Links  
Online Sentaurus help: <http://www.sentaurus.dsod.pl/index.html>

Notes

4.26.2018

* UCSD Resources for using Putty & VNC

<https://crl.ucsd.edu/handbook/vnc/>

Our Sentaurus server password: Che8ron-

* Access Sentaurus through Putty and VNC or Xming.
* Elastochemical potential term to add to the diffusion equation. This additional term is expected to be negligible compared to drift and diffusion.

5.23.2018

* PVRD1: Device simulations will be needed later. Work currently done on process simulations (Na+ diffusion, elastochemical potential, etc.), later will need to work on device. Full, different architectures to simulate → 6 to 9 months from now.

|  |  |
| --- | --- |
| Sentaurus process | Sentaurus device |
| Na+ diffusion, Na+ concentration as a function of time, elastochemical potential | Lifetime, bulk recombination as a function of time.  Model full solar cells and obtain efficiency as a function of Na+ concentration. |

* Idea of PVRD1: ASU produces data, we extract from this data: segregation, diffusion coefficient.

7.23.2018

Sentaurus setup

* UCSD Resources for using Putty & VNC: <https://crl.ucsd.edu/handbook/vnc/>
* Need xming or vnc viewer. Putty help accessing unix.
* If using xming, just start it before login in to Putty. Type swb & to access sentaurus workbench and sde to access sentaurus structure editor.
* If using vnc viewer, first log in to Putty and type vncserver. Log. Then go to vnc viewer and log in to localhost0.
* To kill VNC viewer: see <https://acms.ucsd.edu/info/vncgnome.html>

Logging off

Once you are finished, you need to kill the GNOME session. Please run:

vncserver -kill X-display-#

You will be notified what your X-display is when you first start your VNC process. Once you run vncserver you will see output similar to the following:

New 'ieng6-244.ucsd.edu:24 (okenobi)' desktop is ieng6-244.ucsd.edu:24

Please take note of the bolded section, as you will need this to cancel your job once you are finished. In this example, we would kill our VNC process with:

vncserver -kill :24

Long-running jobs

Any processes older than 2 days with a connection to software (e.g. ADS, Cadence) will be killed. Be sure to save your work and exit the program when you're finished.

To find open vncserver connections: <https://superuser.com/questions/549386/what-is-the-correct-way-to-kill-a-vncsession-in-linux>

Command: **ps -ef |grep vnc** and find line with Xvnc :m where m is the session to kill.

* Our Sentaurus server password: Che8ron-

Hostname to use: [na299x@ieng6-242.ucsd.edu](mailto:na299x@ieng6-242.ucsd.edu)

* Go to DB folder in winscp
* Go back to Putty and run Sentaurus workbench by writing “swb &” in the command line.

In vnc server: localhost:0

* Google “UCSD vnc” or visit <https://acms.ucsd.edu/info/vncgnome.html> to find a list of accessible servers.
* Type vncserver in the command line.
* Once in sentaurus workbench, can create new projects (can choose between Sentaurus device and Sentaurus process).
* See help in the workbench → documentation is there
* Kill VNC server: vncserver -kill :58, where 58 is the number of the session in the workbench (found at the top of the window in the workbenck). **Note the space after “kill”!** See instructions at <https://acms.ucsd.edu/info/vncgnome.html>.
* If using xming instead of VNC server, no need to kill the session.

Run Sentaurus from the command line:

* Command to run Sentaurus Device from the command line: sdevice. See Sentaurus Device manual p. 1363: Command-Line Options.
* Command to run Sentaurus Structure Editor from the command line: sde. See Structure Editor manual p. 5: Command-Line options.
* Option -e is used to run in batch mode, meaning without the GUI. Option -e -l is used to run a script in batch mode. For instance: sde -e -l.
* Gtree is the simulation workflow: tells what Sentaurus is going to run
* See example file in erick’s folder 2d\_cell.
* Different nodes. To view results, right click on a node in a SDEVICE column in the Sentaurus workbench, and select “Quick vizualize”. Opens the plots, from there can select parameter to plot and export it.
* To use matlab in the linux server: type “matlab -nodesktop” in the Putty command line.
* Look first at structure editor, then look at device example.

Note 1/13/2020:

Problems with killing a server.

Can't find file /home/linux/ieng6/na299x/na299x/.vnc/ieng6-242.ucsd.edu:14.pid

You'll have to kill the Xvnc process manually

<https://stackoverflow.com/questions/28755379/getting-error-with-vnc-session-is-already-running>

<https://lists.gt.net/vnc/list/58565>

Solution found at: <https://crl.ucsd.edu/handbook/vnc/index.php>

→ type in the PuTTY command line “**ps -ef |grep vnc**” and find line with Xvnc :## where ## is the session to kill.  
The Processor ID (PID) number is at the beginning of the line. For instance in the following example, the server window is 2 and the PID number is 6788:

ee15wi2+ 6788 1 0 Jan08 ? 00:02:47 /software/common/bin/Xvnc :2 -desktop ieng6-242.ucsd.edu:2 (ee15wi20cv) -auth /home/linux/ieng6/ee15wi20/ee15wi20cv/.Xauthority -geometry 1024x768 -rfbwait 30000 -rfbauth /home/linux/ieng6/ee15wi20/ee15wi20cv/.vnc/passwd -rfbport 64037 -fp catalogue:/etc/X11/fontpath.d -pn

To kill this process, type “kill 6788” in this case, where 6788 is the PID number.

8.14.2018

Sentaurus:

* Sentaurus workbench tutorial available at <file:///C:/Users/Guillaume/Documents/%23UCSD/Sentaurus/Jonathan_documents/Sentaurus_Training/swb/swb_b.html> or from Sentaurus help (or copy and paste the files from WinSCP).
* To open the project “SimpleMOS”, go to ieng6/na299x/na299x/DB/GettingStarted/swb.
* One parameter corresponds to one node. A complete set of nodes constitute and experiment. Nodes can be selected independently is only a specific number of parameters need to be simulated.

8.15.2018

* To run Sentaurus:

Define the device’s geometry in Sentaurus structure editor (shortcut sde). The user-defined file is a scheme script file (.scm). Other input files are needed too (see <file:///C:/Users/Guillaume/Documents/%23UCSD/Sentaurus/Jonathan_documents/Sentaurus_Training/sse/sse_a.html>). See also introduction\_to\_Sentaurus\_TCAD by David Pennicard, university of Glasgow.

Input for Sentaurs process: fps.cmd.

The input files for Sentaurus Structure Editor are named \*\_dvs.cmd. See Beneventi tutorial.

