# MSL Tutorial

Course: EE735

25/09/2019

# Group Ids:

Username : EE735_group_no ;  Password is same as username

Ex: for group 1 , username is : EE735_1

hostID : 10.107.106.x  (x=21,22,13,16,17)

Ex: For terminal 21, hostID is 10.107.106.21

Connection to MCL servers:

**Linux system :**

ssh -X  username@hostID

**Windows:**

Mobaxterm

# Overview

- Structure specification
- Numerical Solution
- Material parameter Specifications
- Result analysis

# Simulation Flow

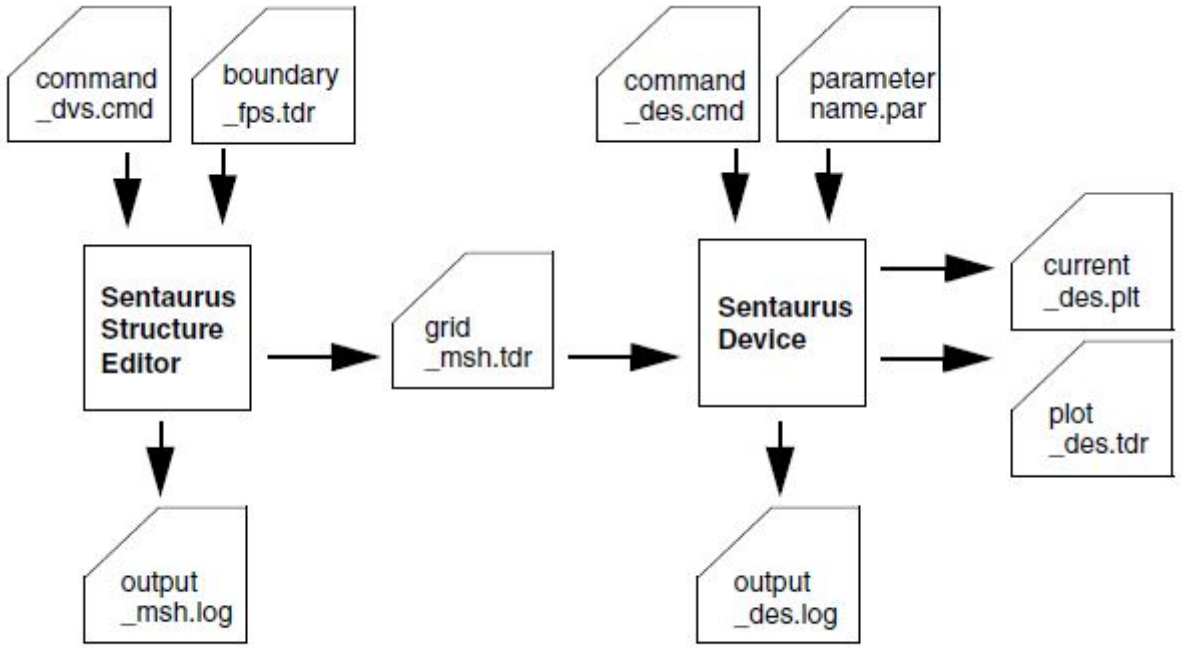



Figure 3 Typical tool flow with device simulation using Sentaurus Device

# Structure specification
# SDE or Sprocess

- SDE: Sentaurus Structure Editor can be used as a two-dimensional (2D) or three-dimensional (3D) structure editor, and a 3D process emulator to create TCAD devices

- SProcess: Sentaurus Process is an advanced 1D, 2D, and 3D process simulator suitable for silicon and non-silicon semiconductor devices. It features modern software architecture and state-of-the-art models to address current and future process technologies

# Structure Device Editor

- Sentaurus Structure Editor can be used interactively by either using the GUI menu bar and toolbars, or entering the Scheme commands in the command-line window

- To run Sentaurus Structure Editor in interactive mode, type in a command prompt: ***sde***

- To run a Scheme script file, for example MyScript.cmd, type in command prompt:

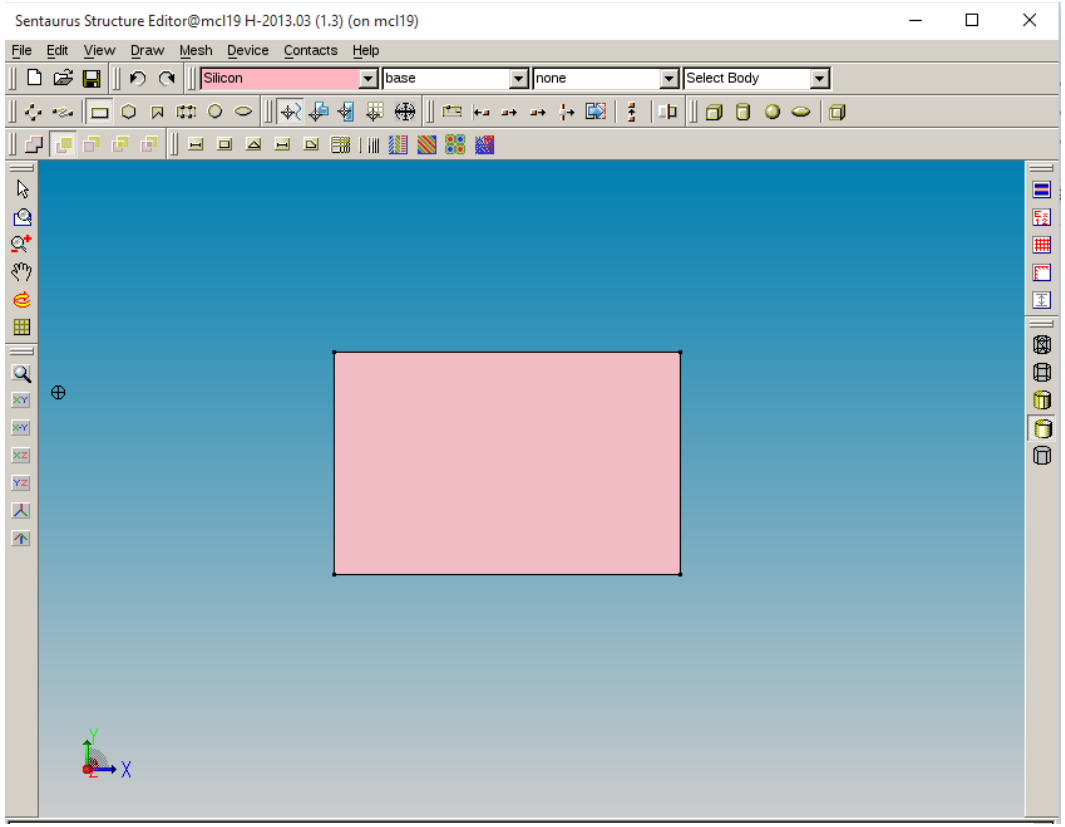
: ***sde -e -l MyScript.cmd***

# SDE

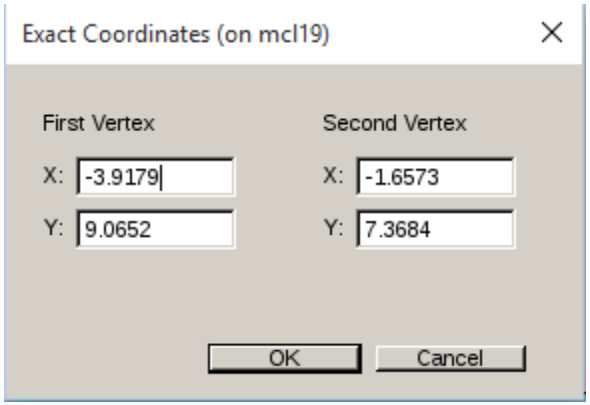
- Drawing a region
- Doping
- Contact placement
- Meshing

http://www.sentaurus.dsod.pl/sse/sse_a.html

# Drawing a region




You can set exact co-ordinates using Draw -> Exact Co-ordinates.

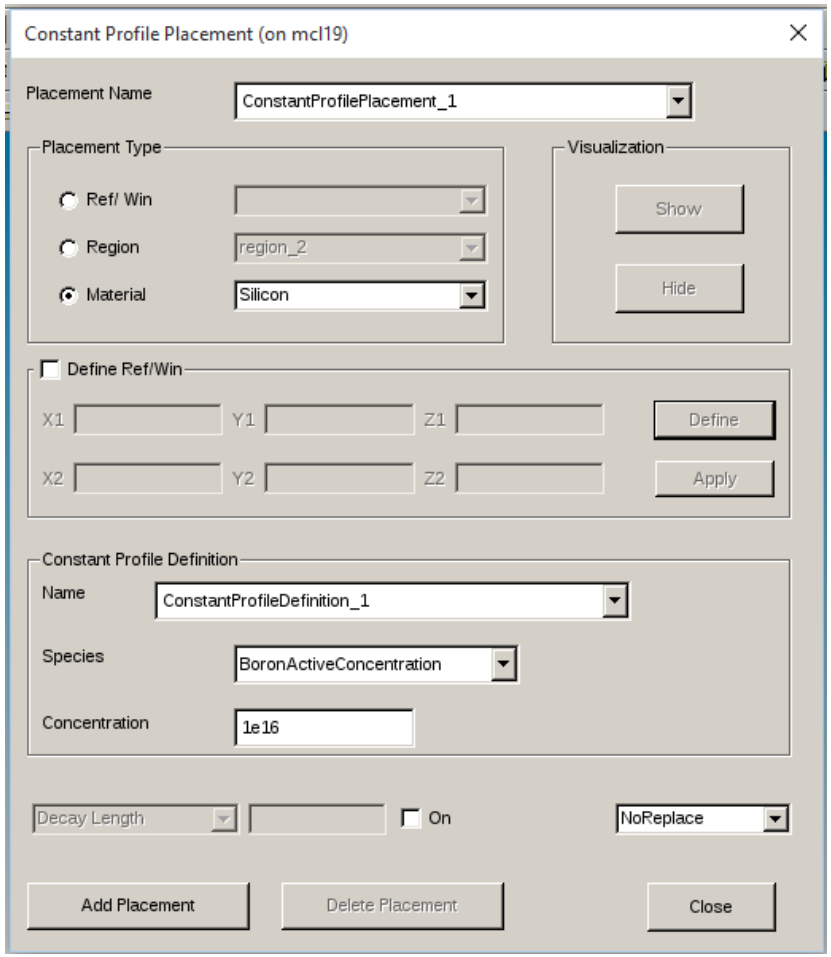



Code:

