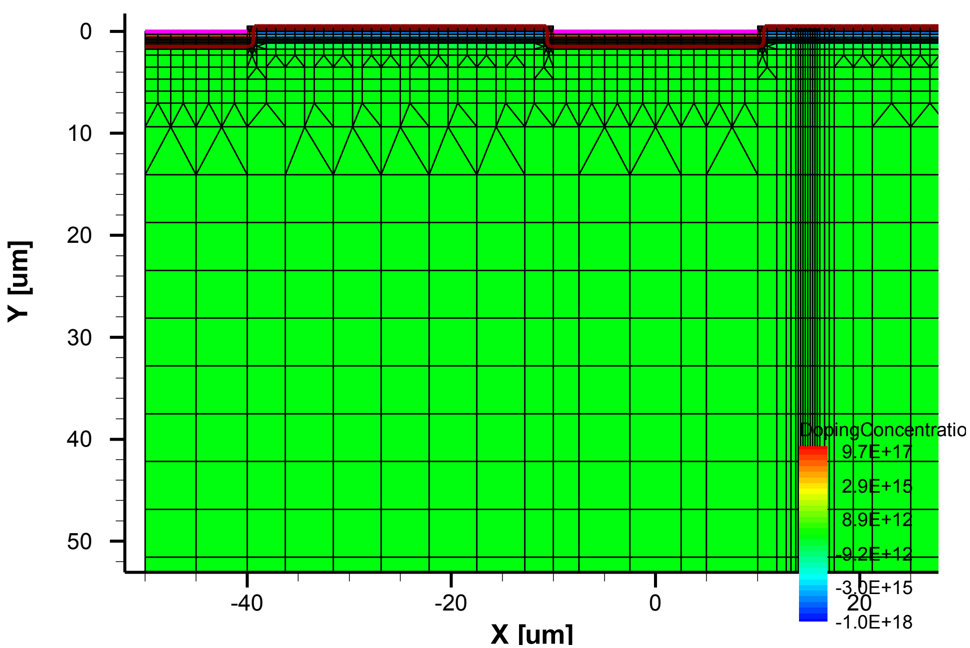
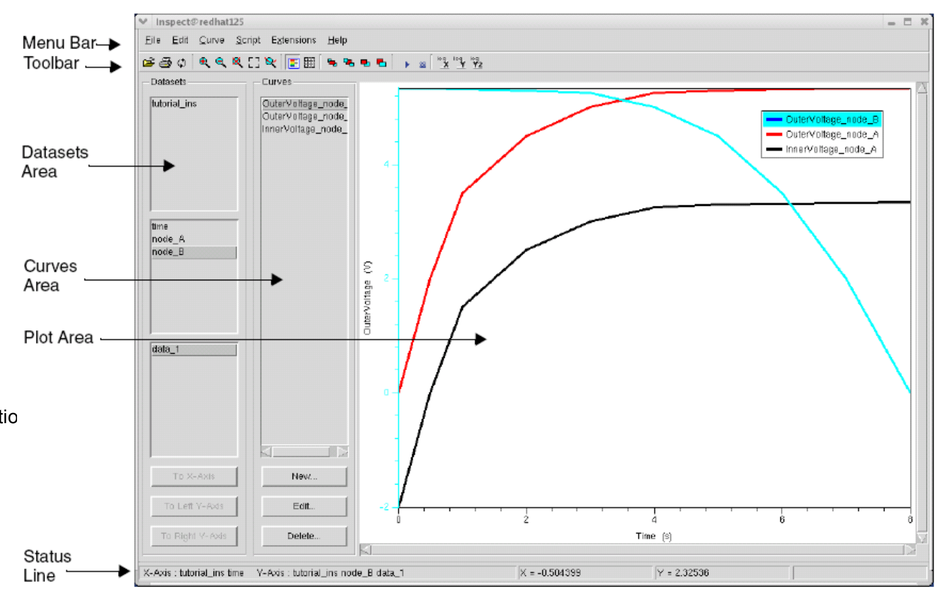
Introduction to Sentaurus TCAD

David Pennicard – University of Glasgow - d.pennicard@physics.gla.ac.uk

For more information on Sentaurus TCAD, visit:

http://ppewww.physics.gla.ac.uk/det_dev/activities/threedee/Documents/BarcelonaSeminar.html





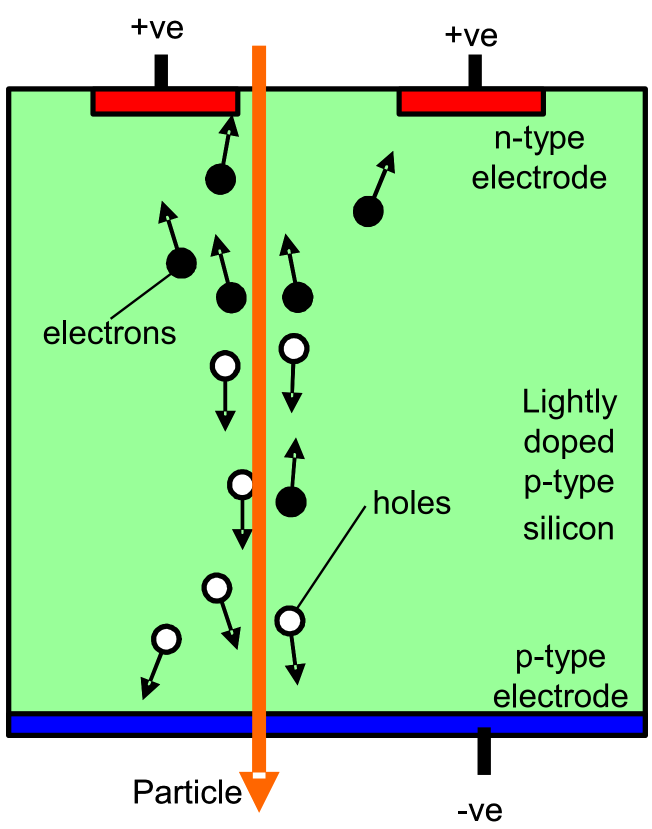
Overview

- Introduction to Sentaurus TCAD software
- Building the device structure
- Running the simulation
- Viewing results
- Other software

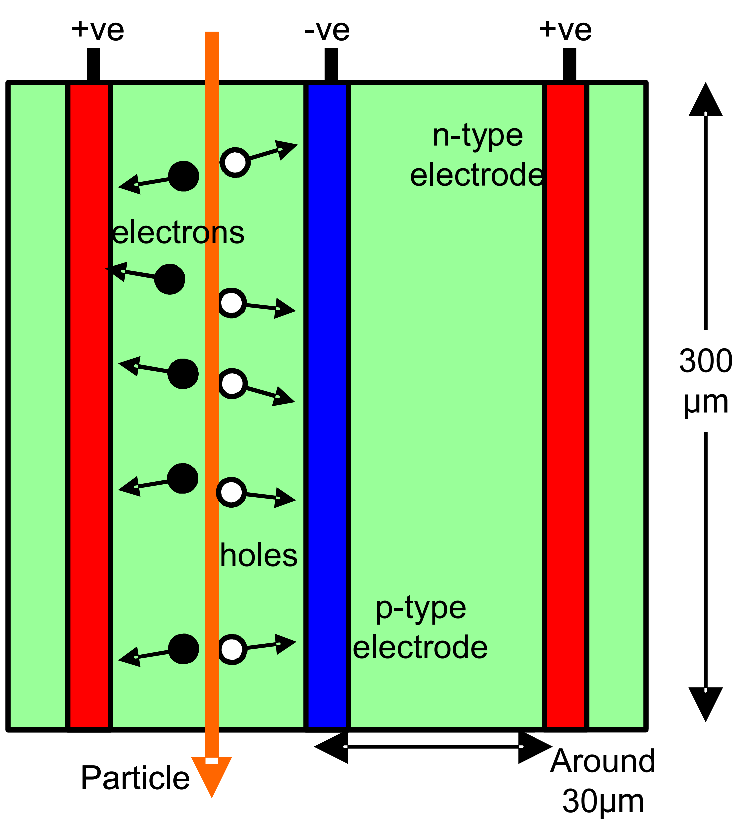
Example simulations – 3D detectors

- 3D detector – photodiode detector with electrode columns passing through substrate
  - Small electrode spacing gives fast collection, low V_{dep}
  - Radiation hardness