Then run Sentaurus device for the physics part (\*\_des.cmd). The input files for Sentaurus device are the command file (\*\_des.cmd), the parameter file (\*\_desSi.par), and the device structure file (\*\_msh.tdr). The number of input files can vary, Sentaurus expects at least one to define device structure and field values. The .tdr file is optional and is used to declare material properties and physical model parameters. See Sentaurus help, Sentaurus Device → 1. Basics.

Example of a scheme command: (sdegeo:create-cuboid (position 0 0 0) (position 1 1 1) "Silicon" "region\_1") ; To create a cube

* See Sentaurus help file, Sentaurus Structure Editor → 5. Scripting and Parameterization.
* To load Sentaurus structure editor file in the command line: (sde -h -e -l test\_sde\_commandline.scm). See Sentaurus structure editor user manual, chapter 2, p. 5.
* The @ sign is used for a parameter: @parameter@ in the code.
* Next: implement the diode example by Giovanni Betti Beneventi (sde file, sdevice file) and try to make it work. See .pdf file “04\_TCAD\_laboratory\_pn\_junction…”
* Next: Do it for a MOS and plot band diagrams.
* Next: Do it for an Al-BSF solar cell.
* Sentaurus: see Beneventi PN junction examples and see also the solar cell back contact example
* See also youtube video (saved in favorites) about solar cell simulation.

11.29.2018

* PVRD1, Sentaurus:

“sdegeo” used to create a geometrical region in Sentaurus editor

“sdedr” usd to create a doping region

* Contact definition (see chap. 7 Sentaurus device editor):
* First declare and activate the contact
* Then assign an edge or a face to a contact
* Contacts  
  Three different commands to be used successively to build contacts:
* (sdegeo:define-contact-set contact-name edge-thickness (color:rgb red green blue) pattern) **; Definition of contact**
* (sdegeo:set-current-contact-set contact-name) **; Activation of contact**
* (sdegeo:define-2d-contact edge|edge-list contact-name) **; Assignment of contact**
* Example of contact building:
* (sdegeo:create-rectangle (position 0.0 0.0 0) (position 1.0 1.0 0) "Silicon" "region\_1")
* (sdegeo:define-contact-set "cont" 4 (color:rgb 1 0 0) "##") **; Definition of contact**
* (sdegeo:set-current-contact-set "cont") **; Activation of contact**
* (sdegeo:define-2d-contact (find-edge-id (position 0.5 0.0 0)) **; Assignment of contact**
* (sdegeo:get-current-contact-set) **; Reference the contact automatically**
* To add contacts in only one part of a device, first define two vertices and then use find-edge-id to find this vertices and place the contact at the adequate position. See Sentaurus structure editor manual p. 189 (”creating new edges or faces for use in contacts”).
* Definition of doping (for a constant profile) are done in three steps:

1. Define a ref/eval windows OR a region OR a material (see detail in point 3)
2. Use the command “sdedr:define-constant-profile”. Then define the name of the doping profile (“p-doping-profile”), the doping element to choose from a set of predefined dopants including boron, phosphorus, arsenic, etc (“BoronActiveConcentration”) and the doping density

(sdedr:define-constant-profile "p-doping-profile" "BoronActiveConcentration" @p\_doping@)

1. Then place the profile in the desired location. There are three ways to place a doping profile (Sentaurus structure editor p. 219, see also example p. 223):

* By Ref/Eval window. Command (sdedr:define-constant-profile-placement placement-name definition-name ref-win-name).  
  Note that the window must have been defined previously in the code.
* By region, using a region previously defined in the code. Command (sdedr:define-constant-profile-region placement-name definition-name region-name).
* By material, using the materials previously defined the code. Command define-constant-profile-material.

In the PN junction examples by Beneventi:

Set this doping profile in the p-type region “p-region” defined previously in the dvs file

(sdedr:define-constant-profile-region "p-doping-placement" "p-doping-profile" "p-region") **; place the doping profile in the region p-region**

(sdedr:define-constant-profile-placement "p-doping-placement" "p-doping-profile" "p-doping-window") **; use window placement, not sure why the doped region is defined twice in this example. (related to mesh refinement definition, see Sentaurus structure manual p. 204).**

Edit 1/28/2020: it seems that the sdedr:define-constant-profile-placement command above is useless, as the p-doping-window has not been defined previously in the code. The sdedr:define-constant-profile-region command should be enough to place the profile.

* For other profiles, for instance a Gaussian profile, this would be a bit different and the command to use would be sdedr:define-gaussian-profile.

12.5.2018

* Mesh refinement:
* First define a window for refinement placement: (sdedr:define-refeval-window ref-eval-window { single-shape | multi-shape }). Ref-eval-window is the name given to this window, then come arguments to define the shape and location of the window.
* Then define the size of the refinement (sdedr:define-refinement-size definition-name max-x max-y max-z min-x min-y min-z) where max-x, max-y etc are the mesh dimensions and “definition-name” is the name given to the refinement.
* Then, place the refined region: (sdedr:define-refinement-placement refinement-name definition-name ref-eval-window)where “refinement-name” is the name given to this refinement, “definition-name” is the previously defined refinement size, and “ref-eval-window” is the name of the previously defined window for refinement placement.
* Mesh refinement in the Beneventi PN-junction example:
* Define ref/eval window of the size of the whole device for refinement placement: (sdedr:define-refeval-window "domain-ref" "Rectangle" (position 0 0 0) (position L (+ Wn Wp) 0))
* Size definition:(sdedr:define-refinement-size "domain-ref-size" xmax ymax xmin ymin)
* Placement: (sdedr:define-refinement-placement "domain-ref-pl" "domain-ref-size" "domain-ref")
* Sentaurus output files:
* .plt contains the electrical characteristics of the electrodes (currents, voltages, charges)
* .tdr contains the distributed variables (doping profiles, charge, field, carrier densities, etc)
* Visualize data stored in .tdr file: svisual n1\_des.tdr &

1.10.2019

PVRD1: Sentaurus

* See p. 663 of Sentaurus Device manual for explanation of naming of the .tdr files.  
  Each time a solution is plotted, a tdr file such as n2\_000001\_des.tdr in Beneventi’s examples is created. In Beneventi’s example “plot { range=(0, 1) intervals=1 }”, the interval chosen for plotting is 1 and will determine the name of the created tdr file.

1.17.2019

* Arithmetic operations in Sentaurus, addition and subtraction. See structure editor manual p. 273. Sentaurus device editor uses the Polish notation.

2.11.2019

* Sentaurus: assign contacts. See examples in sdevice manual p. 191.

2.13.2019

* Sentaurus memory issue. Maryam gave the command “du -a . | sort -n -r |head -n 10” to find the 10 files occupying the most memory on linux. Note that the results are given in kB.

Try on 2/13/2019:

du -a . | sort -n -r |head -n 10

<https://www.howtogeek.com/50714/list-the-10-largest-files-or-directories-on-linux/>

<http://www.labtestproject.com/linuxcmd/du_command.html>

4483112 .

