRONS. CARRIERS=2 implies that solutions will be obtained for both electrons and holes.

ELECTRONS specifies that only electrons will be simulated for single carrier simulation.

HOLES specifies that only holes will be simulated for single carrier simulations.

Solution Tolerance Parameters

The default convergence criteria used in ATLAS consists of a combination of relative and absolute values. The program will converge if either criterion is met. This is particularly useful when low-carrier concentrations would not converge using just relative criteria.

Current convergence criteria are also used. Terminal currents are monitored at each iteration and overall convergence is allowed if currents converge along with absolute potential error.

BQPX.TOL is the convergence criterion for the update vector in the BQP model.

BQPR.TOL is the convergence criterion for the Right hand side (residual) in the BQP model.

CR.**TOLER** specifies an absolute tolerance for the continuity equation.

CX.**TOL** or **C.TOL** is the relative tolerance for the continuity equation. The XNORM parameter uses the CX.TOL and PX.TOL parameters to calculate convergence criteria.

EF.TOL specifies the Schrodinger solver tolerance when NEW.EI is specified.

ETR.TOLE is an alias for TCR.TOL.

ETX.TOLE is an alias for TCX.TOL.

HCIR.**TOL** is the absolute current convergence criteria for energy transport models.

HCIX.**TOL** is the relative current convergence criteria for energy transport models.

IR.**TOL** specifies absolute current convergence criteria.

IX.**TOL** specifies relative current convergence criteria.

LTR.TOLE is an alias for TLR.TOL.

LTX.TOLE is an alias for TLX.TOL.

PR.**TOLER** specifies an absolute tolerance for the Poisson Equation.

PX.**TOL** is the relative tolerance for the potential equation. The XNORM parameter uses the CX.TOL and PX.TOL parameters to calculate convergence criteria.

RHSNORM specifies that only absolute errors will be used to determine convergence. If RHSNORM is selected Poisson error are measured in C/μm and the continuity error is measured in A/μm.

RXNORM specifies that both relative and absolute convergence criteria will be used in the solution method. This is the equivalent of specifying both XNORM and RHSNORM. This is the default and it is not recommended to change this.

TCR.**TOL** specifies the absolute (RHSNORM) tolerance for convergence of the carrier temperature equations. The alias for this parameter is ETR.TOL.

TCX.**TOL** specifies the relative (XNORM) tolerance for convergence of the carrier temperature equations. The alias for this parameter is ETX.TOL.

TOL.**TIME** specifies maximum local truncation error for transient simulations.

TOL.**LTEMP** specifies the temperature convergence tolerance in block iterations using the lattice heat equation.

TOL.**RELAX** specifies a relaxation factor for all six Poisson, continuity, and current convergence parameters (PX.TOL, CX.TOL, PR.TOL, CR.TOL, IX.TOL, and IR.TOL).

TLR.**TOL** specifies the relative (XNORM) tolerance for convergence of the lattice temperature equation.The alias for this parameter is LTR.TOL.

TLX.**TOL** specifies the relative (XNORM) tolerance for convergence of the lattice temperature equation. The alias for this parameter is LTX.TOL.

WEAK specifies the multiplication factor for weaker convergence tolerances applied when current convergence is obtained.

XNORM specifies that only the relative errors will be used to determine convergence for the drift-diffusion equations. If XNORM is used Poisson updates are measured in kT/q. and carrier updates are measured relative to the local carrier concentration.

Note: Generally, the solution tolerances should not be changed. Convergence problems should be tackled by improving the mesh or checking the model and method combinations. Chapter 2: "Getting Started with ATLAS" has useful hints.

General Parameters

ACONTINU is an alias for ATRAP.

ATRAP specifies the multiplication factor which reduces the electrode bias steps when a solution starts to diverge. This parameter has no effect unless the TRAP parameter is specified. The alias for this parameter for ACONTINU.

CLIM.**DD** is analogous to CLIMIT except it is expressed in a dimensional value representing the minimum carrier concentration that can be resolved.

CLIM.**EB** can be treated as a regularization parameter for the case of very small carrier concentrations for energy balance simulation. It specifies the minimum value of carrier concentration for which the relaxation term in the energy balance equation will still be properly resolved. Carrier temperatures for points where the concentration is much less than CLIM.EB, will tend to the lattice temperature.

CLIMIT specifies a concentration normalization factor. See Chapter 18: “Numerical Techniques”, the “Carrier Concentrations and CLIM.DD (CLIMIT)” Section on page 18-8 for a complete description.

CONT.RHS is an alias for TRAP.

CONTINU is an alias for TRAP.

CUR.PROJ enables the use of projection method for initial guesses with current boundary conditions.

FIX.**QF** fixes the quasi-Fermi potential of each non-solved for carrier to a single value, instead of picking a value based on local bias.

ITLIMIT or **GITLIMIT** specifies the maximum number of allowed outer loops (Newton loops or Gummel continuity iterations).

LAS.PROJ extrapolates that photon rates for the initial guess during bias ramp.

MAXTRAPS specifies the number of times the trap procedure will be repeated in case of divergence. The value of MAXTRAPS may range from 1 to 10. The alias for this parameter is STACK.

MIN.**TEMP** and **MAX**.**TEMP** are specified to control the absolute range of lattice temperatures allowed during Gummel loop iterations with lattice temperature. These parameters help insure that lattice temperatures converge during the outer loop iterations.

NBLOCKIT specifies the maximum number of BLOCK iterations. If METHOD BLOCK NEWTON is specified, the solver will switch to Newton’s method after NBLOCKIT Block-Newton iterations.

NEG.CONC flag allows negative carrier concentrations.

NO.POISSON flag allows you to omit the solution of the Poisson’s Equation (see Chapter 3: “Physics”, Equation 3-1). This feature doesn’t work with Gummel’s Method (see Chapter 18: “Numerical Techniques”, Section 18.5.2: “Gummel Iteration”).

STACK is an alias for MAX.TRAPS.

TMIN.**FACT** specifies the minimum electron or hole temperature allowable during non-linear iteration updates. TMIN.FACT is normalized to 300K.

TRAP specifies that if a solution process starts to diverge, the electrode bias steps taken from the initial approximation are reduced by the multiplication factor ATRAP. The aliases for this parameter is CONT.RHS and CONTINU.

TTNMA specifies the maximum allowable electron carrier temperature.

TTPMA specifies the maximum allowable hole carrier temperature.

VSATMOD.**INC** specifies that the derivatives of the negative differential mobility (MODEL FLDMOB EVSATMOD=1) will not be included into the Jacobian until the norm of Newton update for potential is less than the value specified by VSATMOD.INC. This is useful since the negative differential mobility model is highly nonlinear and causes numerical stability problems.

Gummel Parameters

DVLIMIT limits the maximum potential update for a single loop.

GCARR.**ITLIMIT** specifies the maximum number of interations that the carrier continuity equations will solve when using GUMMEL.

GCON.ITLIMIT specifies the number of extra GUMMEL loop iterations that will be performed after all equations have converged.

GUM.**INIT** specifies the maximum number of Gummel iterations in order to obtain an initial approximation for successive Newton iterations. This parameter is used when METHOD GUMMEL NEWTON is specified

GUMITS specifies the maximum number of Gummel iterations.

LU1CRI, **LU2CRI** specifies amount of work per Poisson loop. The inner norm is required to either decrease by at least LU1CRI before returning, or reach a factor of LU2CRI below the projected Newton error, whichever is smaller. If the inner norm is exceeds the projected Newton error, the quadratic convergence is lost.

SINGLEPOISSON specifies that only a single Poisson iteration is to be performed per Gummel loop. In the default state, the continuity equation is only treated after the Poisson iteration has fully converged. This technique is useful where the carrier concentration and potential are strongly coupled but the initial guess is poor precluding the use of NEWTON.

NITGUMM, **NT0**, **NT1**, **NT2**, and **NT3** specify Gummel iteration control parameters described in Chapter 18: "Numerical Techniques", Section 18.5.10: "Detailed Convergence Criteria".

Newton Parameters

2NDORDER specifies that second-order discretization will be used when transient simulations are performed.

AUTONR implements an automated Newton-Richardson procedure, which attempts to reduce the number of LU decompositions per bias point. We strongly recommend that you use this parameter to increase the speed of NEWTON solutions.Iterations using AUTONR will appear annotated with an A in the run-time output. Often an extra iteration is added when using this parameters since the final iteration of any converged solution cannot be done using AUTONR.

DT.MAX specifies the maximum time-step for transient simulation.

DT.MIN specifies the minimum time-step for transient simulations.

DVMAX sets the maximum allowed potential update per Newton iteration. Large voltage steps are often required when simulating high voltage devices. If any simulation requires voltage steps of 10V or more, set DVMAX to 100,000. Reducing DVMAX may serve to damp oscillations in solutions in some cases leading to more robust behavior. However excessive reduction in DVMAX is not recommended since the maximum voltage step allowed will be limited by DVMAX*ITLIMIT. The alias for this parameter is N.DVLIM.

EXTRAPOLATE specifies the use of second-order extrapolation to compute initial estimates for successive time-steps for transient simulations.

L2NORM specifies the use of L2 error norms rather than infinity norms when calculating time steps for transient simulations.

NRCRITER specifies the ratio by which the norm from the previous Newton loop must decrease in order to be able to use the same Jacobian (LU decomposition) for the current Newton loop.

N.DVLIM is an alias for DVMAX.

TAUTO selects automatic, adaptive timesteps for transient simulations from local truncation error estimates. Automatic time-stepping is the default for second-order discretization, but is not allowed for first-order.

TSTEP.INCR specifies the maximum allowable ratio between the time step sizes of successive (increasing) time steps during transient simulation.

QUASI specifies a quasistatic approximation for transient simulations. This is useful in simulating transient simulations with long timescales where the device is in equilibrium at each timestep.

Quantum Parameters

BQP.NOFERMI forces the BQP model to only use its formulation derived using Maxwell Boltzmann statistics. If this is not set, then the BQP formulation used will be the same as the carrier statistics specified for the electrons or holes or both.

BQP.ALTEB enables an alternative iteration scheme that can be used when solving the BQP equation and the energy balance equation. This may produce a more robust convergence than for the default scheme. The scheme used with BQP.ALTEB set, however, will be slower.

DG.INIT enables an algorithm to initialize the Density Gradient method within the SOLVE INIT statement. This is not self-consistent with the solution of Poisson's equation, which makes it limited in usefulness. You can specify the carrier type to be initialized using either DG.N.INIT or DG.P.INIT. DG.INIT will enable both DG.N.INIT and DG.P.INIT.

Numerical Method Definition Example

The default numerical method is the equivalent of:

```
METHOD NEWTON CARRIERS=2
```

For more complex problems including those involving floating regions the following is recommended

```
METHOD GUMMEL NEWTON GUM.INIT=5
```

When impact ionization is combined with floating regions as in SOI or guard ring breakdown simualation the above syntax can also be used. Quicker solutions, however, can be obtained using the following SINGLEPOISSON technique:

```
METHOD GUMMEL NEWTON GUM.INIT=5 SINGLE
```

TRAP Parameter Example

This example illustrates the trap feature (often used to capture knees of I-V curves for junction breakdown).

The first SOLVE statement solves for the initial, zero bias case. In the second SOLVE statement, we attempt to solve for V2=3 volts and V3=5 volts. If such a large bias change caused the solution algorithms to diverge for this bias point, the bias steps would be multiplied by ATRAP (0.5).

An intermediate point (V2=1.5 volts, V3=2.5 volts) would be attempted before trying to obtain V2=3 volts and V3=5 volts again. If the intermediate point can not be solved for either case, then the program will continue to reduce the bias step (the next would be V2=0.75 volts and V3=1.25 volts) up to MAXTRAPS times.

```
METHOD TRAP ATRAP=0.5

SOLVE INIT

SOLVE V2=3 V3=5 OUTFILE=OUTA
```

Transient Method Example

In this transient simulation example, second-order discretization is used (by default), but the required LTE (10^{-3}) is smaller than the default. Since the Jacobian is exact for the second part (BDF-2) of the composite timestep, there should be very few factorizations for the BDF-2 interval when AUTONR is specified.

```
METHOD NEWTON TOL.TIME=1E-3 AUTONR
```

Note: For recommendations on METHOD parameters for different simulations, see Chapter 2: "Getting Started with ATLAS" or the on-line examples.

19.28: MOBILITY

MOBILITY allows specification of mobility model parameters.

Syntax

```
MOBILITY [NUMBER=<n>] [REGION=<n>] [MATERIAL=<name>]
         [NAME=<region_name>] <parameters>
```

Parameter	Type	Default	Units
ACCN.SF	Real	0.87	
ACCP.SF	Real	0.87	
ALBRCT.N	Logical	False	
ALBRCT.P	Logical	False	
AL1N.WATT	Real	-0.16	
AL1P.WATT	Real	-0.296	
AL2N.WATT	Real	-2.17	
AL2P.WATT	Real	-1.62	
AL3N.WATT	Real	1.07	
AL3P.WATT	Real	1.02	
ALPHAN	Real	0.0	
ALPHAN.ARORA	Real	-0.57	
ALPHAN.CAUG	Real	0.0	
ALPHAP	Real	0.0	
ALPHAP.ARORA	Real	-0.57	
ALPHAP.CAUG	Real	0.0	
ALPHAN.FLD	Real	2.4×10^{7}	cm/s
ALPHAP.FLD	Real	2.4×10^{7}	cm/s
ALPHA1N.KLA	Real	0.68	
ALPHA1P.KLA	Real	0.719	
ALPHAN.TAS	Real	2	
ALPHAP.TAS	Real	3.4	
ALPHN.CVT	Real	0.680	
ALPHP.CVT	Real	0.71	
ALN.CVT	Real	6.85×10^{-21}	

Parameter	Type	Default	Units
ALP.CVT	Real	7.82×10^{-21}	
ALPN.UM	Real	0.68	
ALPP.UM	Real	0.719	
AN.PLBRCT	Real		
AN.CCS	Real	4.61×10^{17}	cm^{-3}
AN.CVT	Real	2.58	
AN.IIS	Real	4.61×10^{17}	cm^{-3}
AP.IIS	Real	1.0×10^{17}	cm^{-3}
ANALYTIC.N	Logical	False	
ANALYTIC.P	Logical	False	
AP.CCS	Real	1.0×10^{17}	cm^{-3}
AP.CVT	Real	2.18	
ARORA.N	Logical	False	
ARORA.P	Logical	False	
ASN.YAMA	Real	1.54×10^{-5}	cm/V
ASP.YAMA	Real	5.35×10^{-5}	cm/V
B1N.TAS	Real	1.75	
B1P.TAS	Real	1.5	
B2N.TAS	Real	-0.25	
B2P.TAS	Real	-0.3	
BETAN	Real	2.0	
BETAP	Real	1.0	
BETAN.ARORA	Real	-2.33	
BETAP.ARORA	Real	-2.33	
BETAN.CAUG	Real	-2.3	
BETAP.CAUG	Real	-2.2	
BETAN.CVT	Real	2.00	
BETAP.CVT	Real	2.00	
BETAN.TAS	Real	2	
BETAP.TAS	Real	1	
BN.CCS	Real	1.52×10^{15}	cm^{-3}

Parameter	Type	Default	Units
BP.CCS	Real	6.25×10^{14}	cm^{-3}
BN.CVT	Real	4.75×10^{7}	cm/(K·s)
BN.IIS	Real	1.52×10^{15}	cm^{-3}
BN.LSM	Real	4.75×10^{7}	$cm^2/(V*s)$
BP.CVT	Real	9.925×10^{6}	cm/(K·s)
BP.IIS	Real	6.25×10^{14}	cm^{-3}
BP.LSM	Real	9.925×10^{6}	$cm^2/(V*s)$
CA.KLA	Real	0.50	
CCSMOB.N	Logical	False	
CCSMOB.P	Logical	False	
CD.KLA	Real	0.21	
CONMOB.N	Logical	False	
CONMOB.P	Logical	False	
CVT.N	Logical	False	
CVT.P	Logical	False	
CN.ARORA	Real	1.432×10^{17}	cm^{-3}
CN.CVT	Real	1.74×10^{5}	
CN.LSM	Real	1.74×10^{5}	
CN.UCHIDA	Real	0.78125	$cm^2/Vs\mu m^6$
CP.ARORA	Real	$2,67\times10^{17}$	cm^{-3}
CP.CVT	Real	8.842×10^{5}	
CP.LSM	Real	8.842×10^{5}	
CP.UCHIDA	Real	0.78125	$cm^2/Vs\mu m^6$
CRN.CVT	Real	9.68×10^{16}	cm^{-3}
CRN.LSM	Real	9.68×10^{16}	cm^{-3}
CRP.CVT	Real	2.23×10^{17}	cm^{-3}
CRP.LSM	Real	2.23×10^{17}	cm^{-3}
CSN.CVT	Real	3.43×10^{20}	cm^{-3}
CSN.LSM	Real	3.43×10^{20}	cm^{-3}

Parameter	Type	Default	Units
CSP.CVT	Real	6.10×10^{20}	cm^{-3}
CSP.LSM	Real	6.10×10^{20}	cm^{-3}
DELN.CVT	Real	5.82×10^{14}	V/s
DELP.CVT	Real	2.0546×10^{14}	V/s
DELTAN.CAUG	Real	0.73	
DELTAP.CAUG	Real	0.70	
DEVICE	Character		
DP.CVT	Real	1/3	
DN.CVT	Real	1/3	
DN.LSM	Real	3.58×10^{18}	$cm^2/(V*s)$
DN.TAS	Real	3.2×10^{-9}	
DP.LSM	Real	4.1×10^{15}	$cm^2/(V*s)$
DP.TAS	Real	2.35×10^{-9}	
EON	Real	4.0×10^{3}	V/cm
E1N.SHI	Real	6.3e3	V/cm
E2N.SHI	Real	0.77e6	V/cm
EOP	Real	4.0×10^{3}	V/cm
E1P.SHI	Real	8.0e3	V/cm
E2P.SHI	Real	3.9e5	V/cm
ECRITN	Real	4.0×10^{3}	V/cm
ECRITP	Real	4.0×10^{3}	V/cm
EGLEY.N	Logical	False	
EGLEY.P	Logical	False	
EGLEY.R	Real	2.79	
EN.CVT	Real	1.0	
EP.CVT	Real	1.0	
ESRN.TAS	Real	2.449×10^{7}	
ESRP.TAS	Real	1.0×10^{9}	
ETAN.CVT	Real	0.0767	
ETAP.CVT	Real	0.123	

Parameter	Type	Default	Units
ETAN	Real		
ETAN.WATT	Real	0.50	
ETAP.WATT	Real	0.33	
EVSATMOD	Real	0	
EX1N.SHI	Real	0.3	
EX1P.SHI	Real	2.9	
EXN1.ARO	Real	-0.57	
EXN1.LSM	Real	0.680	
EXN2.ARO	Real	-2.33	
EXN2.LSM	Real	2.0	
EXN3.ARO	Real	2.546	
EXN3.LSM	Real	2.5	
EXN4.LSM	Real	0.125	
EXN8.LSM	Real		
EXP1.ARO	Real	-0.57	
EXP1.LSM	Real	0.71	
EXP2.ARO	Real	-2.33	
EXP2.LSM	Real	2.0	
EXP3.ARO	Real	2.546	
EXP3.LSM	Real	2.2	
EXP4.LSM	Real	0.0317	
EXP8.LSM	Real		
EXP.WATT.N	Logical	False	
EXP.WATT.P	Logical	False	
F.CONMUN	Character		
F.CONMUP	Character		
F.ENMUN	Character		
F.ENMUP	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.VSATN	Character		
F.VSATP	Character		

Parameter	Type	Default	Units
FBH.KLA	Real	3.828	
FCW.KLA	Real	2.459	
FLDMOB.N	Logical	False	
FLDMOB.P	Logical	False	
FLDMOB	Integer	0	
GAMMAN	Real	4.0	
GAMMAP	Real	1.0	
GAMMAN.ARORA	Real	2.546	
GAMMAP.ARORA	Real	2.546	
GAMMAN.CAUG	Real	-3.8	
GAMMAP.CAUG	Real	-3.7	
GAMN.CVT	Real	2.5	
GAMP.CVT	Real	2.2	
GN.YAMA	Real	8.8	
GP.YAMA	Real	1.6	
HOPMOB.N	Logical	False	
HOPMOB.P	Logical	False	
HVSATMOD	Real	0	
KAPPAN.CVT	Real	1.7	
KAPPAP.CVT	Real	0.9	
KLA.N	Logical	False	
KLA.P	Logical	False	
INVN.SF	Real	0.75	
INVP.SF	Real	0.75	
MATERIAL	Character		
MMNN.UM	Real	52.2	$cm^2/(V \cdot s)$
MMNP.UM	Real	44.9	$cm^2/(V \cdot s)$
MMXN.UM	Real	1417.0	$cm^2/(V \cdot s)$
MMXP.UM	Real	470.5	$cm^2/(V \cdot s)$
MOBMOD.N	Real	1	
MOBMOD.P	Real	1	

Parameter	Type	Default	Units
MOD.WATT.N	Logical	False	
MOD.WATT.P	Logical	False	
MREF1N.WATT	Real	481.0	$cm^2/(V \cdot s)$
MREF1P.WATT	Real	92.8	$cm^2/(V \cdot s)$
MREF2N.WATT	Real	591.0	$cm^2/(V \cdot s)$
MREF2P.WATT	Real	124.0	$cm^2/(V \cdot s)$
MREF3N.WATT	Real	1270.0	$cm^2/(V \cdot s)$
MREF3P.WATT	Real	534.0	$cm^2/(V \cdot s)$
MSTI.PHOS	Logical	False	
MTN.ALPHA	Real	0.68	
MTN.BETA	Real	2.0	
MTN.CR	Real	9.68×10^{16}	cm^{-3}
MTN.CS	Real	3.34×10^{20}	cm^{-3}
MTN.MAX	Real	1417	$cm^2/(Vs)$
MTN.MIN1	Real	52.2	$cm^2/(Vs)$
MTN.MIN2	Real	52.2	$cm^2/(Vs)$
MTN.MU1	Real	43.4	$cm^2/(Vs)$
MTN.PC	Real	0.0	cm^{-3}
MTP.ALPHA	Real	0.719	
MTP.BETA	Real	2.0	
MTP.CR	Real	2.23×10^{17}	cm^{-3}
MTP.CS	Real	6.1×10^{20}	cm^{-3}
MTP.MAX	Real	470.5	$cm^2/(Vs)$
MTP.MIN1	Real	44.9	$cm^2/(Vs)$
MTP.MIN2	Real	0.0	$cm^2/(Vs)$
MTP.MU1	Real	29.0	$cm^2/(Vs)$
MTP.PC	Real	9.23×10^{16}	cm^{-3}
MU0N.SHI	Real	1430.0	$cm^2/(V \cdot s)$
MU0P.SHI	Real	500.0	$cm^2/(V \cdot s)$

Parameter	Type	Default	Units
MU1N.ARORA	Real	88.0	$cm^2/(V \cdot s)$
MU1P.ARORA	Real	54.3	$cm^2/(V \cdot s)$
MU2N.ARORA	Real	1252.0	$cm^2/(V \cdot s)$
MU2P.ARORA	Real	407.0	$cm^2/(V \cdot s)$
MU1N.CAUG	Real	55.24	$cm^2/(V \cdot s)$
MU1P.CAUG	Real	49.7	$cm^2/(V \cdot s)$
MU2N.CAUG	Real	1429.23	$cm^2/(V \cdot s)$
MU2P.CAUG	Real	479.37	$cm^2/(V \cdot s)$
MU0N.CVT	Real	52.2	$cm^2/(V \cdot s)$
MU0P.CVT	Real	44.9	$cm^2/(V \cdot s)$
MU1N.CVT	Real	43.4	$cm^2/(V \cdot s)$
MU1P.CVT	Real	29.0	$cm^2/(V \cdot s)$
MUBN.TAS	Real	1150	
MUBP.TAS	Real	270	
MUDEG.N	Real	1.0	
MUDEG.P	Real	1.0	
MULN.YAMA	Real	1400.0	$cm^2/(V \cdot s)$
MULP.YAMA	Real	480.0	$cm^2/(V \cdot s)$
MUMAXN.CVT	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.CVT	Real	470.5	$cm^2/(V \cdot s)$
MUMAXN.KLA	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.KLA	Real	470.5.0	$cm^2/(V \cdot s)$
MUMINN.KLA	Real	52.2	$cm^2/(V \cdot s)$
MUMINP.KLA	Real	44.9	$cm^2/(V \cdot s)$
MUN	Real	1.0	$cm^2/(V \cdot s)$
MUN.MAX	Real	1429.23	$cm^2/(V \cdot s)$
MUN.MIN	Real	55.24	$cm^2/(V \cdot s)$
MUN0.LSM	Real	52.2	$cm^2/(V \cdot s)$

Parameter	Type	Default	Units
MUN1.ARO	Real	88.0	$cm^2/(V \cdot s)$
MUN2.ARO	Real	1252.0	$cm^2/(V \cdot s)$
MUN1.LSM	Real	43.4	$cm^2/(V \cdot s)$
MUN2.LSM	Real	1417.0	$cm^2/(V \cdot s)$
MUP	Real	1.0	$cm^2/(V \cdot s)$
MUP.MAX	Real	479.37	$cm^2/(V \cdot s)$
MUP.MIN	Real	49.7	$cm^2/(V \cdot s)$
MUP0.LSM	Real	44.9	$cm^2/(V \cdot s)$
MUP1.ARO	Real	54.3	$cm^2/(V \cdot s)$
MUP1.LSM	Real	29.0	$cm^2/(V \cdot s)$
MUP2.ARO	Real	407.0	$cm^2/(V \cdot s)$
MUP2.LSM	Real	470.5	$cm^2/(V \cdot s)$
N.ANGLE	Real	0.0	Degrees
N.MASETTI	Logical	False	
N2N.TAS	Real	1.1×10^{21}	
N2P.TAS	Real	1.4×10^{18}	
N1N.TAS	Real	2.0×10^{19}	
N1P.TAS	Real	8.4×10^{16}	
NAME	Character		
NCRITN.ARORA	Real	1.432×10^{17}	cm^{-3}
NCRITP.ARORA	Real	2.67×10^{17}	cm^{-3}
NCRITN.CAUG	Real	1.072×10^{17}	cm^{-3}
NCRITP.CAUG	Real	1.606×10^{17}	cm^{-3}
NEWCVT.N	Logical	False	
NEWCVT.P	Logical	False	
NREF1N.KLA	Real	9.68×16	cm^{-3}
NREF1P.KLA	Real	2.23×17	cm^{-3}
NREFD.KLA	Real	4.0×20	cm^{-3}
NREFA.KLA	Real	7.2×20	cm^{-3}

Parameter	Type	Default	Units
NREFN	Real	1.072×10^{17}	cm^{-3}
NREFN.YAMA	Real	3.0×10^{16}	cm^{-3}
NREFP	Real	1.606×10^{17}	cm^{-3}
NREFP.YAMA	Real	4.0×10^{16}	cm^{-3}
NRFN.UM	Real	9.68×10^{16}	cm^{-3}
NRFP.UM	Real	2.23×10^{17}	cm^{-3}
NUN	Real	-2.3	
NUP	Real	-2.2	
OLDSURF.N	Logical	False	
OLDSURF.P	Logical	False	
OXLEFTN	Real		microns
OXLEFTP	Real		microns
OXRIGHTN	Real		microns
OXRIGHTP	Real		microns
OXBOTTOMN	Real		microns
OXBOTTOMP	Real		microns
P.MASETTI	Logical	False	
P1N.SHI	Real	0.28	
P1P.SHI	Real	0.3	
P2N.SHI	Real	2.9	
P2P.SHI	Real	1.0	
P1N.TAS	Real	0.09	
P1P.TAS	Real	0.334	
P2N.TAS	Real	4.53×10^{-8}	
P2P.TAS	Real	3.14×10^{-7}	
P.ANGLE	Real	0.0	Degrees
PC.LSM	Real	9.23×10^{16}	cm^{-3}
PCN.CVT	Real	0.0	cm^{-3}
PCP.CVT	Real	0.23×10^{16}	cm^{-3}
PFMOB.N	Logical	False	

Parameter	Type	Default	Units
PFMOB.P	Logical	False	
PRINT	Logical	False	
R1.KLA	Real	0.7643	
R2.KLA	Real	2.2999	
R3.KLA	Real	6.5502	
R4.KLA	Real	2.3670	
R5.KLA	Real	-0.8552	
R6.KLA	Real	0.6478	
RN.TAS	Real	2	
RP.TAS	Real	3	
REGION	Integer		
S1.KLA	Real	0.89233	
S2.KLA	Real	0.41372	
S3.KLA	Real	0.19778	
S4.KLA	Real	0.28227	
S5.KLA	Real	0.005978	
S6.KLA	Real	1.80618	
S7.KLA	Real	0.72169	
SCHWARZ.N	Logical	False	
SCHWARZ.P	Logical	False	
SHI.N	Logical	False	
SHI.P	Logical	False	
SN.YAMA	Real	350	
SP.YAMA	Real	81.0	
STRUCTURE	Character		
SURFMOB.N	Logical	False	
SURFMOB.P	Logical	False	
TASCH.N	Logical	False	
TASCH.P	Logical	False	
TAUN.CVT	Real	0.125	
TAUP.CVT	Real	0.0317	
TETN.UM	Real	-2.285	

Parameter	Type	Default	Units
TETP.UM	Real	-2.247	
THETAN.FLD	Real	0.8	
THETAP.FLD	Real	0.8	
THETAN.KLA	Real	2.285	
THETAN.SHI	Real	2.285	
THETAP.KLA	Real	2.247	
THETAP.SHI	Real	2.247	
TMUBN.TAS	Real	2.5	
TMUBP.TAS	Real	1.4	
TMUN	Real	1.0	
TMUP	Real	1.0	
TNOMN.FLD	Real	600.0	K
TNOMP.FLD	Real	600.0	K
TN.UCHIDA	Real	0.0	µm
TP.UCHIDA	Real	0.0	µm
ULN.YAMA	Real	4.9×10^{6}	cm/s
ULP.YAMA	Real	2.928×10^{6}	cm/s
VSATN	Real		cm/s
VSATP	Real		cm/s
VSN.YAMA	Real	1.036×10^{7}	cm/s
VSP.YAMA	Real	1.200×10^{7}	cm/s
UCHIDA.N	Logical	False	
UCHIDA.P	Logical	False	
XIN	Real	-3.8	
XIP	Real	-3.7	
XMINN.WATT	Real	-1.0×32	microns
XMAXN.WATT	Real	1.0×32	microns
YMAXN.WATT	Real	-1.0x32	microns
XMINP.WATT	Real	-1.0×32	microns
XMAXP.WATT	Real	1.0×32	microns
YMAXP.WATT	Real	-1.0×32	microns

Parameter	Type	Default	Units
YCHARN.WATT	Real	1.0×32	microns
YCHARP.WATT	Real	1.0×32	microns
Z11N.TAS	Real	0.0388	
Z11P.TAS	Real	0.039	
Z22N.TAS	Real	1.73×10^{-5}	
Z22P.TAS	Real	1.51×10^{-5}	

Description

MATERIAL specifies which material from Table B-1 in Appendix B: "Material Systems" should be applied to the MOBILITY statement. If a material is specified, then all regions defined as being composed of that material will be affected.

Note: You can specify the following logical parameters to indicate the material instead of assigning the MATERIAL parameter: SILICON, GAAS, POLYSILI, GERMAINU, SIC, SEMICOND, SIGE, ALGAAS, A-SILICO, DIAMOND, HGCDTE, INAS, INGAAS, INP, S.OXIDE, ZNSE, ZNTE, ALINAS, GAASP, INGAP and MINASP.

DEVICE specifies the device in MIXEDMODE simulation (see Chapter 12: "MixedMode: Mixed Circuit and Device Simulator") to be applied to the MOBILITY statement. The synonym for this parameter is STRUCTURE.

NAME specifies the name of the region to be applied to the MOBILITY statement. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.

REGION specifies the number of the region to be applied to the MOBILITY statement.

PRINT acts the same as PRINT on the MODELS statement.

STRUCTURE is a synonym for DEVICE.

MASETTI Model Parameters

Mobility Model Flags

ALBRCT.N and **ALBRECT.P** enable the Albrecht mobility model (see Chapter 5: "Blaze: Compound Material 2D Simulator", "The Albrecht Model" Section on page 5-36).

ANALYTIC.**N** specifies that the analytic concentration dependent model is to be used for electrons (see Chapter 3: "Physics", Equation 3-161).

ANALYTIC.**P** specifies that the analytic concentration dependent model is to be used for holes (see Chapter 3: "Physics", Equation 3-162).

ARORA.**N** specifies that the Arora analytic concentration dependent model is to be used for electrons (see Chapter 3: "Physics", Equation 3-163).

ARORA.**P** specifies that the Arora analytic concentration dependent model is to be used for electrons (see Chapter 3: "Physics", Equation 3-164).

CONMOB.**N** specifies that doping concentration dependent model to be used for electrons.

CONMOB.**P** specifies that a doping concentration dependent model is to be used for holes.

CCSMOB.N specifies that carrier-carrier scattering model is to be used for electrons (see Chapter 3: "Physics", Equations 3-167-3-171).

CCSMOB.P specifies that carrier-carrier scattering model is to be used for holes (see Chapter 3: "Physics", Equations 3-167-3-171).

EGLEY.N enables the low field strained silicon mobility model for electrons (see Section 3.6.12).

EGLEY.P enables the low field strained silicon mobility model for holes (see Section 3.6.12).

EVSATMOD specifies which parallel field dependent mobility model (see Chapter 3: "Physics", Equation 3-260 and Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-50) should be used for electrons as follows:

1. Use the standard saturation model (`EVSATMOD=O`, see also Chapter 3: "Physics", Equation 3-260).
2. Use the negative differential mobility saturation model (`EVSATMOD=1`, see also Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-50).
3. Use simple field-dependent velocity mode (`EVSATMOD=2`, see also Chapter 3: "Physics", Equation 3-274).

EXP.WATT.N turns on exponertial modification to Watt's mobility model for electrons (see Chapter 3: "Physics", Equation 3-256).

EXP.WATT.P turns on exponential modification to Watt's mobility model for holes (see Chapter 3: "Physics", Equation 3-257).

FLDMOB specifies transverse field degradation for electrons as follows:

1. No transverse degradation.
2. Use the Watt or Tasch transverse field models depending on the settings of `FIELDMOB1` and `FIELDMOB2`.
3. Use the Yamaguchi transverse field dependent model.
4. Use the CVT transverse field dependent model.

FLDMOB.N specifies a lateral electric field dependent model for electrons (see Chapter 3: "Physics", Equation 3-252).

FLDMOB.P specifies a lateral electric field dependent model for holes (see Chapter 3: "Physics", Equation 3-253).

KLA.N turns on Klaassen's mobility model for electrons (see Chapter 3: "Physics", Equations 3-179 through 3-204).

KLA.P turns on Klaassen's mobility model for holes (see Chapter 3: "Physics", Equations 3-179 through 3-204).

MOBMOD.N specifies transverse field degradation for electrons as follows:

1. No transverse degradation.
2. Use the Watt or Tasch transverse field models depending on the settings of `FIELDMOB1` and `FIELDMOB2`.
3. Use the Yamaguchi transverse field dependent model.
4. Use the CVT transverse field dependent model.

MOBMOD.P specifies transverse field degradation for holes as follows:

1. No transverse degradation.
2. Use the Watt or Tasch transverse field models depending on the settings of `FIELDMOB1` and `FIELDMOB2`.
3. Use the Yamaguchi transverse field dependent model.
4. Use the CVT transverse field dependent model.

MOD.**WATT**.**N** turns on modified Watt mobility model for electrons (see Chapter 3: “Physics”, Equation 3-256).

MOD.**WATT**.**P** turns on modified Watt Mobility Model for holes (see Chapter 3: “Physics”, Equation 3-257).

MSTI.PHOS selects the parameter set for Phosphorous doping in Masetti model. For more information about this parameter, see Chapter 3: “Physics”, Tables 3-23, 3-24 and 3-25.

N.ANGLE specifies angle for application of electron mobility parameters in simulation of anisotropic mobility.

N.MASETTI enables Masetti mobility model for electrons. For more information about this parameter, see Chapter 3: “Physics”, Tables 3-23, 3-24 and 3-25.

NEWCVT.N specifies that a new surface mobility degradation calculation is used in the CVT mobility model for electrons in place of the original (see Chapter 3: “Physics”, the “Darwish CVT Model” Section on page 3-61).

NEWCVT.P specifies that a new surface mobility degradation calculation is used in the CVT mobility model for holes in place of the original (see Chapter 3: “Physics”, the “Darwish CVT Model” Section on page 3-61).

P.ANGLE specifies angle for application of hole mobility parameters in simulation of anisotropic mobility.

P.MASETTI enables Masetti mobility model for holes. For more information about this parameter, see Chapter 3: “Physics”, Tables 3-23, 3-24 and 3-25.

SURFMOB.**N** invokes the effective field based surface mobility model for electrons (see Equation 3-252).

SURFMOB.**P** invokes the effective field based surface mobility model for holes (see Equation 3-253).

SCHWARZ.N specifies the use of transverse electric field-dependent mobility models for electrons. See `TFLDMB1` or `SCHWARZ` in the `MODELS` statement.

SCHWARZ.P specifies the use of transverse electric field-dependent mobility models for holes. See `TFLDMB1` or `SCHWARZ` in the `MODELS` statement.

SHI.**N** turns on Shirahata’s Mobility Model for electrons (see Chapter 3: “Physics”, Equation 3-258).

SHI.**P** turns on Shirahata’s Mobility Model for holes (see Chapter 3: “Physics”, Equation 3-259).

TASCH.N specifies a transverse electric field dependent mobility model for electrons based on Tasch [146,147] (see Chapter 3: “Physics”, Equations 3-233 through 3-251).

TASCH.P specifies a transverse electric field dependent mobility model for holes based on Tasch [146,147] (see Chapter 3: “Physics”, Equations 3-233 through 3-251).

Temperature Dependent Low Field Mobility Parameters

For information about these parameters, see Chapter 3: “Physics”, Table 3-19.

Albrecht Model Parameters

For information about these parameters, see Chapter 5: “Blaze: Compound Material 2D Simulator”, “The Albrecht Model” Section on page 5-36.

Caughey-Thomas Concentration Dependent Model Parameters

For more information about these parameters, see Chapter 3: “Physics”, Table 3-21.

Arora Concentration Dependent Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-22.

Carrier-Carrier Scattering Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-26.

Parallel Field Dependent Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-46 and Chapter 5: "Blaze: Compound Material 2D Simulator", Tables 5-3 and 5-10.

Yamaguchi Transverse Field Dependent Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-41.

CVT Transverse Field Dependent Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-37.

Darwish CVT Transverse Field Dependent Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-38.

Watt Effective Transverse Field Depenedent Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-41.

Tasch Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-42.

Klaassen's Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Tables 3-30 and 3-35.

The Modified Watt Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-44.

Shirahata's Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-45.

Uchida's Mobility Model Parameters

For more information about these parameters, see Chapter 3: "Physics", Table 3-36.

SCHWARZ and TASCH Transverse Field Dependent Model Parameters

ACCN.SF specifies the accumulation saturation factor which describes the ratio of the electron concentration in the accumulation layer before and after bending of conductivity and valence bands for electron mobility.

ACCN.SF specifies the accumulation saturation factor which describes the ratio of the hole concentration in the accumulation layer before and after bending of conductivity and valence bands for hole mobility.

INVN.SF specifies the inversion saturation factor which describes the ratio of the electron concentration in the inversion layer before and after the bending of conductivity and valence bands for electron mobility.

INVN.SF specifies the inversion saturation factor which describes the ratio of the hole concentration in the inversion layer before and after the bending of conductivity and valence bands for hole mobility.

OXBOTTOMN specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for electron mobility.

OXBOTTOMP specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for hole mobility.

OXLEFTN specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for electron mobility.

OXLEFTP specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for hole mobility.

OXRIGHTN specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for electron mobility.

OXRIGHTP specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for hole mobility.

Miscellaneous Mobility Model Parameters

EGLEY.R specifies the ratio of unstrained light hole mobility model to the unstrained net mobility.

Interpreter Functions

F.CONMUN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent electron mobility model.

F.CONMUP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent hole mobility model.

F.ENMUN specifies a C-Interpreter file for the electron mobility as a function of Electron Temperature and perpendicular field as well as other choice variables. The function itself is named `endepmun` and only applies to ATLAS2D.

F.ENMUP specifies a C-Interpreter file for the hole mobility as a function of Hole Temperature and perpendicular field as well as other choice variables. The function itself is named `endepmup` and only applies to ATLAS2D.

F.MUNSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent electron mobility model for velocity saturation.

F.MUPSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent hole mobility model for velocity saturation.

F.VSATN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent electron saturation velocity models.

F.VSATP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent electron saturation velocity models.

Mobility Model Aliases

Aliases are available for certain parameters on the `MOBILITY` statement. Table 19-2 lists these aliases.

Table 19-2. Mobility Model Aliases			
Parameter	**Alias**	**Parameter**	**Alias**
MU1N.CAUG	MUN.MIN	THETAN.KLA	TETN.UM*
MU1P.CAUG	MUP.MIN	THETAP.KLA	TETP.UM*
MU2N.CAUG	MUN.MAX	MUMAXN.KLA	MMXN.UM
MU2P.CAUG	MUP.MAX	MUMAXP.KLA	MMXP.UM
BETAN.CAUG	NUN	MUMINN.KLA	MMNN.UM
BETAP.CAUG	NUP	MUMINP.KLA	MMNP.UM
DELTAN.CAUG	ALPHAN	NREF1N.KLA	NRFN.UM
DELTAP.CAUG	ALPHAP	NREF1P.KLA	NRFP.UM
GAMMAN.CAUG	XIN	ALPHA1N.KLA	ALPN.UM
GAMMAP.CAUG	XIP	ALPHA1P.KLA	ALPP.UM
NCRITN.CAUG	NREFN	BN.CVT	BN.LSM
NCRITO.CAUG	NREFP	BP.CVT	BP.LSM
MU1N.ARORA	MUN1.ARO	CN.CVT	CN.LSM
MU1P.ARORA	MUP1.ARO	CP.CVT	CP.LSM
MU2N.ARORA	MUN2.ARO	TAUN.CVT	EXN4.LSM
MU2P.ARORA	MUP2.ARO	TAUP.CVT	EXP4.LSM
ALPHAN.ARORA	EXN1.ARO	MU0N.CVT	MUN0.LSM
ALPHAP.ARORA	EXP1.ARO	MU0P.CVT	MUP0.LSM
BETAN.ARORA	EXN2.ARO	MU1N.CVT	MUN1.LSM
BETAP.ARORA	EXP2.ARO	MU1P.CVT	MUP1.LSM
GAMMAN.ARORA	EXN3.ARO	CRN.CVT	CRN.LSM
GAMMAP.ARORA	EXP3.ARO	CRP.CVT	CRP.LSM
NCRITN.ARORA	CN.ARORA	CSN.CVT	CSN.LSM
NCRITP.ARORA	CP.ARORA	CSP.CVT	CSP.LSM
AN.CCS	AN.IIS	ALPHAN.CVT	EXN1.LSM
AP.CCS	AP.IIS	ALPHAP.CVT	EXP1.LSM
BN.CCS	BN.IIS	BETAN.CVT	EXN2.LSM
BP.CCS	BP.IIS	BETAP.CVT	EXP2.LSM
P1N.SHI	EX1N.SHI	MUMAXN.CVT	MUN2.LSM
P1P.SHI	EX1P.SHI	MUMAXP.CVT	MUP2.LSM

Table 19-2. Mobility Model Aliases			
Parameter	**Alias**	**Parameter**	**Alias**
ECRITN	EON	GAMN.CVT	EXN3.LSM
ECRITP	EOP	GAMP.CVT	EXP3.LSM
KN.CVT	EXN8.LSM	DELN.CVT	DN.LSM
KP.CVT	EXP8.LSM	DELP.CVT	DP.LSM
PCP.CVT	PC.LSM		

Note: These aliases are negated with respect to the primary parameter.

Modified Watt Model Example

The following example sets the Modified Watt Surface mobility model for MOSFETs. The MOBILITY statement is used to set the models and to specify the value of the depth of action of the Modified Watt model.

```
MODELS CONMOB FLDMOB SRH  MIN.SURF  PRINT
MOBILITY WATT.N MOD.WATT.N YMAXN.WATT=0.01
```

19.29: MODELS

MODELS specifies model flags to indicate the inclusion of various physical mechanisms, models, and other parameters such as the global temperature for the simulation.