Planar

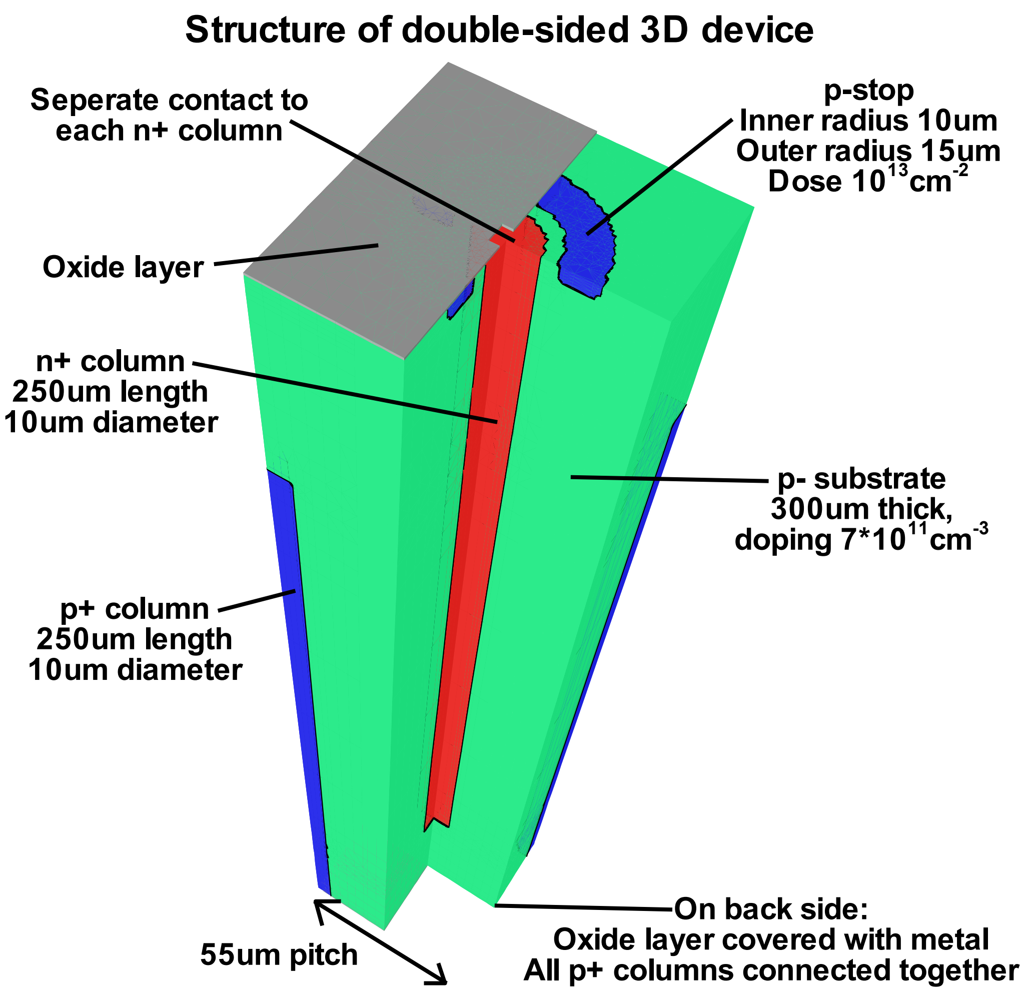


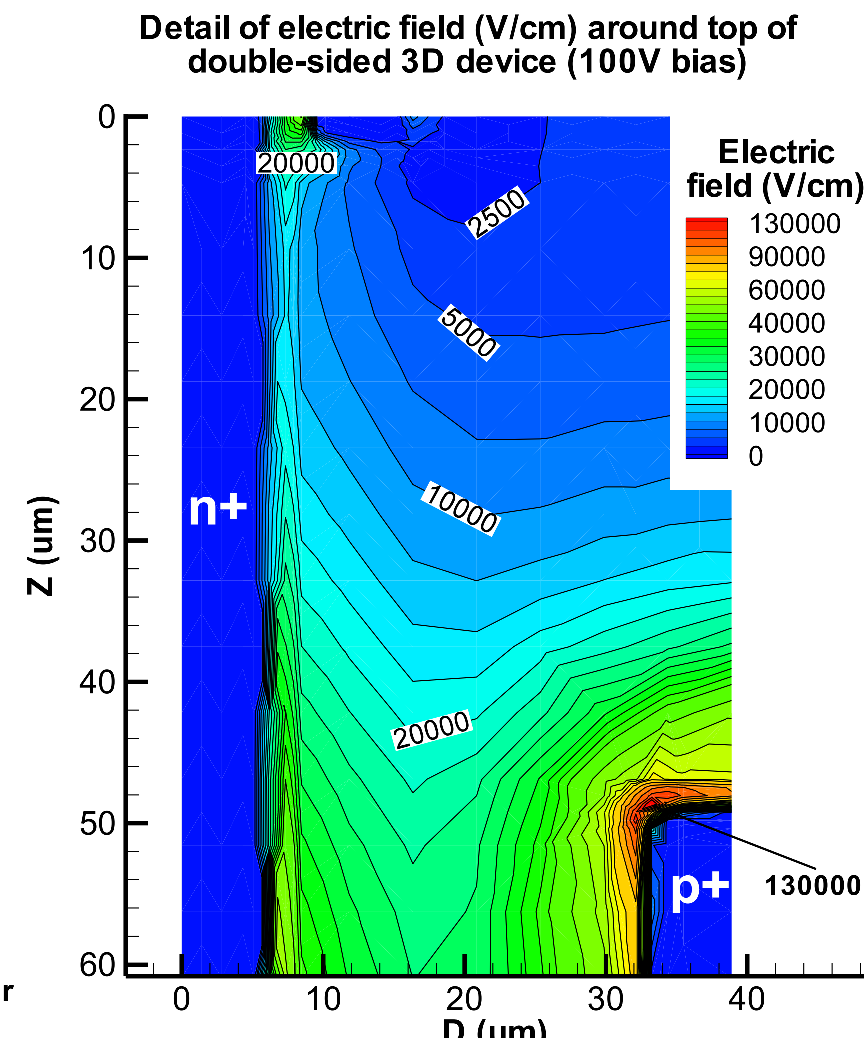
3D



Example simulations – 3D detectors

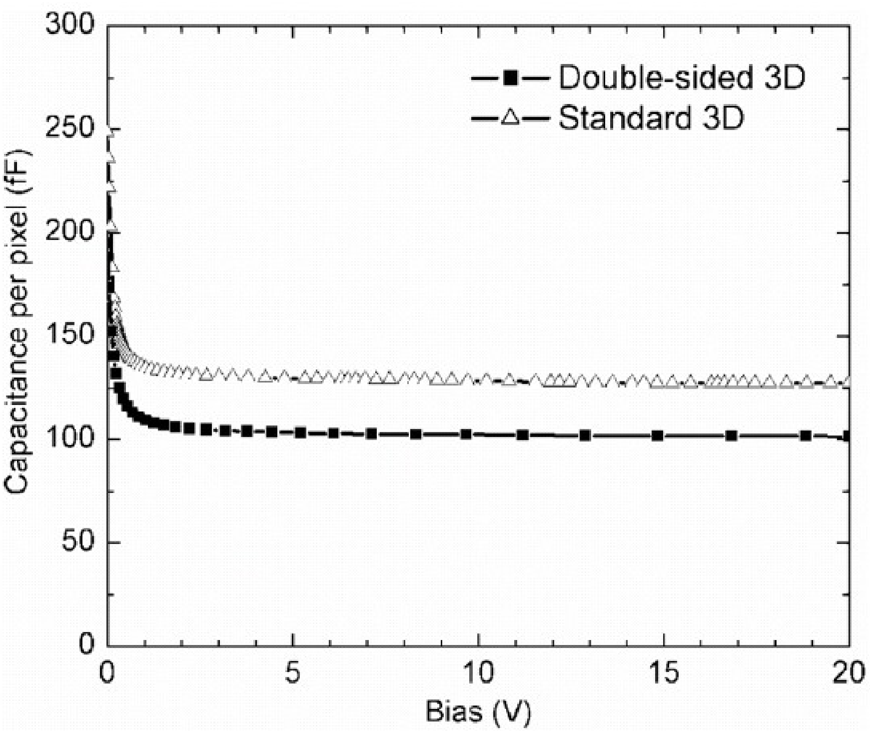
Electric field pattern in a new device structure



