DESIGN OF MULTI-JUNCTION SOLAR CELLS INCORPORATING SILICON-GERMANIUM-TIN ALLOYS WITH FINITE-ELEMENT ANALYSIS AND DRIFT-DIFFUSION MODEL

LAURIER BARIBEAU

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Science Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Ottawa

ABSTRACT

This study explores in detail design options and simulations of multi-junction solar cells that utilize silicon-germanium-tin ($Si_xGe_{1-x-y}Sn_y$ or SiGeSn) to achieve high-efficiency solar power conversion devices. $Si_xGe_{1-x-y}Sn_y$ is an emerging system of alloys that can lattice match with germanium and gallium arsenide and can provide a bandgap higher than that of germanium; useful in the development of multi-junction solar cells. The results herein include designs of four devices: a triple-junction, a quadruple-junction, a seven-junction, and a six-junction, with estimated efficiencies of 41.6%, 42.6%, 41.2%, and 39.2% respectively under 1000x concentrated AM1.5D illumination, where the seven- and six-junction devices relax the thickness requirement of the germanium layer, and have room for improvement via the development of an advanced tunnel-junction component. Visualizations of the potentially available SiGeSn bandgaps are developed. The documentation supports further work in modelling additional compositions of SiGeSn. Loss mechanisms of the devices are calculated and plotted, enabling the design of the device layer components. Tools and techniques are developed to determine and control the resultant output error, and a generalized simulation mesh definition is given that efficiently controls the primary source of error of the calculation, which is related to the optical interaction. Lateral currents and surface recombination effects are included. The software is modularized to enable the development of higher-order segmented devices.

TABLE OF CONTENTS

ABSTRACT ..... **II**

TABLE OF CONTENTS ..... **III**

TABLE OF ACRONYMS ..... **VI**

TABLE OF TABLES ..... **VII**

TABLE OF FIGURES ..... **VIII**

ACKNOWLEDGEMENTS ..... **XII**

STATEMENT OF ORIGINAL CONTRIBUTIONS ..... **XIII**

1: INTRODUCTION **1**

2: PHOTOVOLTAIC DEVICE MECHANICS **6**

2-1: Energy Levels 6

2-2: Complex Refractive Index 8

2-3: Mobility 9

2-4: Doping 10

2-5: Example Device Structure 11

2-6: *P-N* Junction Optoelectronics 12

2-7: Fill Factor 15

2-8: Minority Carrier Injection 15

2-9: Recombination 15

2-10: Surface Field Layers 17

2-11: Ohmic Contacts 18

2-12: Multi-Junction Solar Cells 18

2-13: Standard Spectra 19

2-14: Black Body Radiation 21

2-15: Concentrators and Satellites 24

2-16: Design Approaches 24

3: SILICON-GERMANIUM-TIN $Si_xGe_{1-x-y}Sn_y$ **26**

3-1: Growth 26

3-2: Band Structure 27

3-3: SiGeSn Composition Space Regions of Interest 29

3-4: Bandgap Function 34

3-5: SiSn Bowing Parameter b_{SiSn_Γ} Values and Validity 39

3-6: General Approach To Electron Affinity 43
3-7: SiGeSn Electron Affinity 46
3-8: Lattice Constant Matching Germanium 49
3-9: Permittivity 49
3-10: Electron Effective Mass 50
3-11: Complex Refractive Index 51
4: SIMULATION TOOLFLOW 53
4-1: Epi 53
4-1-1: Simulation Width 53
4-1-2: Layer CSV Table 54
4-1-3: Interfaces 55
4-1-4: Data Structure Setup 56
4-1-5: Mesh Refinement 57
4-2: Sentaurus Structure Editor 58
4-2-1: Cathode and Anode 59
4-2-2: Additional Contacts 60
4-3: Sentaurus Device 61
4-3-1: Optics 61
4-3-2: Poisson, Continuity, Drift and Diffusion Equations 61
4-3-3: Sentaurus Device Log File Tools 63
4-3-4: Solver Preferences 65
4-3-5: Plot Error Locations 67
4-3-6: Mesh Adaptation 69
4-4: Wavelength Ramp Circuit – Calculation of J_{sc} 69
4-4-1: Minority Carrier Loss at Contacts 72
4-4-2: External Quantum Efficiency 74
4-4-3: Internal Quantum Efficiency 74
4-4-4: Generation and Recombination Rates 74
4-5: Voltage Ramp Functions 75
4-6: Sentaurus Visual Plot Development 77
4-7: Plot Development with Python Matplotlib 81
4-8: Workbench Project Settings with gtooldb.tcl 83
4-9: Other tools 83
4-10: Development Practices 84
5: RESULTS 85
5-1: Four-Junction Cell with Front Three Subcells Current-Matched 85
5-1-1: Error Reduction by Mesh Adaptation 94
5-2: Four-Junction Device Current-Matched by Thinning Front Subcells 101

5-2-1: Germanium Subcell Back Surface Field 107
5-2-2: Germanium Subcell Front Surface Field 108
5-2-3: SiGeSn SRH Lifetime Parameter Sweep 109
5-3: Seven-Junction Design by 2:2:2:1 Segmentation 110
5-4: Optimization 118
5-4-1: Procedure 118
5-4-2: General Mesh for Two-Dimensional Structure 122
5-4-3: Three-junction device 123
5-4-4: Four-junction device 127
5-4-5: Seven-junction device 130
5-4-6: Six-junction device 134
6: CONCLUSION 137
WORKS CITED 140
APPENDIX A: CONDUCTION BAND OFFSET CALCULATION 145
APPENDIX B: EFFECT OF LATTICE CONSTANT BOWING ON GERMANIUM LATTICE MATCHING SIGESN COMPOSITIONS 146
APPENDIX C: SIGESN ELECTRON EFFECTIVE MASS 148
APPENDIX D: DIVIDING MULTIBOX MESH CUSTOMIZABLE WITH WORKBENCH VARIABLES 149
APPENDIX E: PLOT COORDINATES FROM STANDARD INPUT 151
APPENDIX F: OBJECT WITH DATA AND CONVERSION FACTORS 152
APPENDIX G: MERCURIAL REPOSITORY IGNORE FILE .HGIGNORE 153
APPENDIX H: AUTOMATIC SCALE AND GRID WITH MATPLOTLIB 154
APPENDIX I: PLOTTING ENTIRE DATASET TO INSPECT DISCRETIZATION 155
APPENDIX J: TCL INTERFACE TO SDE SCHEME FOR 1D MESH REFINEMENT 156
APPENDIX K: ONE-DIMENSIONAL MESH FOR LOW ERROR 157
APPENDIX L: GENERAL 2D MESH REFINEMENT 158

TABLE OF ACRONYMS

AM1.5D	Air Mass 1.5, Direct
AM1.5G	Air Mass 1.5, Global
ARC	Anti-Reflection Coating
ASTM G173	American Society for Testing and Measurement, Radiometry subcommittee
BSF	Back Surface Field (layer)
CBO	Conduction Band Offset
CMOS	Complimentary Metal Oxide Semiconductor
CPV	Concentrator Photovoltaics
CSV	Comma-Separated Variables
EQE	External Quantum Efficiency
FSF	Front Surface Field (layer)
GUI	Graphical User Interface
IQE	Internal Quantum Efficiency
LCOE	Levelized Cost of Energy
MJSC	Multi-Junction Solar Cell
RHS	Right Hand Side
SDE	Sentaurus Structure Editor (Device)
SRH	Shockley-Read-Hall
SRV	Surface Recombination Velocity
TCAD	Technology Computer-Aided Design
TCL	Tool Command Language
TMM	Transfer Matrix Method
VBO	Valence Band Offset

TABLE OF TABLES

Table 1: Input Values for Solar Black Body Radiation Calculation. 23
Table 2: Solar Black Body Radiation Calculation. 23
Table 3: Example Epi CSV Table. 54
Table 4: Legend for Quantum Efficiency Plots. 87
Table 5: Legend for Figure 27 a-d and Figure 33. 88
Table 6: Four-Junction Device Design with Thinned Front Subcells. 101
Table 7: $Si_{14}Ge_{82.2}Sn_{3.8}$ Valence Band Offset Calculation. 108
Table 8: Seven-Junction Epi CSV 111
Table 9: Triple-Junction Device Epi Layer Specifications 124
Table 10: Layer Specifications for Four-Junction Solar Cell 128
Table 11: Layer Specifications for Seven-Junction Device 131
Table 12: Layer Specifications for Six-Junction Device 134
Table 13: Summary of Simulated Devices' Subcell Total Thicknesses 137
Table 14: Summary of Device Characteristics for two SRV values, with 1 μs SiGeSn SRH lifetime. 137

TABLE OF FIGURES

1	Calculated current-voltage and power-voltage characteristics by device area of the subcell components in a $Ga_{0.5}In_{0.5}P$/GaAs/Ge structure under 1000x concentrated standard AM1.5D sunlight.	2
2	Portions of the solar spectrum collected by each of the GaInP, GaAs, and Ge components when shorted.	3
3 a, b	a) Energy-level diagram for hydrogen, colour added from b) to the appropriate transitions. [5] b) Hydrogen lamp and its emission separated by wavelength after diffraction through grooves of DVD. [6]	7
4	Symmetry element of a single *p*-*n* junction device structure with names of regions.	11
5	Picture of a III-V multijunction solar cell at the Fraunhofer Institute for Solar Energy Systems. [13]	12
6	A schematic for the top electrode of concentrator solar cell. [56]	12
7	Example $J(V)$ and $P(V)$ characteristics from the simulated structure of Figure 4.	13
8	AM0, AM1.5D, and AM1.5G standard data, [19], [20], with the result of the Planck blackbody irradiance calculation of the sun's energy radiance onto Earth added.	20
9	Scanning electron microscope image of GeSn (top) on Ge virtual substrate on silicon wafer (bottom) grown at Polytechnique Montréal. Image taken by Ross Cheriton and Laurier Baribeau in 2018.	27
10	Band structures of silicon, germanium, and tin, as calculated and depicted in [25], cropped, modified labels and grid.	29
11	Bandgap of strained SiGeSn lattice-matched to germanium from tight-binding calculation and depiction in [24].	31
12 a, b	SiGeSn bandgap bowing formula output as a function of alloy composition.	38
13 a, b	SiGeSn bandgap model with bSiSn = 3.26 eV.	40
14	Bandgap vs Lattice Constant with bSiSn = 3.26 eV.	41
15	Bandgap vs Lattice Constant with bSiSn = 13.2 eV.	42
16	Diagram of energy band edge levels to calculate temperature dependence of valence band offset with respect to GaAs reference.	45
17 a-c	SiGeSn compositions that lattice match to germanium a) on triangular axes b) on rectilinear axes. c) Difference due to bowing term in tin composition needed to lattice match to germanium.	49
18	Calculated K result based on data from the two methods in [50] versus energy.	52
19	Calculated complex refractive index of SiGeSn versus wavelength. Composition is estimated in [50] as 14 atomic % Si, lattice matched to germanium.	52
20 a-c	Mesh vertices indicated by `CNormPrint` during a full wavelength ramp to be locations of maximum estimated error found from a) the Poisson equation, b) the electron continuity equation, and c) the hole continuity equation.	68
21	Schematic of external circuit showing current loops for Kirchhoff's Current Law.	70

22	Vector and colour contour plot of hole current density near the top contact with 300 nm incident light.	73
23	Sentaurus Visual GUI.	78
24	Current density *J(V)*, power density, and power density derivative curves for device with current-matched front three subcells.	85
25	Device EQE multiplied by AM1.5D spectrum converted to unit of photons/s/cm^2/μm.	87
26	External quantum efficiency.	88
27 a-d	Integrated a) optical generation, b) radiative, c) SRH, and d) Auger recombination in each layer.	89
28	Internal quantum efficiency.	90
29	Summation of optical generation by layer.	91
30	Difference between the total integrated optical generation and the probability of light absorption according to the TMM optics result.	91
31	Summation of optical generation renormalized by reflectivity and grid shade.	92
32	Discrepancy in optical generation total, renormalized to IQE unit.	92
33	Optical generation by layer, normalized with respect to reflectivity, log scale.	93
34	Discrepancy in optical generation total (overlap in Figure 31) in IQE unit, log scale.	93
35	IQE discrepancy plot (overlap in Figure 28), log scale.	94
36	Discrepancy in optical generation integration with a mesh multibox added in the front eighth of the germanium base.	95
37	Discrepancy in integrated generation normalized by reflectivity for device meshed entirely with ratio method, 50 pm minimum element size, and 1.15 ratio.	95
38	Visualization of mesh point spacing through the first three subcells of the device.	97
39	Plot of many physical quantities to inform mesh adaptation.	98
40	Discrepancy in generation normalized by reflection produced with mesh of Figure 38.	99
41	Subcell current and power density characteristics current-matched by layer thinning.	102
42	Current-matched four-junction device J-V characteristic.	103
43	Calculated EQE of four-junction design with thinned front subcells and tunneling model disabled.	104
44	Four-junction device EQE multiplied with the AM1.5D standard spectrum.	105
45	Recombination overlaid against optical generation by device layer.	105
46	Discrepancy in four-junction device EQE with tunneling model enabled and not plotted.	106
47	Discrepancy in four-junction device EQE with tunneling model disabled.	107
48	Energy band edge levels of SiGeSn and $Ge_{1-x}Sn_x$, lattice-matched, for *x* in 0, 5, 8, 10, and 12%. Merged data of figures from [46] and added $x = 0$.	108
49	Band diagram of SiGeSn BSF, tunnel junction, SiGeSn FSF in germanium subcell, and germanium emitter.	109

50	Calculated effect of SRH lifetime parameter on SiGeSn subcell current and power versus voltage curves.	110
51	Current density [A/cm^2], power density [W/cm^2], and power density derivative [$W \cdot cm^{-2} \cdot V^{-1}$] characteristics with respect to controlled voltage ramping of virtual contacts across the seven subcells under 1000x the AM1.5D spectrum with one-dimensional mesh.	112
52	Seven-junction device EQE.	113
53	Tally of carriers reaching contacts, recombining, with non-generative absorption.	114
54	EQE result from two-dimensional mesh, surface recombination coefficient set to $5*10^4$ cm/s, surface recombination model error acceptable from Sentaurus Device version *P*.	115
55	Subcell *J(V)*, *P(V)* under 1000x AM1.5D with two-dimensional mesh. Surface recombination velocity is set to $5*10^4$ cm/s.	116
56	Current density magnitude.	117
57	Current density magnitude and direction. Arrow-head points are located on mesh vertices.	117
58	Electron current density.	118
59	Hole current density.	118
60	Cathode location in one-dimensional structure, collecting more minority carriers.	119
61	Current density at the front of the seven-junction device.	120
62	Substrate layer mesh with more *x*-dimension coverage on surfaces than bulk, specified by using line and window refinement multiboxes.	123
63	Device *J-V* and *P-V* characteristics.	125
64	Subcell *J-V* and *P-V* characteristics.	125
65	EQE response with losses also shown, SRV=10^5 cm/s.	126
66	Spectral response of triple-junction device with SiGeSn bottom subcell.	126
67	Depiction of the growth portion of the device. Optical generation is shown.	128
68	EQE of four-junction device with SRV=10^3 cm/s at short-circuit condition.	128
69	EQE of four-junction device with SRV=$5*10^4$ cm/s at short-circuit condition.	129
70	EQE of four-junction device with SRV=10^6 cm/s, showing that the surface recombination model and mesh sources of error continue to pass the self-consistency test for a high degree of surface recombination.	129
71	Four-junction subcell *J-V* and *P-V* characteristics with two SRV values.	130
72	Depiction of seven-junction device.	131
73	Seven-junction device *J(V)* and *P(V)* characteristics.	132
74	Seven-junction subcell *J(V)*, *P(V)*, and $dP(V)/dV$.	132
75	Optical generation in each layer of seven-junction device, with bulk recombination integrated by region and superimposed. Surface recombination does not appear on this plot.	133
76	Seven-junction segment *J(V)* and *P(V)* characteristics with SiGeSn lifetime parameter sweep.	133
77	Six-junction device illuminated current and power density characteristics with respect to bias voltage for SRV=10^3 and SRV=$5*10^4$.	135

78	Six-junction device subcell current and power density characteristics for two SRV values, showing excess current available to the bottom subcells.	135
79	Six-junction device EQE with loss mechanisms.	136

ACKNOWLEDGEMENTS

I would like to express gratitude to everyone at the SUNLAB, our collaborators, and my professors for all their support, expertise, and informative conversations. I would like to extend utmost thanks in particular to Christopher Valdivia and to Professor Karin Hinzer for their outstanding mentorship and guidance.