4300236 ./DB

2798324 ./DB/Jonathan

2798320 ./DB/Jonathan/Matab

2181844 ./DB/Jonathan/Matab/BreakthroughTime

1461416 ./DB/Jonathan/Matab/BreakthroughTime/EVA\_ExpTesting

1385652 ./DB/Jonathan/Matab/BreakthroughTime/EVA\_ExpTesting/BreakthroughData

808032 ./DB/Guillaume

691124 ./DB/Guillaume/Matlab

691120 ./DB/Guillaume/Matlab/GD\_Fit\_Vfb

* Results in kB. For instance, “DB/Guillaume” is 808 MB.
* Entity types in Sentaurus: body, face, edge, vertex, other (Sentaurus device editor page 486).
* Order box for CV setup circuit from Hawk Electronics inc. Order number 25025.

3.1.2019

* Ramping boundary conditions, for instance when ramping voltage from -1 V to +1.5 V (InitialStep, MaxStep, MinStep): see Sentaurus Device manual p. 119. The initial step is the step Sentaurus uses at the beginning to ramp voltage. Then, if convergence is not reached, it can change the step automatically. The max and min steps are set by the user to control the limit of the steps used by Sentaurus. If the step becomes smaller than MinStep, Sentaurus will stop and give an error.
* Saving and plotting during a quasistationary: Sdevice user manual p. 125.
* Language to set in notepad++ (for ; as comments): LISP. See https://udl20.weebly.com/delimiters.html.
* For illumination on solar cell: see example file SolarCell\_SingleJunct\_GaAs in Sentaurus examples.

3.2.2019

* Illumination window: see Sentaurus Device manual p. 603. In 2D, Y-position does not need to be defined. The origin of the illumination window can be defined by Origin = (x0, y0). See manual p. 605.
* Explanation of LayerStackExtraction in Sentaurus Device manual p. 672. Related to the single junction GaAs solar cell in the Sentaurus example, p. 11.
* Meshing for the TMM solver in the Single junction GaAs solar cell. The mesh is defined in the csv file with $global dYmax=0.2. Local mesh refinements are defined using xref, yref and zref, which are used for constant refinements in the xyz direction (Sentaurus Structure Editor manual p. 261). Graded refinement (dense at interface and relaxing towards bulk) are realized using the command mbox. See Sentaurus mesh user guide.  
  In the single junction example, mesh refinements are of 5-10 nm in each layer (and gradually increasing to 1.1 um towards the bulk of thick regions).
* See StandingWave versus Envelope options for TMM solving in surface vs bulk regions.
* Q: Is it needed to implement the TMM method to take into account interferences in the SiNx layer or is it done by default by Sentaurus?? Looks like the TMM method needs to be used.

See “Using the Transfer Matrix Method” p. 671 of sdevice manual.  
See extracting the layer stack p. 612 of sdevice manual.

* Next: need to change meshing to allow for standingwave TMM solving (need to decrease meshing size in SiNx and near-Si surface). Also need to specify the regions in which standingwave vs Envelope is used.

From the example “SolarCell\_SingleJunct\_GaAs”:

Physics (Region="base") {

\* for thick regions avoid interpolation errors due to non resolved standing wave patterns

Optics(OpticalSolver(TMM(IntensityPattern = Envelope)))

}

3.4.2019

* Created a new language template in Notepad++ to highlight sdevice code. Called “sdevice”. For structure editor code (\_dvs.cm), use LISP predefined language.
* See Sentaurus Mesh user manual p. 75 to perform interface meshing refinement. Note that the functions presented in this manual are meant to be used in the SentaurusMesh software, not in Sentaurus structure editor or in Sentaurus Device. They recommend not to use mbox (multibox) anymore.
* Scheme “sdedr:define-refinement-size”: see structure editor manual p. 568. Refers to Mesh editor user manual p. 12. The manual only explains that MaxElementSize and MinElementSize refer to the max and min sizes of the grid elements. It seems that Sentaurus automatically optimizes the mesh based on these values.
* See sdevice manual region-specific models.

Physics (region="region-name") {

<physics-body>

}

It looks like there is no logical “and” that can be used with this command. For instance, if the same model needs to be applied to two different material regions:

Physics (MaterialInterface= "GaAs/GaAs") {

Recombination( surfaceSRH )

}

Physics (MaterialInterface= "GaAs/AlGaAs") {

Recombination( surfaceSRH )

}

* Duplicate vertices and elements are vertices or elements which occur more than once in the mesh. <http://webcache.googleusercontent.com/search?q=cache:uQCDNvZPY7QJ:www.iue.tuwien.ac.at/pdf/ib_2015/hashed_links/p54PCkr3wuWnsrCY_us.pdf+&cd=6&hl=en&ct=clnk&gl=us>
* Sentaurus didn’t like when the refined meshing window was defined starting at a location where a vertex was (end of the metal contact). Had to start from x=0 instead.
* Error when using origin=() in excitation.

6.9.2019

* As in the single junction GaAs file, the illumination file must be specified at the beginning of the sdevice file. It is currently not in the Al-BSF file.  
  The syntax is the following:

IlluminationSpectrum= "[spectrumpath.txt]".

IlluminationSpectrum is the name of the Sentaurus command used to assign the path of the spectrum file.

* The bug with the illumination included is that @ are missing around the parameters defined in the structure file. Try running again with the @ added.

Hi,

Could you indicate where I can find the most recent Sentaurus user manuals, in particular for Sentaurus Device, as well as example code?

I am currently using Sentaurus Device 2016 user manual but it seems some features have changed. I tried defining the transfer matrix method in the optical solver section and received the following message "Error: The following feature has been discontinued: Specification of TMM solver in Optics section. Please use syntax of unified interface for optical generation computation to specify TMM solver !".

It would be useful if you could share information on how to use this unified interface.

Best regards,

Guillaume von Gastrow

7/27/2019

* Path to Sentaurus directory: /home/linux/ieng6/na299x/na299x/DB/Guillaume

**Notes to add an external optical generation profile**

* To input external optical generation profile, see example of cSi\_BackContOPt and Steven Ning’s Master’s Thesis p. 81-86.