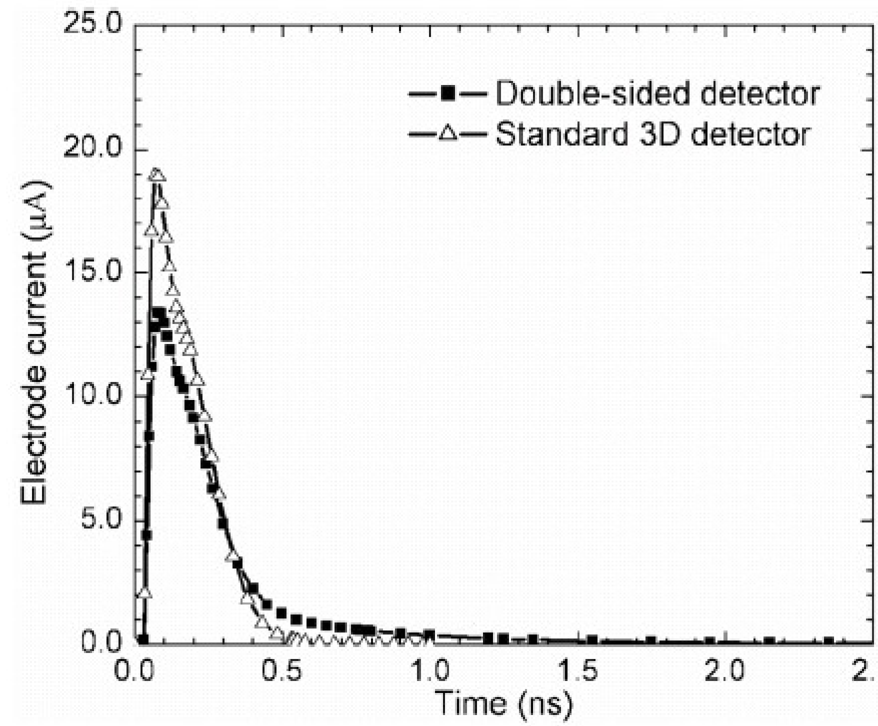


Example simulations – 3D detectors

Capacitance-voltage characteristics

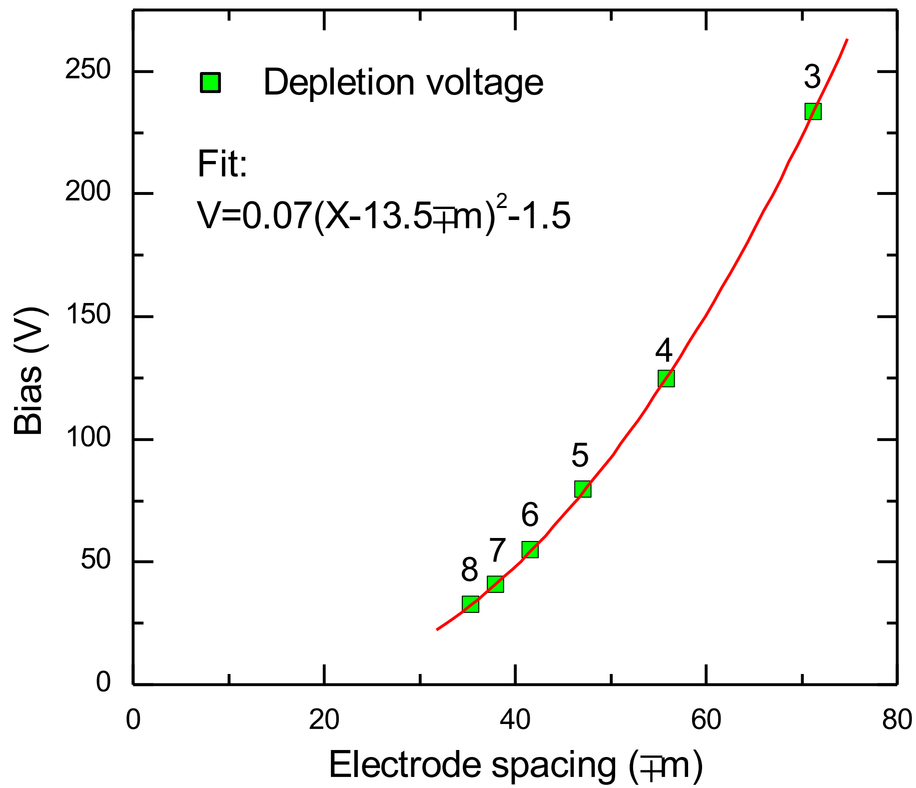


Current pulse produced over time as particle hits detector

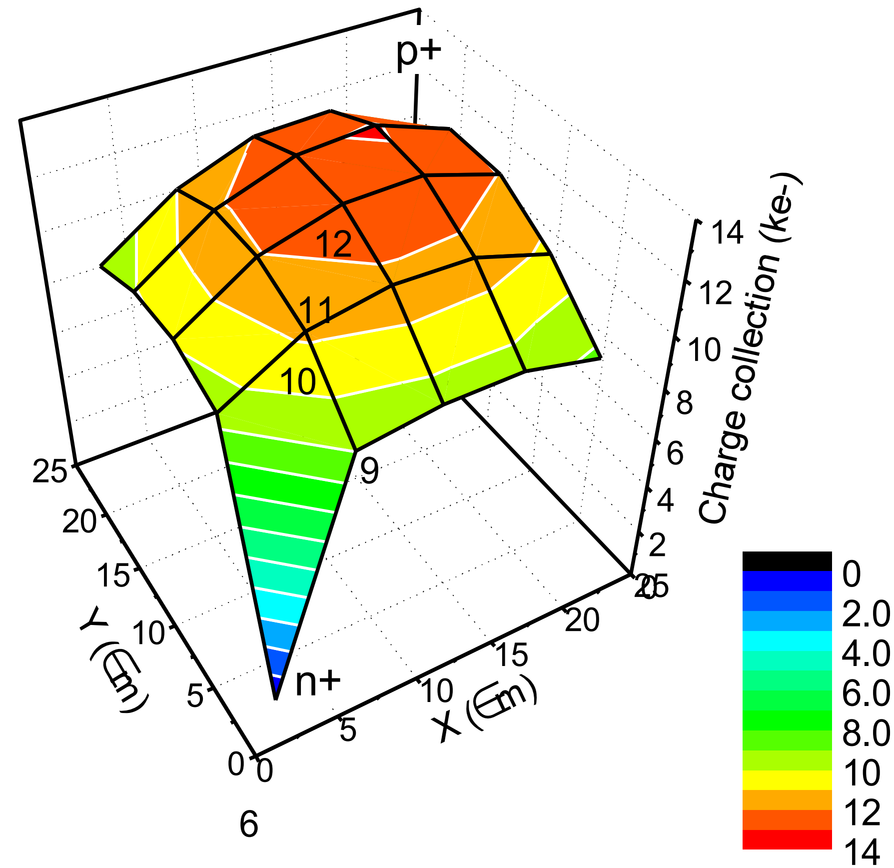


Example projects – 3D detectors

Multiple depletion simulations find V_{dep} for possible ATLAS 3D structures (after radiation damage)



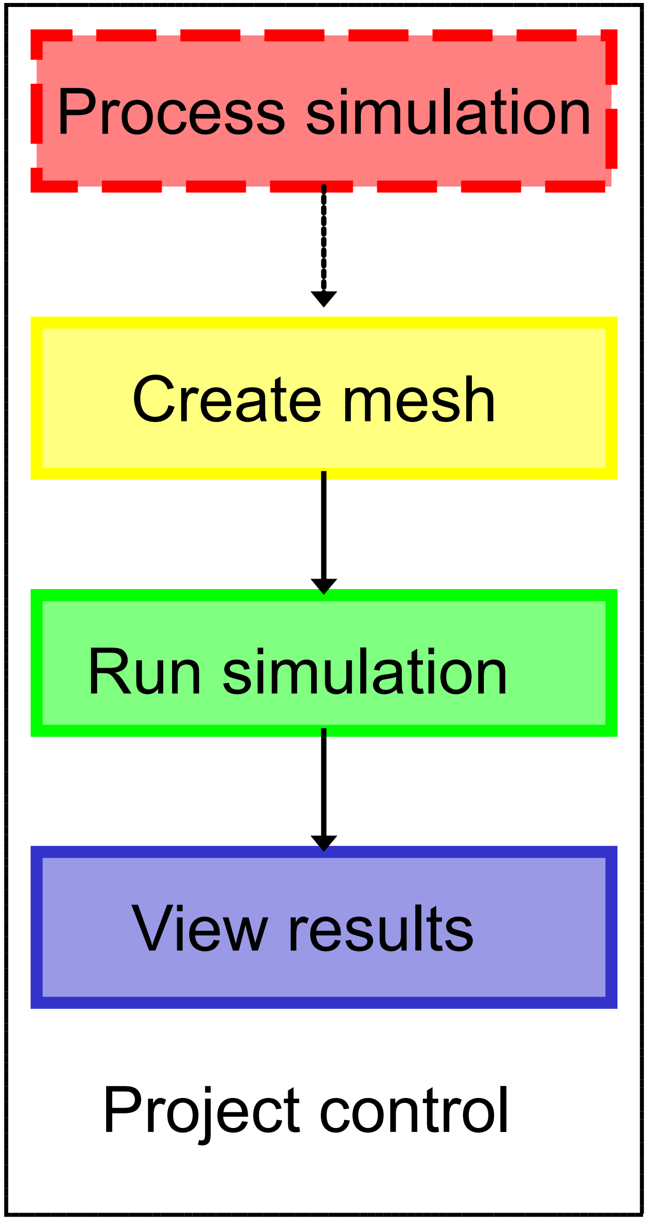
Multiple particle track sims map charge collection with hit position (after radiation damage)



Basics of simulation

- The structure of a device is approximated by a “mesh” consisting of a large number of discrete elements
  - This can be 1D, 2D or 3D
  - Choice depends on symmetry of device
- Differential equations describing the electric potential and carrier distributions are applied to each element
  - End up with very large number of equations!
- Choose boundary conditions for the simulation
  - E.g. potentials at each electrode
- Solve the equations to find the potential and carrier concentrations in each element
  - Software uses a numerical solver – iterates repeatedly until solution is accurate enough

Simulation packages



Sentaurus Process (optional) {dios}

Ligament can generate command files for Process

Mesh, noffset3d create device meshes using a command file

Sentaurus Structure Editor (and MDraw) create meshes with GUI

Sentaurus Device {dessis}

Inspect – Plotting graphs of electrode currents etc.

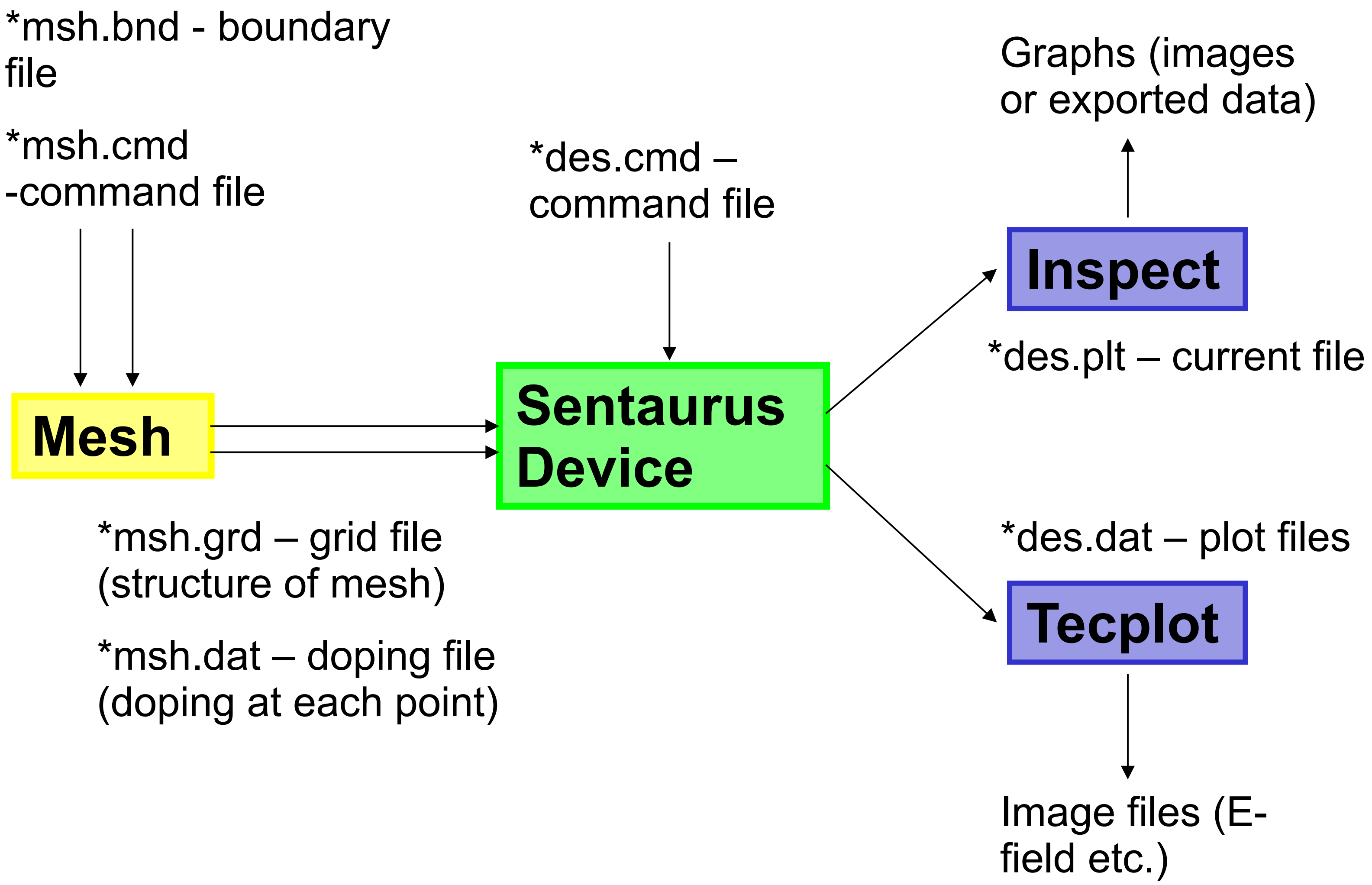
Tecplot – Producing images of electric field patterns etc. throughout device