**(sdegeo:create-rectangle (position x1 y1 0) (position x2 y2 0) "Silicon" "region_1" )**

# Doping

Doping can be done by clicking Device -> Constant Profile Placement



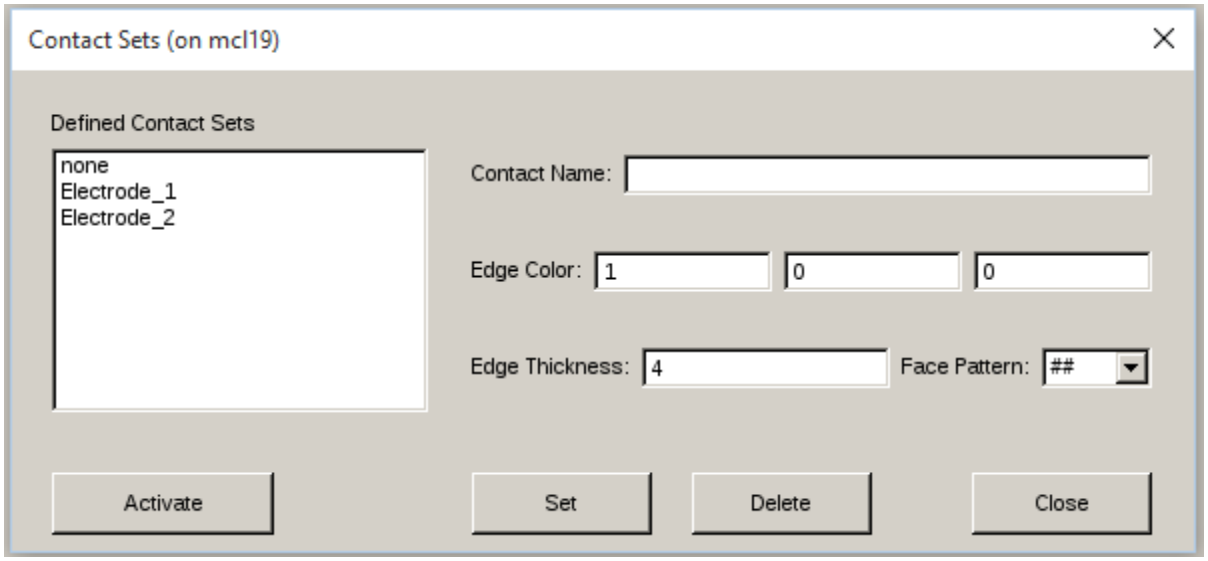

Code:

**(sdedr:define-constant-profile "ConstantProfileDefinition_1" "BoronActiveConcentration" 1e16)**

**(sdedr:define-constant-profile-material "ConstantProfilePlacement_1" "ConstantProfileDefinition_1" "Silicon")**

# Contact placement

- Go to Contacts-> Contact Sets
- Define the contacts.




Code:

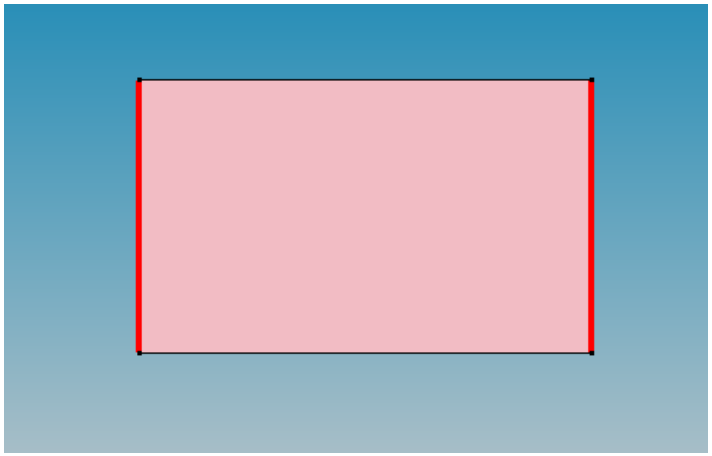
```
(sdegeo:define-contact-set "Electrode_1" 4  (color:rgb 1 0 0 ) "##" )
(sdegeo:define-contact-set "Electrode_2" 4  (color:rgb 1 0 0 ) "##" )
```

# Contact Definition

To define a contact,

- We have to activate the contact set we defined in the previous slide.
- Then we have to select “Select Edge” in the drop-down list


We now have to select which edge we want to define the contact at and then go to Contacts-> Set Edges



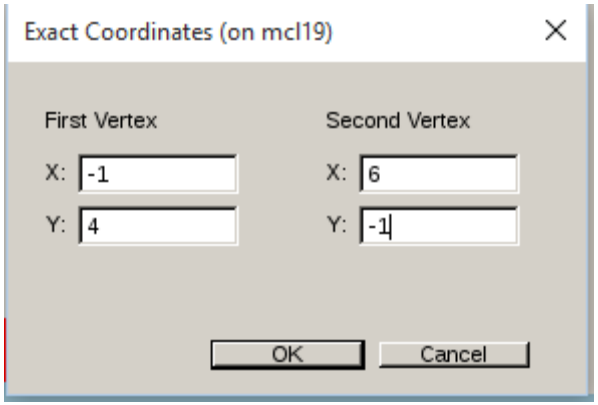
Code:

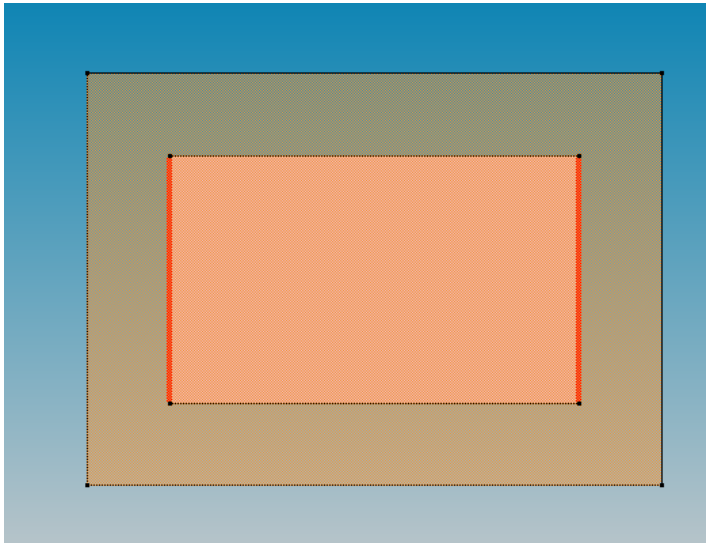
**(sdegeo:define-2d-contact (list (car (find-edge-id (position x1 y 0)))) "Electrode_1")**

**(sdegeo:define-2d-contact (list (car (find-edge-id (position x2 y 0)))) "Electrode_2")**

# Meshing

- Meshing is extremely important for solving equations inside the device.
- To define meshing, we need to define a refinement window to define a global mesh.
- Usually define a larger area than your device



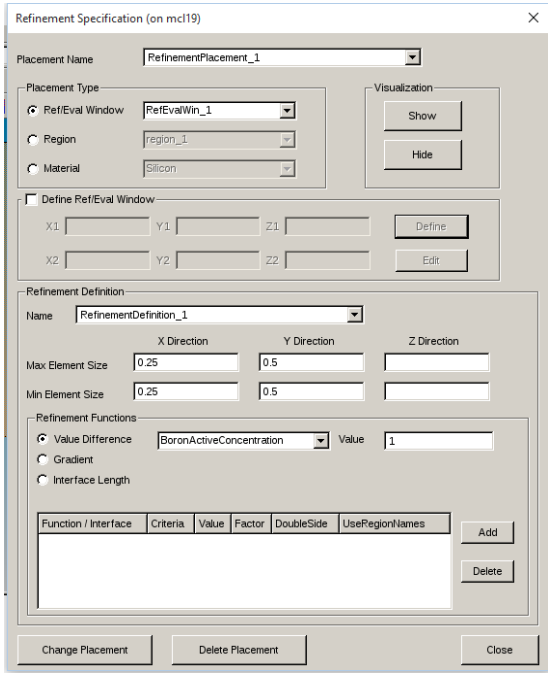



Code:

**(sdedr:define-refeval-window "RefEvalWin_1" "Rectangle" (position -1 4 0) (position 6 -1 0))**

# Meshing(Contd..)

- We now need to associate the global refinement window with the refinement box
- Select **Mesh** > **Refinement Placement**