Optics(

OpticalGeneration(

ReadFromFile(

TimeDependence(

WaveTime= (1, 2)

WaveTSlope= 0.05

)

)

)

)

* Copied the file “input\_optical\_generation.plx” into the Al\_BSF folder. Need to update the code and try running it.$
* <http://nanohub.org/> for resources.
* Example documentation p.6: “Both optical and doping profiles will be stored in a \*\_msh.tdr file as part of the Sentaurus Structure Editor output”. This is done by creating a one-dimensional external profile in the mesh command file using the command sdedr:define-1d-external-profile.
* Optical generation: see p. 81-87 in Steven ning’s Master’s Thesis.
* Lambda command is used to create local variables. See Sentaurus Structure Editor guide p. 276.
* Summary to add an external optical generation profile:

1. Import the profile

Place the profile in the device:

1. Define a refinement window (finer mesh for calculation of generation)
2. Create a 1D external profile in the mesh command file
3. Place the profile in the correct window (defined above) in the mesh command file

Note: Loop/if condition creation. To create an if-else block (Sentaurus device editor p. 275):

(define val -1)

(if (< val 0) ; Test

(begin ; Execute if condition is true

(display "val is negative") (newline)

)

(begin ; Execute if condition is false

(display "val is positive") (newline)

)

) ;-> "val is negative"

* Explanation of the cSi\_BackContOpt example for external optical generation positioning:

; 1) Define an optical generation profile in the mesh file based on variables win\_name, place\_name, Xll, Yll, Xur, Yur which are window name, window place name, and window positioning parameters. It can be reused later by calling PlaceOpt with the correct variables, for instance (PlaceOpt "opt\_win" "opt\_place" HalfFrontCntSize 0 CellW CellL). This step is similar to a function definition.

(define PlaceOpt

(lambda (win\_name place\_name Xll Yll Xur Yur)

; Xll/Yll (lower-left), Xur/Yur (upper-right), Ext\_xy (extension of the refinement window along X/Y direction)

(begin

(sdedr:define-refinement-window win\_name "Rectangle" (position Xll Yll 0) (position Xur Yur 0))

(sdedr:define-1d-external-profile "1d\_opt\_def2" CntFile "Scale" 1.0 "Range" 0 1000 "Erf" "Factor" 0)

(sdedr:define-analytical-profile-placement place\_name "1d\_opt\_def2" win\_name "Positive" "Replace" "Eval")

)

)

)

; 2) Define a global optical generation window

(sdedr:define-refinement-window "globalopt\_window" "Rectangle" (position HalfFrontCntSize 0 0)

(position CellW CellL 0))

(sdedr:define-1d-external-profile "1d\_opt\_def" noCntFile "Scale" 1.0 "Range" 0 1000 "Erf" "Factor" 0)

(sdedr:define-analytical-profile-placement "1d\_opt\_place" "1d\_opt\_def" "globalopt\_window" "Positive" "NoReplace" "Eval")

; 3) Define the optical generation window depending on whether whole back contact (FlagAllCnt=1) or point back contacts (FlagAllCnt=0) are used.

(if (= FlagAllCnt 1)

(begin ;If the condition is true, just place the optical generation ;window starting from the halfcontact size and ending at the back of ;the cell

(PlaceOpt "opt\_win" "opt\_place" HalfFrontCntSize 0 CellW CellL)

)

(begin ;Else if the condition is false, place the optical generation window at the right locations depending on the locations of the point contacts

(define optWinName "opt\_win\_")

(define optPlaceName "opt\_place\_")

(define optWinPrefix "opt\_win\_")

(define optPlacePrefix "opt\_place\_")

(define totalCnt 0)

(set! restSpaceX (- CellW ShiftX))

(set! currentLocX ShiftX)

(define Xstart currentLocX)

(do ((x 0 (+ x 1))) ; do-loop for X direction (there may be several point contacts on the back)

( (= x numCnt\_x) )

(begin

(set! optWinName (string-append optWinPrefix (number->string totalCnt)))

(set! optPlaceName (string-append optPlacePrefix (number->string totalCnt)))

(cond

( (and (= x 0) (= ShiftX 0) ) ;; first condition: if it's the first half-contact

(begin ; only room to place half contact in X direction

(if (< HalfFrontCntSize (+ currentLocX HalfRearCntSize))

(begin

(set! Xstart (max currentLocX HalfFrontCntSize))

(PlaceOpt optWinName optPlaceName Xstart currentLocY

(+ currentLocX HalfRearCntSize) (+ currentLocY HalfRearCntSize))

)

)

(set! totalCnt (+ totalCnt 1))

(set! currentLocX (+ currentLocX (- RearCntPitch HalfRearCntSize)))

(set! restSpaceX (- restSpaceX (- RearCntPitch HalfRearCntSize)))

)

) ; end of first condition

( (= restSpaceX HalfRearCntSize) ;; second condition: if it's the last half-contact

(begin ; only room to place half contact in X direction

(PlaceOpt optWinName optPlaceName currentLocX currentLocY

(+ currentLocX HalfRearCntSize) (+ currentLocY HalfRearCntSize))

)

) ; end of second condition

( else ;; place whole contact

(begin

(if (< HalfFrontCntSize (+ currentLocX RearCntSize))

(begin

(set! Xstart (max currentLocX HalfFrontCntSize))

(PlaceOpt optWinName optPlaceName Xstart currentLocY

(+ currentLocX RearCntSize) (+ currentLocY HalfRearCntSize))

)

)

(set! totalCnt (+ totalCnt 1))

(set! currentLocX (+ currentLocX RearCntPitch))

(set! restSpaceX (- restSpaceX RearCntPitch))

)

)

)

)

)

)

)

8.3.2019

* Plot range: see structure editor manual p. 125. Plot is used to save and plot data during a quasistationary ramping process.

Plot {Range = (0 1) Intervals=5})

Here, six plot files are saved at five intervals: t = 0, 0.2, 0.4, 0.6, 0.8, and 1.0.

t is the time

* Note: The plot controls of the Transient command are the same as for the Quasistationary

command, except the is the real time (in seconds), and is not restricted to an interval from 0

to 1.

* If using the command

Plot {Range = (0 1) Intervals=1})

the files will be saved and plotted at 0 and 1.

* Questions to answer:
* What is the x column in the optical generation file (depth in um?)
* Do we need to make a transient calculation in the solver before the quasistationary calculation?

8.4.2019

* Sentaurus structure editor does read the external optical generation profile and adds it to the structure (verified in Sentaurus visual). Possible problems with the optical generation files:
* Sentaurus device cannot find the extracted mesh file, see if the correct mesh filename is imported (@tdr@).

Edit: not the problem, as even when including a fixed optical generation rate into sdevice, the solar cell curve is not shifted.

8.5.2019

* Plan for Sentaurus simulations framework based on DoE proposal:

1. Simulate stacking faults by including regions in the pn junctions with surface defects. Add Schottky junctions causing shunting.
2. Simulate with PC1D the effect of Na on lifetime. Then can extract SRH parameters and include them in Sentaurus.