Workbench – Can run large numbers of simulations conveniently

Example of simulation flow

Strip detector simulation

Files available in SENTAURUS/Seminar/Introduction

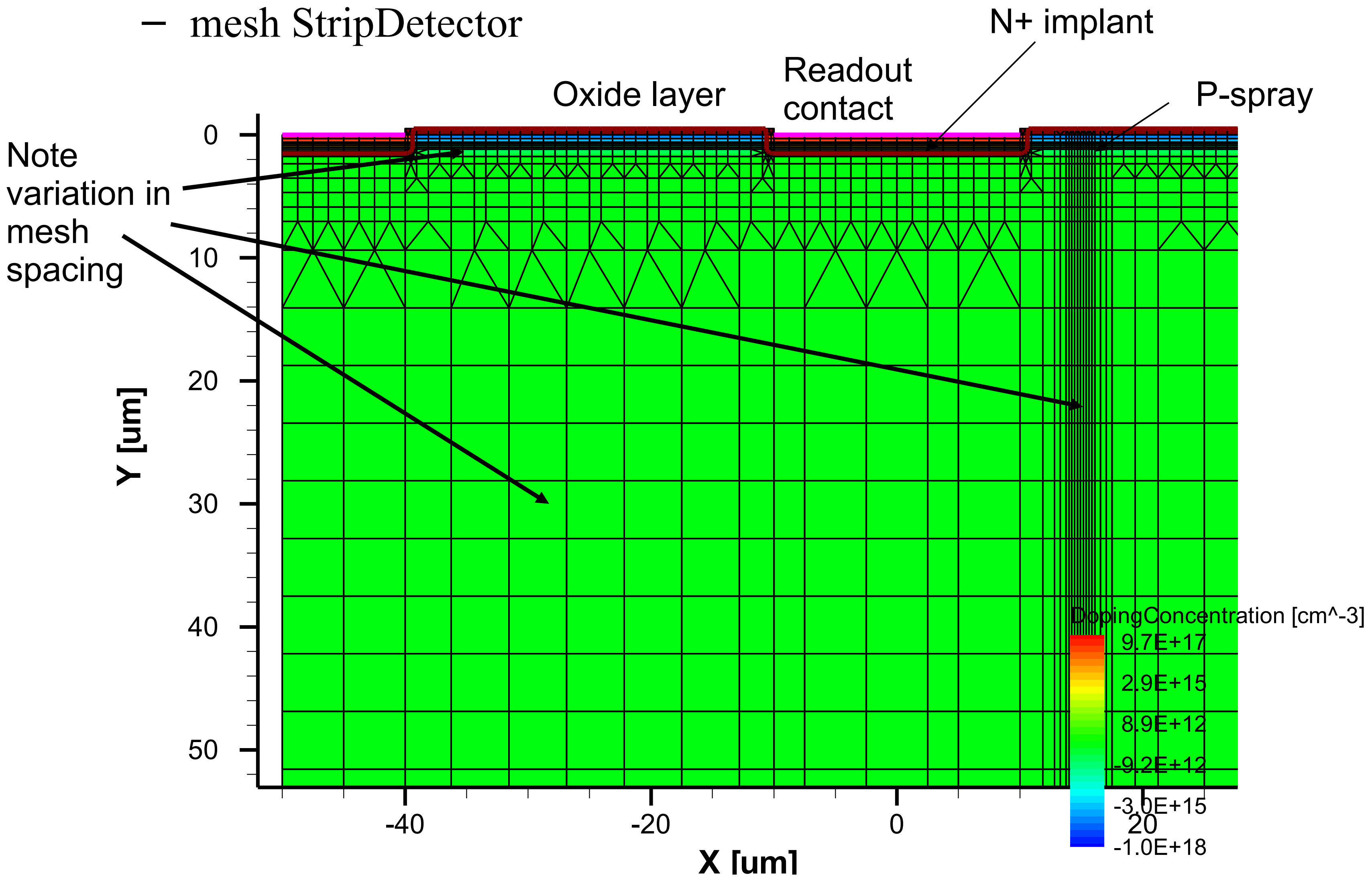


Overview

- Introduction to Sentaurus TCAD software
- Building the device structure
- Running the simulation
- Viewing results
- Other software

Mesh

- Mesh takes a “boundary” (.bnd) and command (.cmd) files as arguments:
  - mesh StripDetector



Mesh input files

- *StripDetector.bnd* boundary file describes materials & contacts

```
Silicon "substrate" {rectangle [(-50,0) (50,300)]}

Oxide "TopOxide1" {rectangle [(-40,-0.5) (-10,0)]
          rectangle [(10,-0.5) (40,0)]}

Contact "nplus2" {line [(-10,0) (10,0)]}…….
```

- *StripDetector.cmd* describes doping profiles and mesh refinement
  - These are defined first, then placed
  - Doping:

```
Definitions {
    AnalyticalProfile "n-plus electrode"
        {
        Species="PhosphorusActiveConcentration"
        Function = Gauss(PeakPos=0, PeakVal=1e18, ValueAtDepth=1e+12, Depth=1)
        Lateralfunction=Gauss(Factor=0.8)
        }
    ……
}
```

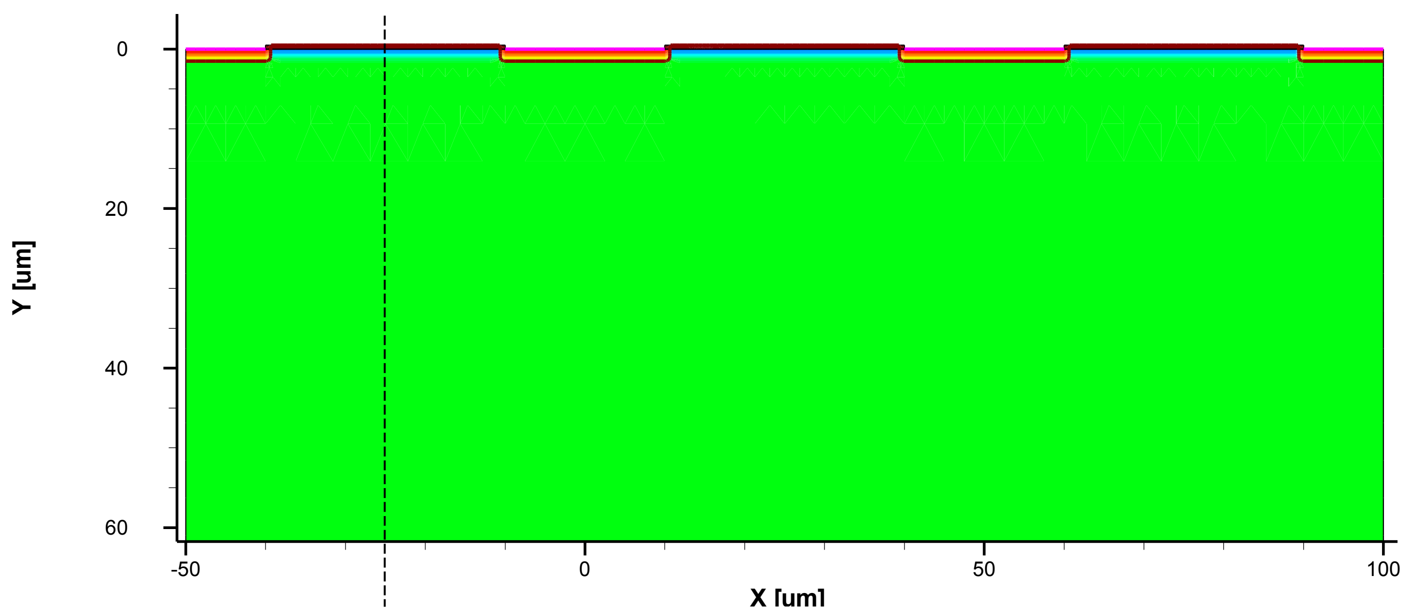
Mesh input files

- Mesh refinements:
  - Small elements = more accurate but slower simulation
  - So, use refinement statements to get smallest spacing in regions with doping profiles, high electric fields, charge generation etc.
  - 3D simulations have more elements, and run far slower, so good mesh design is crucial!

```
Definitions {
    Refinement "n-electrode"
        {
        MaxElementSize = (2.5 2)
        MinElementSize = (0.1 0.1)
        Refinefunction = MaxTransDifference(Variable="DopingConcentration", Value=2)
        }
……}

Placements {
    Refinement "n-electrodes instance"
        {
        Reference = "n-electrode"
        RefineWindow = rectangle [(-50 0), (50 3)]
        }
.......}
```

Mesh design considerations

- Boundary conditions
  - Default boundary conditions are that E and carrier currents perpendicular to boundary are zero
  - So, boundaries should either be far enough from active region not to have any effect, or along a line of symmetry
  - Mesh design depends on application – a simple electric field simulation can simply use the smallest repeating region of the device
    - Mesh for simulating charge sharing in strip detector below: has 2 full electrodes which we use to measure charge sharing, and 2 half-electrodes to approximate the "rest of the device"



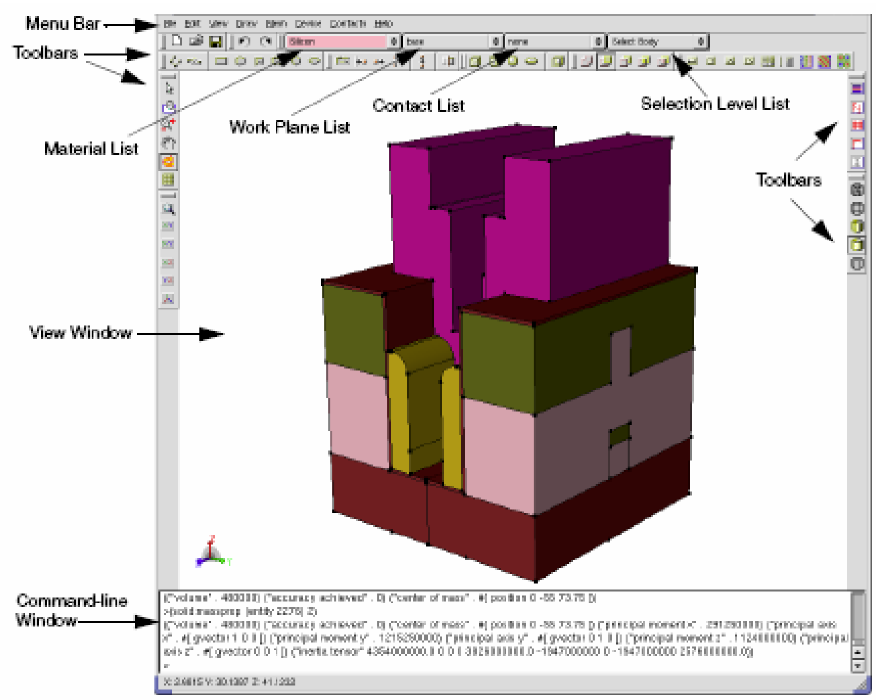
MDraw

- Old graphical interface for designing meshes
- Can be used in “boundary” and “doping” modes
- Materials, contacts, doping and refinement regions can be drawn in
- Then, will call on “mesh” to build the mesh