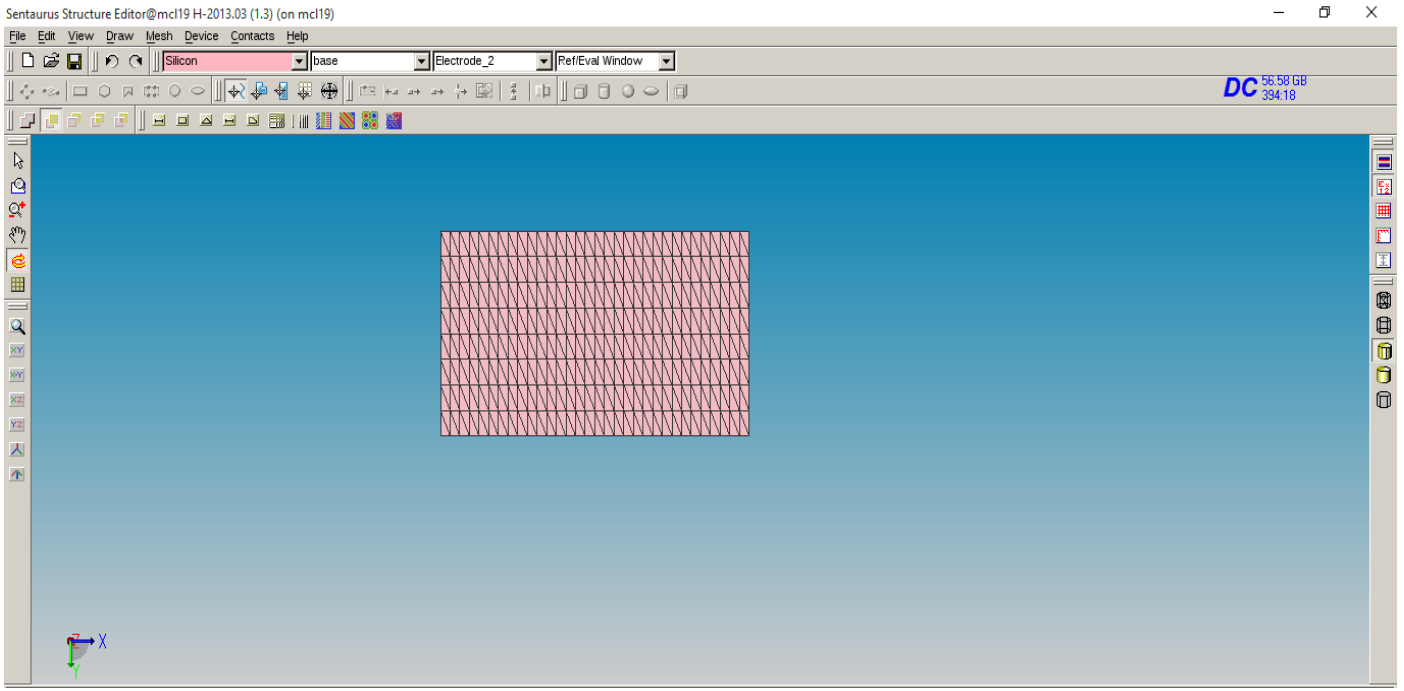

Code:

**(sdedr:define-refinement-size "RefinementDefinition_1" 0.25 0.5 0.25 0.5 )**

**(sdedr:define-refinement-placement "RefinementPlacement_1" "RefinementDefinition_1" "RefEvalWin_1" )**

# Meshing(Contd..)

- Then go to Mesh-> Build Mesh



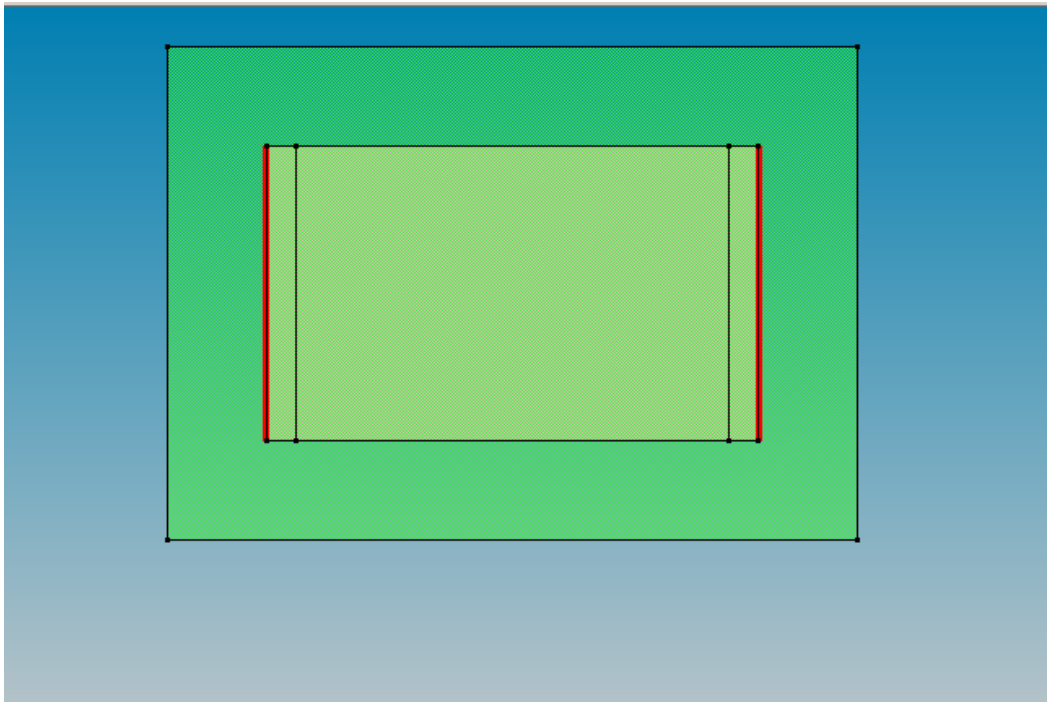

Code:

```
(sde:build-mesh "snmesh" "-a -c boxmethod" "sdemodel")
```

Your .tdr file is saved as sdemodel.tdr

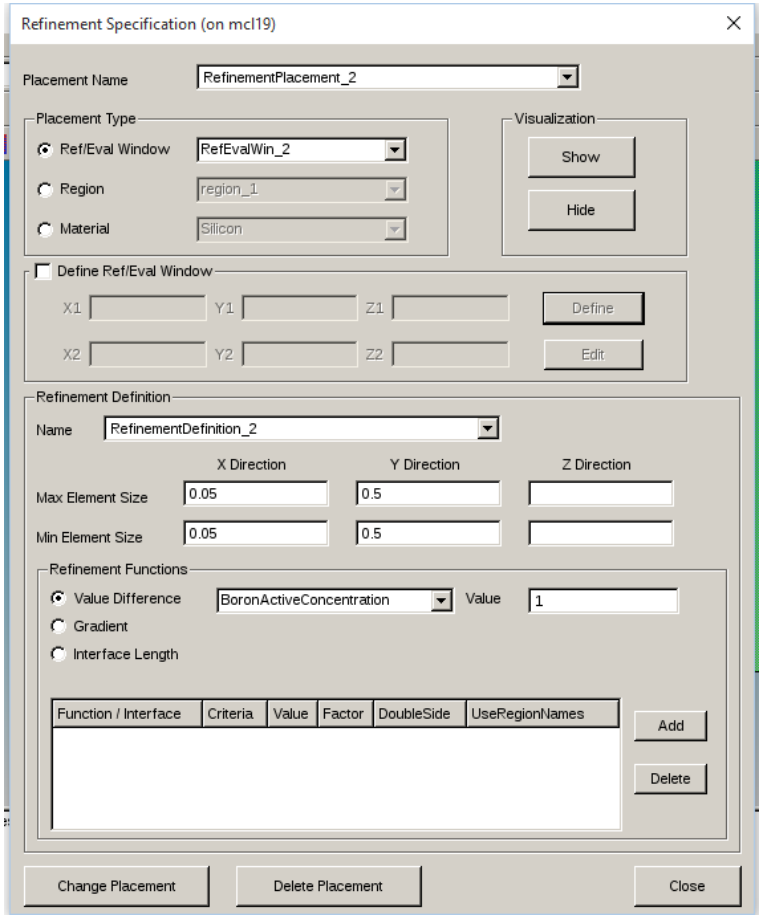
# Meshing(Contd..)

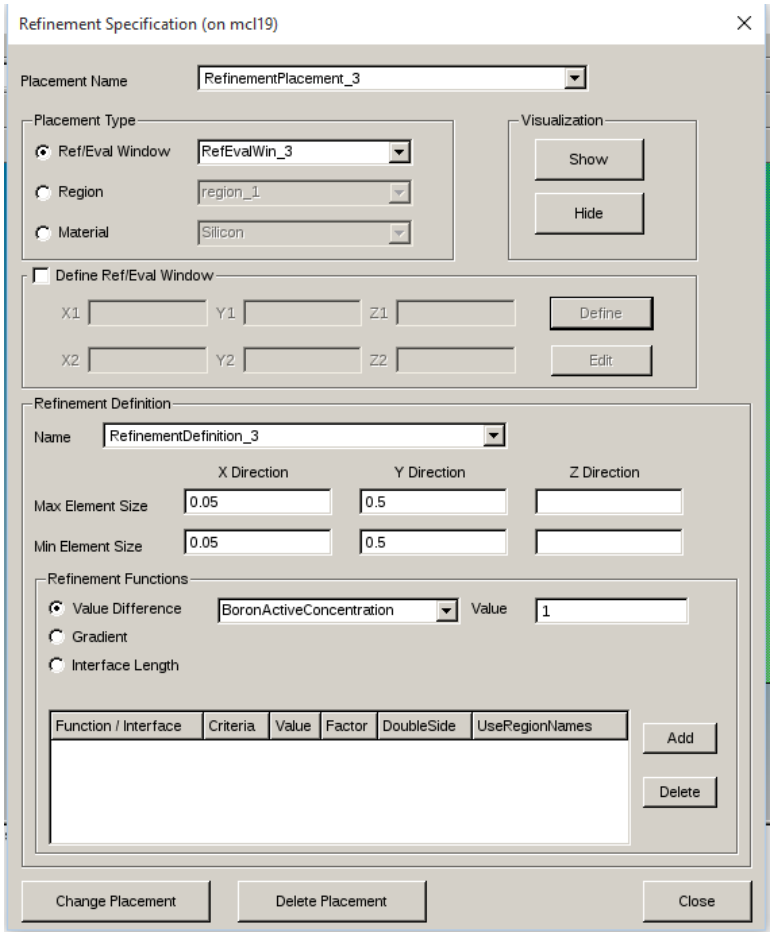
- You can change meshing by going to Mesh-> Refinement Placement again.
- Suppose we need different meshing at some specific regions which we consider the most ‘sensitive’ regions
- Lets take this current example and suppose we need finer meshing at the edges.
- We define similar refinement windows at the edges



# Meshing(Contd..)

- Similar to previous refinement windows, we have to define refinement placements for these regions.