* Need to know how fast Na gets into stacking faults. Is there some experimental data available in the literature?
* 2 effects of Na on solar cells: lifetime and shunts (see literature)
* Questions:
* How much Na enters the stacking faults? At what rate? See Ziebarth et al (ref 73 in PID review)
* Can we model the formation of a Schottky junction as a function of sodium density in stacking faults? What should be assumed for the defect states?
* To add dislocation: add a region inside some parts of the silicon and place defects at the interface between these region and silicon.

8.16.2019

* Include generation in Sentaurus, based on the file from MIT/ISE:
* First create the structure
* Then run an IV sweep in the dark (may not be needed for the light curve)
* Then run a file under illumination but without bias
* Then finally run a file using the illumination .tdr file created in the run under illumination, and sweep bias.

But that shouldn’t be necessary (optical generation and voltage sweep done in the same file in the Sentaurus example Optbackcontact)

Edit: actually the problem was just that the contact area was defined too small (so the photogenerated current was already calculated before but very small so not visible on the IV curve).

* Problem might be that the cell area is very small and so is the photogenerated current → Check in the MIT example how they calculate the current (what is the unit?) and what is the value of the device area.
* Area is determined from contact area. The area is defined in the contact section as it is needed to calculate the total current going through the contacts. See “AreaFactor” in Sentaurus device manual p. 10.
* In MIT file, the area is set in the electrode definition. Area=1.67e8 um with a back contact width of 600 um gives a total area of 0.1 m2, ie 1e3 cm2. So the current going out of the cell is in mA/cm2 without the need to rescale.
* If we have a back contact size of 50 um (our case), to obtain a contact area of 1000 cm2 (which will output results in mA/cm2 since Sentaurus outputs current in A), we need a correction of 1000e-4/50e-6 = 2e3 m = 2e9 um. So AreaFactor = 2e9 since Sentaurus distance units are in um.

8.18.2019

* To do next:
* ~~Compare results with and without illumination and see if there is a shift.~~
* Make a real solar cell with real optical generation file (use the one from MIT)
* Make a batch processing file using the existing template (use matlab script to replace @ parameters, so that the file can still be read in Sentaurus workbench)

8.21.2019

* To run Sentaurus visual from a command line:

See Sentaurus visual user manual p. 1.

Svisual n2\_fps.tdr

* Metal resistivity: see sdevice user manual p. 1290. To implement a user-defined resistivity model, Sentaurus PMI tool must be used (PMI = Physical Model Interface). Note that the PMI is used to implement the model, while the resistivity itself is implemented in the parameter file. A PMI is an interface that can be used to add own models to Sentaurus, see sdevice manual p. 1077. See Beneventi’s lecture slide, chapter 3. See sdevice manual p. 1079 for instructions on how to write and run the PMI.
* PMI for transport in metal, see p. 281.
* Also p. 329: the resistivity of metals can be changed in the parameter file. The manual mentions that the metal conductivity is calculated directly in the contact equation. Also mention that “No specific keyword is required in the command file because Sentaurus Device recognizes all conductor regions and applies the appropriate equations to these regions and interfaces”. Does that mean that it will be applied even if it is not a contact? How about when defining a metallic region outside of the contact area? For instance a shunt.
* Electric boundary condition for metals p. 282. Displacement current is based on Maxwell’s equation (second term in Faraday’s law).
* Using tcl language to extract parameters: see sdevice manual p. 162 and 163y. Example on how to compute average electron conductivity in the channel region of a transistor.

CurrentPlot {

Tcl (

Dataset = "Ave\_channel eConductivity"

Function = "Conductivity"

Formula = "set q 1.602e-19

set n [tcl\_cp\_ReadScalar eDensity]

set mu [tcl\_cp\_ReadScalar eMobility]

set value [expr $q \* $n \* $mu]"

Operation = "Average Region = channel"

* Simulation results and current files p. 153 sdevice manual.

→ the current file contains the terminal characteristics obtained during the numeric experiment

→ plot files allow to visualize device internal quantities

→ log files allow to investigate the simulation procedure itself

* Note: there are two types of currentplot commands, not to be confused:

- the currentplot *statement*, called in the solve section of sdevice, which determines exactly which points are written to the current file (sdevice manual p. 153)

- the currentplot section, called in the top level of the .cmd file (not in the solve section), and used to track additional data in the current file (sdevice manual p. 156).

8.27.2019

* Note: Xming needs to be open to use matplotlib if running python from the Unix server. Otherwise will display an error “could not open display”.
* To try to change Al resistivity: create a parameter file for Al and change resistivity there.
* Add a new material to sde: apparently sde does not need a material input, it is only defined in the parameter file and called in sdevice: <https://www.researchgate.net/post/how_can_i_include_a_new_material_in_sentaurus_structure_editor_and_how_to_export_the_new_parameter_file_to_the_tcad>

Jun-Sik Yoon

added an answer

To add some comments, sde does not need parameter file. After you assign new materials in sde, sdevice will calculate physics using tdr file made by sde and parameter file you wrote.

If the material that you want to include was not in the "include parameter files" option, then you need to assign that material in the parameter file.

Best regards,

Jun-Sik Yoon

* Actually sdevice needs a material that is in the database, so the names cannot be changed in sdevice and in the .par file. List of materials recognized by Sentaurus, found from the GUI of Sentaurus device editor (output when sdevice gave an error when trying to define a modified Al material in the parameter file):

|  |  |  |  |
| --- | --- | --- | --- |
| 4HSiC  6HSiC  AlAs  AlGaAs  AlGaN  AlGaP  AlInGaAs  AlInGaN  AlInGaP  AlInN  AlInP  AlN  AlP  Aluminum  Ambient  Anyinsulator  Anymaterial  Anysemiconductor  BN  BSG  CdTe  Ceramic  Cobalt  CobaltSilicide  Copper  Default | GaAs  GaAsP  GaAsSb  GaInP  GaInSb  GaN  GaP  GaSb  Gas  GatePolySilicon  Germanium  Gold  HfO2  HgCdTe  HgTe  InAlAs  InAs  InAsP  InAsSb  InGaAs  InGaAsP  InGaAsSb  InGaN  InN  InP  InSb  Insulator1 | Insulator2  Insulator3  InsulatorX  InterfacialOxide  Iron  Metal  Molybdenum  Nickel  NickelSilicide  Nitride  Oxide  OxideAsSemiconductor  Oxynitride  Photoresist  Platinum  PolySi  PolySilicon  Resist  Si3N4  SiC\_4H  SiC\_6H | SiO2  Silicide  Silicon  SiliconCarbide  SiliconGermanium  Silver  Solder60\_40  StrainedSilicon  Tantalum  TiN  TiNitride  TiSi2  Titanium  Tungsten  TungstenSilicide  Vacuum |

* If a new material name is really needed, custom materials can be added, see “registering custom materials for device simulations” in sdevice manual p. 48.

