

Notes on Silvaco TCAD Simulations

January 2, 2024

Contents

1	About	2
2	Atlas	2
2.1	Meshing!	3
2.1.1	Circular mesh	3
2.1.2	Regrid/remesh	3
2.2	Material properties	4
2.2.1	Interface properties	4
2.3	Models	4
2.4	Numerical Methods	5
2.5	Examples	5
2.5.1	Optoelectronics	5

Colour legend

This document will follow a highlighting scheme where text highlighted in different colors mean the following:

Yellow coloured text - Doubts, or sentences that are to be clarified/understood later.

Green coloured text - Some takeaways.

Purple coloured text - Interesting and unique points, that often are not mentioned explicitly in textbooks/literature.

1 About

Silvaco TCAD is one of the major commercial softwares in the current market that offer tools for the simulation of silicon devices. This note in particular will focus on the tools that facilitate the simulation of Silicon sensors utilized in high energy physics experiments. Particular focus will be towards Athena and Atlas. Athena is a "device-technology" simulation tool, and is widely used for simulating the electrical behavior of semiconductor devices. Athena on the other hand, is a "process-technology" simulation tool. It focuses on the simulation of the manufacturing processes involved in creating semiconductor devices, including deposition, etching, diffusion, etc. Both tools are often used hand-in-hand to offer a comprehensive study of the device to be manufactured.

2 Atlas

The primary reference used is Silvaco's Atlas User's Manual [1]. Infact, this section will be a condensed version of the User's Manual, the only differences being the highlighting of parts relevant+useful for Silicon detector simulations, and additional examples provided by the authors. The workflow as given in [1] is shown below in Figure 1. In a very primitive sense: we provide inputs to Atlas through Athena output files, and code in DeckBuild. Atlas then provides outputs that can be visualized in TonyPlot. Most of the following subsections will involve working in the DeckBuild level.

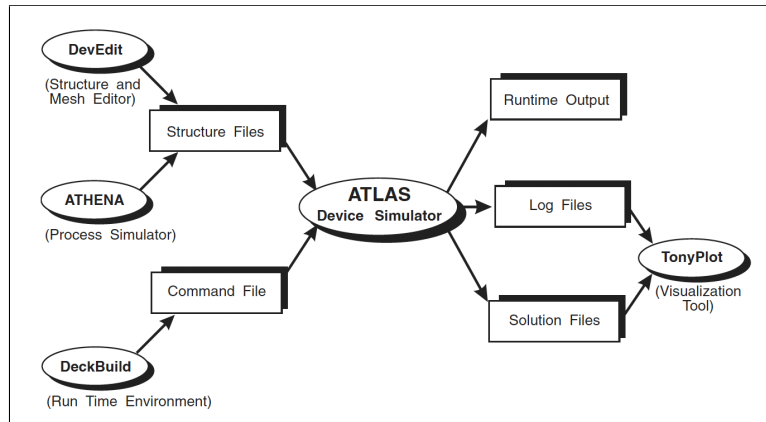


Figure 1: Source: [1]

Every code line for Atlas will be of the following form:

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

An example of a statement line is:

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

The statement is DOPING, and 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 parameter with floating point input. REGION is a parameter taking integers as input.

A crucial thing to note in Atlas code is the order of which statements need to be called in. Typically the

<i>Group</i>	<i>Statements</i>
1. Structure Specification	MESH REGION ELECTRODE DOPING
2. Material Models Specification	MATERIAL MODELS CONTACT INTERFACE
3. Numerical Method Selection	METHOD
4. Solution Specification	LOG SOLVE LOAD SAVE
5. Results Analysis	EXTRACT TONYPLOT

Figure 2: Ordering of statements in Atlas. Source: [1]

spatial units are in microns.

Chapter 21 (Statements) of [1] contains the syntax and parameter information of most, if not all statements used in Silvaco TCAD simulations.

2.1 Meshing!

2.1.1 Circular mesh

Some structures can benefit more with a circular mesh than a rectangular mesh, we have been working with, until now. An example of how circular meshes can be defined in Atlas is shown below:

```
MESH CIRCULAR
R.MESH LOC=0.0 SPAC=0.05
R.MESH LOC=0.2 SPAC=0.05
R.MESH LOC=0.35 SPAC=0.025
R.MESH LOC=0.4 SPAC=0.01
A.MESH LOC=0 SPAC=30
A.MESH LOC=360 SPAC=30
```

The compiler is informed that the mesh will be a circular in the first mesh definition command. The commands that follow are locations and spacings of the mesh the user wants, which is similar to the rectangular meshes (X, Y), except, this time, we are working in circular coordinates (radius \equiv R, theta \equiv A).