I am also grateful to my parents, my partner, my siblings, and my other friends.

I was able to pursue this research thanks to financial support from grant NSERC STPGP 413276, the University of Ottawa Graduate Scholarship, and Electricity Human Resources Canada. Thanks also to CMC microsystems for licensing Synopsys Sentaurus TCAD.

STATEMENT OF ORIGINAL CONTRIBUTIONS

The following developments have occurred based on original work by me during the course of my M.A.Sc. program.

This conference paper and poster presentation contain a preliminary four-junction design and results following initial development of the SiGeSn material parameter models:

> L. S. Baribeau, C. E. Valdivia, M. M. Wilkins, K. Hinzer, "Modelling of Four-Junction Solar Cells with a Silicon-Germanium-Tin Subcell," in *19th International Conference on Numerical Simulation of Optoelectronic Devices (NUSOD-19)*, Ottawa, ON, Canada, 2019. [Poster]

I contributed to this paper in CPV-17:

> R.F.H. Hunter, C. E. Valdivia, L. S. Baribeau, K. Hinzer. "Unlocking the Potential for Extreme Solar Concentration via Subcell Segmentation," in *17th International Conference on Concentrator Photovoltaic Systems (CPV-17)*, 2021. *In print.*

This paper based on a design I completed has been submitted to PVSC-49:

> L. S. Baribeau, R.F.H. Hunter, C. E. Valdivia, K. Hinzer. "Drift-Diffusion Modelling of Four-Junction InGaP/InGaAs/SiGeSn/Ge Solar Cells" in *49th IEEE Photovoltaic Specialists Conference (PVSC)*, Philadelphia, PA, USA, 2022. *Under review.*

These technical developments in the simulation environment are crucial to the findings reported above and in this thesis. Device designs are presented herein are based on these developments of the modelling project.

Four designs with thicknesses and requirements analyses

- Three-, four-, seven-, and six-junction solar cell designs
- Layer thicknesses for current matching
- Reported performance metrics (J_{sc}, V_{oc}, Fill factor, efficiency)
- Includes surface recombination
- Includes lateral effects (2D)
- Parameter sweeps in four-junction and seven-junction

The six- and seven-junction designs are a manifestation of a novel concept in the field, segmentation, so the drift-diffusion modelling of the devices using that approach are of particular interest. My work builds upon a simulation project structure developed previously; I have made the following developments.

Implementation of SiGeSn material parameter models discussed in literature

- Bandgap model implementation based on bowing parameters
- Band offset model
- Complex refractive index for $Si_{0.14}Ge_{0.822}Sn_{0.038}$
- Electron effective mass
- Permittivity

SiGeSn Bandgap Bowing plots:

- Bandgap vs alloy composition in 3D triangular axes with angled tick marks
- Bandgap vs lattice constant showing L, Γ , and Δ regions

These take bowing parameters as input, which still have highly significant uncertainties in the literature

Mesh strategy to control error

- Low mesh spacing in front to control error from optical generation
    - Thin layers use constant, small mesh spacing
- Growing mesh spacing is to occur in large layers
    - Location of primary mesh error is then known to be within the large layer
    - Mesh asymmetrically
        - Control optical generation error in front,
        - Save time with coarse mesh at the back of the device
- X-line X-multiboxes on surfaces control surface recombination error while allowing coarse meshing in bulk
    - Significantly reduces problematic calculation time for large active substrate layer

TCL to SDE Scheme mesh description interface

- Meshing done in code, not table,
    - Automates meshing process for many designs
    - More flexible than Epi-table mesh refinement
- TCL API written to interface with SDE Schema mesh refinement functions
- Allows for
    - Asymmetric mesh in absorber region
    - X-line X-multiboxes

Project architecture extensible for segmented design

- Epi CSV file initializes TCL data structures that set remainder of project in motion
    - List of regions, list of important surfaces, number of segments
- Loops in Sentaurus Device, Sentaurus Visual, and Python command files need not also be edited

Band offset utility functions

- Material-independent generic implementations of calculations of Chi_relative given conduction band offset or valence band offset

IV_Ramp TCL function that writes two quasistationary ramps

- Reduces amount of code to manageable, maintainable level

Python and Sentaurus Visual code that processes Sentaurus Device output into EQE unit

- Minority carrier loss
- Calculation of current lost due to absorption in tunnel layers
- Surface recombination

1: INTRODUCTION

Increasingly, climate change proves to be a persistent and difficult problem before everyone on Earth today. Climate scientists realize that a reduction on the reliance of burning fossil fuels for electricity is advisable to discontinue irreversible and unpredictable changes occurring to the Earth's global ecosystem. At the current rate of greenhouse gas emissions, global warming of 2.0 ^{o}C relative to 1850 to 1900 is expected in 2050, and is currently 1.2 ^{o}C. [1] The problem has been known and has persisted for decades already, because actions to address it can be expensive, unpopular, or have little effect. It is simultaneously a social issue and a technological one. As the political will and economic environment come together and the necessary bold steps are taken, such as the commissioning of large, sustainable power stations and infrastructure, the most advanced, highly effective technologies must be made available, and must therefore continue to be discovered as rapidly as possible. Options to address this problem may yet be discovered in research, such as in concentrated photovoltaics (CPV.) This field contains many known avenues for development and improvement of system performance, and the future may hold cost-competitive or better performance compared to single-junction silicon solar technology. Working to increase the achievable conversion efficiency of the solar cell is a powerful way to improve system performance with respect to net system cost in both CPV and space applications. In this study, the emerging technology of silicon-germanium-tin ($Si_xGe_{1-x-y}Sn_y$ or SiGeSn) semiconductor alloy growth is considered in the design and optimization of high-efficiency multijunction solar cells.

A standard high-performance multi-junction solar cell uses three junctions to collect the power carried by sunlight while minimizing the energy loss in the conversion to electrical power. There is a clear path for improvement upon this design, which does not distribute the photons evenly among the three junctions. The bandgap of the semiconductor determines the share of photons that it can absorb. This standard cell uses GaInP, (In)GaAs, and germanium, with bandgaps of 1.86, 1.4, and 0.66 eV. This partition allocates approximately double the number of photons to be available to the bottom subcell compared to the front two subcells, as shown in Figures 1 and 2. This calculation is done with a simulation setup based on previous work [2-4] and has been further developed in this work.

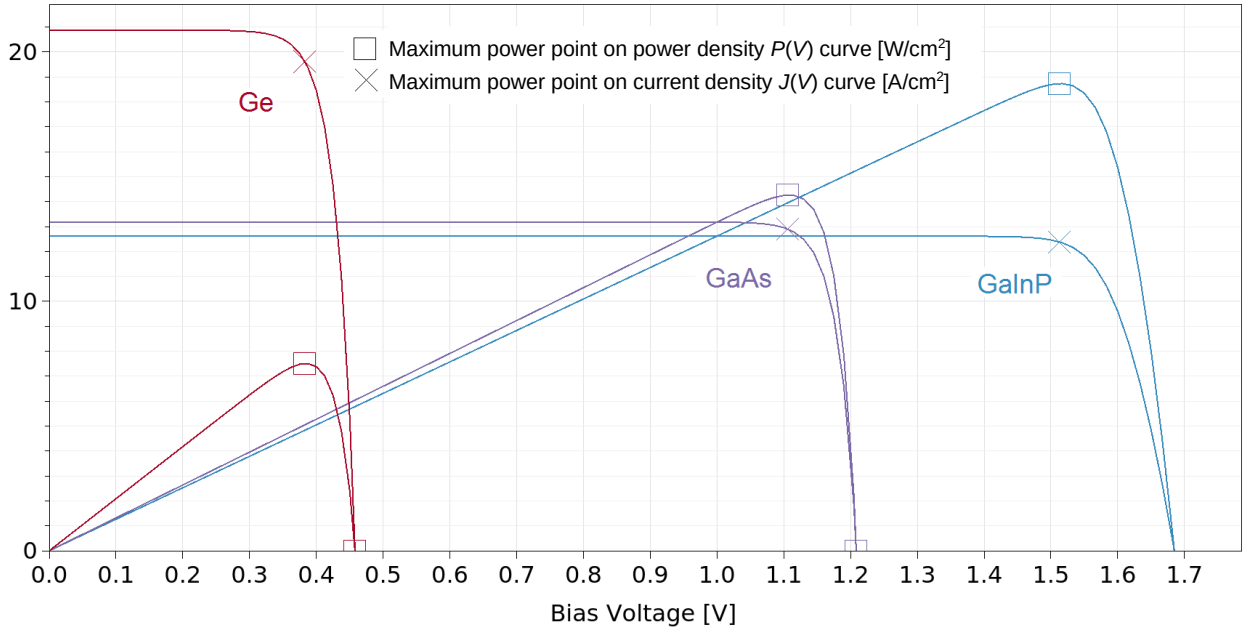

Figure 1: Calculated current-voltage and power-voltage characteristics by device area of the subcell components in a $Ga_{0.5}In_{0.5}P$/GaAs/Ge structure under 1000x concentrated standard AM1.5D sunlight. The additional current in the germanium represents unrealized potential energy, since each component must operate with the same current when grown on one wafer and therefore series connected.

In these calculations, the currents produced by the three subcells are separated out by a virtual diagnostic circuit. The germanium component shows more current to be available than the GaInP and (In)GaAs. However, these components are to be connected in series, and must therefore all operate with the same current according to Kirchoff's Current Law. Current moves between adjacent components via a tunnel junction, where conduction energy level electrons move into the valence energy level of the adjacent subcell. This can only occur at the rate that vacancies or holes are provided at that energy level by the subcell adjacent in the circuit. Therefore, the germanium subcell operates at a point well below the maximum power point indicated in Figure 1 as determined by the other two subcells. Many of the electrons energized by the sunlight must fall back into the covalent bonds in the germanium because the other components limit the current in the circuit. The germanium then produces slightly higher voltage, but so much less current that the

potential energy of the photons that it has collected has not been fully realized. These physical concepts are described more completely in chapter 2.

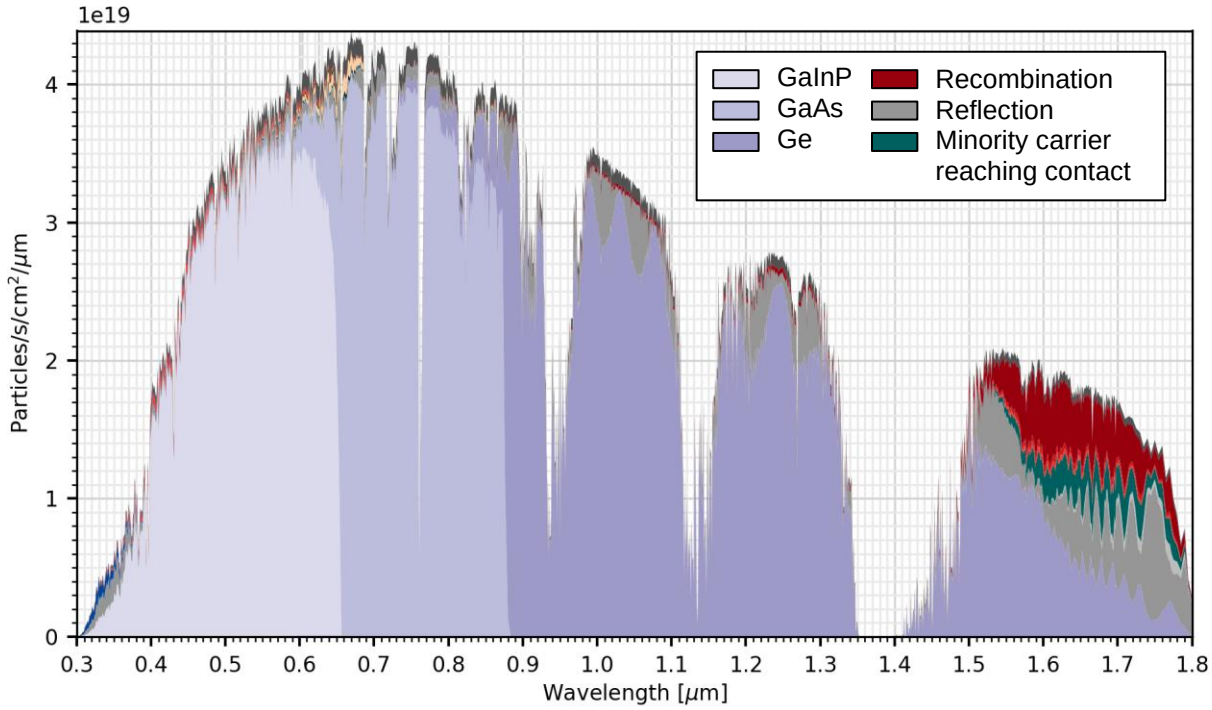

Figure 2: Portions of the solar spectrum collected by each of the GaInP, GaAs, and Ge components when shorted. The colored areas correspond to the short circuit current of the components plotted at $V = 0$ in Figure 1, and shows that the germanium component has more photons available to it than the other two.

If a semiconductor can be produced with higher bandgap than germanium, around 1.0 eV, then the germanium component could be replaced by that semiconductor, which would be able to produce higher voltage if the material can be produced with high quality. In this trade, the higher bandgap semiconductor interacts with less of the solar photons, reducing the current achievable by the component under the short-circuit ($V = 0$) condition. This is a desirable trade, since the current is limited by the other subcells in the circuit anyway. This arrangement appears likely to improve the power produced by the device, and likewise its efficiency in the photovoltaic conversion of solar power.

Additionally, it appears possible to design a four-junction device with such a semiconductor. If the current shown to be producible by germanium is shared by two components on the bottom of the device, then the current of the GaInP and GaAs can almost be achieved, with approximately double the voltage being produced on the bottom. This is another strong candidate design explored in detail herein.

For these reasons, attempts to develop a 1.0 eV subcell are often undertaken. However, the choice of materials is limited, and a semiconductor with this particular bandgap requires alloying or other techniques to produce. Silicon-germanium-tin ($Si_xGe_{1-x-y}Sn_y$ or SiGeSn) is a novel alloy of Group-IV materials in the diamond-cubic atomic geometry that is expected to produce a bandgap at or

near to this value, and the four-junction device incorporating a silicon-germanium-tin subcell with the other three standard subcells is the primary consideration of this study. The material behavior can be adjusted by tuning the relative amounts of silicon, germanium, and tin, as well as by introducing strain into the lattice. These degrees of freedom can be used to set the lattice constant of the material to match germanium, and to set the bandgap of the material. Additionally, there is a compositional range where the bandgap is direct, which would provide improved absorption characteristics.

These options can only be considered approximately without powerful device design tools. Models of the material physics are implemented here for the Synopsys TCAD Sentaurus device simulation toolset using the drift-diffusion carrier transport models. The tools are used to execute detailed finite-element calculations to simulate the performance of the device and design its components. The output of the simulations is a calculation of the expected movement of the charge carriers within the device and of power loss mechanisms, informing the optimization process and allowing device parameters such as layer thicknesses and doping levels to be determined.

This research is part of a collaboration with *Polytechnique Montréal* and *l'Université de Sherbrooke*, where the former provides novel expertise in the epitaxial growth of the Group-IV alloys forming the bottom part of the device, and the latter grows the III-V semiconductors at the front of the device, and completes the final fabrication steps.

The results of this study contain four proposed designs furthering four possible avenues to leverage the unique properties of SiGeSn in multijunction solar cells – a three-junction, a four-junction, a seven-junction, and a six-junction device. The three-junction device is the simplest and can serve as a practical initial prototype, the four-junction is the most intuitive arrangement that leverages the capabilities of the new material system, and the six- and seven-junction devices use the segmentation approach to reduce the thickness requirements of the component layers, especially the germanium. The six-junction device is less complex and requires less growth than the seven-junction. Both of them reduce cost risk related to the price of bulk germanium used in the four-junction design, and are more resilient with respect to the possibility of crystalline defects within the new alloy, as well as the possibly temporary limitations in achievable composition, and can introduce flexibility with respect to any material property.