9.3.2019

* Note: Sentaurus PMI can be used for instance to implement a new Auger recombination model (such as Richter’s), see example in sdevice manual, standard C++ interface p. 1079.
* See classes and derived classes in C++ at <https://www.cs.rochester.edu/~nelson/courses/csc_173/review/CppClasses.html>.
* See sdevice manual to create a class to define linear metal resistivity.
* It seems that it is not possible to add a resistivity profile varying in depth, as the resistivity.
* Note that the spatial distribution of traps can be modified using a PMI, see sdevice manual p. 1224.
* See also p. 284 to set the positional dependency of the workfunction in metals (specified directly in the physics section). Can also specify a dataset name from which the workfunction values will be taken using the SFactor parameter. Or a space factor PMI model can be specified to calculate the workfunction values directly. See space factor p. 1224.
* The space distribution of the following parameters only can be modified by a space factor PMI, according to p. 1224:
* Metal workfunction
* Traps
* Bond concentration
* Extended nonradiative multiphonon model precursor concentration
* Piezoresistance enhancement factors
* See Appendix B p. 1359 for Sentaurus Device syntax. See p. 1369 for Data and Plot names
* P. 1495, SFactor is defined as being used to input dataset for variation.
* Strategy to adopt for now for spatial variation of conductivity in a shunted region: define several regions in sde, each region corresponding to a material with a different resistivity.
* See PMI\_SpatialDistributionFunction class defined in Sentaurus’ PMI. It can be used to implement a Gaussian spatial distribution function. See sdevice manual p. 1289. Actual seems to be specifically for the case of heavy ion impinging on semiconductor surface.
* Other idea: change resistivity by doping with a metal?? See sde

9.5.2019

* Sever 242 seems to be working fine, no disconnections.
* Next steps for modelling:
* Make function to add new materials to the index based on the number of discretized shunt regions defined in the sde file (choose a fixed step for region depth and find the number of regions depending on the sodium profile depth from Erick’s simulations).

9.18.2019

* Instructions on 1D external profile selection in Sentaurus Structure Editor manual, see p. 221. The keyword needs to be written at the beginning of the .plx file (they refer to it as the species). *“For a 1D profile, the species is read from the .plx data file and the selection in the Species field is ignored”.*
* Explanation on how to access the complete list of species for introduction of an external profile. First open the user interface of sde (by typing “sde” in the command line), then go to device → analytical profile placement and select the dropdown menu “species”, then select “other” to see the complete list of all DATEX variables (different doping species, metal conductivity, etc). See sde user manual p. 218: *“Select a species from the list, which contains the most common dopant species. Select Other to access the complete list of DATEX variables. Enter a value for the concentration of the dopant.”*
* To change a region conductivity, define a metal and include an external .plx file with “MetalConductivity” as the header.

9.24.2019

* Analytical profile placement for the external conductivity profile: command define-analytical-profile placement.
* The symmetry can be “negative”, “positive” or “both”. Explanation of the parameters in Sentaurus Mesh manual p. 28: “Placing Analytic Profiles”. It seems “positive” and “negative” control the direction of the profile. “By default, values are computed on both sides of the Element. If Direction is specified, function values are computed only on the positive or negative side of the Element.”

10.18.2019

* Next things to do with Sentaurus:
* Fix optical generation, currently it is does not go beyond a few nanometers from the surface
* The shunt window depth is not set correctly, check why (might be because the shunt profile is deeper than the shunt region)
* Approaches to fix problem:
* Defined a base cell depth larger than the optical generation file. But that is probably not the issue here.
* Corrected the shunt depth in the batchshunt function (previously using a too small shunt depth instead of the value multiplied by 100).

Problem: no values in the conductivity file created in the last simulations (testdir5).

11.2.2019

* Setting simulation temperature:

See sdevice manual p. 25. Temperature is set to 300 K by default.

* Rotate ylabel in Matlab

set(get(gca,'Ylabel'),'Rotation',0)

11.27.2019

* “not a mixed-element geometry”:

https://www.edaboard.com/showthread.php?187762-sdevice-error-while-reading-grid

“It's a long time ago but I think it has something to do with the meshing. try tensor or axis aligned meshing and this should somehow work”

* Sentaurus mesh user manual p. 57:

“Converting Input Mesh to Mixed-Element Mesh This option converts an input mesh containing only tetrahedral elements to a mixed-element mesh containing hexahedra, prisms, pyramids, and tetrahedra:”

1.14.2020

* Email from Maryam on 11/1/2019 regarding server instability

Hi Guillaume,  
Thank you for sending the screenshot.  
From the message you're getting it's a network issue where the session is idle and connection gets lost. A couple of things to try:

1. In Putty under Connection -> Seconds between keepalives (0 to turn off) -> enter 30

2. SSH to [ieng6-702.ucsd.edu](http://ieng6-702.ucsd.edu/) with this connection option added

Traffic on ieng6-702 is low and we can better monitor your job. Please try this and let me know how it goes

Thank you

\_Maryam

1.15.2020

* Implementation of SRH recombination in Sentaurus.

Instructions to define SRH recombination in specific regions in example 1 “Optimization of rear contact design in monocrystalline silicon solar cell using 3D TCAD simulations”. See p. 7 and p. 8 of the pdf.  
See also example 2 “Three-dimensional All-Back-Contact Monocrystalline Silicon Solar-Cell Simulation” p.5 for implementation of trap energy levels.

* In example 1, the parameters Si\_Al\_SRH\_S0, Si\_Silver\_SRH\_S0, Si\_FrontArc\_SRH\_S0 and Si\_RearRc\_SRH\_S0 (and their values) are defined in the Sentaurus Workbench interface. They are read by Sentaurus through the command Parameter= “@parameter@” in the sdevice\_des.cmd file.  
  See sdevice manual p. 65, Physical Model parameters.  
  Apparently this refer to the .par file, which contains the physical recombination model and the recombination parameters for each interface.

Currently in the PID simulations, the .par file contains several recombination model (Auger, CDL, etc) and, the Scharfetter SRH model and a surface SRH recombination model with a S0 value of 1000 cm/s.  
→ Need to add hole and electron recombination parameters manually to the .par file?

1.16.2020

* It seems that SRH recombination parameters are added in the following way in the cSi\_BackContOpt example.