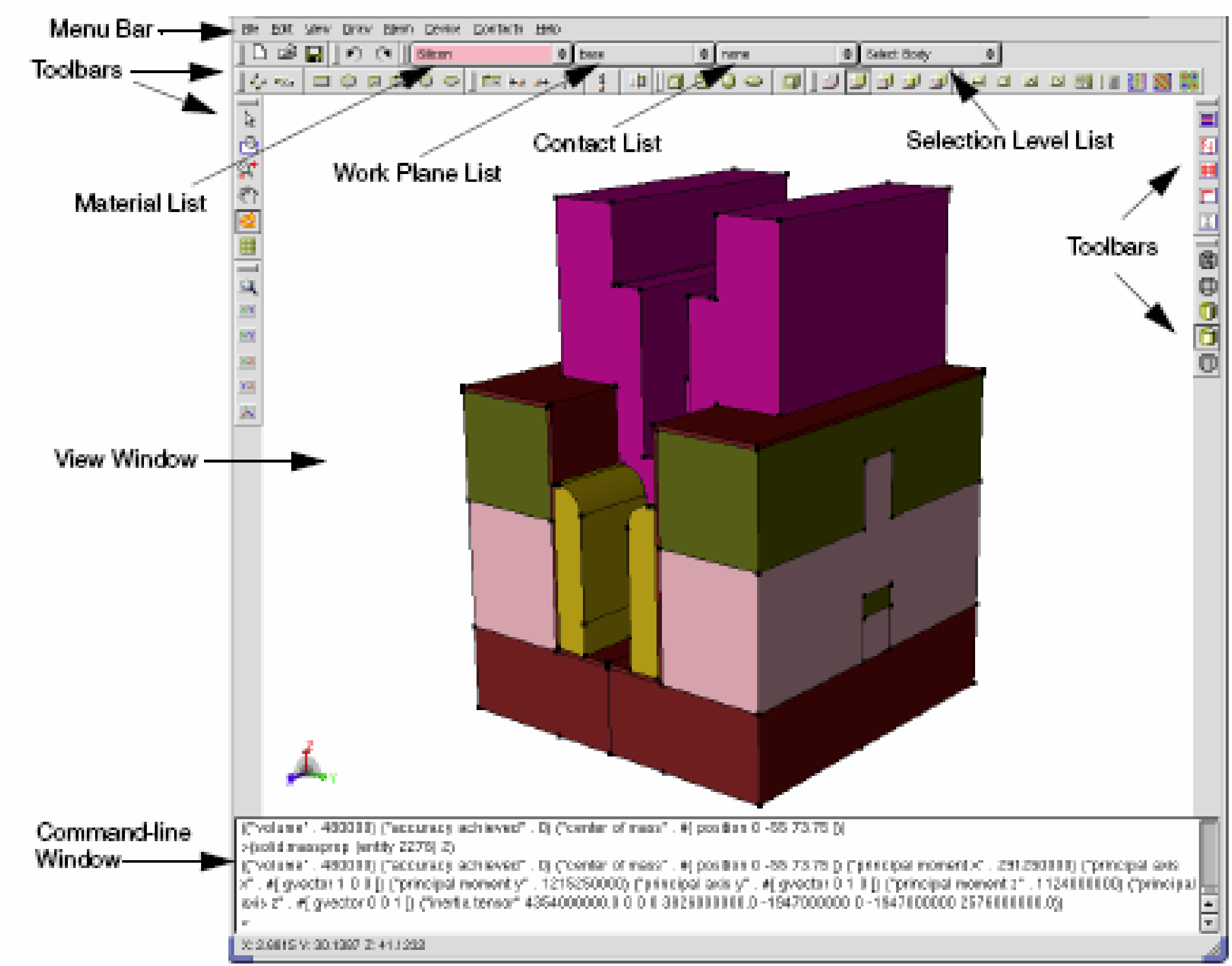
- Downsides:
  - Can’t be used for 3D meshes
  - Can’t add parameters in Workbench
  - Can’t be used with NOffset3D (see later)

Sentaurus Device Editor

- New feature of Sentaurus TCAD
- Start with sde
- Can work in 2D and 3D modes
- Has functions for complicated shapes like circles, spheres etc.
  - In command files or MDraw, these must be built up point-by-point, which is very inconvenient
- Has a built-in command line, and can be controlled with scripts
  - In 3D, easier than using mouse!
  - Possible to insert parameters using Workbench

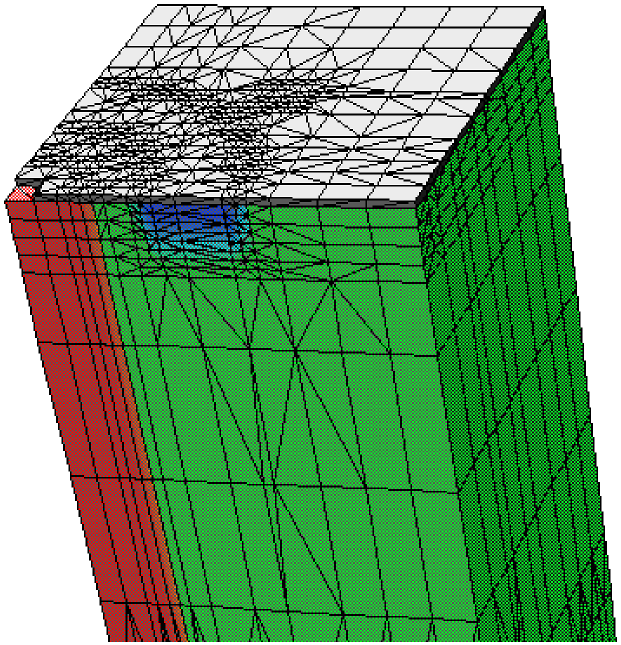


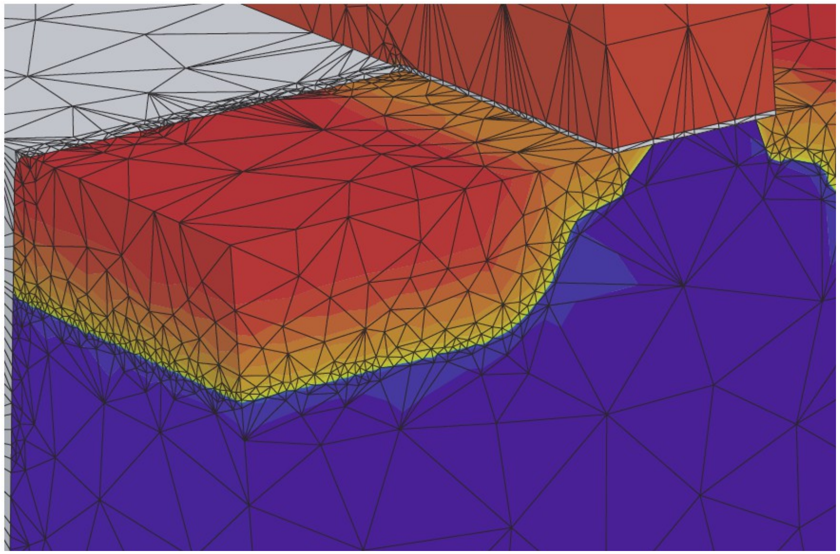
Sentaurus Device Editor



New mesh tool – NOffset3D

- noffset3d can be run using command files or through Structure Editor, just like mesh
- Other mesh tool produce axis aligned meshes
- This tool produces unstructured meshes
  - More effective for creating curved structures
- Input command files more complicated – see “Mesh Generation Tools User Guide”





Overview

- Introduction to Sentaurus TCAD software
- Building the device structure
- Running the simulation
- Viewing results
- Other software

Sentaurus Device

- Takes mesh, applies semiconductor equations and boundary conditions (in discrete form) and solves
- **Physics models:** Works by modelling electrostatic potential (Poisson's equation) and carrier continuity

Poisson $$\nabla \cdot \varepsilon_s E = -\varepsilon_s \nabla^2 \psi = q(p - n + N)$$

Electron continuity $$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot J_n + (G - R) \quad \text{where} \quad J_p = q\mu_n E - qD_p \nabla p$$

Hole continuity $$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot J_p + (G - R) \quad \text{where} \quad J_n = q\mu_n E + qD_n \nabla n$$

See Fichtner, Rose, Bank, "Semiconductor Device Simulation", IEEE Trans. Electron Devices 30 (9), pp1018, 1983

- Different versions of physics models available
  - Different models of mobility, bandgap…
  - Generation and recombination rates may include avalanche effects, charge generation by high-energy particles…

Sentaurus Device – Command file

- Controlled by a file *_des.cmd
  - Run with   sdevice whatever_des.cmd
- File specifies the following:
  - ***File*** - Input and output files
  - ***Electrode*** - List of the device's contacts
  - ***Physics*** - Physics models used in simulation -
  - ***Plot*** and ***CurrentPlot*** - Variables in included in output files
  - ***Math*** - Controls for solver
  - ***Solve*** - Simulation conditions

- See example command file
  - SENTAURUS/Sim folders/Seminar/Workbench/StripDetector_des.cmd
  - Simulates a strip detector – IV ramp followed by charge collection sim

Sentaurus Device – Physics

- Basic physics models
  - Mobility – reduced by doping concentration, velocity saturates at high field
  - Recombination – Shockley-Read-Hall: generation and recombination due to defects in midgap
  - EffectiveIntrinsicDensity – models narrowing of bandgap at high doping concentration and high temps

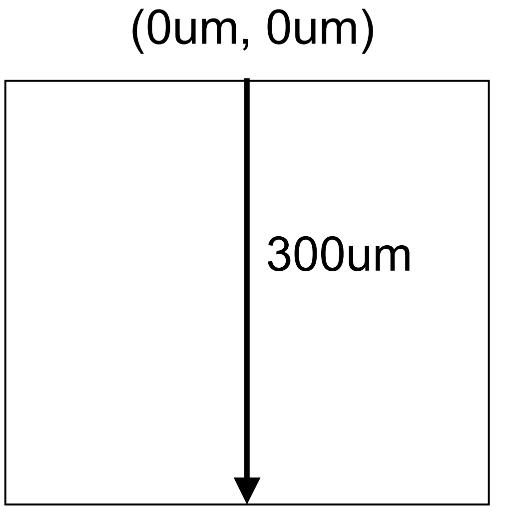
```
Physics {
    # Standard physics models - no radiation damage or avalanche etc.
    Temperature=300
    Mobility( DopingDep HighFieldSaturation Enormal )
    Recombination(SRH(DopingDep))
    EffectiveIntrinsicDensity(Slotboom)
    }
```

- Alternative models for parameters such as mobility, recombination etc. are available – see manual

Sentaurus Device – Useful physics models

- Heavy Ion
  - Flexible model for simulating charge generation produced by particle

```
HeavyIon (
    Direction=(0,1)
    Location=(0,0)
    Time=0.02e-9
    Length = [0 0.001 300 300.001]
    wt_hi = [1.0 1.0 1.0 1.0]
    LET_f = [0 1.282E-5 1.282E-5 0]
    Gaussian
    Picocoulomb )
}
```



  - “Length” is an array. The width of the profile (wt_hi) and the charge generation per unit distance (LET_f) are piecewise-linear
  - In Math section, use RecBoxIntegr command to improve accuracy of charge generation.
    - RecBoxIntegr(5e-3 50 5000)
  - When designing mesh, mesh spacing should be small compared to width of ion track, to ensure accurate generation

Sentaurus Device – Useful physics models

- Avalanche
  - Recombination(SRH(DopingDep) Avalanche(Okuto))
  - Simulates increase in generation from impact ionization
  - Different models available, aside from Okuto – see manual
  - In Plot, eAvalancheGeneration and hAvalancheGeneration
  - If breakdown occurs during an IV ramp, simulation can become very slow: set BreakCriteria in Math section

- Oxide charge
  - Physics models can be specified for particular device regions by inserting a separate physics section:

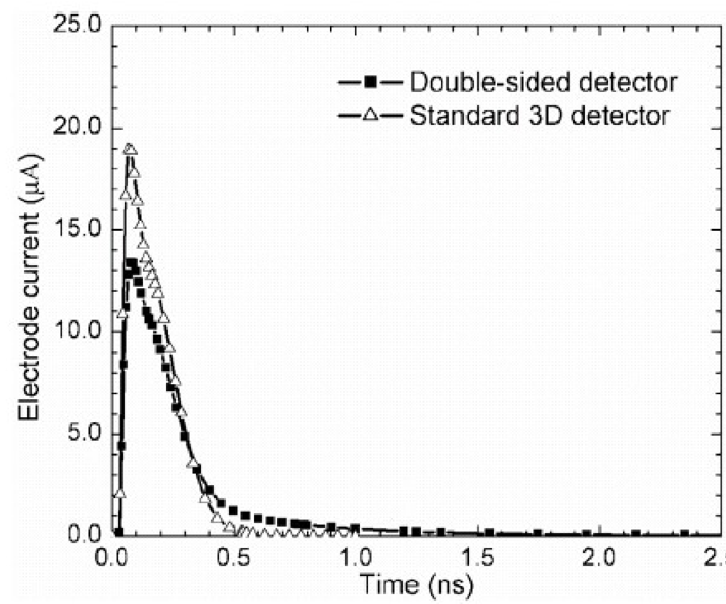
    ```
    Physics(MaterialInterface="Oxide/Silicon") {
    Charge(Conc=4e11)
          }
    ```