Syntax

```
MODELS <mf> <gp> <mdp>
```

Parameter	Type	Default	Units
2DXY.SCHRO	Logical	False	
A.TEMP	Logical	False	
ACC.SF	Real	0.87	
ALBRCT	Logical	False	
ALL	Logical	True	
ALN1	Real	-0.160	
ALN2	Real	-2.17	
ALN3	Real	1.07	
ALP1	Real	-0.296	
ALP2	Real	-1.62	
ALP3	Real	1.02	
ALT.SCHRO	Logical	False	
ANALYTIC	Logical	False	
AR.MU1N	Real	88	
AR.MU2N	Real	1252	
AR.MU1P	Real	54.3	
AR.MU2P	Real	407	
ARORA	Logical	False	
ASUB	Real	1.0	Å
AUGER	Logical	False	
AUGGEN	Logical	False	
AUTOBBT	Logical	False	
B.DORT	Real	1.0	
B.ELECTRONS	Real	2	
B.HOLES	Real	1	
BARLN	Real	2.59×10^{-4}	$(V\cdot cm)^{0.5}$

Parameter	Type	Default	Units
BARLN	Real	2.59×10^{-4}	$(V\cdot cm)^{0.5}$
BB.A	Real	4.0×10^{14}	$cm^{-1/2}V^{-5/2}s^{-1}$
BB.B	Real	1.97×10^{7}	$cm^{-1/2}V^{-2}s^{-1}$
BB.GAMMA	Real	2.5	V/cm
BBT.FORWARD	Logical	False	
BBT.HURKX	Logical	False	
BBT.KL	Logical	False	
BBT.NLDERIVS	Logical	False	
BBT.NONLOCAL	Logical	False	
BBT.STD	Logical	False	
BBT3	Logical	False	
BC.QUANT	Logical	False	
BGN	Logical	False	
BGN2	Logical	False	
BGN.KLA	Logical	False	
BGN.KLASSEN	Logical	False	
BGN.SLOTBOOM	Logical	False	
BH.FNONOS	Real	3.07	eV
BIPOLAR	Logical	False	
BIPOLAR2	Logical	False	
BOLTZMANN	Logical	True	
BOUND.TRAP	Logical	False	
BQP.N	Logical	False	
BQP.P	Logical	False	
BQP.NALPHA	Real	0.5	
BQP.NGAMMA	Real	1.2	
BQP.NEUMANN	Logical	true	
BQP.PALPHA	Real	0.5	
BQP.PGAMMA	Real	1.0	
BQP.QDIR	Integer	2	
BTBT	Logical	False	

Parameter	Type	Default	Units
BX	Real	0.0	Tesla
BY	Real	0.0	Tesla
BZ	Real	0.0	Tesla
C0	Real	2.5×10^{-10}	
CALC.FERMI	Logical	False	
CALC.STRAIN	Logical	False	
CARRIERS	Real		
CAVITY.LENGTH	Real		μm
CCS.EA	Real	4.61×10^{17}	
CCS.EB	Real	1.52×10^{15}	
CCS.HA	Real	1.0×10^{17}	
CCS.HB	Real	6.25×10^{14}	
CCSMOB	Logical	False	
CCSNEW	Logical	False	
CCSSURF	Logical	False	
CGATE.N	Real	0.025	
CGATE.P	Real	150.0	
CHI.HOLES	Real	4.6×10^{4}	
CHIA	Real	3e5	
CHIB	Real	5.0×10^{4}	
CHUANG	Logical	False	
CONMOB	Logical	False	
CONSRH	Logical	False	
CVT	Logical	False	
D.DORT	Real	2.5×10^{-6}	
DG.N	Logical	False	
DG.P	Logical	False	
DGN.GAMMA	Real	3.6	
DGP.GAMMA	Real	3.6	
DEVDEG.E	Logical	False	
DEVDEG.H	Logical	False	

Parameter	Type	Default	Units
DEVDEG.B	Logical	False	
DEVICE	Character		
DGLOG	Logical	True	
DGROOT	Logical	False	
DRIFT.DIFF	Logical	True	
DT.CBET	Logical	False	
DT.CUR	Logical	False	
DT.METH	Real	1	
DT.VBET	Logical	False	
DT.VBHT	Logical	False	
E.BENDING	Logical	False	
E.TAUR.VAR	Logical	False	
E0_TCOEF	Real		
E0.TCOEF	Real		
ECRIT	Real	4000.0	V/cm
EF.TOL	Real	10×10^{-10}	
EIGENS	Integer	15	
ELECTRONS	Logical	True	
ENERGY.STEP	Real	0.0025	eV
ERASE	Logical	False	
ET.MODEL	Logical	False	
ETA.FNONOS	Real	1.0	
ETAN	Real	0.50	
ETAP	Real	0.33	
EXCITONS	Logical	False	
EVSATMOD	Integer	0	
FERMIDIRAC	Logical	False	
FERRO	Logical	False	
FIXED.FERMI	Logical	False	
FLDMOB	Logical	False	
F.AE	Real	1.82×10^{-7}	$CV^{-2}s^{-1}$

Parameter	Type	Default	Units
F.BE	Real	1.90×10^{8}	V/cm
F.AH	Real	1.82×10^{-7}	$CV^{-2}s^{-1}$
F.BH	Real	1.90×10^{8}	V/cm
F.EFEMIS	Character		
F.HFEMIS	Character		
F.KSN	Character		
F.KSP	Character		
F.TRAP.COULOMBIC	Character		
F.TRAP.TUNNEL	Character		
FAST.EIG	Logical	False	
FG.RATIO	Real		
FN.CUR	Logical	False	
FNHOLES	Logical	False	
FNHPP	Logical	False	
FNONOS	Logical	False	
FNORD	Logical	False	
FNPP	Logical	False	
G.EMPIRICAL	Logical	False	
G.STANDARD	Logical	False	
GAINMOD	Logical	0	
GATE1	Logical	False	
GATE2	Logical	False	
GR.HEAT	Logical	False	
HANSCHQM	Logical	False	
H.BENDING	Logical	False	
H.TAUR.VAR	Logical	False	
HCTE	Logical	False	
HCTE.EL	Logical	False	
HCTE.HO	Logical	False	
HEAT.FULL	Logical	False	
HEAT.PETHOM	Logical	False	

Parameter	Type	Default	Units
HEI	Logical	False	
HHI	Logical	False	
HOLE	Logical	False	
HOPMOB	Logical	False	
HVSATMOD	Integer	0	
IG.ELINR	Real	6.16×10^{-6}	cm
IG.HLINR	Real	6.16×10^{-6}	cm
IG.ELINF	Real	9.2×10^{-7}	cm
IG.HLINF	Real	9.2×10^{-7}	cm
IG.EBETA	Real	2.59×10^{-4}	$(V.cm)^{0.5}$
IG.HBETA	Real	2.59×10^{-4}	$(V.cm)^{0.5}$
IG.EETA	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
IG.HETA	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
IG.EEOX	Real	9.0×10^{4}	V/cm
IG.HEOX	Real	9.0×10^{4}	V/cm
IG.EB0	Real	3.2	V
IG.HB0	Real	4.0	V
IG.LRELE	Real	3.0×10^{-6}	[Q=1] cm
IG.LRELH	Real	2.0×10^{-6}	[Q=1] cm
II.NODE	Logical	False	
II.VALDI	Logical	False	
IMPACT	Logical	False	
INC.ION	Logical	False	
INCOMPLETE	Logical	False	
INFINITY	Real	0.001	
IONIZ	Logical	False	
INV.SF	Real	0.75	
ISHIKAWA	Logical	False	
JOULE.HEAT	Logical	True	
K.P	Logical	False	

Parameter	Type	Default	Units
KAUGCN	Real		
KAUGCP	Real		
KAUGDN	Real		
KAUGDP	Real		
KSRHCN	Real		
KSRHCP	Real		
KSRHGN	Real		
KSRHGP	Real		
KSRHTN	Real		
KSRHTP	Real		
KSN	Integer	-1	
KSP	Integer	-1	
KLA	Logical	False	
KLAAUG	Logical	False	
KLASRH	Logical	False	
L.TEMP	Logical	False	
LAMHN	Real	9.2×10^{-7}	cm
LAMHP	Real	9.2×10^{-7}	cm
LAMRN	Real	6.16×10^{-6}	cm
LAMRP	Real	6.16×10^{-6}	cm
LAS.ABSOR_SAT	Logical	False	
LAS.ABSORPTION	Logical	False	
LAS.EFINAL	Real		eV
LAS.EINIT	Real		eV
LAS.ESEP	Real		eV
LAS.ETRANS	Logical	False	
LAS.FCARRIER	Logical	False	
LAS.GAIN_SAT	Logical	False	
LAS.ITMAX	Integer	30	
LAS.LOSSES	Real	0	cm^{-1}
LAS.MIRROR	Real	90	%

Parameter	Type	Default	Units
LAS.MULTISAVE	Logical	False	
LAS.NEFF	Real	3.57	
LAS.OMEGA	Real	2.16×10^{15}	1/sec
LAS.NMODE	Integer	1	
LAS.NX	Real		
LAS.NY	Real		
LAS.REFLECT	Logical	False	
LAS.RF	Real		
LAS.RR	Real		
LAS.SIN	Real	1.0×10^{4}	cm^{-1}
LAS.SPECSAVE	Integer	1	
LAS.SPONTANEOUS	Real		
LAS.TAUSS	Real	0.05	
LAS.TIMERATE	Logical	True	
LAS.TOLER	Real	0.01	
LAS.TRANS_ENERGY	Real		
LAS.XMAX	Real		
LAS.XMIN	Real		
LAS.YMAX	Real		
LAS.YMIN	Real		
LAS_MAXCH	Real	2.5	
LASER	Logical	False	
LAT.TEMP	Logical	False	
LED	Logical	False	
LI	Logical	False	
LMODE	Logical	False	
LORENTZ	Logical	False	
LSMMOB	Logical	False	
MASS.TUNNEL	Real	0.25	
MASETTI	Logical	False	
MATERIAL	Character		
MIN.SURF	Logical	False	

Parameter	Type	Default	Units
MOBMOD	Integer		
MOBTEM.SIMPL	Logical	False	
MOS	Logical	False	
MOS2	Logical	False	
MREFN1	Real	481.0	cm^2/Vs
MREFN2	Real	591.0	cm^2/Vs
MREFN3	Real	1270.0	cm^2/Vs
MREFP1	Real	92.8	cm^2/Vs
MREFP2	Real	124.	cm^2/Vs
MREFP3	Real	534.	cm^2/Vs
N.CONCANNON	Logical	False	
N.DORT	Logical	False	
N.KSI.VAR	Logical	False	
N.KSI_VAR	Logical	False	
N.QUANTUM	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
NEARFLG	Logical	False	
NEW.EIG	Logical	False	
NEW.SCHRO	Logical	False	
NEW.DORT	Logical	False	
NEWIMPACT	Logical	False	
NI.FERMI	Logical	False	
NT.FNONOS	Real	0.0	μm
NUM.BAND	Real		
NUM.DIRECT	Real		
NUMBER	Integer	0	
NUMCARR	Real		
OLDIMPACT	Logical	False	
OPTR	Logical	False	
ORTH.SCHRO	Real	10^{-3}	

Parameter	Type	Default	Units
OX.BOTTOM	Real	0.0	μm
OX.LEFT	Real		μm
OX.MARGIN	Real	0.0003	μm
OX.POISSON	Logical	False	
OX.RIGHT	Real		μm
OX.SCHRO	Logical	False	
P.CONCANNON	Logical	False	
P.DORT	Logical	False	
P.KSI_VAR	Logical	False	
P.KSI.VAR	Logical	False	
P.SCHRODING	Logical	False	
P.TEMP	Logical	False	
PASSLER	Logical	False	
PATH.N	Real	3.4×10^{7}	microns
PATH.P	Real	2.38×10^{-7}	microns
PEFF.N	Real	3.2	eV
PEFF.P	Real	4.8	eV
PFMOB	Logical	False	
PHOTON.ENERGY	Real		eV
PHUMOB	Logical	False	
PIEZOELECTRIC	Logical	False	
PIEZO.SCALE	Real	1.0	
POISSON	Logical	False	
POLAR.CHARGE	Real	0.0	cm^{-2}
POLAR.SCALE	Real	1.0	
POLARIZATION	Logical	False	
POST.SCHRO	Logical	False	
PRINT	Logical	False	
PRINT.REGION	Integer		
PRINT.MATERIAL	Character		
PRT.BAND	Logical	False	

Parameter	Type	Default	Units
PRT.COMP	Logical	False	
PRT.FLAGS	Logical	False	
PRT.IMPACT	Logical	False	
PRT.RECOM	Logical	False	
PRT.THERM	Logical	False	
PROGRAM	Logical	False	
PRT.BAND	Logical	False	
PRT.COMP	Logical	False	
PRT.FLAGS	Logical	False	
PRT.IMPACT	Logical	False	
PRT.RECOM	Logical	False	
PRT.THERM	Logical	False	
PT.HEAT	Logical	False	
QMINCONC	Real	1.0e-10	cm^{-3}
QTNL.DERIVS	Logical	False	
QTNLSC.BBT	Logical	False	
QTNLSC.EL	Logical	False	
QTNLSC.HO	Logical	False	
QTUNN	Logical	False	
QTUNN.BBT	Logical	False	
QTUNN.DIR	Real	0	
QTUNN.EL	Logical	False	
QTUNN.HO	Logical	False	
QTUNNSC	Logical	False	
QUANTUM	Logical	False	
QWELL	Logical	False	
QX.MAX	Real	RHS of device	μm
QX.MIN	Real	LHS of device	μm
QY.MAX	Real	Bottom of device	μm
QY.MIN	Real	Bottom of device	μm

Parameter	Type	Default	Units
REGION	Integer	0	
RH.ELEC	Real	1.1	
RH.HOLE	Real	0.7	
SBT	Logical	False	
SCHRO	Logical	False	
SHAPEOX	Logical	False	
SINGLET	Logical	False	
SHI	Logical	False	
SP.BAND	Logical	False	
SP.DIR	Integer	0	
SP.SMOOTH	Logical	False	
SPEC.NAME	Character		
SPONTANEOUS	Logical	False	
SRHIRAMOB	Logical	False	
STRESS	Logical	False	
STRUCTURE	Character		
SUBSTRATE	Logical	False	
SURFMOB	Logical	False	
TAT.NLDEPTH	Real	$(E_c - E_v)/2$	eV
TAT.NONLOCAL	Logical	False	
TAUMOB	Logical	False	
TAUTEM	Logical	False	
TAYAMAYA	Logical	False	
TEMPERATURE	Real	300	K
TFLDMB1	Logical	False	
TFLDMB2	Logical	False	
THETA.N	Real	60.0	degrees
THETA.P	Real	60.0	degrees
TRAP.COULOMBIC	Logical	False	
TRAP.TUNNEL	Logical	False	
TRIPLET	Logical	False	

Parameter	Type	Default	Units
TUNLN	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
TUNLP	Real	2.0×10^{-5}	$V^{1/3}\cdot cm^{2/3}$
UBGN	Logical	False	
UST	Logical	False	
UNSAT.FERRO	Logical	False	
VALDINOCI	Logical	False	
WELL.CNBS	Integer	1	
WELL.DEBUG	Logical	False	
WELL.FIELD	Logical	True	
WELL.GAIN	Real	1.0	
WELL.NX	Integer	10	
WELL.NY	Integer	10	
WELL.OVERLAP	Real		
WELL.VNBS	Integer	1	
YAMAGUCHI	Logical	False	
YAN	Logical	False	

Description

mf is one or more of the model flags described on the next page.

gp is one or more of the general parameters described in the "General Parameters" Section on page 19-178. These parameters are used to specify general information used during the simulation.

mdp is one or more of the model dependent parameters described in the "Model Dependent Parameters" Section on page 19-179.

Mobility Model Flags

ALBRCT enables the Albrecht mobility model for electrons and holes (see Chapter 3: "Physics", Equation 3-163).

ANALYTIC specifies an analytic concentration dependent mobility model for silicon which includes temperature dependence (see Chapter 3: "Physics", Equations 3-161 and 3-162).

ARORA specifies an analytic concentration and temperature dependent model according to Arora (see Chapter 3: "Physics", Equations 3-163 and 3-164).

CCSMOB specifies carrier-carrier scattering model according to Dorkel and Leturq (see Chapter 3: "Physics", Equations 3-167-3-171).

CONMOB specifies that a concentration dependent mobility model be used for silicon and gallium arsenide. This model is a doping versus mobility table valid for 300K only.

CVT specifies that the CVT transverse field dependent mobility model is used for the simulation. The alias for this parameter is LSMMOB.

FLDMOB specifies a lateral electric field-dependent model (see Chapter 3: "Physics", Equation 3-260 and Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-50). The EVSATMOD parameter may be used to define which field dependent equation is used.

HOPMOB specifies that the hopping mobility model will be used for electrons and holes (see Chapter 15: "OTFT/OLED: Organic and Polymer Materials Simulators").

KLA specifies that the Klaassen Mobility Model (see Chapter 3: "Physics", Equations 3-179-3-204) will be used for electrons and holes.

LSMMOB is an alias for CVT.

MASETTI enables the Masetti mobility model for electrons and holes (see Chapter 3: "Physics", Equations 3-165 and 3-166).

MIN.SURF specifies that the WATT, TASCH, or SHI mobility models should only apply to minority carriers.

MOBMOD specifies mobility degradation by longitudinal electric field only (MOBMOD=1), or by both longitudinal and transverse electric fields (MOBMOD=2). When MOBMOD=2, specify the ACC.SF, INV.SF, OX.BOTTOM, OX.LEFT, and OX.RIGHT parameters. This parameter is only used with TFLDMB1 and TFLDMB2. The alias for this parameter is PHUMOB.

PFMOB specifies that the Poole-Frenkel mobility model will be used for electrons and holes (see Chapter 15: "OTFT/OLED: Organic and Polymer Materials Simulators").

PHUMOB is an alias for KLA.

SHIRAMOB specifies that the Shirahata Mobility Model (see Chapter 3: "Physics", Equation 3-259) will be used for electrons and holes.

SURFMOB or **WATT** invokes the effective field based surface mobility model (see Chapter 3: "Physics", Equations 3-252 and 3-253). SURFMOB parameters are used in the calculation of surface mobility according to the Watt Model [173]. Do not specify this parameter unless S-PISCES is installed on your system.

TFLDMB1 or **SCHWARZ** specifies the use of transverse electric field-dependent mobility models. The electron model is based on the Schwarz and Russe equations [139] and is implemented in [146]. The hole model, which is used only when MOBMOD=2, is based on the Watt and Plummer equations [174].

TFLDMB2 or **TASCH** specifies a transverse electric field dependent mobility model for electrons and holes based on a semi-empirical equation [146, 147].

YAMAGUCHI specifies that the Yamaguchi transverse field dependent mobility model is used in the simulation.

Note: See Section 19.28: "MOBILITY" for alternative ways to set mobility models.

Recombination Model Flags

AUGER specifies Auger recombination (see Chapter 3: "Physics", Equation 3-294).

AUGGEN specifies that the Auger recombination model will be used as a generation and a recombination term.

CONSRH Specifies Shockley-Read-Hall recombination using concentration dependent lifetimes (see Equations 3-282 and).

KLAAUG enables Klaassen's model for concentration dependent auger coefficients (see Chapter 3: "Physics", Equations 3-292, 3-293, and 3-294).

KAUGCN, **KAUGCP**, **KAUGDN** and **KAUGDP** specify parameters of Klaassen's Auger recombination model (see Chapter 3: "Physics", Equations 3-162 and 3-163).

KLASRH enables Klaassen's model for concentration dependent lifetimes for Shockley Read Hall recombination (Chapter 3: "Physics", Equations 3-295 and 3-296).

KSRHCN, **KSRHCP**, **KSRHGN**, **KSRHGP**, **KSRHTN** and **KSRHTP** specify parameters of Klaassen's SRH recombination model (see Chapter 3: "Physics", Equations 3-274 and 3-275).

NI.FERMI includes the effects of Fermi statistice into the calculation of the intrinsic concentration in expressions for SRH recombination.

OPTR selects the optical recombination model (see Chapter 3: "Physics", Equation 3-294). When this parameter is specified, the COPT parameter of the MATERIAL statement should be specified.

SRH specifies Shockley-Read-Hall recombination using fixed lifetimes (see Chapter 3: "Physics", Equation 3-281).

TRAP.COULOMBIC specifies that the Poole-Frenkel barrier lowering for columbic wells will be used for traps.

TRAP.TUNNEL specifies that the trap-assisted tunneling model will be used for traps.

TAT.NLDEPTH gives the energy level of the trapping levels as the energy depth below the conduction band edge. If TAT.NLDEPTH is not specified then the trap energy level is set to be the midgap energy level. The units of TAT.NLDEPTH are eV. TAT.NLDEPTH applies only to the TAT.NONLOCAL model.

TAT.NONLOCAL enables the non-local tunneling model in the calculation of the field effect enhancement factors. Setting this flag will also set the TRAP.TUNNEL flag.

Direct Tunneling Flags

An alternative set of direct quantum tunneling models for calculating gate current can be set using DT.CUR on the MODELS statement. If you set this parameter, then select the particular model using DT.METH. The allowed values of DT.METH are 1, 2 and 3. The default value of DT.METH is 1.

DT.METH=1 selects a Fowler-Nordheim type model, with a correction for the trazepoidal, rather than the triangular shape of the potential barrier.

DT.METH=2 selects a WKB model with the assumption of a linearly varying potential. It involves a calculation of the classical turning points and integration between those limits.

DT.METH=3 selects an exact formula for a trapezoidal barrier that involves Airy function solutions.

You can also need to specify the type of tunneling taking place by using at least one of the following flags.

DT.CBET specifies electron tunneling from conduction band to conduction band.

DT.VBHT specifies holes tunneling from valence band to valence band.

DT.VBET specifies Band-to-Band tunneling.

Note: Specifying DT.CUR on the MODELS statement will invoke the self-consistent version of these models. You can also specify DT.CUR and associated parameters on the SOLVE statement, which will invoke the post processing version of this model for the particular SOLVE statement.

Note: The effective masses used in the tunneling calculations are the default or specified effective masses for the materials in question, unless you set the ME.DT and MH.DT parameters on the MATERIAL statement.

Generation Model Flags

AUTOBBT causes band-to-band tunneling parameters to be calculated from first principals (see Chapter 3: "Physics", Equations 3-369, 3-370, and 3-373).

BBT.FORWARD avoids convergence problems (if using BBT.NONLOCAL in a forward biased junction).

BBT.HURKX enables the Hurkx model (see Chapter 3: "Physics", Equation 3-369).

BBT.KL specifies a band-to-band tunneling model according to Klaassen.

BBT.NONLOCAL enables the non-local band-to-band tunneling model. No other band-to-band tunneling models should be enabled in the same region.

BBT.NLDERIVS enables the inclusion of non-local derivatives in the Jacobian matrix (if you set BBT.NONLOCAL).

BBT.STD specifies a standard band-to-band tunneling model (see Chapter 3: "Physics", Equation 3-373). The alias for this parameter is BTBT.

BH.FNONOS is the barrier height parameter for the FNONOS model.

BTBT is an alias for BBT.STD.

DEVDEG.E activates the device degradation caused by hot electron injection current. HEI should also be specified in this case.

DEVDEG.H activates the device degradation caused by hot hole injection current. HHI should also be specified in this case.

DEVDEG.B activates the device degradation caused by both the hot electron and hot hole injection currents. HEI and HHI should also be specified in this case.

Note: The specification of either DEVDEG.E, DEVDEG.H, or DEVDEG.B will initialize the density of electron (hole) like traps on the interface and will clear out the trapped electron (hole) density on the interface if any.

E.BENDING specifies that electron band bending will be taken into account for electron injection (see Chapter 3: "Physics", Equation 3-390).

ETA.FNONOS is the trap capture efficiency for the FNONOS model.

FERRO enables the ferroelectric permittivity model (see Section 3.6.8: "The Ferroelectric Permittivity Model").

FG.RATIO for 2D simulation specifies the ratio of the floating gate extent in the Z direction to the channel width in Z direction.

FN.CUR enables both FNORD and FNHOLES.

FNHOLES enables self-consistent analysis of hole Fowler-Nordheim currents.

FNONOS enables model for gate tunnelling in SONOS structures.

F.TRAP.TUNNEL specifies the name of a file containing a C-interpreter function to simulate trap assisted tunneling. The alias for this parameter is F.TRAP.DIRAC.

F.TRAP.COULOMBIC specifies the name of a file containing a C-interpreter function to simulate trap assisted tunneling with Coulombic wells.

FNORD selects a self-consistent Fowler-Nordheim tunneling model for electrons (see Chapter 3: "Physics", Equation 3-384). This is the recommended approach for calculating Fowler-Nordheim current.

FNHPP selects a post-processing Fowler-Nordheim tunneling model for holes (see Chapter 3: "Physics", Equation 3-385). Generally, the Fowler-Nordheim current does not cause convergence problems so this approach is not required.

FNPP selects a post-processing Fowler-Nordheim tunneling model for electrons (see Chapter 3: "Physics", Equation 3-384). Generally, the Fowler-Nordheim current does not cause convergence problems so this approach is not required.

GATE1 enables both HEI and HHI.

GATE2 is an alias for GATE1.

H.BENDING specifies that hole band bending will be taken into account for hole injection (see Chapter 3: "Physics", Equation 3-392).

HEI specifies the calculation of hot electron injection into oxides. This parameter can be used to simulate MOS gate current or EPROM programming. In transient mode, the oxide current is self-consistently added to the floating gate charge (see Chapter 3: "Physics", Equation 3-387).

HHI specifies the calculation of hot hole current into an oxide in a similar manner to HEI (see Chapter 3: "Physics", Equation 3-388).

II.NODE is an alias for OLDIMPACT.

II.VALDI is an alias for VALDINOCI.

IMPACT invokes the empirical impact ionization model with ionization coefficients taken from [61]. More rigorous impact ionization models may be specified with the IMPACT statement.

MASS.TUNNEL specifies the effective mass for band to band tunneling (see Chapter 3: "Physics", Equations 3-82 and 3-83).

N.CONCANNON enables the Concannon gate current model for electrons.

NEARFLG specifies the model used for oxide transport when HEI, HHI or FNSONOS are used. The default is false which sets a purely drift based model assigning gate current to the electrodes where the electric field lines through the oxide terminate. Setting NEARFLG replaces this model with one assuming the gate current is flowing to the electrode physically nearest the point of injection. This assumes a purely diffusion transport mechanism.Setting NEARFLG for devices with only one gate or a coarse mesh is advised.

NEWIMPACT specifies that impact ionization should use the form described in Chapter 3: "Physics", Equation 3-297.

NT.FNONOS is the trap area density for saturation mode of the FNONOS model.

OLDIMPACT specifies that the impact ionization should use the form described in Chapter 3: "Physics", Equation 3-296. The alias for parameter is II.NODE.

P.CONCANNON enables the Concannon gate current model for holes.

QTNL.DERIVS enables non-local couplings in the Jacobian matrix in the self-consistent direct quantum tunneling models.

QTNLSC.BBT enables band-to-band mode of the self-consistent direct quantum tunneling model.

QTNLSC.EL enables the self-consistent direct quantum tunneling model for electrons.

QTNLSC.HO enables the self-consistent direct quantum tunneling model for holes.

QTUNN.BBT enables the band-to-band mode of the direct quantum tunneling model. This does not work with the quantum confined variant of quantum tunneling.

QTUNN.DIR specifies the tunneling current direction if you specify a rectangular mesh using the QTX.MESH and QTY.MESH statements. A value of 0 means the y-direction (default) and a value of 1 means the x-direction.

QTUNN.EL enables direct quantum tunneling model for electrons (see Chapter 3: “Physics”, Section 3.6.6: “Gate Current Models”).

QTUNN.HO enables direct quantum tunneling model for holes.

QTUNN is the same as specifying QTUNN.EL, QTUNN.HO, and QTUNN.BBT. These models calculate the quantum tunneling current through an oxide.

QTUNNSC enables the self-consistent versions of the direct quantum tunneling models for gate current. These include the tunneling current self-consistently in the carrier continuity equations. QTUNNSC is the same as specifying QTNLSC.EL, QTNLSC.HO, and QTNLSC.BBT.

SBT is an alias for UST.

SHAPEOX specifies which algorithm will be used for calculating quantum tunneling current on an non-rectangular mesh. If this parameter is not specified, then the current is calulated in a direction normal to each segment of the semiconductor/oxide interface. If SHAPEOX is specified, then the minimum distance is found between each segment and a contact or a polysilicon region. The quantum tunneling current is calculated along the direction that gives this minimum distance.

UST enables the universal Schottky tunneling model (see Chapter 3: “Physics”, Section 3.5.2: “Schottky Contacts”). The alias for this parameter is SBT.

VALDINOCI enables the Valdinoci impact ionization model. See Chapter 3: “Physics”, Equations 3-312, 3-313, and 3-314. The alias for this parmeter is II.VALDI.

Carrier Statistics Model Flags

BGN specifies band-gap narrowing (see Chapter 3: “Physics”, Equation 3-46).

BGN2 is an alias for BGN.

BGN.KLA specifies that the bandgap narrowing model will use the Klaassen coefficients. The alias for this parameter is BGN2.

BGN.KLASSEN enables the use of Klaassen's parameters for defaults for bandgap narrowing parameters (see Chapter 3: “Physics”, Table 3-3).

BGN.SLOTBOOM specifies band-gap narrowing using the original Slotboom coefficients (see Chapter 3: “Physics”, Equation 3-46).

BOLTZMANN specifies that Boltzmann carrier statistics be used (see Chapter 3: “Physics”, Equations 3-25 and 3-26).

BQP.N enables the BQP model for electrons.

BQP.P enables the BQP model for holes.

BQP.NGAMMA sets the value of BQP γ factor for electrons.

BQP.NALPHA sets the value of BQP α factor for electrons.

BQP.NEUMANN uses explicitly enforced Neumann boundary conditions instead of the implicit implementation.

BQP.PALPHA sets the value of BQP α factor for holes.

BQP.PGAMMA sets the value of BQP γ factor for holes.

BQP.QDIR sets the direction of principal quantization.

FERMIDIRAC specifies that Fermi-Dirac carrier statistics be used (see Chapter 3: “Physics”, Equations 3-25 – 3-78).

INCOMPLETE accounts for incomplete ionization of impurities in Fermi-Dirac statistics (see Chapter 3: “Physics”, Equations 3-52 and 3-53).

INC.ION specifies that Equations 3-57 and 3-58 should be used for calculations of incomplete ionization.

IONIZ accounts for complete ionization of heavily doped silicon when using INCOMPLETE.

UBGN enables the universal bandgap narrowing model given in Equation 3-50.

Quantum Carrier Statistics Model Flags

2DXY.SCHRO is the flag that tells ATLAS to solve a 2D Schrodinger equation in x-y plane when you also specify the SCHRODINGER or P.SCHRODINGER flag. When it is off, 1D Schrodinger equations will be solved at each slice perpendicular to x-axis.

ALT.SCHRO specifies usage of an alternate Schrodinger solver.

BC.QUANT specifies Dirichlet boundary conditions for all insulator interfaces for the quantum transport equation.

CALC.FERMI specifies that the Fermi level used in calculation of the quantum carrier concentration in the Schrodinger-Poisson model is calculated from the classic carrier concentrations (Chapter 13: "Quantum: Quantum Effect Simulator", Table 13-2).

DG.N is an alias for QUANTUM (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.3: "Density Gradient (Quantum Moments Model)").

DG.P is an alias for P.QUANTUM (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.3: "Density Gradient (Quantum Moments Model)").

DGN.GAMMA specifies the fit factor for electrons for the electron density gradient model (see Chapter 13: "Quantum: Quantum Effect Simulator", Equation 13-5).

DGP.GAMMA specifies the fit factor for holes for the hole density gradient model (see Chapter 13: "Quantum: Quantum Effect Simulator", Equation 13-6).

DGLOG specifies that the gradient of the log of concentration is to be used in the calculation of the quantum potential in the density gradient model (see Chapter 13: "Quantum: Quantum Effect Simulator", Equation 13-7).

DGROOT specifies that the gradient of the root of the concentration is to be used in the calculation of the quantum potential in the density gradient model (see Chapter 13: "Quantum: Quantum Effect Simulator", Equation 13-8).

EIGENS specifies the number of eigenstates solved for by the Poisson-Schrodinger solver (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model").

EF.TOL controls the eigenvector tolerance on numerical accuracy for the alternate eigenvalue solver enabled by the NEW.EIG parameter.

EV.TOL controls eigenvalue tolerance on numerical accuracy for the alternate eigenvalue solver enabled by the NEW.EIG parameter.

FAST.EIG enables a computationally efficient version of the eigen-value solver.

FIXED.FERMI specifies that a constant Fermi level is used along each individual slice in the Y direction for carrier concentration calculation in the Schrodinger-Poisson Model (Chapter 13: "Quantum: Quantum Effect Simulator", Table 13-2).

HANSCHQM turns on the Hansch quantum effects approximation model for N channel MOS devices (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.5.1: "Hansch's Model").

N.DORT turns on the Van Dort quantum effects approximation model for N channel MOS devices (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.5.2: "Van Dort's Model").

NEW.EIG enables an alternate eigenvalue solver that obtains eigenvectors using the INVERSE ITERATION method.

N.QUANTUM is an alias for QUANTUM (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.3: "Density Gradient (Quantum Moments Model)").

NEW.SCHRO specifies that the new non-rectilinear self-consistent Schrodinger-Poisson solver will be used for an ATLAS mesh (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model"). When this is specified, the Schrodinger solver mesh should also be specified using the SX.MESH and SY.MESH statements.

ORTH.SCHRO specifies the critical value used for determination of the orthogonality of eigenvectors (wavefunctions) resulting from Schrodinger equation solutions. The orthogonality measurement is done by taking the sum of the product of individual samples from two eigenvectors in question and dividing the sum by the number of samples. If that measure exceeds the value specified by ORTH.SCHRO, then an error message will be issued during any subsequent output of structure files. This message indicates that you should carefully check any wavefunction data contained in the structure file and make adjustments as necessary.

NUM.BAND specifies the number of valence bands to retain for Schrodinger Poisson solutions.

NUM.DIRECT specifies the number of conduction band directions in the k space to retain for Schrodinger Poisson solutions.

OX.MARGIN is the maximum electron penetration into oxide. You can use it with OX.SCHRO in non-planar semiconductor-oxide interface or nonrectangular channel crossection (ATLAS3D) to reduce the number of nodes used in the solution of the Schrodinger equation.

OX.POISSON calculates the quasi-Fermi levels in insulators for Schrodinger Poisson solutions.

OX.SCHRO specifies that the band edges of insulators are included in the solution of Schrodinger's Equation in self-consistent solutions of the Schrodinger-Poisson Model (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model").

P.DORT turns on the Van Dort quantum effects approximation model for P channel MOS devices (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.5.2: "Van Dort's Model").

POST.SCHRO specifies to calculate the Schrodinger Equation (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model") for quantum effects as a post processing step.

P.SCHRODING computes the self consistent Schrodinger Poisson for holes.

QMINCONC specifies the minimum carrier concentration considered in the gradient calculations of the density gradient model or in Poisson's Equation in the self-consistent Schrodinger-Poisson Model (see Chapter 13: "Quantum: Quantum Effect Simulator", Sections 13.2: "Self-Consistent Coupled Schrodinger Poisson Model" and 13.3: "Density Gradient (Quantum Moments Model)").

QUANTUM enables the density gradient (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.3: "Density Gradient (Quantum Moments Model)").

QX.MAX, QX.MIN specify the minimum and maximum extent of the Poisson-Schrodinger solver along the x-axis direction (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model").

QY.MAX, QY.MIN specify the minimum and maximum extent of the domain of the Schrodinger solver (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model").

SCHRO enables the Poisson-Schrodinger solver (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model"). This can be used for zero carrier solutions only (specified by METHOD CARRIERS=0). This is typically combined with the

EIGENS parameter to control the number of eigenstates calculated. This is a 1D solver that is solved within the mesh between limits set by QX.MIN and QX.MAX.

SP.BAND is a flag that turns on a banded matrix solver used to solve the 2D Schrodinger equation in ATLAS3D. Otherwise, a full matrix solver is used.

SP.DIR specifies quantization direction in 1D and 2D Schrodinger Equations. It is used in materials with single electron band but with anistropic effective mass. It is active only when Schrodinger is True and NUM.DIRECT=1. Different values of SP.DIR for effective mass are given below.

In ATLAS, quantization is in y-direction:

SP.DIR=0 my = mc, dos_mass= mc

SP.DIR=1 my = ml, dos_mass= $\sqrt{mt1*mt2}$

SP.DIR=2 my = mt1, dos_mass= $\sqrt{m1*mt2}$

SP.DIR=3 my = mt2, dos_mass= $\sqrt{m1*mt1}$

In ATLAS3D, quantization is in y-z plane:

SP.DIR=0 (my,mz)= (mc,mc), dos_mass= mc

SP.DIR=1 (my,mz)= (mt1,mt2), dos_mass= ml

SP.DIR=2 (my,mz)= (ml,mt2), dos_mass= mt1

SP.DIR=3 (my,mz)= (mt1,ml), dos_mass= mt2

SP.SMOOTH enables smoothing of quantum carrier calculations. When you set the SP.SMOOTH parameter on the MODELS statement, the SP solver will use the degeneracy factor and effective mass evaluated at the location of the maximum value of the primary wave function for calculation of carrier concentration. This removes any discontinuities due to abrupt changes in mass or degeneracy.

Energy Balance Simulation Flags

A.TEMP is an alias for HCTE.

E.TAUR.VAR specifies that electron temperature dependent energy relaxation time is used. Use the TRE.T1, TRE.T2, TRE.T3, TRE.W1, TRE.W2, and TRE.W3 parameters on the material statement to specifies the energy relaxation time.

ET.MODEL is an alias for HCTE.

H.TAUR.VAR specifies that hole temperature dependent energy relaxation time is used. Use TRH.T1, TRH.T2, TRH.T3, TRH.W1, TRH.W2, and TRH.W3 parameters on the material statement to specifies the energy relaxation time.

HCTE specifies that both electron and hole temperature will be solved. The aliases for this parameter are A.TEMP and ET.MODEL.

HCTE.EL specifies that electron temperature will be solved.

HCTE.HO specifies that hole temperature will be solved.

F.KSN specifies the name of a file containing a C-INTERPRETER function specifying the electron Peltier coefficient as a function of electron energy.

F.KSP specifies the name of a file containing a C-INTERPRETER function specifying the hole Peltier coefficient as a function of hole energy.

GR.HEAT includes generation/recombination heat source terms in the lattice heat flow equation.

JOULE.HEAT includes the Joule heat source terms in the lattice heat flow equation.

KSN specifies which hot carrier transport model will be used for electrons. KSN=0 selects the hydrodynamic model and KSN=-1 selects the energy balance model.

KSP specifies which hot carrier transport model will be used for holes. KSP=0 selects the hydrodynamic model and KSP=-1 selects the energy balance model.

TAUMOB specifies the dependence of relaxation times with carrier temperature in the mobility definition. If TAUMOB is specified, the values of MATERIAL statement parameters TAUMOB.EL and TAUMOB.HO are dependent on the carrier temperature.

TAUTEM specifies the dependence of relaxation times with carrier temperature. If TAUTEM is specified, the values of MATERIAL statement parameters TAUREL.EL and TAUREL.HO are dependent on carrier temperature.

N.TEMP or **HCTE.EL** specifies that the electron temperature equation will be solved.

P.TEMP or **HCTE.HO** specifies that the hole temperature equation will be solved.

PASSLER enables Passler's bandgap versus temperature model.

PT.HEAT includes Peltier-Thomson heat sources in the lattice heat flow equation.

Lattice Heating Simulation Flags

HEAT.FULL enables all thermal sources and sinks (Joule heat, generation-recombination heat, and Peltier Thomson heat).

HEAT.PETHOM can be used to turn off the Peltier-Thomson heat source in the HEAT.FULL option.

LAT.TEMP or **L.TEMP** specifies that the lattice temperature equation will be solved. For lattice heating simulation there must be at least one thermal contact defined using the THERMCONTACT statement.

Note: These parameters should only be specified if GIGA or GIGA3D is enabled on your system.

Magnetic Field Model

BX is the X Component of Magnetic Field Vector (Tesla).

BY is the Y Component of Magnetic Field Vector (Tesla).

BZ is the Z Component of Magnetic Field Vector (Tesla).

RH.ELEC is the Hall scattering factor for electrons .

RH.HOLE is the Hall scattering factor for holes.

Model Macros

BIPOLAR selects a default set of models which are used when simulating bipolar devices. The bipolar models are CONMOB, FLDMOB, BGN, CONSRH, and AUGER.

BIPOLAR2 selects an alternative default set of models which are used when simulating bipolar devices. The bipolar models are FERMI, KLA, KLASRH, KLAAUG, FLDMOB, and BGN.

ERASE specifies a default set of models which are used to simulate EEPROM erasure. When it's specified, the MOS, FNORD, IMPACT, and BBT.KL models will be used.

MOS specifies a default set of models for MOS devices. The MOS models are CVT, SRH and FERMI.

MOS2 specifies a default set of models for MOS devices. The MOS2 models are KLA, SHI, SRH, FERMI and BGN.

PROGRAM specifies a default set of models used when writing to EEPROMS. When PROGRAM is specified, the MOS, HEI, and IMPACT models will be used.

General Parameters

ALL enables solution for both electrons and holes.

ASUB specifies the substrate lattice constant for strain calculations as described in Section 3.6.9: "Epitaxial Strain Tensor Calculation in Wurtzite".

BOUND.TRAP enables modeling of traps coincident with ohmic boundaries.

CARRIERS specifies the number of carriers to simulate, which should be 0, 1 or 2. The alias is NUMCARR.

CHUANG enables the Chuang gain/radiative recombination model.

DEVICE specifies which device in MIXEDMODE simulation the MODELS statement should apply to. The synonym for this parameter is STRUCTURE.

DRIFT.DIFF specifies that the drift-diffusion transport model is to be used. This implies that the electron and hole carrier temperature equations will not be solved.

ELECTRONS specifies that the electron continuity equation is included in simulation.

G.EMPIRICAL enables the empirical gain model from Chapter 3: "Physics", Section 3.9.4: "The Empirical Gain Model" (GAINMOD=2).

G.STANDARD enables the gain model from Chapter 3: "Physics", Section 3.9.3: "The Standard Gain Model" (GAINMOD=1).

HOLE specifies that the hole continuity equation is included in simulation.

ISHIKAWA enables Ishikawa's strain model for InGaAs/InGaAsP/InGaP (see Chapter 3: "Physics", Section 3.9.10: "Ishikawa's Strain Effects Model").

K.P enables using the k*p model effective masses and band edge energies for drift-diffusion simulation.

LI enables the Li gain/radiative recombination model.

LORENTZ enables Lorentz gain broadening.

MATERIAL specifies which material from Table B-1 in Appendix B: "Material Systems", will apply to the MODELS statement. If a material is specified, then all regions defined as being composed of that material will be affected.

NAME specifies which region will be applied to the MODELS statement. Note that the name must match the name specified in the NAME parameter of the REGION statement or the region number.

PRT.BAND enables run-time output of regional band parameter model information.

PRT.COMP enables run-time output of regional composition information.

PRT.FLAGS enables run-time output of regional model flag information.

PRT.IMPACT enables run-time output of regional impact ionization model information.

PRT.RECOM enables run-time output of regional recombination model information.

PRT.THERM enables run-time output of regional thermal model information.

PRINT prints the status of all models, a variety of coefficients, and constants. We recommended that you include this parameter in all ATLAS runs.

PRINT.REGION prints the model status and coefficients for the specified region.

PRINT.MATERIAL prints the model status and coefficients for the specified material.

REGION specifies the region number for this MODELS statement. The alias is NUMBER.

SPEC.NAME specifies the name of a file to collect the emission spectrum from the LASER simulator.

SPONTANEOUS includes the radiative recombination rates derived from the LI, YAN or CHUANG model into the drift diffusion.

Note: You need additional computational resources since the radiative rate must be numerically integrated for each grid point.

STRUCTURE is a synonym for DEVICE.

STRESS enables the silicon stress dependent bandgap model from Section 3.6.11.

SUBSTRATE is a logical parameter indicating that the specified region should be considered as the substrate for strain calcualtions described in Section 3.6.9: “Epitaxial Strain Tensor Calculation in Wurtzite”.

TAYAMAYA enables the model from Chapter 3: “Physics”, Section 3.9.5: “Tayamaya's Gain Model” (GAINMOD=3).

TEMPERATURE specifies the temperature in Kelvin.

UNSAT.FERRO enables unsaturated loop modeling for ferroelectrics.

YAN enables the Li gain/radiative recombination model.

Model Dependent Parameters

ARORA Model Parameters

The parameters that can only be used with the ARORA model (see Chapter 3: “Physics”, Equations 3-163 and 3-164) are AR.MU1N, AR.MU2N, AR.MU1P, and AR.MU2P.

BBT.KL and BBT.STD Model Parameters

The parameters used with the BBTKL model are BB.A1 and BB.B. The BB.A2 and BB.B parameters are used with the BBT.STD model. The BB.A1 and BB.A2 parameters are the pre-exponential coefficients in the band-to-band tunneling models. The BB.B parameter is the exponential coefficient used in both models. The default value of BB.B depends on which model is chosen (see Chapter 3: “Physics”, Equation 3-373).

CCSMOB Model Parameters

The CCS.EA, CCS.EB, CCS.HA, and CCS.HB parameters describe the dependence of mobility on doping, carrier concentration, and temperature (see Chapter 3: “Physics”, Equations 3-167 – 3-171).

FLDMOB Model Parameters

B.ELECTRONS is used in the field-dependent mobility expression for EVSATMOD=0 (see Chapter 3: “Physics”, Equation 3-260).

B.HOLES is used in the field-dependent mobility expression for EVSATMOD=0 (see Chapter 3: “Physics”, Equation 3-261).

E0 is used in the field dependent mobility model for EVSATMOD=1 (see Chapter 5: “Blaze: Compound Material 2D Simulator”, Equation 5-50).

EVSATMOD specifies which parallel field dependent mobility model (see Chapter 3: “Physics”, Equation 3-260 and Chapter 5: “Blaze: Compound Material 2D Simulator”, Equation 5-50) should be used for electrons. The value of EVSATMOD should be assigned as follows:

1. Use the standard saturation model. You also can apply the option parameter, MOBTEM.SIMPL (see Chapter 3: “Physics”, the “Carrier Temperature Dependent Mobility” Section on page 3-74 for more

information).

2. Use the negative differential velocity saturation model

3. Use a simple velocity limiting model.

In most cases the default value of 1 should be used.

HVSATMOD specifies which parallel field dependent mobility model (see Chapter 3: "Physics", Equation 3-261 and Chapter 5: "Blaze: Compound Material 2D Simulator", Equation 5-51) should be used for holes. The value of HVSATMOD should be assigned as follows:

1. Use the standard saturation model

2. Use a simple velocity limiting model.

In most cases the default value of 1 should be used.

Fowler-Nordheim Tunneling Model Parameters

The parameters used in this model are F.AE and F.BE. F.AE is the pre-exponential factor and F.BE is the exponential coefficient (see Chapter 3: "Physics", Equation 3-384).

WATT or SURFMOB Model Parameters

The parameters that can be used with the WATT model, include: ALN1, ALN2, ALN3, ALP1, ALP2, ALP3, ETAN, ETAP, MREFN1, MREFN2, MREFN3, MREFP1, MREFP2, and MREFP3 (see Chapter 3: "Physics", Equations 3-252 and 3-253).

TFLDMB1 and TFLDMB2 Model Parameters

ACC.**SF** specifies the accumulation saturation factor which describes the ratio of the majority carrier concentration in the accumulation layer before and after bending of conductivity and valence bands.

INV.**SF** specifies the inversion saturation factor which describes the ratio of the majority carrier concentration in the inversion layer before and after the bending of conductivity and valance bands.

OX.**BOTTOM** specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor.

OX.**LEFT** specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor.

OX.**RIGHT** specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor.

CONCANNON Model Parameters

CGATE.**N** specifies an empirical tuning factor for electrons in the Concannon gate current model.

CGATE.**P** specifies an empirical tuning factor for holes in the Concannon gate current model.

PEFF.**N** specifies an effective barrier height for electrons in the Concannon gate current model. An alias for IG.EB0.

PEFF.**P** specifies an effective barrier height for holes in the Concannon gate current model. An alias for IG.HB0.

THETA.**N** specifies the critical rejection angle for electrons in the Concannon gate current model.

THETA.**P** specifies the critical rejection angle for electrons in the Concannon gate current model.

C0 specifies the electron distribution weight factor in the Concannon gate current model.

CHIA specifies the electron distribution function constant in the Concannon gate current model.

CHIB specifies the electron distribution function constant in the Concannon gate current model.

CHI.**HOLES** specifies the hole distribution function constant in the Concannon gate current model.

ENERGY.**STEP** specifies the energy step for numeric integration in the Concannon gate current model.

INFINITY specifies the highest energy in numeric integration in the Concannon gate current model.

PATH.**N** specifies the mean free path in the oxide for electrons in the Concannon gate current model.

PATH.**P** specifies the mean free path in the oxide for holes in the Concannon gate current model.

HEI Model Parameters

The parameters that may be used with the HEI model include: IG.ELINR, IG.HLINR, IG.ELINF, IG.HLINF, IG.EBETA, IG.HBETA, IG.EETA, IG.HETA, IG.EEOX, IG.HEOX, IG.EB0, IG.HB0.

Table 19-3 lists aliases for parameters of the HEI model.

Table 19-3. HEI model aliases

Parameter	Alias	Parameter	Alias
IG.ELINF	LAMHN	IG.EBETA	BARLN
IG.ELINR	LAMRN	IG.HBETA	BARLP
IG.HLINF	LAMHP	IG.EETA	TUNLN
IG.HLINR	LAMRP	IG.HETA	TUNLP

N.DORT and P.DORT Model Parameters

B.DORT user-specifiable model parameter for the Van Dort quantum effects approximation model.

D.DORT user-specifiable model parameter for the Van Dort quantum effects approximation model.

LASER Simulation Parameters

CAVITY.**LENGTH** specifies the cavity length in the longitudinal direction (in μm).

GAINMOD specifies the local optical gain model to be used. GAINMOD=0 specifies that no optical gain model will be used. GAINMOD=1 specifies that the complex frequency-dependent gain model will be used (see Chapter 3: “Physics”, Section 3.9.4: “The Empirical Gain Model”). GAINMOD=2 specifies that the simple gain model will be used (see Chapter 3: “Physics”, Section 3.9.3: “The Standard Gain Model”).

LAS.ABSOR_SAT enables the non-linear absorption loss saturation model.

LAS.ABSORPTION enables the absorption loss model in LASER.

LAS.**EINIT**, **LAS**.**EFINAL** specify the lower and upper photon energies. LASER will calculate multiple longitudinal photon rates within this range. Using wide ranges can slow down simulation.

LAS.ESEP specifies the photon energy separation. If this is not specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range.

LAS.ETRANS specifies that the Helmholtz solver should use the transverse mode with the eigen-energy closes to the energy specified by the LAS.TRANS_ENERGY parameter.

LAS.FCARRIER enables the free carrier loss model in LASER.

LAS.GAIN_SAT enables the non-linear gain saturation model.

LAS.**ITMAX** specifies the maximum number of iterations allowed for LASER simulation at each bias point.

LAS.**LOSSES** specifies the total losses in Equation 8-5.

LAS.MIRROR specifies the percentage facet reflectivity (both facets are assumed to have this value of reflectivity) for the mirror loss in Laser. 100% reflectivity is equivalent to no mirror loss.

LAS.**NEFF** specifies the effective refractive index in Equation 9-2.

LAS.NMODE specifies the number of transverse modes simulated.

LAS.NX specifies the number of discrete samples in the laser mesh in the X direction.

LAS.NY specifies the number of discrete samples in the laser mesh in the Y direction.

LAS.**OMEGA** specifies the lasing frequency to be used in Chapter 8: "Laser: Edge Emitting Simulator", Equation 8-1. If model 2 is used for simulation, then this parameter specifies an estimate of the lasing frequency. Instead of using this parameter, PHOTON.ENERGY can be used to specify photon energy.

LAS.REFLECT specifies that the Helmholtz equation should be solved with the Y axis (X=0) as an axis of symmetry.

LAS.RF specifies the front mirror reflectivity in percent.

LAS.RR specifies the rear mirror reflectivity in percent.

LAS.**SIN** specifies an initial photon density in the fundamental lasing mode. This value provides an initial guess for subsequent iterations. This parameter is used only when the single frequency model has been selected.

LAS.SPECSAVE the spectrum file will be saved after every LAS.SPECSAVE LASER solution steps.

LAS.SPONTANEOUS enables the physically based model for spontaneous emission to be used.

LAS.**TAUSS** specifies the iteration parameter to be used for the photon rate equation when solving Equation 9-8.

LAS.TIMERATE specifies that the time dependent photon rate equation will be used in a transient laser simulation.

LAS.**TOLER** specifies the desired accuracy in photon areas.

LAS.TRANS_ENERGY specifies the transverse mode energy for mode selection with LAS.ETRANS set.

LAS.XMAX, **LAS.XMIN**, **LAS.YMAX** and **LAS.YMIN** specify the maximum and minimum limits on the laser Helmholtz solution mesh.

LAS_MAXCH specifies the maximum allowable relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.

LASER enables the LASER simulation.

LMODE specifies that multiple longitudinal models are to be taken into account during the LASER simulation.

PHOTON.**ENERGY** specifies the energy of photons. If model 2 is used for simulation, this parameter specifies only an initial estimate of the photon energy. Instead of using this parameter, LAS.OMEGA can be used to specify the lasing frequency.

SPEC.**NAME** specifies the name of a spectrum file, which LASER will produce for each bias point if the LMODES parameter has been specified.

Quantum Model Parameters

LED specifies that the region is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 11: "LED: Light Emitting Diode Simulator".

PIEZO.SCALE is an alias for POLAR.SCALE.

PIEZOELECTRIC is an alias for POLARIZATION.

POLARIZATION enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. The alias for this parameter is PIEZOELECTRIC. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

POLAR.CHARG specifies polarization charge densities to replace the density calculated using Equations 3-295 and 3-296.

POLAR.SCALE specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the REGION statement. The alias for this parameter is PIEZO.SCALE. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

QWELL specifies that the region is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.

Exciton Model Flags

EXCITONS specifies that singlet and triplet exciton continuity equations will be solved.

SINGLET specifies that the singlet exciton continuity equation will be solved.

TRIPLET specifies that the triplet exciton continuity equation will be solved.

Model Selection Example

This example selects concentration and field dependent mobilities, SRH recombination, and Auger recombination. This is a typical model set for bipolar simulation. The simulation temperature is 290K.

```
MODELS CONMOB FLDMOB SRH AUGER TEMP=290
```

Confirming Model Selection

To echo back model selections, parameters and material constants use

```
MODELS PRINT
```

Note: For the best model selection for different applications, see the Standard Examples set (see also Chapter 2: "Getting Started with ATLAS", Section 2.4: "Accessing The Examples" for more information about to how to access these examples).

19.30: MQW

MQW defines multiple quantum well structures in an existing device structure. This statement should follow all other structure definition statements (i.e., MESH, REGION, ELECTRODE, and DOPING).

Note: The method for defining quantum wells using the MQW statement described in this section has been superceeded by the method described in Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.7: "Multiple Quantum Well Model". Therefore, we highly recommended that you use to the newer syntax.

Syntax

```
MQW <parameters>
```

Parameter	Type	Default	Units
ACCEPTORS	Real	0.0	cm-3
ALT.SCHRO	Logical	False	
ASTR	Real	0.0	
BSTR	Real	0.0	
CSTR	Real	0.0	
DELTA	Real	0.341	eV
DONORS	Real	0.0	cm-3
DSTR	Real	0.0	
EPSILON	Real	0.0	
ESTR	Real	0.0	
FSTR	Real	0.0	
GAIN0	Real	1.0	
GAMMA0	Real	2.0×10-3	eV
GSCALE	Real	1.0	
ISHIKAWA	Logical	False	
LI	Logical	False	
LORENTZ	Logical	False	
MATERIAL	Character		
MC	Real	0.067	
MHH	Real	0.62	
MLH	Real	0.087	
MSTAR	Real	0.053	

NBS	Integer	15	
NWELL	Integer	1	
NX	Integer	10	
NY	Integer	10	
RESOLVE	Logical	True	
SPONTANEOUS	Logical	False	
STRAIN	Real	0.0	
STABLE	Integer	0	
SY	Real		microns
TAU.IN	Real	3.3×10-13	seconds
TE.MODES	Logical	True	
WW	Real		microns
WB	Real		microns
XCOMP	Real	0.0	
XMAX	Real		microns
XMIN	Real		microns
YAN	Logical	False	
YCOMP	Real	0.0	
YMAX	Real		microns
YMIN	Real		microns

Description

The MQW regions will be added to ATLAS when you specify the MQW statement. This statement is necessary to correctly model the recombination and gain effects of these MQW regions.

The multiple quantum well structure is restricted to place the quantum wells only parallel to the X axis (perpendicular to the Y axis). Also, if more than one well is specified, then each well will have the same width defined by the WW parameter. Each barrier between wells will then have the same width defined by the WB parameter.

The MQW structure is located within the rectangle defined by the XMIN, XMAX, YMIN, and YMAX parameters. In the Y direction, the wells are centered within the span of YMIN to YMAX. Make sure that the difference between YMIN and YMAX is larger than the sum of the well width (WW) and the barrier width (WB) multiplied by the number of wells (NWELL). In the X direction, the wells extend the entire length from XMIN to XMAX.