The software was developed in a general manner. The generalization allows any of the four devices and others to be loaded by only changing the initial structure definition file. Subsequent calculations and visualizations do not require additional changes due to their generalized implementation with respect to the number of subcells or segments in the device. This supports maintainability and the possibility of further development of the code base, since developments need not be made on any device individually. This applies to command file definitions for the device calculation and data visualization modules. The structure can facilitate the development of additional devices with similar planar heterostructure layouts, with an arbitrary number of layers, enabling further development and exploration into higher-order multi-junction solar cell designs, such as those applying the segmentation design strategy [4]. The approach is documented here.

The second chapter of this document provides an overview of the physics important to the device design process. This includes descriptions of the parameters determining the movement of charge

carriers within the device, detailing the processes by which a solar cell produces power when absorbing incident sunlight. The third chapter outlines many of the complex material parameters of silicon-germanium-tin reported in the literature, and their bearing on the multi-junction solar cell design possibilities. In line with this analysis, these material parameters, including the band structure, band offset, and effective mass models, are implemented in the manner required by the TCAD software. In the fourth chapter, the command implementation of the other modules in the simulation toolflow is described. The project is structured to allow for the exploration of the segmented design approach, which can compound the number of layers in the device. This impacts the structure of every module of the project. Additionally, tools for defining and adapting the finite-element mesh over the device are described. In the fifth chapter, the output of the software is explored. This includes reports of the optimized device designs, and discussion and mitigation strategies regarding sources of error. Chapter 6 contains concluding remarks about the four designs, the developed meshing strategy and software architecture, and describes material quality and thickness the requirements for SiGeSn growth.

2: PHOTOVOLTAIC DEVICE MECHANICS

2-1: ENERGY LEVELS

The conductivity of any material depends on whether its electrons can move. This is governed by the quantum mechanics inside the material. In a hydrogen atom, one electron buzzes around a tiny nucleus. Just like the harmonic modes of a guitar string, it can only buzz in ways that correspond to how it is contained. A brass instrument can play a few notes without pressing any keys, but it cannot play every note. In the same way, electrons in materials experience gaps, which are ranges of energy that they cannot play in.

An electron can drop from an upper energy level to a lower one, and the transition creates a photon that travels away. The frequency and energy of the photon correspond to the energy difference of the transition. Electrons moving in larger scale antennas also create photons, which are electro-magnetic waves, created when charge moves quickly. The allowed energy levels of electrons correspond to quantum states that the electron is allowed to have, in consequence of its surroundings. Just like the electron can drop from an upper energy level to a lower one, if the appropriate amount of energy can be given to the electron, it can move up from a low to a high state, perhaps by absorbing an incident photon with that energy. However, the electron cannot exist at an energy level in between the states, just as the notes available to a guitar string are constrained by its length.

The limitation on electrons to exist only in predefined states is observable in the light emitted by a hydrogen lamp. In this kind of lamp, voltage is applied to electrodes within a sealed tube containing the hydrogen gas. This imparts energy onto electrons within the hydrogen atoms, boosting them from the lowest energy level into the higher allowed states. Shortly thereafter, the electrons can fall back into lower available states. The light this creates consists primarily of a few specific wavelengths, called the hydrogen emission spectrum. The energy of a photon is related directly to its wavelength as described by the Planck-Einstein formula, $E = hf = hc/\lambda$, where h is Plank's constant, f is the photon's frequency, and λ is its wavelength. [5]

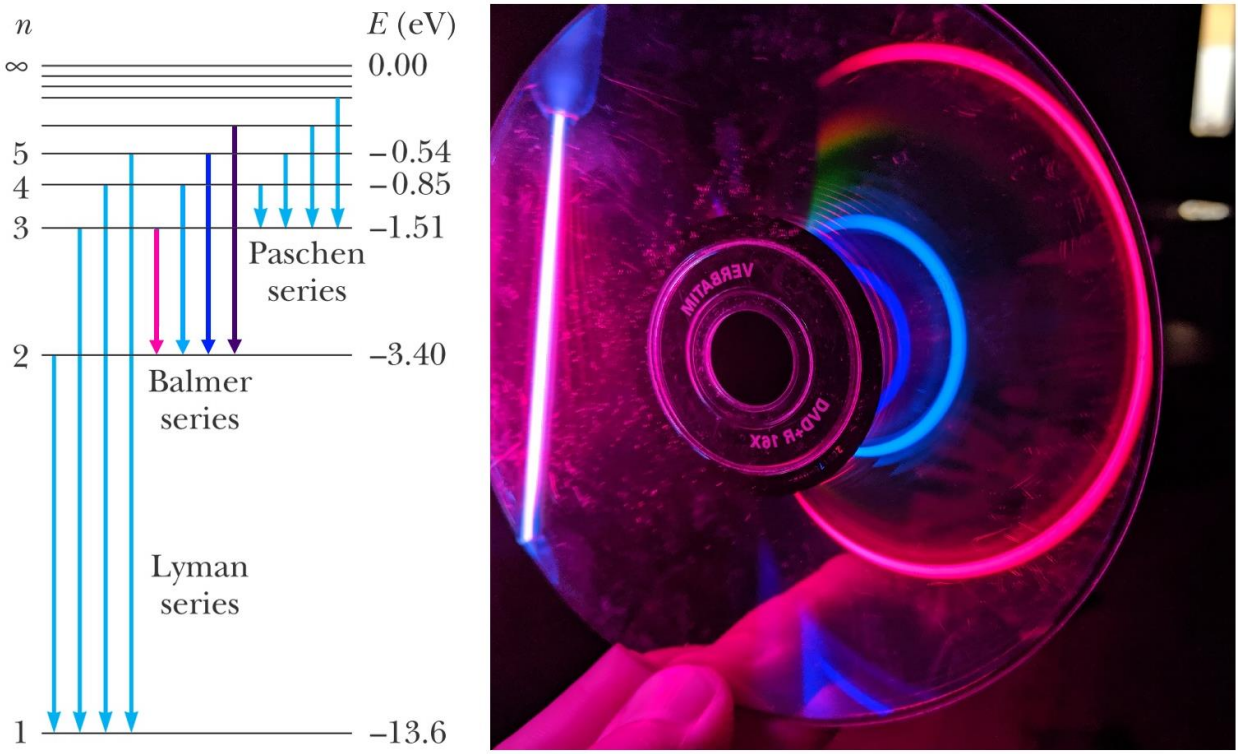

Figure 3 a) Energy-level diagram for hydrogen, colour added from b) to the appropriate transitions. [5] *b) Hydrogen lamp and its emission separated by wavelength after diffraction through grooves of DVD.* [6]

The hydrogen gas has singular, isolated states. However, in a crystalline material, with bonded atoms at regular intervals, the number of states increases, and they form bands of energies that are available to electrons. Electrons in the bonds of insulators take a large amount of energy to be freed from the bond, and this corresponds to the large energy gap between the valence band, where the electrons exist in covalent bonds between nuclei, and the conduction band, within which the electrons are free to move. In a semiconductor, the energy gap exists and is small relative to insulators, and it does take a significant amount of energy to free the electrons and create conductivity. In metals, energy states in which the electron is free to move exist at the same energy level as the bonded states, and no significant extra energy is required to enable electrical conductivity. [7]

The conducting states available to electrons in 3D space differ depending on the material, and present constraints on the electron's momentum and wavelength. The simplest transition, where an electron only requires energy, and not momentum, to shift into a conduction band wavefunction, is called **direct**. Semiconductors that have such a state as the lowest available conduction band state are also called direct, for example, gallium arsenide (GaAs). Other transitions require momentum differences in addition to the energy difference, such as along a diagonal or lateral direction within the lattice, and these are called **indirect** transitions. For a photon to create an electron-hole pair in an indirect semiconductor, it must interact with a valence band electron when a phonon of the right momentum is also present, resulting in a lower absorption probability and

higher penetration depth. This energy is much higher than the average temperature-related energy of phonons, lowering the likelihood of this photon absorption process to occur.

The conduction band states are higher potential energy states because they correspond to wavefunctions other than that of the covalent crystalline bonds. The states within the bonds are lower energy states, where the electron is as close as possible to nearby stationary ions of opposite charge that each exert a pulling Coulomb force on the electron, creating the lower potential energy area that the electron's wave function can occupy. The conduction band is composed of states away from these confined locations. An electron is free to move in the conduction band state because of the low occupancy of many spatially adjacent states. Electrons in valence band states are also able to move, but only when an adjacent state is vacant. This vacancy is called a hole, and the hole moves when an adjacent valence band electron moves into it. In this way, the motion of electrons in valence band states is appropriately described mathematically in terms of these holes and their motion. Effectively, a hole is a positively charged particle, representing the absence of a valence band electron. If a conduction band electron moves back into a hole (a vacant valence band state) the electron has lost its potential energy. In the transition from the higher energy wavefunction to the lower, a photon is released containing the difference in energy. This is the radiative recombination process, more likely in direct semiconductors, where the conduction and valence band states can be occupied with the same momentum. It is the inverse physical transition of a photon interacting with a valence band electron, imparting its energy to it, and causing a transition into a conduction band state. This interaction occurs as the electron is affected by and affects the electric field of the photon, creating a complementary traveling field in the transition between states that effectively destroys the photon as part of the absorption process.

2-2: Complex Refractive Index

Snell's law provides an equation that describes the bending or refracting, of light as it travels into and out of transparent media. The equation contains a coefficient, *n*, that is the real part of a material's refractive index, which describes the path of the light. [8] Concepts including the focal lengths of lenses, and total internal reflection, where light reflects off of the internal surface within a high *n* medium when incident at a nearly parallel angle, follow from this law.

The imaginary part of the complex refractive index contains another number, K, the extinction coefficient, that contains information about how the light's power is affected when traveling through the medium. Light of photon energy E and intensity I_0 attenuates exponentially within a medium according to the Beer-Lambert law [9],

$$I(E,x) = I_0(E)e^{-\alpha(E)x}$$

where x is the depth into the material, and the attenuation coefficient, α, relates directly to *K*, which is called the extinction coefficient, through [10]

$$\alpha = 2Kk_0$$

where k_0 is the photon wave number, the magnitude of its angular wave vector in vacuum. The wave number is related to the photon wavelength via $k_0 = 2\pi/\lambda_0$, which leads to

$$\alpha = \frac{4\pi K}{\lambda_0}$$

Alpha (α) has units of inverse length, and the reciprocal of α, which has units of length, provides a characteristic length scale with which one can estimate the penetration of the light. For example, if α is 0.1 μm^{-1}, then most of the light attenuates within 10 μm, and about 99% of it attenuates after 50 μm.

The complex permittivity of a material is related to the complex refractive index, N, by a complex square root function,

$$N = n - jK = \sqrt{\epsilon_{relative}}$$

where $\epsilon_{\text{relative}}$ is the complex relative permittivity of the material. [10] The absolute permittivity, ϵ, is related to the relative permittivity $\epsilon_{\text{relative}}$ by scaling with the permittivity of vacuum space, ϵ_0:

$$\epsilon = \epsilon_{relative}\epsilon_0$$

The imaginary parts generally must be negative, representing loss of energy of the propagating wave, and K is then expected to be positive by definition, and the imaginary part of the complex permittivity is defined similarly. Minus signs in the definitions allow the imaginary parts to be plotted and tabulated without the negative sign.

Materials with negative refractive index are called metamaterials and are a subject of modern, somewhat unrelated research, and we expect n, the real part of the square root of $\epsilon_{\text{relative}}$, to be positive. The real part of the permittivity it not so restricted. To visualize the complex square root function, it is best to consider the polar representation of the complex numbers. The magnitude of the number or vector is a real number, and the angle is also given. The square root function affects the magnitude as a real square root, and the angle is halved, according to $(Me^{i\Theta})^{1/2} = M^{1/2}e^{i\Theta/2}$. If the imaginary parts are plotted as positive, the permittivity can occupy the first or second quadrants, and the real refractive index will remain positive. The correct square root must be taken, given that two exist, and this is the one where the sign of the imaginary part is preserved. The other square root is found by interpreting Θ as negative.

Within a semiconductor, the light interacts with the matter by interacting with an electron. For example, a valence electron can be given the energy of the photon, the photon disappears, and the electron changes its state and wavefunction to a conduction band state. This relates the absorption coefficient with the states available to the electron, which are a function its wave vector, $\vec{k}$. Photons promote electrons to the conduction band as the light loses intensity. When in the conduction band, the electron is more free to move, since many conduction band states are unoccupied in adjacent areas. The electron leaves behind a hole in one of the bonds of the semiconductor, and the hole can also move, so it also contributes to the overall conductivity of the semiconductor.

2-3: MOBILITY

Usually, a hole has less mobility than electrons in the conduction band. The mobility of conducting electrons and holes varies greatly depending on the material and doping. The mobility is a

coefficient that can be used to calculate the effective drift velocity of charge carriers within a given applied electric field.

The mobility of charge carriers, (i.e. electrons or holes,) also manifests in their diffusion behaviour. Diffusion occurs when a large group of randomly vibrating particles have inconsistent concentration. The concept applies to macroscopic gases, as well as to concentrations of electrons and holes within a semiconductor. The Einstein relation shows the dependence of the diffusion coefficient, *D*, on carrier mobility. [11] The mobility is, therefore, relevant whether the carrier is moving around randomly as it does in diffusion, or whether it is undertaking guided motion due to an electric field, (drifting.)

2-4: DOPING

This term 'intrinsic' refers to semiconductors that haven't been doped, and doping is the introduction of another element into the crystalline structure of the semiconductor. This affects the properties of the material. For example, it can improve the conductivity by supplying conducting electrons or holes at room temperature. In this way, semiconductors have relatively poor intrinsic conductivity, but are more conductive when doped. The small amount of conductivity that semiconductors do intrinsically provide is largely caused by electrons excited into the conduction band due to the temperature-related energy that they have. The energy gap between the valence and conduction band is large compared to the energy of room temperature, and it is a small proportion of electrons that have sufficient energy to escape the bond and conduct. Consequently, the conductivity of an intrinsic semiconductor depends on the temperature.

An intrinsic semiconductor has no extra electrons apart from those that come from bonds. Each silicon atom has four valence electrons that form bonds with neighboring silicon atoms to form a crystalline matrix. Just like carbon or germanium, this occurs because it is in the fourth group of the periodic table. Tin can make this arrangement as well, as it does in SiGeSn, though pure tin usually takes on a hexagonal arrangement. Carbon has a very small interatomic spacing, and tin is the largest of those four. Gallium arsenide (GaAs) exhibits a similar crystalline structure, called diamond cubic, and it is an example of a III-V semiconductor, while the previous materials and alloys of them are group IV semiconductors. Gallium contributes three valence electrons and arsenic contributes five, and in GaAs, the two substances are present equally, resulting in a similar arrangement of electrons in bonds. That is, four bonds adjacent to each atom, and each bond having two electrons.

Typical dopant elements are those from group III or V, and they are introduced into the crystalline matrix. A group V dopant has five valence electrons. Four of them become parts of bonds, and the fifth is weakly attached to the site of the dopant atom. Near 0 Kelvin, this extra electron still cannot contribute to conductivity, just like the valence electron in an intrinsic semiconductor. However, at normal operating temperatures, electrons from group V dopants have much higher availability than valence band, bonded electrons. The background energy from temperature is generally enough to free these electrons completely from the dopant sites. At medium temperatures, the concentration of conducting electrons is approximately equal to the concentration of the group V dopant.

A group III dopant has only three electrons, and therefore produces a hole in one of its bonds. It similarly takes minimal energy for an electron from an adjacent bond to fill that hole, and for the hole to escape from the dopant and begin freely contributing to electrical conductivity. This is called *p*-type doping, since it provides holes, which can be thought of as positive charge carriers. The equations of semiconductor devices deal largely with the dynamic concentrations of conducting electrons and holes, which produce current when they move. They can have drift motion caused by an electric field, or diffusion motion, which occurs due to random particle motion when there is a gradient in the concentration level.