  - Oxide charge attracts layer of electrons to interface – need narrower mesh spacing to model this accurately
  - Oxide charge increases after irradiation
- Radiation damage – Talk tomorrow

.plt files and CurrentPlot

- Sentaurus uses the word “plot” far too much – don’t get confused!
- *.plt files
  - Contain the electrode potentials and currents throughout the simulation
  - Can be graphed in Inspect to give IV curves, electrode signals, etc.
- CurrentPlot section allows you to add data to these files
  - Hole density at back surface to test Vdep
  - Max electric field as a rough guide to breakdown

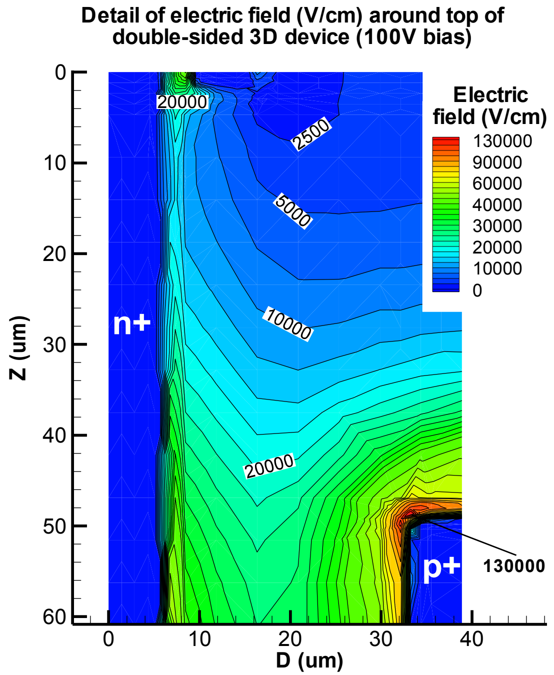
```
CurrentPlot {
    hDensity((25 295))
    ElectricField(Maximum(Material="Silicon"))
}
```



.dat files and Plot

- *.dat files
  - Contain variables such as electric potential and carrier concs. at *every* mesh point in the device
  - Loaded into Tecplot to show electric field distribution, etc
  - One .dat file is produced when sim finishes – commands in the Solve section let you produce more
  - “Plot” section allows you to choose which variables are added
    - See manual – some physics models have particular Plot variables

```
Plot {
    eDensity hDensity eCurrent/Vector hCurrent/Vector
    Potential SpaceCharge ElectricField/Vector Doping
}
```



Math section

- Controls solving the simulation
  - Many useful keywords are now default – not really needed in file!
    - Extrapolate, Derivatives, RelErrControl, NewDiscretization
- Can choose numerical solver
  - “Pardiso” is default, and works well
  - Solvers user guide lists others
- Certain physics models use extra keywords
  - E.g. RecBoxIntegr improves accuracy of charge generation from HeavyIon or optical generation
- Can set break criteria (e.g. to stop simulation if device breaks down)
  - BreakCriteria {Current(Contact=“pplus1" Absval=1e-6)

Solve section

- Various different processes can be done
- Basic solve of Poisson equation, or Poisson Electron Hole
  - Simply solves device under steady bias conditions applied
- Quasistationary
  - Ramps a parameter (usually bias voltage) from one value to another in series of steps
  - At each point, device is solved for a “steady state”
  - E.g. simulating an IV ramp for a photodiode, or response of a transistor
- Transient
  - Simulation over time
  - E.g. signals produced in a radiation detector when hit by a particle
- For both of these, we can control stepping conditions – see file
  - Smaller step sizes
- During the solve, we can produce .dat files (so we can view the state of the simulation at a particular moment)

Solve

```
Solve {
# Get initial state of the device without a bias applied.
Poisson
Coupled{Poisson Electron Hole}

# Ramp-up the voltage to -100V in a series of small steps. While doing this, create data files at 0V, 25V, 50V, 75V, 100V.
# The Quasistationary ramp is controlled by a variable sweeping from 0 to 1. So, the max step corresponds to 0.05*100V = 2.5V.
# As well as "Plot", you can Save and Load the state of the simulation.
Quasistationary (
    InitialStep=1e-3 MaxStep=0.025 Minstep=3e-5 Increment=1.2
    Goal {Voltage=-100 Name=pplus1 }
    )
    {
    Coupled (iterations=8, notdamped=15) {Poisson Electron Hole}
    Plot ( FilePrefix = "StripDetector_" Time = (0; 0.25; 0.5; 0.75; 1) NoOverwrite )
    }
……………
```

Solve

……………………

This statement creates a new current plot file, with its name starting with "transient". This can be useful if you're doing a few different solve phases

```
NewCurrentPrefix = "transient_"
```

Do a simulation over time, to get the current signal produced by the MIP. The "iterations=8" means that if we take more than 8 iterations to solve a step, it'll reduce the step size and try again

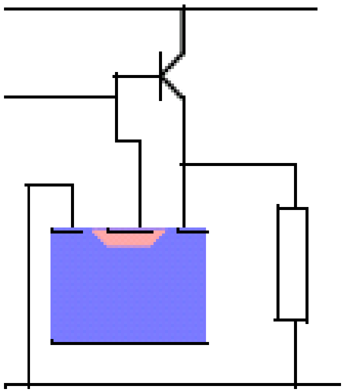
```
Transient(
      InitialTime = 0.0
      FinalTime=40.0e-9
      InitialStep=0.5E-11
      MaxStep=2E-9
      Increment=1.1
      Decrement=1.5
       )
{Coupled (iterations=8, notdamped=15) { Poisson Electron Hole }
Plot (Time = (0.05e-9; 0.2e-9; 1e-9; 5e-9; 10e-9; 20e-9) noOverwrite
FilePrefix="StripDetector_transient")
}

}
```

Solve – Iteration tips

- Sentaurus solves each step by an iterative process
- We set a limit to the no of iterations
  - Success: move on to next step with increased step size
  - Failure – try again with a smaller step
- Generally better to keep number of iterations small (say, 8-10)
  - More accurate, and frequently quicker, to do small steps with few iterations than large steps with many iterations
- “Increment” and “decrement” control changes in step size
  - Default increment is 2 (double step size after success). If sim starts fine, but we get repeated failure later on, useful to reduce Increment
- des.log files record output
  - Typing dessisstat whatever.log will look through a log file and summarise the information

Mixed mode simulation

- In standard simulation, we have a single device with boundary conditions
- Mixed mode simulates one or more devices, plus extra components such as resistors, voltage sources etc., modelled by Spice
  - See Compact Models User guide for details of components
  - E.g. can have AC-coupled detector with strips biased through resistors
- Can do transient simulations with *time-varying* voltage sources
  - E.g switching behaviour of a transistor
  - CCD simulation – use HeavyIon to generate charge within one pixel, then a time-varying voltage to transfer charge to next pixel
- Mixed-mode is needed to do C-V simulation
  - "ACCoupled" command



- Main difference – Sentaurus file has to describe all the devices present, and how they are connected
  - See StripDetector_CV_des.cmd
  - Sentaurus Device manual also has examples

Mixed mode simulation file

```
Device strip {                    # Set up strip detector
      Electrode { ……}
      File {….}
      Physics {…….}
}

File {
Output    = "StripDetector_CV"
ACExtract = "StripDetector_CV"
}

# Describe all the components, and how they connect

System {
strip sample (nplus1=c1 nplus2=c2 nplus3=c3 pplus1=cp)
Vsource_pset vc1 (c1 0) {dc=0}
Vsource_pset vc2 (c2 0) {dc=0}
Vsource_pset vc3 (c3 0) {dc=0}
Vsource_pset vcp (cp 0) {dc=0}
}
```

Mixed mode simulation file

```
Device strip {                    # Set up strip detector
      Electrode { ……}
      File {….}
      Physics {…….}
}

File {
Output    = "StripDetector_CV"
ACExtract = "StripDetector_CV"
}

# Describe all the components, and how they connect

System {
strip sample (nplus1=c1 nplus2=c2 nplus3=c3 pplus1=cp)
Vsource_pset vc1 (c1 0) {dc=0}
Vsource_pset vc2 (c2 0) {dc=0}
Vsource_pset vc3 (c3 0) {dc=0}
Vsource_pset vcp (cp 0) {dc=0}
}
```

Mixed mode – typical CV commands

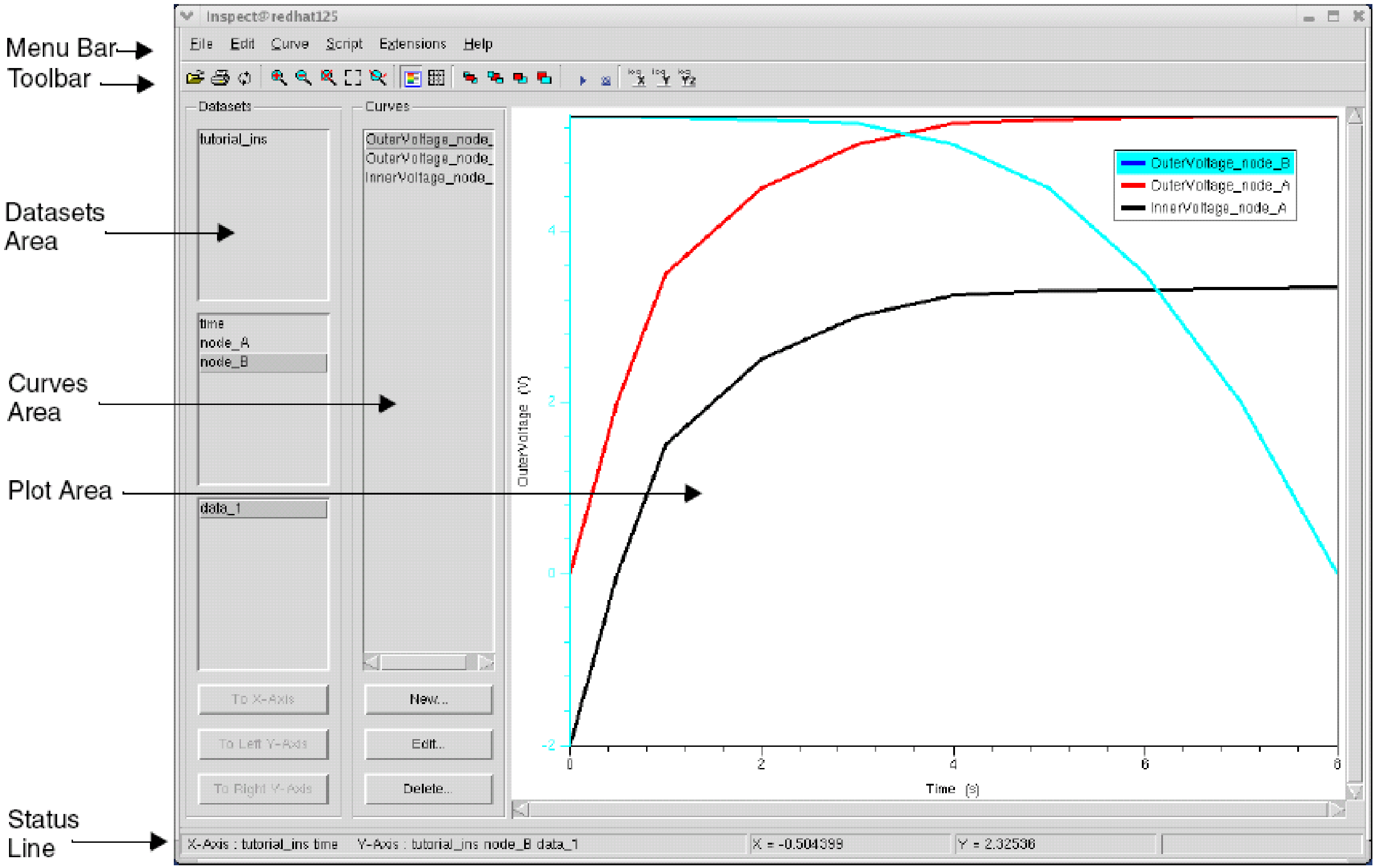
```
Solve {
    # Get initial state of the device without a bias applied.
    .................

    # Then, use a combination of Quasistationary and ACCoupled to do CV
sim while ramping bias to -100V
    Quasistationary (
    InitialStep=1e-3 MaxStep=0.025 Minstep=3e-5 Increment=1.2
    Goal { Parameter=vcp.dc Voltage=-100 }
    )
    { ACCoupled (
    Iterations=10
    StartFrequency=1e4 EndFrequency=1e4
    NumberOfPoints=1 Decade
    # Specify which nodes we look at AC behaviour between. Exclude all
voltage sources
    Node(c1 c2 c3 cp) Exclude(vc1 vc2 vc3 vcp)
    )
    { Poisson Electron Hole }
    }
}
```