First, the parameters Si\_Al\_SRH\_S0, Si\_Silver\_SRH\_S0, Si\_FrontArc\_SRH\_S0 and Si\_RearRc\_SRH\_S0 are created manually in Sentaurus Workbench (“Parameter” → ”Add parameter/Value”). Then values for these parameters are entered manually by creating experiments in Sentaurus workbench (“Experiment” → ”Add New Experiment”). Then the sdevice.par file was probably edited manually to include these parameters. For instance, the sdevice.par file contains the following command:

RegionInterface = "frontArc/substrate" {

SurfaceRecombination {

S0 = @Si\_FrontArc\_SRH\_S0@, @Si\_FrontArc\_SRH\_S0@ \* [cm/s]

Sref = 0 \* [1]

}

}

Note that the model of doping-dependence of surface recombination is used here, but the doping-dependence term is set to zero by Sref=0. So the only term remaining is S0=Si\_FrontArc\_SRH, which is the user-defined fundamental SRH surface recombination velocity. See surface SRH recombination model p. 446 and p. 447.  
So each time Sentaurus is run, the value of recombination is input to the parameter file from Sentaurus Workbench to calculate corresponding recombination.

Q: 1) Where is the link between the parameters defined in Sentaurus Workbench and the recombination model in the parameter file? Does it have to be defined manually?  
→ Maybe it is a general feature that can be applied to the parameter file? See Sentaurus online help at <http://www.sentaurus.dsod.pl/sd/sd_7.html#6>, “7.6 Parameterized Sentaurus Device Parameter Files”: “All the techniques discussed regarding parameterization in Sentaurus Workbench also can be applied to the Sentaurus Device parameter file.” → Parameterization of the sdevice .cmd file can also be applied to the .par file as well by adding “@” characters.  
2) Is there a more straightforward way for batch simulations? For instance by editing directly the .cmd file?  
3) Where are these instructions in the Sentaurus manual?  
4) Is the modified .par file going to be used? It seems a new one is automatically created during each simulation? Need to test if we see an effect of a change of Sn in a test simulation.

Next steps with Sentaurus:

* propose to ASU the model s = S0 ( 1 + Sref ( N/Nref )^gamma ) for Sn and Sp as a function of sodium concentration, as in the papers by Cuevas and King and Swanson (see sdevice manual p.409).
* Implement surface recombination. In the device.par files, there is a SurfaceRecombination component automatically created with S0=1000 cm/s by default. Is it going to be overwritten by adding a surface recombination value at a region interface, as in the example above? Probably as it is the way it is done in the example cSi\_backcontactOpt?

See <http://www.sentaurus.dsod.pl/sd/sd_4.html>, “Regionwise Physics Section”:

A Physics section with a region qualifier activates additional physical models in the specified region in addition to those defined in the global Physics section:

Physics( Region= "AlGaAs\_Emitter" ) {

MoleFraction(

XFraction=0.3

Grading(

(RegionInterface = ( "GaAs\_Emitter\_Cap" "AlGaAs\_Emitter" )

XFraction=0.0 GrDistance=0.10)

(RegionInterface = ( "GaAs\_Base" "AlGaAs\_Emitter" )

XFraction=0.0 GrDistance=0.025)

)

)

}

A regionwise Physics section overrides a materialwise Physics section if their definitions overlap.

* Note: splitting parameter files for better readability:

<http://www.sentaurus.dsod.pl/sd/sd_7.html#6>

“7.5 Including Files

As previously mentioned, Sentaurus Workbench preprocesses the Sentaurus Device parameter file only if it is called sdevice.par. However, it may be advantageous to maintain a separate parameter file for each material or region for better readability and then to include all the parameter files in the sdevice.par file using the #include preprocessor directive.

Click to view the parameter file sdevice.par.

For example, the parameter file sdevice.par includes the following #include clause:

Material = "Silicon" {

#include "Silicon.par"

}”

* Plan for surface recombination velocities:

Model surface recombination at first using phosphorus parameterization.  
Do not model recombination at shunt based on Needleman’s model for now.

If possible model effect of Na recombination in the Si bulk, if we get data from ASU’s DCPM measurements.

1.24.2020

* The .tdr output files from Sentaurus are HDF files and can be opened with the h5py package. The structure is the following.

>> hf['collection']['geometry\_0'].keys()

>> Out[75]: <KeysViewHDF5 ['offset', 'reference coordinate system', 'region\_0', 'region\_1', 'region\_2', 'region\_3', 'state\_0', 'transformation', 'vertex']>

where the region numbers seem to represent different Sentaurus regions defined in the .sdr file.

* Datasets are included in the key ‘state\_0’ . Additionally, each dataset has a name that can be obtained from the attribute.

>> hf['collection']['geometry\_0']['state\_0']['dataset\_34'].attrs['name']

>> Out[71]: b'eQuasiFermiEnergy'

* To see the list of attributes:

>> list(hf['collection']['geometry\_0']['state\_0']['dataset\_33'].attrs)

>> Out[81]:

['conversion factor',

'location type',

'name',

'number of values',

'quantity',

'region',

'structure type',

'unit:exponent',

'unit:long name',

'unit:name',

'unit:significand',

'value type']

* Threshold of Na concentration for bottom-up PID modelling. Based on preliminary PID degradation model presented in the DoE Q10 meeting. Degradation starts at the point t=12096 s = 3.36 hours.

1.26.2020

* Memory error when running too many simulations:

Errno 122: disk quota exceeded

* Created a backup folder in google Drive for heavy files. Path is G:\My Drive\Sentaurus\_backup.

1.27.2020

* It seems that Sentaurus sometimes does not find the interfaces when an interface surface recombination is defined. Approx. 25 simulations went fine with the SRV model and then an error message was returned during the next simulation.

'.//20200127\_curve\_SRH2//n\_t37920\_msh.tdr' ...

coordinate system: DFISE, 2d\_dfise (y is device down direction, 3d extension right-handed)

use coordinate system as is (no transformation)

TDR format

Number of grid points is 33819.

done.

Adding interfaces and contacts ...

Check UNKNOWN interface in Physics of the input file: "SiNx-region/emitter-region"

or use Math{-ExitOnUnknownParameterRegion} to skip this error message !

Mon Jan 27 17:22:35 2020: checked in 1 sdevice license(s)

Data in .//20200127\_curve\_SRH2.

1.28.2020

* Error function profile defined in Sentaurus Structure Editor manual p. 555.

(sdedr:define-erf-profile definition-name species "SymPos" symmetry-position

"MaxVal" max-value | "Dose" dose "Junction" junction

"ValueAtDepth" value-at-depth "Depth" depth "Length" length

"StdDev" standard-deviation "Gauss" | "Erf" "Factor" factor

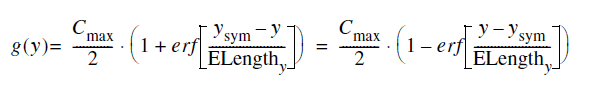
"StdDev" lateral-standard-deviation "Length" lateral-length)

* Parameters for Sentaurus analytical profiles (error function, Gaussian, etc) are detailed in the Sentaurus Mesh user manual, “General implantation Models”, starting p. 114.
* *The minimum set of parameters to define an error function as a doping profile is:*

*■ Maximum concentration (Cmax ) [cm-3]*

*■ Symmetry position (ysym) [um]*

*■ Length (ELengthy) [um]*

*Using these parameters, the error function is defined by:*

* Modelling of an emitter profile assuming infinite source diffusion:

Where c’/2 is the surface concentration.