2-5: Example Device Structure

Figure 4 shows a cross section of a portion of a single-junction device. The portion selected is a symmetry element used by the simulation. [12] The symmetry is by mirror image. Extending symmetrically to the right by 60 additional micrometers, the next contact, or grid finger, will be reached. Only half of the width of the cap and cathode is shown on the left, as the mirror symmetry creates the other half. The interface between *p*- and *n*-type layers has characteristics that allow for the collection of electrons freed by and holes created by incident light. The cathode and anode only exist as boundaries in the figure and in the simulation.

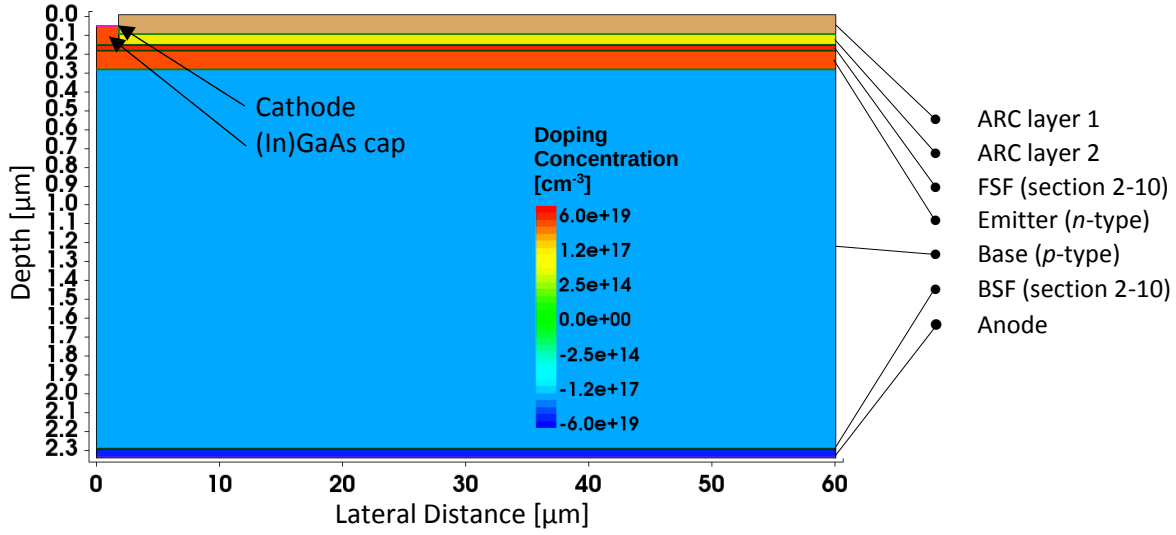

Figure 4: Symmetry element of a single p-n junction device structure with names of regions.

Figures 5 and 6 show the geometry of the other dimension of the device.

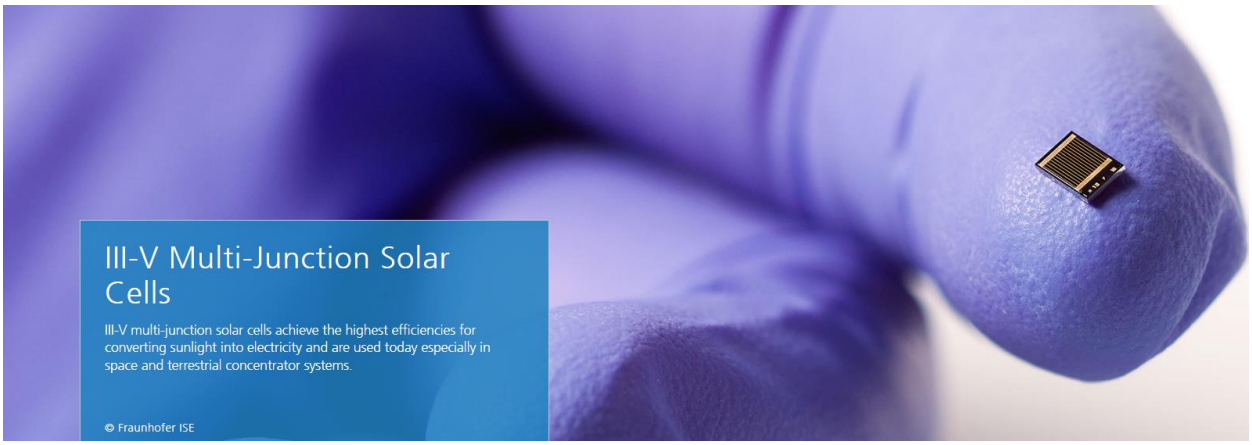

Figure 5: Picture of a III-V multijunction solar cell at the Fraunhofer Institute for Solar Energy Systems. [13]

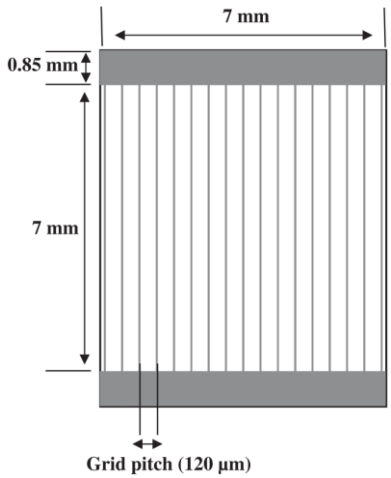

Figure 6: A schematic for the top electrode of concentrator solar cell. [56]

The resistive loss in the grid fingers is not calculated here, but is an important power loss mechanism in concentrator photovoltaics. It typically produces the primary limitation on the concentration factor of the light. However, if the segmentation design strategy is applied, the device will produce higher voltage in exchange for lower current, for similar device power. This reduces the series resistive power loss in the grid fingers, since the reduced current reduces $P = I^2R$. In this way, a segmented device allows higher concentration to be utilized in the system, which benefits the efficiency of the solar cell and the system.

Regarding the effects neglected with respect to this physical dimension by assuming the symmetry element of Figure 4, many are beneficial, since they comprise of the carriers taking the path of least resistance. At the edge, charge carriers can travel directly to the pad on the side. Additionally, the carriers may travel off-perpendicular to the grid fingers, taking a slightly angled path of less resistance. However, these effects are negligible due to the high conductivity of the grid fingers. If the grid fingers are removed, the fill factor of the device suffers greatly to the lateral resistance within the device at high concentration. The resistive loss of the grid fingers is not being calculated, and these beneficial effects would constitute a slight reduction of that loss mechanism.

2-6: *P-N* JUNCTION OPTOELECTRONICS

When an *n*-doped region and *p*-doped region are adjacent, the majority carriers diffuse into each other, and form a "depletion region," where conduction band electrons have filled the holes, and the concentrations of both are very low. In this region, also called the space-charge region, an electric field develops, since the *p*-dopants acquire full bonds, and the *n*-dopants have lost their extra conducting electrons, and there is an extremely low number of conducting electrons and holes. The extent of the depletion region depends on the temperature, the dopant concentrations, and the permittivity inside the semiconductor.

When reverse biased, a small current flows, called the reverse saturation current, since the field at the junction only blocks majority carriers, and the small, often negligible amounts of minority carriers present flow freely through the junction. The applied bias causes the depletion region to widen.

As the junction is forward biased, the depletion width reduces. In digital circuitry, diodes are made from p-n junctions, and can be thought of as gates that allow current in one direction and not in the other. The p-n junction allows current to pass when the forward bias level is sufficient to reduce the depletion width completely. In this situation, since the bias level is far above the turn-on level, the diode does not limit the current. However, the p-n junction current-voltage characteristic curve is important to the operation of a solar cell. In the dark, the forward biased p-n junction blocks most current until the depletion region is reduced to nothing. Exponential expressions follow from these processes. While the depletion region is large, an ideality factor of 2 may be observed in a logarithmic plot of the small allowed current, and when it is smaller, processes leading to an ideality factor of 1 may dominate, and in the intermediate region, an ideality factor between 1 and 2 may be perceived.

When photons are incident, they can interact with valence electrons and impart all of their energy to them. An ultraviolet photon might impart energy to the electron in excess of the bandgap. The extra energy is quickly lost in a process called thermalization, and the electron falls to a low energy state in the conduction band through a rapid series of lossy interactions with the lattice, creating heat energy. Photons can also interact with electrons already in the conduction band, which is not desirable, since all of that photon's energy will likely be lost to thermalization.

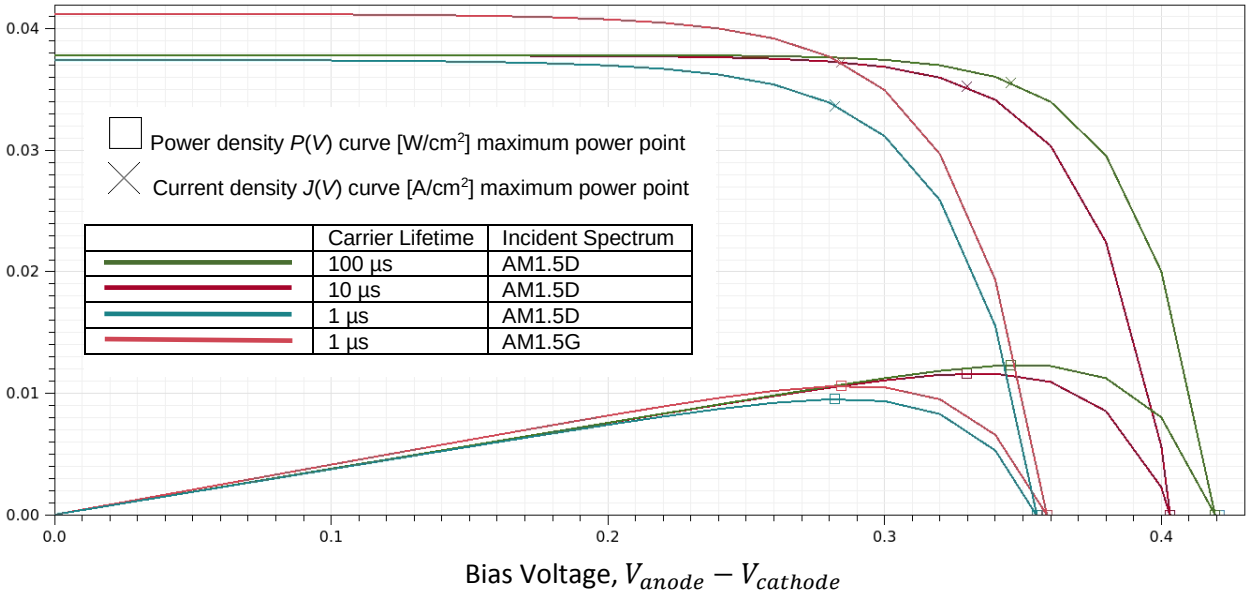

Figure 7: Example $J(V)$ and $P(V)$ characteristics from the simulated structure of Figure 4.

The desired outcome is that an electron-hole pair is created in or near the depletion region. Inside the depletion region, an electric field is present, which separates the electrons and holes, and creates most of the current. If an absorption event occurs outside of the depletion region, there is no field present, and the two particles begin to contribute to diffusion motion. The diffusion effect

after pair generation is to push carriers in both directions as a net effect of their random motion and the created concentration gradient. The desired direction for minority carriers is toward the depletion region, and for majority carriers it is to move to the outside of the cell. Regarding motion in the undesirable direction, minority carriers should be contained by surface layers that are engineered to prevent them from reaching the metallic contact, where they would lose their potential energy. Majority carriers diffusing in the wrong direction are contained by the field of the depletion region. As such, even pair generation away from the junction promotes the separation of the charge carriers, since the minority carriers are bounded by a barrier on one side and an accelerating field on the other side. Likewise, majority carriers are bounded by the built-in field on one side and will therefore discharge via the external load, given that their concentration has been increased by the photogeneration and by the influx from across the junction.

The n region of the device is called the emitter, since it will emit electrons, and the p side is called the base. When the emitter is above the base, it is more highly doped, and this allows this layer to be much thinner and more transparent, allowing more photons to pass. Lower doping in the base layer extends the depletion region to a larger depth. The base layer is also called an absorber layer, since this is the layer that is engineered to be large, which gives light more opportunity to have absorbing interactions that create electron-hole pairs. The direction of the field in the depletion region pushes carriers towards the layer where they are the majority carrier, electrons into the emitter, and holes towards the p side. The extended penetration of the field in the absorber region helps encourage electron-hole pairs to be separated, over a distance long enough to allow the light to interact, within a region where recombination is less likely. Electron-hole pair creation outside of the field can contribute to diffusion that promotes motion of majority carriers towards the contact and minority carriers towards the depletion region field. High mobility and diffusion length is desirable for minority carriers in an absorber region, since it extends the length over which they can diffuse into the depletion region.

A solar cell under operating conditions is connected to the equivalent of a load resistance. As the current flows through the resistance, a forward biasing voltage appears in counteraction. An operational design is optimized when the load is near the "turn-on" point, comparable to the built-in voltage of the diode. Open circuit conditions correspond to the highest concentrations of majority carriers, and occur when the load resistance is extremely high. The optimal design usually maximizes the power produced by the solar cell, which corresponds to a specific equivalent load resistance. The current direction is to come out of the p side of the diode. If the p-n junction is forward biased with a voltage source of higher voltage, it can enter the diode-on regime, where the current direction is opposite. When both regimes are under consideration, the solar-cell operation current is considered negative, and is placed in the fourth quadrant of the plot. If the forward bias applied to the diode is great enough to exceed the open-circuit voltage of the solar cell, the diode current then becomes greater than the countering photocurrent, corresponding to the first quadrant of the plot.

The forward bias current being linearly additive with the photocurrent, opposite in direction, is interesting. It's possible that the reason that superposition is applicable takes its roots in Newton's third law being applicable to the moving electrons. The resistance forces presented by the load, that imply the forward bias, exist only as a consequence of the push forces that create the

photocurrent. The forward bias forces are co-linear and directly oppose the forces of the carriers caused by the junction field acting through the resistance, and the vector addition amounts to a scalar reduction of the effects that cause the diode characteristics. A series-connected voltage source would apply a similar constraint as a load resistance with equal voltage drop, as in any circuit. In this way, the effects contributed by the forward bias are totally linked to the effects of the photocurrent in an additive fashion: as a macroscopic effect that is the sum total of microscopic effects, all of which are subject to Newton's third law. As such, the diode effect is a consequence of Newton's third law with respect to the photocurrent effect, and the net effect of both must be the summation of the two. This may serve as an explanation for why the photocurrent can be added to the characteristics of a diode in the dark to achieve the net effect.

The open circuit voltage is causally secondary to the short circuit current. If the short circuit current is improved by increasing the illumination, the diode component to the cell's current-voltage characteristic remains identical, and a logarithmic improvement to the open circuit voltage is implied and observed.

2-7: Fill Factor

The concept of the fill factor numerically connects the short circuit current and open circuit voltage to the operational power output of the device. Multiplied together, I_{sc} and V_{oc} produce a fictitious quantity of power. One can imagine a rectangle on an I-V plot bounded by these quantities. The actual maximum power point occurs at a point receded from the fictional power point at the corner of that rectangle. The fill factor is the ratio of the actually accessible maximum power point to this fictional, larger quantity. The fill factor can indicate the magnitude of many sources of performance loss, and serve as a metric for the effectiveness of many processes within the cell, however, the plot of the device's I-V curve contains more information.

2-8: Minority Carrier Injection

At equilibrium with no applied bias, the depletion region is formed as the freely moving (diffusing) conduction-band electrons of the n side fill the nearby holes of the neighboring p side. Steady state is reached when the diffusion tendencies of the two groups of carriers are counteracted by the electric field that is formed by the departure of carriers across the junction. Under forward bias, the majority carriers of one side can bypass the weakened electric field by diffusion and establish some presence on the opposite side, as part of the process of reducing the field. The concentration gradient is reduced to the point where the diffusion tendency is balanced again with the modified field. Therefore, there exists an above-normal minority carrier concentration near a p-n junction under forward bias.