# Meshing(Contd..)

- Code:

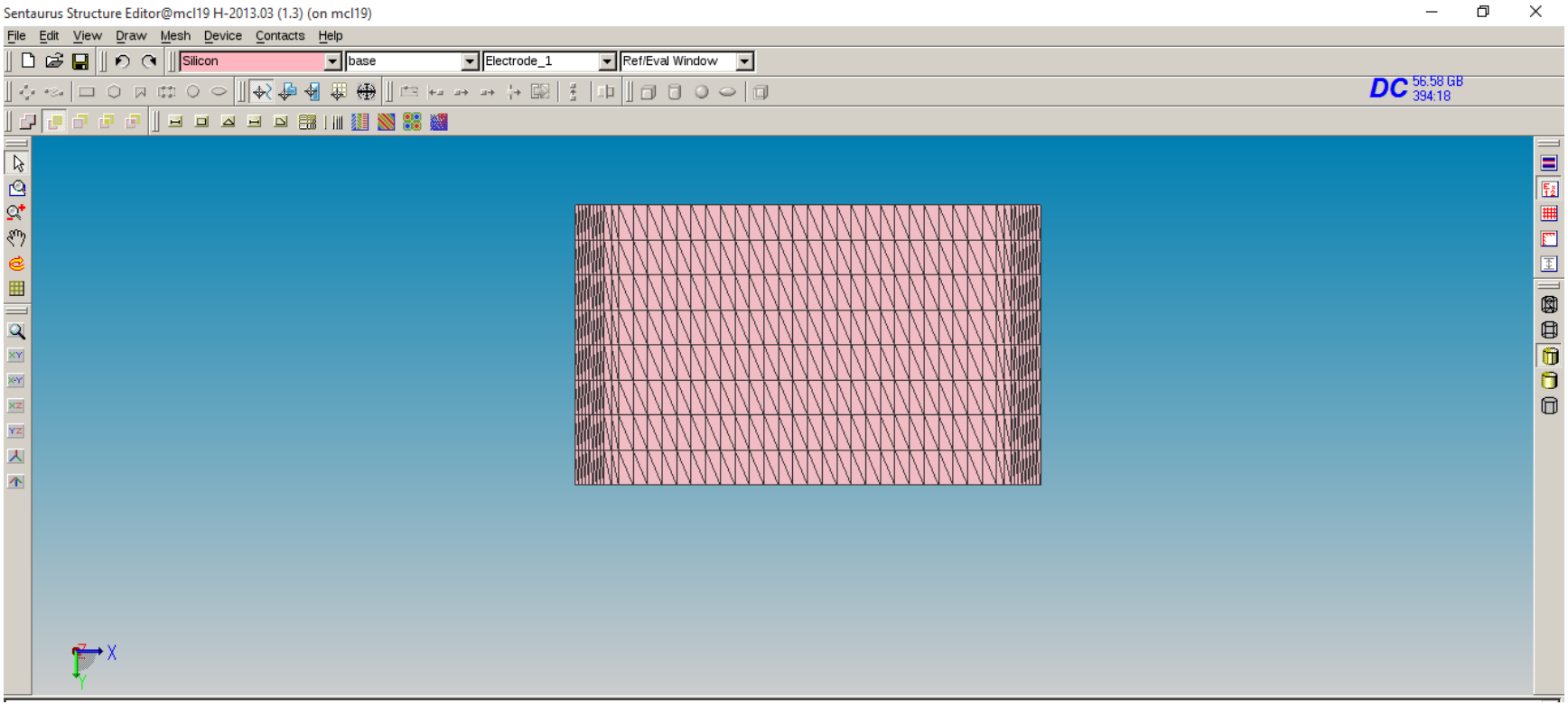
```
(sdedr:define-refinement-placement "RefinementPlacement_2"
"RefinementDefinition_2" "RefEvalWin_2" )
(sdedr:define-refinement-size "RefinementDefinition_2"
0.05 0.5 0.05 0.5 )

(sdedr:define-refinement-placement "RefinementPlacement_3"
"RefinementDefinition_3" "RefEvalWin_3" )
(sdedr:define-refinement-size "RefinementDefinition_3"
0.05 0.5 0.05 0.5 )
```

Here, we have used finer meshing in the x-direction near the edges.

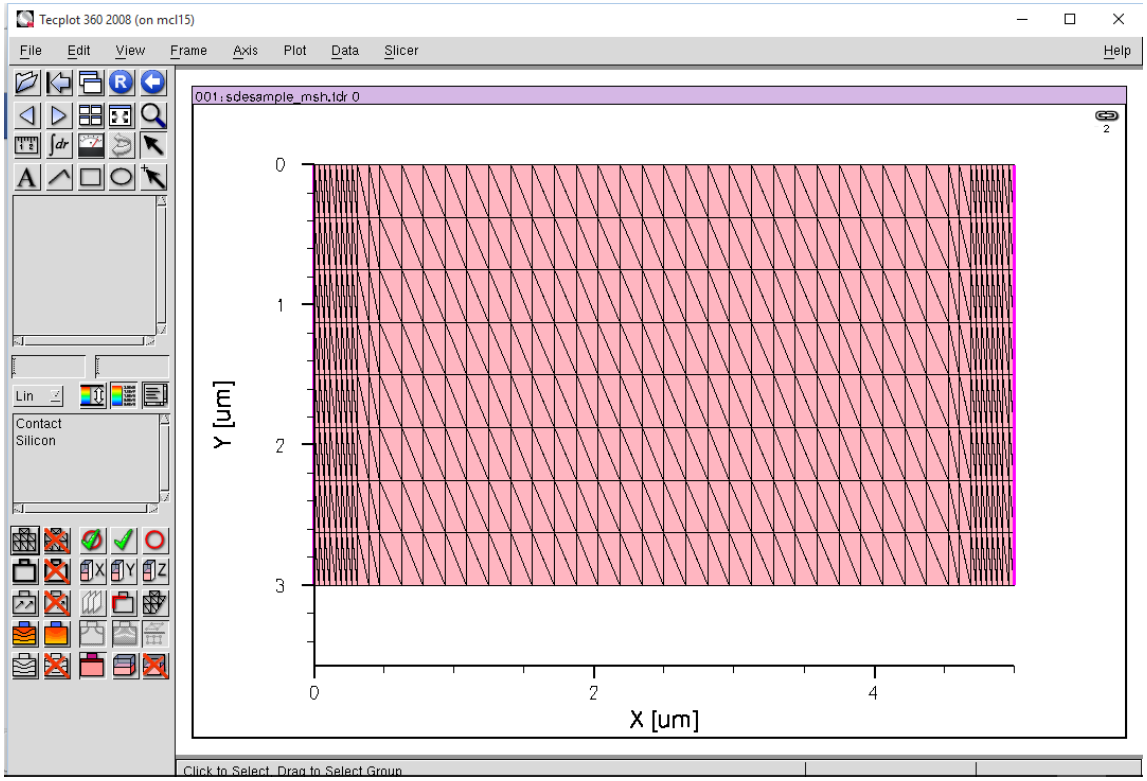
# Meshing(Contd..)

We build the mesh again and here is what we get



# Visualizing the model

- We can view our model in tecplot (command: ***tecplot_sv***) or svisual (command: ***svisual***)



# Basics of scripting

- An alternative way to use the SDE is scripting, which is often a convenient way.
- SDE uses LISP-like programming.
- It allows us to define our own set of variables for later use

**(define i 3) ;Variable definition**
**(define nm 1e-3) ;nanometer definition**
**(define thick (* 2 nm)) ; Definition of 2 nm thickness**

- Arithmetic expressions are defined as

**(define c (+a b)) ; c=a+b**

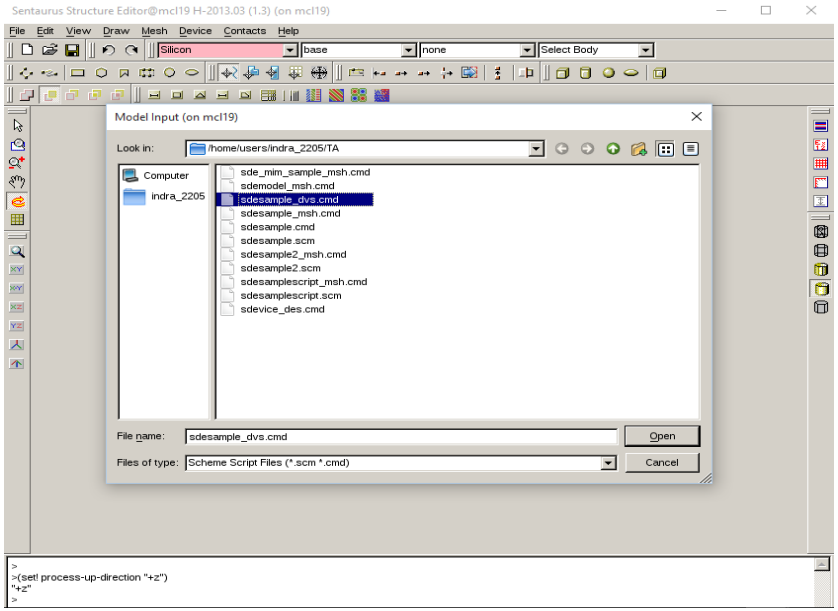
- These definitions help us defining co-ordinates for meshing and refinement windows.