The quantum well material is specified by the MATERIAL parameter, modified by the composition fractions: XCOMP and YCOMP. The barrier material is given by the material specified in the region, where the wells are defined.

This implementation solves the Schrodinger's equation (see Chapter 13: “Quantum: Quantum Effect Simulator”, Equations and 13-2) for each quantum well, in order to calculate the quantum well-bound state energies and carrier distributions in the wells. You can specify the number of bound states to use

for this calculation with the NBS parameter. For laser gain calculations, only the lowest bound state energy in the conduction band and the highest bound state energy in the valence band are used. These bound state energies feed directly into the gain model and spontaneous emission models are shown in Chapter 3: "Physics", Section 3.9: "Optoelectronic Models"

MQW Parameters

ACCEPTORS specifies a uniform density of ionized acceptors in the quantum wells.

ALT.SCHRO specifies that the alternative Schrodinger solver should be used in the calculations.

ASTR, **BSTR**, **CSTR**, **DSTR**, **ESTR**, and **FSTR** specify coefficients to calculate the effective masses in the strained quantum well.

DELTA specifies the spin orbital splitting energy in the quantum well.

DONORS specifies the uniform density of ionized donors in the quantum wells.

EPSILON specifies the high frequency permittivity to used in calculating the gain and radiative recombinations.

GAIN0 scaling factor for the MQW gain models.

GAMMA0 specifies the width in eV (Note: GAMMA0=h/TAU.IN).

GSCALE specifies a scale factor to be multiplied by the optical gain.

ISHIKAWA specifies that the Ishikawa model [75] is used to account for strain effects (see Chapter 3: "Physics", Section 3.9.10: "Ishikawa's Strain Effects Model" for more information).

LI specifies that the Li model [33] be used for spontaneous emission in multiple quantum wells. The DELTA, MSTAR, MC and MHH parameters can be used with this model.

LORENTZ is the Lorentzian line broadening model for MQW gain. The half width of the Lorentzian shape function is controlled by the GAMMA0 parameter of the MQW statement. You can also set the TAU.IN parameter to specify the intraband relaxation time in seconds. For more information about the Lorentzian line broadening model, see Chapter 3: "Physics", Section 3.9.9: "Lorentzian Gain Broadening".

MATERIAL specifies the name of the material to be used in the wells. The barrier material is taken from whatever material is already in the structure at the MQW location.

MC is the conduction band effective mass.

MHH is the heavy hole effective mass.

MLH is the light hole effective mass.

MSTAR disperses energy in the conduction band. This doesn't necessarily equate to the conduction band effective mass, see [187].

NBS defines the number of bound states used in solving the Schrodinger equation.

NWELL specifies the number of quantum wells.

NX, **NY** specify the number of mesh points in the X and Y direction for resolving the MQW structure. Make sure that these are specified large enough to resolve the individual quantum wells. NY is defined as:

$$\mathrm{NY} = \left[\frac{\mathrm{YMAX} - \mathrm{YMIN}}{\langle YGridSpacing(\mu m)\rangle}\right] + 1 \qquad 19\text{-}4$$

RESOLVE specifies whether the band edge discontinuities at the well edges are resolved by the device equations.

SPONTANEOUS specifies whether the quantum well spontaneous recombination is included in the device continuity equations.

STABLE specifies the table row to be used for coefficients to calculate the effective masses in strained quantum wells.

STRAIN specifies the strain percentage in the quantum well.

SY specifies the maximum mesh spacing to be applied in the y direction through the quantum wells.

TAU.IN specifies the intraband relaxation time in seconds (Note: `GAMMA0=h/TAU.IN`).

TE.MODES specifies whether TE or TM modes will be used in calculating the asymmetrical factors in the LI model.

WB specifies the individual barrier width between wells in microns.

WW specifies the individual quantum well width in microns.

XCOMP defines the X composition fraction for the wells for ternary and quaternary materials.

XMIN, **XMAX**, **YMIN**, **YMAX** specify a rectangular box where the quantum wells are to be simulated. The units of these parameters are in microns.

YCOMP defines the Y composition fraction for the wells for quaternary materials.

YAN specifies that the Yan model [187] be used for spontaneous emission in multiple quantum wells. The `DELTA`, `MSTAR`, `MC` and MHH parameters can be used with this model.

Note: The MQW gain model is now enabled with `GAINMOD=4`, specified in the `MODELS` statement.

19.31: ODEFECTS

ODEFECTS actives the bandgap defect model for organic polymer materials and sets the parameter values. This model can be used when simulating organic polymer devices using the OTFT module (see Chapter 15: "OTFT/OLED: Organic and Polymer Materials Simulators").

Syntax

```
ODEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	false	
DEVICE	Character		
DFILE	Character		
EA	Real	0	eV
ED	Real	0	eV
F.OTFTACC	Character		
F.OTFTDON	Character		
HA	Real	0	cm^3
HD	Real	0	cm^3
HOPPING	Logical	False	
INT_LIM1	Real	0.0	eV
INT_LIM2	Real	0.0	eV
MATERIAL	Character		
NA	Real	0	cm^{-3}
ND	Real	0	cm^{-3}
NIA	Real	0	cm^{-3}
NID	Real	0	cm^{-3}
NUMA	Real	12	
NUMD	Real	12	
REGION	Real	All	
SIGAE	Real	1.0e-16	cm^2
SIGAH	Real	1.0e-14	cm^2
SIGDE	Real	1.0e-14	cm^2

Parameter	Type	Default	Units
SIGDH	Real	1.0e-16	cm^2
SIGMA	Real	0	eV
SIGMD	Real	0	eV
SIGMAIA	Real	0	eV
SIGMAID	Real	0	eV
STRUCTURE	Character		
TCA	Real	0.0	K
TCD	Real	0.0	K
X.MIN	Real		microns
X.MAX	Real		microns
Y.MIN	Real		microns
Y.MAX	Real		microns
Z.MIN	Real		microns
Z.MAX	Real		microns

Description

AFILE specifies the acceptor state density output file.

CONTINUOUS specifies that the continuous defect model will be used.

DEVICE specifies which device in MIXEDMODE simulation will apply to the ODEFECTS statement. The synonym for this parameter is STRUCTURE.

DFILE specifies the donor state density output file.

EA specifies the energy shift between the intrinsic and dopant states for acceptor-like traps.

ED specifies the energy shift between the intrinsic and dopant states for donor-like traps.

F.OTFTACC specifies the C-Interpreter file for the acceptor states energy distrubution function.

F.OTFTDON specifies the C-Interpreter file for the donor states energy distrubution function.

HA specifies the peak acceptor-like trap density.

HD specifies the peak donor-like trap density.

HOPPING specifies the effective transport hopping energy model is to be used.

INT_LIM1 specifies the lower limit for the numerical integration of the CONTINUOUS model.

INT_LIM2 specifies the upper limit for the numerical integration of the CONTINUOUS model.

MATERIAL specifies which material will apply to the ODEFECTS statement. If a material is specified, then all regions defined as being composed of that material will be affected.

NA specifies the total dopant density for acceptor-like traps.

ND specifies the total dopant density for donor-like traps.

NIA specifies the total intrinsic dopant density for acceptor-like traps.

NID specifies the total intrinsic dopant density for donor-like traps.

NUMA specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.

NUMD specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.

REGION specifies the region to which the ODEFECTS statement applies.

SIGAE specifies the electron capture cross-section for acceptor traps.

SIGAH specifies the hole capture cross-section for acceptor traps.

SIGAE specifies the electron capture cross-section for donor traps.

SIGAH specifies the hole capture cross-section for donor traps.

SIGMAA specifies the Gaussian width for the acceptor-like traps dopant distribution.

SIGMAD specifies the Gaussian width for the donor-like traps dopant distribution.

SIGMAIA specifies the Gaussian width for the acceptor-like traps intrinsic distribution.

SIGMAID specifies the Gaussian width for the donor-like traps intrinsic distribution.

STRUCTURE is a synonym for DEVICE.

TCA specifies the characteristic temperature for the acceptor-like trap states.

TCD specifies the characteristic temperature for the acceptor-like trap states.

X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN** and **Z.MAX** specify the bounding box for the ODEFECTS statement.

19.32: OINTDEFECTS

OINTDEFECTS actives the bandgap interface defect model for organic polymer materials and sets the parameter values. This model can be used when simulating organic polymer devices using the OTFT module (see Chapter 15: “OTFT/OLED: Organic and Polymer Materials Simulators”).

Syntax

```
OINTDEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	false	
DEVICE	Character		
DFILE	Character		
EA	Real	0	eV
ED	Real	0	eV
F.OTFTACC	Character		
F.OTFTDON	Character		
HA	Real	0	cm^3
HD	Real	0	cm^3
HOPPING	Logical	False	
INT_LIM1	Real	0.0	eV
INT_LIM2	Real	0.0	eV
MATERIAL	Character		
NA	Real	0	cm^{-3}
ND	Real	0	cm^{-3}
NIA	Real	0	cm^{-3}
NID	Real	0	cm^{-3}
NUMA	Real	12	
NUMD	Real	12	
REGION	Real	All	
SIGAE	Real	1.0e-16	cm^2
SIGAH	Real	1.0e-14	cm^2
SIGDE	Real	1.0e-14	cm^2

Parameter	Type	Default	Units
SIGDH	Real	1.0e-16	cm^2
SIGMA	Real	0	eV
SIGMD	Real	0	eV
SIGMAIA	Real	0	eV
SIGMAID	Real	0	eV
STRUCTURE	Character		
TCA	Real	0.0	K
TCD	Real	0.0	K
X.MIN	Real		microns
X.MAX	Real		microns
Y.MIN	Real		microns
Y.MAX	Real		microns
Z.MIN	Real		microns
Z.MAX	Real		microns

Description

AFILE specifies the acceptor state density output file.

CONTINUOUS specifies that the continuous defect model will be used.

DEVICE specifies which device in MIXEDMODE simulation will apply to the OINTDEFECTS statement. The synonym for this parameter is STRUCTURE.

DFILE specifies the donor state density output file.

EA specifies the energy shift between the intrinsic and dopant states for acceptor-like traps.

ED specifies the energy shift between the intrinsic and dopant states for donor-like traps.

F.OTFTACC specifies the C-Interpreter file for the acceptor states energy distrubution function.

F.OTFTDON specifies the C-Interpreter file for the donor states energy distrubution function.

HA specifies the peak acceptor-like trap density.

HD specifies the peak donor-like trap density.

HOPPING specifies the effective transport hopping energy model is to be used.

INT_LIM1 specifies the lower limit for the numerical integration of the CONTINUOUS model.

INT_LIM2 specifies the upper limit for the numerical integration of the CONTINUOUS model.

MATERIAL specifies which material will apply to the OINTDEFECTS statement. If a material is specified, then all regions defined as being composed of that material will be affected.

NA specifies the total dopant density for acceptor-like traps.

ND specifies the total dopant density for donor-like traps.

NIA specifies the total intrinsic dopant density for acceptor-like traps.

NID specifies the total intrinsic dopant density for donor-like traps.

NUMA specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.

NUMD specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.

REGION specifies the region to which the OINTDEFECTS statement applies.

SIGAE specifies the electron capture cross-section for acceptor traps.

SIGAH specifies the hole capture cross-section for acceptor traps.

SIGAE specifies the electron capture cross-section for donor traps.

SIGAH specifies the hole capture cross-section for donor traps.

SIGMAA specifies the Gaussian width for the acceptor-like traps dopant distribution.

SIGMAD specifies the Gaussian width for the donor-like traps dopant distribution.

SIGMAIA specifies the Gaussian width for the acceptor-like traps intrinsic distribution.

SIGMAID specifies the Gaussian width for the donor-like traps intrinsic distribution.

STRUCTURE is a synonym for DEVICE.

TCA specifies the characteristic temperature for the acceptor-like trap states.

TCD specifies the characteristic temperature for the acceptor-like trap states.

X.MIN, **X.MAX**, **Y.MIN**, **Y.MAX**, **Z.MIN** and **Z.MAX** specify the bounding box for the OINTDEFECTS statement.

19.33: OPTIONS

OPTIONS sets options for an entire run.

Syntax

```
OPTIONS  [<rcp>]
```

Parameter	Type	Default	Units
CINT.DLL	Logical	False	
CINT.MODE	Integer		
FAIL.QUIT	Logical	False	
NORMAL	Logical	True	
QUIET	Logical	False	
TRANSITION	Logical	False	
VERBOSE	Logical	False	

Description

CINT.DLL specifies that ATLAS should treat subsequent C-Interpreter files with the extension ".so" (on UNIX based machines) or ".dll" (on Windows based machines) as shared object libraries instead of C code files. The shared object libraries will be dynamically loaded at run-time.

CINT.MODE specifies the C-Interpreter compilation mode. If CINT.MODE is set to 0, all subsequent C-Interpreter files will be compiled in optimize mode. If CINT.MODE is set to 1, all subsequent C-Interpreter files will be compiled in command-line debug mode. If CINT.MODE is set to 2, all subsequent C-Interpreter files will be compiled in GUI debug mode. The default compliation mode is optimize.

FAIL.QUIT if enabled will cause the simulation to quit if the number of cutbacks exceeds the number specified by the MAXTRAP parameter of the METHOD statement and TRAP is enabled.

rcp is one or more of the run control parameters described below. These parameters, which are not normally used, specify debugging options.

NORMAL is the default specification for run-time output filtering. At this setting ATLAS prints out the most relevant information (e.g., mesh statistics, terminal voltages, currents, warnings, and error messages).

QUIET specifies the maximum of filtering of run-time output.

TRANSITION specifies that during calculation of bound state energies for optical transitions. The allowable transitions are printed to the run-time output.

VERBOSE specifies the minimum filtering of run-time output. You should specify VERBOSE if you want output of residual norms.

Example

The following syntax can be used to provide extra debugging output.

```
OPTION VERBOSE
```

19.34: OUTPUT

OUTPUT allows you to specify the data that will be stored in standard structure format files.

Syntax

```
OUTPUT <parameters>
```

Parameter	Type	Default	Units
ANGLE	Logical	False	
AREA	Logical	False	
BAND.PARAM	Logical	False	
BAND.TEMP	Logical	false	
CHARGE	Logical	False	
CON.BAND	Logical	False	
CONTACT	Integer		
DELTAV	Real	0.1	Volts
DEVDEG	Logical	False	
E.FIELD	Logical	True	
E.LINES	Logical	False	
E.MOBILITY	Logical	False	
E.TEMP	Logical	True	
E.VELOCITY	Logical	False	
EFIELD	Logical	True	
EIGENS	Integer	0	
EX.FIELD	Logical	True	
EX.VELOCITY	Logical	False	
EY.FIELD	Logical	True	
EY.VELOCITY	Logical	False	
EZ.FIELD	Logical	True	
EZ.VELOCITY	Logical	False	
FLOWLINES	Logical	False	
H.MOBILITY	Logical	False	
H.TEMP	Logical	True	
H.VELOCITY	Logical	False	
HCTE.JOULE	Logical	False	

Parameter	Type	Default	Units
HEI	Logical	False	
HHI	Logical	False	
HX.VELOCITY	Logical	False	
HY.VELOCITY	Logical	False	
HZ.VELOCITY	Logical	False	
IMPACT	Logical	True	
INV.AREA	Logical	False	
INV.ANGLE	Logical	False	
INAME	Character		
J.CONDUC	Logical	True	
J.DISP	Logical	False	
J.DRIFT	Logical	False	
J.DIFFUSION	Logical	False	
J.ELECTRON	Logical	True	
J.HOLE	Logical	True	
J.TOTAL	Logical	True	
JX.CONDUC	Logical	False	
JX.ELECTRON	Logical	True	
JX.HOLE	Logical	True	
JX.TOTAL	Logical	True	
JY.CONDUC	Logical	False	
JY.ELECTRON	Logical	True	
JY.HOLE	Logical	True	
JY.TOTAL	Logical	True	
JZ.CONDUC	Logical	True	
JZ.ELECTRON	Logical	True	
JZ.HOLE	Logical	True	
JZ.TOTAL	Logical	True	
KSN	Logical	False	
KSP	Logical	False	
L.TEMPER	Logical	True	
LRATIO	Real	1.0	

Parameter	Type	Default	Units
MINSET	Logical	False	
N.LINES	Integer	20	
NOISE	Logical	False	
NOISE.IMP	Logical	False	
NOISE.ALL	Logical	False	
OPT.INTENS	Logical	False	
OX.CHARGE	Logical	False	
OLD.AVG	Logical	True	
P.QUANTUM	Logical	False	
PERMITTIVITY	Logical	False	
PHOTOGEN	Logical	True	
POLAR.CHANGE	Logical	False	
QFN	Logical	True	
QFP	Logical	True	
QSS	Logical	False	
RECOMB	Logical	True	
SCHOTTKY	Logical	False	
T.QUANTUM	Logical	False	
TAURN	Logical	False	
TAURP	Logical	False	
TOT.DOPING	Logical	False	
TRAPS	Logical	True	
TRAPS.FT	Logical	False	
U.AUGER	Logical	False	
U.BBT	Logical	False	
U.LANGIVIN	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
U.TRAP	Logical	False	
VAL.BAND	Logical	False	

Parameter	Type	Default	Units
VECTORS	Logical	False	
X.COMP	Logical	True	
Y.COMP	Logical	True	

Description

BAND.**PARAM** specifies that the band parameters (e.g., ni, Ne, Nv, and c) are included in the standard structure file.

BAND.TEMP outputs the following temperature dependent band parameters.

- Sqrt(electron concentration * hole concentration) (cm^{-3})
- Conduction band effective density of states (cm^{-3})
- Valence band effective density of states (cm^{-3})
- Energy band gap (eV)
- Conduction Band Energy (eV)
- Valence Band Energy (eV)

CHARGE specifies that the net charge will be included in the standard structure file.

CON.**BAND** specifies that the conduction band edge will be included in the standard structure file.

DEVDEG causes the distribution of acceptor/donor like traps on the interface, hot electron/hole current density on the interface, and trapped electron/holes to be written to the structure file.

E.**FIELD** or **EFIELD** specifies that total electric field will be included in the standard structure file.

E.**LINES** specifies the electric field lines will be included in the standard structure file.

E.**MOBILITY** specifies that electron mobility will be included in the standard structure file.

E.**TEMP** specifies that the electron temperature will be included in the standard structure file.

E.**VELOCITY** specifies that the total electron velocity will be included in the standard structure file.

EIGENS specifies the maximum number of eigen energies and eigen functions to be written to the structure file from a Poisson- Schrodinger solution.

EX.**FIELD** specifies that the x-component of electric field will be included in the standard structure file.

EX.**VELOCITY** specifies that the x-component of electron velocity will be included in the standard structure file.

EY.**FIELD** specifies that the y-component of electric field will be included in the standard structure file.

EY.**VELOCITY** specifies that the y-component of electron velocity will be included in the standard structure file.

EZ.FIELD specifies that the z-component of the electric field will be included in the standard structure file.

EZ.VELOCITY specifies that the z-component of the electron velocity will be included in the standard structure file.

FLOWLINES specifies that the current flowlines will be included in the standard structure file.

H.**MOBILITY** specifies that hole mobility will be included in the standard structure file.

H.**TEMP** specifies that the hole temperature will be included in the standard structure file.

H.**VELOCITY** specifies that the total hole velocity will be included in the standard structure file.

HCTE.JOULE specifies that the volumetrically averaged Joule heating will be included in the standard structure file.

HEI specifies that the hot electron current density will be included in the standard structure file.

HHI specifies that the hot hole current density will be included in the standard structure file.

HX.**VELOCITY** specifies that the x-component of hole velocity will be included in the standard structure file.

HY.**VELOCITY** specifies that the y-component of hole velocity will be included in the standard structure file.

HZ.VELOCITY specifies that the z-component of the hole velocity will be included in the standard structure file.

IMPACT specifies that the impact ionization rate will be included in the standard structure file.

J.**CONDUC** specifies that the total conduction current density will be included in the standard structure file.

J.**DISP** specifies that the total displacement current density will be included in the standard structure file.

J.**ELECTRON** specifies that the total electron current density will be included in the standard structure file.

J.**HOLE** specifies that the total hole current density will be included in the standard structure file.

J.**TOTAL** specifies that the total current density will be included in the standard structure file.

JX.**CONDUC** specifies that the x-component of the total conduction current density will be included in the standard structure file.

J.DRIFT specifies that the drift current density will be included in the standard structure file.

J.DIFFUSION specifies that diffusion current density will be included in the standard structure file.

JX.**ELECTRON** specifies that the x-component of electron current density will be included in the standard structure file.

JX.**HOLE** specifies that the x-component of hole current density will be included in the standard structure file.

JX.**TOTAL** specifies that the x-component of total current density will be included in the standard structure file.

JY.**CONDUC** specifies that the y-component of the total conduction current density will be included in the standard structure file.

JY.**ELECTRON** specifies that the y-component of electron current density will be included in the standard structure file.

JY.**HOLE** specifies that the y-component of hole current density will be included in the standard structure file.

JY.**TOTAL** specifies that the y-component of total current density will be included in the standard structure file.

JZ.CONDUC specifies that the z-component of the conduction current density will be included in the standard structure file.

JZ.ELECTRON specifies that the z-component of the electron current density will be included in the standard structure file.

JZ.HOLE specifies that the z-component of the hole current density will be included in the standard structure file.

JZ.TOTAL specifies that the z-component of the total current density will be included in the standard structure file.

KSN specifies that electron Peltier coefficients are to be written to any saved structure file.

KSP specifies that hole Peltier coefficients are to be written to any saved structure file.

L.TEMPER specifies that lattice temperature will be included in the standard structure file.

MINSET that a minimum set of data (potential, carrier concentration, and electric field) will be included in the standard structure file.

OPT.**INTENS** specifies that optical intensity is included in the standard structure file.

OX.**CHARGE** specifies that fixed oxide charge is include in the standard structure file.

P.QUANTUM specifies that the Bohm quantum potential is included in the solution file.

PERMITTIVITY specifies the dielectric permittivity is saved.

PHOTOGEN specifies that the photogeneration rate will be included in the standard structure file.

POLAR.CHARGE specifies that polarization charge will be included in the structure file.

QFN specifies that the electron quasi-fermi level will be included in the standard structure file.

QFP specifies that the hole quasi-fermi level will be included in the standard structure file.

QSS specifies that the surface charge will be included in the standard structure file.

RECOMB specifies that the recombination rate will be included in the standard structure file.

SCHOTTKY specifies that recombination velocities and barrier lowering will be included in the standard structure file.

T.QUANTUM specifies that quantum temperatture from the density gradient model is included in the solution file.

TAURN specifies that electron relaxation times are to be written to any saved structure file.

TAURP specifies that hole relaxation times are to be written to any saved structure file.

TOT.**DOPING** specifies that total doping will be included in the standard structure file.

TRAPS specifies that trap density information will be included in the standard structure file.

TRAPS.FT specifies that the trap probability of occupation will be included in the standard structure file.

U.**AUGER** specifies that the Auger component of recombination is to be written to solution files.

U.BBT specifies that the band to band tunneling rate will be included in the standard structure file.

U.LANGIVIN specifies that the Langivin recombination rate will be included in the standard structure file.

U.**RADIATIVE** specifies that the radiative component of recombination is to be written to solution files.

U.**SRH** specifies that the SRH component of recombination is to be written to solution files.

U.TRAP specifies that the trap recombination/generation rate will be included in the standard structure file.

VAL.**BAND** specifies that the valence band edge will be included in the standard structure file.

VECTORS specifies that only vector components will be included in the standard structure file.

X.COMP specifies that the composition fraction, x , is to be written to solution files.

Y.COMP specifies that the composition fraction, y, is to be written to solution files.

Ionization Integral Parameters

INAME specifies the name of a contact for which electric field lines are calculated.

CONTACT specifies a contact number for which electric field lines are calculated.

LRATIO specifies the spacing ratio between adjacent electric field lines. Defaults to 1.0 for uniform spacing.

N.LINES specifies the number of electric field lines.

DELTAV since the electric field is near zero at the contact, the electric field line calculations begin at a distance from the contact at which the contact voltage has changed by DELTAV. Defaults to 0.1 V.

Note: See Section 19.44: "SOLVE" and the on-line examples for instructions on using ionization integrals.

NOISE Parameters

NOISE selects the total local noise source for output.

NOISE.IMP selects the impedance fields for output.

NOISE.ALL selects everything for output. Currently, the local noise source, the impedance fields, the short-circuit current Green's function, the individual microscopic noise sources, and the individual local noise sources.

Averaging Parameters for Vector Quantities

OLD.AVG specifies that the current and field quantities will be averaged using an older algorithm (from version 3.0.0.R and back). By default the new method is used.

ANGLE specifies that averaging of current and fields will be weighted by the size of the angle of triangles intersecting at the node.

INV.ANGLE specifies that averaging of current and fields will be weighted by the inverse of the size of the angle of triangles intersecting at the node.

AREA specifies that averaging of current and fields will be weighted by the areas of triangles intersecting at the node.

INV.AREA specifies that averaging of current and fields will be weighted by the inverse of areas of triangles intersecting at the node.

Note: Certain quantities that can be output into the structure file and subsequently displayed using TONYPLOT need special mention. These quantities are evaluated within ATLAS along the links between grid points. They are represented in the structure file at the grid points themselves. As such these quantities are subject to averaging. In particular, electric field and currents are averaged so as to take into account the vector nature of these values. Mobility is simply summed up over all the links surrounding the grid point and divided by the total number of links. Carrier velocities are derived by dividing the averaged current by the carrier density at the grid point and the fundamental electron charge, q.

An Example of Combining OUTPUT with SOLVE and SAVE

The OUTPUT statement is often used in conjunction with the SAVE statement. The following statement lines specify that current flowlines and electron velocity components are saved in all subsequent standard structure solution files.

```
OUTPUT FLOWLINES EX.VELO EY.VELO

SOLVE  PREVIOUS V5=2 OUTF=data1.str MASTER

SAVE  OUTF=data2.str
```

19.35: PHOTOGENERATE

PHOTOGENERATE statement specifies photogenetion of carriers inside the device. It ensures TMA compatibility with the PHOTOGEN statement.

Syntax

```
PHOTOGENERATE <parameters>
```

Parameter	Type	Default	Units
A1	Real	0.0	cm^{-3}
A2	Real	0.0	cm^{-3}/um
A3	Real	0.0	cm^{-3}
A4	Real	0.0	um^{-1}
CONSTANT	Real	0.0	cm^{-3}
EXPONENT	Real	0.0	um^{-1}
FACTOR	Real	0.0	cm^{-3}
LINEAR	Real	0.0	cm^{-3}/um
RADIAL	Real	∞	um
R.CHAR	Real	∞	um
X.END	Real	Min x of device	um
X.MAX	Real	Max x of device	um
X.MIN	Real	Min x of device	um
X.ORIGIN	Real	Min x of device	um
X.START	Real	Min x of device	um
Y.END	Real	Max y of device	um
Y.MAX	Real	Max y of device	um
Y.MIN	Real	Min y of device	um
Y.ORIGIN	Real	Min y of device	um
Y.START	Real	Min y of device	um

A1 is the synonym for CONSTANT. Added for TMA compatibility.

A2 is the synonym for LINEAR. Added for TMA compatibility.

A3 is the synonym for FACTOR. Added for TMA compatibility.

A4 is the synonym for EXPONENT. Added for TMA compatibility.

CONSTANT specifies the constant photogeneration rate.

EXPONENT specifies the exponential coefficient for depth dependent photogeneration.

FACTOR specifies the pre-exponential factor for depth dependent photogeneration.

LINEAR specifies the linear factor for depth dependent photogeneration.

RADIAL specifies the characteristic radial distance for Gaussian dependence of photogeneration rate in the direction perpendicular to the line along which the carriers are injected.

R.CHAR is the synonym for RADIAL. Added for TMA compatibility.

X.END specifies the x-coordinate of the end of the line along which photogenerated carriers are injected.

X.MAX is the maximum x-coordinate at which photogeneration occurs. Generation of carriers is zero for x>X.MAX.

X.MIN is the minimum x-coordinate at which photogeneration occurs. Generation of carriers is zero for x<X.MIN.

X.ORIGIN specifies the x-coordinate of the origin of the line along which photogenerated carriers are injected.

X.START is the synonym of X.ORIGIN. Added for TMA compatibility.

Y.END specifies the y-coordinate of the end of the line along which photogenerated carriers are injected.

Y.MAX is the maximum y-coordinate at which photogeneration occurs. Generation of carriers is zero for y>Y.MAX.

Y.MIN is the minimum y-coordinate at which photogeneration occurs. Generation of carriers is zero for y<Y.MIN.

Y.ORIGIN specifies the y-coordinate of the origin of the line along which photogenerated carriers are injected.

Y.START is the synonym of Y.ORIGIN. Added for TMA compatibility.

19.36: PROBE

PROBE allows you to output the value of several distributed quantities to the log file. The value at a specified location or the minimum, maximum, or integrated value within a specified area of the device will be saved to the log file at each bias or time point.

Note: PROBE is the most accurate way to determine the value of many parameters calculated by ATLAS. Parameters stored on node points in the structure files for TonyPlot are often interpolated and subject to noise.

Syntax

```
PROBE[MIN|MAX|INTEGRATED|x=<n> y=<n> z=<n> [DIR=<n>]][POLAR=<n>]<parameters>
```

Parameter	Type	Default	Units
ACC.TRAP	Logical	False	
ALPHAN	Logical	False	
ALPHAP	Logical	False	
APCURRENT	Logical	False	
AUGER	Logical	False	
BACK	Real	maximum z coord	µm
BAND	Integer	1	
BAND.GAP	Logical	False	
BEAM	Integer		
BND.ENER	Logical	False	
BOTTOM	Real	maximum y coord	µm
CHARGE	Logical	False	
COMPLIANCE	Real		
CON.BAND	Logical	False	
CONCACC.TRAP	Logical	False	
CONCDON.TRAP	Logical	False	
DEVICE	Character		
DIR	Real	0.0	degrees
DON.TRAP	Logical		
EMIN	Real		eV
EMAX	Real		eV
ENERGY	Real		eV

Parameter	Type	Default	Units
EXCITON	Logical	False	
FIELD	Logical	False	
FRONT	Real	minimum z coord	µm
FTACC.TRAP	Logical	False	
FTDON.TRAP	Logical	False	
GENERATION	Logical	False	
GR.HEAT	Logical	False	
H.CONC	Logical	False	
INTEGRATED	Logical	False	
J.CONDUCTION	Logical	False	
J.DISP	Logical	False	
J.ELECTRON	Logical	False	
J.HOLE	Logical	False	
J.PROTON	Logical	False	
J.TOTAL	Logical	False	
JOULE.HEAT	Logical	False	
KX	Logical	False	
KY	Logical	False	
KZ	Logical	False	
LMIN	Real		microns
LMAX	Real		microns
LANGEVIN	Logical	False	
LASER.INTENSITY	Logical	False	
LASER.GAIN	Logical	False	
LASER.MODE	Real		
LAT.TEMP	Logical	False	
LEFT	Real	minimum x coord	µm
LUMINOUS.INTENSITY	Logical	False	
MATERIAL	Character		
MAX	Logical	False	
MIN	Logical	False	
MAX.XITION	Logical	False	

Parameter	Type	Default	Units
MIN.XITION	Logical	False	
N.CONC	Logical	False	
N.MOB	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
NWELL	Logical	False	
P.CONC	Logical	False	
P.MOB	Logical	False	
P.TEMP	Logical	False	
PERMITTIVITY	Logical	False	
PHOTOGEN	Logical	False	
POLARIZATION	Logical	False	
POTENTIAL	Logical	False	
PT.HEAT	Logical	False	
PWELL	Logical	False	
QFN	Logical	False	
QFP	Logical	False	
R.TRAP	Logical	False	
R.BBT	Logical	False	
RADIATIVE	Logical	False	
RAUGER	Logical	False	
RECOMBIN	Logical	False	
REGION	Integer		
RIGHT	Real	maximum x coord	µm
RNAME	Character		
SRH	Logical	False	
STATE	Integer	1	
STRUCTURE	Character		
TOTAL.HEAT	Logical	False	
TOP	Real	minimum y coord	µm
VAL.BAND	Logical	False	
VEL.ELECTRON	Logical	False	

Parameter	Type	Default	Units
VEL.HOLE	Logical	False	
WAVE.FUN	Logical	False	
WAVELENGTH	Logical	False	
X.MIN	Real	maximum x coord	μm
X.MAX	Real	maximum x coord	μm
Y.MIN	Real	maximum y coord	μm
Y.MAX	Real	maximum y coord	μm
Z.MIN	Real	maximum z coord	μm
Z.MAX	Real	maximum z coord	μm

Description

ACC.TRAP specifies that the ionized acceptor trap density is to be probed. If multiple trap levels are present the ENERGY parameter should be set to the desired energy level that is to be probed.

ALPHAN specifies that the electron impact ionization coefficient is to be probed. The DIR parameter should also be specified as this is a vector quantity.

ALPHAN, **ALPHAP** specify that the electron or hole impact ionization constant alpha is to be probed.

ALPHAP specifies that the hole impact ionization coefficient is to be probed. The DIR parameter should also be specified as this is a vector quantity.

APCURRENT specifies that the available photocurrent is probed.

AUGER specifies that Auger recombination rate is probed.

BAND is the number of the electron or hole band characterized by effective mass.

The following shows the value of effective mass in the quantization direction y for different values of the BAND parameter on the PROBE statement and the SCHRODINGER, P.SCHRODINGER, NUM.DIRECT, NUM.BAND and SP.DIR parameters on the MODELS statement.

In ATLAS:

```
ml              (BAND=1 AND SCHRODINGER AND NUM.DIRECT>1)
mt1             (BAND=2 AND SCHRODINGER AND NUM.DIRECT>1)
mt2             (BAND=3 AND SCHRODINGER AND NUM.DIRECT=3)
mc              (BAND=1 AND SCHRODINGER AND NUM.DIRECT=1)


mhh             (BAND=1 AND P.SCHRODINGER AND NUM.BAND>1)
mlh             (BAND=2 AND P.SCHRODINGER AND NUM.BAND>1)
mso             (BAND=3 AND P.SCHRODINGER AND NUM.BAND=3)
mv              (BAND=1 AND P.SCHRODINGER AND NUM.BAND=1)
```

In ATLAS3D, the confinement is in y-z plane and the band is characterized by (mx,my, and mz).

```
(ml,mt1,mt2)          (BAND=1 AND SCHRODINGER AND NUM.DIRECT>1)
(mt1,ml,mt2)          (BAND=2 AND SCHRODINGER AND NUM.DIRECT>1)
```

```
(mt2,mt1,ml)          (BAND=3 AND SCHRODINGER AND NUM.DIRECT=3)
(mc,mc,mc)            (BAND=1 AND SCHRODINGER AND NUM.DIRECT=1)

(mhh,mhh,mhh)    (BAND=1 AND P.SCHRODINGER AND NUM.BAND>1)
(mlh,mlh,mlh)    (BAND=2 AND P.SCHRODINGER AND NUM.BAND>1)
(mso,mso,mso)    (BAND=3 AND P.SCHRODINGER AND NUM.BAND=3)
(mv,mv,mv)       (BAND=1 AND P.SCHRODINGER AND NUM.BAND=1)
```

If BAND<0 , it defaults to BAND=1.

If BAND>NUM.DIRECT and SCHRODINGER, it defaults to BAND=NUM.DIRECT.

If BAND>NUM.BAND and P.SCHRODINGER, it defaults to BAND=NUM.BAND.

For materials with one band but with anisotropic electron effective mass, you can change the effective mass in the case of BAND=1 and SCHRODINGER=1 and NUM.DIRECT=1 using the SP.DIR parameter on the MODELS statement.

BAND.GAP specifies that the bandgap is to be probed. If the probe is located in a equation well, the probe returns the energy associated with the minimum allowable transition.

BEAM specifies a beam number to focus the application of the probe to that specific beam.

BND.ENER probes the bound state energy characterized by the BAND and STATE parameters.

CHARGE specifies that the net charge is to be probed.

COMPLIANCE specifies the probed quantity compliance value. When the probed quantity reaches the specifies value, the SOLVE statement (or .DC or .TRANS statement in MIXEDMODE) will terminate and will execute the next statement.

CON.BAND probes the conduction band.

CONCACC.TRAP specifies that the acceptor trap concentration is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all acceptor energy levels will be probed.

CONCDON.TRAP specifies that the donor trap concentration is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all donor energy levels will be probed.

DEVICE specifies which device in MIXEDMODE simulation the probe statement should apply to. The synonym for this parameter is STRUCTURE.

DIR specifies the direction relative to the x axis in degrees associated with certain directed quantities. These quantities include FIELD, N.MOB, P.MOB, and POLARIZATION.

EMIN, **EMAX** specifies the range of energies to search for peak emission wavelength when the WAVELENGTH parameter is set on the PROBE statement.

DON.TRAP specifies that the ionized donor trap density is to be probed. If multiple trap levels are present the ENERGY parameter should be set to the desired energy level that is to be probed.

Note: The algorithm used finds the triangle in the mesh containing the specified X and Y values. Then the value of the DIR parameter is used to find which edge of the triangle lies in the direction nearest that value.

EXCITON specifies that the exciton density is probed.

FIELD specifies that a value of electric field is probed. The DIR parameter should also be specified if FIELD is used.

FTACC.TRAP specifies that the acceptor trap probability of occupation is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all acceptor energy levels will be probed.

FTDON.TRAP specifies that the donor trap probability of occupation is to be probed. You can use the ENERGY parameter to specify the trap energy level to be probed. If ENERGY is not specified, then the total value of all donor energy levels will be probed.

GENERATION specifies that the generation rate due to impact ionization is probed.

GR.HEAT requests a PROBE of Generation-Recombination heat in GIGA.

H.CONC specifies that proton concentration is probed.

INTEGRATED specifies that the probed value is to be integrated over all mesh points inside a box defined with the parameters X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN and Z.MAX.

J.CONDUCTION requests a PROBE of Conduction current (J.ELEC+J.HOLE).

J.DISP requests a PROBE of Displacement current.

J.ELECTRON requests a PROBE of Electron Current density.

J.HOLE requests a PROBE of Hole Current density.

J.PROTON specifies that the proton current density is probed.

J.TOTAL requests a PROBE of Total Current density (J.TOTAL+J.DISP).

JOULE.HEAT requests a PROBE of Joule-Heating in GIGA.

KX, **KY** and **KZ** specify directions in k space associated with probing conduction and valence band bound state energies. See also NWELL and PWELL.

LANGEVIN specifies that the Langevin recombination rate is probed.

LASER.GAIN specifies that the probe will operate on the LASER optical gain. For more about information LASER, see Chapter 8: "Laser: Edge Emitting Simulator".

LASER.INTENSITY specifies that the probe will operate on the LASER optical intensity. For more about information LASER, see Chapter 8: "Laser: Edge Emitting Simulator".

LASER.MODE specifies the optical mode for LASER.INTENSITY and LASER.GAIN.

LAT.TEMP specifies that the probe will operate on lattice temperature.

LMIN, **LMAX** specifies the range of wavelengths to search for peak emission wavelength when the WAVELENGTH parameter is set on the PROBE statement.

LUMINOUS.INTENSITY specifies that the probe will operate on the LUMINOUS optical intensity. For more information about LUMINOUS, see Chapter 10: "Luminous: Optoelectronic Simulator".

MATERIAL is assigned to a specific material name that will act to direct the probe to only address regions composed of the specified material.

MAX specifies that the probe will find the maximum value on the mesh.

MIN specifies that the probe will find the minimum value on the mesh.

MAX.XITION specifies that the maximum allowable transition between bound state energies is to be probed. If there are no allowable transitions between bound state energies, the band gap is probed.

MIN.XITION specifies that the minimum allowable transition between bound state energies is to be probed. If there are no allowable transitions between bound state energies, the band gap is probed.

NAME sets a character string that allows you to specify the description displayed by TONYPLOT.

N.CONC specifies that the probe will operate on electron concentration.

N.MOB specifies that the probe will operate on the electron mobility. The DIR parameter should also be specified if N.MOB is used.

N.TEMP specifies that the probe will operate on electron temperature.

NWELL specifies that the a specified conduction bound state energy is to be probed. To specify the bound state, you must also specify which bound state is to be probed with the STATE parameter (where 0 specifies the lowest bound state) and the direction in k space using the KX, KY or KZ parameters.

P.CONC specifies that the probe will operate on hole concentration.

P.MOB specifies that the probe will operate on the hole mobility. The DIR parameters should also be specified if P.MOB is used.

P.TEMP specifies that the probe will operate on hole temperature.

PT.HEAT requests a PROBE of Peltier-Thomson Heat in GIGA.

PERMITTIVITY specifies that material permittivity is probed.

PHOTOGEN specifies that photogeneration rate is probed.

POLARIZATION specifies that the probe will operate on ferroelectric polarization.The DIR parameter should also be specified if POLARIZATION is used.

POTENTIAL specifies that the probe will operate on electrostatic potential.

PWELL specifies that the a specified valence bound state energy is to be probed. To specify the bound state, you must also specify which bound state is to be probed with the STATE parameter (where 0 specifies the highest bound state) and the direction in k space using the KX, KY or KZ parameters.

QFN specifies that the probe will operate on the electron quasi-Fermi level.

QFP specifies that the probe will operate on the hole quasi-Fermi level.

R.TRAP specifies that the trap recombination is to be probed.

R.BBT specifies tha the band-to-band tunneling rate is to be probed.

RADIATIVE specifies that the probe will operate on radiative recombination rate.

RECOMBIN specifies that the probe will operate on net recombination rate.

SRH specifies that the probe will operate on SRH recombination rate.

STATE is the number of the bound state. It can take values from 1 to the maximum number of bound states, which is specified by the EIGENS parameter on the MODELS statement.

STRUCTURE is a synonym for DEVICE.

TOTAL.HEAT requests a PROBE of Total heat in GIGA (GR.HEAT+PT.HEAT+JOULE.HEAT).

VAL.BAND probes the valence band.

VEL.ELECTRON specifies that the probe will operate on electron velocity.

VEL.HOLE specifies that the probe will operate on hole velocity.

WAVE.FUN probes the wavefunction characterized by the BAND and STATE parameters.

WAVELENGTH specifies that the radiative emission wavelength is probed.

Region Parameters

BACK is a synonym for Z.MAX.

BOTTOM is a synonym for Y.MAX..

FRONT is a synonym for Z.MIN.

LEFT is a synonym for X.MIN.

RIGHT is a synonym for X.MAX.

TOP is a synonym for Y.MIN.

X.MIN specifies the minimum X coordinate for the minimum, maximum, or integrated probe.

X.MAX specifies the maximum X coordinate for the minimum, maximum, or integrated probe.

Y.MIN specifies the minimum Y coordinate for the minimum, maximum, or integrated probe.

Y.MAX specifies the maximum Y coordinate for the minimum, maximum, or integrated probe.

Z.MIN specifies the minimum Z coordinate for the minimum, maximum, or integrated probe.

Z.MAX specifies the maximum Z coordinate for the minimum, maximum, or integrated probe.

Example of Probing the Maximum Value

The following line will cause the maximum electron concentration on the grid to be output to the log file:

```
PROBE NAME=peak_electrons MAX N.CONC
```

Probing at a location Example

This syntax will cause the potential at the location X=0.5 Y=0.1 to be output to the log file.

```
PROBE NAME=mypotential X=0.5 Y=0.1 POTENTIAL
```

Vector Quantity Example

For vector quantities, the direction parameter, DIR, must be specified. These two lines allow a lateral mobility and vertical field in a MOSFET.

```
PROBE NAME=channel_mobility x=1 y=0.001 N.MOB DIR=0
PROBE NAME=channel_field x=1 y=0.001 FIELD DIR=90
```

In ATLAS3D, you can add a parameter POLAR to the PROBE statement. Its default value is 90°, which means the direction lies within the x-y plane. POLAR gives the angle respect to the z-axis. A value of zero means that the probed direction is along the z-axis. A value of POLAR between 0 and 90° will PROBE along a direction vector with a z-component of COS(POLAR).

```
PROBE NAME=electroncurrent dir=45 polar=60 j.electron  x=0.5 y=0.5 z=0.0
```

This probes the electron current at the specified position along the direction (0.61237, 0.61237, 0.5).

19.37: QTX.MESH, QTY.MESH

QTX.MESH and QTY.MESH allow you to specify a rectangular mesh in order to calculate quantum tunneling current in either the x-direction or y-direction. It is required in some situations for the direct quantum tunneling model for gate insulators. It is also required for non-local band-to-band and trap assisted tunneling models.