2-9: Recombination

Carriers existing in excess of equilibrium levels will tend to fall back into the equilibrium state. This corresponds to energetic electrons falling back into the valence band, filling holes. This is not desirable in a solar cell, which aims to push conduction band electrons and valence band holes apart as they are generated by incident light, and have them maintain that potential energy as much as possible, such that they will be attracted through the load in the external circuit to finally meet back with the opposite type. Recombination is a manner in which these carriers can find a partner

without having to travel through the load. It counteracts the initial separation of the charge carriers created by the incident light, and constitutes a loss of potential energy.

Notice how recombination via the load is desirable. The field created by the *p*-*n* junction pushes electrons into the cathode and holes into the anode. The direction of electron flow is clear at the cathode. At the anode, hole flow towards the contact corresponds to valence band electrons traveling away from the contact. In this manner, the *p*-*n* junction is pushing conduction band electrons through the circuit via the anode, while pulling the valence band electrons through the circuit via the anode. So, ignoring the hole concept for a moment, electrons are pushed and pulled the same way around the circuit on both sides of the *p*-*n* junction. Returning to holes, as they reach the anode, they do not continue to exist within the metal circuit. What is happening is that electrons are arriving via the metal and entering the valence band of the semiconductor, at a rate equal to the rate at which the holes arrive at the contact. This is a kind of recombination in that the electrons from the metal are entering the valence band holes, however, they are doing so via the external circuit, after having lost their potential energy in that desirable manner.

The way the *p*-*n* junction has influence at a distance away from its location is via diffusion. Locally, the *p*-*n* junction is sorting the new charge carriers, constantly placing holes on one side and electrons on the other side. This produces a build-up and increased concentration of both as long as the light is incident. The increased concentration at steady state encourages the carriers through the load due to the diffusion effect. The field of the *p*-*n* junction holds the majority carriers at bay, so their only option is to collect until they reach a certain voltage and begin discharging through the load. This is how the carriers are encouraged to move through the circuit with a certain degree of force originating at the electric field of the *p*-*n* junction where electron-hole pairs generated by incident photons are separated.

There are a few processes by which recombination can occur within the semiconductor region. The rates of these processes can be modelled and estimated based on the concentrations of both types of carriers and parameters of the material. Understanding these processes is useful while defining the device and aiming to improve its performance.

When a conducting electron arrives at a hole and falls in, it is called radiative recombination, since a photon is produced when an electron changes state in this manner. Auger recombination occurs when two conducting electrons collide. One loses its energy in the collision, and the other gains excessive energy that it quickly loses again to thermalization. A third process is called Shockley-Read-Hall (SRH) or trap-assisted recombination. If a defect exists in the semiconductor's crystalline lattice, such as a contaminant at one lattice point or geometric imperfection, the usual band structure can be affected, creating a quantum state within the bandgap associated with the defect that an electron or hole can occupy. A charge carrier arriving at that location will likely fall to the lower available energy state, like falling into a trap, and can subsequently fall the rest of the way when a carrier of the opposite type arrives. Without the interaction with that trap, the carriers would have remained in their excited state. [11]

The photon emitted by radiative recombination can be re-absorbed before it leaves the device, and this is called photon recycling, or also called luminescent coupling if occurring in a multijunction device, [14] so radiative recombination is slightly preferable to the other recombination processes.

In indirect materials, the holes and electrons usually have different wave vectors and radiative recombination can be unlikely and unimportant, [15] however it is a significant effect in direct semiconductors. Auger recombination can be a dominant process when carrier concentrations are high. SRH recombination is usually relevant, and solar cells are known to benefit from excellent crystalline quality. This kind of recombination reduces the fill factor and V_{oc}.

The net recombination time, the average amount of time that a minority carrier exists before recombining, depends on the rates of all these processes, according to $1/\tau_{net} = \text{sum}(1/\tau)$ from each process. [16] Often, one process has the shortest expected timeframe and dominates, having the strongest effect on the minority carrier lifetime. The length, L, that a generated minority carrier can be expected to travel depends on the diffusion coefficient, D, and the carrier lifetime, τ_{net}. This diffusion length is an important parameter in the design of solar cells, potentially limiting the thickness of certain layers, especially if the light is expected to penetrate deeply before being absorbed, or if recombination is high. [15][9]

$$L = \sqrt{D\tau_{\text{net}}}$$

Surface recombination can be classified as trap-assisted recombination. At the surfaces of the solar cell, perhaps less so at surfaces between layers of the solar cell, and more so at connections to metal and other outer edges, the regular crystalline structure is disrupted. Quantum mechanically, this results in allowed states for the carriers at many other points in the band structure, due to the high degree of tiny irregularities present at the surface, and can greatly increase recombination. The reduction in carrier concentrations near the surface results in diffusion current towards the surface, and recombination continues. Mathematically, this results in a coefficient with units of velocity, called the surface recombination velocity (SRV), that quantifies the quality of a surface and its tendency to facilitate the loss of mobile charge carriers. An excellent quality surface could have a surface recombination velocity of 10 cm/s, and value of 10^7 would be appropriate for a fairly low quality surface.

2-10: Surface Field Layers

Front surface and back surface fields (FSF, BSF,) are staple concepts in solar cell design, due in part to the increased recombination observed at the surfaces of a solar cell. Minority carriers need to get to across the junction to help generate power, and they are not useful near the edges. In fact, their presence is required for recombination processes at the surface. Their only usefulness is the possibility that they may diffuse all the way to the junction. These are the reasons to engineer the surface layers of the solar cell. This can be done by altering the doping or the material. Ideally, the characteristics of the surface layer affect the minority carriers and not the majority carriers. On the *n* side, this means that the valence band energy level should be affected and not the conduction band energy level. Majority carrier energy level differences would constitute blocking the number of carriers that can pass, or allowing them to lose some potential energy to thermalization. The BSF dissuades the presence of minority carriers, reflecting them towards the junction if they approach, such that they are not present near the back contact, and recombination there is reduced. The front surface field layer, which is also called a window layer, accomplishes a similar goal, repelling the minority carriers, reducing surface recombination, and it must also be transparent to the incoming light. It can offer some physical protection to the internal layers of the solar cell,

where material quality is important to maintain a long diffusion length. In a multi-junction solar cell, surface recombination is possible at the edge of a subcell component, and the surface field layer repels minority carriers, reducing that possibility, but the interface between the layers must be of good quality, since surface recombination can also occur at the boundary surface field layer where minority carriers are present.

2-11: OHMIC CONTACTS

When metal is connected to a semiconductor, it can create a Schottky diode or an Ohmic contact, depending on how highly doped the semiconductor is near the interface. High performance Ohmic contacts are desired for solar cell contacts. Very high doping at the edge accomplishes this. Diode behavior is observed at lower doping levels, which can be useful in other applications. [15]

All these components and concepts cover most of the important basic aspects of a single junction solar cell, such as the two surface field layers on the outside, the emitter and base layers that form the p-n junction inside, and the connection to the metal.

2-12: MULTI-JUNCTION SOLAR CELLS

Sunlight contains many photons that have a very broad range of energies. If a high energy photon is absorbed by a low bandgap solar cell, much of the energy does not become available as power to the load. If one uses a high bandgap solar cell to efficiently convert the high energy photons, then the lower energy photons are not captured at all. A multi-junction solar cell combines high-bandgap and low-bandgap solar cells to more efficiently collect all of the photons. The high-bandgap solar cell is placed on the top. It collects the high energy photons, and it is transparent to the lower energy photons, which pass through it, to be collected by a lower bandgap cell beneath it.

There are a few ways to accomplish this. The ‘subcells’ can be made separately and simply mounted on top of each other, with a small separation. An electrical terminal between them can be made available. However, it can be more cost-effective to design the stacked subcells into a singular device, grown on top of each other on the same substrate. Even within this design space, there are different avenues available, and technologies to explore. With no terminal available in between subcells, when they are connected to a loading circuit, their stacking leads to a series connection. It follows that each subcell must produce the same current.

In between subcells within a device, a tunnel junction must be utilized. Without it, the *p*-region from one cell would connect to the *n*-region of the next cell, and create a *p*-*n* junction in the opposite orientation to the one within the subcell, which would create a blocking diode. A tunnel junction consists of a pair of highly doped *p*- and *n*- layers, placed between adjacent subcells, where each subcell has the four components of Figure 4: emitter, base, FSF, and BSF. The highly doped layers of the tunnel junction create a depletion width narrow enough for electrons to bypass it via the quantum-mechanical tunneling effect, which follows from the wave-like nature of tiny particles. The majority carriers from each subcell flow into each other via the tunnel junction. The conduction band electrons from the n-doped side of one subcell tunnel and flow into the valence band of the p-doped side of the next subcell, as is necessary for holes to flow on that side, since hole flow corresponds inherently to valence-band electron flow. Alternatively, the process can be

described as holes tunneling from the p-side valence band into the conduction band of the n-side, corresponding to those electrons departing the solar cell as current. Given that minority carrier concentration is minimal at the edges of the subcells, the total current is comprised primarily of majority carrier current. As such, the net total current is the same throughout every layer of the device, whether it is caused by electrons or holes.

Since carriers are flowing from the conduction band of one subcell to the valence band of the next subcell with a small voltage drop related to the tunnel junction characteristics, the energy levels of these two locations become nearly equal. In this way, the voltages produced by subcells are additive as in a series connection.

If subcells are mechanically stacked, and a terminal is made available in between the subcells, then the subcells do not need to be series-connected, since the extra terminal(s) can be utilized to optimally bias the different sections of the stack. [17] However, in the case of this thesis, the focus is on lattice-matched epitaxy, and only two terminals are available. In this configuration, every subcell should produce precisely the same current. Due to the series connection, and Kirchhoff's Current Law, which follows from the conservation of the number of electrons throughout the device in steady state, the subcell that produces the least current will limit the other subcells to that level. This is somewhat of an approximation, as it does not account for luminescent coupling, from which the limiting subcell will benefit the most, since the excess carriers in the other subcells lead to additional radiative recombination. [14] Ultimately, significant energy is lost according to the degree to which the set of subcells do not produce the same current.

The amount of current produced by a subcell depends upon the number of photons incident upon it, and follows from all of the physical processes described previously. This includes the absorption profile of the materials. A direct bandgap material has a steep absorption profile, meaning that, photons with energies even slightly above the bandgap will be absorbed without penetrating too deeply. A weaker absorbing material, such as an indirect bandgap semiconductor, will have photons penetrate more deeply, and many could pass completely through. If the photon is absorbed far from the depletion area by the junction, then the diffusion mechanics become important. The carrier recombination processes also have an important effect on the probability that the solar cell will convert the photon into electrical power. The current finally produced by the operational subcell depends on all these processes.

As an approximation, it is possible to assume that each subcell will absorb its share of photons above its bandgap, and below the bandgap of the subcell above it. This assumption corresponds to assuming a full quantum efficiency of 1, and a step absorption function. The actual deviation from the assumption can be calculated. It's often a useful assumption for making estimates, or to discuss the design space, but an optimized multi-junction solar cell should rely upon more precise calculations. The equations are not analytically solvable, and simulation software is routinely used to enable the design of these devices.

2-13: STANDARD SPECTRA

The number of photons available for the cell to convert depends on the solar spectrum incident on the device. Not only are the physics within the device fairly complex, but the content profile of the solar spectrum is an important constraint on the design of a multi-junction solar cell (MJSC.) There

are a few standard spectra that are often used for consistency in the field, such as AM0, for space, AM1.5D, for direct, and AM1.5G, for global. AM is air mass, and the number, i.e. 0, or 1.5, stands for the number of atmospheric distances that the sunlight has passed through, where 1.0 corresponds to the amount of atmospheric distance or mass that the light passes through from a sun that is 90 degrees, directly overhead, which is unusual. 1.5 corresponds to a more commonly encountered, or closer to average, spectrum, where the sun is shining from 48^{o}, and passes through that much more atmospheric mass than if it were 90^{o} overhead. The global term refers to the light that can come from all angles of the sky, and the direct data shows only the component that comes directly from the sun. [18] The spectra are a part of a standard developed by the American Society for Testing and Measurement (ASTM), called ASTM G173, where this "G" is the identifier for an ASTM radiometry subcommittee. These standard spectra are shown in Figure 8.

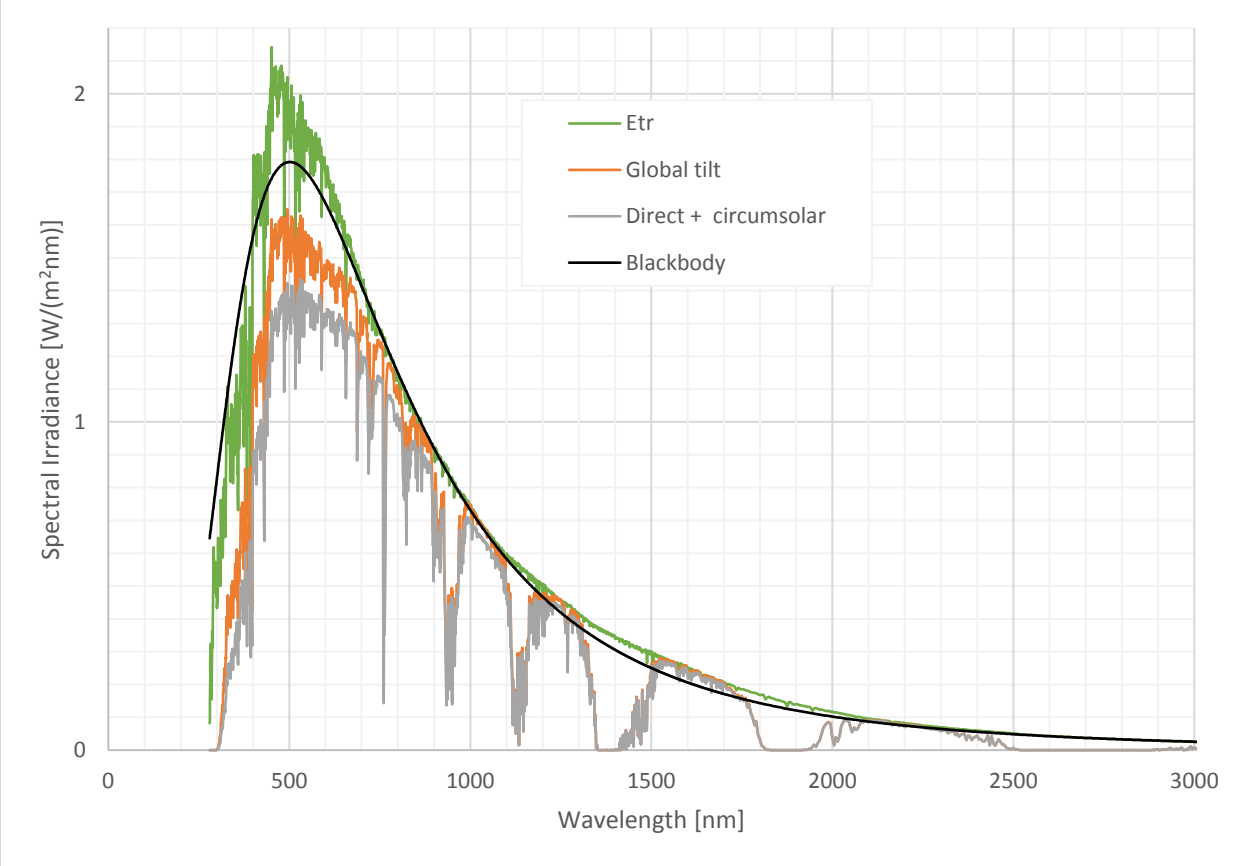

Figure 8: AM0, AM1.5D, and AM1.5G standard data, [19], [20]*, with the result of the Planck blackbody irradiance calculation of the sun's energy radiance onto Earth added.*

The AM0 measurements lack certain spectral power deficits present in the AM1.5D and AM1.5G datasets. These deficits are caused by atmospheric absorption. They are not considered in the black body radiation model. The black body radiation spectral distribution was determined first by Max Planck, which approximately matches the measurements, as shown. The deviation is from less-dominant physical processes, such as from the non-uniformities throughout the sun, in temperature, composition, and physical processes, as is the case in the photosphere.