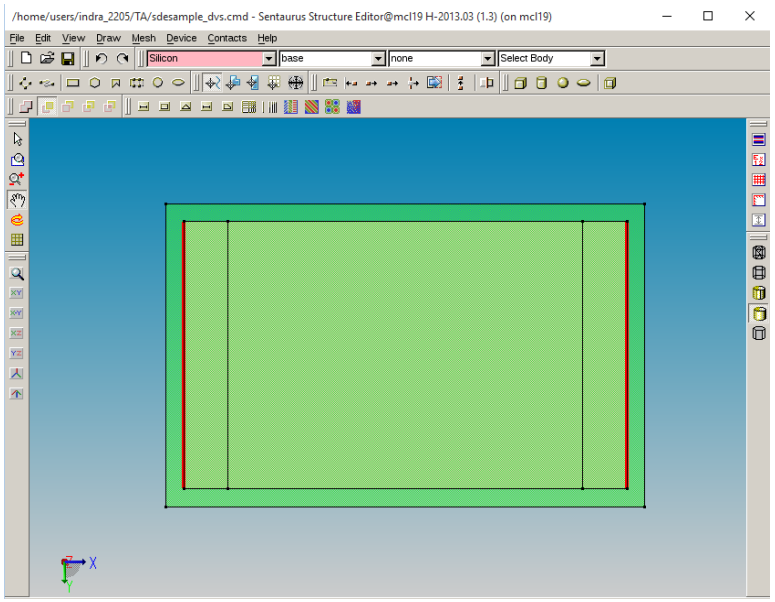
For e.g. if we want a rectangular window extending ‘x’ distance from edge of the large rectangle(as used in example),

**(define rec_edge 2)**
**(define window_edge (+x rec_edge))**
**(sdedr:define-refeval-window "RefEvalWin_n" "Rectangle" (position rec_edge y1 0) (position window_edge y2 0))**

# Importing a script into SDE

- Select File->Import->``abc.cmd''

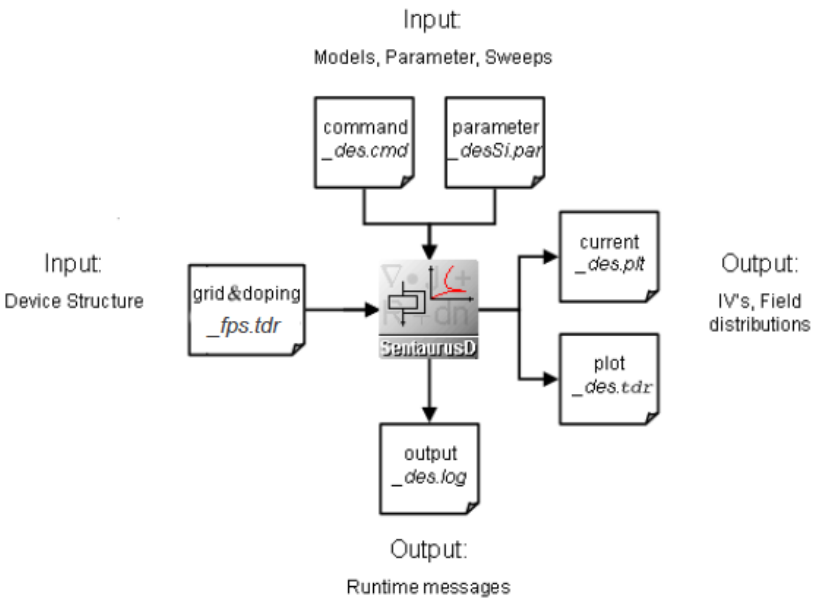


# Numerical solution

- Done using Sdevice
- The Sentaurus Device command file is organized in command or statement sections that can be in any order

**Flow of Input and Output in SDevice**



# Command File

A typical device file contains following sections:

- *File*
- *Electrode*
- *Physics*
- *Plot*
- *Math*
- *Solve*

# Command File

## File section:

This section contains input files included and output files of the simulation

```
File {
    *Input Files*
    Grid = "_msh.tdr"
    Parameter = “abc.par"
    *Output Files*
     Current = “abc"
    Plot = “abc"
    Output = “abc”
 }
```

# Command File

## Electrode section:

The electrical device contacts are declared in the Electrode section together with the initial boundary conditions (bias) and other optional specifications.

```
Electrode {
        { Name= "n1contact"   Voltage= 0.0 Schottky  WorkFunction  = wf1 }
        { Name= “n2contact"   Voltage= 0.0 Schottky  WorkFunction  = wf2 }
}
```

By default, Sentaurus Device treats an electrode as an ideal Ohmic contact

# Command File

**Physics section:**

The physical models that needs to be included in the simulation should be specified in the "Physics" section. The models you include should completely reflect the physical phenomenon happening in the device.

```
Physics {
    Mobility ( DopingDep )
    EffectiveIntrinsicDensity(BandGapNarrowing(oldSlotboom))
    Recombination (SRH Auger )
}
```

# Command File

## Plot section:

The parameters that you want to visualize in the plot file after the device simulation, need to be included here

```
Plot {
        Doping DonorConcentration AcceptorConcentration
        BandGap BandGapNarrowing ElectronAffinity
        ConductionBandEnergy ValenceBandEnergy
        eQuasiFermiEnergy hQuasiFermiEnergy
        eDensity hDensity
        EffectiveIntrinsicDensity IntrinsicDensity
        ElectricField/Vector
        Potential
        eMobility hMobility
        SRHRecombination AugerRecombination TotalRecombination SurfaceRecombination
 }
```

# Command File

## Math section:

This section directs the solver with information line, which numerical method to use during the solution, what should be the initial guess for each bias point, How many iteration it should wait to reduce the step size, if the convergence is not met. etc.,

```
Math {
        Extrapolate                 * switches on solution extrapolation along a bias ramp
        Iterations= 8               * maximum-allowed number of Newton iterations (3D)
        Method= Blocked             *  default solvers for Coupled
         RelErrControl
}
```

# Command File

## Solve section:

This is the actual section in which the solver is directed with the set of equations to solve. It consists of a series of simulation commands to be performed that are activated sequentially.

```
Solve{
        Poisson
        Electron
        Hole
        Coupled { Poisson Electron Hole }
        NewCurrentFile="forwardbias"
        Quasistationary (Initialstep= 0.025 MaxStep = 0.01 MinStep = 1e-8
                Goal { Name = “n1contact" Voltage =-5})
                { Coupled { Poisson Electron Hole } }
    }
```

***Order of commands is important here***

# Tdr files

- We can generate several tdr files by inserting regular halting commands.
- For e.g, if we want to halt the simulation at 0.1V as visualize the tdr at that voltage, we can use

```
 Quasistationary (Initialstep= 0.025 MaxStep=0.05
MinStep=0.0001
     Goal{ Name="n2contact" Voltage= 0.1})
     { Coupled { Poisson Electron hole}
     plot(FilePrefix="0.1V_neg" )
     CurrentPlot ( Time = (range = (0 1) intervals = 50))
      }
```

- The plot Fileprefix command saves a tdr file at that voltage.

# Parameter File

- Command to generate the default parameter file:

  ***sdevice -P sdevice_file.cmd***

- This command generates the file models.par, which contains all the default model parameters for the silicon material.

- To change the parameters for a model, for example, the parameters used by the Shockley–Read–Hall (SRH) model and to force Sentaurus Device to use the modified model parameters instead of the built-in defaults.

# Parameter File

- A sample section in parameter file looks like

```
* Region = "region_1" Material = "Silicon"
* Electrode = "n1contact"
* Electrode = "n2contact"


Material = "Silicon" {

Epsilon
{ *  Ratio of the permittivities of material and vacuum

  * epsilon() = epsilon
  epsilon = 11.7  # [1]
}


Epsilon_aniso
{ *  Ratio of the permittivities of material and vacuum

  * epsilon() = epsilon
  epsilon = 11.7  # [1]
}
```

- We can change these values for our own convenience.