Syntax

```
QTX.MESH NODE=<n> LOCATION=<n>

QTX.MESH SPACING=<n> LOCATION=<n>

QTY.MESH NODE=<n> LOCATION=<n>

QTY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE specifies the mesh line index. These mesh lines are assigned consecutively.

LOCATION specifies the location of the grid line.

SPACING specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

Example

```
qtx.mesh loc=0.0 spac=0.25

qtx.mesh loc=1.0 spac=0.25


qty.mesh loc=1.5 spac=0.05

qty.mesh loc=2.0 spac=0.01

qty.mesh loc=3.0 spac=0.01

qty.mesh loc=3.5 spac=0.05
```

will set up a mesh in the rectangle bounded by x=0.0, x=1.0, y=1.5 and y=3.5. The mesh is finer in the y-direction because it is intended to calculate the tunneling current in this direction.

19.38: QUIT

QUIT stops execution of ATLAS.

Syntax

QUIT

Synonyms

STOP

END

EXIT

Description

The QUIT statement may be placed anywhere in an input file. ATLAS will stop execution upon encountering the QUIT statement. All input lines after the occurrence of the QUIT statement will be ignored for that execution of ATLAS.

Note: To quit and immediately restart ATLAS inside of DECKBUILD, use the GO ATLAS statement. Full details on the GO syntax statement are found in the DECKBUILD USER'S MANUAL.

19.39: REGION

REGION specifies the location of materials in a previously defined mesh. Every triangle must be defined as a material.

Syntax

```
REGION NUMBER=<n> <material> [<position>]
```

Parameter	Type	Default	Units
A-SILICO	Logical	False	
ACCEPTORS	Real	0.0	cm^{-3}
ALGAAS	Logical	False	
ALINAS	Logical	False	
ASUB	Real	1.0	Å
BOTTOM	Logical	False	
CALC.STRAIN	Logical	False	
COMPX.BOTTOM	Real	0.0	
COMPY.BOTTOM	Real	0.0	
COMPX.TOP	Real	0.0	
COMPY.TOP	Real	0.0	
CONDUCTOR	Logical	False	
CONVERT	Logical	False	
DEC.D2	Real	0.0	eV
DEC.D4	Real	0.0	eV
DEC.ISO	Real	0.0	eV
DEV.HH	Real	0.0	eV
DEV.LH	Real	0.0	eV
DEV.SO	Real	0.0	eV
DEV.ISO	Real	0.0	eV
DIAMOND	Logical	False	
DONORS	Real	0.0	cm^{-3}
FIXED.FERMI	Logical	False	
GAAS	Logical	False	
GAASP	Logical	False	
GERMANIU	Logical	False	

Parameter	Type	Default	Units
GRAD.12	Real		μm
GRAD.23	Real		mm
GRAD.34	Real		μm
GRAD.41	Real		μm
GRADED	Logical	False	
HGCDTE	Logical	False	
INGAAS	Logical	False	
INAS	Logical	False	
INASP	Logical	False	
INGAP	Logical	False	
INP	Logical	False	
INSULATO	Logical	False	
IX.LOW	Integer	left of structure	
IX.HIGH	Integer	right of structure	
IX.MAX	Integer	right of structure	
IX.MIN	Integer	left of structure	
IY.LOW	Integer	top of structure	
IY.HIGH	Integer	bottom of structure	
IY.MAX	Integer	top of structure	
IY.MIN	Integer	bottom of structure	
IZ.HIGH	Real		
IZ.LOW	Real		
IZ.MAX	Integer		
IZ.MIN	Integer		
LED	Logical	False	
MATERIAL	Character		
MODIFY	Logical	False	
NAME	Character		
NA.TOP	Real	0.0	cm^{-3}
ND.TOP	Real	0.0	cm^{-3}
NA.BOTTOM	Real	0.0	cm^{-3}

Parameter	Type	Default	Units
ND.BOTTOM	Real	0.0	cm^{-3}
NITRIDE	Logical	False	
NUMBER	Integer		
NX	Real	2	
NY	Real	2	
OXIDE	Logical	False	
OXYNITRI	Logical	False	
PIEZOELECTRIC	Logical	False	
PIEZO.SCALE	Real	1.0	
POLAR.CHARGE	Real	0.0	cm^{-2}
POLAR.SCALE	Real	1.0	
POLARIZATION	Logical	False	
POLYSILICON	Logical	False	
QWELL	Logical	False	
S.OXIDE	Logical	False	
SAPPHIRE	Logical	False	
SEMICOND	Logical	False	
SELF	Logical	False	
SI3N4	Logical	False	
SIC	Logical	False	
SIGE	Logical	False	
SILICON	Logical	False	
SIO2	Logical	False	
STAY	Logical	False	
STRAIN	Real	0.0	
STR.TOP	Real	0.0	
STR.BOT	Real	0.0	
SUBSTRATE	Logical	False	
SX	Real	0.0	
SY	Real	0.0	
THICKNESS	Real	0.0	microns
TOP	Logical	False	

Parameter	Type	Default	Units
USER.MATERIAL	Character		
WELL.CNBS	Integer	1	
WELL.DEBUG	Logical	False	
WELL.FIELD	Logical	True	
WELL.GAIN	Real	1.0	
WELL.NX	Integer	10	
WELL.NY	Integer	10	
WELL.OVERLAP	Real		
WELL.VNBS	Integer	1	
X.COMP	Real	0.0	
X.MAX	Real	right of structure	μm
X.MIN	Real	left of structure	μm
X.MOLE	Real	0.0	
Y.MOLE	Real	0.0	
Y.COMP	Real	0.0	
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MAX	Real		
Z.MIN	Real		
ZNSE	Logical	False	
ZNTE	Logical	False	

Description

n specifies a region number from 1 to 200.

material is one or more of the material names described below.

position is one or more of the position parameters described below.

Material Parameters

ACCEPTORS specifies a uniform density of ionized acceptors in the region.

ASUB specifies the substrate lattice constant for strain calculations as described in Section 3.6.9: "Epitaxial Strain Tensor Calculation in Wurtzite".

CALC.STRAIN specifies that the strain in the region is calculated from the lattice mismatch with adjacent regions.

COMPX.BOTTOM, **COMPY.BOTTOM**, **COMPX.TOP** and **COMPY.TOP** specify the composition fractions at the top and bottom of a region when linearly graded. Linear grading is specified by the

GRADED parameter. See Chapter 2: “Getting Started with ATLAS”, Section 2.6.4: “Automatic Meshing (Auto-meshing) Using The Command Language”.

CONDUCTOR specifies that the region is to be simulated as a conductor. This means that the conduction equation is solved for the region.

CONVERT is an alias for MODIFY.

DEC.D2 is a conduction band shift in a region for $\Delta 2$ electrons.

DEC.D4 is a conduction band shift in a region for $\Delta 4$ electrons.

DEC.ISO is a conduction band shift in a region for isotropic electrons.

DEV.HH is a valence band shift in a region for heavy holes.

DEV.LH is a valence band shift in a region for light holes.

DEV.SO is a valence band shift in a region for split-off holes.

DEV.ISO is a valence band shift in a region for isotropic holes.

DONORS specifies a uniform density of ionized donors in the region.

GRAD.<n> specifies the compositional gradings for heterojunctions along each side of the region rectangle or quadrilateral. The value of the GRAD parameters specifies the distance at which the composition fraction reduces to zero. A value of 0.0 specifies that the heterojunction is abrupt. The value of <n> can be the numbers: 12, 23, 34, and 41. The meanings of these numbers is down below.

- 12 = top surface
- 23= right hand side
- 34= bottom surface
- 41 = left hand side

These correspond to the point indices around the rectangular region working clockwise from top left.

GRADED specifies that the region has linear compositional or doping variation or both. See also COMPX.BOTTOM, COMPY.BOTTOM, COMPX.TOP, COMPY.TOP, ND.BOTTOM, ND.TOP, NA.BOTTOM and NA.TOP.

LED specifies that the region is to be treated as a light emitting region and included in postprocessing for LED analysis. See Chapter 11: “LED: Light Emitting Diode Simulator”.

MATERIAL specifies the material used for the region. Valid material names are listed in Appendix B: “Material Systems”, Table B-1. All materials are divided into three classes: semiconductors, insulators and conductors. Also see Appendix B for important information about requirements for each material class.

Note: You can specify the following logical parameters to indicate the region material instead of assigning the MATERIAL parameter: SILICON, GAAS, POLYSILI, GERMAINU, SIC, SEMICOND, SIGE, ALGAAS, A-SILICO, DIAMOND, HGCDTE, INAS, INGAAS, INP, S.OXIDE, ZNSE, ZNTE, ALINAS, GAASP, INGAP, INASP, OXIDE, SIO2, NITRIDE, SI3N4, INSULATO, SAPPHIRE and OXYNITRI.

MODIFY is used to modify the characteristics of regions imported into the simulator. See Chapter 2: “Getting Started with ATLAS”, Section 2.6.5: “Modifying Imported Regions”. The alias for this parameter is CONVERT.

NAME specifies the name of the region. The name can be used in the MODELS, MATERIAL, and IMPACT statements to provide regionally dependent models. This name is just a label. It doesn’t imply any material parameter settings.

ND.BOTTOM, **ND.TOP**, **NA.BOTTOM** and **NA.TOP** specify the doping concentrations at the top and bottom of a region when linearly graded. Linear grading is specified by the GRADED parameter. See Chapter 2: "Getting Started with ATLAS", Section 2.6.4: "Automatic Meshing (Auto-meshing) Using The Command Language".

NUMBER assigns a region number. Multiple REGION lines with the same number can be used to define region shapes made from several rectangles.

PIEZO.SCALE is an alias for POLAR.SCALE.

PIEZOELECTRIC is an alias for POLARIZATION.

POLARIZATION enables the automatic calculation of added interface charge due to spontaneous and piezoelectric polarization. The alias for this parameter is PIEZOELECTRIC. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

POLAR.CHARG specifies polarization charge densities to replace the density calculated using Equations 3-295 and 3-296.

POLAR.SCALE specifies a constant scale factor multiplied by the calculated spontaneous and piezoelectric polarization charges when you enable polarization by setting the POLARIZATION parameter of the REGION statement. The alias for this parameter is PIEZO.SCALE. See Chapter 3: "Physics", Section 3.6.10: "Polarization in Wurtzite Materials [23]".

QWELL specifies that the region is treated as a quantum well for calculation of radiative recombination or gain or both for certain optoelectronic models.

STRAIN specifies the strain in the region.

STR.BOT and **STR.TOP** specify the strain at the bottom and top of the region for calculations of the k.p model. The strain values between the top and bottom of the region are linearly interpolated.

SUBSTRATE is a logical parameter indicating that the specified region should be considered as the the substrate for strain calculations as described in Section 3.6.9: "Epitaxial Strain Tensor Calculation in Wurtzite".

USER.MATERIAL specifies a user-defined material name. The specified material name can be any name except that of a default material, such as Silicon. You should define each material with an accompanying definition in the MATERIAL statement. You can define up to 50 user-defined materials.

WELL.CNBS and **WELL.VNBS** specify the number of bound states retained for calculation of radiative recombination or gain if the region is treated as a quantum well as specified by the QWELL parameter.

WELL.FIELD specifies that the calculations of bound state energies should include the effects of the local field.

WELL.GAIN specifies a constant scale factor multiplied by the calculated gain to give the net gain used for certain optoelectronic calculations.

WELL.NX and **WELL.NY** specify the number of slices (WELL.NX) and the number of samples per slice (WELL.NY) are used in the solution of Schrodinger's equation to obtain the local bound state energies for calculation of radiative recombination or gain or both for certain optoelectronic models.

WELL.OVERLAP specifies the wavefunction overlap integral. If WELL.OVERLAP is not specified, then the overlap integral is calculated from the wavefunctions.

Note: The highest region number takes precedence if REGION definitions overlap.

Position Parameters

You can use grid indices to define a region only when the mesh is rectangular. To define a region with a rectangular mesh, use the X.MESH and Y.MESH statements to specify grid indices.

You can also use the IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters to specify x and y mesh line number values.

Note: To add regions to irregular meshes, such as those from ATHENA, specify the boundaries with the X.MAX, X.MIN, Y.MAX, and Y.MIN parameters.

BOTTOM specifies that the region is to be added at the bottom (starting at the maximum previously specified Y coordinate and extending in the positive Y direction) of the structure.

IX.**HIGH** specifies the maximum x value of the grid index. The alias for this parameter is IX.MAX.

IX.**LOW** specifies the minimum x value of the grid index. The alias for this parameter is IX.MIN.

IY.**HIGH** specifies the maximum y value of the grid index. The alias for this parameter is IY.MAX.

IY.**LOW** specifies the minimum y value of the grid index. The alias for this parameter is IY.MIN.

IZ.**HIGH** specifies the maximum z value of the grid index. The alias for this parameter is IZ.MAX.

IZ.**LOW** specifies the minimum z value of the grid index. The alias for this parameter is IZ.MIN.

IX.MAX, **IX.MIN**, **IY.MAX**, **IY.MIN**, **IZ.MAX**, and **IZ.MIN** are aliases for IX.HIGH, IX.LOW, IY.HIGH, IY.LOW, IZ.HIGH, and IY.LOW.

NX specifies the number of uniformly spaced mesh lines to be added to resolve the region in the X direction.

NY specifies the number of uniformly spaced mesh lines to be added to resolve the region in the Y direction.

STAY is used in automeshing to create material variations in the x-direction. See Chapter 2: “Getting Started with ATLAS”, Section 2.6.4: “Automatic Meshing (Auto-meshing) Using The Command Language”.

SX specifies the spacing between mesh lines to be applied at the edges of the region in the X direction.

SY specifies the spacing between mesh lines to be applied at the edges of the region in the Y direction.

THICKNESS specifies the thickness of the region in the Y direction. This parameter must accompany the specification of the TOP or BOTTOM parameters.

TOP specifies that the region is to be added at the top (starting at the minimum previously specified Y coordinate and extending in the negative Y direction) of the structure.

X.**COMP**, **X.MOLE** is the composition fraction (X) for a region with a composition dependent cations (e.g., AlGaAs).

X.**MAX** specifies the maximum x-boundary.

X.**MIN** specifies the minimum x-boundary.

Y.**COMP**, **Y.MOLE** is the composition fraction (Y) for a region with a composition dependent anions(e.g., InGaAsP).

Y.**MAX** specifies the maximum y-boundary.

Y.**MIN** specifies the minimum x-boundary.

Z.**MIN** specifies the minimum z-boundary.

Z.MAX specifies the maximum z-boundary.

Grid Indices Example

Define a silicon region extending from nodes 1 to 25 in the x direction and from nodes 1 to 20 in the y direction.

```
REGION NUM=1 IX.LO=1 IX.HI=25 IY.LO= 1 IY.HI=20 MATERIAL=SILICON
```

Non-Rectangular Region Example

Define a region that's composed of two separate rectangular areas. Note that REGION statements are cumulative.

```
REGION NUM=1 IX.LO=4 IX.HI=5 IY.LO=1 IY.HI=20 MATERIAL=OXIDE
REGION NUM=1 IX.LO=1 IX.HI=30 IY.LO=1 IY.HI=37 MATERIAL=OXIDE
```

Typical MOS Example

Define regions for a typical MOS structure.

```
REGION NUM=1 Y.MAX=0 MATERIAL=OXIDE
REGION NUM=2 Y.MIN=0 MATERIAL=SILICON
```

3D Region Definition Example

Define a cube of oxide within a region silicon in 3D.

```
REGION NUM=1 MATERIAL=SILICON
REGION NUM=2 Y.MAX=0.5 X.MIN=0.5 \
X.MAX=1.0 Z.MIN=0.5 Z.MAX=1.0 MATERIAL = OXIDE
```

Graded Heterojunction Definition Example

Define a graded heterojunction of AlGaAs/GaAs.

```
REGION NUM=1 MATERIAL=GaAs Y.MIN=1
REGION NUM=2 MATERIAL=AlGaAs Y.MAX=0.9 X.COMP=0.2 GRAD.34=0.1
```

In this case, the area between y=0.9 and 1.0 is graded in composition from 0.2 to 0.0. The Y.MAX parameter refers to the bottom of the uniform composition region. The actual bottom of the AlGaAs region is Y.MAX+GRAD.34

19.40: REGRID

REGRID allows you to refine a crude mesh. A triangle is refined when the chosen variable changes by more than one specified criteria.

Syntax

```
REGRID RATIO=<n> <var> [<lp>] [<cp>] [<io>]
```

Parameter	Type	Default	Units
ABSOLUTE	Logical	False	
ASCII	Logical	False	
CHANGE	Logical	True	
COS.ANG	Real	2.0	
DOPFILE	Character		
DOPING	Logical		cm^{-3}
E.TEMP	Logical	False	
EL.FIELD	Logical		V/cm
ELECTRON	Logical		cm^{-3}
H.TEMP	Logical	False	
HOLE	Logical		cm^{-3}
IGNORE	Integer		
IN.GREEN	Character		
LOCALDOP	Logical	False	
LOGARITHM	Logical	True	
MAX.LEVEL	Integer	1+ maximum level of grid	
MIN.CARR	Logical		cm^{-3}
NET.CARR	Logical		cm^{-3}
NET.CHRG	Logical		cm^{-3}
PISCES.OUT	Logical	False	
OUT.GREEN	Character	value of OUTFILE + tt	
OUTFILE	Character		
POTENTIAL	Logical		V
QFN	Logical		V
QFP	Logical		V
REGION	Integer		All

Parameter	Type	Default	Units
SMOOTH.K	Integer		
STEP	Real		
X.MAX	Real	Right	μm
X.MIN	Real	Left	μm
Y.MAX	Real	Bottom	μm
Y.MIN	Real	Top	μm

Description

RATIO or **STEP** specifies the maximum allowed variance across one element.

var is one of the variable parameters described below. The selected parameter is used as the basis for regridding.

lp is one of more of the location parameters described below. These parameters are used to select the areas which are to be refined.

cp is one or more of the control parameters. These parameters are used to control the plotted output.

io is one or more of the File I/O parameters.

Variable Parameters

DOPING selects net doping.

E.**TEMP** select electron temperature.

EL.**FIELD** selects electric field.

ELECTRON selects electron concentration.

H.**TEMP** selects hole temperature.

HOLE selects hole concentration.

MIN.**CARR** selects minority carrier concentration.

NET.**CARR** selects net carrier concentration.

NET.**CHRG** selects net charge.

POTENTIAL selects mid-gap potential.

QFN selects the electron quasi-Fermi level.

QFP selects the hole quasi-Fermi level.

Location Parameters

If no location parameters are specified, refinement will include:

- All regions for potential and electric field regrids
- All semiconductor regions for regrids which depend on the other variables

IGNORE specifies regions that are not to be refined.

REGION specifies regions which are refined according to the specified criterion. Other regions may be refined to maintain well-shaped triangles.

X.**MAX** uses device coordinates to specify the maximum x-value for refinement.

X.MIN uses device coordinates to specify the minimum x-value for refinement.

Y.MAX uses device coordinates to specify the maximum y-value for refinement.

Y.MIN uses device coordinates to specify the minimum y-value for refinement.

Control Parameters

ABSOLUTE specifies that the absolute value of the variable be used.

CHANGE determines whether to use the magnitude or the difference of a triangle variable as the refinement criterion. This parameter defaults to “difference”.

COS.ANGLE limits the creation of obtuse angles in the mesh by specifying obtuse criterion. If this parameter is used, nodes are added to the mesh so that the number of obtuse triangles is reduced.

Note: Be careful when using the COS.ANGLE parameter. Recommended values are from 0.8 to 0.95. Smaller values may dramatically increase the number of nodes.

LOCALDOP specifies that if minority carrier concentration exceeds local doping, the grid will be refined. This parameter is used in conjunction with minority carrier regrids.

LOGARITHM specifies a logarithmic refinement scale. Since many of the quantities may become negative, numerical problems are avoided by using $\log(x)=\text{sign}(x)\cdot\log10(1+|x|)$. If you wish to obtain the true logarithm of a quantity, the ABSOLUTE parameter must be specified before the LOGARITHM parameter is specified. The absolute value of a quantity is computed first, thereby eliminating negative arguments.

MAX.LEVEL specifies the maximum level of any triangle relative to the original mesh. This parameter defaults to one more than the maximum level of the grid, but can be set to a smaller value to limit refinement. Values less than or equal to zero are interpreted relative to the current maximum grid level.

SMOOTH.KEY specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows:

1. Triangle smoothing: All region boundaries remain fixed.
2. Triangle smoothing: Only material boundaries are maintained.
3. Node averaging.
4. Improved triangle smoothing method: This method uses diagonal flipping to reduce the number of obtuse triangles.
5. Aligns triangles with electric field gradient.

Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSUPREM4. Option 4 is similar to option 1, but option 4 usually creates less obtuse triangles.

File I/O Parameters

ASCII specifies that mesh files and triangle trees will be written in an ASCII rather than a binary format. This parameter has no effect on the device doping file (see the DOPFILE parameter).

DOPFILE specifies the name of a file, which contains device doping information. This file is created on the DOPING statement. Specifying DOPFILE avoids linear interpolation of doping values at newly created grid points by using the initial doping specification to apply doping to the new grid points.

IN.GREEN specifies a triangle tree for the mesh which will be used in this regrid. If this parameter is not specified, the program will look for a file with the same name as the current mesh plus, *tt*, at the end. If no such file exists, the program will not use a triangle tree for the previous mesh.

MASTER.OUT saves mesh and doping information in a standard structure file format.

OUTFILE specifies the name of a standard structure output file where mesh information will be stored. This parameter must be specified if the mesh is to be used for subsequent runs.

OUT.GREEN specifies the name of the file that holds the history of the triangle tree. This history is used in further regrid steps.

PISCES.OUT saves mesh and doping information in a binary PISCES-II format.Files in this format cannot be displayed in TONYPLOT.

Doping Regrid Examples

Starting with an initial grid, we refine twice so that all triangles with large doping steps are refined.

```
REGRID LOG DOPING RATIO=6 OUTF=grid1 DOPF=dopxx SMOOTH=4

REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx SMOOTH=4
```

A similar effect could be obtained with just one regrid statement.

```
REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx MAX.LEVEL=2
```

In both cases, two levels of refinement are performed. The first example is recommended because new doping information is introduced at each level of refinement. This produces better refinement criterion and fewer triangles.

Potential Regrid Example

Next, an initial solution is produced and triangles which exhibit large potential steps are refined.

```
SOLVE INIT

REGRID POTENTIAL RATIO=0.2 OUTF=grid3.str SMOOTH=4
```

Re-initializing after Regrid example

Often you need to re-solve the same bias point after a REGRID using the following style of syntax.

```
SOLVE VDRAIN=3.0

REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str

SOLVE PREV
```

Occasionally, you need to quit and restart ATLAS with the new mesh. To do this, type syntax such as:

```
SOLVE VDRAIN=3.0

REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str

go atlas

MESH INF=mygrid.str
```

After this MESH statement all models, material parameters and numerical methods have to be re-specified before any SOLVE statement.

19.41: SAVE

SAVE saves all node point information into an output file.

Note: In all cases the region boundaries, electrodes, mesh and doping are saved. If a SOLVE statement has preceded the SAVE statement all electrical data from the last solution is stored.

Syntax

```
SAVE OUTFILE=<filename> [MASTER]
```

Parameter	Type	Default	Units
ABSORB	Logical	False	
ANGPOWER	Character		
COMPARE	Logical	False	
COUPLING3D	Logical	False	
DIPOLE	Logical	False	
EMIN	Real		eV
EMAX	Real		eV
INTERFERE	Logical	False	
L.WAVE	Real	0.8	microns
LMAX	Real		microns
LMIN	Real		microns
MASTER	Character	True	
MIN.POWER	Real	1E-4	
MIR.BOTTOM	Logical	False	
MIR.TOP	Logical	False	
NSAMP	Integer	100	
NUMRAYS	Integer	180	
OUT.FILE	Character		
OUTFILE	Character		
PATTERNS	Character		
PISCES	Logical	False	
POLAR	Real	45.0	
RAYPLOT	Character		
REFLECTS	Integer	0	
SIDE	Logical	False	

Parameter	Type	Default	Units
SPECTRUM	Character		
STRUCTURE	Character	False	
TEMPER	Real	300.0	
X	Real	0.0	microns
XMAX	Real	0.0	microns
XMIN	Real	0.0	microns
XNUM	Integer	0	

Description

ABSORB takes absorption into account in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). The absorption is assumed to be constant specified for each material by the imaginary part of the refractive index.

ANGPOWER enables the reverse ray-tracing algorithm (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing") for analysis of output coupling of light from the structure of a Light Emitting Diode. ANGPOWER specifies the name of the output file for the angular power density vs. output angle dependence.

COUPLING3D calculates an output coupling coefficient, assuming a cylindrically symmetric light output with the light source located on the axis of symmetry (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). Takes into account the 3D nature of the problem with output coupling calculation.

DIPOLE specifies a particular angular distribution of the internal radiating field that corresponds to a preferred in-plane orientation of dipoles often relevant to OLED devices (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

EMAX, **EMIN** specify the energy range for saving a spectrum file.

INTERFERE specifies that rays originating in the same point are fully coherent in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). In this case, the phase information upon propagation is preserved. Phase change upon reflection is also considered. Thus, interference of rays exiting the device at the same angle are taken into account.

LMAX, **LMIN** specify the range of wavelength for saving spectrum files.

L.WAVE specifies the wavelength for reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

MASTER specifies that the output file will be written in a standard structure format.Files in this format can be plotted in TONYPLOT.

MIN.POWER specifies the minimum relative power of a ray (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). The ray is not traced after its power falls below MIN.POWER value. This is useful to limit the number of rays traced. The default value is MIN.POWER=1e-4.

MIR.TOP specifies that the top surface of the device be treated as an ideal mirror in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

MIR.BOTTOM specifies that the bottom surface of the device be treated as an ideal mirror in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

NSAMP specifies the number of samples to use for a spectrum plot.

NUMRAYS specifies the number of rays starting from the origin in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). The default is 180. Acceptable range is 36-3600.

OUT.FILE is a synonym for OUTFILE.

OUTFILE specifies the name of an output file name. The synonym for this parameter is OUT.FILE.

PISCES specifies that the output file will be written in the original PISCES-II format.

PATTERNS specifies a character string representing the root of the file names where near and far field patterns are written for LASER or VCSEL. The near field pattern file is appended with the string *.nfp* and the far field pattern file is appended with the string *.ffp*.

POLAR specifies polarization of the emitted photons in degrees while linearly polarized light is assumed (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). Parallel (TM-mode, POLAR=0.0) and perpendicular (TE-mode, POLAR=90.0) polarizations result in significantly different output coupling values. Use the default value (POLAR=45.0) if there is no preferred direction of polarization of emitted photons (unpolarized light emission).

RAYPLOT specifies the name of the output file containing the information on each ray exiting the device in single origin reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). This file is only created when the single origin for all rays is assumed. The information includes ray output angle, relative ray power (TE and TM-polarization and total), and initial internal angle at the origin (only if INTERFERE parameter is unspecified). 0° angle corresponds to the rays in the X axis direction. 90° angle corresponds to the rays in the Y axis direction.

REFLECTS specifies a number of reflections to be traced for each ray in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing"). The default value is REFLECTS=0. The maximum allowed value is REFLECTS=10. Setting the number of reflections to 3 or 4 is often a good choice.

SIDE specifies that the rays reaching the sides of the device are terminated there and do not contribute to the total light output (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

SPECTRUM specifies the name of a file for saving spectrum plots.

STRUCTURE is a synonym for SAVE.

TEMPER is the temperature (needed for using appropriate refractive indexes of the materials in reverse ray-tracing). The default setting of 300 K will be used if TEMPER is unspecified.

X is the x-coordinate of light origin for a single point reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

XMAX is the maximum x-coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

XMIN is the minimum x-coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

XNUM specifies the number of points along x-axis within a rectuangular area in multiple origin reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

Y is the y-coordinate of light origin for a single point reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

YMAX is the maximum y-coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

YMIN is the minimum y-coordinate of a rectangular area containing multiple origin points in reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

YNUM specifies the number of points along y-axis within a rectuangular area in multiple origin reverse ray-tracing (see Chapter 11: "LED: Light Emitting Diode Simulator", Section 11.5: "Reverse Ray-Tracing").

Basic Save Example

```
SOLVE V1=5

SAVE OUTF=data1.str
```

is equivalent to

```
SOLVE V1=5 OUTF=data1.str MASTER
```

Save Example with User-Defined Output

In the second example, the SAVE and OUTPUT commands are used to produce two output files for the same bias. The OUTPUT statement selects which data will be stored in each file. The first file (data1.str) contains the default contents, total electric field, and components of electron velocity. The second file (data2.str) contains components of hole velocity and band edge potentials. Note that the EX.VELO and EY.VELO parameters are used to prevent electron velocity components from being stored in file data2.

```
OUTPUT E.FIELD EX.VELO EY.VELO

SAVE OUTF=data1.str

OUTPUT HX.VELO HY.VELO CON.BAND VAL.BAND ^EX.VELO ^EY.VELO

SAVE OUTF=data2.str
```

Note: You can customize the contents of the saved file by using the OUTPUT statement.

19.42: SET

The SET statement is used to define variables for substitution into ATLAS syntax. SET commands are executed by DECKBUILD.

Note: Full documentation of the SET statement is found in the DECKBUILD USER'S MANUAL.

Numeric Variable Example

Define a numerical variable. Use it in a calculation and substitute it into the ATLAS syntax for REGION definition

```
SET MYLENGTH=0.1

SET HALFLENGTH= $"MYLENGTH"*0.5

...

REGION NUM=1 MATERIAL=SILICON X.MIN=$"HALFLENGTH" X.MAX=$"MYLENGTH"
```

String Variable Example

Define a string variable to use as part of a filename

```
SET LABEL=testcase1

..

LOG OUTF=$"LABEL".log

..

SAVE OUTF=bias_$"LABEL"_25.str
```

This will produce files called `testcase1.log` and `bias_testcase1_25.str`

19.43: SINGLEEVENTUPSET

SINGLEEVENTUPSET specifies the values of parameters used in Single Event Upset modeling.

Syntax

```
SINGLEEVENTUPSET <parameters>
```

Parameter	Type	Default	Units
A1	Real		
A2	Real		
A3	Real		
A4	Real		
B1	Real		
B2	Real		
B3	Real		
B4	Real		
B.DENSITY	Real	0	cm^{-3}
BEAM.RADIUS	Real	0	μm
DENSITY	Real		cm^{-3}
ENTRYPOINT	Real	vector	μm
EXITPOINT	Real	vector	μm
PCUNITS	Logical	False	
RADIALGAUSS	Logical	False	
RADIUS	Real		μm
RESCALE	Logical	False	
TFINAL.SEU	Real		
T0	Real		c
TC	Real		c
TF	Real		sec
UNIFORM	Logical	False	

Description

A1, **A2**, **A3** and **A4** are the first set of parameters for the length dependence of the charge generation pulse.

B1, **B2**, **B3** and **B4** are the second set of parameters for the length dependence of the charge generation pulse.

B.DENSITY specifies the number of electron-hole pairs per unit volume or generated charge in pico Coulumbs per micron if the PCUNITS parameter is specified.

BEAM.RADIUS is the radius of the beam where the generation rate is maintained constant. Beyond this point the generation will decay by either an exponential or by a Gaussian function.

DENSITY specifies the number of electron-hole pairs per unit volume generated along the alpha particle track.

ENTRYPOINT specifies the x, y, and z coordinates of the beginning of the alpha particle track. The specified point should belong to the semiconductor region.

EXITPOINT specifies the x, y, and z coordinates of the end of the alpha particle track. The specified point should belong to the semiconductor region.

PCUNITS sets the units of B.DENSITY to be pC per micron.

RADIALGAUSS specifies the Gaussian radial dependence of the charge generation pulse. By default the exponential dependence is used.

RADIUS specifies the radius of the alpha particle track.

RESCALE causes nodal generation rates to be scaled by the ratio of the integral of the analytic generation rate divided by the numerically integrated value.

Note: This may cause problems when the track radius approaches device dimensions.

TFINAL.SEU specifies the finish time for the track, this defaults to the finish time of the transient simulation if TFINAL.SEU is not specified.

T0 specifies the peak in time of the charge generation pulse.

TC specifies the width of the charge generation pulse.

UNIFORM specifies that a uniform generation rate corresponding to that specified by the DENSITY parameter is applied.

SEU Example

This statement specifies a track path, radius and density:

```
SINGLEEVENTUPSET ENTRYPOINT="1.5,2.0,0.0"\
                 EXITPOINT="1.0,1.0,4.0 RADIUS=0.05" \
                 DENSITY=1E18
```

Note: For user-defined Single Event Generation profiles, the C-INTERPRETER function, F3.RADIATE, on the BEAM statement can be used.

19.44: SOLVE

SOLVE instructs ATLAS to perform a solution for one or more specified bias points.

Syntax

```
SOLVE [<ion>] <dc> [<fp>][<ep>][<tp>][<ac>][<photo>] [<thermal>]
```

Parameter	Type	Default	Units
AC.ANALYSIS	Logical	False	
ANAME	Character		
AR.INDEX	Real	1.0	
AR.THICKNESS	Real	0.0	microns
ASCII	Logical	False	
AUTO	Logical	True	
B<n>	Real	0.0	W/cm^2
BRANIN	Logical	True	
C.IMAX	Real	10^{-4}	A
C.IMIN	Real	-10^{-4}	A
C.VSTEP	Real	0.0	V
C.VMAX	Real	5.0	V
C.VMIN	Real	-5.0	V
CNAME	Character		
COMPLIANCE	Real		
CONTINUE	Logical	False	
CURVETRACE	Logical	False	
CYCLES	Integer	1	
CYCLIC.BIAS	Logical	False	
CYCLIC.RELAX	Real	0.2	
CYCLIC.TOL	Real	1.0×10^{-5}	
DECAY	Real		s
DELTAV	Real	0.1	V
DIRECT	Logical	False	
DT	Real	0	s
DT.CBET	Logical	False	

Parameter	Type	Default	Units
DT.CUR	Logical	False	
DT.METH	Real	1	
DT.VBET	Logical	False	
DT.VBHT	Logical	False	
E.COMPL	Real		
E.CRIT	Real	1.0×10^{-8}	
ELECTRODE	Integer		
EMAX	Real	0.0	eV
EMIN	Real	0.0	eV
ENDRAMP	Real		s
FREQUENCY	Real		Hz
FSTEP	Real	0	Hz
GRAD	Logical	False	
I<n>	Real		A/μm
IFINAL	Real		A/μm
IMULT	Logical	False	
INAME	Character		
INDEX.CHECK	Logical	False	
INITIAL	Logical	False	
ION.CRIT	Real	1.0	
ION.ELEC	Integer	see Description	
IONIZINT	Logical	False	
IONLINES	Integer	50	none
IONSTOP	Logical	True	
ISTEP	Real	0.0	A/μm
LAMBDA1	Real		μm
LIT.STEP	Real		W/cm^2
LMAX	Real	0.0	μm
LMIN	Real	0.0	μm
LOCAL	Logical	False	
LRATIO	Real	1.0	
L.WAVE	Real		μm

Parameter	Type	Default	Units
MASTER	Logical	False	
MAX.INNER	Integer	25	
MLOCAL	Logical	False	
MULT.FREQ	Logical	False	
N.BIAS	Real		V
NAME	Character		
NB1	Real		V
NB2	Real		V
NB3	Real		V
NB4	Real		V
NB5	Real		V
NB6	Real		V
NB7	Real		V<B>
NB8	Real		V
NFSTEPS	Integer	0	
NLAYERS	Real	15	
NOCONTACT	Logical	False	
NOCURRENT	Logical	False	
NOINTERFACE	Logical	False	
NOISE.SS	Logical	False	
NSAMP	Integer	100	
NSTEPS	Integer	0	
ONEFILEONLY	Logical	See Description	
OUT.FILE	Character		
OUTFILE	Character		
P.BIAS	Real		
PB1	Real		V
PB2	Real		V
PB3	Real		V
PB4	Real		V
PB5	Real		V
PB6	Real		V

Parameter	Type	Default	Units
PB7	Real		V
PB8	Real		V
PIEZSCALE	Real	1.0	
POWER<n>	Real		Ω
POWERFINAL	Real		Ω
PREVIOUS	Logical	False	
PROJECT	Logical	False	
PULSE.WIDTH	Real		s
Q<n>	Real		C/μm
QFACTOR	Real	1.0 or previous value	
QFINAL	Real		C/μm
QSCV	Logical	False	
QSTEP	Real		C/μm
RAMPTIME	Real		s
RAMP.LIT	Logical	False	
RELATIVE	Logical	False	
SCAN.SPOT	Real		
S.OMEGA	Real	1.0	
SINUAMP.COMP	Real	0.0	
SINUVAR.COMP	Real	0.0	
SOR	Logical	False	
SPECTRUM	Character		
SQPULSE	Logical	False	
SS.LIGHT	Real	0.001	W/cm^2
SS.PHOT	Logical	False	
T <n>	Real		K
T.COMP	Real	0	K
T.SAVE	Real	0.0	s
TABLE	Character		
TDELAY	Real	0.0	s
TFALL	Real	0.0	s

Parameter	Type	Default	Units
TRISE	Real	0.0	s
TSAVE.MULT	Real	1.0	
TERMFINAL	Real		K
TERMINAL	Integer	all contacts	
TOLERANCE	Real	1.0×10^{-5}	
TRANS.ANALY	Logical	False	
TSTOP	Real		
TWOFILESONLY	Logical	False	
V<n>	Real		V
VFINAL	Real		V
VSTEP	Real	0.0	V
VSS	Real	0.1	V
WAVE<n>	Logical	False	
WAVEFORMS	Character		

Description

Each SOLVE statement must specify an initial bias condition. Once any DC condition has been solved, either a transient or AC analysis may be performed. You may also solve for carrier generation due to incident light under DC, or AC analysis transient conditions.

dc is one or more of the DC bias parameters

fp is one or more of the file parameters

ep is one or more of the initial guess or estimate parameters. Estimate parameters are used to specify how the initial approximation for the solution is to be obtained.

tp is one or more of the transient parameters. These parameters are used to specify data for transient analysis.

ac is one or more of the AC parameters. AC parameters are used to specify data for AC analysis.

ion is a set of the ionization integral parameters.

photo is one or more of the photogeneration parameters. Photogeneration parameters are used to specify illumination data.

therm is one or more of the thermal parameters. Thermal parameters are used for obtaining solutions in THERMAL3D.

DC Parameters

C.IMAX specifies the maximum current for curve tracing.

C.IMIN specifies the minimum current for curve tracing.

C.VSTEP specifies the initial voltage step for curve tracing.

C.VMAX specifies the maximum voltage for curve tracing.

C.VMIN specifies the minimum voltage for curve tracing.

CONTINUE is an alias for CURVETRACE.

CURVETRACE initiates curve tracing. See also the CURVETRACE statement. The alias for this parameter is CONTINUE.

ELECTRODE specifies electrodes that you wish to add voltage and current increments (VSTEP and ISTEP) to. If n electrodes are to be stepped, ELECTRODE should be an n-digit integer, where each of the digits is a separate electrode number. See also the NAME parameter.

I<name> or **I(<name>)** specifies the applied current for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel, and ground. No other user- defined names are allowed. This parameter is used when current boundary conditions are selected (see Section 19.4: "CONTACT")

I<n> specifies the terminal current for electrode, n. This parameter is used when current boundary conditions are selected (see Section 19.4: "CONTACT"). Normally, I defaults to the current from the previous bias point.It is more usual to use electrode names rather than numbers. This parameter is superseded by I<name>.

IFINAL specifies the final current value for a set of bias increments. If IFINAL is specified, either ISTEP or NSTEPS must be specified.

IMULT specifies that the current (for current boundary conditions) be multiplied by ISTEP rather than incremented.

ISTEP specifies a current increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If ISTEP is specified, either IFINAL or NSTEPS must also be specified.

N.BIAS specifies fixed electron quasi-Fermi potentials if electron continuity is not being solved. If N.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. But if FIX.QF is set in the METHOD statement, the quasi-Fermi levels will be set to the maximum bias.

NB<n> allows region by region specification of N.BIAS. The n index corresponds to the region index for which the specified value of electron quasi-fermi level applies.

NAME specifies that the named electrode is to be ramped. Custom electrode names are supported by name. See also the V<name> parameter.

NSTEPS specifies the number of DC bias increments.

P.BIAS specifies fixed hole quasi-Fermi potentials if hole continuity is not being solved. If P.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. But if FIX.QF is set in the METHOD statement, the quasi-Fermi levels will be set to the minimum bias.

PB<n> allows region by region specification of P.BIAS. The n index corresponds to the region index for which the specified value of hole quasi-fermi level applies.

Q<name> or **Q(<name>)** specifies the charge on a named electrode. These names are as follows: gate, gg, fgate, cgate, ngate, and pgate. No other user-defined names are allowed.This parameter is used when floating or charge boundary conditions are selected (see Section 19.4: "CONTACT").

Q<n> specifies the charge on electrode number, n. It is more usual to use electrode names rather than numbers. This parameter is superseded by Q<name>.

QFINAL specifies the final charge for a set of bias increments. If QFINAL is specified, either QSTEP or NSTEPS must also be specified.

QSCV specifies the Quasi-static Capacitance calculation. This only has meaning when the SOLVE statement is ramping a bias. This requires a small voltage increment between solutions for best results.

QSTEP specifies a charge increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If QSTEP is specified, either QFINAL or NSTEPS must also be specified.

V<name> or **V(<name>)** specifies the bias voltage for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel, and ground. No other user-defined names are allowed.

V<n> specifies the bias voltage for electrode, n. Normally, Vn, defaults to the potential from the previous bias point. It is more usual to use electrode names rather than numbers. This parameter is superseded by V<name>.

VFINAL specifies the final voltage for a set of bias increments. If VFINAL is specified, either VSTEP or NSTEPS must also be specified.

VSTEP specifies a voltage increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If VSTEP is specified, you must either specify VFINAL or NSTEPS.

File Output Parameters

EMIN, **EMAX** specify the energy range for output of spectral data in the file named by the SPECTRUM parameter.

LMIN, **LMAX** specify the wavelength range for spectral output in the file named by the SPECTRUM parameter.

MASTER specifies that the output file selected by the OUTFILE parameter will be written in a standard structure file rather than in the older PISCES-II binary format.

ONEFILEONLY specifies that only one filename will be used to save solutions during a bias ramp. If this parameter is specified, filename incrementing described under OUTFILE is not applied.This parameter is true by default unless the NAME or ELECTRODE parameters or transient simulations specified. When CURVETRACE is used, you need to explicitly set this parameter to false using ^ONEFILEONLY to save a different filename at each bias point.

OUT.FILE is an alias for OUTFILE.

OUTFILE specifies the name of the output file where bias point solution information will be stored. If an electrode is stepped so that more than one solution is generated by the SOLVE statement, the ASCII code of the last non-blank character of the supplied file name will be incremented by one for each bias point in succession, resulting in a unique file per bias point. The alias for this parameter is OUT.FILE.

SPECTRUM specifies the name of an output file where LED spectral data are stored after each step.

Note: The output file specified by OUTFILE has a limit of 132 characters on a UNIX system and eight characters on a PC system.

Initial Guess Parameters

INITIAL sets all voltages to zero. If this parameter is not specified for the first bias point in a given structure a SOLVE INIT statement is automatically inserted. The SOLVE INIT statement is always solved in zero carrier mode with no external elements attached.It is used to provide a good initial guess to subsequent solutions.

LOCAL specifies that the initial approximation should use local values of quasi-Fermi levels. See Chapter 18: "Numerical Techniques", Section 18.6: "Initial Guess Strategies" for more detailed information.

MLOCAL specifies the modified local initial guess. This parameter is used when solved for carrier temperatures in the energy balance models. If any energy balance model is specified, MLOCAL defaults to true.

NOCURRENT specifies an initial guess strategy that solves the non-linear Poisson equation and uses this solution as the initial guess. This will typically give good results for situations where the quasi-fermi levels are almost flat and the current is consequently low. Under these conditions, you can solve almost any desired bias point directly.

PREVIOUS specifies that the previous solution as the initial approximation.

PROJ specifies that an extrapolation from the last two solutions will be used as an initial approximation. This parameter may be used if there are two or more existing solutions and equivalent bias steps are taken on any electrodes that are changed.

Note: If no initial guess parameters are specified, ATLAS will use PROJ wherever possible.

QFACTOR specifies as scaling factor to improve initial guesses when the QUANTUM model is used. This parameter should be ramped slowly from zero to unity at the start of quantum effect simulations.

Compliance Parameters

Compliance parameters define a current limit for a DC bias ramp or transient simulation. You can terminate the simulation by monitoring the current and checking against a user-defined limit.

COMPLIANCE sets a limit on the current from the electrode which has been specified by the CNAME or E.COMPLIANCE parameter. When the COMPLIANCE value is reached, any bias ramp is stopped and the program continues with the next line of the input file. The COMPLIANCE parameter is normally specified in A. If the GRAD parameter is specified, COMPLIANCE is specified in A/V.

E.COMPLIANCE specifies the electrode number to be used by the COMPLIANCE parameter. See also CNAME.

CNAME specifies the name of the electrode used by the COMPLIANCE.

GRAD specifies that the compliance value is a current/voltage gradient, and not a current value.

Transient Parameters

CYCLES specifies the number of periods to be simulated (both FREQUENCY and TRANS.ANALY must be specified when this parameter is used). The synonym for this parameter is PERIODS.

CYCLIC.BIAS specifies that a cyclic bias [157] simulation is being performed. Cyclic biasing allows the effects of repeated trapezoidal/square transient pulses (see SQPULSE) on the output characteristics to be determined, without having to do the extremely large numbers of cycles. When CYCLIC.BIAS is specified, ATLAS will calculate a series of trapezoidal/square pulses using the SQPULSE parameters. At the end of the third cycle (and at all subsequent cycles) the values of the potential (V), electron concentration (n), hole concentration (p) and trap probability of occupation (f_T) will be modified according to the equation:

$$x(k+1) = x(k) - \mathrm{CYLIC\cdot RELAX}*((x(k)-x(k-1))*dx(k)/(dx(h)-dx(k-1)) \qquad 19\text{-}5$$

where x is a parameter such as V, n, p or f_T at every node point, x(k) is the updated value calculated from the equation, k is the cycle number (with k+1 being the new update for parameter x), CYCLIC.RELAX is the relaxation factor, δx(k) is the difference between the simulated values of x at the start and beginning of the current cycle, while δx(k-1) is the equivalent for the previous cycle.

Steady state cyclic convergence is determined by comparing the normalizing sum of the updated values of V, n, p, and fT with a tolerance value (CYCLIC.TOL).

CYCLIC.RELAX specifies the **CYCLIC.BIAS** relaxation factor (the recommended range for this parameter to ensure stable convergence is between 0.2 and 1).

CYCLIC.TOL specifies the **CYCLIC.BIAS** tolerance factor.

DECAY specifies the time constant used for defining an exponential change in bias for transient simulation.

ENDRAMP applies any bias changes as linear ramps. ENDRAMP specifies the exact end of the ramp in running time (i.e., the ramp will start at $t=t_0$ and end at t=ENDRAMP).

NSTEPS can be used to signal the end of the run (i.e., the final time would be $t = t$ + NSTEPS*DT). You can use this parameter instead of the TSTOP parameter.

PULSE.WIDTH specifies the time constant used for sinusoidal non-linear time domain simulation. If SQPULSE is specified, PULSE.WIDTH is the width of the trapezoidal/square pulse not including the rise and fall times.

RAMPTIME applies any bias changes as linear ramps. RAMPTIME specifies a ramp interval in seconds (i.e., the ramp will begin at $t=t_0$ and ends at $t=t_0$ +RAMPTIME).

RELATIVE specifies that the TSTOP value is relative to the current time value.

SINUAMP.COMP specifies that sinusoidal amplitude compliance will be used. The transient simulation will stop if the amplitude of a sinusoidal waveform is less than the value of SINUAMP.COMP.

SINUVAR.COMP specifies that sinusoidal variance compliance will be used. The transient simulation will stop if the change in amplitude of a sinusoidal waveform is less than SINUVAR.COMP.

SQPULSE specifies that multiple trapezoidal/square transient pulses will be applied. The pulse is controlled by the TDELAY, TRISE, PULSE.WIDTH, TFALL, and FREQUENCY parameters.

T.COMP specifies a temperature for temperature compliance. Once the specified temperature is obtained at some location on the mesh the solution process is discontinued.

T.SAVE specifies a time increment at which the device structure files are saved. Note: Actual time steps may not correspond to the user-specified increment.

TABLE specifies the filename of a table of time and bias values for the specified electrode. The following format must be used for the TABLE file.

```
time1 bias1
time2 bias2
time3 bias3
....  ....
end
```

The time value should be in seconds. The value of the bias should be in the same units as displayed in ATLAS (i.e., Volts for a voltage controlled electrode, A/um for a current controlled electrode, and A for a current controlled electrode in DEVICE3D or ATLAS with the WIDTH parameter specified on the MESH statement). There's no limit to the number of time/bias pairs that can be defined in the table file. The keyword, END, must be placed after the last time/bias pair. Linear interpolation will be used to determine the bias for time points that aren't specified in the table file.

Note: An electrode must also be specified when using the TABLE parameter. The electrode can be specified either by name (e.g., NAME=anode) or by number (e.g., ELECTRODE=1).

TDELAY specifies the time delay before the first cycle of multiple trapezoidal/square transient pulses (SQPULSE) will be applied.

TFALL specifies the fall time for trapezoidal/square transient pulses (SQPULSE).

TRISE specifies the rise time for trapezoidal/square transient pulses (SQPULSE).

TRANS.ANALY specifies that transient analysis is being performed and that a sinusoid with a frequency, specified by the FREQUENCY parameter, should be applied.

TSAVE.MULT specifies a multiplier the save time increment, T.SAVE, is multiplied by after each time a structure is saved.

TSTEP or **DT** specifies the time-step to be used. For automatic time-step runs, DT is used to select only the first time step (see Section 19.27: "METHOD").

TSTOP specifies the end of the time interval. If simulation begins at $t=t_0$ and RELATIVE is specified, it will end at $t=TSTOP+t_0$.

WAVE<n> specifies that the WAVEFORM, n (where n is between 1 and 10), will be enabled for the current SOLVE statement. This parameter should appear on every SOLVE statement where the specified WAVEFORM is to be applied.

Note: You can solve more than one WAVEFORM at the same time.

WAVEFORMS specifies multiple WAVEFORMs to be enabled for the current SOLVE statement. The WAVEFORM numbers should be specified as a comma separated list. For example, SOLVE WAVEFORMS= "1, 2, 12" TFINAL=1e-6. This will enable WAVEFORM numbers 1, 2, and 12.

AC Parameters

AC.ANALYSIS specifies that AC sinusoidal small-signal analysis should be performed after solving for the DC condition. The full Newton method must be used for this analysis. This is typically specified with the statement:

```
METHOD NEWTON CARRIERS=2
```

ANAME specifies the name of the electrode to which the AC bias will be applied. See also TERMINAL.If no ANAME is specified, all electrodes have AC bias applied in turn.

AUTO selects an automatic AC analysis method. This method initially uses SOR. The DIRECT method will be used if convergence problems occur. We strongly recommend the use of the AUTO parameter.

DIRECT selects the direct AC solution method. This method is robust at all frequencies, but slow.

FREQUENCY specifies the AC analysis frequency. Analysis may be repeated at a number of different frequencies (without solving for the DC condition again) by specifying FSTEP. FREQUENCY can also be used to specify a sinusoidal or square pulse frequency for transient simulations.

FSTEP specifies a frequency increment which is added to the previous frequency. If MULT.FREQ is specified, the frequency will be multiplied by FSTEP.

MAX.INNER specifies the maximum number of SOR iterations.

MULT.FREQ specifies that the frequency will be multiplied by FSTEP, instead of added to FSTEP.

NFSTEPS specifies the number of times that the frequency is to be incremented by FSTEP.

S.OMEGA specifies the SOR parameter. This parameter is not the AC frequency.

SOR selects the SOR AC solution method. Although SOR is fast, it should only be used when you are performing simulations at low frequencies. Low frequency can be defined here as at least an order of magnitude below the cutoff frequency.

TERMINAL specifies the electrode number to which the AC bias will be applied. Although more than one contact number may be specified (via concatenation), each will be solved separately. Each contact that is specified generates a column of the admittance matrix. If no TERMINAL is specified, all electrodes have AC bias applied in turn. See also ANAME.

TOLERANCE specifies SOR convergence criterion.

VSS specifies the magnitude of the applied small-signal bias. The approach used for small-signal analysis constructs a linear problem based on derivatives calculated during Newton iterations, so adjusting VSS will generally not affect the results.

NOISE Parameters

BRANIN specifies that Branin's method should be used in the calculation of the impedance field. The direct method (which is the alternative to Branin's method) is very slow. Therefore, we recommend that BRANIN is never set to False.

NOINTERFACE specifies that any element which is adjacent to an interface should be ignored in the noise calculation.

NOCONTACT specifies that any element which is adjacent to a contact should be ignored in the noise calculation.

NOISE.SS specifies that a small-signal noise simulation should be performed.

Direct Tunneling Parameters

An alternative set of direct quantum tunneling models for calculating gate current can be set using DT.CUR on the SOLVE statement. If you set this parameter, then select the particular model using DT.METH. The allowed values of DT.METH are 1, 2 and 3. The default value of DT.METH is 1.

DT.METH=1 selects a Fowler-Nordheim type model, with a correction for the trazepoidal, rather than the triangular shape of the potential barrier.

DT.METH=2 selects a WKB model with the assumption of a linearly varying potential. It involves a calculation of the classical turning points and integration between those limits.

DT.METH=3 selects an exact formula for a trapezoidal barrier that involves Airy function solutions.

You can also need to specify the type of tunneling taking place by using at least one of the following flags.

DT.CBET specifies electron tunneling from conduction band to conduction band.

DT.VBHT specifies holes tunneling from valence band to valence band.

DT.VBET specifies Band-to-Band tunneling.

Note: Specifying DT.CUR on the SOLVE statement invokes the post-processing version of Direct tunneling calculation of Gate current. Specifying the same parameters on the MODELS statement enables a self-consistent version.

Ionization Integral Parameters

Ionization parameters are used to calculate ionization integrals. No calculation will take place unless the IONIZINT parameter is specified.

DELTAV since the electric field can be near zero at electrodes, the electric field line calculations begin at a distance from the electrode. A potential contour is drawn around the electrode at a distance where the potential is DELTAV less than the applied bias on the contact. Defaults to 0.1 V, but will typically need to be increased especially for power devices and heavily doped contact regions.

E.CRIT specifies the minimum electric field used to calculate integration integrals. Field lines will terminate when the field drops below E.CRIT.

INAME specifies the electrode name from which the electric field lines are calculated.The default is to use the same as electrode as specified in NAME.

ION.CRIT specifies the critical value of the ionization integral used by IONSTOP to terminate the simulation. When the critical value is reached, any bias ramp will be terminated and the next line of the input file will be executed.

ION.ELEC specifies the electrode from which the electric field lines are calculated. This parameter defaults to the electrode which is being ramped (if any).

IONIZINT enables the calculation of ionization integrals along electric field lines.

IONLINES specifies the number of electric field lines to be calculated. Ionization integrals are calculated along each line.

IONSTOP stops the bias ramp if integral is greater than 1.0

LRATIO specifies the ratio between the starting points of the electric field lines. An LRATIO value of between 0.5 and 1.5 is recommended. LRATIO=1 means that the spacing between the starting points of the electric field lines is equal. LRATIO < 1 means more lines start towards the left hand side of the structure. LRATIO > 1 means more lines start towards the right hand side.

Photogeneration Parameters

All these parameters require LUMINOUS to be licensed for correct operation.

AR.INDEX specifies the index of refraction for an anti-reflective coating.

AR.THICKNESS specifies the thickness in microns of an anti-reflective coating.

B<*n*> specifies the optical spot power associated with optical beam number, n. The beam number must be an integer from 1 to 10.

BEAM specifies the beam number of the optical beam when AC photogeneration analysis is performed. Unlike the ELECTRODE parameter, this parameter can only be used to specify a single beam.

INDEX.CHECK specifies that the real and imaginary refractive indices used in the ray tracing of each beam will be printed to the run-time output. This parameter can be used as confirmation of the input of user-defined refractive indices or to check the default parameters. The indices are only printed when you perform the ray trace at the first reference to a given beam on the SOLVE statement.

L.WAVE specifies the luminous wavelength to use to calculate the luminous power estimate for radiative recombination. If a positive value of L.WAVE is specified the luminous power will be saved in any log file specified for that solution.

LAMBDA1 specifies the wavelength of the optical source (beam) number 1 for this solution. This parameter can be used to perform analysis of spectral response as a function of wavelength.

LIT.STEP selects the light intensity increment of all optical beams which have been specified. This parameter is used when light intensity varies by stepping (similar to the VSTEP parameter). If LIT.STEP is specified, the NSTEP parameter should be used to select the number of steps.

RAMP.LIT specifies that the light intensity is to be ramped when transient simulations are performed. If RAMP.LIT is specified, transient mode parameters such as RAMPTIME, TSTEP, and TSTOP must also be specified. The RAMP.LIT parameter affects all specified optical beams (i.e., all beams are ramped).

SCAN.SPOT specifies a beam number for spot scanning. Spot scanning requires you to specify the RAYS parameter of the BEAM statement. With this specification, the incident light is split into the user-specified number of rays. During the spot scan, solutions are obtained with the beam energy applied to each of the rays in sequence.

SS.LIGHT specifies the intensity in the small signal part of the optical beam when small signal AC photogeneration analysis is performed.

SS.PHOT specifies that small signal AC analysis will be performed on the optical beam selected by the BEAM parameter. When SS.PHOT is specified, other AC parameters (e.g., FREQUENCY, FSTEP, MAX.INNER, MULT.FREQ, NFSTEPS, S.OMEGA, and TOLERANCE) should also be specified. You don't need to specify the AC.ANALYSIS parameter when performing small signal AC analysis on optical beams.

Thermal3D Parameters

POWER<n> specifies the power in watts on the n-th heat source.

POWERFINAL specifies the final power for a linearly ramped heat source.

T<n> specifies the temperature in K on the n-th heat sink.

TEMPFINAL specifies the final temperature for a linearly ramped heat sink.

For more information about these parameters, see also Chapter 17: "Thermal 3D: Thermal Packaging Simulator", Section 17.4: "Obtaining Solutions In THERMAL3D".

DC Conditions Example

The following statement solves for a defined bias point and saves the solution to output file, *OutA*. The voltages on electrodes other than the gate will keep the value from the previous SOLVE statement.