2-14: BLACK BODY RADIATION

A black body is an object made of material that is free to resonate at any frequency. This property makes it a black object because it will absorb all incoming frequencies of light, while colored objects are reflecting certain frequencies of light. The term is important in the inverse situation, when the object is heated and begins to emit light. It is an ideal concept of an object, that does not indicate any preference or constraint with respect to the vibration and resonance spectrum that occurs when heat energy is added to it. This property is an idealistic assumption from which the study of the phenomenon can proceed. Non-black non-ideal objects can resonate similarly, however, the resulting frequency will deviate from this result. Therefore, this theory is applicable to many non-ideal objects, which will emit a similar spectrum when heated, such as a hot stove element, or a flame. The black body problem begins from the observation that nearly any object emits a similar color when the temperature is raised, and the problem was to determine the spectrum that is emitted.

Planck's contribution to this problem contained one of the earliest discoveries in quantum mechanics. Even though the object does not have any inherent constraint on the frequencies at which it can resonate, Planck supposed that there was a quantum constraint on the emission.

The resonated spectrum is caused by the atoms transitioning between the states that are possible for them to occupy, while following the constraints of quantum mechanics. Planck's work involves the introduction of the assumption that only specific energy levels are possible, according to $E = nhf$, where n is any integer, h is Planck's constant, and f is the resonator frequency, which becomes the emitted photon frequency. This restricts the spectrum at high frequencies, where the temperature-related energy, averaging at $k_B T$, becomes too low to attain the energy transition of magnitude hf. Without this restriction, the temperature energy could create a contribution to that part of the spectrum, contributed to in small part by all of the resonators that would be free to produce light even at those high frequencies. The average kinetic energy of the atoms in the system at $T = 5778\ \text{K}$ is $k_B T = 8.617 * 10^{-5}\ \text{eV/K} * 5778\ \text{K} = \mathbf{0.5\ eV}$. The energy of a photon at $500\ \text{nm}$ is $E = hc/\lambda = 1240\ \text{eV} \cdot \text{nm} / 500\ \text{nm} = \mathbf{2.48\ eV}$. According to Boltzmann statistics, this makes the states at wavelengths lower than $500\ \text{nm}$, (energies above $2.48\ \text{eV}$), become exponentially less likely to be attained by the atoms in the system of the given temperature. This is how quantum mechanical proposition that the energy states available to resonators being restricted to $E = nhf$ dampens the spectrum on the ultraviolet side.

The earlier purely electromagnetic theory allows for electromagnetic waves of any energy. Since light is composed of photons, the energy of a photon must be attained by a resonator in order to create light at the given frequency. On the ultraviolet side of the spectrum, photon creation is restricted because the temperature-related energy becomes insufficient for the atoms to reach the necessary energy level to create photons. Without the assumption $E = nhf$, this black body emission expression could not be derived. Einstein expanded upon the work of Plank to discover that light itself is quantized and is composed of quantities of the elementary light particle, the photon. [5]

The spectral emission quantities can take many forms: as a function of frequency, wavelength, or energy. Here, the wavelength spectrum is applied. The energy density in a black body cavity per

wavelength band, $u(\lambda, T)$, is distributed over the wavelength range λ, and is a function of temperature, T, given by

$$u(\lambda, T) = \frac{8\pi hc}{\lambda^5 \left(e^{\frac{hc}{\lambda k_B T}} - 1 \right)} \left[\frac{\mathrm{J}}{\mathrm{m^3 m}} \right]$$

where h is Plank's constant, k_B is Boltzmann's constant, and c is the speed of light.

This is related to the total power radiated by the black body per emitter surface area per wavelength band, [5]

$$J(\lambda, T) = u(\lambda, T) * \frac{c}{4} = \frac{2\pi hc^2}{\lambda^5 \left(e^{\frac{hc}{\lambda k_B T}} - 1 \right)} \left[\frac{\mathrm{W}}{\mathrm{m^2 m}} \right]$$

To apply this model to the sun's radiation on Earth, note that the total power amount is uniformly emitted spherically out from the sun. To achieve the irradiance on the Earth in the international system of units, consider an area of $1\ \mathrm{m^2}$ at the Earth's distance from the sun, R_{orbit}. Incidentally, the Earth's radius is negligible compared to the orbit radius. Likewise, the Earth's elliptical orbit's aspect ratio of 0.9966 is of little consequence here. The proportion of power reaching $1\ \mathrm{m^2}$ is the ratio of that $1\ \mathrm{m^2}$ to the area of the entire sphere of radius R_{orbit}. The area of any sphere is $4\pi r^2$, where r is the radius of the sphere. The quantity $J(\lambda, T)$ becomes spectral power when multiplied by the sun's surface area. The resulting unit is $[\mathrm{W/m}]$ if $1\ \mathrm{m^2}$ is assumed, or $[\mathrm{W/m^2 m}] = [\mathrm{W/m^3}]$, which can be multiplied with other irradiated area sizes, and the wavelength unit is finally converted to nm for $[\mathrm{W/m^2 nm}]$.

Irradiance $=$

$$J(\lambda, T) * 4\pi R_{sun}^2 (1\mathrm{m^2}) / 4\pi R_{orbit}^2$$

$$= J(\lambda, T) \frac{R_{sun}^2}{R_{orbit}^2} \left[\frac{\mathrm{W}}{\mathrm{m^2 m}} \right]$$

The temperature of $5778\ K$ is used as the temperature of the sun. Input values are shown in Table 1, an excerpt of the calculated data table is shown in Table 2, and the result is shown in Figure 8.

Table 1: Input Values for Solar Black Body Radiation Calculation.

h	4.136E-15	eV·s
c	2.998E+08	m/s
k_B	1.381E-23	J/K
T_{sun}	5.778E+03	K
R_{orbit}	1.496E+11	m
R_{Earth}	6.371E+09	m
R_{sun}	6.957E+08	m
R_{orbit}-R_{sun}	1.489E+11	m

Table 2: Solar Black Body Radiation Calculation.

Wavelength	λ		2.80E+02 2.80E-01 2.80E-07	2.81E+02 2.81E-01 2.81E-07	… … …	5.00E+02 5.00E-01 5.00E-07	nm µm m
Etr			**8.20E-02**	**9.90E-02**	…	**1.25E+00**	$\frac{W}{m^2nm}$
Global tilt			**4.73E-23**	**1.23E-21**	…	**7.97E-01**	$\frac{W}{m^2nm}$
Direct + circumsolar			**2.54E-26**	**1.09E-24**	…	**5.85E-01**	$\frac{W}{m^2nm}$
Energy	E	$\frac{hc}{\lambda}$	4.43E+00	4.41E+00	…	2.48E+00	eV
Black Body Energy Density (per volume) per wavelength band	$u(\lambda,T)$	$\frac{8\pi hc}{\lambda^5 \exp(hc/\lambda kT)-1}$	3.98E+05	4.04E+05	…	1.11E+06	$\frac{J}{m^3m}$
Black Body Power Radiated per emitter area per wavelength band	$J(\lambda,T)$	$\frac{u(\lambda,T)c}{4}$	2.99E+13	3.03E+13	…	8.29E+13	$\frac{W}{m^2m}$
Calculated Irradiance		$\frac{JR_{sun}^2}{R_{orbit}^2}$	6.46E+08	6.55E+08	…	1.79E+09	$\frac{W}{m^2m}$
Convert λ unit			**6.46E-01**	**6.55E-01**	…	**1.79E+00**	$\frac{W}{m^2nm}$

2-15: CONCENTRATORS AND SATELLITES

Most of the interest in multi-junction solar cells comes either from their potential for deployment to power devices in space, or for use on Earth in concentrator photovoltaics (CPV). In space, the increased cost of development and materials of these devices is justified since it is a more efficient use of the weight budget for items launched into orbit. On Earth, silicon technologies currently lead the market in terms of cost, generally, however, there is potential for the more complex MJSCs to become economically competitive when deployed in systems that highly concentrate the sun on a higher-efficiency cell. There are a few cost metrics that can be used to make detailed comparisons of these and related technologies, such as $/W, which has a focus on the up-front costs for installation, and the LCOE, levelized cost of energy, which has units of $/energy, where system lifetime is important part of the equation.

When designing MJSCs to be used in in concentrator systems on Earth, it is common practice to make and report calculations with the standard AM1.5D as the input spectrum. Adjustments should be made when designing for a specific real system. The location of the system is very important, since in northern latitudes, a higher air mass is more appropriate on average. One should consider what is typical over the entire year. Additionally, at dawn and dusk, higher air masses will be apparent to the cell than at midday. If it turns out that a cell optimized for the AM1.5D spectrum has the front, high-bandgap subcells produce the lowest, limiting current, then that cell will be sensitive to the changes in the spectrum that occur at dawn and dusk, since travel through the air attenuates the shorter wavelength photons more than the longer ones. This phenomenon is likely to be relevant, since the front subcells produce the most voltage, so to get the most overall power from AM1.5D, the solar cell should not have any of the lower subcells limit the current, since an inefficient use of the highest energy photons would follow.

2-16: DESIGN APPROACHES

While addressing these subtle factors in MJSC design, it can seem at first as if there are not enough tools in the toolkit. That is to say, the selection of semiconductors available is not large and diverse enough. The desired bandgap or lattice constant is not always available. This is especially true for lattice-matched approaches, which prevent complexity at the system level, but are the most highly constrained by the materials that are available. This is one reason for excitement over new materials and materials development. For a long period, a triple-junction MJSC was the highest efficiency solar cell available, using InGaP, InGaAs, and germanium, with bandgaps of 1.88 eV, 1.40 eV, and 0.68 eV. In this thesis, we explore the possibility of a high-efficiency four-junction solar cell, which has been elusive to the community, due to the shortage of materials with 1.0 eV bandgap. This concept would be a clear step forward, and seems within reach, since there are photons sufficient in the spectrum for the bottom germanium subcell to produce about double the current that is necessary to match the front two subcells, and so, it appears that none of these three subcells need any changes, and a new 1.0 eV subcell could simply be inserted in the third position, for significant improvement.

Dilute nitride semiconductors, such as InGaAsN, have been explored for similar reasons. The strong non-linear bowing shape of the bandgap function with respect to the composition

parameters lead to an area in the material space where the bandgap can be tuned to 1.0 eV with low nitride composition. [21]

Segmentation is another tool that may enable MJSC designers to achieve new levels of efficiency. As an illustrative example of the concept, an early candidate design utilizing the idea is to design a four-junction solar cell with two germanium cells on the bottom, making full use of all of the available photocurrent. [17] If materials with ideal bandgaps are not available, more junctions of the available materials can be used in the design, and the imperfection in the division of photocurrent can be addressed by controlling the ratio of the number of copies of junctions. The current of each junction is matched by controlling the thicknesses of the layers. The front segment of a set of segments of the same material must be very thin in order to share the photocurrent equally, and there are advantages when layers of a device are thin. That section of the device requires less material, and will produce additional voltage due to higher carrier density. The resulting device produces much more voltage, due to the additional junctions, and much less current, which is beneficial to concentrator systems, since the concentration level can be limited by the implicit series resistance of the thin grid fingers, and the emitter layer inside the cell. [4] This thesis aims to inform and contribute to the MJSC design world by developing this novel approach.

3: SILICON-GERMANIUM-TIN $Si_XGe_{1-X-Y}Sn_Y$

3-1: GROWTH

Silicon-germanium-tin is a new and interesting material being grown experimentally at *Polytechnique Montréal* and elsewhere. At this point, the most accessible compositions of SiGeSn are those with a high amount of germanium and a low amount of tin. The composition space of silicon-germanium-tin alloys is complex with many potentially attractive features. There is a great deal of interest in the direct-bandgap region, partially because direct bandgap semiconductors have high absorption, allowing photovoltaic devices to be very thin and sensitive, and also because they can be used to produce efficient lasers and other CMOS (complimentary metal oxide semiconductor) process compatible electronics.

Growers report that the tin has a tendency to collect, forming an island of pure tin instead of organizing randomly into the alloy of desired composition and evenly distributed components. [22] The crystalline arrangement of this alloy is called diamond-cubic, similar in structure to diamond (pure carbon,) and gallium arsenide. The lattice constant is the distance between two nodes in the structure along a specific direction. In this case, the structure can be described as a repeating pattern consisting of a cube interlaced with another cube. All these materials form the diamond cubic atomic arrangement with different lattice constant or size. Actual diamond, composed only of carbon, is the tightest, tiniest manifestation of this structure, since carbon has a very low atomic number, few protons and electrons, and low atomic radius. Carbon is also a group IV element, like silicon, germanium, and tin. CSiGeSn is a possible alloy, but in practice so far, pairs of carbon atoms connect tightly, [23] since the bond length in diamond is normally much less than the other elements, preventing the orderly random arrangement of all the components, unless the carbon amount is extremely minimal. The growth problem presented by tin is somewhat similar, but is proving to be surmountable, and recently, successfully produced SiGeSn lattice samples are being studied and explored. The growth techniques allowing the tin to become incorporated involves lower than normal growth temperature and reduced pressure.

Tin (Sn) is known to form into two types of crystalline arrangement near room temperature. The white, malleable, more dense form, β-tin, has a tetragonal structure. This can transform into the gray, brittle, α-tin structure when below 13.2 ^oC, which has the diamond cubic arrangement similar to silicon and germanium that it adopts in the SiGeSn alloy. The tin atom is large compared to silicon, which is small, and germanium, which is medium.

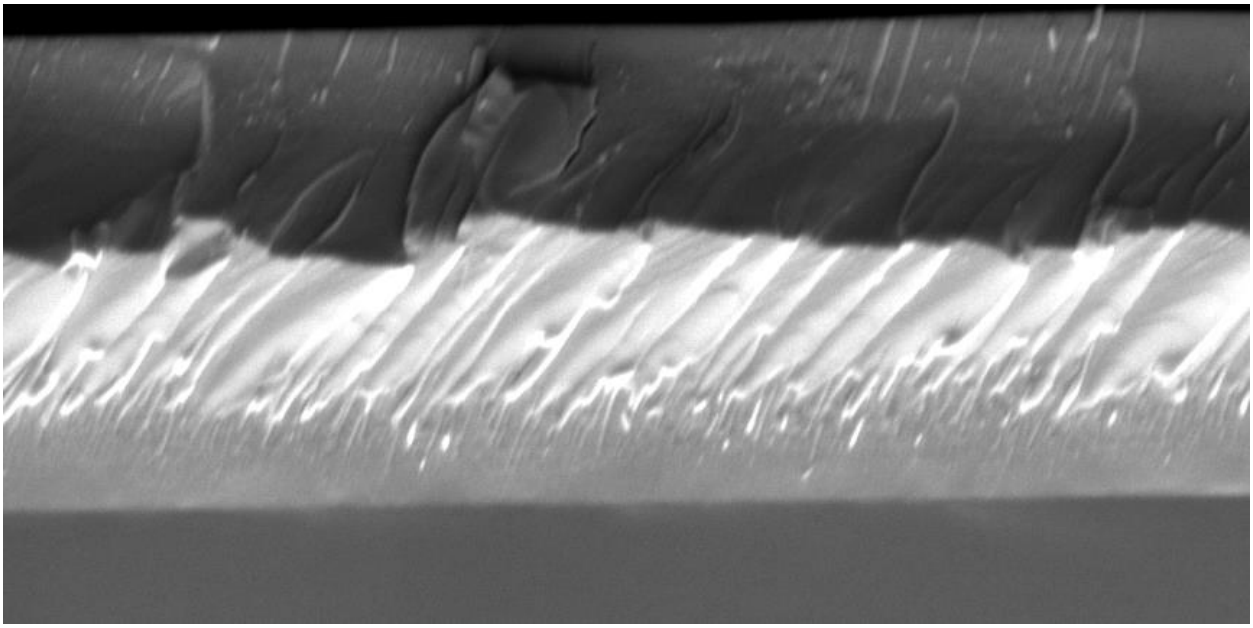

Figure 9: Scanning electron microscope image of GeSn (top) on Ge virtual substrate on silicon wafer (bottom) grown at Polytechnique Montréal. An Energy-dispersive X-ray spectroscopy (EDS) measurement shows tin content of approximately 10% in the top layer. Image taken by Ross Cheriton and Laurier Baribeau in 2018.