2.1.2 Regrid/remesh

After generating a coarse mesh along the structure the user has defined, Atlas provides the user with the capability of generating a finer mesh in a localized region. A typical usage of this feature is in generating a finer mesh near a doping layer where the doping concentration is varying rapidly along space.

```
REGRID LOGARITHM DOPING RATIO=2 SMOOTH.KEY=4 DOPFILE=<file1> OUTFILE=<file2>
```

The example statement given above involves a regridding that will resolve doping profiles to two orders of magnitude in change. The doping file, file1, must be specified in the first DOPING statement. The results of the regrid are saved in the file, file2. 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.

Please note that this statement must be used **after** the MESH, REGION, MATERIAL, ELECTRODE, and DOPING statements.

2.2 Material properties

A region can be associated with a material and properties can be defined for that material. The MATERIAL statement can also set properties for a material of a certain type present throughout the device. The following lines are how you perform the previously stated functions in this paragraph:

```
MATERIAL MATERIAL=Silicon EG300=1.12 MUN=1100  
MATERIAL REGION=2 TAUN0=2e-7 TAUP0=1e-5
```

2.2.1 Interface properties

The INTERFACE statement is used to define the interface charge density and surface recombination velocity at interfaces between semiconductors and insulators. The interface of interest can also be restricted to a specific region by specifying the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. These parameters define a rectangle inside which the interface properties apply. For example:

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

specifies that all interfaces between semiconductors and insulators within the rectangular region specified, have a fixed charge of 3.1010cm^{-2} .

2.3 Models

The MODELS statement defines the physics models to be used by Atlas to perform the corresponding simulation. As one would guess, this forms a crucial step in obtaining realistic simulations. To help with this, Atlas provides a method to select the right models for the technology the user wants to simulate: MOS (models for MOSFET) and BIP (bipolar devices), PROGRAM (programming), and ERASE (erasing programmable devices). The reader is expected to keep note of the compatibility tables (for mobility models) described in the User's manual: models that are/are-not compatible with other models.

The user can also select+run material-specific models by specifying the material in the MODELS runline. For example:

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

It is important to note that the impact ionization models however will have to be called for in a separate runline using the IMPACT statement.

As a check, one can use the PRINT parameter which prints out (in the runtime output) the models and parameters that were used in the simulation.

2.4 Numerical Methods

A system of unknowns can be solved by three techniques in Atlas, depending upon the coupling and the order of convergence (See wikipedia article for more info), namely: GUMMEL, NEWTON, and BLOCK. A combination of these methods can also be used, however the statement calls in such scenarios will depend on the model it is being applied to. A detailed description on guidelines to these methods are available in Chapter 20 of [1]. For basic drift-diffusion calculations, using

```
METHOD GUMMEL NEWTON
```

will cause the solver to start with GUMMEL iterations and then switch to NEWTON if convergence is not achieved.

A relevant statement to silicon detector simulations is controlling for how many types of carriers (between electrons and/or holes), the continuity equations need to be solved. The examples statements involved are:

```
METHOD CARRIERS=2 (default)
METHOD CARRIERS=1 HOLE
METHOD CARRIERS=0 (solve potential)
```

2.5 Examples

This subsection aims to serve as a better look-up section for the example library of Silvaco TCAD to help save the reader time from looking through the example directories, one folder at a time. This section will also look to make note of the nuances that were not explained in the code's comments.

2.5.1 Optoelectronics

opteoex01 is an example simulation for a PIN photodiode. The mesh, region, and electrode definitions are straightforward and self-explanatory. The main takeaway is from the doping and the models used for these doping levels in optoelectronics' use-case.

The bulk detector volume is n-type and there are two other doping regions (gaussian) towards the top and bottom.

The example uses the models:

```
srh auger conmob fldmob
```

The SRH (Shockley-Read-Hall, for minority carrier lifetime and not conc dependent) and Auger (Auger recombination) are non-radiative (not involving a photon) recombination models which are needed in photodiode applications. A simple and introductory reference for these two models can be found in this link. Wikipedia also has a good article towards the same. conmob and fldmob are mobility models which define mobility as a function of concentration and electric field respectively. According to [1], the later is required to model any type of velocity saturation effect.

References

- [1] Atlas User's Manual. Technical report, Silvaco, Inc., Santa Clara, CA, October 2013.