```
SOLVE VGATE=0.1 OUTFILE=OutA
```

Bias Stepping Example

In the next example, the bias stepping is illustrated. The two SOLVE statements produce the following bias conditions:

Bias Point	Vgate	Vdrain	Vsub
1	0.0	0.5	-0.5
2	1.0	0.5	-0.5
3	2.0	0.5	-0.5
4	3.0	0.5	-0.5
5	4.0	0.5	-0.5
6	5.0	0.5	-0.5

The solutions for these bias points will be saved to the files: OUT1, OUTA, OUTB, OUTC, OUTD, and OUTE. The initial approximation for each bias point is obtained directly from the preceding solution.

For bias points 4, 5, and 6, the program will use a PROJ to obtain an initial approximation. Since, starting with bias point 4, both of its preceding solutions (bias points 2 and 3) only had the same electrode bias (number 1) altered.

```
SOLVE Vdrain=.5 Vsub=-.5 OUTF=OUT1

SOLVE Vgate=1 VSTEP=1 VFINAL=5  NAME=gate OUTF=OUTA
```

Transient Simulation Example

The following sequence is an example of a time dependent solution. The METHOD statement specifies second-order discretization, automatic time-step selection, and an automated Newton-Richardson procedure.

The first SOLVE statement then computes the solution for a device with 1V on the base electrode and 2V on the collector in steady-state. The second SOLVE statement specifies that the base electrode is to be ramped to 2V over a period of 10 ns and is left on until 25 ns. Each solution is written to a file. The name of the file is incremented in a manner similar to that described for a DC simulation (UP1, UP2, and so on). Note that an initial time step had to be specified in this statement.

The third SOLVE statement ramps ther base down from 2V to - 0.5V in 20 ns (end of ramp is at =45 ns). The device is then solved at this bias for another 55 ns (out to 100 ns). Each solution is again saved in a separate file (DOWN1, DOWN2, and so on).

No initial timestep was required since one had been estimated from the last transient solution from the previous SOLVE statement.

Finally, the fourth SOLVE statement performs the steady-state solution at Vbe = -0.5V and Vce = 2V.

```
METHOD 2ND TAUTO AUTONR

SOLVE  Vbase=1 Vcollector=2

SOLVE  Vbase=2 DT=1E-12 TSTOP=25E-9 RAMPTIME=10E-9 OUTF=UP1

SOLVE  Vbase= -0.5 TSTOP=100E-9 RAMPTIME=20E-9 OUTF=DOWN1

SOLVE  Vbase= -0.5 Vcollector=2
```

AC Analysis Example

The following example illustrates an application of the SOLVE statement for AC analysis. It is assumed that the device has three electrodes. This analysis is being performed after DC conditions are solved at V1 = 0V, 0.5V, 1.0V, 1.5V, and 2.0V. A series of 10mV AC signals with frequencies of 1 MHz, 10 MHz, 100 MHz, 1 GHz, 10 GHz, and 100 GHz are applied to each electrode in the device.

Ninety AC solutions will be performed (5 X 6 X 3 = 90).

```
SOLVE V1=0 V2=0 V3=0 VSTEP=0.5 NSTEPS=4 ELECT=1 \

AC FREQ=1E6 FSTEP=10 MULT.F NFSTEP=5 VSS=0.01
```

Photogeneration Examples

The following statement simulates two DC optical beams incident on the device with optical spot powers of 15 and 25 W/cm^2.

```
SOLVE B1=15 B3=25
```

The next example shows how DC spot power of the two optical beams can be stepped simultaneously. Beam #1 will increase to 35 W/cm^2 and beam #3 will increase to 45 W/cm^2.

```
SOLVE LIT.STEP=5.0 NSTEP=4
```

In the next example, the beams are ramped in the time domain. Beam #1 is ramped down to 0 W/cm^2 and beam #3 is ramped up to 100 W/cm^2. The duration of the ramp is 1 ns. After the ramp, simulation will continue for 4 ns.

```
SOLVE B1=0 B3=100 RAMP.LIT TSTEP=2E-11 RAMPTIME=1E-9 TSTOP=5E-9
```

Next, the small signal response of a single beam is analyzed. First, ATLAS will solve the DC characteristics at the specified optical spot powers. Then, the AC response of beam #1 will be calculated at a frequency of 10 MHz.

```
SOLVE B1=10 B3=20 BEAM=1 FREQUENCY=1e7 SS.PHOT SS.LIGHT=0.01
```

Finally, frequency stepping is used to look at the small signal AC frequency response of one of the beams. AC response is calculated at frequencies from 1kHz to 100MHz at each decade. The MULT.F parameter is used to geometrically increase the frequency.

```
SOLVE B1=10 B3=5 BEAM=1 SS.PHOT SS.LIGHT=0.01 \
      MULT.F FREQUENCY=1E3 FSTEP=10 NFSTEP=6
```

Ionization Integral Example

Ionization integrals are used to estimate the breakdown voltage from analysis of the electric field. They can be used in zero carrier mode providing much faster simulation that conventional breakdown analysis. The ionization syntax uses the parameter IONIZ to enable the ionization integral calculation. An equipotential contour is calculated at a potential DELTAV from the contact NAME (or INAME). Electric field lines are started at distances along this potential contour. IONLINES sets the number of lines and LRATIO is the ratio of the distance between the starting points of the lines. The following syntax is for a PMOS transistor. See the on-line examples for other cases using ionization integrals.

```
IMPACT SELB

METHOD CARR=0

SOLVE VDRAIN=-1 VSTEP=-1 VFINAL=-20.0 NAME=DRAIN \

IONIZ IONLINES=50 LRATIO=0.9 DELTAV=1.2
```

Note: There are over 300 on-line examples supplied with ATLAS to provide examples of sequences of SOLVE statements applied to practical problems for a variety of device technologies.

19.45: SPX.MESH, SPY.MESH

S<n>.MESH specifies the location of grid lines along the <n>-axis in a rectangular mesh used in self-consistent Schrodinger-Poisson Model simulation (see Chapter 13: "Quantum: Quantum Effect Simulator", Section 13.2: "Self-Consistent Coupled Schrodinger Poisson Model"). The syntax is equivalent for x and y directions.

Syntax

```
SPX.MESH NODE=<n> LOCATION=<n>

SPX.MESH SPACING=<n> LOCATION=<n>

SPY.MESH NODE=<n> LOCATION=<n>

SPY.MESH SPACING=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
SPACING	Real		μm

Description

NODE specifies the mesh line index. These mesh lines are assigned consecutively.

LOCATION specifies the location of the grid line.

SPACING specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, then the NODE parameter should not be specified.

Note: The mesh defined in these statements for the Self-Consistent Schrodinger-Poisson Model is entirely separate from the electrical device simulation mesh defined on the MESH statement.

Setting Locally Fine Grids Example

This example shows how to space grid lines closely around a topological feature such as a junction at y=0.85 microns.

```
SPY.MESH  LOC=0.0  SPAC=0.2

SPY.MESH  LOC=0.85 SPAC=0.01

SPY.MESH  LOC=2    SPAC=0.35
```

19.46: SYMBOLIC

Note: In versions 3.0 and greater, the SYMBOLIC statement is no longer needed. All functions have been moved to the METHOD statement.

19.47: SPREAD

SPREAD distorts rectangular grids defined by ATLAS in the vertical direction to follow surface and junction contours.

Note: The use of this parameter is not recommended by SILVACO.

Syntax

```
SPREAD LEFT|RIGHT WIDTH=<r> UPPER=<i> LOWER=<i>
Y.LOWER|THICKNESS [<options>]
```

Parameter	Type	Default	Units
ENCROACH	Real	1.0	
GRADING	Real	1.0	
GR1	Real	1.0	
GR2	Real	1.0	
LEFT	Logical	False	
LOWER	Integer		
MIDDLE	Integer	Halfway between UPPER and LOWER	
RIGHT	Logical	False	
THICKNESS	Real		μm
UPPER	Integer		
VOL.RATIO	Real	0.44	
WIDTH	Real		μm
Y.LOWER	Real		μm
Y.MIDDLE	Real	0.50	μm

Description

SPREAD can reduce the grid complexity of specific simulations. Since the SPREAD statement is somewhat complicated, we suggest that you follow the supplied examples carefully until you're confident to understand the workings of this statement.

Mandatory Parameters

LEFT distorts the left side of the grid. If LEFT is specified, RIGHT must not be specified.

LOWER specifies the lower y-grid line above which distortion will take place.

RIGHT distorts the right side of the grid. If RIGHT is specified, LEFT must not be specified.

THICKNESS specifies the thickness of the distorted region. Unless VOL.RATIO is set to 0 or 1, THICKNESS will usually move the positions of both the UPPER and LOWER grid lines. The Y.LOWER and

THICKNESS parameters define the distorted grid region. Only one of these parameters should be specified.

UPPER specifies the upper y-grid line under which distortion will take place.

WIDTH specifies the width from the left or right edge (depending on whether LEFT or RIGHT is selected) of the distorted area. The actual x coordinate specified by WIDTH (min[*x*]+ WIDTH for LEFT, max[*x*] + WIDTH for RIGHT) will lie in the middle of the transition region between distorted and undistorted grid regions.

Y.LOWER specifies the physical location in the distorted region to which the line specified by LOWER will be moved. The line specified by UPPER is not moved. The Y.LOWER and THICKNESS parameters define the distorted grid region. Only one of these parameters should be specified.

Optional Parameters

ENCROACH defines the abruptness of the transition between a distorted and non-distorted grid. The transition region becomes more abrupt with smaller ENCROACH factors (the minimum is 0.1).

Note: Depending on the characteristics of the undistorted grid, long, thin, or obtuse triangles may result if too low an ENCROACH value is used.

GRADING specifies a grid ratio which produces a non-uniform grid in the distorted region. This parameter is identical to the RATIO parameter in the X.MESH and Y.MESH statements.

GR1 and **GR2** may be used instead of the GRADING parameter. GR1 and GR2 may be specified in conjunction with MIDDLE (y grid line) and Y.MIDDLE (location) so that GR1 specifies grading in the spread region from UPPER to MIDDLE, and GR2 specifies grading from MIDDLE to LOWER.

MIDDLE specifies the y-grid line that serves as a boundary between the grading specified by GR1 and the grading specified by GR2.

VOL.RATIO specifies the ratio of the downward displacement of the lower grid line to the net increase in thickness. The default (0.44) should be used for oxide-silicon interfaces. VOL.RATIO is ignored if Y.LOWER is specified.

Y.MIDDLE specifies the physical location in the distorted grid to which the line specified by MIDDLE will be moved.

Examples

This example spreads a uniform 400 Å of oxide to 1000 Å on the left side of the device. This increases oxide thickness by 600 Å. Because VOL.RATIO is not specified, the default (0.44) is used. Therefore, 0.44 X 600 = 264 Å of the net increase will lie below the original 400 Å and 0.56 X 600 = 336 Å of the net increase will lie above the original 400 Å. The width of the spread region is 0.5 μm, and the oxide taper is quite gradual because of the high encroachment factor. The grid is left uniform in the spread region.

```
             # *** Mesh definition ***
MESH NX=30 NY=20 RECT
X.M N=1  L=0
X.M N=30 L=5
Y.M N=1  L=-.04
Y.M N=5  L=0
Y.M N=20 L=1 R=1.4
```

```
# *** Thin oxide ***
REGION IY.H=5 OXIDE NUM=1
# *** Silicon substrate ***
REGION IY.L=5 SILICON NUM=2
# *** Spread ***
SPREAD LEFT W=0.7 UP=1 LO=4 THICK=0.1 MID=2 Y.MID=0.05
```

In the second example, the right side of the grid is distorted in order to follow a junction contour. The initial grid is assumed to be above. Y.LOWER is used so that there is no increase in the size of the device, just grid redistribution. When Y.LOWER is set to the junction, choose the ENCROACH parameter so that the lower grid line (LOWER=10) follows the junction as closely as possible. The grid is graded so that grid lines are spaced closer together as they approach the junction. Because the point specified by WIDTH lies in the middle of the transition region, it should be chosen to be slightly larger than the width of the doping box.

```
# *** Doping ***
    DOPING UNIFORM N.TYPE CONC=1E15
    DOPING GAUSS P.TYPE X.LEFT=1.5 X.RIGHT=2 \
    PEAK= CONC=1.e19  RATIO=.75 JUNC=0.3
    # *** Spread ***
    SPREAD RIGHT W=0.7 UP=1 LO=4 THICK=0.3 MID=2 Y.MID=0.10
```

19.48: SYSTEM

SYSTEM allows execution of any UNIX command within an input file

---

Note: The SYSTEM statement is executed by DECKBUILD and is fully documented in the DECKBUILD USER'S MANUAL. To enable SYSTEM command, select **Category: Options** in the DeckBuild Main Control menu.

Examples

The following command will remove all files test*.str before a SOLVE statement where the OUTF parameter is used.

```
system \rm -rf test*.str

SOLVE  .... OUTF=test0
```

The system command and the UNIX commands are case sensitive.

UNIX commands may be concatenated on a single line using the semicolon (;) operator. For example to run a third party program that reads and writes Silvaco format files with fixed names input.str and output.str.

```
SAVE OUTF=mysave.str

system mv mysave.str input.str; source myprog.exe; mv output.str
myrestart.str

EXTRACT INIT INF=myrestart.str
```

The UNIX re-direct symbol > is not supported by the system command. The UNIX echo and sed syntax can be used instead to output values or variables to a given filename. For example to save the extracted value of variable $myvariable to the file myfile.

```
system echo $"myvariable" | sed -n " w myfile"
```

19.49: THERMCONTACT

THERMCONTACT specifies the position and properties of thermal contacts. This statement must be used when lattice heating solutions are specified using GIGA or GIGA3D.

Syntax

```
THERMCONTACT NUMBER=<n> <position> [EXT.TEMPER=<n>] [ALPHA=<n>]
```

Parameter	Type	Default	Units
ALPHA	Real	•	W/(cm2·K)
BOUNDARY	Logical	true	
DEVICE	Character		
ELEC.NUMBER	Integer		
EXT.TEMPER	Real	300	K
F.CONTEMP	Character		
NAME	Character		
NUMBER	Integer	1	
STRUCTURE	Character		
X.MAX	Real	Right side of structure	μm
X.MIN	Real	Left side of structure	μm
Y.MAX	Real	Bottom of structure	mμ
Y.MIN	Real	Top of structure	μm
Z.MIN	Real	Front	microns
Z.MAX	Real	Back	microns

Description

At least one THERMCONTACT statement must be specified when simulating lattice heating effects (MODELS LAT.TEMP). The THERMCONTACT statement must appear in the input deck before any METHOD statement.

position is a set of the position parameters described below. Use either the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters to specify the exact position of the contact or the ELEC.NUMBER parameter to specify an electrode number that coincides with the thermal contact.

NUMBER specifies a thermal contact number from 1 to 20. Contact numbers should be specified in increasing order. This parameters must be specified on all THERMCONTACT statements.

ALPHA specifies the reverse value of thermal resistance ($\alpha=1/R_{TH}$). This parameter may not be used if ELEC.NUMBER parameter has been specified.

EXT.TEMPER specifies the external temperature. The synonym for this parameter is TEMPERATURE.

F.CONTEMP specifies the name of a file containing a C-Interpreter function describing the contact temperature as a function of time.

Position Parameters

BOUNDARY specifies which parts of the thermal contact the thermal boundary conditions are applied at. See Chapter 7: "Giga: Self-Heating Simulator", Section 7.2.4: "Thermal Boundary Conditions" for a full description.

NAME specifies which region the THERMCONTACT statement applies to. Note that the name must match the name specified in the NAME parameter of the REGION statement.

DEVICE specifies which device in MIXEDMODE simulation the THERMCONTACT statement applies to. The synonym for this parameter is STRUCTURE.

ELEC.NUMBER specifies an electrode number that the thermal contact is coincident with.

STRUCTURE is a synonym for DEVICE.

X.MIN specifies the left edge of the contact.

X.MAX specifies the right edge of the contact.

Y.MIN specifies the top edge of the contact.

Y.MAX specifies the bottom edge of the contact.

Z.MIN specifies the location of the front edge of the thermal contact.

Z.MAX specifies the location of the rear edge of the thermal contact.

Coordinate Definition Example

A thermal contact is located where y coordinate values range from 10 μm to the bottom side of the structure and x coordinate values range from the left edge of the structure to the right edge of the structure (be default). The external temperature is set to 300K and a thermal resistance of 1 is added. Thus, the temperature at y = 10 μm will be greater than 300K once lattice heating effects occur.

```
THERMCONTACT NUM=1 Y.MIN=10 EXT.TEMP=300 ALPHA=1
```

Setting Thermal and Electrical Contacts Coincident Example

The next statement line creates a thermal contact at the location of electrode #4. An external temperature of 400K is specified.

```
THERMCONTACT NUM=2 ELEC.NUM=4 EXT.TEMP=400
```

Note: Location and Parameters of thermal contacts are not stored in the ATLAS solution files. Therefore, THERMCONTACT statements must be defined in each ATLAS run involving lattice heating.

19.50: TITLE

TITLE specifies the title (up to 29 characters) that will appear in the standard output. If used, the TITLE command should be the first statement in the ATLAS input file.

Syntax

```
TITLE <string>
```

Example

This example causes the text, *** CMOS p-channel device ***, to be printed at the top of all ATLAS printouts and screen displays.

```
TITLE *** CMOS p-channel device ***
```

Note: TITLE cannot be used with the automatic ATHENA to ATLAS interface feature of DECKBUILD

19.51: TONYPLOT

TONYPLOT starts the graphical post-processor TONYPLOT

Note: The TONYPLOT statement is executed by DECKBUILD and is fully described in the DECKBUILD USER'S MANUAL.

Examples

All graphics in ATLAS is performed by saving a file and loading the file into TONYPLOT. The TONYPLOT command causes ATLAS to automatically save a structure file and plot it in TONYPLOT. The TonyPlot window will appear displaying the material boundaries. Use the **Plot:Display** menu to see more graphics options.

This command will display the file, *myfile.str*.

```
tonyplot -st myfile.str
```

This command will overlay the results of *myfile1.str* and *myfile2.str*.

```
tonyplot -overlay myfile1.str myfile2.str
```

Note: For documentation of the extensive features of TONYPLOT for graphical display and analysis, see TONYPLOT USER'S MANUAL.

19.52: TRAP

TRAP activates bulk traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
TRAP DONOR|ACCEPTOR E.LEVEL=<r> DENSITY=<r> DEGEN=<v>
<capture parameters> REGION|NUMBER
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
DEVICE	Character		
DEGEN.FAC	Real	undefined	
DENSITY	Real		cm^{-3}
DONOR	Logical	False	
E.LEVEL	Real		eV
FAST	Logical	False	
F.DENSITY	Character		
MATERIAL	Character		
NUMBER	Integer	0	
REGION	Integer	0	
SIGN	Real		cm^2
SIGP	Real		cm^2
STRUCTURE	Character		
TAUN	Real		s
TAUP	Real		s
X.MIN	Real	Minimum X coord	µm
X.MAX	Real	Maximum X coord	µm
Y.MIN	Real	Minimum Y coord	µm
Y.MAX	Real	Maximum Y coord	µm
Z.MIN	Real	Minimum Z coord	µm
Z.MAX	Real	Maximum X coord	µm

Description

ACCEPTOR specifies an acceptor-type trap level.

DEGEN.FAC specifies the degeneracy factor of the trap level used to calculate the density.

DENSITY sets the maximum density of states of the trap level.

DEVICE specifies which device the statement applies to in MIXEDMODE simulation. The synonym for this parameter is STRUCTURE.

DONOR specifies a donor-type trap level.

E.LEVEL sets the energy of the discrete trap level. For acceptors, E.LEVEL is relative to the conduction band edge. For donors, it depends on the valence band edge.

F.DENSITY specifies the name of a file containing a C-Interpreter function describing the density of donor/acceptor traps as a function of position.

FAST specifies that the trap densities are static even during transient simulation (i.e., they reach equilibrium instantaneously).

MATERIAL specifies which material from the table in Appendix B: "Material Systems" will apply to the TRAP statement. If a material is specified, then all regions defined as being composed of that material will be affected.

NUMBER is the synonym for REGION.

REGION specifies which region the traps apply to. If unspecified, the traps apply to all regions.

STRUCTURE is a synonym for DEVICE.

X.MIN specifies the minimum X coordinate of a box where traps are to be applied.

X.MAX specifies the maximum X coordinate of a box where traps are to be applied.

Y.MIN specifies the minimum Y coordinate of a box where traps are to be applied.

Y.MAX specifies the maximum Y coordinate of a box where traps are to be applied.

Z.MIN specifies the minimum Z coordinate of a box where traps are to be applied.

Z.MAX specifies the maximum Z coordinate of a box where traps are to be applied.

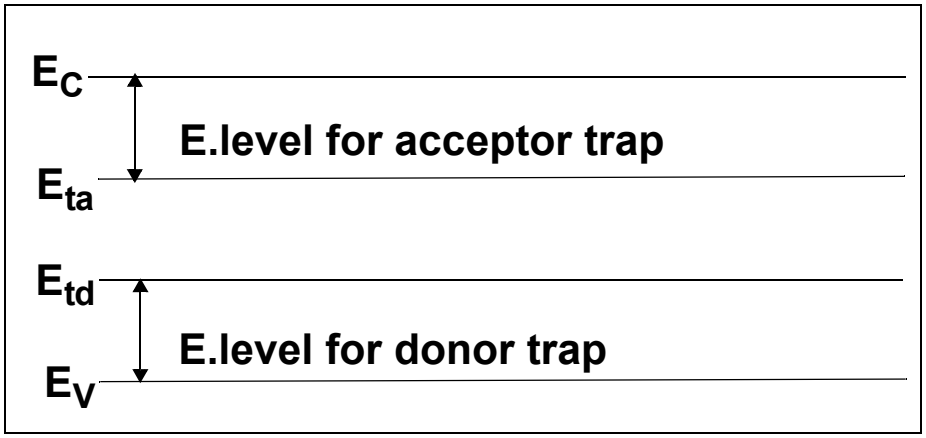


Figure 19-8: Acceptor and Donor Trap Energy Levels

Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN specifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level.

Multiple Trap Level Definition Example

The following example sets three discrete trap levels within the silicon bandgap. These trap levels will capture carriers, slowing the switching speed of any device. In this example the capture cross sections are used to define the properties of each trap.

```
trap e.level=0.49 acceptor density=2.e15 degen=12 \
sign=2.84e-15 sigp=2.84e-14
trap e.level=0.41 acceptor density=1.e15 degen=12 \
sign=7.24e-16 sigp=7.24e-15
trap e.level=0.32 donor density=1.e15 degen=1 \
sign=1.00e-16 sigp=1.00e-17
```

Note: For distributed trap levels, see Section 19.7: “DEFECTS”. For interface traps, see Section 19.19: “INTTRAP”. Also, spatial `TRAP` distributions in x, y or z directions must be defined in the `DOPING` statement.

19.53: UTMOST

UTMOST converts data from a ATLAS logfile format into an UTMOST logfile format.

Note: UTMOST 12.3.4 or later can read ATLAS format logfiles directly in batch mode. This statement is obsolete and its use is not recommended.

Syntax

```
UTMOST BIP|DIODE|MOS <input> OUTFILE=<fn> WIDTH=<n>
[<elec>] <cntrl>
[TRANSLATE INFILE1=<filename>][<analysis>] [<parasites>]
```

Parameter	Type	Default	Units
AC	Logical	False	
ANODE	Integer	1	
APPEND	Logical	False	
BASE	Integer		
BIP	Logical	False	
BULK	Integer		
CATHODE	Integer		
COLLECTOR	Integer		
DEVICE	Integer	1	
DIODE	Logical	False	
DRAIN	Integer		
EMITTER	Integer		
GATE	Integer		
INFILE	Character		
INTERNAL	Logical	False	
LENGTH	Real	1.0	μm
MESFET	Logical	False	
MINSTEP	Real	0.01	V
MOS	Logical	False	
OUTFILE	Character		
POLARITY	Integer	1 (n-type)	
ROUTINE	Integer	1	

Parameter	Type	Default	Units
SOURCE	Integer		
TEMPERATURE	Real	300	K
WELL	Integer		
WIDTH	Real	1.0	μm

Description

Specify a technology parameter (BIP, DIODE, or MOS), an input file (INFILE1=), and an output file (OUTFILE=). You can also specify one or more of the optional electrode parameters.

INFILE is used to convert up to nine ATLAS logfiles into a single UTMOST logfile. The ATLAS logfiles must be specified in the form:

```
INFILE1=<fn> INFILE2=<fn> INFILE3=<fn> . . .
```

where fn is the name of the logfile that you wish to convert.

OUTFILE specifies the name of a file where I-V data will be stored in an UTMOST file format.

APPEND specifies that I-V data should be appended to the output file specified by OUTFILE2.

WIDTH is used to specify the width of the device. Electrode current is multiplied by the value of WIDTH before being saved in the logfile.

Technology Parameters

BIP creates an UTMOST bipolar transistor log file.

DIODE creates an UTMOST diode log file.

MOS creates an UTMOST MOSFET log file.

MESFET creates an UTMOST MESFET log file.

Electrode Parameters

Different electrode parameters may be specified with different technologies.

BIP Technology

BASE specifies the base electrode number.

COLLECTOR specifies the collector electrode numb.

EMITTER specifies the emitter electrode numb.

POLARITY indicates the device type. Polarity=1 specifies npn. Polarity=-1 specifies pnp.

MOS And **MESFET** Technologies

Note: If you have used the NAME parameter of the ELECTRODE statement to assign standard electrode names (bulk, drain, gate, and source), you don't need to re-specify these names in the UTMOST statement.

BULK or **SUBSTRATE** specifies the bulk electrode number.

DRAIN specifies the drain electrode number.

GATE specifies the gate electrode number.

POLARITY indicates the device type. POLARITY=1 specifies nmos. POLARITY=-1 specifies pmos.

SOURCE specifies the source electrode number.

Control Parameters

The optional control parameters are used to specify what type of information will be converted to an UTMOST logfile.

AC specifies that input logfiles contain AC parameters and that the UTMOST routine numbers refer to the UTMOST capacitance routines.

DEVICE specifies the device (or row) number used to identify different devices in UTMOST. The synonym for this parameter is ROW.

INTERNAL specifies that internal contact voltage will be used instead of applied bias.

LENGTH specifies the length of the device.

MINSTEP specifies the minimum voltage step between data points.

TEMPERATURE specifies the simulation temperature.

ROUTINE specifies which UTMOST routine number data will be saved for. The following routines are supported.

BIP Technology

ROUTINE=1 specifies I_C vs. V_{CE} curves. Each input file holds I-V data for a solution with a fixed base current and V_{CE} stepped. (UTMOST IC/VCE routine).

ROUTINE=9 specifies a B_F vs. V_{BE} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BE} stepped. (UTMOST BF vs. IC routine)

ROUTINE=10 specifies a B_F vs. V_{BC} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BC} stepped. (UTMOST BR routine)

ROUTINE=14 specifies a I_C,I_B vs. V_{BE} (Gummel) plot. Each input file holds I-V data for a solution with constant V_{CE} and V_{BE} stepped. (UTMOST gummel routine)

ROUTINE=15 specifies a I_E,I_B vs. V_{BC} (Reverse Gummel) plot. Each input file holds I-V data for a solution with constants V_{EC} and V_{BC} stepped. (UTMOST rgummel routine)

ROUTINE=29 specifies I_E vs. V_{EC} plot. Each input file holds I-V data for a solution with a fixed base current and V_{EC} stepped. (UTMOST IE/VEC routine)

MOS Technology

ROUTINE=1 specifies a I_D vs. V_{DS} plot. V_B is constant and V_G is stepped. (UTMOST ID/VD-VG routine)

ROUTINE=2 specifies a I_D vs. V_{GS} plot. V_D is constant and V_B is stepped. (UTMOST ID/VG-VB routine)

ROUTINE=3 specifies the L_{EFF}, R_{SD} routine. Two devices of different lengths are needed. The I_D/V_{GS} data should be used.

ROUTINE=5 specifies a I_D vs. V_{GS} plot. V_D is constant and V_B=0. (UTMOST VTH routine)

ROUTINE=9 specifies a I_D vs. V_{GS} plot. V_D is constant and V_B is stepped. (UTMOST NSUB routine).

ROUTINE=10 specifies the IS routine. The forward diode characteristics of the drain to bulk junction are required.

ROUTINE=11 specifies the LAMBDA routine. An I_D/V_D curve is required.

ROUTINE=12 specifies the ETA routine. A set of I_D/V_{GS} data for different VDS is required.

ROUTINE=13 specifies the VMAX routine. A set of I_D/V_{GS} curves for small changes in V_{DS} are required.

ROUTINE=14 specifies the U0 routine. A set of I_D/V_{GS} data is required.

ROUTINE=26 specifies a I_D vs. V_{GS} plot. V_B is constant and V_D is stepped. (UTMOST ID/VG-VD routine).

MOS Capacitances

The following routines may be specified if both MOS and AC parameters are selected.

ROUTINE=1 specifies the CGSO routine. C_{GS} vs. V_{GS} data is required.

ROUTINE=2 specifies the CGDO routine. C_{GD} vs. V_{GS} data is required.

ROUTINE=5 specifies the CJ routine. C_{BD} vs. V_{DS} data is required.

ROUTINE=6 specifies the CJSW routine. C_{BD} vs. V_{DS} data is required.

ROUTINE=12 specifies the CJ/CJSW routine. C_{GD} vs. V_{GS} data is required for two different size drain areas.

Diode Technology

ROUTINE=1 specifies an I vs. V plot. V is stepped. (UTMOST ID vs. VD routine).

19.54: VCSEL

The VCSEL statement specifies certain VCSEL simulation parameters.

Syntax

```
VCSEL <parameters>
```

Parameter	Type	Default	Units
ABSORPTION	Logical	False	
CHECK	Logical	False	
EFINAL	Real		eV
EINIT	Real		eV
FCARRIER	Logical		
INDEX.BOTTOM	Real	1.0	
INDEX.TOP	Real	1.0	
ITMAX	Integer		
MAXCH	Real		
LOSSES	Real		
NEFF	Real		
NMODE	Integer	1	
NSPEC	Integer	100	
OMEGA	Real		
PERTURB	Logical	False	
PHOTON.ENERGY	Real		eV
PHOJ	Logical	False	
TAUSS	Real		s
TOLER	Real		
VCSEL.CHECK	Logical	False	
VCSEL.INCIDENCE	Integer	1	

Description

ABSORPTION enables the absorption loss model in VCSEL.

CHECK enables reflectivity test simulation of the VCSEL structure

EINIT, **EFINAL** specify the lower and upper photon energies in reflectivity test. The default values are EINIT=0.8 PHOTON.ENERGY and EFINAL=1.2 PHOTON.ENERGY.

INDEX.BOTTOM specifies the refractive index of the medium below the structure. The default value is 1.0.

INDEX.TOP specifies the refractive index of the medium above the structure. The defaul value is 1.0.

ITMAX sets maximum number of external VCSEL iterations during photon density calculation. The default value is 30.

MAXCH specifies the maximum allowed relative change in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.

LOSSES specifies additional losses in Chapter 9: "VCSEL Simulator", Equation 9-16.

NEFF specifies the effective refractive index in Chapter 9: "VCSEL Simulator", Equation 9-11.

NMODE specifies the number of transverse modes of interest. The default value is `NMODE=1`.

NSPEC specifies the number of sampling points between `EINIT` and `EFINAL` in reflectivity test. The default value is `NSPEC=100`.

OMEGA specifies the reference frequency (see Equation 9-2 in Chapter 9: "VCSEL Simulator"). The reference frequency can change after the reflectivity test to correspond to the resonant frequency of the structure.

PERTURB enables faster solution of the Helmholtz equation. When it is specified, the program solves the longitudinal wave equation only once. The perturbational approach is used to account for the refractive index changes.

PHOTON.ENERGY specifies the reference photon energy. The reference photon energy can change after reflectivity test to correspond to the resonant photon energy of the structure. This parameter is related to `OMEGA` parameter.

PROJ enables faster solution of the photon rate equations below threshold. You should disable this solution scheme when the bias reaches lasing threshold. To do this, specify `^PROJ` in a `VCSEL` statement.

TAUSS is an iteration parameter used in the calculation of photon densities. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.

TOLER sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in a faster but less accurate calculations.

VCSEL.CHECK same as `CHECK`.

VCSEL.INCIDENCE specifies the direction of light incident on the structure. `VCSEL.INCIDENCE=1` is the light incident from the top. `VCSEL.INCIDENCE=0` is the light incident from the bottom. `VCSEL.INCIDENCE=2` or `>2` means both directions of light incidence are considered. By default, light is incident from the top of the structure.

19.55: WAVEFORM

WAVEFORM defines transient waveforms, which can then be enabled by specifying WAVE1 . . WAVE10 on the SOLVE statement.

Syntax

```
WAVEFORM [<parameters>]
```

Parameter	Type	Default	Units
AMPLITUDE	Real	0	
CONTINUE	Logical	True	
DECAY	Real	0	
ELEC.NAME	Character		
ELEC.NUMBER	Real	0	
F.WAVE	Character		
FREQUENCY	Real	0	Hz
NUMBER	Real	0	
OFFSET	Logical	False	
PERIODS	Real		
PHASE	Real	0	Deg
SINUSOID	Logical	False	

Description

AMPLITUDE specifies the sinusoidal amplitude.

CONTINUE specifies that the waveform will continue (i.e. retain the phase) across mulitple SOLVE statements. This is the default.

DECAY specifies an optional decay value.

ELEC.NAME specifies the electrode name.

ELEC.NUMBER specifies the electrode number (either ELEC.NAME or ELEC.NUMBER must be specified).

F.WAVE specifies the C-Interpreter file. The function prototype is:

```
int workf(double time, double dt, double *bias)
```

where time is the transient time, dt is the time step and bias is the bias calculated by the C-Interpreter function.

FREQUENCY specifies the frequency.

NUMBER specifies the waveform number. This must be specified on each WAVEFORM statement.

OFFSET specifies that a phase offset is to be used. If you use a non-zero phase in a sinusoidal definition, then SOLVE statement will go immediately to this bias. For example, if PHASE=90, AMPLITUDE=1 and the DC bias is 1V, the transient SOLVE statement will start at 2V and vary from 2V to 0V. If you specify OFFSET, then the bias will vary from 1V to -1V.

PERIODS specifies the maximum number of sinusoidal periods (default is unlimited).

PHASE specifies the phase.

SINUSOID specifies that a sinusoidal waveform will be generated according to:

```
AMPLITUDE*exp(-time*DECAY)*sin(2*PI*(FREQUENCY*time+PHASE/360.0))
```

19.56: X.MESH, Y.MESH, Z.MESH

<n>.MESH specifies the location of grid lines along the <n>-axis in a rectangular mesh for 2-D or 3-D simulation.

Note: The commands are equivalent in the x, y or z directions.

Syntax

```
X.MESH LOCATION=<l> (NODE=<n> [RATIO=<r>])| SPACING=<v>
```

Parameter	Type	Default	Units
DEPTH	Real		μm
H1	Real		μm
H2	Real		μm
H3	Real		μm
LOCATION	Real		μm
N.SPACES	Integer		
NODE	Integer		
RATIO	Real	1	
SPACING	Real		μm
WIDTH	Real		μm
X.MAX	Real		μm
X.MIN	Real	0.0	μm
Y.MAX	Real		μm
Y.MIN	Real	0.0	μm

Description

DEPTH specifies the extent of a mesh section in the Y direction.

H1 specifies the spacing between consecutive mesh lines at the beginning of a mesh section.

H2 specifies the spacing between consecutive mesh lines at the end of a mesh section.

H3 specifies the spacing between consecutive mesh lines at the middle of a mesh section.

LOCATION specifies the location of the grid line.

N.SPACES specifies the number of mesh spacings within a given mesh section.

NODE specifies mesh line index. There is a limit of 120 mesh lines. These mesh lines must be assigned in increasing order.

RATIO specifies the ratio to use when interpolating grid lines between given locations. Spacing between adjacent grid lines will increase or decrease by the factor assigned to the RATIO parameter. A

RATIO value of between 0.667 and 1.5 is recommended. RATIO should not be used if SPACING is specified.

SPACING specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, the NODE and RATIO parameters should not be specified. If the SPACING parameter is used to specify mesh spacings, the NX, NY, and NZ parameters of the MESH statement should not be specified. When the mesh spacings are specified using the SPACING parameter, the mesh size will be calculated.

WIDTH specifies the extent of a mesh section in the X direction.

X.MIN/Y.MIN specify the location of the beginning of a given mesh section (should only be specified for the first section).

X.MAX/Y.MAX specify the location of the end of a given mesh section (should not be specified if DEPTH/WIDTH is specified).

Setting Fine Grid at A Junction Example

This example shows how to space grid lines closely around a junction at y=0.85 microns.

```
Y.MESH  LOC=0.0  SPAC=0.2

Y.MESH  LOC=0.85 SPAC=0.01

Y.MESH  LOC=2    SPAC=0.35
```

This page is intentionally left blank.

Appendix A: C-Interpreter Functions

A.1: Overview

ATLAS has a built-in C language interpreter that allows many of the models contained in ATLAS to be modified. To use the SILVACO C-INTERPRETER, (SCI), you must write C language functions containing analytic descriptions of the model. The C-INTERPRETER uses the ANSI standard definition of C. If you are not familiar with the C language, we suggest that you refer to any of the popular C language references such as the one written by Kernighan and Ritchie [83]. Additional information about the C-INTERPRETER can be found in the SILVACO C-INTERPRETER MANUAL.

The function arguments of the C-INTERPRETER functions and return values are fixed in ATLAS. Make sure that the arguments and the return values for the functions match those expected by ATLAS. To help you, this release of ATLAS includes a set of templates for the available functions. The C-INTERPRETER function template can be obtained by typing:

```
atlas -T filename
```

where `filename` is the name of a file where you want the template to be copied. You can also get the C-INTERPRETER function template by accessing the template file through DECKBUILD.

This template file should be copied and edited when implementing user-defined C-INTERPRETER functions. To use the C-INTERPRETER function, specify the file name containing this function in the appropriate ATLAS statement. The relevant statement names and parameters are listed in the template file. The template function prototypes are defined in the include file `template.in`.

The following example shows how the C-INTERPRETER function `munsat` can be used to describe velocity saturation for electrons.

First, examine the template file and find the template for `munsat`. The template should look something like this:

```
/*
* Electron velocity saturation model.
* Statement: MATERIAL
* Parameter: F.MUNSAT
*/
/* e          electric field (V/cm) */
/* v          saturation velocity (cm/s) */
/* mu0        low field mobility (cm^2/Vs) */
/* *mu        return: field dependent mobility (cm^2/Vs) */
/* *dmde      return: derivative of mu with e */


int munsat(double e, double v, double mu0, double *mu, double *dmde)
{
    return(0);                /* 0 - ok */
}
```

Here, we see that the function is passed the electric field, the saturation velocity and the low field mobility. The function returns the field dependent mobility and the derivative of mobility with respect to electric field.

Note: It is important to properly calculate and return derivative values when specified, since the convergence algorithms in ATLAS require these derivatives.

The return value is an error flag. In this case, it is set to zero indicating that the function was successful (0=OK, 1=fail).

In this example, the electron saturation velocity characteristic is changed to one, which is akin to that used for holes in silicon. You can specify this model by setting the B.ELE parameter to 1 but for example purposes, use the C-INTERPRETER function munsat instead. This will enable comparison between the built-in model and the C-INTERPRETER model.

To implement the model, two lines additional lines of C code in the body of the function are specified as follows:

```
/*
* Electron velocity saturation model.
* Statement: MATERIAL
* Parameter: F.MUNSAT
*/
/* e          electric field (V/cm) */
/* v          saturation velocity (cm/s) */
/* mu0        low field mobility (cm^2/Vs) */
/* *mu        return: field dependent mobility (cm^2/Vs) */
/* *dmde      return: derivative of mu with e */


int munsat(double e, double v, double mu0, double *mu, double *dmde)
{
  *mu  = mu0/(1.0+mu0*e/v);
  *dmde = (*mu)*(*mu)/v;
  return(0);                /* 0 - ok */
}
```

The function will then need to be stored in a file. For example, the function can be stored in the file: test.lib. The function is then introduced into a specific example by specifying the file name on the F.MUNSAT parameter in the MATERIAL statement as follows:

```
MATERIAL F.MUNSAT="test.lib"
```

When the input deck is executed, your C-INTERPRETER functions will be used in place of the built in function. When trying this example, place PRINT statements (using the printf C command) in the function to check that the function is working correctly.

Table A-1 shows a complete list of all the interpreter functions available in ATLAS.

Table A-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
BEAM	F.RADIATE	radiate ()	Generation rate as a function of position.
BEAM	F3.RADIATE	radiate3 ()	Generation rate as a function of position (3D).
BEAM	F.REFLECT	reflect ()	Reflection coefficient.
DEFECT	F.TFTACC	tftacc ()	TFT acceptor trap density as a function of energy.
CONTACT	F.ETUNNEL	fetunnel ()	Schottky contact electron tunneling.
CONTACT	F.HTUNNEL	fhtunnel ()	Schottky contact hole tunneling.
CONTACT	F.WORKF	workf ()	Electrical Field dependent workfunction.
DEFECT	F.TFTDON	tftdon ()	TFT donor trap density as a function of energy.
DEGREDAT	F.NTA	devdeg_nta ()	Interface acceptor trap density as a function of location.
DEGREDAT	F.NTD	devdeg_ntd ()	Interface donor trap density as a function of location.
DEGREDAT	F.SIGMAE	devdeg_sigmae ()	Interface electron capture cross-section.
DEGREDAT	F.SIGMAH	devdeg_sigmah ()	Interface hole capture cross-section.
DOPING	F.COMPOSIT	composition ()	Position dependent composition fractions.
DOPING	F.DOPING	doping ()	Position dependent net doping.
DOPING	F3.DOPING	doping3 ()	Position dependent net doping.
INTERFACE	F.QF	int_fixed_charge ()	Interface fixed charge as a function of position.
INTERFACE	F.HTE.N	hten()	Electron heterojunction thermionic emission current.
INTERFACE	F.HTE.P	htep()	Hole heterojunction thermionic emission current.
INTERFACE	F.HTFE.N	htfen()	Electron heterojunction thermionic field emission delta term.

Table A-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
INTERFACE	F.HTFE.P	htfep()	Hole heterojunction thermionic field emission delta term.
INTTRAP	F.DENSITY	inte_acc_trap ()	Interface ACCEPTOR trap density as a function of position.
INTTRAP	F.DENSITY	inte_don_trap ()	Interface DONOR trap density as a function of position.
IMPACT	F.EDIIN	ediin ()	Electron temperature coefficients for impact ionization.
IMPACT	F.EDIIP	ediip ()	Hole temperature coefficients for impact ionization.
INTTRAP	F.INTACCEPTO	intacceptor ()	Initial interface acceptor trap density and so on as a function of position.
LASER	F.MIRROR	mirror	Mirror facet reflectivities on a function of wavelength.
MATERIAL	F.BANDCOMP	bandcomp ()	Temperature and composition dependent band parameters.
MATERIAL	F.BGN	bgn ()	Composition, temperature, and doping dependent band gap narrowing.
MATERIAL	F.CBDOSFN	cbdosfn	Electron temperature dependent effective conduction band density of states.
MATERIAL	F.CONMUN	conmun ()	Composition, temperature, position, and doping dependent electron mobility.
MATERIAL	F.CONMUP	conmup ()	Composition, temperature, position, and doping dependent hole mobility.
MATERIAL	F.COPT	copt ()	Radiative recombination rate as a function of composition and temperature.
MATERIAL	F.EPSILON	epsilon ()	Composition and temperature dependent permittivity.
MATERIAL	F.FERRO	ferro ()	Position and field dependent permittivity.
MATERIAL	F.GAUN	gaun ()	Electron Auguer rate as a function of composition and temperature.

Table A-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.GAUP	gaup ()	Hole Auger rate as a function of composition and temperature.
MATERIAL	F.INDEX	index ()	Wavelength dependent complex index of refraction.
MATERIAL	F.MNSNDIFF	mnsndiff ()	Microscopic noise source for electron diffusion noise.
MATERIAL	F.MNSPDIFF	mnspdiff ()	Microscopic noise source for hole diffusion noise.
MATERIAL	F.MNSNFLICKER	mnsnflicker ()	Microscopic noise source for electron flicker noise.
MATERIAL	F.MNSPFLICKER	mnspflicker ()	Microscopic noise source for hole flicker noise.
MATERIAL	F.MUNSAT	munsat ()	Electron velocity saturation model.
MATERIAL	F.MUPSAT	mupsat ()	Hole velocity saturation model.
MATERIAL	F.RECOMB	recomb ()	Position, temperature, and concentration dependent recombination.
MATERIAL	F.TARUP	taurp ()	Energy dependent hole relaxation time.
MATERIAL	F.TAUN	taun ()	Position, temperture, and doping dependent electron SRH lifetime.
MATERIAL	F.TAUP	taup ()	Position, temperture, and doping dependent hole SRH lifetime.
MATERIAL	F.TAURN	taurn ()	Energy dependent electron relaxation time.
MATERIAL	F.TCAP	TCAP()	Lattice Thermal Capacity as a function of lattice temperature, position, doping, and fraction composition.
MATERIAL	F.TCOND	TCOND()	Lattice Thermal Conductivity as a function of lattice temperature, position, doping, and fraction composition.
MATERIAL	F.VSATN	vsatn ()	Composition and temperature dependent electron saturation velocity.
MATERIAL	F.VSATP	vsatp ()	Composition and temperature dependent hole saturation velocity.

Table A-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.VBDOSFN	vbdosfn	Hole temperature dependent effective valence band density of states.
MATERIAL/ MOBILITY	F.ENMUN	endepmun	Electron mobility as a function of electron temperature and a few other choice variables.
MATERIAL/ MOBILITY	F.ENMUP	endepmup	Hole mobility as a function of hole temperature and a few other choice variables.
MODELS	F.ALPHAA	bulk_absorb()	Bulk absorption coefficient versus carrier density and photon energy.
MODELS	F.KSN	fksn()	Energy dependent electron Peltier coefficient.
MODELS	F.KSP	fksp()	Energy dependent hole Peltier coefficient.
TRAP	F.DENSITY	acc_trap()	ACCEPTOR trap density as a function of position.
TRAP	F.DENSITY	don_trap()	DONOR trap density as a function of position.

Appendix B: Material Systems

B.1: Overview

ATLAS understands a library of materials for reference to material properties and models of various regions in the semiconductor device. These materials are chosen to represent those most commonly used by semiconductor physicists today. BLAZE or DEVICE3D users will have access to all of these materials. S-PISCES users will have only access to Silicon and Polysilicon.

S-PISCES is designed to maintain backward compatibility with the standalone program, SPISCES2 Version 5.2. In the SPISCES2 syntax, certain materials can be used in the REGION statement just by using their name as logical parameters. This syntax is still supported.

B.2: Semiconductors, Insulators, and Conductors

All materials in ATLAS are strictly defined into three classes: semiconductor materials, insulator materials, or conductors. Each class of materials has particular properties.

B.2.1: Semiconductors

All equations specified by your choice of models are solved in semiconductor regions. All semiconductor regions must have a band structure defined in terms of bandgap, density of states, affinity, and so on. The parameters used for any simulation can be echoed to the run-time output using `MODELS PRINT`. For complex cases with mole fraction dependent models, these quantities can be seen in TONYPLOT by specifying `OUTPUT BAND.PARAM` and saving a solution file.

Any semiconductor region that is defined as an electrode is then considered to be a conductor region. This is typical for polysilicon gate electrodes.

B.2.2: Insulators

In insulator materials, only the Poisson and lattice heat equations are solved. Therefore for isothermal simulations, the only parameter required for an insulator is dielectric permittivity defined using `MATERIAL PERM=<n>`.

Materials usually considered as insulators (e.g., SiO_2) can be treated as semiconductors using BLAZE, however all semiconductor parameters are then required.

B.2.3: Conductors

All conductor materials must be defined as electrodes, and all electrode regions are defined as conductor material regions. If a file containing regions of a material known to be a conductor are read in, these regions will automatically become un-named electrodes. As noted below, if the file contains unknown materials, these regions will become insulators.

During electrical simulation, only the electrode boundary nodes are used. Nodes that are entirely within an electrode region are not solved. Any quantities seen inside a conductor region in TONYPLOT are spurious. Only optical ray tracing and absorption for LUMINOUS and lattice heating are solved inside of conductor/electrode regions.

B.2.4: Unknown Materials

If a mesh file is read containing materials not in Table B-1, these will automatically become insulator regions with a relative permittivity of 3.9. All user-defined materials from ATHENA, regardless of the material name chosen, will also become such insulator materials.

B.3: ATLAS Materials

This section lists the materials in ATLAS. The superscripted numbers (e.g., Silicon[(1)]) on these tables refer to the notes on the next page.