3-2: BAND STRUCTURE

Pure germanium has an indirect bandgap of 0.68 eV and direct bandgap of 0.80 eV. Silicon has an indirect bandgap of 1.12 eV, and the direct bandgap is very high. Tin is called a semi-metal, because it has some conduction band states at energies as low as the valence band, making the "gap" a negative value near 0.4 eV. The three-way mixture of these characteristics in the alloy does roughly follow an intuitive interpolation of these values. Observed deviations from linear interpolations in this space are called bowing, because what most simply would have been a straight line function is observed to be bow shaped. The extreme bowing effect of nitrogen composition in the InGaAsN material space is so non-linear that a surprising dip in the function allows for compositions where the bandgap falls as low as 1.0 eV and lower, while both of the extreme points of the function are above 1.0 eV. [21] Less prominent bowing is observed in SiGeSn, with the strongest bowing appearing along the SiSn side of the material space.

The direct bandgap is observed where silicon content is very low or zero (i.e. direct-bandgap GeSn.) Conditions required to produce this bandgap are studied in detail. It can be challenging to determine the achieved material composition or bandgap in a sample. As measurements of the new material are published, the figure for the tin composition *y* required for $Ge_{1-y}Sn_y$ to transition from the Ge-like indirect characteristic to direct is beginning to settle around 10%. The tight-binding calculations of [24] have shown that the direct bandgap is accessible at lower levels of tin composition when the material is strained, however, another effect of the strain is to decrease the bandgap. They plot the calculated effects of both kinds of strain, compressive and tensile, on both the valence and conduction band edges, and it appears that all conduction band minima usually decrease, however, in some cases the direct conduction band edge decreases the most quickly,

which is what allows the for the transition to direct. Their calculations show that the highest direct band gap attainable is 0.72 eV.

There are clearly a large number of important variables, including the composition levels of silicon and tin relative to germanium, the level of strain, both the indirect and direct band gaps, and the band offsets, which are important at interfaces in the solar cell heterostructure. It is important to properly consider all these factors, and when making simplifying assumptions, or limiting the scope of a project, to do so carefully. In this area, there are many promising scenarios. However, a bandgap of 0.72 eV is much lower than the desired level for the third position in a four-junction solar cell. Its role would be to convert photons in the 1.0 eV to 1.4 eV range, so a 1.0 eV direct material would clearly exhibit less thermalization loss. Even so, it is an interesting scenario, and might even still be able to out-perform the popular and robust three-junction design, and may additionally reap benefits available to thin solar cells. Here, the focus is on the indirect regime SiGeSn, where the 1.0 eV bandgap value is attainable, and there may be a difficult absorption profile, but there is not as much efficiency loss to thermalization.

Band structures of materials are a deep subject within solid state physics. Only certain states, corresponding to wavelength modes, are allowed to be occupied by the quantum mechanical electron, like the guitar string. Schrödinger's equation can be solved for the energy levels allowed for electrons. These solutions come with a dependence on the momentum of the electron, which is related to the wavenumber, $\vec{k}$, which is proportional to the momentum, $\vec{p}$, and inversely proportional to the electron's wavelength within the crystal. The direct bandgap is the energy required to move an electron into the conduction band with zero change in momentum. The electron and hole might retain zero momentum, with different wavefunctions, for only an instant, before random thermal energy separates them. The indirect transition is much less likely, since a significant change in momentum must be imparted onto the electron for it to transition to a state away from the centre of the $E(\vec{k})$ plot.

$E(\vec{k})$ diagrams illustrate the constraints by which the electrons in the material must abide. Inside metals, these constraints are few, and electrons are relatively free to occupy any set of quantities for its physical variables, like the magnitude and direction of momentum. Semiconductors exhibit much stronger constraints, as is implied by the concept of the bandgap, which means that an electron can only exit a lattice bond if a certain minimum amount energy is available, unlike in a metal. An indirect bandgap requires not only that amount of energy, say, provided by a photon, like in a solar cell, or by an applied voltage, as in electroluminescence, but also a relatively large increase in momentum. The direction of the momentum is constrained as well. The band diagram is not usually plotted in 3D, since it becomes very difficult or impossible to illustrate the constraints in detail, however, the 2D band diagram with respect to the components of $\vec{k}$ can allow one to roughly infer the structure of the physical parameter sets available to an electron in 3D space. The momentum required is large compared to the small momentum that photons possess. The thermal vibrational motion of the large ions can provide the necessary momentum. This corresponds to phonons travelling within the material. Simultaneous interaction with an incident photon and phonon with proper momentum is necessary for an indirect semiconductor to absorb the photon, greatly reducing the probability of absorption and absorption coefficient compared to direct materials. The direction does not have to be exactly along a lattice matrix axis or diagonal, however

the extreme points of the energy function do occur along those lines. An electron travelling in an off-diagonal or off-axis direction therefore has more potential energy, and might relax into a lower energy state parallel to these lines if able.

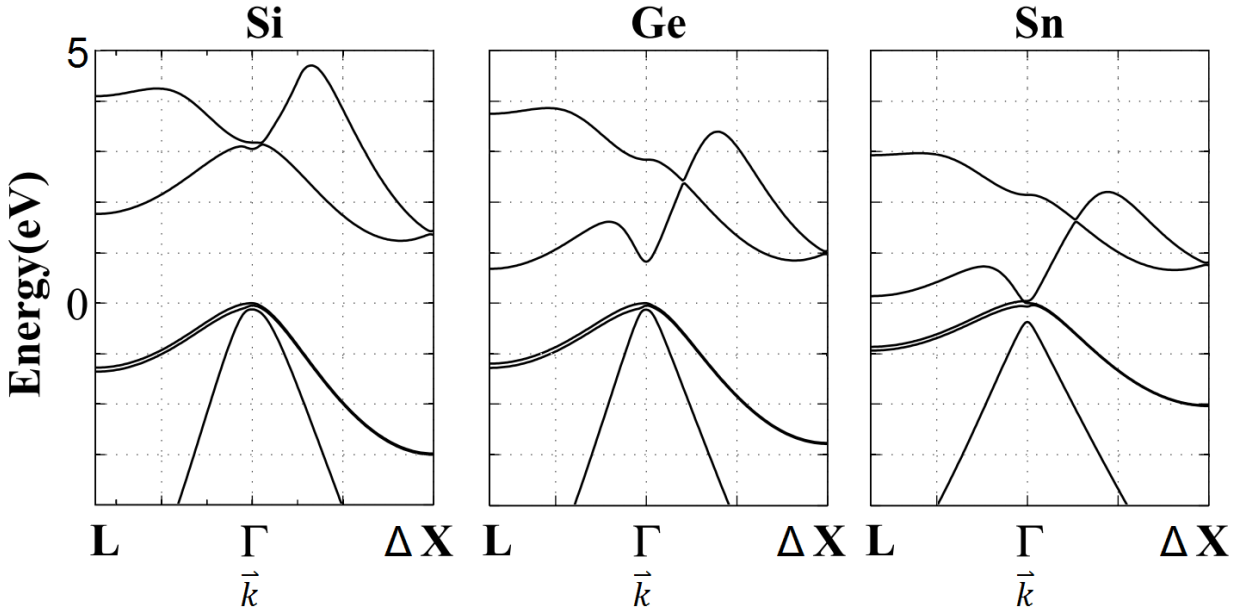

Figure 10: Band structures of silicon, germanium, and tin, as calculated and depicted in [25]*, cropped, modified labels and grid.*

To get a stationary electron moving, it achieves non-zero $\vec{k}$, and momentum, and that slides it up the direct band edge to the side. That is kinetic energy. The $E(k)$ curve shows how much energy it takes to get it moving. The curvature of the plot therefore correlates to a quantity very much like mass, called effective mass. It does not correspond to actual physical matter mass, but it does correspond to inertia, the coefficient of motion with respect to force. In that way, electrons at the bottom of the direct bandgap minimum are stationary, and moving ones occupy states higher up on the line. In an indirect semiconductor, there is no comfort for conduction band electrons unless they have significant momentum, since the state at $k = 0$ has higher potential energy than a moving state, and this also significantly affects the absorption process. A steeply increasing $E(k)$ function indicates that a small increase in momentum corresponds to a very large amount of energy, much like moving a heavy object, and a slowly changing function with a very large curvature, corresponds to an effectively light particle with low effective mass.

3-3: SiGeSn Composition Space Regions of Interest

Whether it's from the results of tight-binding calculations, or more simply from interpolation of the better-known band structures of silicon, germanium, and tin, one can begin to identify areas of engineering interest in the SiGeSn landscape. In a solar cell, the higher probability of absorption in the direct interaction allows for a thinner resultant design. This regime is of interest, but due to the low bandgap, it is not chosen here for the third position of a four-junction cell. It could substitute germanium cells and subcells, offering a similar bandgap, however direct, not indirect, which is an improvement. In the lattice-matched, four-junction solar cell, it is reasonable to restrict the compositions of SiGeSn to those on the line that match the lattice constant of germanium,

5.658 Å. Along this line, the indirect bandgap increases from 0.68 eV, and the direct bandgap increases unfortunately faster from 0.80 eV. An electron absorbed at the interval of the direct bandgap can quickly lose some of its energy by falling into the state corresponding to the indirect bandgap. Therefore, deviations from the zero-strain line matching germanium become interesting, to mitigate thermalization losses of this kind. Diving deeply into the material space, we can follow a line where the indirect bandgap remains constant at 1.0 eV, and the direct bandgap decreases, to leverage as much of the direct style absorption as possible. Remaining photons that do not interact could be absorbed by the subsequent germanium subcell, and these are likely to be the lowest energy ones present. Before we encounter a material where the direct and indirect bandgaps both equal 1.0 eV, we find a region where another indirect bandgap associated with another direction becomes the minimum available energy state.

Group theory symbols assigned to these different directions within the crystal can be used to facilitate the discussion and analysis. Γ refers to the zero-momentum point, X refers to a point in a lateral or vertical direction parallel to an axis of the repeating crystalline structure, and the L symbol indicates a point along a diagonal direction through the cubic repeating unit. The indirect bandgap of silicon is associated with the X directions, and the indirect bandgap of germanium is associated with the diagonal L directions.

There is a line of interest within the SiGeSn material space where the energy levels of the L and X states are equal, and a point on that line where they are equal to 1.0 eV. This material could offer improved absorption over a material with less available indirect transitions. The direct bandgap is slightly larger there, so another point of interest if 1.0 eV absorption is desired is where the X and L bandgaps are each chosen as ~0.9 eV, where the direct bandgap is 1.0 eV. This would not eliminate this kind of thermalization, but it could significantly mitigate it.

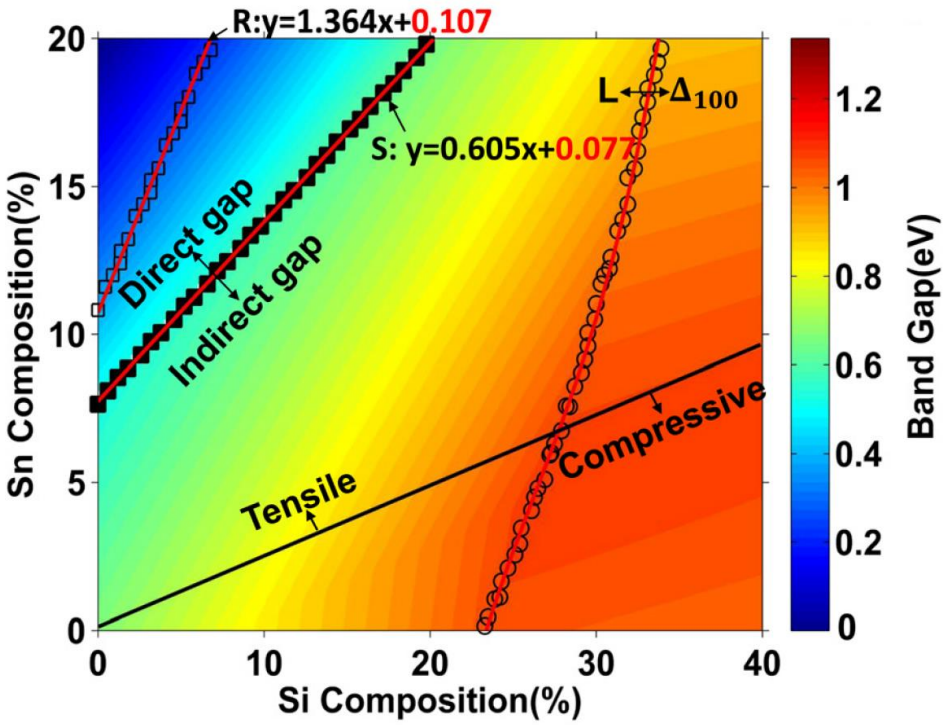

Figure 11: Bandgap of strained SiGeSn lattice-matched to germanium from tight-binding calculation and depiction in [24].

All of these compositions, unfortunately, are well off of the zero-strain line when grown on pure germanium, and would require strain engineering, since such detailed control over the band structure requires all available degrees of freedom, and constitute significant deviation from the lattice constant of germanium. If the direct bandgap is low enough, a solution to the strain problem might be more available, since the thickness requirement for sufficient absorption might remain very low. Additionally, these deep compositions with very high amounts of silicon and tin are very difficult to grow and have not yet been demonstrated.

The prospect of SiGeSn growth at a different lattice constant on a different substrate, such as that of GaAs, is of course an interesting proposition. The gallium arsenide lattice constant is slightly smaller than that of germanium. The introduction of 1.173% indium to gallium arsenide creating $In_{0.01173}Ga_{0.98827}As$ is done in the solar cell lattice-matched to germanium for that reason. This reduces the bandgap of the material from 1.42 eV to 1.40 eV. At the smaller lattice constant of GaAs, less strained SiGeSn growth would be on the side of more silicon and less tin. This makes the direct interaction even less attainable. One might consider complete departure from a design relying upon direct absorption, and look to the binary SiGe. The *X*-related indirect bandgap of Si has low dependence on composition, so most of the SiGe material space has a bandgap similar to that of silicon. One point of interest might be at the *L* / *X* indirect band gap crossover point, where the two bandgaps are equal, near 75% germanium. Nearer to the point of 100% germanium, the alloy has a germanium-like band structure, except that the bandgap increases. This could be desirable for the same reason as SiGeSn, since a 1.0 eV bandgap exists in this range, however SiGeSn can remain lattice-matched to pure germanium, making a high-quality substrate available. SiGe could be incorporated into a strained heterostructure to balance the strain introduced by high

Sn content SiGeSn that accesses the direct bandgap. The difference in lattice constant between GaAs and germanium is very small, and there is a similarly small degree of difference when comparing the properties of SiGeSn lattice matched to germanium and to GaAs, though the lattice constant of germanium is closer to the regions where the direct bandgap might contribute beneficially. Inverted growth, where the top subcells are grown first, prioritizes the material quality of the upper subcells, and is viable, as is growth from the bottom on germanium.

When considering the band structure of the $Si_xGe_{1-x-y}Sn_y$ alloy, in choosing silicon and tin compositions (x and y,) there are several areas of interest. Here the four-junction scenario is considered especially, while keeping in mind the possibility of applying the segmented design approach. First, SiGeSn makes available indirect bandgap selection higher than germanium without changing the lattice constant. The intent of a choice on or near strain line is to divide the solar spectrum evenly in a four-junction design, and produce a higher subcell voltage than germanium does, noting the potential to improve upon the InGaP/InGaAs/Ge/Ge idea. It may require either very thick growth of SiGeSn to compensate for the poor absorption provided by the indirect bandgap, or for the direct bandgap to play a role, as it does in germanium, providing better absorption, for those photons that are at a high enough energy level (above 0.8 eV in germanium).