Overview

- Introduction to Sentaurus TCAD software
- Building the device structure
- Running the simulation
- Viewing results
- Other software

Inspect

- Creates graphs from *.plt files created by Sentaurus Device
  - Contains electrode voltages, currents etc, and “time”
  - Contains data produced by CurrentPlot (e.g. max electric field)
  - Can graph any pair of data sets



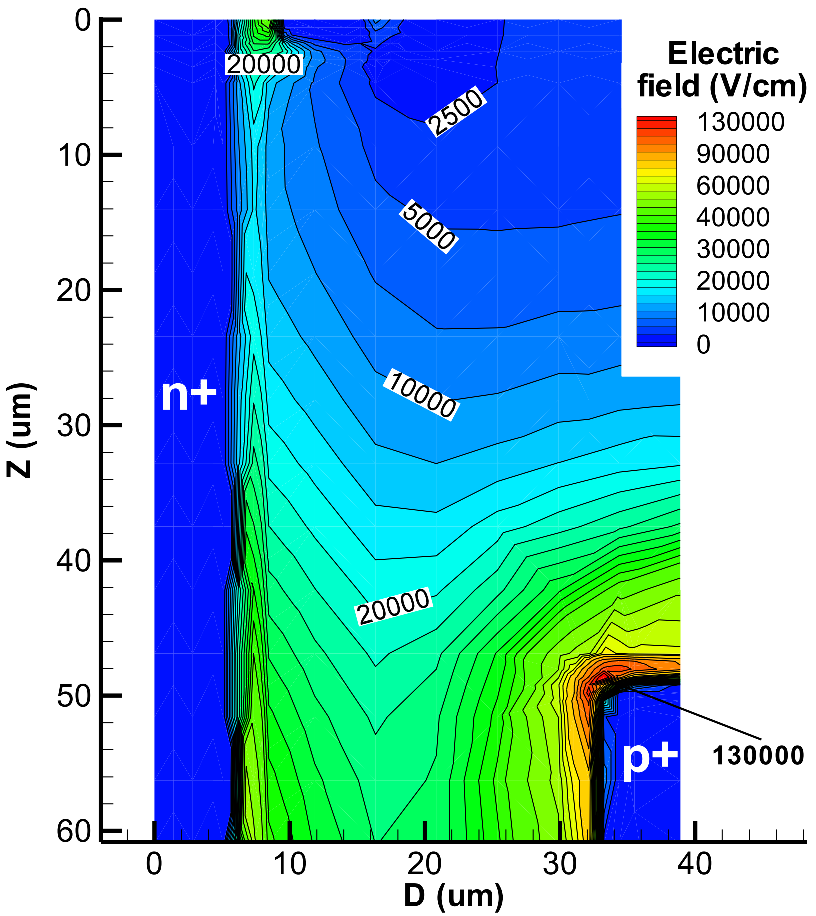
Inspect

- After creating a curve, can use File->Export->XGraph to export a file with x,y data points
  - Can then be used in other programs like Origin
- Can do mathematics on graphs using “new” button
  - integr(<currentgraph>) to integrate a current pulse over time
- Scripting language is available to control Inspect
  - *ins.cmd script file can be loaded
  - See Inspect manual for language
  - Scripts->Record allows you to carry out a series of steps by hand, and writes the corresponding commands to a file
- Scripting language allows you to extract data from projects when using Workbench

Tecplot SV

- Can load *.grd and *.dat files
  - View meshes
  - View field distributions etc produced by Sentaurus device
  - Contour plots, vectors…
- tecplot_sv command
- Tecplot is a general-purpose package for viewing 2D and 3D plots
  - http://www.tecplot.com
  - Synopsys then added on extra functions
- Manuals
  - Tecplot SV user guide only describes extra functions (58 pages)
  - Tecplot User Manual describes regular version of package (632 pages)
  - Tecplot Reference Manual gives script commands (320 pages)

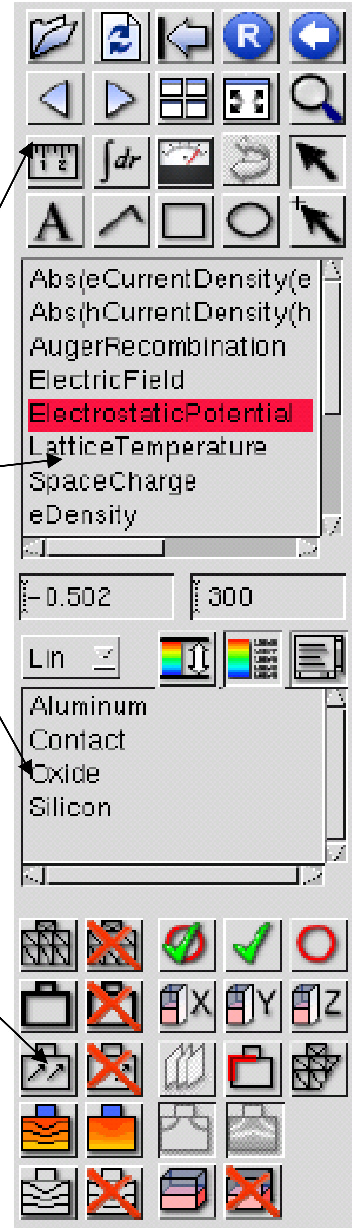
Detail of electric field (V/cm) around top of double-sided 3D device (100V bias)



Tecplot SV

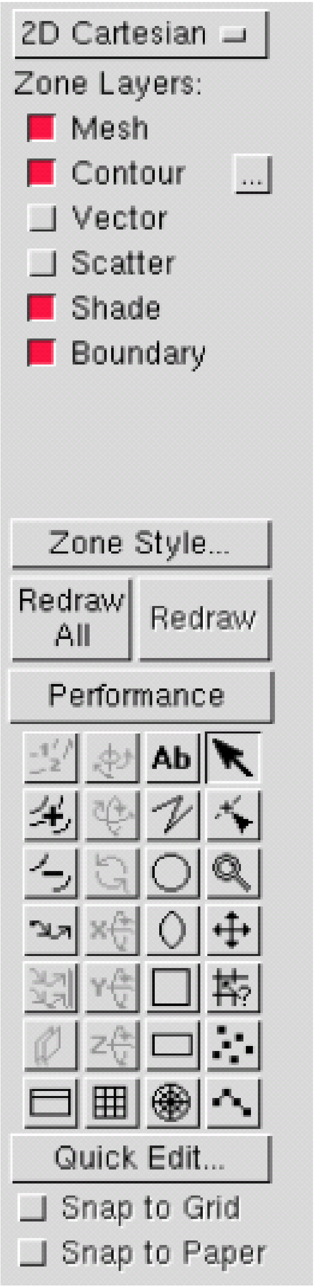
- Two *different* sidebars are available: Synopsys and Tecplot
  - Switch using the View menu at the top
- Synopsys sidebar - useful basic commands
  - Basic commands for loading files, zooming, inserting text/lines etc
  - List of variables to plot, and ranges to set
    - Can also add legend etc
  - Menu to choose which material we apply commands to
  - Options to make the following visible/invisible
    - Mesh structure
    - Boundary
    - Vector plots (for quantities like E-field)
    - Contour lines, flood
  - (On right) options to make 2D cuts from 3D device simulations
  - (On right) display or hide p-n junctions and depletion regions