Table B-1. The ATLAS Materials			
Single Element Semiconductors			
Silicon[(1)]	Poly[(2)]	Germanium	Diamond
Binary Compound Semiconductors			
GaAs [(3)]	GaP	CdSe	SnTe
SiGe	InP	CdTe	ScN
SiC-6H	InSb	HgS	GaN
SiC-3C	InAs	HgSe	AlN
SiC-4H			
AlP	ZnS	HgTe	InN
AlAs	ZnSe	PbS	BeTe
AlSb	ZnTe	PbSe	Zn0
GaSb	CdS	PbTe	
Ternary Compound Semiconductors			
AlGaAs	GaSbP	InAlAs	GaAsP
InGaAs	GaSbAs	InAsP	HgCdTe
InGaP	InGaN	AlGaN	CdZnTe
InAlP	InGaSb	InAlSb	AlGaSb
InAsSb	GaAsSb	AlAsSb	InPSb
AlPSb	AlPAs	AlGaP	
Quaternary Compound Semiconductors			
InGaAsP	AlGaAsP	AlGaAsSb	InAlGaN
InGaNAs	InGaNP	AlGaNAs	AlGaNP
AlInNAs	AlInNP	InAlGaAs	InAlGaP
InAlAsP			
Insulators			
Vacuum	Oxide	Nitride	Si3N4
Air	SiO2	SiN	Sapphire
Ambient	InPAsSb	InGaAsSb	InAlAsSb

Table B-1. The ATLAS Materials			
CuInGaSe2			
Conductors[4]			
Polysilicon[2]	Palladium	TiW	TaSi
Aluminum	Cobalt	Copper	PaSi
Gold	Molybdenum	Tin	PtSi
Silver	Lead	Nickel	MoSi
AlSi	Iron	WSi	ZrSi
Tungsten	Tantalum	TiSi	AlSi
Titanium	AlSiTi	NiSi	Conductor
Platinum	AlSiCu	CoSi	Contact
ITO			
Organics			
Organic	**Peutancene**		**Ala3**
TPD	PPV		Tetracene

Notes

- The material models and parameters of Silicon are identical to those of S-PISCES2 Version 5.2. Be aware that although these band parameters may be physically inaccurate compared to bulk silicon measurements, most other material parameters and models are empirically tuned using these band parameters.
- Polysilicon is treated differently depending on how it is used. In cases where it's defined as an electrode, it's treated as a conductor. It can also be used as a semiconductor such as in a polysilicon emitter bipolars.
- The composition of SiGe is the only binary compound that can be varied to simulate the effects of band gap variations.
- Conductor names are only associated with electrodes. They are used for the specification of thermal conductivities and complex index of refraction and for display in TONYPLOT.

B.3.1: Specifying Compound Semiconductors Rules

The rules for specifying the order of elements for compound semiconductors are derived from the rules used by the International Union of Pure and Applied Chemistry [113]. These rules are:

1. Cations appear before anions.
2. When more than one cation is present the order progresses from the element with the largest atomic number to the element with the smallest atomic number.
3. The order of anions should be the in order of the following list: B, Si, C, Sb, As, P, N, H, Te, Se, S, At, I, Br, Cl, O, and F.
4. The composition fraction x is applied to the cation listed first.
5. The composition y is applied to the anion listed first.

To accommodate popular conventions, there are several exceptions to these rules (see Table B-2).

Table B-2. Exceptions to the Specifying Compound Semiconductor Rules

Material	Description
`SiGe`	The composition fraction x applies to the Ge component. SiGe is then specified as $Si_{(1-x)}Ge_{(x)}$, an exception to rule #4.
`AlGaAs`	This is specified as $Al_{(x)}Ga_{(1-x)}As$. This is an exception to rule #2.
`InGaAsP` (system)	The convention $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ as set forth by Adachi [1] is used. This is an exception to rule #4.
`InAlAs`	The convention $In_{(1-x)}Al_{(x)}As$ as set forth by Ishikawi [75] is used. This is an exception to rule #4.

B.4: Silicon and Polysilicon

The material parameters defaults for Polysilicon are identical to those for Silicon. The following paragraphs describe some of the material parameter defaults for Silicon and Polysilicon.

Note: A complete description is given of each model Chapter 3: "Physics". The parameter defaults listed in that chapter are all Silicon material defaults.

Table B-3. Band parameters for Silicon and Poly

Material	Eg300 eV	a	b	Nc300 per cc	Nv300 per cc	χ eV
Silicon	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17
Poly	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17

Table B-4. Static dielectric constants for Silicon and Poly

Material	Dielectric Constant
Silicon	11.8
Poly	11.8

The default mobility parameters for Silicon and Poly are identical in all cases. The defaults used depend on the particular mobility models in question. A full description of each mobility model and their coefficients are given in Chapter 3: "Physics", Section 3.6.1: "Mobility Modeling".

Table B-5 contains the silicon and polysilicon default values for the low field constant mobility model.

Table B-5. Lattice Mobility Model Defaults for Silicon and Poly

Material	MUN cm^2/Vs	MUP cm^2/Vs	TMUN	TMUP
Silicon	1000.0	500.0	1.5	1.5
Poly	1000.0	500.0	1.5	1.5

Table B-6 shows the silicon and polysilicon default values for the field dependent mobility model.

Table B-6. Parallel Field Dependent Mobility Model Parameters for Silicon and Poly		
Material	**BETAN**	**BETAP**
Silicon	2	1
Poly	2	1

Table B-7 shows the default values used in the bandgap narrowing model for Silicon and Polysilicon.

Table B-7. Bandgap Narrowing Parameters for Silicon and Poly			
Statement	**Parameter**	**Defaults**	**Units**
MATERIAL	BGN.E	6.92×10^{-3}	V
MATERIAL	BGN.N	1.3×10^{17}	cm^{-3}
MATERIAL	BGN.C	0.5	—

Table B-8 shows the default parameters for Schockley-Read-Hall (SRH) recombination.

Table B-8. SRH Lifetime Parameter Defaults for Silicon and Poly				
Material	**TAUN0 (s)**	**TAUP0 (s)**	**NSRHN (cm^{-3})**	**NSRHP (cm^{-3})**
Silicon	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}
Poly	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}

The default parameters for Auger recombination are given in Table B-9.

Table B-9. Auger Coefficient Defaults for Silicon and Poly		
Material	**AUGN**	**AUGP**
Silicon	8.3×10^{-32}	1.8×10^{-31}
Poly	8.3×10^{-32}	1.8×10^{-31}

Table B-10 shows the default values for the SELB impact ionization coefficients.

Table B-10. Impact Ionization Coefficients for Silicon and Poly	
Parameter	**Value**
EGRAN	4.0×10^5
BETAN	1.0
BETAP	1.0
AN1	7.03×10^5
AN2	7.03×10^5
BN1	1.231×10^6
BN2	1.231×10^6
AP1	6.71×10^5
AP2	1.582×10^6
BP1	1.693×10^6
BP2	2.036×10^6

Table B-11 shows the default Richardson cofficients.

Table B-11. Effective Richardson Coefficients for Silicon and Poly		
Material	**ARICHN ($A/cm^2/K^2$)**	**ARICHP ($A/cm^2/K^2$)**
Silicon	110.0	30.0
Poly	110.0	30.0

B.5: The $Al_{(x)}Ga_{(1-x)}As$ Material System

Table B-12 shows the default recombination parameters for AlGaAs.

Table B-12. Default Recombination Parameters for AlGaAs		
Parameter	**Value**	**Equation**
TAUN0	1.0×10^{-9}	3-213
TAUP0	1.0×10^{-8}	3-213
COPT	1.5×10^{-10}	3-226
AUGN	5.0×10^{-30}	3-227
AUGP	1.0×10^{-31}	3-227

Table B-13 shows the default values for the SELB impact ionization coefficients used for GaAs. AlGaAs uses the same values as GaAs.

Table B-13. Impact Ionization Coefficients for GaAs	
Parameter	**Value**
EGRAN	0.0
BETAN	1.82
BETAP	1.75
EGRAN	0.0
AN1	1.889×10^{5}
AN2	1.889×10^{5}
BN1	5.75×10^{5}
BN2	5.75×10^{5}
AP1	2.215×10^{5}
AP2	2.215×10^{5}
BP1	6.57×10^{5}
BP2	6.57×10^{5}

The default values for the effective Richardson coefficients for GaAs are 6.2875 $A/cm^2/K^2$ for electrons and 105.2 $A/cm^2/K^2$ for holes.

B.6: The $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ System

The default material thermal models for InGaAsP assumes lattice-matching to InP. The material density is then given by

$$\rho = 4.791 + 0.575y.composition + 0.138y.composition$$

The specific heat for InGaAsP is given by

$$C_p = 0.322 + 0.026y.composition - 0.008y.composition$$

The thermal resistivities of InGaAsP are linearly interpolated from Table B-14.

Table B-14. Thermal Resistivities for InGaAsP Lattice-Matched to InP

Composition Fraction y	Thermal Resistivity (deg(cm/w)
0.0	1.47
0.1	7.05
0.2	11.84
0.3	15.83
0.4	19.02
0.5	21.40
0.6	22.96
0.7	23.71
0.8	23.63
0.9	22.71
1.0	20.95

The default thermal properties for the ternary compounds in the InGaAsP system: $In_{(1-x)}Ga_{(x)}As$, $In_{(1-x)}Ga_{(x)}P$, $InAs_{(y)}P_{(1-y)}$, and $GaAs_{(y)}P_{(1-y)}$, are given as a function of composition fraction by linear interpolations from these binary compounds.

B.7: Silicon Carbide (SiC)

Table B-15 shows the default values for the SELB impact ionization coefficients used for SiC.

Table B-15. Impact Ionization Coefficients for SiC	
Parameter	**Value**
EGRAN	0.0
BETAN	1.0
BETAP	1.0
AN1	1.66×10^{6}
AN2	1.66×10^{6}
BN1	1.273×10^{7}
BN2	1.273×10^{7}
AP1	5.18×10^{6}
AP2	5.18×10^{6}
BP1	1.4×10^{7}
BP2	1.4×10^{7}

Table B-16 shows the default thermal parameters used for both 6H-SiC and 3C-SiC.

Table B-16. Default Thermal Parameters for SiC		
Parameter	**Value**	
	3C-SiC	**6H-SiC**
TCA	0.204	0.385

B.8: Material Defaults for GaN/InN/AlN System

Tables B-17 through B-20 show the default values for the band structure parameters for InN, GaN and AlN for each of the user-defined parameter sets. To choose the parameters sets, specify KP.SET1 (Table B-20), KP.SET2 (Table B-21), KP.SET3 (Table B-22), and KP.SET4 (Table B-24). By default, the parameters in Table B-17 are used.

Table B-17. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET1 (All Parameters Values are from [122] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN	GaN	AlN
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.11	0.20	0.33
Electron eff. mass (t)	m_{ct}	MTT	m0	0.11	0.18	0.25
Hole eff. mass param.	A1	A1		-9.24	-7.24	-3.95
Hole eff. mass param.	A2	A2		-0.60	-0.51	-0.27
Hole eff. mass param.	A3	A3		8.68	6.73	3.68
Hole eff. mass param.	A4	A4		-4.34	-3.36	-1.84
Hole eff. mass param.	A5	A5		-4.32	-3.35	-1.92
Hole eff. mass param.	A6	A6		-6.08	-4.72	-2.91
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.89	3.42	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.019	-0.164
Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01413	0.01913
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Elastic constant	C33	C33	GPa	200	392	382
Elastic constant	C13	C13	GPa	94	100	127

Table B-17. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET1 (All Parameters Values are from [122] unless otherwise noted.)						
Hydrostatic deformation potential	ac	AC	eV	-4.08	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-0.89	-0.89	-0.89
Shear deform. potential	D2	D2	eV	4.27	4.27	4.27
Shear deform. potential	D3	D3	eV	5.18	5.18	5.18
Shear deform. potential	D4	D4	eV	-2.59	-2.59	-2.29

Table B-18. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET2 (All Parameters Values are from [169] unless otherwise noted.)						
Parameter	**Symbol**	**Syntax**	**Units**	**InN**	**GaN**	**AlN**
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.12	0.20	0.32
Electron eff. mass (t)	m_{ct}	MTT	m0	0.12	0.20	0.28
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95
Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27
Hole eff. mass param.	A3	A3		7.57	5.65	3.68
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92
Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.507	6.23
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.019	-0.164

Table B-18. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET2 (All Parameters Values are from [169] unless otherwise noted.)

Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01413	0.01913
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Elastic constant	C33	C33	GPa	224	398	373
Elastic constant	C13	C13	GPa	92	106	108
Hydrostatic deformation potential	ac	AC	eV	-4.08	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-0.89	-3.0	-0.89
Shear deform. potential	D2	D2	eV	4.27	3.6	4.27
Shear deform. potential	D3	D3	eV	5.18	8.82	5.18
Shear deform. potential	D4	D4	eV	-2.59	-4.41	-2.29

Table B-19. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET3 (All Parameters Values are from [36] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN [169]	GaN	AlN
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.12	0.20	0.32
Electron eff. mass (t)	m_{ct}	MTT	m0	0.12	0.20	0.28
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95 [169]
Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27 [169]
Hole eff. mass param.	A3	A3		7.57	5.65	3.68 [169]
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84 [169]
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92 [169]

Table B-19. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET3 (All Parameters Values are from [36] unless otherwise noted.)

Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91 [169]
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.44	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.016	-0.0585
Crystal-field split energy	D2	DELTA2	eV	0.0013	0.01213	0.020413
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112 [169]
Elastic constant	C33	C33	GPa	224	267	395
Elastic constant	C13	C13	GPa	92	158	120
Hydrostatic deformation potential	ac	AC	eV	-3.5	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-3.0	0.7	0.7
Shear deform. potential	D2	D2	eV	3.6	2.1	2.1
Shear deform. potential	D3	D3	eV	8.82	1.4	1.4
Shear deform. potential	D4	D4	eV	-4.41	-0.7	-0.7

Table B-20. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET4 (All Parameters Values are from [38] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN [169]	GaN	AlN
Electron eff. mass (z)	m_{cz}	MZZ	m0	0.12	0.20	0.33
Electron eff. mass (t)	m_{ct}	MTT	m0	0.12	0.18	0.25
Hole eff. mass param.	A1	A1		-8.21	-6.56	-3.95 [169]

Table B-20. Electronic Band-Structure Parameters for InN, GaN and AlN Using KP.SET4 (All Parameters Values are from [38] unless otherwise noted.)

Hole eff. mass param.	A2	A2		-0.68	-0.91	-0.27 [169]
Hole eff. mass param.	A3	A3		7.57	5.65	3.68 [169]
Hole eff. mass param.	A4	A4		-5.23	-2.83	-1.84 [169]
Hole eff. mass param.	A5	A5		-5.11	-3.13	-1.92 [169]
Hole eff. mass param.	A6	A6		-5.96	-4.86	-2.91 [169]
Valence band reference	Ev0		eV	-1.59	-2.64	-3.44
Direct band gap (300K)	Eg(300)		eV	1.994	3.44	6.28
Spin-orbit split energy	D1	DELTA1	eV	0.041	0.021	-0.0585
Crystal-field split energy	D2	DELTA2	eV	0.00113	0.01618	0.020413
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112 [169]
Elastic constant	C33	C33	GPa	224	392	395
Elastic constant	C13	C13	GPa	92	100	120
Hydrostatic deformation potential	ac	AC	eV	-35	-4.08	-4.08
Shear deform. potential	D1	D1	eV	-3.0	-0.89	-0.89
Shear deform. potential	D2	D2	eV	3.6	4.27	4.27
Shear deform. potential	D3	D3	eV	8.82	5.18	5.18
Shear deform. potential	D4	D4	eV	-4.41	-2.59	-2.29

Tables B-21 through B-23 show the default polarization parameters for InN, GaN and AlN for each of the user-defined parameter sets. To choose the parameter sets, specify POL.SET1 (Table B-24), POL.SET2 (Table B-25), or POL.SET3 (Table B-26) on the MATERIAL statement. By default, the parameter values in Table B-24 are used.

Table B-21. Polarization Parameters for InN, GaN and AlN Using POL.SET1 (All Parameters Values are from [122] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN	GaN	AlN
Lattice constant	a_0	ALATTICE	Å	3.548	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.042	-0.034	-0.09
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.81	0.67	1.5
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.41	-0.34	-0.53

Table B-22. Polarization Parameters for InN, GaN and AlN Using POL.SET2 (All Parameters Values are from [169] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN	GaN	AlN
Lattice constant	a_0	ALATTICE	Å	3.545	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.032	-0.029	-0.081
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.97	1.27	1.79
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.57	-0.35	-0.5

Table B-23. Polarization Parameters for InN, GaN and AlN Using POL.SET3 (All Parameters Values are from [37] unless otherwise noted.)

Parameter	Symbol	Syntax	Units	InN	GaN	AlN
Lattice constant	a_0	ALATTICE	Å	3.54	3.189	3.112
Spontaneous polarization	P_{sp}	PSP	C/m^2	-0.032	-0.029	-0.081
Piezoelectric const.(z)	e_{33}	E33	C/m^2	0.97	0.73	1.46
Piezoelectric const.(x,y)	e_{31}	E31	C/m^2	-0.57	-0.49	-0.6

B.9: Material Defaults for Compound Semiconductors

Tables B-24a and B-27b, and B-27c show the material defaults for binary compound semiconductors. Table B-28 shows Material defaults for ternary compound semiconductors.

Note: `x.comp` is set to 0.3. See Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.3: "Material Dependent Physical Models" for more information.

Table B-24a. Materials Defaults for binary compound semiconductors

Material	GaAs	GaP	CdSe	SnTe	InP	CdTe	ScN
Epsilon	13.2	11.1	10.6	0	12.5	10.9	0
Eg (eV)	1.42	2.26	1.74	0.18	1.35	1.5	2.15
Chi (eV)	4.07	4.4	0	0	4.4	4.28	0
Nc (per cc)	4.35E+17	1.76E+18	1.18E+18	1	5.68E+17	1	1
Nv (per cc)	8.16E+18	8.87E+18	7.57E+18	1	8.87E+18	1	1
ni (per cc)	2.12E+06	0.409	7.22E+03	0.0308	1.03E+07	2.51E-13	8.70E-19
Gc	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Lifetime (el)	1.00E-09	1	1	1	1	1	1
Lifetime (ho)	2.00E-08	1	1	1	1	1	1
Auger cn	5.00E-30	0	0	0	0	0	0
Auger cp	1.00E-31	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0
Copt	1.50E-10	0	0	0	0	0	0
An**	6.29	20.4	15.6	1.40E-11	9.61	1.40E-11	1.40E-11
Ap**	105	60.1	54.1	1.40E-11	60.1	1.40E-11	1.40E-11
betan	1.82	1	1	1	1	1	1
betap	1.75	1	1	1	1	1	1
egran	0	0	0	0	0	0	0
an1	1.90E+05	7.30E+04	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn1	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.90E+05	7.30E+04	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
an2	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	2.22E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	6.57E+05	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	2.22E+05	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	6.57E+05	2.40E+05	2.40E+05	2.04E+06	2.40E+05	2.04E+06	2.04E+06
Vsatn (cm/s) :	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s) :	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
mun (cm^2/Vs)	8000	300	800		4600	1050	
tmun	1	1.5	1.5		1.5	1.5	
mup (cm^2/Vs)	400	100			150	100	
tmup	2.1	1.5			1.5	1.5	

Table B-27b. Materials Defaults for binary compound semiconductors

Material	6H-SiC	4H-SiC	3C-SiC	InSb	HgS	InAs	HgSe	AlP
Epsilon	9.66	9.7	9.72	18	0	14.6	25	9.8
Eg (eV)	2.9	3.26	2.2	0.17	2.5	0.35	0	2.43
Chi (eV)	0	0	4.2	4.59	0	4.13	0	0
Nc (per cc)	7.68E+18	1.66E+19	6.59E+18	4.16E+16	1	9.33E+16	1	1
Nv (per cc)	4.76E+18	3.30E+19	1.68E+18	6.35E+18	1	8.12E+18	1	1
ni (per cc)	2.64E-06	9.07E-07	1.1	1.92E+16	9.99E-22	9.99E+14	1	3.87E-21
Gc	2	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Lifetime (el)	1	1	1	1	1	1	1	1
Lifetime (ho)	1	1	1	1	1	1	1	1
Auger cn	2.80E-31	5.00E-29	0	0	0	0	0	0
Auger cp	9.90E-31	9.90E-32	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0	0
Copt	0	0	0	0	0	0	0	0
An**	54.6	91.3	49.3	1.68	1.40E-11	2.88	1.40E-11	1.40E-11
Ap**	39.7	144	19.8	48.1	1.40E-11	56.6	1.40E-11	1.40E-11
betan	1	1	1	1	1	1	1	1
betap	1	1	1	1	1	1	1	1
egran	-1	-1	-1	0	0	0	0	0
an1	1.66E+06	1.66E+06	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn1	1.27E+07	1.27E+07	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.66E+06	1.66E+06	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05
bn2	1.27E+07	1.27E+07	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	5.18E+06	5.18E+06	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	1.40E+07	1.40E+07	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	5.18E+06	5.18E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	1.40E+07	1.40E+07	2.04E+06	2.40E+05	2.04E+06	2.40E+05	2.04E+06	2.04E+06
Vsatn (cm/s) :	2.00E+07	1.25E+07	2.00E+07	1.00E+06	1.00E+06	1.00E+06	1.00E+06	7.70E+06
Vsatp (cm/s) :	1.00E+06	1.00E+07	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	7.70E+06
mun (cm^2/Vs)	330	460	1000	78000		33000	5500	80
tmun	1.5	1	1.5	1.5		1.5	1.5	1.5
mup (cm^2/Vs)	60	124	50	750		460		
tmup	1.5	1	1.5	1.5		1.5		

Table B-27c. Material Defaults for binary compound semiconductors

Material	ZnS	HgTe	AlAs	ZnSe	PbS	BeTe	AlSb	ZnTe	PbSe	GaSb	PbTe	SiGe
Epsilon	8.3	20	12	8.1	170	0	11	9.7	250	15.7	412	11.8
Eg (eV)	3.8	0	2.16	2.58	0.37	2.57	1.52	2.28	0.26	0.72	0.29	0.78
Chi (eV)	0	0	3.5	4.09	0	0	3.65	3.5	0	4.06	4.6	4.17
Nc (per cc)	6.35E+18	1	4.35E+17	1	3.14E+18	1	6.79E+18	1	1	5.68E+18	1.76E+18	1.92E+19
Nv (per cc)	1	1	8.16E+18	1	3.14E+18	1	6.35E+18	1	1	2.95E+18	2.24E+18	8.20E+18
ni (per cc)	3.02E-23	1	1.35	2.13E-22	2.45E+15	2.58E-22	1.12E+06	7.04E-20	0.00655	3.66E+12	7.28E+15	3.52E+12
Gc	2	2	2	2	2	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters												
Lifetime (el)	1	1	1.00E-09	1	1	1	1	1	1	1	1	3.00E-05
Lifetime (ho)	1	1	2.00E-08	1	1	1	1	1	1	1	1	1.00E-05
Auger cn	0	0	5.00E-30	0	0	0	0	0	0	0	0	8.30E-32
Auger cp	0	0	1.00E-31	0	0	0	0	0	0	0	0	1.80E-31
Auger kn	0	0	0	0	0	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0	0	0	0	0	0
Copt	0	0	1.50E-10	0	0	0	0	0	0	0	0	0
An**	48.1	1.40E-11	8.05	1.40E-11	30	1.40E-11	50.3	1.40E-11	1.40E-11	44.6	20.4	110
Ap**	1.40E-11	1.40E-11	56.8	1.40E-11	30	1.40E-11	48.1	1.40E-11	1.40E-11	28.8	24	30
Impact Ionization Parameters												
betan	1	1	1.82	1	1	1	1	1	1	1	1	1
betap	1	1	1.75	1	1	1	1	1	1	1	1	1
egran	0	0	0	0	0	0	0	0	0	0	0	4.00E+05
an1	7.03E+05	7.03E+05	1.90E+05	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05	7.30E+04	7.30E+04	7.03E+05
bn1	1.23E+06	1.23E+06	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	7.03E+05	7.03E+05	1.90E+05	7.03E+05	7.30E+04	7.03E+05	7.30E+04	7.03E+05	7.03E+05	7.30E+04	7.30E+04	7.03E+05
an2	1.23E+06	1.23E+06	5.75E+05	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	6.71E+05	6.71E+05	2.22E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	1.69E+06	1.69E+06	6.57E+05	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	1.58E+06	1.58E+06	2.22E+05	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	2.04E+06	2.04E+06	6.57E+05	2.04E+06	2.40E+05	2.04E+06	2.40E+05	2.04E+06	2.04E+06	2.40E+05	2.40E+05	2.04E+06
Saturation Velocities												
Vsatn (cm/s)	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+07
Vsatp (cm/s)	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	7.98E+06
Mobility Parameters												
mun (cm^2/Vs)	165	22000	1000	100	600		200	7	1020	4000	6000	1430
tmun	1.5	1.5	1.5	1.5	1.5		1.5	1.5	1.5	1.5	1.5	1.5
mup (cm^2/Vs)	5	100	100		700		550		930	1400	4000	480
tmup	1.5	1.5	1.5		1.5		1.5		1.5	1.5	1.5	1.5

Note: `x.comp` is set to 0.3. See Chapter 5: “Blaze: Compound Material 2D Simulator”, Section 5.3: “Material Dependent Physical Models” for more information.

Table B-28. Material Default for ternary compound semiconductors

Material	AlGaAs	GaSbP	InAlAs	GaAsP	InGaAs	GaSbAs	InAsP	HgCdTe	InGaP
Epsilon	12.3	0	0	11.7	14.2	0	13.1	16.4	12.1
Band Parameters									
Eg (eV)	1.8	2.01	2.1	2.31	0.571	0.665	1.03	0.291	1.61
Chi (eV)	3.75	0	0	4.32	4.13	4.06	4.32	2.54	4.4
Nc (per cc)	4.35E+17	1	1	1.04E+18	1.15E+17	1	2.86E+17	7.58E+16	7.18E+17
Nv (per cc)	8.16E+18	1	1	8.54E+18	8.12E+18	1	8.54E+18	1.02E+19	8.87E+18
ni (per cc)	1.37E+03	1.24E-17	2.29E-18	0.125	1.56E+13	2.61E-06	3.56E+09	3.18E+15	7.43E+04
Gc	2	2	2	2	2	2	2	2	2
Gv	4	4	4	4	4	4	4	4	4
Ed (eV)	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV)	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters									
Lifetime (el)	1.00E-09	1	1	1	1	1	1	1	1
Lifetime (ho)	2.00E-08	1	1	1	1	1	1	1	1
Auger cn	5.00E-30	0	0	0	0	0	0	0	0
Auger cp	1.00E-31	0	0	0	0	0	0	0	0
Auger kn	0	0	0	0	0	0	0	0	0
Auger kp	0	0	0	0	0	0	0	0	0
Copt	1.50E-10	0	0	0	0	0	0	0	0
An**	8.05	1.40E-11	1.40E-11	14.4	3.32	1.40E-11	6.08	2.51	11.2
Ap**	56.8	1.40E-11	1.40E-11	58.6	56.6	1.40E-11	58.6	66.1	60.1
Impact Ionization Model Parameters (Selberherr Model)									
betan	1.82	1	1	1	1	1	1	1	1
betap	1.75	1	1	1	1	1	1	1	1
egran	0	0	0	0	0	0	0	0	0
an1	1.90E+05	7.03E+05	8.60E+06	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05
bn1	5.75E+05	1.23E+06	3.50E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2	1.90E+05	7.03E+05	8.60E+06	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05
an2	5.75E+05	1.23E+06	3.50E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1	2.22E+05	6.71E+05	2.30E+07	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1	6.57E+05	1.69E+06	4.50E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2	2.22E+05	1.58E+06	2.30E+07	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2	6.57E+05	2.04E+06	4.50E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06
Saturation Velocities									
Vsatn (cm/s)	7.70E+06	1.00E+06	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s)	7.70E+06	1.00E+06	1.00E+06	1.00E+06	7.70E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06

Note: `x.comp` is set to 0.3. See Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.3: "Material Dependent Physical Models" for more information.

Table B-29 shows material defaults for quaternary compound semiconductors.

Table B-29. Material Defaults for quaternary compound semiconductors

Material :	InGaAsP	AlGaAsP	AlGaAsSb	InGaNAs	InGaNP	AlGaNAs	AlGaNP	AlInNAs	AlInNP	InAlGaAs	InAlGaP	InAlAsP
Epsilon :	12.7	0	0	0	0	0	0	0	0	0	0	0
Band Parameters												
Eg (eV) :	1.27	0	0	0	0	0	0	0	0	1.34	0	0
Chi (eV) :	4.32	0	0	0	0	0	0	0	0	1.46	0	0
Nc (per cc) :	3.61E+17	1	1	1	1	1	1	1	1	1	1	1
Nv (per cc) :	8.54E+18	1	1	1	1	1	1	1	1	1	1	1
ni (per cc) :	3.72E+07	1	1	1	1	1	1	1	1	5.04E-12	1	1
Gc :	2	2	2	2	2	2	2	2	2	2	2	2
Gv :	4	4	4	4	4	4	4	4	4	4	4	4
Ed (eV) :	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
Ea (eV) :	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045	0.045
Recombination Parameters												
Lifetime (el):	1	1	1	1	1	1	1	1	1	1	1	1
Lifetime (ho):	1	1	1	1	1	1	1	1	1	1	1	1
Auger cn :	0	0	0	0	0	0	0	0	0	0	0	0
Auger cp :	0	0	0	0	0	0	0	0	0	0	0	0
Auger kn :	0	0	0	0	0	0	0	0	0	0	0	0
Auger kp :	0	0	0	0	0	0	0	0	0	0	0	0
Copt :	0	0	0	0	0	0	0	0	0	0	0	0
An** :	7.11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11
Ap** :	58.6	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11	1.40E-11
Impact Ionization Parameters (Selberherr model)												
betan :	1	1	1	1	1	1	1	1	1	1	1	1
betap :	1	1	1	1	1	1	1	1	1	1	1	1
egran :	0	0	0	0	0	0	0	0	0	0	0	0
an1 :	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05
bn1 :	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
an2 :	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05	7.03E+05
an2 :	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06	1.23E+06
ap1 :	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05	6.71E+05
bp1 :	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06	1.69E+06
ap2 :	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06	1.58E+06
bp2 :	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06	2.04E+06
Saturation Velocities												
Vsatn (cm/s) :	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
Vsatp (cm/s) :	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06

Note: `x.comp` and `y.comp` are both set to 0.3. See Chapter 5: "Blaze: Compound Material 2D Simulator", Section 5.3: "Material Dependent Physical Models" for more information.

B.10: Miscellaneous Semiconductors

The remainder of the semiconductors available have defined default parameter values to various degrees of completeness [1]. The following tables describe those parameter defaults as they exist. Since many of the material parameters are currently unavailable, we recommended that you should be careful when using these materials. Make sure the proper values are used

Note: You can use the MODEL PRINT statement to echo the parameters used to the run-time output.

Table B-30. Band Parameters for Miscellaneous Semiconductors

Material	Eg(0)eV	Eg(300)eV	a	b	m_c	m_v	χeV
Silicon							
Polysilicon							
Ge	0.7437		4.77×10^{-4}	235.0	0.2225	0.2915	4.0
Diamond		5.45	4.77×10^{-4}	0.0	(a)	(b)	7.2
6H-SiC	2.9	2.9	0.0	0.0	0.454	0.33	
4H-SiC	3.26	3.26	0.0	0.0	0.76	1.20	
3C-SiC	2.2	2.2	0.0	0.0	0.41	0.165	
A1P	2.43	2.43	0.0	0.0			
A1As	2.16	2.16	0.0	0.0			3.5
A1Sb	1.6		2.69×10^{-4}	2.788	(c)	0.4	
GaSb	0.81		3.329×10^{-4}	-27.6622	(c)	0.24	3.65
InSb	0.235		2.817×10^{-4}	90.0003	0.014	0.4	4.06
ZnS	3.8	3.8	0.0	0.0	0.4		4.59
ZnSe	2.58	2.58	0.0	0.0	0.1	0.6	
ZnTe	2.28		0.0	0.0	0.1	0.6	4.09
Cds	2.53	2.53	0.0	0.0	0.21	0.8	3.5
CdSe	1.74	1.74	0.0	0.0	0.13	0.45	4.5
CdTe	1.5	1.5	0.0	0.0	0.14	0.37	
HgS	2.5	2.5	0.0	0.0			4.28
HgSe							
HgTe							
PbS	0.37	0.37	0.0	0.0	0.25	0.25	

Table B-30. Band Parameters for Miscellaneous Semiconductors							
Material	**Eg(0)eV**	**Eg(300)eV**	**a**	**b**	**m_c**	**m_v**	**χeV**
PbSe	0.26	0.26	0.0	0.0	0.33	0.34	
PbTe	0.29	0.29	0.0	0.0	0.17	0.20	4.6
SnTe	0.18	0.18	0.0	0.0			
ScN	2.15	2.15	0.0	0.0			
BeTe	2.57	2.57	0.0	0.0			

Notes

- $Nc300 = 5.0 \times 10^{18}$
- $Nv300 = 1.8 \times 10^{19}$
- $m_c(X) = 0.39$

  $m_c(G) = 0.09$

  $Nc = Nc(X) + Nc(G)$
- $m_c(G) = 0.047$

  $m_c(L) = 0.36$

  $Nc = Nc(G) + Nc(L)$

Table B-31. Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
Ge	16.0
Diamond	5.5
6H-SiC	9.66
4H-SiC	9.7
3C-SiC	9.72
AlP	9.8
AlAs	12.0
AlSb	11.0
GaSb	15.7
InSb	18.0
ZnS	8.3
ZnSe	8.1
CdS	8.9

Table B-31. Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
CdSe	10.6
CdTe	10.9
HgS	
HgSe	25.0
HgTe	20.
PbS	170.0
PbSe	250.0
PbTe	412.0
SnTe	
ScN	
BeTe	

Table B-32. Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cmcm/s)**
Ge	3900.0(a)	1900.0(b)		
Diamond	500.0	300.0	2.0×10^{7}	
6H-SiC	330.0	300.0	2.0×10^{7}	1×10^{7}
4H-SiC	460	124		
3C-SiC	1000.0	50.0	2.0×10^{7}	1×10^{7}
AlP	80.0			
AlAs	1000.0	100.0		
AlSb	200.0	550.0		
GaSb	4000.0	1400.0		
InSb	7800.0	750.0		
ZnS	165.0	5.0		
ZnSe	100.0	16		
CdS	340.0	50.0		
CdSe	800.0			
CdTe	1050.0	100.0		
HgS				

Table B-32. Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cmcm/s)**
HgSe	5500.0			
HgTe	22000.0	100.0		
PbS	600.0	700.0		
PbSe	1020.0	930.0		
PbTe	6000.0	4000.0		
SnTe				
ScN				
BeTe				

Notes

- Uses Equation B-4 with TMUN=1.66.
- Uses Equation B-4 with TMUP=2.33.

B.11: Insulators

The following tables show the default material parameters for insulator materials. As noted in the Section B.2: "Semiconductors, Insulators, and Conductors", the only parameter required for electrical simulation in insulator materials is the dielectric constant. Thermal and optical properties are required in GIGA and LUMINOUS respectively.

Table B-33. Default Static Dielectric Constants of Insulators

Material	Dielectric Constant
Vacuum	1.0
Air	1.0
Ambient	1.0
Oxide	3.9
Si02	3.9
Nitride	7.5
SiN	7.5
Si3N4	7.55
Sapphire	12.0

Table B-34. Default Thermal Parameters for Insulators

Material	Thermal Capacity (J/cm^3)	Thermal Conductivity (deg(cm/W))	Reference
Vacuum	0.0	0.0	
Air	1.0	0.026×10^{-2}	7
Ambient	1.0	0.026×10^{-2}	7
Oxide	3.066	0.014	4
Si02	3.066	0.014	4
Nitride	0.585	0.185	4
SiN	0.585	0.185	4
Si3N4	0.585	0.185	4
Sapphire			

B.12: Metals/Conductors

Table B-35 shows the default values for resistivities of metals and the temperature coefficients of resistivity [see also 175].

Table B-35. Resistivity and Temperature Coefficient of Metals

Material	Resistivity (μΩ·cm)	Temperature Coefficient (μΩ·cm/K)
Aluminum	2.6540	0.00429
Gold	2.35	0.004
Silver	1.59	0.0041
Tungsten	5.65	
Titanium	42.0	
Platinum	10.6	0.003927
Palladium	10.8	0.00377
Cobalt	6.24	0.00604
Molibdinum	5.2	
Lead	20.648	0.00336
Iron	9.71	0.00651
Tantalum	12.45	0.00383
Copper	6.24	0.00604
Tin	11.0	0.0047
Nickel	6.84	0.0069

B.13: Optical Properties

The default values for complex index of refraction in LUMINOUS are interpolated from the tables in [68, 117, 189]. Rather than print the tables here, the ranges of optical wavelengths for each material are listed in Table B-36.

Table B-36. Wavelength Ranges for Default Complex Index of Refraction

Material	Temperature (K)	Composition Fraction	Wavelengths (microns)
Silicon	300	NA	0.0103-2.0
AlAs	300	NA	0.2213 - 50.0
GaAs	300	NA	0.0 - 0.9814
InSb	300	NA	0.2296 - 6.5
InP	300	NA	0.1689 - 0.975
Poly	300	NA	0.1181 - 18.33
SiO2	300	NA	0.1145 - 1.7614

Table B-37 show the default parameters for the Sellmeier refractive index model (see Chapter 3: "Physics", Equation 3-505) for various materials.

Table B-37. Default Parameter Values [122] for the Sellmeier Refractive Index Model

Parameter	S0SELL	S1SELL	L1SELL	S2SELL	L2SELL
Si	3.129	8.54297	0.33671	0.00528	38.72983
Ge	9.282	6.7288	0.66412	0.21307	62.20932
GaAs	3.5	7.4969	0.4082	1.9347	37.17526
InP	7.255	2.316	0.6263	2.756	32.93934
AlAs	2.616	5.56711	0.2907	0.49252	10.0
GaSb	13.1	0.75464	1.2677	0.68245	10.0
InAs	11.1	0.71	2.551	2.75	45.6618
GaP	3.096	5.99865	0.30725	0.83878	17.32051
InSb	15.4	0.10425	7.15437	3.47475	44.72136
GaN	3.6	1.75	0.256	4.1	17.86057
AlN	3.14	1.3786	0.1715	3.861	15.0333

Table B-38 shows the default parameters for the Adachi refractive index model [1] (see Chapter 3: "Physics", Equation 3-506) for various materials.

Table B-38. Default Parameter Values [105] for the Adachi Refractive Index Model			
Parameter	**AADACHI**	**BADACHI**	**DADACHI**
GaAs	6.3	9.4	0.34
InP	8.4	6.6	0.11
AlAs	25.3	-0.8	0.28
GaSb	4.05	12.66	0.82
AlSB	59.68	-9.53	0.65
InAs	5.14	10.15	0.38
GaP	22.25	0.9	0.08
AlP	24.1	-2.0	0.07
InSb	7.91	13.07	0.81

Note: You can add the INDEX.CHECK parameter to the SOLVE statement to list the values of real and imaginary index being used in each solution. The indices are only printed when you perform the ray trace at the first reference to a given beam on the SOLVE statement.

B.14: User Defined Materials

The current version of ATLAS doesn't directly support user-defined materials. A simple workaround can be done using the already existing user specifications. This workaround is based on the use of an already existing material name and modifying the material parameters as appropriate.

In ATLAS, material names are defined to give you a reasonable set of default material parameters. You can override any of these defaults using the MATERIAL, IMPACT, MODEL, and MOBILITY statements.

The key to defining new materials is to choose a material name defined in ATLAS. Then, modify the material parameters of that material to match the user material. Here, you should choose a material that has default parameter values that might best match the user material, while making sure to choose a material that is not already in the user device. Then, associate this material name with the device regions where the new material is present. To do this, either specify the chosen material name on the appropriate REGION statements (when the device is defined in the ATLAS syntax) or choose the material name from the materials menu when defining the region in DEVEDIT. Next, modify the material statements using MATERIAL, IMPACT, MOBILITY, and MODEL statements. When doing this, the MATERIAL parameter of the given statement should be assigned to the chosen material name.

For materials with variations in composition fraction, choose a defined material with X and/or Y composition fractions (i.e., a ternary or quaternary material). You may also find it convenient to use C interpreter functions to define the material parameters as a function of composition.

The C interpreter functions that are useful for this approach are: F.MUNSAT, F.MUPSAT, F.BANDCOMP, F.VSATN, F.VSATP, F.RECOMB, F.INDEX, F.BGN, F.CONMUN, F.CONMUP, F.COPT, F.TAUN, F.TAUP, F.GAUN, and F.GAUP.

When defining new materials, there is a minimum set of parameters that should be defined. This set includes bandgap (EG300), electron and hole density of states (NC300 and NV300), dielectric permittivity (PERMITIVITY), and electron and hole mobilities (MUN and MUP). For bipolar devices, certain recombination parameters should also be defined such as: lifetimes (TAUN and TAUP), radiative recombination rates (COPT), and Auger coefficients (AUGN and AUGP) .

For devices with variations in material composition, certain band-edge alignment parameters should also be defined: either electron affinity (AFFINITY) or edge alignment (ALIGN). If impact ionization is considered, the impact ionization coefficients should also be defined.

As an example, consider the case where a device is simulating with an AlInGaP region. In Table B-1, we see that this material system is not defined in ATLAS. We then choose a material that's defined in ATLAS, which has default material parameters that best approximate the material parameters of the new material. In this case, we choose InGaAsP since, at least for this example, we feel that this material is closest to the AlInGaP. Next, we must specify InGaAsP as the material of the region(s) that is/are composed of AlInGaP. This can be done either on the REGION statement if the structure is defined in ATLAS syntax or from the material menu when the region is defined in DEVEDIT.

Supposing that we're satisfied with the default values of the parameters from the minimum set discussed above and that we're principally concerned with the recombination and heat flow parameters defaults, the following section of the input deck illustrates how these parameter defaults can be modified:

```
# new material AlInGaP
MATERIAL MATERIAL=InGaAsP
# SRH
MATERIAL  MATERIAL=InGaAsP TAUN0=1.1e-9 TAUP0=2.3e-8
# Auger
MATERIAL MATERIAL=InGaAsP AUGN=5.8e-30 AUGP=1.1e-31
# Optical
material material=InGaAsP COPT=1.7e-30
# Thermoconductivity
```

```
MATERIAL MATERIAL=InGaAsP TC.A=2.49
# Heat capacity
MATERIAL  MATERIAL=InGaAsP HC.A=1.9
```

This page is intentionally left blank.

Appendix C: Hints and Tips

This appendix is a collection of answers to commonly asked questions about the operation of ATLAS. This information has been previously published in articles in The Simulation Standard™, Silvaco's trade publication. The original articles can be viewed at Silvaco's home page at http://www.silvaco.com.

Question:

Can ATLAS handle photogeneration from non-normally incident light? What syntax is used to determine the correct photogeneration rate?

Answer:

The models with LUMINOUS allow simulation of photogeneration within ATLAS. Arbitrary light sources are available within Luminous using the BEAM statement. The program uses geometric ray tracing to determine the path of a light beam includingrefraction and reflection of non-normally incident light. It then applies models for the absorption of the light to determine the photogeneration rate.

To use luminous the use should first define a light source using the BEAM statement and then choose the light intensity using a SOLVE statement. The BEAM statement can be explained by a simple example. The following syntax generates the ray trace in Figure C-1.

```
BEAM NUM=1 X.ORIGIN=3.0 Y.ORIGIN=-5.0 ANGLE=60.0 \
MIN.W=-1 MAX.W=1 WAVEL=0.6 REFLECT=2
```

The parameter NUM sets the beam number. LUMINOUS can applications up to 10 independent beams. X.ORIGIN and Y.ORIGIN define the initial starting point for the light beam. This must be outside the device coordinates and for non-normal beams it is important to keep this point well away from the device. The ANGLE parameter determines the direction of the light beam relative to the x-axis. ANGLE=90 gives normal incidence from the top of the device. The light is defined as coming from a line perpendicular to the direction set by ANGLE and passing through (X.ORIGIN, Y.ORIGIN). MAX.W and MIN.W set a window along this line through which the light passes. The default for these parameters is +/- infinity.

The wavelength of the light beam is defined in microns using the WAVEL parameter. It is also possible to specify spectral sources of light by replacing WAVEL with POWER.FILE={filename}, where the {file} is a UNIX text file defining the spectrum in terms of an XY list of wavelength vs intensity.

LUMINOUS automatically chooses the number of rays needed to resolve the geometry. In the case of Figure C-1, only one initial ray is needed. The number of rays used by LUMINOUS is purely a function of the geometry and is not related to optical intensity, photogeneration rate, or MIN.W and MAX.W. The REFLECT parameter is used to limit the number of reflections the light beam is allowed. In ATLAS Version 3.0.0.R an alternative parameter MIN.POWER is used to set a relative intensity below which no more rays are traced.

The user has a choice in LUMINOUS as to whether the rays should reflect from the sidewalls and bottom of the device structure. If the simulation is of a partial wafer (such as is typical for CCD simulation) the light should not reflect. This is the default shown in Figure C-1. If the simulation is of a complete device, such as is typical for solar cells, the light should reflect. The parameter BACK.REFL is used to enable back and side reflection. The result of adding this to the previous syntax is shown in Figure C-2. In this figure the limit set by REFLECT=2 is also clear as each ray reflects only two times.

The ray trace is only done once a SOLVE statement is used to turn on the light beam. For example, the syntax `SOLVE B1=0.5` sets the power of beam #1 to 0.5W/cm^2. DC and transient ramps of light intensity can also be performed.

The refraction and reflection are determined by the real portion of the refractive index for each material. The imaginary portion on the refractive index controls the absorption of the light. Wavelength dependent defaults exist in ATLAS for common materials, but can be defined by the user as follows:

```
MATERIAL MATERIAL=Silicon REAL.INDEX={value} IMAG.INDEX={value}
```

From the ray trace information and the imaginary refractive indices, the photogeneration rate is calculated at each node in the structure. An integration method is used to ensure the charge generated is independent of the mesh density. The photogeneration rate from the ray trace in Figure C-2 is shown in Figure C-3. The electrons and holes generated are included in electrical simulations in ATLAS to determine collection efficiency, spectral response and other device parameters.

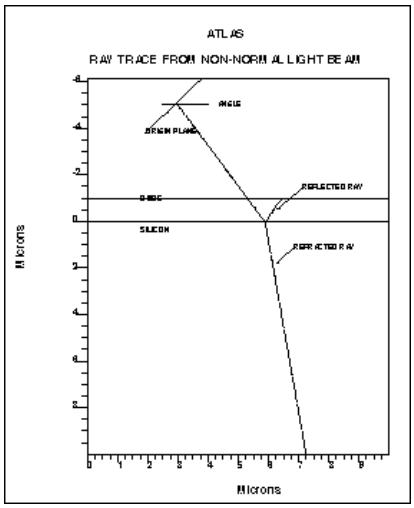


Figure C-1: Simple ray trace in LUMINOUS

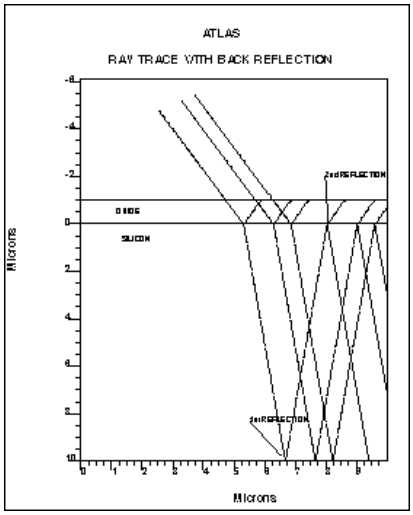


Figure C-2: Addition of back and sidewall reflection to Figure C-1

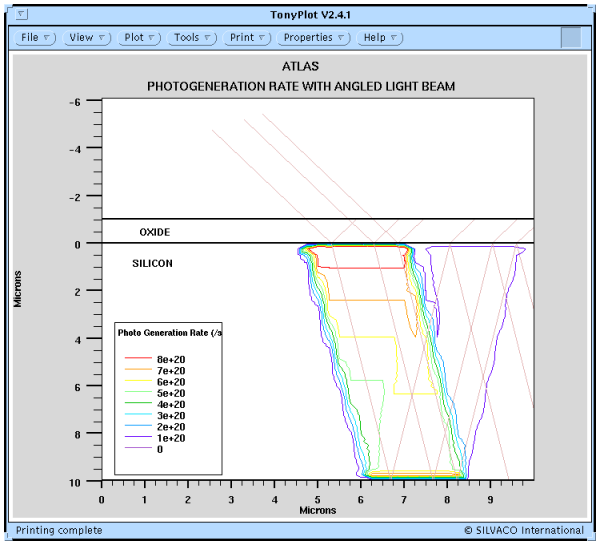


Figure C-3: Photogeneration contours based on ray trace in Figure C-2

Question:

What choices of numerical methods are available in ATLAS? When should each type of method be used?

Answer:

The latest release of ATLAS features more choices of numerical methods for users. It also has a new more logical syntax that clears up some of the previously confusing issues with the choice of numerical method for ATLAS solutions.

ATLAS has the ability to solve up to six equations on the simulation mesh. These are the Poisson equation, two carrier continuity equations, two carrier energy balance equations, and the lattice heat flow equation. The choice of numerical technique in solving these equations can strongly affect both the convergence and CPU time required to complete a simulation run.

In general equations can either be solved in a coupled manner with all equations solved at once or a decoupled manner with a subset of the equation solved whilst others are held constant. The coupled solutions are best when the interactions between the equations is strong (i.e., high current producing significant local heating). However they require a good initial guess to the solution variables for reliable convergence. Unless special techniques, such as projection, are used for calculating initial guesses this would mean that the voltage step size during a bias ramp might have to be rather small.

Decoupled methods can have advantages when the interaction between the equations is small (typically low current and voltage levels). They do not require such good initial guesses for convergence. They tend to either not converge or take excessive CPU time once the interactions become stronger.

ATLAS uses the `METHOD` statement to define numerical methods. The older `SYMBOLIC` statement is no longer required, although existing input files will run as before. To select the decoupled method for two carriers use:

```
method gummel carriers=2
```

To select the coupled method for two carriers use:

```
method newton carriers=2
```

This is the default. In fact users do not need to specify a METHOD statement at all for this case. For the majority of isothermal drift-diffusion simulations this choice of numerics is the recommended one.

ATLAS has the ability to switch automatically between different numerical methods. The syntax

```
method gummel newton
```

would start each bias point using a decoupled method and then switch to a coupled solution. This technique is extremely useful when the initial guess is not good. Typically this happens for devices with floating body effects (i.e., SOI).

For more complex simulation with energy balance or lattice heating other techniques are also available in ATLAS. A mixed technique where the poisson and continuity equations are solved coupled, and then the other equations are decoupled can be applied.

The syntax for this is:

```
method block
```

Typically this mixed technique is quicker and more robust at low lattice and carrier temperatures, whereas the fully coupled technique is better for high lattice and carrier temperatures. The mixed method can be combined very effectively with the fully coupled technique to provide improved speed and convergence for all ranges using:

```
method block newton
```

As an example of this Figure C-4 shows the CPU time taken for individual biasing points for a non-isothermal energy balance simulation of second breakdown of a sub-micron MOSFET. The device was constructed using ATHENA and the mesh contains a low number of obtuse triangles (not zero). The graph clearly shows that at initially the coupled method takes much longer to converge. This is because the initial guess is not good until two bias points are solved so that projection can be used. As the voltage is increased the block method has increasing convergence difficulty whereas the time taken for the newton method is flat. The time for the mixed method has the advantages of block at lower currents, but without the severe increases of block.

The ideal solution for most problems is the use of the mixed "block newton" method initially switching to the fully coupled "newton" method as the simulation proceeds. Of course the cross-over point, which is at x=2.5 in Figure C-4, varies from case to case. To avoid excessive user-interaction the "block newton" could be used throughout the simulation without a very excessive hit in terms of CPU time.