# Result analysis

- Tecplot SV
- Inspect

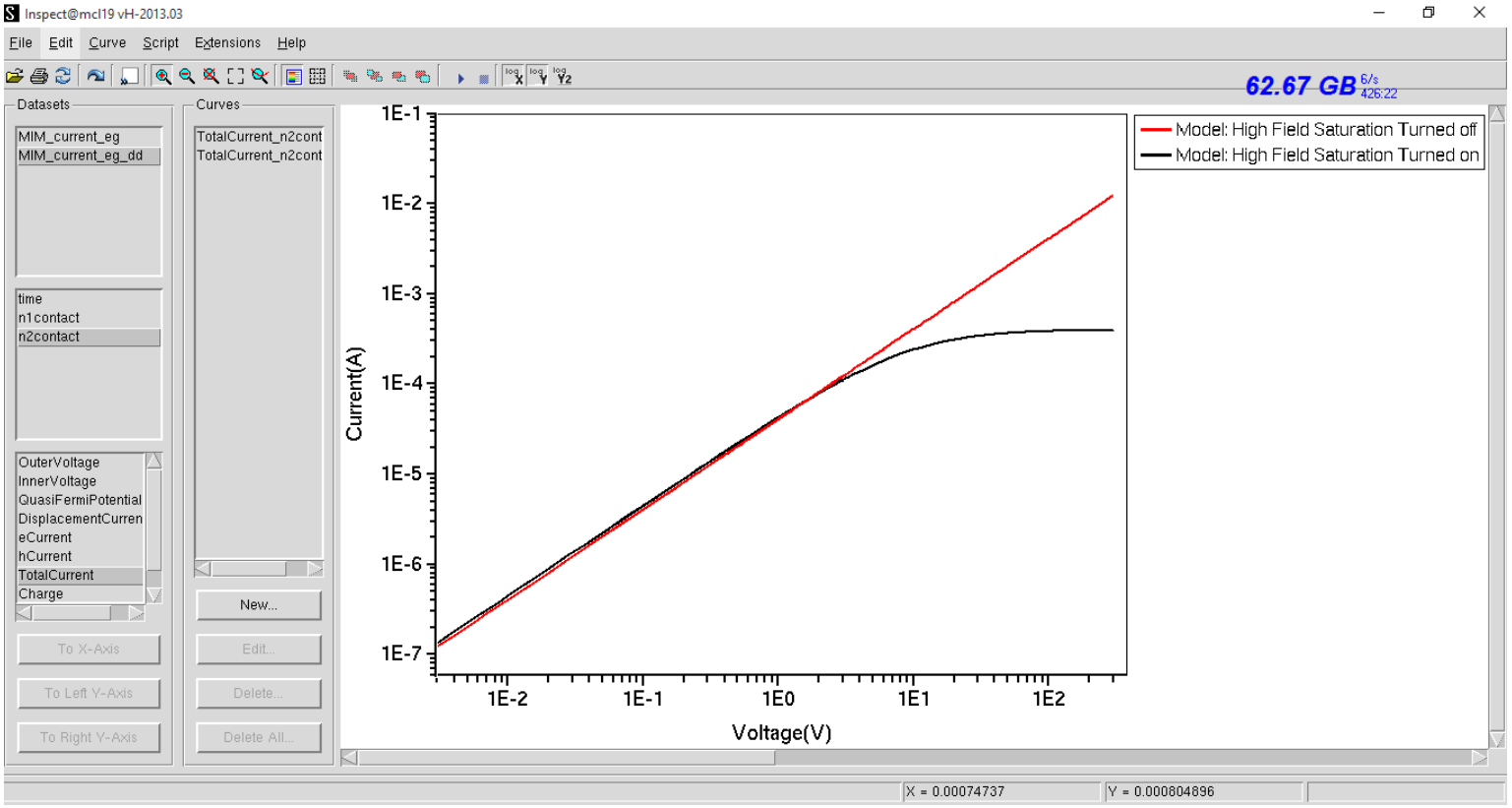
# Inspect

- Inspect is a curve display and analysis program. It works with curves specified at discrete points. Inspect enables users to work interactively with data using both a graphical user interface and a script language

- Inspect can be started from Sentaurus Workbench or from the command line by typing: ***inspect***

# Viewing Output in inspect

File-> Load Dataset-> “abc.plt” -> Select Electrode-> Select Parameter to Plot



# Tecplot SV

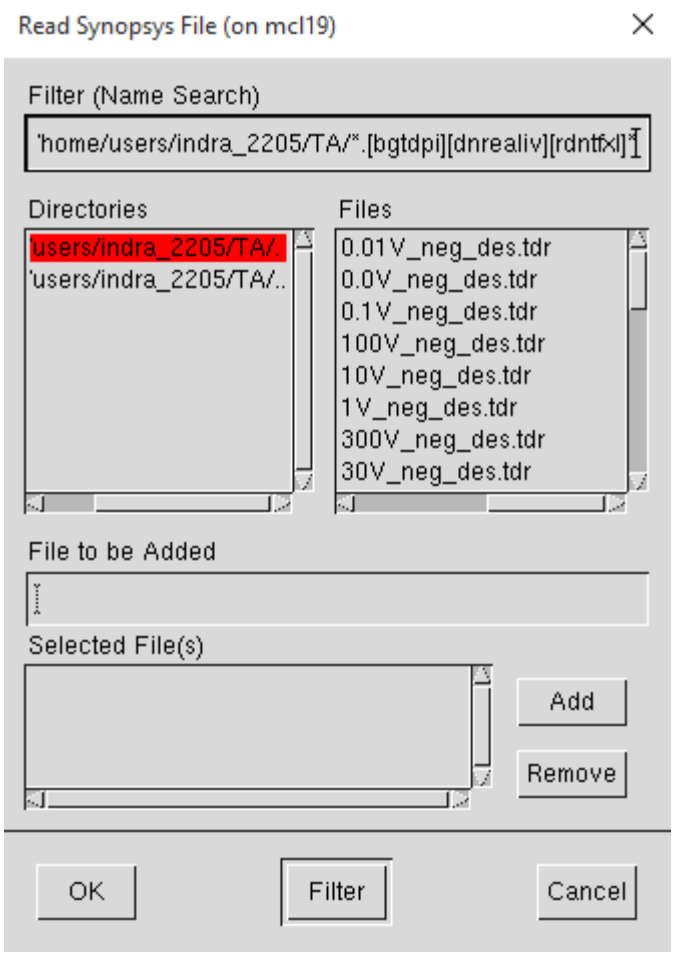
- Tecplot SV is part of Sentaurus Workbench Visualization. It is plotting software with extensive 2D and 3D capabilities for visualizing data

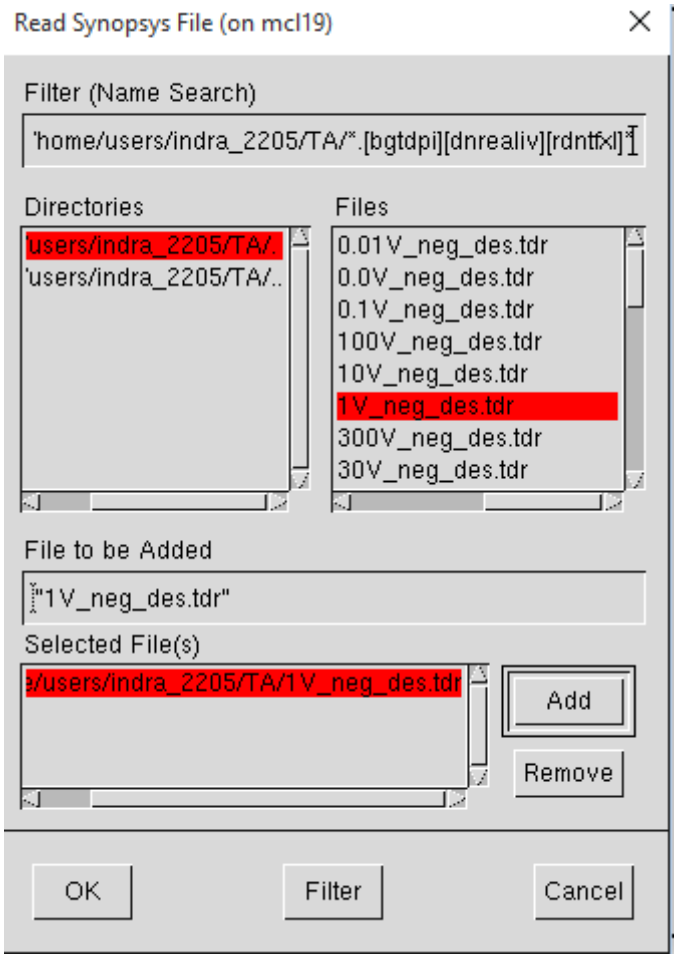
- The command tecplot_sv is used to start Tecplot from the command prompt, for example: ***tecplot_sv n2_fps.tdr***
  where *n2_fps.tdr* is the name of file.