2D/3D Mode



Tecplot SV

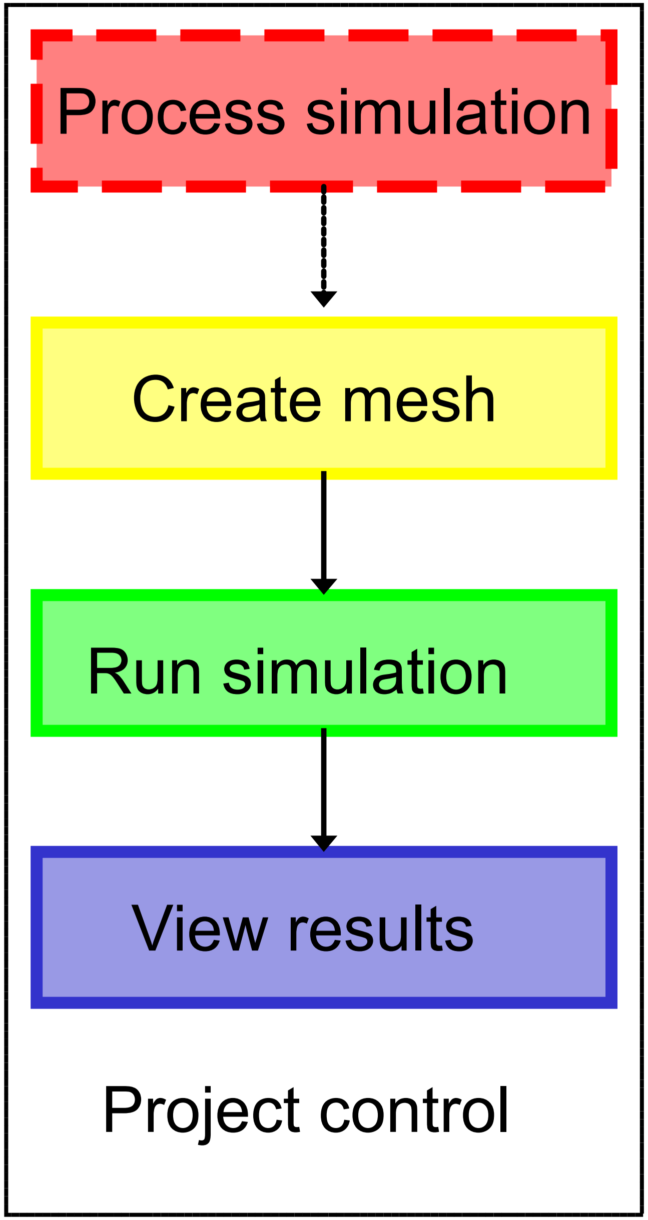
- Tecplot Sidebar
  - Option to switch between 2D and 3D!
  - "Zone style" gives lots of options for controlling contours, vectors etc.
  - Better tools for:
    - Controlling contour lines and streamtraces
    - Moving and rotating 3D objects
- Menus at top of screen – many, many options!
  - Same features as sidebars, but often more control
    - E.g. taking 2D slices at unusual angles
  - Saving states and images
    - Saving a "layout" file saves state of simulation
    - Saving a "style" file just saves formatting – can load another data set then the style file to apply formatting
  - Recording and using macros
  - Creating new variables using existing ones
    - E.g can find the components of a vector in a particular plane



Overview

- Introduction to Sentaurus TCAD software
- Building the device structure
- Running the simulation
- Viewing results
- Other software

Simulation packages



Sentaurus Process (optional) {dios}

Ligament can generate command files for Process

Mesh, noffset3d create device meshes using a command file

Sentaurus Structure Editor (and MDraw) create meshes with GUI

Sentaurus Device {dessis}

Inspect – Plotting graphs of electrode currents etc.

Tecplot – Producing images of electric field patterns etc. throughout device

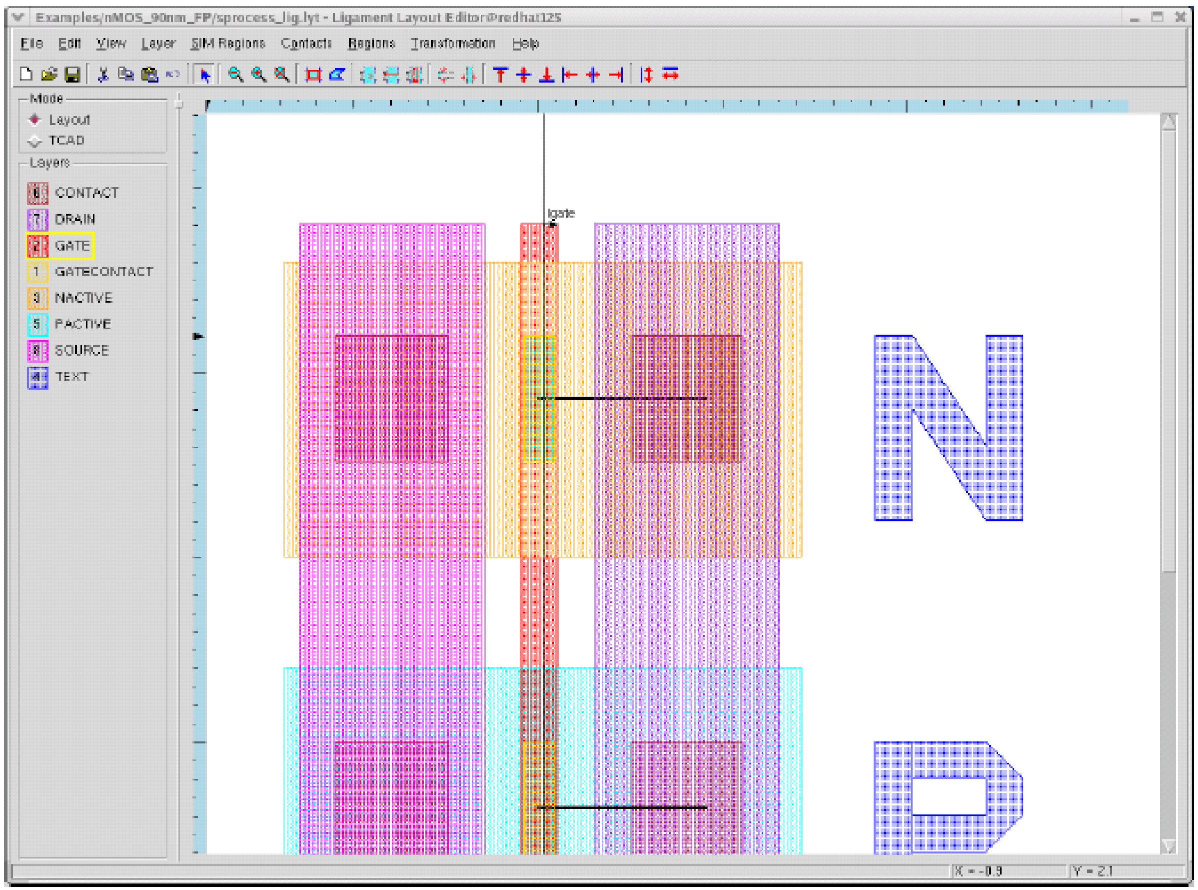
Workbench – Can run large numbers of simulations conveniently

Sentaurus Process

- Process simulation
- Can be controlled interactively (series of commands) or with command file
- Simple case – 1D simulation
  - Start with, say, a doping implant
  - Describe a series of processing steps (time, temp)
  - Simulation finds the 1D doping profile after the stages
  - Produces a .plx doping file, which can then be used in mesh
- 2D and 3D simulation
  - Once again, need to describe a series of processes (implants, annealing, etc.)
  - In this case, also have to define masks used
  - Also, simulation is carried out on a grid of points – similar considerations to refinements in mesh design
  - Finally, automatically produces input files for mesh

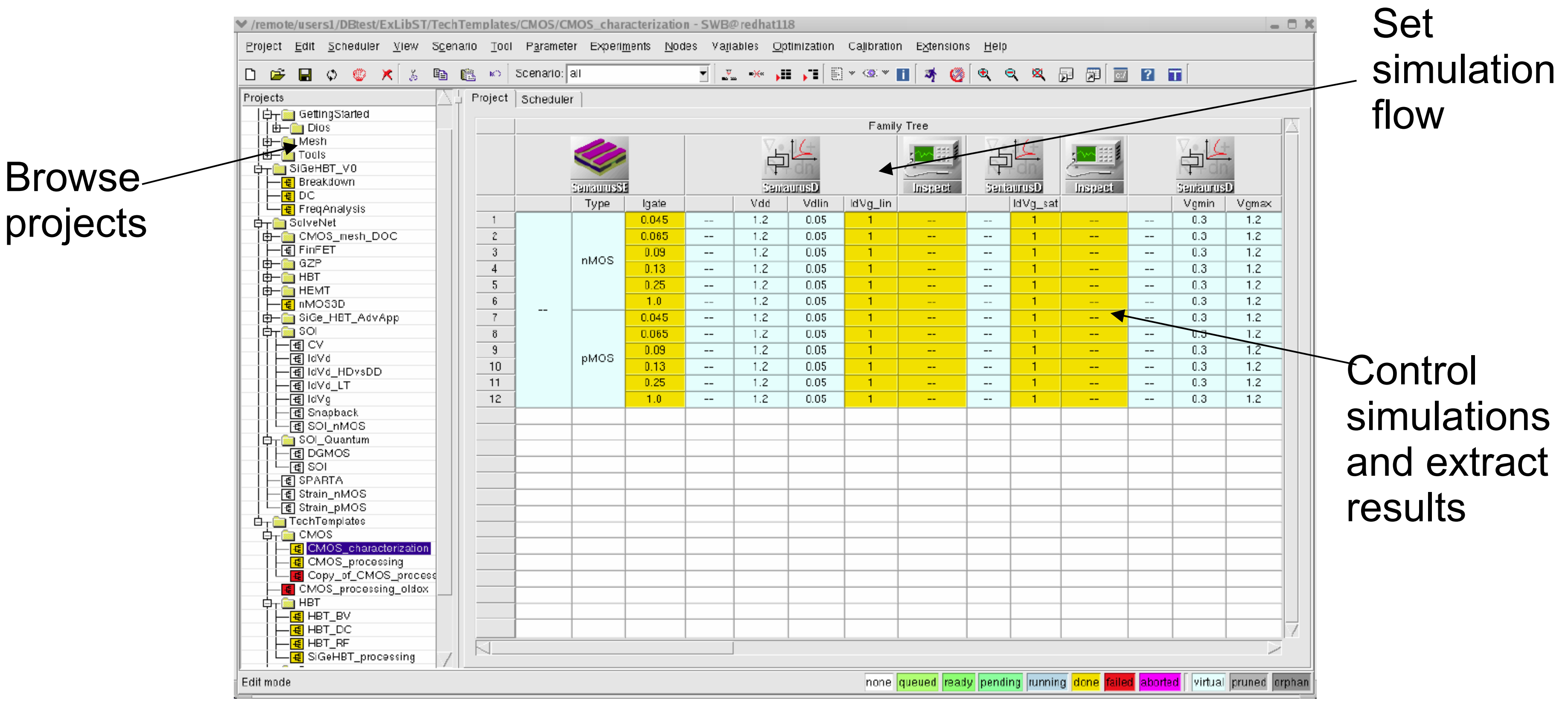
Ligament

- Graphical interface to produce command files for Process
- Flow editor – create a list of process steps
- Ligament layout editor – MDraw-like package for defining "masks" used for each process step



Sentaurus Workbench

- Project control tool
- Allows you to repeatedly run multi-stage simulations using different parameters each time
  - This includes producing graphs of results or extracting data like CCE
- Tutorial next!



Other information

- This is part of a Sentaurus TCAD seminar:
  - http://ppewww.physics.gla.ac.uk/det_dev/activities/threedee/Documents/BarcelonaSeminar.html
  - The page includes a .tgz file with the simulation files referred to in this talk
  - Basic strip detector simulation – folder SENTAURUS/Seminar/Introduction
    - StripDetector.bnd/cmd files for mesh
    - StripDetector_des.cmd file for Sentaurus Device
    - The simulation has already been run, and output files are in the same folder
  - CV simulation
    - Uses the same mesh as before
    - StripDetector_CV_des.cmd

- Synopsys website –
  - https://solvnet.synopsys.com/amserver/UI/Login
  - Can sign up to get a login ID. Can get all the manuals, plus some simulation examples
- Sentaurus Device simulation examples
  - $STROOT/tcad/$STRELEASE/lib/sdevice/GettingStarted
  - Files corresponding to examples in SDevice manual