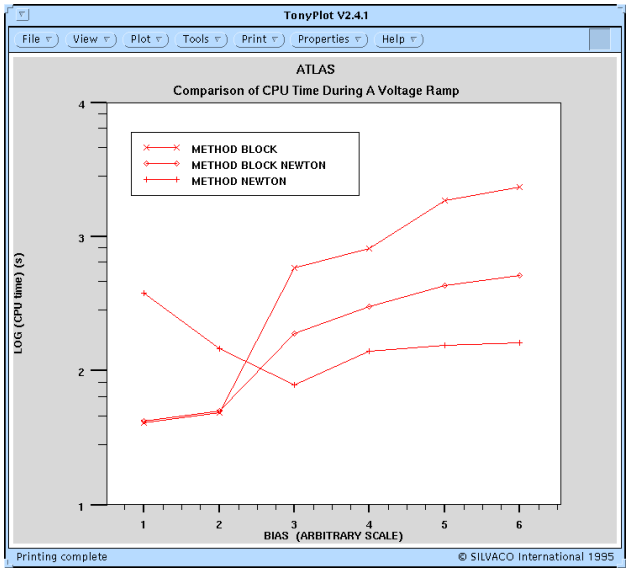


Figure C-4: Comparison of CPU time showing advantages of decoupled methods at low current and coupled method at high currents

Question:

Can the workfunction of the MOS polysilicon gate contact be calculated by ATLAS based on the doping? Can poly depletion effects be simulated in ATLAS?

Answer:

The polysilicon gate contact in MOS devices can be simulated in two distinct ways using ATLAS. These correspond to treating the polysilicon region as:

- A metal-like equipotential region with a specified workfunction.
- A semiconductor region with a potential defined by the doping level.

Most commonly the former approach is adopted. The polysilicon region acting as the gate is defined as an electrode in ATHENA.

The `electrode` statement is used with the X and Y parameters acting as crosshairs to target a particular region of the structure. The whole region, irrespective of shape, is then defined as an electrode.

```
ELECTRODE NAME=gate X={x value} Y={y value}
```

A region defined this way is now treated as equipotential in ATLAS. The potential of this region will be defined by the `VGATE` parameter of the `SOLVE` statement. Hence poly depletion cannot be modeled in gate contacts defined this way. The workfunction of this region must be set by the user. For example, a heavily n doped polysilicon contact can be defined by either of the two following statements:

```
CONTACT NAME=gate N.POLY
CONTACT NAME=gate WORK=4.17
```

The second approach of treating the polysilicon gate region as a semiconductor is achieved by placing a contact on the top of the gate. In ATHENA this is done by depositing a metal (or silicide) layer on top of the polysilicon. The `ELECTRODE` statement is then used to define this metal region as the gate electrode. In ATLAS a workfunction for the gate should not be specified on the `CONTACT` statement as this would give an undesirable workfunction difference between the metal and polysilicon. the

potential on the metal region is defined by the VGATE parameter of the SOLVE statement. The potential within the polysilicon gate region will depend on the doping level of the polysilicon. In non-degenerately doped polysilicon a voltage drop is seen across this region from top to bottom. The workfunction difference between the gate and the substrate can be derived from a potential profile through the channel region.

It is also possible for the polysilicon to be depleted starting at the gate oxide interface. Figure C-5 shows a comparison of high frequency CV curves between a MOS device with a uniform degenerately doped poly gate typical when tube doping is used and a lighter, non-uniformly doped gate typical when source/drain implants are used to dope the polysilicon. In the accumulation region the poly begins to deplete leading to an effectively thicker gate dielectric. This effect is illustrated in Figure C-6. The amount of poly depletion observed is dependent on the doping level. Accurate polysilicon diffusion models are available in ATHENA to simulate the doping. In addition the SILICIDE module allows simulation of the dopant redistribution during gate silicidation. Silicides can typically reduce the effective gate doping making poly depletion more likely.

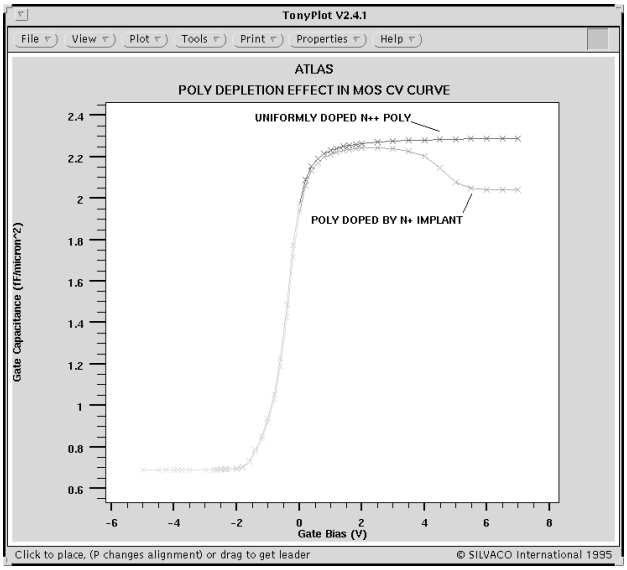


Figure C-5: High frequency CV curve showing poly depletion effects at positive Vgs.

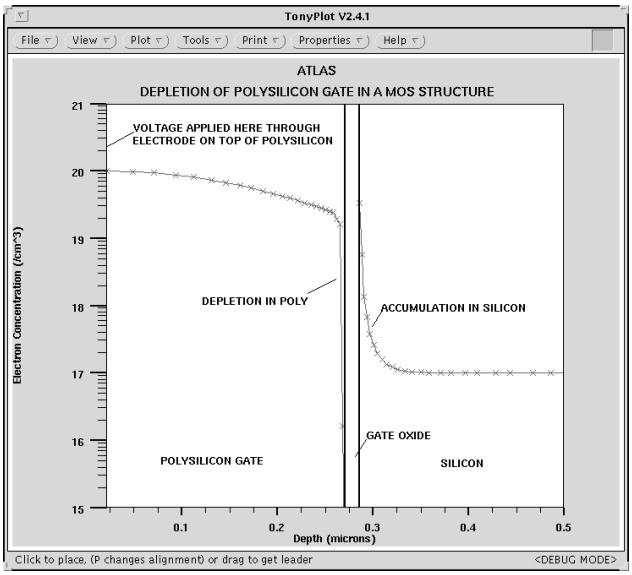


Figure C-6: Figure 2 Electron concentration profile of an NMOS transistor poly depletion occurs at the poly/gate oxide interface.

Question:

How can I remesh my process simulation result for device simulation?

Answer:

The structure editing and griding tool DEVEDIT provides an effective way to remesh structures between process and device simulation.

DEVEDIT is able to read and write files to both ATHENA and ATLAS. In addition to the graphical user interface, DEVEDIT can operate in batch mode under DECKBUILD. The batch mode DEVEDIT features a powerful syntax that can be applied to remesh structures read directly from ATHENA. DEVEDIT employs a heirachical method for remeshing existing structures. Users specify mesh parameters at each stage. Initially a base mesh is applied with the command:

```
BASE.MESH HEIGHT=<h1> WIDTH=<w1>
```

This mesh of h1 by w1 microns provides the coarsest mesh in the structure. On top of this base level mesh DEVEDIT determines which points must be added to ensure the geometry of all regions is preserved. Optional boundary conditioning to smooth region boundaries can be applied using the BOUND.COND statement.

Mesh constraints can be applied to arbitrary boxes with the device using the syntax:

```
constr.mesh x1=<n> x2=<n> y1=<n> y2=<n>\
max.height=<h2> max.width=<w2>
```

This sets the maximum mesh size to h2 by w2 microns the box with diagonal from (x1,y1) to (x2,y2). Using the constraint boxes in critical areas of the device is the most effective way to use DEVEDIT. In MOSFETS the constraint boxes can be applied to the silicon region under the gate. Typically vertical grid spacings of 5 are required for accurate simulation of channel conduction in sub-micron MOSFETs.

Use of multiple constraint boxes can be applied. For MOSFET breakdown constraint boxes can be applied to the drain/gate overlap area.

In addition to constraint boxes, DEVEDIT can refine on quantities such as individual doping species, net doping, potential. The IMP.REFINE statement allows users to select the critical value of a quantity for refinement and an associated minimum grid spacing.

This type of refinement is useful when key effects are located very close to junctions. For example the emitter-base junction in bipolar transistors.

The final level of grid refinement are "manual refine boxes" defined using:

```
Refine Mode={Both|x|y} P1=x1,y1 P2=x2,y2
```

These are boxes with a diagonal from (x1,y1) to (x2,y2), inside which the grid spacing is halved in the given direction. These can be used as a final customization to the mesh. In general, however, the constraint boxes described above are easier to use.

As a demonstration of the effectiveness of regriding using DEVEDIT. Figures C-7 and C-8 show meshes created from the same process simulation structure. Figures C-9 and C-10 show the Id/Vds and breakdown characteristics respectively of the device for each mesh. The mesh at the silicon surface is identical in both cases to a depth of 100A. However the mesh below this is much coarser in Figure C-8.

However the device results show no significant change despite the reduction in the node count by a factor of two.

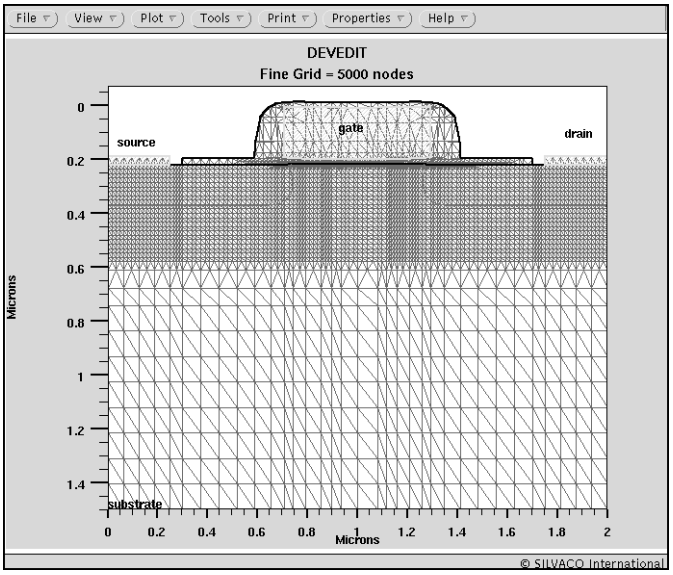


Figure C-7: Fine Grid - 5000 nodes

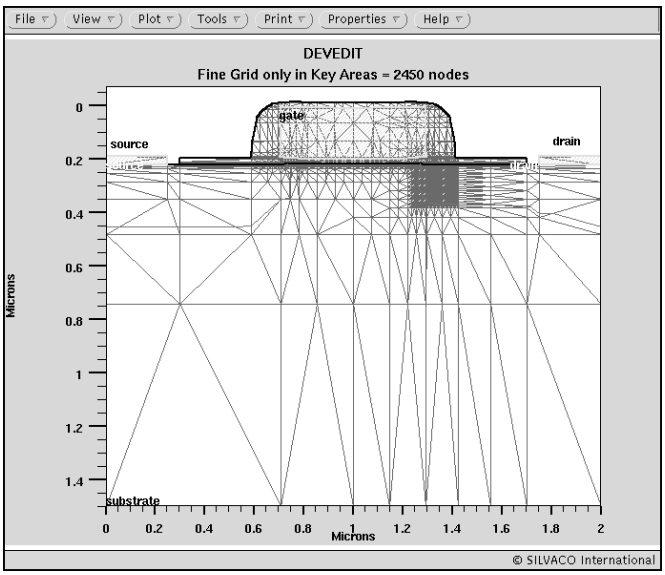


Figure C-8: Fine Grid only in Key Areas - 2450 nodes

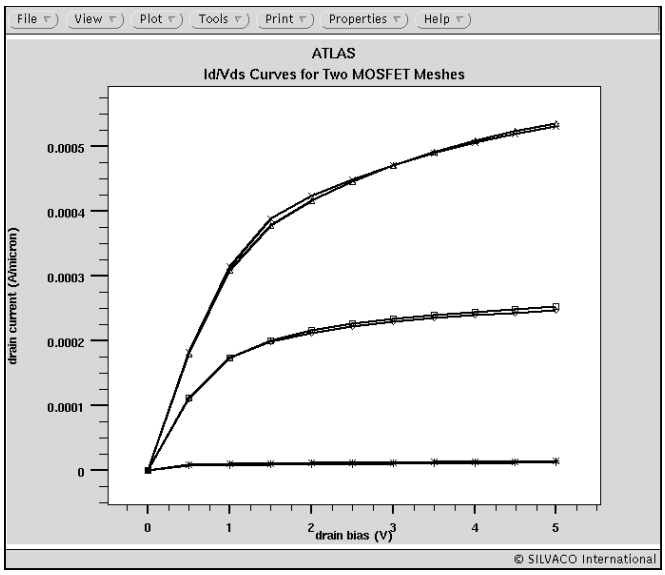


Figure C-9: Id-Vds Curves for two MOSFET Meshes

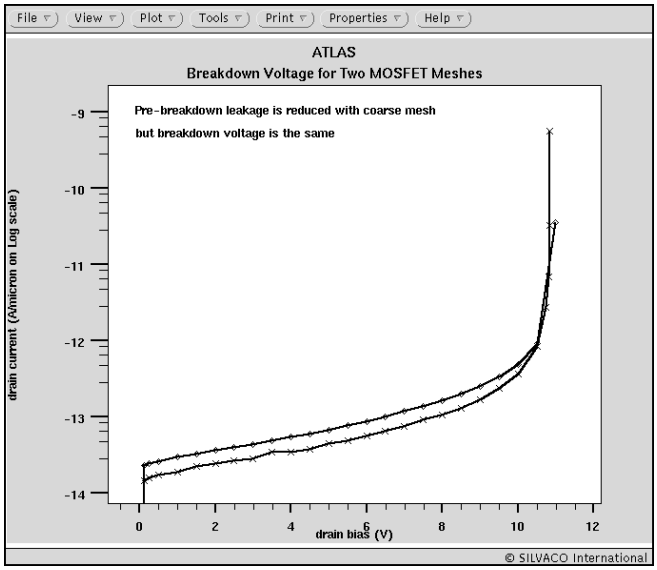


Figure C-10: Breakdown Voltage for Two MOSFET Meshes

Demonstration of optimized griding using constraint boxes in DEVEDIT. Both very fine and optimized mesh produces equivalent results.

Question:

How can solution quantities such as Electric Field be saved for plotting against applied bias?

Answer:

There are two types of output files saved by ATLAS:

- Solution files contain physical quantities mapped to the simulation grid. One solution file is saved per bias point.
- Log files which traditionally have saved the terminal characteristics for all bias points.

A new feature of ATLAS **4.0** is the addition of a capability to save physical quantities at user-specified locations within the device grid to the log files. A new statement PROBE is used to specify the quantity to be saved and the location. For example, to save the electron concentration rate at a location [1,0.1] the syntax is:

```
PROBE NAME=my_e_conc N.CONC X=1.0 Y=0.5
```

The label specified by the NAME parameter is used to label the saved quantity in TONYPLOT and subsequent EXTRACT statements.

For vector quantities the PROBE statement also requires a direction to be given using the DIR parameter. This is specified as an angle in degrees with the X axis as DIR=0. To find the electric field across an oxide layer the syntax is:

```
PROBE X=1.2 Y=0.2 FIELD DIR=90 \ NAME=oxide_field
```

Figure C-11 shows the resultant plot of electric field in a MOSFET gate oxide during a transient ESD pulse. This result shows the probability of oxide breakdown during the ESD stress, without the need to examine many separate solution files.

Another advantage of the probe for vector quantities is that it reads the values directly from the simulator at the closest location to the specified XY coordinates. This avoids many issues of interpolation and averaging of vector quantities onto the simulation grid.

If two physical quantities are probed at the same location it is possible to plot them against each other to examine model settings. For example, impact ionization rate or mobility versus electric field. Figure C-12 shows a plot of channel electron mobility in a submicron NMOS transistor versus the transverse electric field from the gate.

All of the primary solution quantities can be probed. A full list is given in the manual under the PROBE statement. In addition to values at point locations the PROBE statement also supports `MIN` and `MAX` parameters to find the minimum and maximum of a given quantity.

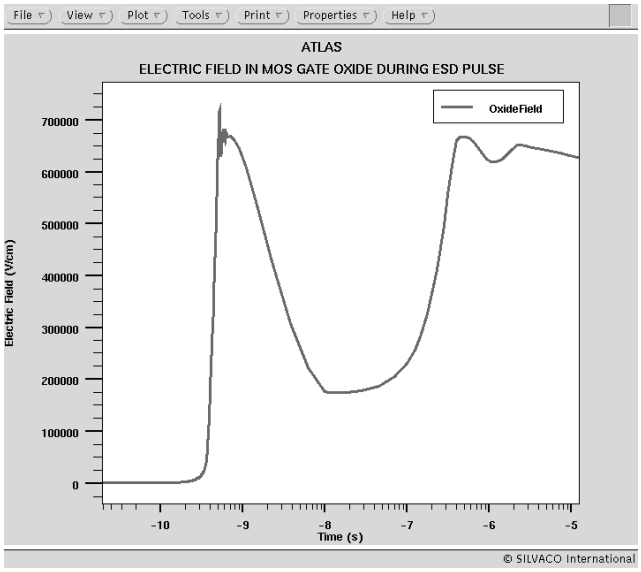


Figure C-11: Electric field in MOS gate oxide during a high current pulse on the drain.

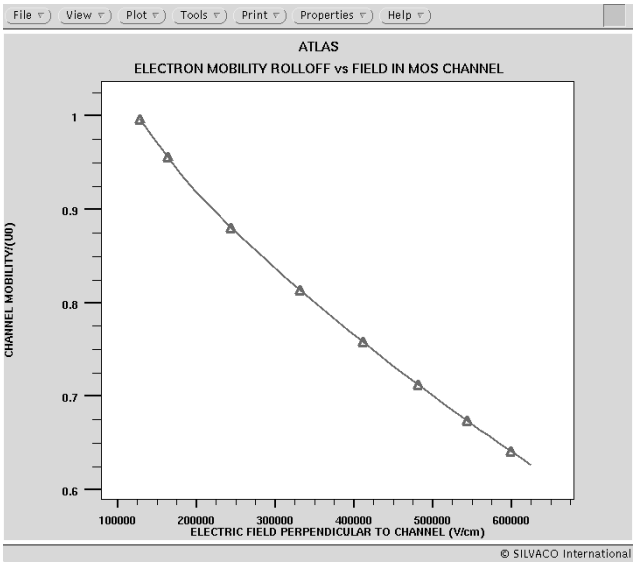


Figure C-12: Mobility (normalized) rolls off as a high gate electric field is applied

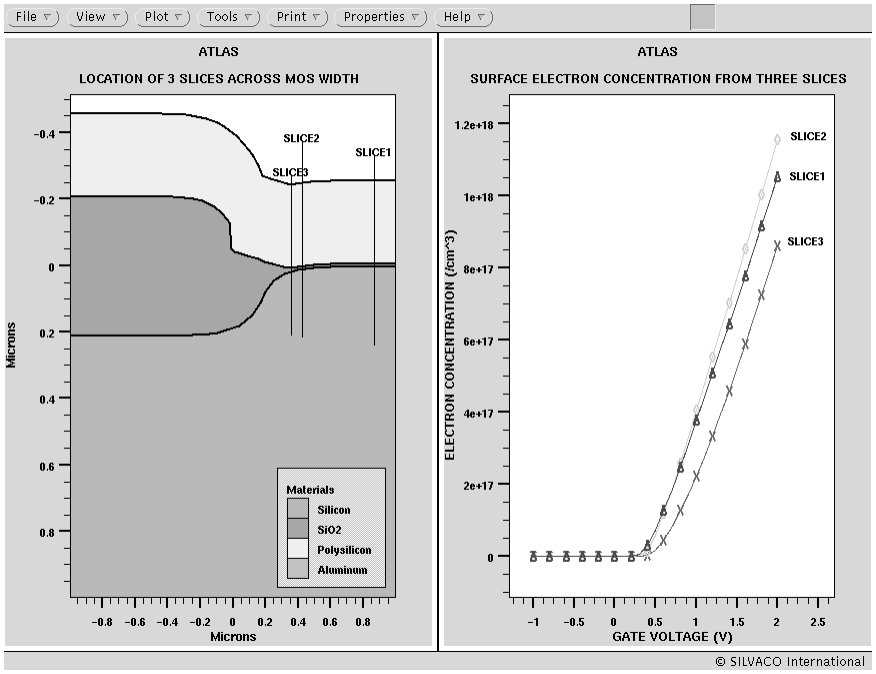


Figure C-13: Using a PROBE of electron concentration allows a study of MOS width effect using 2D simulation. An enhanced electron concentration is seen along slice 2.

Question:

What are the options for generating 3D structures for ATLAS device simulation?

Answer:

Currently there are three options for generating 3D device structures. In all cases the prismatic meshing of ATLAS/DEVICE3D permits arbitrary shaped device regions in 2 dimensions (typically X and Y) and rectangular regions in the other dimension (typically Z).

- Definition though the ATLAS syntax.

This limits the user to defining box shaped regions. Region definition is though statements such as:

```
region num=1 silicon x.min=0  \
x.max=1 y.min=0 y.max=1 z.min=0
\ z.max=1
```

Mesh generation is handled though the `Z.MESH` statement which is analogous to the `X.MESH` and `Y.MESH` statement used in 2D ATLAS simulations. Electrodes and doping can be defined using the same syntax as 2D ATLAS but with `Z.MIN` and `Z.MAX` parameters to control the Z extent. Doping profiles can be read from the formats supported in 2D ATLAS: `ssuprem3, Athena, ascii`.

- Use DEVEDIT to create a 3D mesh structure.

DEVEDIT3D is a 3D structure generation and meshing tool used to generate the mesh, regions, electrodes and doping used in ATLAS/DEVEDIT. It allows users to draw regions by hand in 2D and project them into the third direction.

DEVEDIT3D contains all the sophisticated meshing options available to 2D DEVEDIT. These include: mesh constraints by region, mesh constraints by a user defined box, refinement on doping and other

quantities, mouse controlled refine and unrefine boxes. DEVEDIT3D has both interactive and batch mode operation. This is the recommended approach for 3D power device simulation.

- Use ATHENA and DEVEDIT3D to extend a 2D process simulation to 3D.

A 2D process simulation from ATHENA can be extruded to form a 3D structure using DEVEDIT3D. In this mode DEVEDIT3D can be used to add extra regions and doping if required. This mode us commonly used for modeling MOS width effects in 3D (see Figure C-14). An ATHENA simulation of the isolation bird's beak and field stop implant is performed.

The 2D structure is loaded into DEVEDIT3D and extended in Z direction. The polysilicon gate is truncated in Z and aluminum source/drain contacts are added. The source/drain doping profiles can be read from analytical functions or other process simulation results. A worked example, named mos2ex04.in, is supplied with the Fall 96 release CDROM.

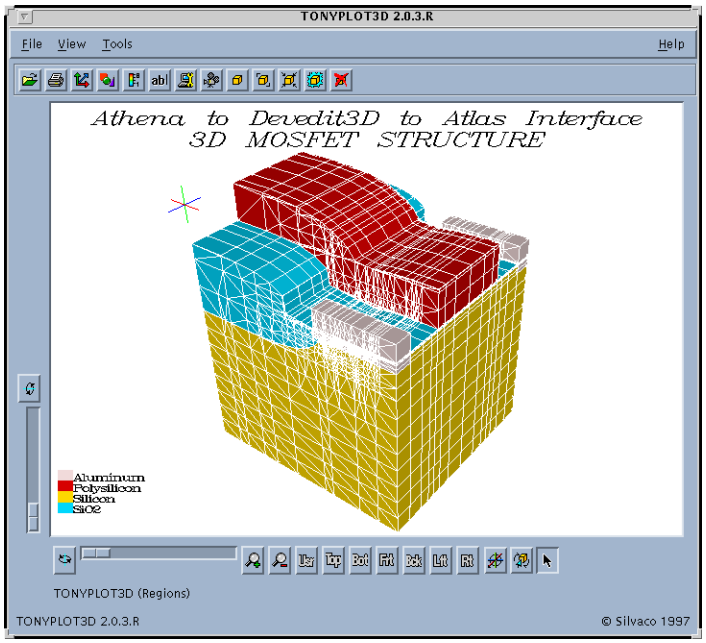


Figure C-14: 3D device simulation of MOS width effect can be performed on structures created Athena.

For devices with non-rectangular regions in the XZ plane such as pillar MOSFETs or cylindrically designed power structures DEVEDIT3D can also be used by drawing in the XZ plane and projecting into the Y direction.

The future release of Silvaco's 3D process/device simulator ODIN will provide a fourth option for 3D simulation problems. ODIN is based on tetrahedral mesh elements and will overcome the mesh restrictions described above.

Note: ODIN is now called Clever.

This page is intentionally left blank.

Appendix D: ATLAS Version History

D.1: Version History

D.1.1: Version 5.10.0.R

General Features

- Introduced input deck syntax for compatibility with other Pisces derivative simulators.
- Introduced FLEXlm licensing.
- Added the ability to probe band-gap.
- Added a C interpreter function for electric field dependent work funciton.
- Improved numerical control over non-linear iteration schemes.
- Introduced two new analytic models for real refractive index for a variety of materials systems.
- Introduced several improved default material models for the InAlGaN system including the effects of bowing and Quaternary composition.
- Introduced a new and improved version of the BICGST iterative linear solver.
- Introduced the ability to grade strain over a region for calculation of gain and radiative recombination using the k.p models.
- Added a new impact ionization rates model for InP.
- Changed the default maximum number of alowable `INTERFACE` statements from 10 to 100.
- Introduced a mobility model to account for carrier-carrier scattering induced by thickness variations in SOI devices.
- Added simple syntax for choosing between various material default parameter sets for calculation of the k.p models for gain and radiative recombination.
- Added simple syntax for choosing between various material default parameter sets for calculation strain dependent polarization.
- Added material parameter defaults for the InAlGaP system.
- Introduced new default impact ionization parameters for the InAlGaN system.
- Introduced a new default radiative rate constant for GaN, InN and AlN.
- Introduced new default models for temperature dependent thermal conductivity and temperature dependent thermal capacity for GaN, InN, AlN, InAaN and AlGaN.
- Made improvements to the Darwish modifications to the CVT mobility model.
- Implemented a new universal band-gap narrowing model.
- Introduced Caughey-Thomas fit for composition dependent mobility in InAlGaN.
- Introduced a field dependent mobility model for the InAlGaN material system.
- Improved convergence of ATLAS3D with energy balance and impact ionization.
- Improved the consistency of the divergence criteria for energy balance between ATLAS2D and ATLAS3D.

New Luminous Features

- Added a new aspherical lenslet in LUMINOUS3D.
- Introduced multilayer anti-reflective coatings in LUMINOUS3D.
- Implemented a new light beam definition in LUMINOUS and LUMINOUS3D where you can define an arbitrary bundle of rays to be traced.
- Anti-reflective coatings can now be introduced on any device interface in LUMINOUS3D.

- Improved calculation of reflectivity and transmissivity in LUMINOUS3D to account for the imaginary part of the refractive index during such calculations.

New MIXEDMODE Features

- Introduced MIXEDMODEXL. This has unlimited circuit nodes, unlimited circuit elements and unlimited ATLAS physical devices.
- Enabled probe compliance in MIXEDMODE.
- Increased the precision in the MIXEDMODE log files to 10 decimal places.
- Introduce transient pulse and tabluar definition into the calculation of time step.

New Quantum Features

- Added output of wavefunctions in REGION based quantum wells.
- Improved control over orthagonality of calculated wavefunctions.
- Introduced a new quantum direct tunneling model for tunneling through thin gate dielectrics.

New VCSEL Features

- Added compositional grading to the DBR regions.

New Features in LED

- Improved LED capabilities to work in cylindrical coordinates.
- Added radiative rate and intensity to the log file.
- Enabled calculation of near field distributions.
- Fixed calculation of output coupling efficiency when 3D nature of light emission is taken into account. Now the results for simple structures agree with analytical estimations (based on escape cone analysis).
- Introduced fast reverse ray-tracing. The new algorithm is based on boundary-to-boundary ray tracing as opposed to element-to-element ray tracing in the old algorithm.
- Output coupling calculations using reverse ray-trace can now account for multilayer anti-reflective coatings.
- Output coupling calculations using reverese ray-trace now account for spectral conten of the emitted light.

New OLED Features

- Improved the stability of the Poole-Frenkel mobility model.

D.1.2: Version 5.8.0.R

Introduced LED

- LED provides new capabilities specifically tailored for the simulation of light emitting diodes.
- Models DC and time dependent responses.
- Uses accurate electronic band structure models based on the k*p method to predict spectral radiative rates.
- Simulates bulk and quantum well devices in all material systems.
- Predicts luminous intensity, peak emission wavelength, spectral output, and efficiency.
- Estimates output coupling and angular output light distribution.

Introduced NOISE

- Calculates equivalent external noise voltage sources for small-signal random behavior of electrons and holes.
- Accounts for random behavior of carrier density and velocity.

New Features of BLAZE

- Added models based on the k*p method to predict band structure and density of states effective masses for Wurtzite InGaN/AlGaN material systems.
- Added models based on the k*p method to predict gain and radiative recombination rates for Wurtzite InGaN/AlGaN material systems.
- Added models to account for piezoelectric effects in the Wurtzite InGaN/AlGaN material systems.
- Added/updated material parameter defaults for numerous material systems.
- Added 3D-SiC as a recognized material.

New Quantum Features

- Added new model/syntax for defining quantum wells. New model is based on the REGION statement and is easier to use and gives greater flexibility in defining quantum wells. The new model also accounts for field effects in spectral characteristics of gain and radiative recombination.
- Added new Bohm quantum potential models in 2D and 3D that allows quantum effects to be included in semi-classical transport modeling.
- Developed Schrodinger-Poisson solver compatible with non-planar and imported structures.
- Added a new eigenvalue solver using the inverse iteration method for Schodinger-Poisson.

New LUMINOUS Features

- Added capability to probe photogeneration in LUMINOUS3D.
- Added capability to output the beam power spectrum.
- Added capability to simulate multi-layer anti-reflective coatings using the transfer matrix formalism.
- Modified calculation of reflection and transmission of light at material interfaces to properly account for imaginary part of refractive index using complex transfer matrix methods.
- Implemented two new types of lenslets for LUMINOUS3D, one ellipsoidal and one a composit of planar front face with cyllindrical edges.
- Added capability to model integrated waveguide photodetectors.
- Moved the complex index of refraction versus wavelength tables into user-modifiable ascii files in the common directory.
- Added capability to simulate diffraction of light and coherent effects using beam propagation method.

New Probe Features

- Added capability to probe a specific region or material for minimum, maximum or integrated value.
- Added capability to probe available photocurrent.
- Added the capability to use raw power spectrum data rather than the default method of integrated averaging.
- Added the capability to probe photogeneration rate.
- Added the capability to probe trap recombination rate.
- Added a compliance capability to the probe.
- Implemented a new initial guess strategy for low/zero current applications in MOS devices.
- Added capability to probe peak emission wavelength for LED.
- Added capability to probe electron and hole velocities.

New VCSEL Features

- Introduced a procedure to check whether a specified VCSEL structure corresponds to general requirements for a VCSEL.

- Added capability to calculate reflectivity of a cold VCSEL cavity as a function of wavelength using transfer matrix method.
- Added capability to find resonanse frequency of a cold VCSEL cavity.
- Implemented a new optical solver for VCSEL based on effective frequency method. The new method allows to simulate oxide aperture VCSELs.
- Added a perturbational method to solve for longitudinal modes of a VCSEL cavity. This method allows to speed up the solution of Helmholtz equation.
- Enabled self-consistent solution for modeling of electrical and optical behavior of VCSELs based on effective frequency method.

New C Interpreter Functions

- Added C-interpreter functions for carrier temperature dependent mobility.
- Added C-interpreter functions for carrier temperature dependent density of states.
- Added carrier concentration to the C interpreter functions forthermionic field emission (tunneling) in Schottkys.
- Added a C-interpreter function for thermal electrode temperature as a function of time for GIGA and GIGA3D.
- Added a C-interpreter function for LASER mirror facet reflectivitiesas a function of wavelength.

New MIXEDMODE Features

- Enabled unlimited circuit nodes and elements in MIXEDMODE and MIXEDMODE3D.
- Implemented BLOCK method in MIXEDMODE.
- Enabled use of modified 2-level Newton scheme in MIXEDMODE.

General Features

- Enabled energy balance/hydrodynamic simulation in 3D.
- Added the capability to modify characteristics of regions in loaded structures.
- Added the capability to load 2D ASCII doping profiles.
- Added quasi-static capacitance calculation in 2D and 3D.
- Implemented curve-tracer in 3D.
- Added capability to print model parameters on a region basis.
- Added random distribution to doping.
- Enabled the use of BLOCK method in transient simulations.
- Reimplemented the Schottky boundary condition with thermionic emission and thermionic field emission.
- Improved the heterojunction thermionic field emission model to work for interfaces in arbitrary directions.
- Enabled region specific specification of the `MOBILITY` statement.
- Enabled transient load and save.
- Added the capability to specify the electron and hole tunneling effective masses to the Schottky boundary model.
- Added the capability to simulate metals as conductors solving a simple conduction equation.
- Implemented new capabilities for user definable materials.

- Made improvements to the Hurkx band-to-band tunneling model.
- Added a new temperature dependent band-gap model.

D.1.3: Version 5.6.0.R

Introduced VCSELS

- VCSELS provides new capabilities that are used in conjunction with BLAZE to simulate cylindrically symmetric vertical cavity surface emitting lasers. This implementation includes numerous new syntactical features, which greatly simplify the specification of complex multilayered epitaxial devices such as VCSELs.

New Features of LUMINOUS and LUMINOUS3D

- Added periodic optical sources to LUMINOUS.
- Added capability to simulate Gaussian sources in LUMINOUS.
- Added capability to simulate Gaussian sources in LUMINOUS3D.
- Added capability to sumilate elliptical sources in LUMINOUS3D.
- Added capability to simulate AC response in LUMINOUS3D.

New Features of LASER

- Added capability to probe laser optical intensity and optical gain.
- Incorporated fully coupled rate equations with device Jacobian.
- Can specify separate front and rear mirror reflectivities.
- Moved all laser models to a new LASER statement.
- Made it so you can specify non-uniform LASER meshes and choose the device mesh for the LASER mesh.
- Added wavelength to the log file.
- Enabled the simulation of symmetric laser structures with half structures.
- Added stimulated recombination to the structure files.

New Features Related to Traps

- Implemented INTDEFECT for interface defects.
- Implemented a bounding box to limit the extent of interface traps or defects.
- Added the capability to probe ionized trap densities.
- Modified continuous and discrete traps to work for both insulators and semiconductors.
- Implemented interface continuous AC traps for ATLAS and ATLAS3D.

New Material System Default Parameters

- Updated material default parameters for the AlGaN system.
- Updated material default parameters for the HgCtTe system.

Miscellaneous Improvements

- Implimented new impact ionization model suggested by Valdinoci et.al.[163].
- Added thermal resistor boundary conditions for energy-balance.
- Added capability to probe impact ionization rates.
- Added capability to probe integrated charge.
- Added capability to specify angled doping profiles in DEVICE3D.
- Added capability to enable "old" Pisces impact ionization model.
- Implemented new discretization scheme suggested by Patil [119]. The scheme promises to avoid the difficulties with obtuse angles inherent in the standard scheme.

- Implemented small signal AC in DEVICE3D.
- Implemented BLOCK method in DEVICE3D.
- Added region and material specific MODELS PRINT capability.
- Implemented CGS, BICGST, and direct solvers in DEVICE3D AC analysis.
- Implemented a new mobility model combining the Klaassen and CVT models.
- Added concentration dependence of surface contributions of mobility.

D.1.4: Version 5.2.0.R

Introduced LUMINOUS3D

- LUMINOUS3D provides a similar functionality to 2D LUMINOUS. This allows simulation of interaction between arbitrary optical sources and semiconductor devices in three dimensions.

Smartlib

- Implemented the SmartSpice Compact Device Model Library (Smartlib). These models replaces the old MIXEDMODE Diode, MOSFET and BJT Models. Additional device models such as TFT and SOI can now be used. Also, additional devices such as JFETS and MESFETs can now be used.

Implemented a New and Improved C-Interpreter

- Implemented a new C-Intrepreter. This completely replaces the old version of the C-Interpreter. The new C-Interpreter is faster than the old version and supports more C functions. The new C-Interpreter conforms more to the ANSI standard.

Hetereojunction Thermionic Emission and Field Emission

- Introduced a new implementation of thermionic emission models for hetero-interfaces. This implementation supercedes the previous one, which is no longer supported.
- Also introduced a new heterojunction tunneling (thermionic field emission) model that is compatible with the new thermionic emission model.

Cyclic Biasing

- Implemented cyclic biasing model to simplify prediction of steady state conditions in cyclically biased devices exhibiting long time constants.

Miscellaneous Improvements

- Added the following new material names: InGaN, AlGaN, InAlGaN, InGaNAs, InGaNP, AlGaNAs, AlGaNP, AlInNAs, AlInNP, InAlGaAs, InAlGaP and InAlAsP.
- Added the following new gain output parameters. ATLAS will now ouput the maximum available gain (Bma) as well as the maximum stable gain (Gms).
- Added capability to add sheet charge along region interfaces for simulation of piezoelectric strain effects.
- Added temperature dependence to the Shirahata mobility model.
- Added capability for multiple trapazodial/square transient pulses.
- Added a more general parameterized temperature-dependent model for saturation velocity for field-dependent mobility.
- Changed the calculation of effective mass used in calculation of thermal velocity for traps in noo-silicon materials. The new approach calculates mass, based on the material density of states.

D.1.5: Version 5.0.0.R

Introduced several new 3D products

Giga3D.

- GIGA3D contains most of the functionality of the 2D Giga, but works with 3D products. This allows modeling of heatflow and self-heating effects in 3D devices. The only functionality not supported in this version of GIGA3D that is supported in 2D is the BLOCK method.

Blaze3D

- This version accounts for spatial variations in band gap due to spatial variations in material composition in 3D. This version supports all the same models that are supported in Blaze 2D with the exception of thermionic emission at hetereojunctions and energy transport. This version also does not support compositional variation in the z direction.

MixedMode3D

- This improvement allows simulation of 3D devices embedded in lumped element circuits. MIXEDMODE3D contains all the functionality of 2D MixedMode simulator.

TFT3D

- This model allows modeling of poly and amorphous semiconductor devices such as TFTs in 3D. This model has all the functionality of the 2D TFT simulator.

Quantum3D

- This allows modeling of the effects of quantum confinement using the quantum moment approach. This model has all the functionality of the 2D Quantum model.

Orchid

- ORCHID is a new product used in conjunction with S-PISCES to model the effects of total radiation dosage on MOS device reliability. Key to this capability is the ability to model the gate oxide as a semiconductor. The user may then define carrier pair generation in the oxide as a function of position, time, and electric fields, using a new C interpreter function. Another C interpreter function can be used to calculate the cumulative fixed charge density as a function of position and electron and hole current densities in the oxide. The deduced fixed oxide charge density directly effects the threshold degradation of the device.

Introduced SiC

- SiC is a new simulation capability for simulating anisotropic materials such as silicon carbide. The basis of this model is the incorporation of a new anisotropic mobility model that allows users to define separate mobilities along normal directions.
- Also added a new default value for thermal conductivity and electron saturation velocity of silicon carbide.

Modifications to Laser

Included additional functionality into Laser. The new features are:

- transient mode (time-dependent photon rate equation)
- far field patterns
- addtitional loss mechanisms including:
- mirror loss
- free carrier loss
- absorption loss (including C interpreter function for bulk absorption)

New Large Signal and Simulation Capabilities

- Added specifications of sinusoidal waveforms to the SOLVE statement.
- Implemented post processing Fourier analysis. This performs a Fast Fourier Transform (FFT) on data within a log file to convert time domain data into frequency domain data.
- Added solution compliance based on sinusoidal analysis. In this mode, transient simulation will stop

if the amplitude of a sinusoidal waveform is less than the user-specified value.

Modifications to Luminous

- Added capability to print the complex indices of refraction as they are calculated for the various materials during the ray trace in Luminous.
- Added functionality to simulate single-layer antireflective coatings in LUMINOUS. This capability calculates the reflection coefficient for a single layer coating under conditions of normal incidence.

Modifications to MixedMode

- Added the capability to specify multiple .TRAN statements in a single input deck.

  Added the extraction of network parameters (S, Z, Y, H, ABCD), stability factor, unilateral power gain, and maximum unilateral transducer power gain in MIXEDMODE.
- Added the capability to use the PROBE in MIXEDMODE.

Modification to GIGA

- Enabled simulations of floating gate structures with heat flow in GIGA.

New Physical models

- Added Klaassen's concentration and temperature-dependent SRH lifetime and Auger rate models.
- Implemented new continuous trap level model in TFT.
- Added new carrier concentration dependent Auger coefficient model.
- Added the capability to calculate the band-to-band tunneling coeffiicients from principles.
- Added the capability to model thermionic field emission tunneling in Schottky contacts.
- Added two new parameters to the single-event upset model. The generated electron/holes pairs (or charge) will not be calculated after tf seconds if this parameter is specified in the input deck. The width of the beam can now be specified as a constant value, followed by an exponential or Gaussian function by using the beam.radius parameter. For radi greater than beam.radius, either the Gaussian or exponential functions is used (if they are specified).

New Numerical Techniques

- Two sparse matrix reordering techniques are now available: Minimum Degree (MD) and Reverse Cuthill McKee (RCM). They are used in the 2D linear solver. MD is the default. Choosing RCM instead of MD sometimes solves the problem of having a zero on the diagonal, which is indicated by the error message:

  ```
  Internal error in linear solver:  Z-#.
  ```

  If this happens, switch from MD to RCM or vice-versa.
- For Parallel ATLAS, two new mesh partitioning techniques are now available:
  - Natural Ordering (NO) and
  - Multi-level Graph Partitioning (MGP)

  These are used to assign sub-meshes to different processor. MGP is the default and should always outperform NO. On the other hand, NO requires less preprocessing time and might perform as well as MGP in the case of meshes generated by the ATLAS statement.
- Made it possible to use projection for initial guesses in conjunction with current boundaries.

Structures and Log File Improvements

- Added the capability to write X and G (rhs) errors in a standard structure file for subsequent visualization using TONYPLOT.
- Added the capability to save electron, hole relaxation times, and Peltier coefficients to structure files.
- Added the capability to save electron, hole, and displacement currents to log files.

New Quantum models

- Implemented a new self-consistent Schrodinger-Poisson solver. This model interactively solves Poisson's and Schrodinger's equations to calculate the potential and carrier concentration, including the effects of quantum confinement. These solutions also produce eigen energies and eigen functions.
- Added Hansch's model for quantum mechanical correction for carrier distributions in N channel MOSFETs inversion layers.
- Added van Dort's model for quantum mechanical correction for carrier distributions in N and P channel MOSFETs in inversion and accumulation layers.

Miscellaneous ATLAS Improvements

- Reconfigured small signal AC post processing to include the effects of parasitics in gain calculations.
- Changed functionality of C interpreter functions, `F.CONMUN` and `F.CONMUP`, to account for donor and acceptor concentrations separately.
- Added trap concentrations to concentration-dependent mobility models and band gap narrowing models.
- Added new C interpreter functions for energy-dependent relaxation times for electrons and holes.
- Added new C interpreter functions for energy-dependent Peltier coefficients for electrons and holes.
- Added a new C interpreter function to specify doping in 3D as a function of position.
- Added extra checking to the AC memory allocation. If there is insufficient memory for the factorization, more will be allocated. A warning message will also be printed.
- Enabled multiple contacts with the same name to have the same work function and other parameters.
- Modified the DOPING statement so that ATHENA master files can now be used in 2D. ATHENA doping is interpolated onto the ATLAS mesh.

D.1.6: Version 4.3.0.R

Introduced Quantum Transport Model

This model accounts for redistribution of carriers near abrupt material transitions due to the effects of quantum confinement. It is also useful when simulating heterojunction devices such as HBTs and HEMTs, as well as short channel thin gate oxide MOS devices.

Introduced Improvements to the Ferroelectric Model

This release includes modifications which allow better modeling of unsaturated loop simulation of ferroelectric materials. It also includes the capability to characterize the AC ferroelectric permitibity affected by the polarization as a fraction of the DC case.

Log File Improvements have been Made for Capturing Data

- PROBE has been enhanced to add the capability of probing minimums and maximums of the directional quantities; electric field, electron mobility, hole mobility and polarization.
- PROBE has been enhanced to allow the measurement of dielectric permitivity and the band-to-band tunneling rate.
- Added the capability to capture simulation time in the log file.

Structure File Improvements

The default method of averaging for capture of impact ionization rate, the electron and hole mobilities to the structure file has been changed. The new averaging techniques provide smoother plots in TONYPLOT.

Improved Parameterization of Band-to-Band Tunneling Models

This release allows user accessibility to the gamma parameter of the standard band-to-band tunneling model.

D.1.7: Version 4.0.0.R

Introduced a ferroelectric permitivity model

This model accounts for ferroelectric polarization and hysteresis. This release includes a new model to model ferroelectric materials. It is useful for simulating ferroelectric devices such as ferroelectric capacitors and FETs.

Introduced Klaassen's mobility model

This release includes a new model for low field electron and hole mobilities in silicon. This model accounts for lattice, impurity, and carrier scattering and screening as a function of temperature, doping and carrier concentrations. This model has been calibrated over a wide range of conditions and should be used as the default low field model for silicon simulations.

Introduced Concannon's gate and substrate current models

This release includes a new model for EPROM simulation. This model estimates the gate current in EPROM devices as a function of carrier temperature by integrating a non-maxwellian distribution function. The model also uses a impact ionization model that uses the same distribution function to ensure a direct relationship between gate and substrate currents.

Introduced Shirahata's mobility model

This release includes a new surface mobility model which accounts for carrier screening effects in the channel. This is done largely by using Klaassen's model for the low field mobility. The model also accounts for parallel and perpendicular fields in the channel.

Introduced a "Probe" capability

A new statement has been implemented which allows users to write local scalar quantities (e.g. potential, electron concentration, hole concentration, etc.) and vectors (e.g. field etc.) to the log file as a function of bias, time etc. The user may specify the probe by location, minimum or maximum, and type.

Modifications to the Watt mobility model

Several modifications were made to the Watt mobility model. First, the user may now apply the model to grid nodes below the interface down to a user specified depth. Also the user can use an exponential scaling on the effective field. Finally, the user can make the models only apply to minority carriers.

Improved vector averaging

Certain quantities used in ATLAS are not defined on grid points but between points (e.g. electric fields). To write these quantities into the structure files they must be averaged. In this implementation an improved averaging scheme was introduced which produces smoother results. The user also has control over various weighting schemes for the averaging.

Traps in MIXEDMODE

New parameters were introduced to the `TRAP`, `INTTRAP`, `DEFECT`, and `INTERFACE` statements to enable proper handling of these physical models in MIXEDMODE.

Modified run time output

The ATLAS run time output was modified to be more concise and informative.

Misalignment and delta CD in INTERCONNECT3D

INTERCONNECT3D now allows users to specify misalignment and variations in critical dimensions with interconnects specified by mask sets from MASKVIEWS.

Lumped elements and current boundary conditions in DEVICE3D

DEVICE3D now supports lumped element and current boundary conditions.

GaAs in DEVICE3D

DEVICE3D now supports GaAs material models.

Electric field lines.

Electric field lines can now be plotted in TONYPLOT. The specification of where to elines lines are calculated was improved for both plotting and for evaluation of ionization integrals.

Energy dependent impact ionization coefficients.

Two new C interpreter functions were introduced to enable user specification of impact ionization coefficients as a function of carrier temperature for Selberherr's model.

Removed obsolete syntax

The `FIT`, `COLOR`, `CONTOUR`, `LABEL`, `LOOP`, `L.END`, `PLOT.1D`, `PLOT.2D`, and `PLOT.3d` statements are no longer supported by ATLAS. Equivalent functionalities are available through DECKBUILD and TONYPLOT.

DEVICE3D improvements for SEU simulation

Introduced a new C interpreter function, `F3.RADIATE`. This function allows users to specify arbitrary generation rate versus spatial position and time. The `SINGLEEVENTUPSET` statement was changed to allow users to control whether the generation rate is scaled by the ratio of the numerical and analytic integrals of the analytic generation function. This permits conservation of total generation rate in a device. By default the rescaling is now turned off.

MIXEDMODE Improvements

The `.DC` statement now allows linear and logarithmic stepping of sources and nested sweeps.

D.1.8: Version 3.0.0.R

First release of "six equation solver" version of ATLAS.

In this release an advanced Non-Isothermal Energy Balance Model was implemented. The model allows self-consistent solution of a set of up to 6 partial differential equations for electrostatic potential, electron and hole concentrations, electron and hole carrier temperatures, and lattice temperature. All the important combinations of equations are supported.

DEVICE3D introduced.

DEVICE3D is introduced as the first truly 3D simulator in the ATLAS framework. It is a virtual analogy to S-PISCES. With it three dimensional steady-state silicon device simulation can be performed to calculate electrical behavior of bipolar and MOS transistors, as well as EPROM devices.

- Syntactically and behaviorally like ATLAS S-PISCES.
- Poisson, one carrier or two carrier analysis can be performed.
- Uses Gummel, Newton or combined Gummel/Newton algorithms.
- Supports semi-implicit scheme for transient simulation. Significantly faster than TR-BDF.
- Supports the essential set of physical models for silicon modeling including: SRH recombination,

Auger recombination and radiative recombination; concentration dependent mobility models (Conmob,Arora, Analytical); longitudinal and transverse electric field dependent mobility models

(Caughey-Thomas, CVT, Yamaguchi); Band Gap Narrowing (BGN), Boltzman statistics, Fermi-Dirac statistics, and incomplete ionization; Fowler-Nordheim current calculation, Lucky electron model for hot carrier injection and transient programming and erasing of EPROM devices.

- The three dimensional simulated structure can be specified using DEVEDIT3D.
- Uses TONYPLOT3D for postprocessing.
- Single Event Upset simulation is feasible with DEVICE3D. The capability of single event upset/ photogeneration transient simulation has been introduced. It allows specification of multiple tracks and radial, length and time dependence specification of generated charge along tracks.

THERMAL3D introduced.

THERMAL3D is introduced as a new capability for three dimensional simulation using ATLAS. The Poisson equation for temperature is solved in three dimensional regions of specified heat conductivity with defined temperatures on thermal contacts (which are ideal heat conductors). The user may specify the thermal properties of regions heat sources, and heat sinks.

- Four models of temperature dependent thermal conductivity.
- User may specify independent temperatures on all thermal sinks (thermal electrodes).
- User may specify multiple independent heat sources.
- Completely compatible with ATLAS.
- The simulated structure is specified using DEVEDIT3D (including thermal electrodes).
- Temperature distributions can be examined using TONYPLOT3D.

INTERCONNECT3D introduced

INTERCONNECT3D has been introduced as another 3D capability in the ATLAS framework. This simulator is used to extract parasitics from arbitrary 3D interconnect structures.