# tecplot

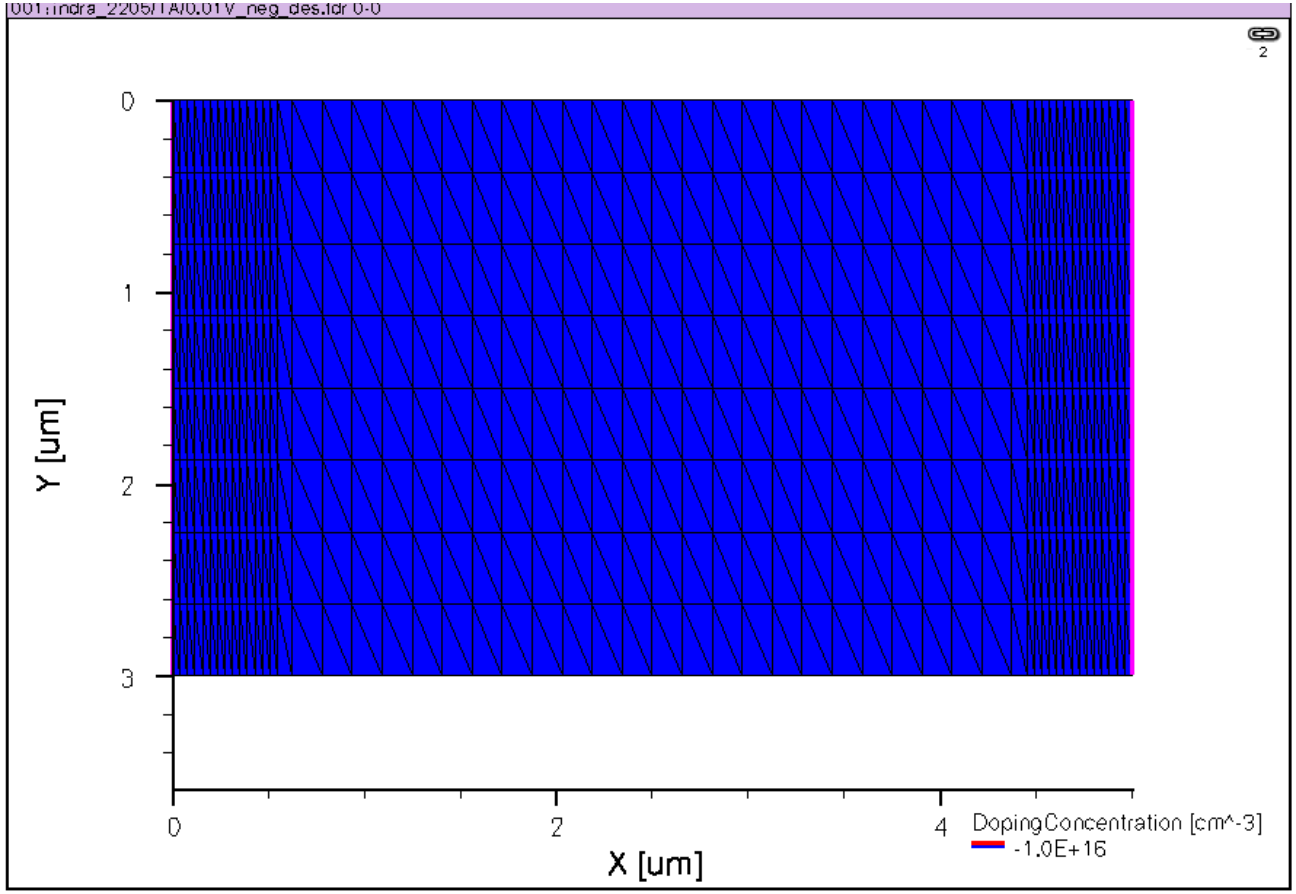
To load data files from an open Tecplot SV interface:

**File** > **Load**



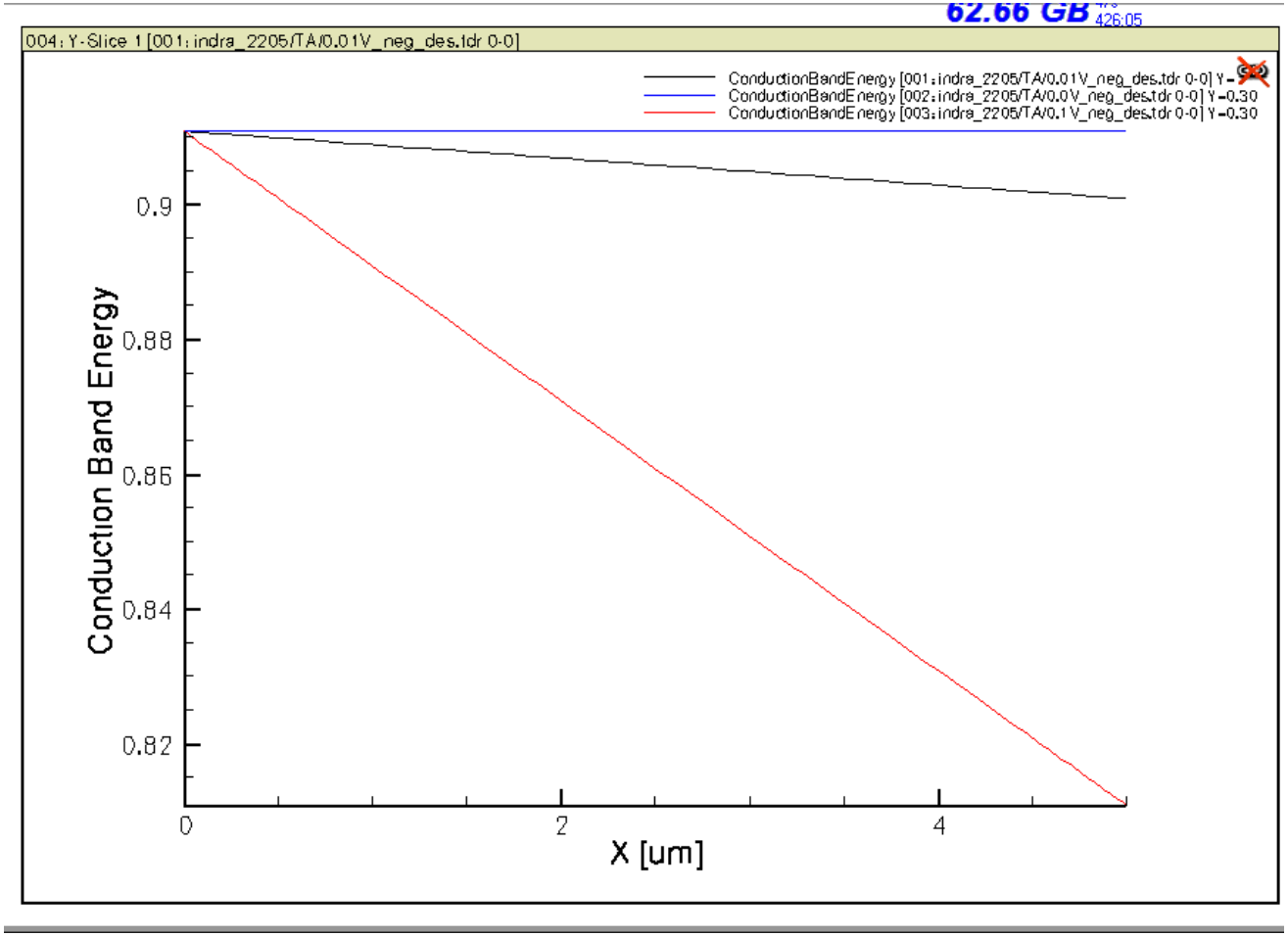
# Visualizing the structure



# Plots from Tdr files

- We can plot all the parameters that we included in the plot section in tecplot/svisual visualization of the tdr file
- For example if we want to plot Conduction Band energy, we need to take a y-cut along the structure and the select Conduction Band energy