Corresponding parameters in Sentaurus:

* c’=4e19 cm-3 for a surface concentration of 2e19 cm-3
* Ysym=0
* ELength=0.2 um

ELength=2sqrt(D\*t)= 2\*sqrt(1e-14\*5\*3600)=0.27 um with some reasonable parameters for B in Si. Gives emitter depth of 800 nm at 1e15 cm-3 base doping.

To reach 400 nm at 1e15 cm-3, ELength=0.15 um

Reach 600 nm at 1e15 cm-3 base doping with ELength=0.2 um (corresponding to about 170 Ohm/sq sheet resistance, see paper kerr et al, JAP 89, 2001)

* See example at <https://github.com/AnastasiiaV/TCAD> for an example of error function profile definition.

1.29.2020

* When defining an analytical profile, the number of points depends on the mesh resolution. Need to add a mesh refinement in the doping region to have a more accurate profile.
* To open the SSH shell directly in the correct directory, the path can be added directly to Putty.

<https://stackoverflow.com/questions/3530433/putty-change-default-ssh-login-directory>

* They is a large lateral smoothing in the external shunt profile definition. Need to remove it. See Sentaurus Mesh user manual p. 120. See “No lateral function” p. 121, “Factor” needs to be set to 0.

1.31.2020

* For Tcl commands (if loops, for loops, etc): <http://www.sentaurus.dsod.pl/tcl/tcl_b.html>
* Actually, tcl is not used in the sde files. See <http://www.sentaurus.dsod.pl/sse/sse_e.html> for scripting inside tcl files. Sde is based on Scheme language, which is a LISP-like programming language. See Scheme syntax in the structure editor manual.  
  Find how to run Scheme directly in the linux terminal.
* New doping profiles from Erick in

G:\Shared drives\FenningLab2\Projects\PVRD1\Simulations\FEniCS\PNP\SWEEP\_D,H

→ idea was first to use the D2 and h in the green figure of the figure 5 in the DoE report Q11 (C0 1e17 cm-3).

But maybe it makes more sense to use C0=1e20 cm-3 and D2=1e-14 cm2/s and h=1e-10 or 1e-11 cm/s for now, even though it's in contradiction to the low concentration hypothesis in the CV paper, because they produce a Vsb shift more similar to what we find experimentally.

* Next Sentaurus steps:
* (Simulate more points with the model used for SRH comparison and confirm that PID catastrophic failure is obtained after 9 hours) maybe try directly the new doping profiles from Erick
* Then, plot Na profile at 9 hours to see the threshold Na concentration

2.1.2010

* Simulations did not converge because the polarity was switched (set to phosphorus emitter and boron base). Worked again when using a boron emitter and phosphorus base.

2.2.2020

* Loss of convergence when adding a rectangle for the metal region with command

(sdegeo:create-rectangle (position shunt1\_pos\_x1 0 0) (position shunt1\_pos\_x2 dshunt1 0) shuntmat "shunt1\_region")

2.3.2020

* Need to add time step in the migration profile to the log file.
* Simulation results showing PID degradation (efficiency drop) with a recent doping profile and uniform emitter: folder sim\_20200126\_test\_3\_backup. Possible explanation for the drop is that the metallic region defined on top of the shunt has its own conductivity set by Sentaurus to some typical metal conductivity.

**→ repeated some simulations (repeat\_20200126\_test and repeat\_20200126\_test\_noregion) and that seems to be the case. The simulation with no region shows no conductivity at all (??) and no degradation**.  
Look in more detail why that is the case.

* Make a function that returns the simulation point corresponding to a certain efficiency (and later corresponding to 5 % eff decrease)
* Clathrate data in G:\Shared drives\FenningLab2\Projects\PVRD1\Literature\Clathrates\CrosArrheniusData.
* Check effect of the metal region (defined separately from the shunt) on the conductivity and PID failure.

Edit: the metal region seems to be the cause of PID. No efficiency decrease is observed when this metal region is removed. That is because the conductivity model returns only 1e-2 S/cm at the Na concentrations considered in these simulations. Only the defined metal region actually affects conductivity, and it seems it has a conductivity defined separately in Sentaurus default. See simulations in “/home/linux/ieng6/na299x/na299x/DB/Guillaume/cell\_AlBSF/Al\_BSF/repeat\_20200126\_test” and in folder “repeat\_20200126\_test\_noregion” (same path).

Edit 2/17/2020: It seems that defining a metal region is necessary to add an external conductivity. However, identical shunting seems to occur no matter the magnitude of the externally added conductivity profile. The metal region should not contribute to shunting, as its conductivity is set to approximately 4.5e-4 S/cm in the parameter file.

→ Try adding just the metal region but no external conductivity profile and see if we see a PID degradation effect (if what is written above is correct we should).

2.17.2020

* Checked whether the defined “Metal” region may contribute to shunting in addition to the externally defined conductivity profile. See note just above, apparently it shouldn’t contribute to any conductivity. Need to find why the same shunting occurs for all conductivity profiles used.
* This shunting may be due to the lateral spreading of the conductivity? Should be avoided by setting “Factor” to 0 in define-1d-external-profile, but still present. Look at “define-analytical-profile-placement” in sde manual and in particular in Sentaurus mesh manual p. 28, “placing analytical profiles”. See “EvaluateWindow” option that is used to restrict placement.
* Trying to run again ASU example (higher up in the folder hierarchy): PIDmodel.runSent.run\_sde('../../ASU\_conductivity\_profile/asu\_conductivity/sde\_dvs.cmd')

→ When the shunt is reduced to a very small region, the same lateral spread of conductivity can be observed. It seems to be an artefact that appears at small dimensions.

Next, will try larger shunt widths in PID simulations.

2.19.2020

* Error handling for Sentaurus device editor

Solution: create a file errfile.txt containing 1 if there is an error and 0 otherwise.

Then can have a try and catch around the code and create an errfile with value 1 if there is an error. The file needs to be initialized with value 0 so that it is reset when a new simulation is started.

Create a file: <https://stackoverflow.com/questions/4181355/file-i-o-operations-scheme>

* Complete documentation to write and read files: <https://www.scheme.com/tspl3/io.html> and see line “procedure: (call-with-output-file filename proc)”.
* To write command to linux from scheme: (system:command [commandname]), see “system calls” p. 278 of the sde manual.
* NEXT: for error handling, add a try and catch around the sde code and write 1 to the error file if there is an error.