- Can extract parasitic capacitance and conductance for arbitrary 3D interconnect structures.
- Provides an interface to MASKVIEWS for simple mask layer definition of interconnect structure.
- Completely compatible with ATLAS.

Curve tracer introduced

Allows automatic tracing of complex IV behavior (including functions that are multivalued) without user intervention. Automatically varies an external resistance on a given contact to ensure optimal steps are chosen.

Enhancements for Energy Balance Model

Aside from the introduction of the "Six Equation Solver", several other improvements were made with respect to the Energy Balance Model (EBM).

- Fermi statistics can be used with the EBM.
- The "GaAs" like mobility model can be used with the EBM.
- The energy relaxation length can be specified for impact ionization in the EBM.
- Carrier temperature projection for initial guesses is supported.

D.1.9: New solution methods

GIGA improvements

Aside from the introduction of the "Six Equation Solver", several other enhancements were made to GIGA.

- GIGA now allows self consistent 5 and 6 equation solutions of lattice temperature with electron and hole temperatures.
- GIGA now supports static projection for initial guesses for lattice temperature.

LUMINOUS improvements

In this release many improvements were made to LUMINOUS.

- LUMINOUS now supports extraction of spatial response to optical sources.
- A MIN.POWER parameter of the BEAM statement can now be used to limit the number of rays traced in LUMINOUS.
- LUMINOUS now recognizes metals (electrodes) as bulk regions.
- Optical properties of bulk metal (electrodes) region can now be defined.
- The C Interpreter function RADIATE for user definable photogeneration was enhanced to include time dependence.

TFT Improvements

Several enhancements were made in this release for TFT.

- Parameters were introduced to allow control of the number of mid-gap states used for TFT simulations.
- TFT now supports output of donor and acceptor trap state densities and energies to user specifiable files for examination by TONYPLOT.
- TFT now provides two new Interpreter functions for specifying the donor and acceptor densities as a function of energy.

Miscellaneous new features

A host of new features were introduced into ATLAS in general.

ATLAS now properly handles touching electrodes and electrodes with the same name.

Incorporated the trap assisted tunneling model given in Hurkx, G.A.M., Klassen, D.B.M. and Knuvers, M.P.G., "A New Recombination Model for Device Simulation Including Tunneling", IEEE Transactions on Electron Devices, V. 39, No. 2. Feb., 1992, pp. 331-338. This model accounts for tunneling via trap transitions using an analytic approximation that modifies the lifetimes in the standard Schockley-Read-Hall recombination model.

Modified the models for hot electrons, and holes and Fowler Nordheim tunneling currents. The new models account for the flow of the electrons/holes in the insulating layers to arbitrary electrodes based on one of two primitive models. The first, is the default model and trys to force the currents to follow the maximum potential gradients to find the proper electrode. In the second model insulator, currents go the the nearest electrode. With these modifications the electrode current run time outputs are modified to better reflect the new models.

Enabled correct small signal simulation when simulating with bulk and interface traps.

Distinction made between active and chemical dopant densities.

Small signal analysis with interface traps now works properly.

Moved the C Interpreter function F.COMPOSIT from the MATERIAL statement to the DOPING statement. In previous releases this function did not work as advertised, since material composition needed to be defined at structure creation. In order to make the function work properly it had to be moved to the DOPING statement.

In addition to CLIMIT parameter which existed earlier, new parameters CLIM.DD and CLIM.EB were added in METHOD statement. CLIM.DD is exact analog of CLIMIT parameter, but in comparison with CLIMIT it specifies dimensional value of minimal concentration which can be resolve using ATLAS. By default CLIM.DD is not specified and ATLAS uses default value of CLIMIT, as in previous versions. Parameter CLIM.EB can be treated as regularization parameter for the case of very small electron (hole) concentration for energy balance model. It specifies the mimimal value of concentration for which relaxation term in energy balance equation will be still properly resolved, otherwise temperature for the points where concentration is much less than CLIM.EB, will tend to lattice temperature.

New syntax

With this release, the ATLAS syntax was augmented to simplify and somewhat rationalize some of the more confusing problems.

- The SYMBOLIC statement is no longer needed.
- The numerical method can now be specified on the MODEL statement. The user may choose from Gummel, Block and Newton methods. More than one method can be chosen; and ATLAS provides automatic switching between the methods.
- ATLAS now supports a new syntax for specifying which equations are simulated. The syntax is now supported on either the MODEL or METHOD statement and is the preferred syntax for future compatibility. The new syntax provides a more rational, intuitive way of choosing which equations are solved.
- ATLAS still supports syntax from previous versions.

Improved run-time output

Many improvements have been made to the run-time outputs of ATLAS.

- Added multiple levels of verbosity.
- Improved printing of mesh statistics, region and electrode summaries and models.
- Printing of error messages was made more consistent and informative in many places in ATLAS.
- Removed spurious printout from S parameter extraction with the UTMOST statement.

New platforms

ATLAS has now been ported to DEC Alpha and SGI workstations.

Miscellaneous bug fixes

In this release many bugs were removed that existed in the previous version.

- Fixed bug that caused setting the value of the MODEL statement parameter CCS.EA to also set the value of the MATERIAL statement parameter COPTR.
- Fixed bug causing improper association of nodes with regions on BLAZE structures read in from DEVEDIT. In some instances nodes j would be assigned a region at the interface then the composition fraction (e.g.) from the opposing region, thus causing distortions in the band diagram at the interface.
- Fixed bug in static projection. Bug caused by two consecutive bias steps of opposite polarity.
- Fixed bug in TFT. Caused the midgap Gaussian donor defects to have the same distribution as the midgap acceptor defects.
- Fixed bug in UTMOST interface. Bug caused ATLAS not to recognize routines 2, 3, 5, 12, 13, 14, and 26.
- The calculations for the electron and hole mobilities in the InGaAsP system were found to be in error both in the manual as well as the code. Referring to the ATLAS Users manual, Edition 2, in the Equations B-36 and B-37, the "y"s should be replaced by (1-y). This fix was introduced into the code for this release.
- Fixed a bug in the calculation of electric fields for output to structure files for examination in TONYPLOT. This bug caused the estimates of the X and Y components of the electric field to be calculated based on the dielectric constant of silicon for all semiconductors. Although, the calculation properly handled insulators it used the dielectric constant of silicon for all semiconductors regardless of composition. This has been fixed in this version.
- Fixed bug causing extra space in UTMOST format output log files.

New examples

In this release the standard examples were completely reworked and improved. The ATLAS examples are now divided into 21 application specific areas. There are now 117 ATLAS examples in all which illustrate virtually all the functionality of ATLAS.

Improved quality control

In this version, new quality control tools and procedures for automated testing were implemented and used that greatly simplify and expanded ATLAS testing. This version of ATLAS represents the most thoroughly tested version to date. Continued application of the improved quality control procedures will ensure that future versions will continue to be of the highest quality.

D.1.10: Version 2.0.0.R

- **Improved Standard Structure Output** – The standard structure output format has been improved to account for multiple values at region interfaces (e.g., contours projecting into regions where they did not apply).
- **Reliable Metal Regions** – Fixed bug that caused metal regions on standard structure format structures that are read into ATLAS to be lost following modification in ATLAS by either addition of electrodes or regridding.
- **MEASURE Statement** – Because of conflicts with the DECKBUILD extract compatibility syntax, the ATLAS EXTRACT statement has been changed to MEASURE.
- **Error Messages** – Fixed bug that caused infinite loop of error messages during write of structures containing isolated triangles.
- **Larger Allocation** – Fixed bug for "large" version (altas-E) for dynamic allocation to allow up to 6,000 nodes.
- **Ramping** – Fixed bug for case where frequency is ramped in AC analysis during a static ramp. The frequency is now properly reset to its base value after each static solution.
- **Maximum Time Step** – Added DT.MAX parameter to the METHOD statement to limit the maximum time step size during transient solutions.
- **Electron Current Vectors** – Fixed bug that caused electron current vectors to point in the wrong direction in TONYPLOT.
- **Ionization Printout** – Removed spurious printout during ionization integral calculation.
- **Incomplete Ionization** – Fixed bug in incomplete ionization. Bug caused lack of convergence.
- **Interpreter** – Fixed bug in F.DOPING Interpreter function for specifying doping.
- **Material** – Fixed bug to allow Interpreter functions in material statement to be defined for a given material.
- **Region Interface** – Fixed bug to resolve ambiguities in definition of region membership of nodes at interface between two regions.
- **Holes** – Fixed bug that caused holes in regions or embedded regions to be improperly displayed in **TonyPlot**.
- **Trap Recombination** – Fixed bug in recombination calculation for regionally specified traps.
- **Transient Projection Algorithm –** Installed transient projection algorithm.
- **Improved Interpreter Function for Complex Index of Refraction –** The C Interpreter function for complex index of refraction was changed to account for material composition and temperature.
- **New Built-in Index Data –** Added built-in complex index of refraction data for: AlAs, GaAs, InSb, InP, Polysilicon, and Si02.
- **Light Intensity Contours** – Added interface to TONYPLOT for optical intensity. To enable, set the OPT.INTENS parameter of the OUTPUT statement.
- **New C Interpreter Function –** Added C Interpreter function for general generation rate as a

function of position.

- **Cylindrical Coordinates – LUMINOUS** now supports cylindrical coordinates. Specify CYL on the MESH statement.
- **Energy Balance Modeling – LUMINOUS** is now supported in energy balance simulations.
- **Loading and Saving Optical Sources –** LOAD and SAVE statements act to save and restore optical sources.

BLAZE Version 2.0.0.R

- **Improved DevEdit Composition:** Fixed bug that caused DEVEDIT composition fraction information to be lost when the DEVEDIT structure is loaded in MESH statement followed by any ELECTRODE statement.
- **Composition Fraction Modification:** Fixed bug to allow composition fraction modification on loaded structures.
- **Thermionic Emission:** Added capability to specify thermionic emission for heterojunctions in loaded structures.

GIGA Version 2.0.0.R

- **Quasi-Fermi Level:** Fixed bug that caused improper display of the hole quasi-fermi level in **GIGA** generated structure files.
- **Outer Block Iteration:** Fixed bug in GIGA that caused premature exit from the outer block iteration when the maximum temperature update exceeded the starting temperature during the initial solution.
- **Material Thermal Parameters:** The ability to specify the GIGA thermal parameters dependently for different materials/regions was added in this version.
- **Improved Thermal Dependencies:** The temperature dependencies of many models were improved.

LASER Version 2.0.0.R (Initial Release of LASER Under The ATLAS Framework)

- This version supports simulation of various types of Fabry-Perot semiconductor diode lasers.

BLAZE Version 2.0.0.R

- Now supports running MIXEDMODE decks under DECKBUILD.
- Implemented small-signal AC simulation in MIXEDMODE.
- **Energy Balance for Heterojunctions:** Modifications to to the ATLAS energy balance formulas were introduced to account for positionally dependent band structures.
- **Improved Material Defaults:** Some material model parameter defaults were changed to reflect better models in literature. In particular, these improvements affect InGaAsP and related compounds, AlGaAs, and SiGe.
- **Improved DevEdit Interface:** Donor and acceptor impurities were added to the DEVEDIT interface for doping specification in DEVEDIT.

MIXEDMODE Version 2.0.0.R

- User-definition of how often solutions are saved during transient simulation
- The cylindrical coordinate system may now be used with MIXEDMODE.
- Schottky contacts are now supported in MIXEDMODE.
- Tabulated time-dependent voltage and current sources may be specified.
- Previously calculated device solutions can be loaded into MIXEDMODE.
- Circuits may include lossless transmission lines.
- User-defined two-terminal elements may be specified.

TFT Version 2.0.0.R

- User Specifiable Thermal Conductivities of Metals: GIGA now recognizes metals (electrodes) as different materials from semiconductors and insulators. Users can now specify metal thermal conductivity.
- Regional DOS Specification: Added `REGION` parameter to the `DEFECT` statement to allow DOS to be defined differently in different regions.

This page is intentionally left blank.

Bibliography

1. Adachi, S. *Physical Properties of III-V Semiconductor Compounds InP, InAs, GaAs, GaP, InGaAs, and InGaAsP*. New York: John Wiley and Sons, 1992.

2. Adachi, S., "Band gaps and refractive indices of AlGaAsSb, GaInAsSb, and InPAsSb: Key properties for a variety of the 2-4 um optoelectronic device applicatios", *J. Appl. Phys.*, Vol. 61, No. 10, 15 (May 1987): 4896-4876.

3. Adamsone A.I., and B.S. Polsky, "3D Numerical Simulation of Transient Processes in Semiconductor Devices", *COMPEL—The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* Vol. 10, No. 3 (1991): 129-139.

4. Akira Hiroki, S. Odinaka et. al., "A Mobility Model for Submicrometer MOSFET Simulations, including Hot-Carrier Induced Device Degradation", *IEEE Trans. Electron Devices* Vol. 35 (1988): 1487-1493.

5. Albrecht, J.D. et al, "Electron transport characteristics of GaN for high temperature device modeling", *J. Appl. Phys.* Vol. 83 (1998): 4777-4781.

6. Allegretto, W., A.Nathan and Henry Baltes, "Numerical Analysis of Magnetic Field Sensitive Bipolar Devices", *IEEE Trans. Electron devices*, Vol. 10, No. 5 (1991): 501-511.

7. Ambacher, O. et. al., "Two Dimensional Electron Gases Induced by Spontaneous and Piezoelectric Polarization in Undoped and Doped AlGaN/GaN Heterosturctures", *J. Appl. Phys.* Vol. 87, No. 1, (1 Jan. 2000): 334-344.

8. Anderson, D.G., "Iterative Procedure for Nonlinear Integral Equations", *ACM Journal* Vol. 12, No. 4 (n.d.): 547-560.

9. Andrews, J.M., and M.P. Lepselter, "Reverse Current-Voltage Characteristics of Metal-Silicide Schottky Diodes", *Solid-State Electronics* 13 (1970): 1011-1023.

10. Antogneti, P., and G. Massobrio. *Semiconductor Device Modeling With SPICE*. McGraw-Hill Book Company, 1987.

11. Apanovich Y. et. al, "Numerical Simulation of Submicrometer Devices, Including Coupled Non-local Transport and Non-isothermal Effects," *IEEE Trans. Electron Devices* Vol. 42, No. 5 (1995): 890-898.

12. Apanovich Y. et. al, "Stready-State and Transient Analysis of Submircon Devices Using Energy Balance and Simplified Hydrodyamic Models," *IEEE Trans. Comp. Aided Design of Integrated Circuits and Systems* Vol. 13, No. 6 (June 1994): 702-710.

13. Arkhipov, V.I., P. Heremans, E.V. Emelianova, G.J. Adriaenssens, H. Bassler, "Charge carrier mobility in doping disordered organic semiconductors", *Journal of Non-Crystalline Solids*, Vol. 338-340 (2004): 603-606.

14. Arkhipov, V.I., P. Heremans, E.V. Emelianova, G.J. Adriaenssens, H. Bassler, "Charge carrier mobility in doped semiconducting polymers", *Applied Phyics Letters*, Vol. 82, No. 19 (May 2003): 3245-3247.

15. Arora, N.D., J.R. Hauser, and D.J. Roulston, "Electron and Hole Mobilities in Silicon as a function of Concentration and Temperature", *IEEE Trans. Electron Devices* Vol. 29 (1982): 292-295.

16. Baccarani, G., M. Rudan, R. Guerrieri, and P. Ciampolini, "Physical Models for Numerical Device Simulation", *European School of Device Modeling*, University of Bologna, 1991.

17. Bank, R.E., and A.H. Sherman, "A Refinement Algorithm and Dynamic Data Structure for Finite Element Meshes", University of Texas at Austin Technical Report CS TR-159/CNA Tr-166, October 1980.

18. Bank, R., W.M. Coughran, W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith, "Transient Simulation of Silicon Devices and Circuits", *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 1992-2007.

19. Baraff G.A., "Distribution Functions and Ionisation Rates for Hot Electrons in Semiconductors", *Appl. Phys. Rev.* 128 (1962): 2507-2517.

20. Barnes, J.J., R.J. Lomax, and G.I. Haddad, "Finite-element Simulation of GaAs MESFET's with Lateral Doping Profiles and Sub-micron Gates", *IEEE Trans. Electron Devices* Vol. 23 (September 1976): 1042-1048.

21. Beattie, A.R., and A.M. White, "An Analytical Approximation with a Wide Range of Applicability for Electron Initiated Auger Transitions in Narrow-gap Semiconductors." *J. Appl. Phys.* Vol. 79, No. 12 (1996): 802-813.

22. Beinstman P. et al. "Comparision of Optical VCSEL Models on the Simulation at Oxide-Confined Devices." *IEEE J. of Quantum Electron* 37 (2001): 1618-1631.

23. Bernardini, F., and V. Fiorentini, "Spontaneous Polarization and Piezoelectric Constants of III-V Nitrides", *Phys. Rev.* B Vol. 56, No. 16, 15 (Oct. 1997): R10024-R10027.

24. Blom, P.W.M., M.J.M. de Jong, and S. Breedijk, "Temperature Dependent Electron-Hole Recombination in Polymer Light-Emitting Diodes", *Applied Physics Letters* Vol. 71, No. 7 (August 1997): 930-932.

25. Blotekjaer, K., *IEEE Trans. Electron Devices* Vol. 17 (1970): 38.

26. Bonani, F., and G. Ghione. *Noise in Semiconductor Devices - Modeling and Simulation*. Berlin Heidelberg: Springer- Verlag (2001): 24-26.

27. Born, M., and E. Wolf. *Principles of Optics*. 6th ed., Cambridge Press, 1980.

28. Boyd, R.W. *Nonlinear Optics*. New York: Academic Press, 1992.

29. Buturla, E.M., and P.E. Cotrell, "Simulation of Semiconductor Transport Using Coupled and Decoupled Solution Techniques", *Solid State Electronics* 23, No. 4 (1980): 331-334.

30. Canali, C., G.Magni, R. Minder and G. Ottaviani, "Electron and Hole drift velocity measurements in Silicon and their empirical relation to electric field and temperature", *IEEE Trans. Electron Devices ED-22* (1975): 1045-1047.

31. Carre, B.A., "The Determination of the Optimum Acceleration for Successive Over-Relaxation", *Computer Journal* Vol. 4, Issue 1 (1961): 73-79.

32. Caughey, D.M., and R.E. Thomas. "Carrier Mobilities in Silicon Empirically Related to Doping and Field." Proc. IEEE 55, (1967): 2192-2193.

33. Cassi C., and B. Ricco, "An Analytical Model of the Energy Distribution of Hot Electrons", *IEEE Trans. Electron Devices* Vol. 37 (1990): 1514-1521.

34. Chinn, S., P. Zory, and A. Reisinger, "A Model for GRIN-SCH-SQW Diode Lasers", *IEEE J. Quantum Electron.* Vol. 24, No. 11, November 1988.

35. Choo, S.C., "Theory of a Forward-Biased Diffused Junction P-L-N Rectifier -- Part 1: Exact Numerical Solutions", *IEEE Transactions on Electron Device Letters*, ED-19, 8 (1972): 954-966.

36. Chuang S.L., "Optical Gain of Strained Wurtzite GaN Quantum-Well Lasers", *IEEE J. Quantum Electron.* Vol. 32, No. 10 (Oct. 1996): 1791-1800.

37. Chuang, S.L., and C.S. Chang, "k*p Method for Straind Wurtzite Semiconductors", *Phys. Rev.* B Vol. 54, No. 4, 15 (July 1996): 2491-2504.

38. Chuang S.L., and C.S. Chang, "A Band-Structure Model of Strained Quantum-Well Wurtzite Semiconductors", *Semicond. Sci. Technol.*, No. 12 (Nov. 1996): 252-262.

39. Chynoweth A.G., "Ionisation Rates for Electrons and Holes in Silicon", *Phys. Rev.* 109 (1958): 1537-1540.

40. Concannon, A., F. Piccinini, A. Mathewson, and C. Lombardi. “The Numerical Simulation of Substrate and Gate Currents in MOS and EPROMs.” IEEE Proc. IEDM (1995): 289-292.

41. Crofton, J., and S. Sriram, "Reverse Leakage Current Calculations for SiC Schottky Contacts", *IEEE Trans. Electron Devices* Vol. 43, No. 12: 2305-2307.

42. Crowell, C.R., and S.M. Sze, “Current Transport in Metal-Semiconductor Barriers”, *Solid State Electronics* 9 (1966): 1035-1048.

43. Crowell, C.R., and S.M. Sze, “Temperature Dependence of Avalanche Multiplication in Semiconductors”, *Applied Physics Letters* 9 (1966): 242-244.

44. Darwish, M. et al, “An Improved Electron and Hole Mobility Model for General Purpose Device Simulation”, *IEEE Trans. Electron Devices* Vol. 44, No. 9 (Sept. 1997): 1529-1537.

45. De Mari, A., “An Accurate Numerical One-dimensional Solution of the P-N Junction Under Arbitrary Transient Conditions”, *Solid-State Electronics* 11 (1968): 1021-1053.

46. Dorkel, J., and Ph. Leturcq, “Carrier Mobilities in Silicon Semi-Empirically Related to Temperature Doping and Injection Level”, *Solid-State Electronics* 24 (1981): 821-825.

47. Dziewior J. and W. Schmid, “Auger Coefficient for Highly Doped and Highly Excited Silicon”, *Appl. Phys. Lett.* Vol. 31 (1977): 346-348.

48. Grant, W.N., “Electron and Hole Ionization Rates in Epitaxial Silicon at High Electric Fields”, *Solid-State Electronics* 16 (1973): 1189-1203.

49. J.L., Egley, and D., Chidambarrao, “Strain Effects on Device Characteristics: Implementation in Drift-Diffusion Simulators”, *Solid-State Electronics*, Vol. 36, No. 12, (1993): 1653-1664.

50. \Eisenstat, S.C., M.C. Gursky, M.H. Schultz, A.H. Sherman, Computer Science Department, Yale Univ. Tech. Rep. 112, 1977.

51. Engl, W.L., and H.K. Dirks, *Models of Physical Parameters in an Introduction to the Numerical Analysis of Semiconductor Devices and Integrated Circuits*, ed. J.J.H. Miller (Dublin: Boole Press, 1981).

52. Eriksson, J., N. Rorsman, and H. Zirath, "4H-Silicon Carbied Schottky Barrier Diodes for Microwave Applications", *IEEE Trans. on Microwave Theory and Techniques* Vol. 51, No. 3 (2003): 796-804.

53. Farahmand, M. et. al., "Monte Carlo Simulation of Electron Transport in the III-Nitride Wurtzite Phase Materials System: Binaries and Terniaries", *IEEE Trans. Electron Devices* Vol. 48, No. 3 (Mar. 2001): 535-542.

54. Feigna C., and Venturi F., “Simple and Efficient Modelling of EPROM Writing”, *IEEE Trans. Electron Devices* Vol. 38 (1991): 603-610.

55. Fossum, J.G. and D.S. Lee, “A Physical Model for the Dependence of Carrier Lifetime on Doping Density in Nondegenerate Silicon”, *Solid State Electronics* Vol. 25 (1982): 741-747.

56. Fujii, K., Y. Tanaka, K. Honda, H. Tsutsu, H. Koseki, and S. Hotta. “Process Techniques of 15 inch Full Color and High Resolution a-Si TFT LCD.” 5th Int. MicroProcess Conf., Kawasaki, Japan, 1992.

57. Gear, C.W. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.

58. Gill, W.D., "Drift Mobilities in Amorphous Charge-Transfer Complexes of Trinitrofluorenone and Poly-n-vinylcarbazole", *J. Appl. Phys.* Vol. 55, No. 12 (1972): 5033.

59. G.L., Bir, and G.E. Pikus, *Symmetry and Strain-Induced Effects in Semiconductors*, New York, Wiley, 1974.

60. Gossens, R.J. G., S. Beebe, Z. Yu and R.W. Dutton, “An Automatic Biasing Scheme for Tracing Arbitrarily Shaped I-V curves”, *IEEE Trans. Computer-Aided Design* Vol. 13, No. 3 (1994): 310-317.

61. Grove, A.S. *Physics and Technology of Semiconductor Devices*. Wiley, 1967.

62. Gundlach, K. H., *Solid-State Electronics* Vol. 9 (1966): 949.

63. Hack, B.M. and J.G. Shaw. "Numerical Simulations of Amorphous and Polycrystalline Silicon Thin-Film Transistors." Extended Abstracts 22nd International Conference on Solid-State Devices and Materials, Sendai, Japan (1990): 999-1002.

64. Hall, R.N., "Electron Hole Recombination in Germanium", *Phys. Rev.* 87 (1952): 387.

65. Hansch, W., Th. Vogelsang, R. Kirchner, and M. Orlowski, "Carrier Transport Near the Si/SiO2 Interface of a MOSFET", *Solid-State Elec.* Vol. 32, No. 10 (1989): 839-849.

66. Hansch, W. et al., "A New Self-Consistent Modeling Approach to Investigating MOSFET Degradation", *IEEE Trans. Electron Devices* Vol. 11 (1990): 362.

67. Harvey, J.E., "Fourier treatment of near-field scalar diffraction theory". *Am. J. Phys.* Vol. 47, No. 11 (1979): 974.

68. Heavens, O.S. *Optical Properties of Thin Solid Films*. Dover Publishing, 1991.

69. Hurkx, G.A.M, D.B.M. Klaassen, and M.P.G. Knuvers, "A New Recombination Model for Device Simulation Including Tunneling," *IEEE Trans. Electron Devices* Vol. 39 (Feb. 1992): 331-338.

70. Hurkx, G.A.M., D.B.M. Klaasen, M.P.G. Knuvers, and F.G. O'Hara, "A New Recombination Model Describing Heavy-Doping Effects and Low Temperature Behaviour", *IEDM Technical Digest* (1989): 307-310.

71. Hurkx, G.A.M., H.C. de Graaf, W.J. Klosterman et. al. "A Novel Compact Model Description of Reverse Biase Diode Characteristics including Tunneling." ESSDERC Proc. (1990): 49-52.

72. Horowitz, G., "Organic Field-Effect Transistors", *Advanced Materials* Vol. 10, No.5 (1998): 365-377.

73. Iannaccone, G., G. Curatola, and G. Fiori. "Effective Bohm Quantum Potential for device simulation based on drift-diffusion and energy transport." SISPAD 2004.

74. Ieong, M., Solomon, P., Laux, S., Wong, H., and Chidambarro, D., "Comparison of Raised and Schottky Source/Drain MOSFETs Using a Novel Tunneling Contact Model", *Proceedings of IEDM (*1998): 733-736.

75. Ishikawa, T. and J. Bowers, "Band Lineup and In-Plane Effective Mass of InGaAsP or InGaAlAs on InP Strained-Layer Quantum Well", *IEEE J. Quantum Electron.* Vol. 30, No. 2 (Feb. 1994): 562-570.

76. Iwai, H., M.R. Pinto, C.S. Rafferty, J.E. Oristian, and R.W. Dutton, "Velocity Saturation Effect on Short Channel MOS Transitor Capacitance", *IEEE Electron Device Letters* EDL-6 (March 1985): 120-122.

77. Jaeger, R.C and F. H. Gaensslen, "Simulation of Impurity Freezeout through Numerical Solution of Poisson's Equations and Application to MOS Device Behavior", *IEEE Trans. Electron Devices* Vol. 27 (May 1980): 914-920.

78. Joyce, W.B., and R.W. Dixon, "Analytic Approximation for the Fermi Energy of an ideal Fermi Gas", *Appl. Phys Lett.* 31 (1977): 354-356.

79. Kahen, K.B., "Two-Dimensional Simulation of Laser Diodes in Steady State", *IEEE J. Quantum Electron.* Vol. 24, No. 4, April 1988.

80. Katayama, K. and T. Toyabe, "A New Hot Carrier Simulation Method Based on Full 3D Hydrodynamic Equations", *IEDM Technical Digest* (1989): 135.

81. Keeney, S., F. Piccini, M. Morelli, A. Mathewson et. al., "Complete Transient Simulation of Flash EEPROM Devices", *IEDM Technical Digest* (1990): 201-204.

82. Kemp, A.M., M. Meunier, and C.G. Tannous, "Simulations of the Amorphous Silicon Static

Induction Transistor", *Solid-State Elect.* Vol. 32, No. 2 (1989): 149-157.

83. Kernighan, Brian, W. and Dennis M. Ritchie, "The C Programming Language", Prentice-Hall, 1978.

84. Kimura M. et. al., "Development of Poly-Si TFT Models for Device Simulation: In-Plane Trap Model and Thermionic Emission Model", (Proceedings of Asia Display/IDW '01, 2001), 423.

85. Klaassen, D.B.M., *Physical Modeling Bipolar Device Simulation", In: Simulation of Semiconductor Devices and Processes* Vol. 4, ed. W. Fichtner and D. Aemmer (Harting-Gorre, 1991), 23-43.

86. Klaassen, D.B.M., "A Unified Mobility Model for Device Simulation- I. Model Equations and Concentration Dependence", *Solid-State Elect.* Vol. 35, No. 7 (1992): 953-959.

87. Klaassen, D.B.M., "A Unified Mobility Model for Device Simulation - II. Temperature Dependence of Carrier Mobility and Lifetime", *Solid-State Elect.* Vol. 35, No. 7 (1992): 961-967.

88. Klaassen, D.B.M., J.W. Slotboom, and H.C. De Graaff, "Unified Apparent Bandgap Narrowing in n- and p- type Silicon", *Solid-State Elect.* Vol. 35, No. 2 (1992):125-129.

89. Klausmeier-Brown, M., M. Lundstrom, M. Melloch, "The Effects of Heavy Impurity Doping on AlGaAs/GaAs Bipolar Transistors", *IEEE Trans. Electron Devices* Vol. 36, No. 10 (1989): 2146-2155.

90. Kumagai, M., S.L. Chuang, and H. Ando, "Analytical Solutions of the Block-diagonalized Hamiltonian for Strained Wurtzite Semiconductors", *Phys. Rev.* B Vol. 57, No. 24, 15 (June 1998): 15303-15314.

91. Lackner, T., "Avalanche multiplication in semiconductors: A modification of Chynoweth's law", *Solid-State Electronics* Vol. 34 (1991): 33-42.

92. Lades M. and G. Wachutka. "Extended Anisotropic Mobility Model Applied to 4H/6H-SiC Devices." Proc. IEEE SISPAD (1997): 169-171.

93. Lambert, J. *Computational Methods in Ordinary Differential Equations*. Wiley, 1973.

94. Laux, S.E., "Techniques for Small-Signal Analysis of Semiconductor Devices", *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 2028-2037.

95. Law, M.E. et. al., Self-Consistent Model of Minority-Carrier Lifetime, Diffusion Length, and Mobility, *IEEE Electron Device Letters* Vol. 12, No. 8, 1991.

96. Li, Z., K. Dzurko, A. Delage, and S. McAlister, "A Self-Consistent Two-Dimensional Model of Quantum-Well Semiconductor Lasers: Optimization of a GRIN-SCH SQW Laser Structure", *IEEE J. Quantum Electron.* Vol. 28, No. 4 (April 1992): 792-802.

97. Lindefelt U., "Equations for Electrical and Electrothermal Simulation of Anisotropic Semiconductors", *J. Appl. Phys.* 76 (1994): 4164-4167.

98. Littlejohn, M.J., J.R. Hauser, and T.H. Glisson, "Velocity-field Characteristics of GaAs with $\Gamma_6^e - L_6^e - X_6^e$ Conduction Band Ordering", *J. Appl. Phys.* 48, No. 11 (November 1977): 4587-4590.

99. Lombardi et al, "A Physically Based Mobility Model for Numerical Simulation of Non-Planar Devices", *IEEE Trans. on CAD* (Nov. 1988): 1164.

100. Lui O.K.B., and P. Migliorato, "A New Generation-Recombination Model For Device Simulation Including The Poole-Frenkel Effect And Phonon-Assisted Tunneling", *Solid-State Electronics* Vol. 41, No. 4 (1997): 575-583.

101. Lundstrom M., and R. Shuelke, "Numerical Analysis of Heterostructure Semiconductor Devices", *IEEE Trans. Electron Devices* Vol. 30 (1983): 1151-1159.

102. Marques, M., Teles, L., Scolfaro, L., and Leite, J., "Lattice parameter and energy band gap of cubic $Al_xGa_yIn_{1-x-y}N$ quaternary alloys", *Appl. Phys. Lett.*, Vol. 83, No. 5, 4 (Aug. 2003): 890-892.

103. Mars, P. "Temperature Dependence of Avalanche Breakdown Voltage in p-n Junctions", *International Journal of Electronics* Vol. 32, No. 1 (1963): 23-27.

104. Masetti G., M.Severi, and S.Solmi, "Modeling of Carrier Mobility Against Carrier Concentration in Arsenic , Phosphorous and Boron doped Silicon", *IEEE Trans. Elec. Dev. ED-30*, (1983): 764-769.

105. Matsuzawa, K., Uchida, K., and Nishiyama, A., "A Unified Simulation of Schottky and Ohmic Contacts", *IEEE Trans. Electron Devices*, Vol. 47, No. 1 (Jan. 2000): 103-108.

106. Mayaram, K., "CODECS: A Mixed-Level Circuit and Device Simulator", Memo No. UCB/ERL M88/71, University of California, Berkeley, CA, November 1988.

107. Mayaram, K., and D.O. Pederson, "Coupling Algorithms for Mixed-Level Circuit and Device Simulation", *IEEE Transactions on Computer-Aided Design* Vol. 11, No. 8 (August 1992): 1003-1012.

108. Meinerzhagen, B., K. Bach, I. Bozk, and W.L. Eugl. "A New Highly Efficient Nonlinear Relaxation Scheme for Hydrodynamic MOS Simulations." Proc. NUPAD IV (1992): 91.

109. Meinerzhagen, B., and W.L. Engl, "The Influence of Thermal Equilibrium Approximation on the Accuracy of Classical Two-Dimensional Numerical Modeling of Silicon Submicrometer MOS Transistors", *IEEE Trans. Electron Devices* Vol. 35, No. 5 (1988): 689-697.

110. Miller et. al., "Device modeling of ferroelectric capacitors", *J. Appl. Phys.* 68, 12, 15 Dec. 1990.

111. Muth, J., et. al., "Absorption Coefficient, Energy Gap, Exciton Binding Energy, and Recombination Lifetime of GaN Obtained from Transmission Measurements", *Appl. Phys. Lett.* Vol. 71 (1997): 2572-2574.

112. Nakagawa, A., and H. Ohashi, "A Study on GTO Turn-off Failure Mechanism — A Time- and Temperature-Dependent 1-D Model Analysis", *IEEE Trans. Electron Devices* Vol. 31, No. 3 (1984): 273-279.

113. "Nomenclature of Inorganic Chemistry," *Journal of the American Chemical Society* No. 82 (1960): 5525.

114. Oguzman, I., et. al., "Theory of Hole Initiated Impact Ionization in Bulk Zincblende and Wurtzite GaN", *J. Appl. Phys.* V. 81, No. 12 (1997): 7827-7834.

115. Ohtoshi, T., K. Yamaguchi, C. Nagaoka, T. Uda, Y. Murayama, and N. Chionone, "A Two-Dimensional Device Simulator of Semiconductor Lasers", *Solid-State Electronics* Vol. 30, No.6 (1987): 627-638.

116. Okuto, Y., and R. Crowell, "Threshold energy effects on avalanche breakdown Voltage in semiconductor junctions", *Solid-State Electronics* Vol. 18 (1975): 161-168.

117. Palik, E.D., Ed. *Handbook of Optical Constants of Solids*. New York: Academic Press Inc., 1985.

118. Passler, R., *Phys.Stat.Solidi* (b) 216 (1999): 975.

119. Patil, M. B., "New Discretization Scheme for Two-Dimensional Semiconductor Device Simulation on Triangular Grid", *IEEE Trans. on CAD of Int. Cir. and Sys.* Vol. 17., No. 11 (Nov. 1998): 1160-1165.

120. Palankovski, V., Schultheis, R. and Selberherr, S., "Simulation of Power Heterojunction Bipolar Transistors on Gallium Arsenide", *IEEE Trans. on Elec. Dev.*, Vol. 48, No. 6, 6 (June 2001): 1264-1269.

121. Pinto M.R., Conor S. Rafferty, and Robert W. Dutton, "PISCES2 - Poisson and Continuity Equation Solver", Stanford Electronics Laboratory Technical Report, Stanford University, September 1984.

122. Piprek, J., *Semiconductor Optoelectronic Devices: Introduction to Physics and Simulation*. UCSB: Academic Press (2003): 22.

123. Piprek, J., Peng, T., Qui, G., and Olowolafe, J., "Energy Gap Bowing and Refractive Index Spectrum of AlInN and AlGaInN". 1997 IEEE International Symposium on Compound Semiconductors (September 8-11, 1997): 227-229.

124. Polsky, B., and J. Rimshans, "Half-Implicit Scheme for Numerical Simulation of Transient Processes in Semiconductor Devices", *Solid-State Electronics* Vol. 29 (1986): 321-328.

125. Price, C.H., "Two Dimensional Numerical Simulation of Semiconductor Devices", Ph.D. Dissertation, Stanford University, May 1982.

126. Price, P.J., and J. M. Radcliffe, "IBM Journal of Research and Development" 364, October 1959.

127. Rafferty, C.S., M.R. Pinto, and R.W. Dutton, "Iterative Methods in Semiconductor Device Simulation", *IEEE Trans. Electron Devices* Vol. 32 (October 1985): 2018-2027.

128. Rakic, A.D., and M.L. Majewski. *Cavity and Mirror Design for Vertical-Cavity Surface-Emitting Lasers", Vertical-Cavity Surface-Emitting Laser Devices*. Springer Series in Photonics, 2003

129. Rhoderick, E.H. and R.H. Williams. *Metal-Semiconductor Contacts,* 2nd ed. Oxford Science Publications, 1988.

130. Riccobene, C., G.Wachutka, J.Burgler, H.Baltes, "Operating principles of Dual Collector Magnetotransistors studied by Two-Dimensional Simulation", *IEEE Trans. Electron Devices*, Vol. 41, No. 7 (1994): 1136-1148.

131. Roblin, P., A. Samman, and S. Bibyk, "Simulation of Hot-Electron Trapping and Aging of n-MOSFET's", *IEEE Trans. Electron Devices* Vol. 35 (1988): 2229.

132. Rose, D.H., and R.E. Bank, "Global Approximate Newton Methods", *Numerische Mathematik* 37 (1981): 279-295.

133. Roulston, D.J., N.D. Arora, and S.G. Chamberlain, "Modeling and Measurement of Minority-Carrier Lifetime versus Doping in Diffused Layers of n ±p Silicon Diodes", *IEEE Trans. Electron Devices* Vol. 29 (Feb. 1982): 284-291.

134. Ruhstaller, B., S.A. Carter, S. Barth, H. Riel, W. Riess, J.C. Scott, "Transient and Steady-State Behavior of Space Charges in Multilayer Organic Light-Emitting Diodes", *J. Appl. Phys.* Vol. 89, No. 8 (2001): 4575-4586.

135. Sangiorgi, E., C.S. Rafferty, M.R. Pinto, and R.W. Dutton, "Non-planar Schottky Device Analysis and Applications", (Proc. International Conference on Simulation of Semiconductor Devices and Processes, Swansea, U.K.), July 1984.

136. Scharfetter, D.L., and H.K. Gummel, "Large-Signal Analysis of a Silicon Read Diode Oscillator", *IEEE Trans. Electron Devices* Vol. 16 (January 1969): 64-77.

137. Schenk, A., and G. Heiser, "Modeling and simulation of tunneling through ultra-thin gate dielectrics", *J.Appl.Phys.*Vol. 81 (1997): 7900-7908.

138. Schenk, A., "Rigorous Theory and Simplified Model of the Band-to-Band Tunneling in Silicon", *Solid State Electronics* Vol. 36 (1993): 19-34.

139. Schwarz, S.A., and S.E. Russe, "Semi-Empirical Equations for Electron Velocity in Silicon: Part II — MOS Inversion Layer", *IEEE Trans. Electron Devices* Vol. 30, No. 12 (1983): 1634-1639.

140. Scott, J.C., S. Karg, and S.A. Carter, "Bipolar Charge and Current Distributions in Organic Light-Emitting Diodes", *J. Appl. Phys.* Vol. 82, No. 3 (August 1997): 1454-1460.

141. Seki, S., T. Yamanaka, and K. Yokoyama, "Two-Dimensional Analysis of Current Blocking Mechanism in InP Buried Heterostructure Lasers", *J. Appl. Phys.* Vol. 71, No. 7 (April 1992): 3572-3578.

142. Selberherr, S. *Analysis and Simulation of Semiconductor Devices*. Wien, New York: Springer-Verlag, 1984.

143. Selberherr, S., "Process and Device Modeling for VLSI", *Microelectron. Reliab.* 24 No. 2 (1984):

225-257.

144. Shaw, John G., and Michael Hack, "An Analytical Model For Calculating Trapped Charge in Amorphous Silicon", *J. Appl. Phys.* Vol. 64, No. 9 (1988): 4562-4566.

145. Shen J., and J. Yang, "Physical Mechanisms in Double-Carrier Trap-Charge Limited Transport Processes in Organic Electroluminescent Devices: A Numerical Study", *J. Appl. Phys.* Vol. 83, No. 12 (June 1998): 7706-7714.

146. Shin, H., A.F. Tasch, C.M. Maziar, and S.K. Banerjee, "A New Approach to Verify and Derive a Transverse-field-dependent Mobility Model for Electrons in MOS Inversion Layers", *IEEE Trans. Electron Devices* Vol. 36, No. 6 (1989): 1117-1123.

147. Shin, H., G.M. Yeric, A.F. Tasch, and C.M. Maziar. *Physically-based Models for Effective Mobility and Local-field Mobility of Electrons in MOS Inversion Layers* (to be published).

148. Shirahata M., H. Kusano, N. Kotani, S. Kusanoki, and Y. Akasaka, "A Mobility Model Including the Screening Effect in MOS Inversion Layer", *IEEE Trans. Computer-Aided Design* Vol. 11, No. 9 (Sept. 1992): 1114-1119.

149. Shockley W., and W.T. Read, "Statistics of the Recombination of Holes and Electrons", *Phys. Rev.* 87 (1952): 835-842.

150. Simmons, J.G., and G.W. Taylor , *Phys. Rev.* B, 4 (1971): 502.

151. Slotboom, J.W., "The PN Product in Silicon", *Solid State Electronics* 20 (1977): 279-283.

152. Slotboom, J.W. and H.C. De Graaf, "Measurements of Bandgap Narrowing in Silicon Bipolar Transistors", *Solid State Electronics* Vol. 19 (1976): 857-862.

153. Stratton, R., *Phys. Rev.*, 126, 6 (1962): 2002.

154. Stratton, R., "Semiconductor Current-Flow Equations (Diffusion and Degeneracy)", *IEEE Trans. Electron Devices* Vol. 19, No. 12 (1972): 1288-1292.

155. Sutherland, J., and F. Hauser, "A Computer Analysis of Heterojunction and Graded Composition Solar Cells", *IEEE Trans. Electron Devices* Vol. 24 (1977): 363-373.

156. Sze, S.M. "Physics of Semiconductor Devices". Wiley, 1981.

157. Takahashi Y., K. Kunihiro, and Y. Ohno, "Two-Dimensional Cyclic Bias Device Simulator and Its Applications to GaAs HJFET Pulse Pattern Effect Analysis," *IEICE Journal C*, June 1999.

158. Takayama, T., "Low-Noise and High-Power GaAlAs Laser Diode with a New Real Refractive Index Guided Structure," *Japan J. Appl. Phys.* Vol. 34, No. 7A(1999): 3533-3542.

159. Tam, S., P-K Ko, and C. Hu, "Lucky-electron Model of Channel Hot-electron Injection in MOSFET's", *IEEE Trans. Electron Devices* Vol. 31, No. 9, September 1984.

160. Tan, G., N. Bewtra, K. Lee, Xu, and J.M., "A Two-Dimensional Nonisothermal Finite Element Simulation of Laser Diodes", *IEEE J. of Q.E.* Vol. 29, No.3 (March 1993): 822-835.

161. Uchida, K., and Takagi, S., "Carrier scattering induced by thickness fluctuation of silicon-on-insulator film in ultrathin-body metal-oxide-semiconductor field-effect transistors", *Applied Phys. Lett.*, Vol. 82, No. 17, 22 (April 2003): 2916-2918.

162. *UTMOST III Modeling Manual*. SILVACO International, 1993.

163. Valdinoci, M., D. Ventura, M.C. Vecchi, M. Rudan, G. Baccarani, F. Illien, A. Stricker, and L. Zullino, "The Impact-ionization in Silicon at Large Operating Temperature". SISPAD '99, Kyoto, Japan, Sept. 6-8, 1999.

164. Van Dort, M.J., P.H. Woerlee, and A.J. Walker, "A Simple Model for Quantisation Effects in Heavily-Doped Silicon MOSFETs at Inversion Conditions", *Solid-State Elec.* Vol. 37, No. 3 (1994): 411-414.

165. Van Overstraeten, R., and H. Deman, "Measurement of the Ionization Rates in Diffused Silicon

p-n Junctions", *Solid-State Electronics* 13 (1970): 583-608.

166. Van Roey, J., J. Van Der Donk, P.E. Lagasse., "Beam-propagation method: analysis and assessment", *J. Opt. Soc. Am.* Vol. 71, No 7 (1981): 803.

167. Varga, R.S. *Matrix Iterative Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1962.

168. Verlaak, S., Cheyns D., Debucquoy M., Arkhipov, V. and Heremans P., "Numerical Simulation of Tetracene Light-Emitting Transistors: A Detailed Balance of Exciton Processes", *Applied Physics Letters* Vol. 85, No. 12 (Sep 2004): 2405-2407.

169. Vurgaftman, I. and J.R. Meyer, "Band Parameters for III-V Compound Semiconductors and their Alloys", *Applied Physics Review* Vol. 89, No. 11 (June 2001): 5815-5875.

170. Wachutka, G.K., "Rigorous Thermodynamic Treatment of Heat Generation in Semiconductor Device Modeling", *IEEE Trans., Computer-Aided Design* Vol. 9, No. 11 (1990): 1141-1149.

171. Wada, M. et al. "A Two-Dimensional Computer Simulation of Hot Carrier Effects in MOSFETs." Proc. IEDM Tech. Dig. (1981): 223-225.

172. Walker, D., Zhang, X., Saxler, Z., Kung, P., Xj, J., Razeghi, M., "AlxGa(1-x)N(0<=x,=1)Ultraviolet Photodetectors Grown on Sapphire by Metal-Organic Chemical-Vapor Deposition", *Appl. Phys. Lett.* Vol. 70, No. 8 (1997): 949-951.

173. Watt, J.T., Ph.D., Thesis, Stanford University, 1989.

174. Watt, J.T., and J.D. Plummer. "Universal Mobility-Field Curves for Electrons and Holes in MOS Inversion Layers." 1987 Symposium on VLSI Technology, Karuizawa, Japan.

175. Weast, R. Ed. *CRC Handbook of Chemistry and Physics*. CRC Press, 1976.

176. Wenus, J., J. Rutkowski, and A. Rogalski, "Two-Dimensional Analysis of Double-Layer Heterojunction HgCdTe Photodiodes", *IEEE Trans. Electron Devices* Vol. 48, No.7 (July 2001): 1326-1332.

177. Wenzel, H., and H.J. Wünsche, "The effective frequency method in the analysis of vertical-cavity surface-emitting lasers," *IEEE J. Quantum Electron* 33 (1997): 1156-1162.

178. Wettstein, A., A. Schenk, and W. Fichtner, "Quantum Device-Simulation with the Density Gradient Model on Unstructured Grids", *IEEE Trans. Electron Devices* Vol. 48, No. 2 (Feb. 2001): 279-283.

179. White, W.T. et.al., *IEEE Trans. Electron Devices*. Vol. 37 (1990): 2532.

180. Wilt, D.P. and A. Yariv, "Self-Consistent Static Model of the Double-Heterostructure Laser," *IEEE J. Quantum Electron.* Vol. QE-17, No. 9 (1981): 1941-1949.

181. Wimp, T. *Sequence Transformation and their Application*. Academic Press, 1981.

182. Wünsche, H.J., H. Wenzel, U. Bandelow, J. Piprek, H. Gajewski, and J. Rehberg, "2D Modelling of Distributed Feedback Semiconductor Lasers," *Simulation of Semiconductor Devices and Processes* Vol. 4 (Sept. 1991): 65-70.

183. Wu, C.M., and E.S. Yang, "Carrier Transport Across Heterojunction Interfaces", *Solid-State Electronics* 22 (1979): 241-248.

184. Yamada, M., S. Ogita, M. Yamagishi, and K. Tabata, "Anisotropy and broadening of optical gain in a GaAs/AlGaAs multi-quantum-well laser," *IEEE J. Quantum Electron.* QE-21 (1985): 640-645.

185. Yamada, T., J.R. Zhou, H. Miyata, and D.K. Ferry, "In-Plane Transport Properties of Si/Si(1-x)Ge(x) Structure and its FET Performance by Computer Simulation", *IEEE Trans. Electron Devices* Vol. 41 (Sept. 1994): 1513-1522.

186. Yamaguchi, K. "A Mobility Model for Carriers in the MOS Inversion Layer", *IEEE Trans. Electron Devices* Vol. 30 (1983): 658-663.

187. Yan, R., S. Corzine, L. Coldren, and I. Suemune, "Corrections to the Expression for Gain in GaAs", *IEEE J. Quantum Electron.* Vol. 26, No. 2 (Feb. 1990): 213-216.

188. Yang K., J. East, and G. Haddad, "Numerical Modeling of Abrupt Heterojunctions Using a Thermionic-Field Emission Boundary Condition", *Solid-State Electronics* Vol. 36, No. 3 (1993): 321-330.

189. Yariv, A. *Optical Electronics*. CBS College Publishing, 1985.

190. Yu, Z., D. Chen, L. So, and R.W. Dutton, "Pisces-2ET, Two Dimensional Device Simulation for Silicon and Heterostructures", Integrated Circuits Laboratory, Stanford University (1994): 27.

191. Yu, Z., and R.W. Dutton, "SEDAN III-A Generalized Electronic Material Device Analysis Program", Stanford Electronics Laboratory Technical Report, Stanford University, July 1985.

192. Zappa, F., Lovati, P., and Lacaita, A., "Temperature Dependence of Electron and Hole Ionization Coefficients in InP." (Eighth International Conference Indium Phosphide and Related Materials, IPRM '96, Schwabisch Gmund, Germany April 21-25, 1996), 628-631.

193. Zhou, B., Butcher, K., Li, X., Tansley, T., "Abstracts of Topical Workshop on III-V Nitrides". TWN '95, Nagoya, Japan, 1995.

194. Zhou, J.R. and D.K. Ferry, "Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations," *IEEE Trans. Electron Devices* Vol. 39 (Mar. 1992): 473-478.

195. Zhou, J.R. and D.K. Ferry, "Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations - II:Volocity Overshoot", *IEEE Trans. Electron Devices* Vol. 39 (Aug. 1992): 1793-1796.

This page is intentionally left blank.

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y