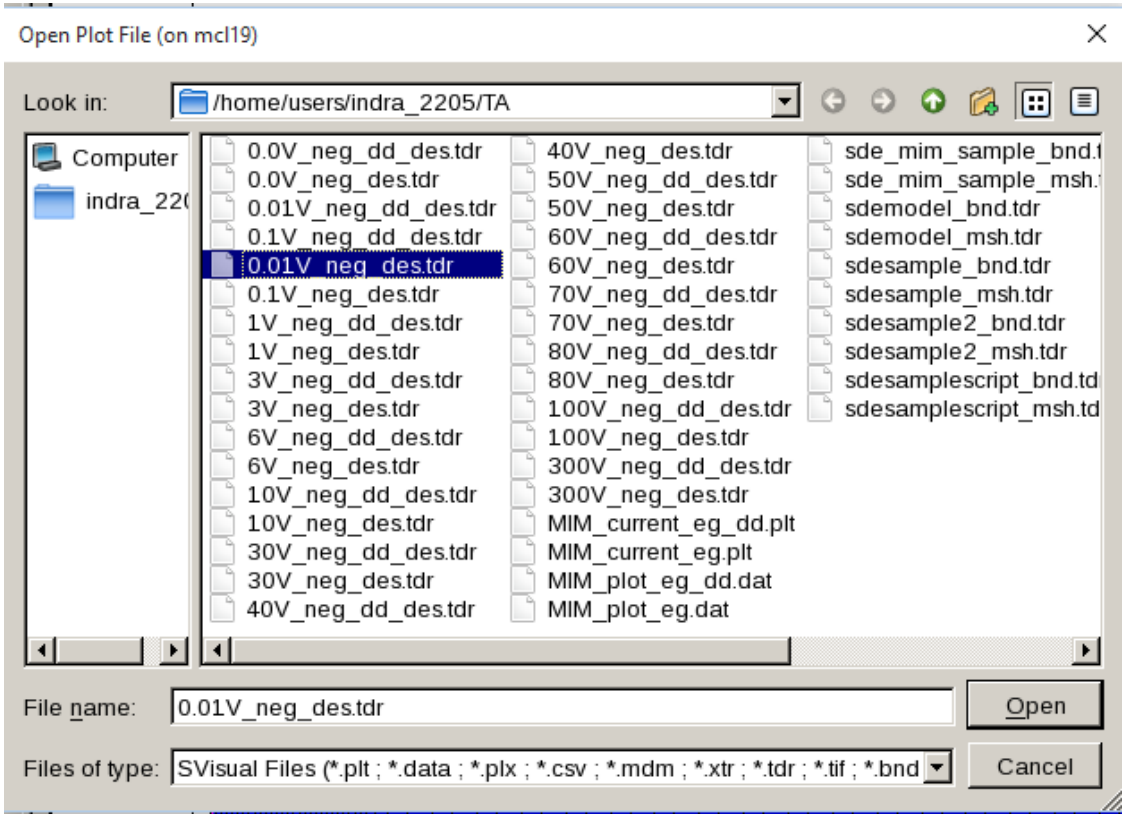
# Conduction Band energies



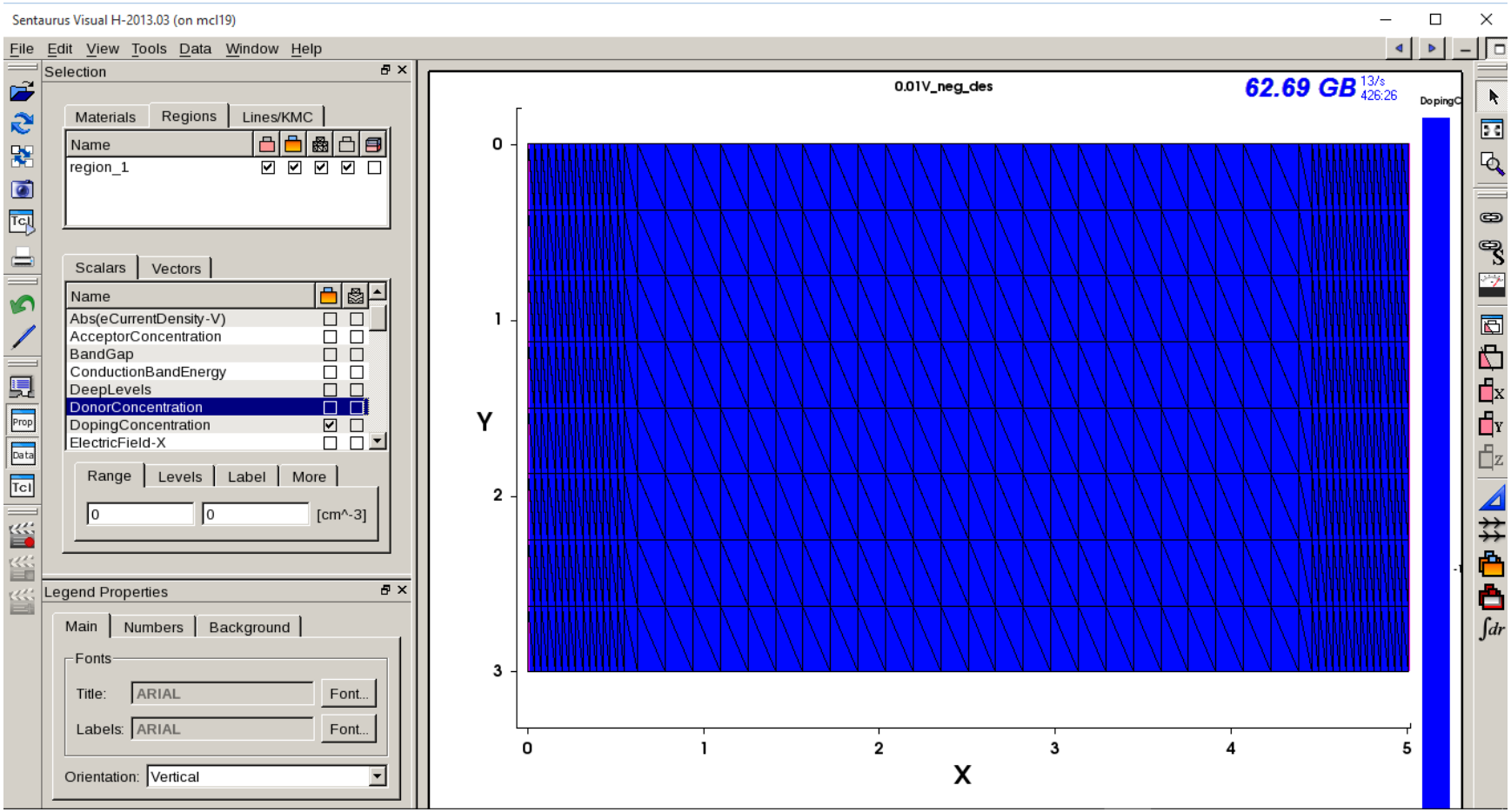
# svisual

- Command to open svisual: ***svisual***
- To load data files from the graphical user interface (GUI) of Sentaurus Visual:

**File > Open**

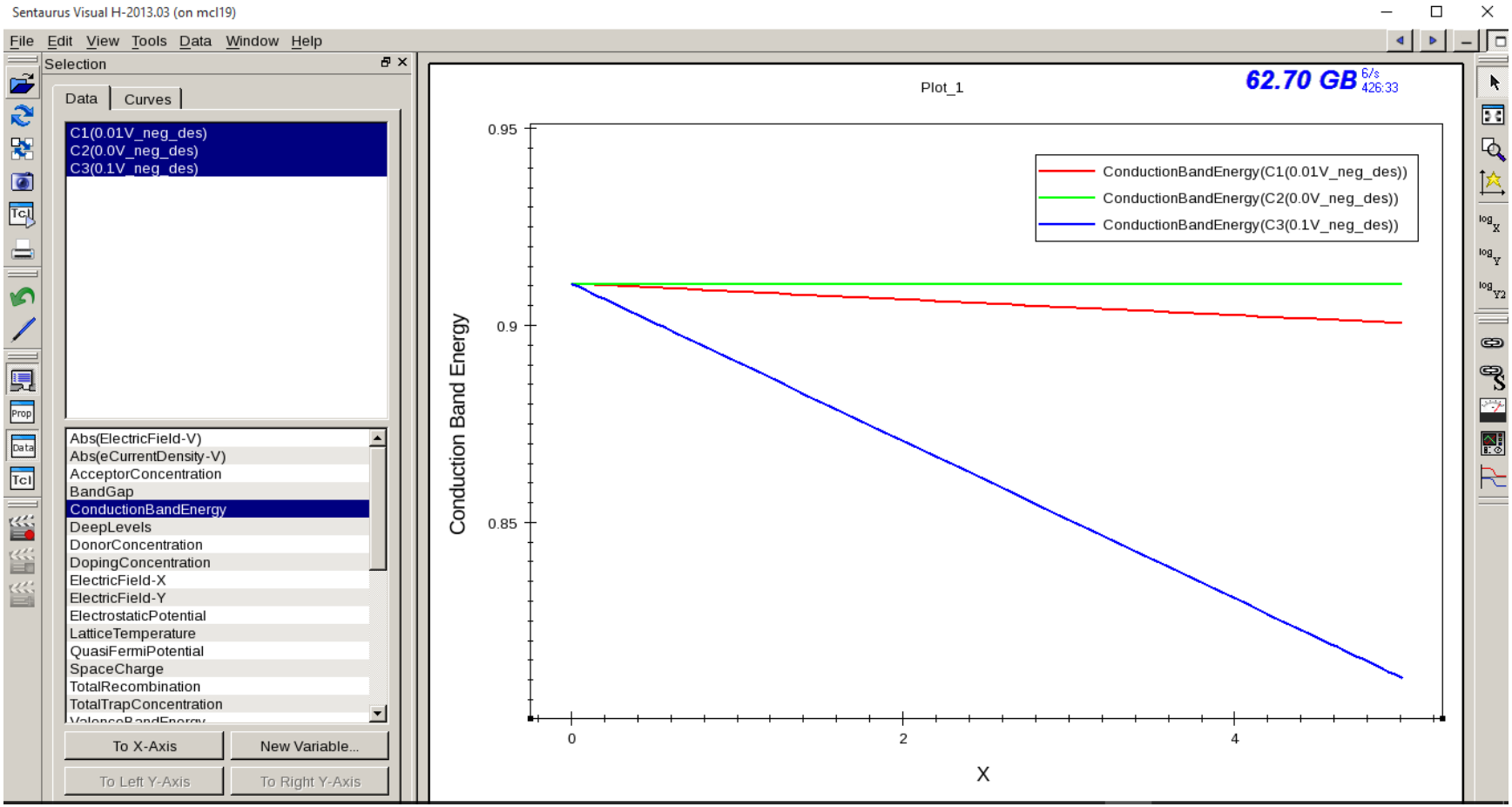


# svisual


Sentaurus Visual H-2013.03 (on mcl19)
File Edit View Tools Data Window Help
Selection
Materials Regions Lines/KMC
Name
region_1
Scalars Vectors
Name
Abs(eCurrentDensity-V)
AcceptorConcentration
BandGap
ConductionBandEnergy
DeepLevels
DonorConcentration
DopingConcentration
ElectricField-X
Range Levels Label More
0
0
[cm^-3]
Legend Properties
Main Numbers Background
Fonts
Title: ARIAL Font...
Labels: ARIAL Font...
Orientation: Vertical
0.01V_neg_des
62.69 GB
13/s
426:26
DopingC
Y
X
0
1
2
3
4
5

# Plotting Tdr files in svisual

- Select tdr file
- Take y-cut
- Select Parameter



# Sentaurus Manuals and Training

- For Sentaurus Training, go to directory mentioned below and open index.html in firefox browser.

  cd /usr/local/Softwares/Sentaurus/Sentaurus_vF_2011.09/tcad/F-2011.09/Sentaurus_Training/

- And manuals are in the following directory

  cd/usr/local/Softwares/Sentaurus/Sentaurus_vF_2011.09/tcad/F-2011.09/manuals

Note: Various versions are available. You can access manuals from any of those. Ex:

cd/usr/local/Softwares/Sentaurus/Sentaurus_vO_2018.06/O_2018.06-SP1/tcad/O-2018.06-SP1/manuals/PDFManual/data/

Thank You