Regarding the indirect-direct crossover in the material space, it seems likely that keeping the *L* (indirect,) and Γ (direct) conduction band energy levels nearly equal would be desirable in most solar cell designs. That is to say that compositions near the crossover line will often be more desirable than nearby regions. The reason is that, as tin content is increased, entering more deeply into the direct regime, the Γ bandgap continues to decrease below 0.7 eV, down to zero. If the fourth subcell will be made of germanium, potentially as in an active substrate, then a decreasing third subcell bandgap reduces the voltage of the subcell that is intended to absorb photons in the 1.0 to 1.4 eV range, with the difference in energy being lost to thermalization. The desired maximum direct bandgap available will reside on the crossover line. Another reason to choose a composition on the direct-indirect crossover line is that having multiple bandgaps at the same value can promote absorption and reduce the necessary thickness of the subcell. Moving into the indirect area from the crossover point is less desirable than having a direct semiconductor, since readily absorbed photons from the direct process can lose energy by transitioning to the indirect state, however, this regime will not be disregarded, since we wish to have a bandgap higher than 0.7 eV. It may be that the direct transition can be utilized to boost the absorption of the otherwise indirect semiconductor.

To consider areas of the compositional space other than the zero-strain line, or direct bandgap crossover, it is best to keep track of both conduction band levels. Near the direct bandgap crossover, even on the indirect side, the direct transition may be low enough to promote absorption. Importantly, the carrier lifetime is longer in indirect materials, since the reverse transition (opposite of photon absorption – radiative recombination,) requires creation of a phonon as well as a photon, to conserve momentum. For example, it is helpful to note that, while traveling along the zero-strain line, beginning at 100% germanium, the indirect bandgap can increase beginning from 0.68 eV, and the direct bandgap increases beginning at 0.8 eV, however, the direct bandgap unfortunately increases faster. Therefore, it seems that in this regime for solar cells, most designs would rely primarily on the low-probability germanium-like indirect transition. Importantly, this can be more

costly in the case of grown SiGeSn, since in the case of germanium, very high-quality crystalline ingots are produced enabling large absorbing thicknesses by incorporating an active substrate into the design, in contrast to the vapour deposition methods used for SiGeSn. Our first proposed designs will work on the line lattice-matched to germanium, such that defects can be kept to a minimum, so substantial grown layer thickness may be required.

Designs involving mechanical strain are likely to benefit greatly from the direct-style photon absorption. Expecting strong defect related effects, a promising area of the design space involves inverted growth, such that the III-V subcells, grown first, do not inherit defects of the SiGeSn. This approach involves substrate removal and, optimally, substrate recycling, with maximized direct bandgap SiGeSn, and may not even involve a germanium subcell, which may be more performant when included as a substrate, since low-bandgap direct SiGeSn would be available and likely ideal in that scenario, grown behind the third SiGeSn subcell. Given that direct bandgap SiGeSn is not available at 1.0 eV, this design would involve thermalization losses, but the layers would be thin, and could have competitive efficiency and cost. Additionally, the thin subcells would have a compensating voltage benefit, due to high carrier concentrations.

If considering the strained, indirect space, that are a couple of other lines that are likely to be appealing. These regions in the design space are deeply far from pure germanium, with compositions reportedly difficult to attain, due to the proclivity of tin to collect in pools of pure alpha-tin, ruining the device. This has been the ubiquitous challenge in SiGeSn growth, which has delayed its availability until recently, and continues to obstruct access to the entire SiGeSn compositional space. However, in these regions, potentially within future reach, we can consider the line where L and X regions intersect. The L-bandgap is the indirect bandgap typical of germanium, however, the indirect bandgap of silicon occurs at the X point in $\vec{k}$-space, and there is a crossover point in the SiGeSn compositional space. This line may be of interest because both bandgaps will then be near (slightly above,) 1.0 eV, and with both indirect bandgaps equal and present, doubled indirect absorption probability may be observable. There is a crossover between this line, and the line matching the lattice constant of germanium. Fairly high diffusion lengths (~5 μm) have been cautiously concluded in SiGeSn growths, [26] which is normal in indirect materials, however, the improved absorption would still be desirable, to help mitigate growth and material costs, and improving the subcell performance.

It has been pointed out that 1.0 eV indirect SiGeSn lattice matched to germanium will have a somewhat large, ineffective, direct bandgap near 1.15 eV. If the lattice constant constraint is relaxed, we can consider larger lattice constant SiGeSn compositions and use the degree of freedom to affect the direct bandgap, while keeping the indirect band level difference at 1.0 eV. This can be used to reduce the direct bandgap slightly. This line then intersects with the L-X crossover, creating an interesting point, where L and X bandgaps are both 1.0 eV, and the direct bandgap is as low as possible.

The extreme depth, low germanium-content space, is far from uninteresting. The bowing model, though potentially inaccurate, shows regions of direct bandgap approaching 1.0 eV, which could lead to good absorption at higher bandgap, leading to higher subcell output voltage than designs

using SiGeSn at compositions within the aforementioned areas nearer to 100% germanium, near the lattice constant of germanium and GaAs.

There is some degree of variance in the literature with respect to the estimates of the energy levels. The above analysis is somewhat low resolution in nature. These intersections of bandgap values certainly exist somewhere in the material space, though exact compositions can be difficult to measure, and are sometimes determined based on an inference of the bandgap, which is not applicable in the precise determination of the bandgap as a function of composition. These properties are discussed in terms of bowing parameters. A bowing parameter indicates a deviation from linear trend at the binary alloy edge in the compositional space. The binary alloys SiGe and GeSn are attainable and actively researched materials that precede SiGeSn. SiSn is the more unattainable edge of the composition space, and the bowing parameter for this edge of the space is the largest. [27] The bandgap of the material is defined as the minimum available difference between a valence band (heavy hole or light hole,) and the conduction band. It may be reasonable to consider the atomic sizes of silicon, (small) germanium (medium), and tin (very large,) and notice that the SiSn binary might be the most unstable or dynamic.

The inclusion of carbon could be helpful in bandgap engineering in the future. Diamond is known to have a very high bandgap of 5.47 eV. The difficulties in incorporating the element aside, it is also important to note that strong bowing has been observed such that dilute levels of carbon actually decrease the bandgap. This was first shown by photoluminescence in [28]. [29] Even so, it is remains a very expansive material system with many regions that could become useful as the growth technology progresses. [30]

3-4: BANDGAP FUNCTION

These models can be implemented in Synopsys TCAD Sentaurus. The bandgap model in the material parameter code files implements the Varshni temperature dependence [31], with the coefficients of the expression being set in the code, making temperature available to be changed as a workbench variable parameter. One of the input values in the expression corresponds to the bandgap at 0 K, and two coefficients control the temperature dependence. Given a bandgap at 300 K, one must set these coefficient values. The $Si_xGe_{1-x-y}Sn_y$ bandgap could correspond to the *X*, *L*, or Γ transition, so the mole-fraction (*x* and *y*) dependent bandgap model file could execute a minimum function with respect to these three possible transitions to implement a SiGeSn bandgap model in Sentaurus. Given that 0.9 eV bandgap SiGeSn has been reported, the first version of the SiGeSn par file can be created by adjusting the germanium file's E_{g0} from 0.744 to 0.85 eV, increasing the bandgap by 0.106 eV. The other parameters, the Varshni coefficients, α and β, are kept the same. The bandgap change will be reflected in the voltage produced by the subcell, but the absorption characteristic is controlled by the imaginary part of the complex refractive index, *k*.

This project uses the Sentaurus module called MatPar to execute the material parameter models, which dictates how the code will be organized. The original method is to create a .par file in the syntax required by Sentaurus Device, the following module, which will solve optical and electrical effects of in the device. These modules and project execution sequence will be visited more completely in later sections. A .par file is not executable, it is a text file containing numerical physical quantities used by Sentaurus Device, and is not suitable to implement a model. A function

must be implemented that will return the bandgap value, given a number of arguments. In initial simulations, the bandgap might be set manually, but the preference is to implement flexible material code that can accept input arguments, and work for more than one composition. This is why .tcl files can also be used to define material parameters. The TCL programming language interpreter will run the .tcl parameter file code, evaluating the model function for given parameters, to automate the production of the .par file.

A Sentaurus project using MatPar usually contains a folder called *par* and a folder called *pardb*. The *par* folder can contain .par files or .tcl files, as some materials might be implemented with the static text of .par files, and some with the dynamic model functions of .tcl files. As such, .par files contain completed information, but the .tcl files contain code that refers to the *pardb* folder. The *par* folder need not contain subfolders. The first module of the project, called *Epi*, uses a .csv file to define the layered structure of the device, and refers to these material parameter definition files. The *pardb* folder organizes many material models in a standardized way, including header folders. The top level of folders are the material names. The next level of folders are groups of parameters, such as *BandStructure*, *Mobility*, *Optics*, *Permittivity*, and *Recombination.* In *BandStructure*, there may be, for example, *Bandgap*, *eDOSMass*, *hDOSMass*, *JainRoulston*, *TableBGN*, etc.. The purpose for this organization is to allow for many different models that may exist for the same material parameter. [32] In the *Bandgap* folder, a .tcl file that implements a function for the SiGeSn bandgap is created as follows.

```
namespace eval SiGeSn::Bandgap {
  proc general_bowing_formula {x y p1 p2 p3 {b12 0} {b13 0} {b23 0}} {
    # Chen [27] equation 2, except Si and Sn are swapped. We are using x for Si, and y for Sn, as in
  SixGe1-x-ySny
    if {[expr $x + $y] > 1} {
      error "SiGeSn bowing function x and y too high $x + $y > 1"
    }
    return [expr {(1.0-$x-$y)*$p1 + $x*$p2 + $y*$p3 - $x*$y*$b23 - (1-$x-$y)*$x*$b12 - (1-$x-$y)*$y*$b13}]
  }
  proc SiGeSn_bowing_formula {Si_fraction Sn_fraction p_Ge p_Si p_Sn {b_GeSi 0} {b_GeSn 0} {b_SiSn 0}} {
    return [general_bowing_formula $Si_fraction $Sn_fraction $p_Ge $p_Si $p_Sn $b_GeSi $b_GeSn $b_SiSn]
  }
  proc delta100_bandgap {x y} {
    set  Eg_Si  1.12  ;# Ioeffe Institute [33], Levinstein [34]
    set  Eg_Ge  0.85  ;# [33], [34]. [25] agrees (Figure 10 Δ).
    set  Eg_Sn  0.867 ;# Madelung [35] α-tin diagram (low significance here).
    return [SiGeSn_bowing_formula $x $y $Eg_Ge $Eg_Si $Eg_Sn]
  }
  proc L_bandgap {x y} {
    set  Eg_Si   2.0    ;# [33], [34]. Chang [36] cites Madelung [37] 1.65 eV (!)
    set  Eg_Ge   0.664  ;# Chang [36] (agreement with Ioeffe [33] of 0.661 eV)
    set  Eg_Sn   0.0092 ;# Chang [36] cites Madelung [37]
    # Chen [27] discusses, using strained values, cites [38], [29], [39], [25]:
    set  b_SiGe  0.169
    set  b_GeSn  1.23
    set  b_SiSn  0.925
    return [SiGeSn_bowing_formula $x $y $Eg_Ge $Eg_Si $Eg_Sn $b_SiGe $b_GeSn $b_SiSn]
  }
  proc Gamma_bandgap {x y} {
    set  Eg_Si   4.185  ;# [36] cites [37]. Γ1 from Ioffe is 3.4 eV, but it's Γ2 that comes down with Sn
  (agrees)
    set  Eg_Ge   0.7985 ;# [36] cites [37], Ioeffe [33] agrees (0.8 eV)
    set  Eg_Sn  -0.413  ;# [36] cites [37]
    # Chen [27] discusses, using strained values, cites [38], [29], [39], [25]:
    set  b_SiGe  0.21
    set  b_GeSn  1.94
    set  b_SiSn  13.2
```

```
    ;# [36] cites [39]. Valid for high Ge fraction [36], rather, it is from a fit on the strain line [39].
    ;# [39] also proposes 3.26 eV (see Figure 13). [40] calls b_SiSn a "distinct uncertainty," all agree
on that.
    return [SiGeSn_bowing_formula $x $y $Eg_Ge $Eg_Si $Eg_Sn $b_SiGe $b_GeSn $b_SiSn]
  }
  # "SiGe bowing parameters are small and consistent"
  proc bandgap {x y} {
    return [expr {min([delta100_bandgap $x $y], [L_bandgap $x $y], [Gamma_bandgap $x $y])}]
  }
  .
  .
  .
```

The development strategy employed here is to keep procedures short. This helps keep tasks small and complexity low. For example, variables used to calculate the *L* bandgap are not required in the function that calculates the Γ bandgap, therefore it is appropriate that they are out of the scope of the `Gamma_bandgap` function. The higher-level "`bandgap`" function is also conceptually simple. It returns the minimum of the previous three bandgap functions. The bottom-level function, `general_bowing_formula`, is tasked with implementing the bowing formula, and takes the necessary values as arguments. This general-purpose function will be used for many material parameters. Other material parameters do not exhibit bowing, and linear interpolation is sufficiently accurate, which corresponds to a bowing coefficient of 0. The TCL syntax of `{b12 0}` sets the default value of that argument, so this function can be called with less arguments to interpolate those other material parameters that do not exhibit bowing. This kind of organization of functions also corresponds to "top-down programming" style.

The first line sets up a namespace in accordance with what Sentaurus's MatPar module expects. MatPar requires "namespace variables" to be set, and refers to them when printing the material parameters for Sentaurus Device. A namespace in TCL is a place where variables and functions can be defined, and not collide with other variables and functions of the same name. For example, this namespace is called `SiGeSn::Bandgap` and defines a function called "`bandgap`". There might be another function called "`bandgap`" in another namespace, such as "`Ge::Bandgap`," and these two functions are in a separate space and do not create a naming collision problem.

Typically, a function called `init` is defined, and its role is to calculate and store the namespace variables that will be read by MatPar. Sentaurus Device expects from the `Bandgap` section of the .par file variables such as `Eg` (the bandgap,) and `Tpar`. `Tpar` is used in most material parameter `init` functions, and will tell Sentaurus Device the temperature corresponding to the value of the other variables. Sentaurus Device implements the temperature dependence model of these variables. `Eg` corresponds to the bandgap at the simulated device temperature. `Eg0` corresponds to temperature `Tpar`. `alpha` and `beta` are also set here, such that Sentaurus Device can implement the Varshni temperature dependence of the material bandgap for temperature-dependent simulations. Finally, Sentaurus Device does expect the electron affinity, `Chi` (X), to be set in a bandgap function, to be used in the production of a band diagrams.

Additional bandgap models can be implemented by doing so within a new file. Again, the role of the top-level TCL file in the par directory is to call and assemble the models of the *pardb* directory. To make use of another model, and have both available on hand, another file is made, and the top-level TCL file is also updated. It would be possible for MatPar to do this by naming procedures (functions) according to the name of the model, instead of relying upon file names, and then calling

models created as functions instead of models created as files. New TCL model files implement the same namespace name, which must be named according to the MatPar convention, and calculate the same variables, that MatPar and Sentaurus Device expect, in a different way.

The above code is converted to python and used to generate data for Matplotlib [41] to produce the bandgap plots of this section. The view angle in Figure 12 a) better shows the boundaries of the bandgap type regions. In Figure 12 b), the view angle reduces the parallax such that the bandgap value can be read on the back scale.

These visualizations are inspired by Attaioui and Moutanabbir [24], who published bandgap calculations including strain and the direct-indirect crossover, by the figures from Moontragoon, Soref, and Ikonic [42], that map the entire SiGeSn material space, and by Chen *et al.* [27] and *Chang et al.* [36], who described among others and reviewed the bandgap model with bowing coefficients that define the curvature of the planes. The plot is only considered accurate for compositions with high germanium content, and is produced here mainly to facilitate somewhat qualitative discussion of the space. The corners of the triangles correspond to silicon (1.12 eV, Δ-type) germanium (0.67 eV *L*-type,) and tin (-0.4 eV, Γ-type.) The triangular axes are a transformation from rectilinear space via $x_T = x + y\sin 30^o$ and $y_T = y\cos 30^o$, where x and y are the silicon and tin compositions respectively, to produce symmetry with respect to the three component elements. The set of compositions that match the lattice constant of germanium is shown with the angled line starting at the 100% germanium corner. The first view shows better the shape of the L region and the intersections of the planes, while the second view angle is better to read the bandgap value with less parallax against the vertical axis, and with color contours also present showing the levels of equal bandgap within each surface. The three-dimensional geometry allows for the extrapolation of the Γ surface to estimate its value within the indirect regions.

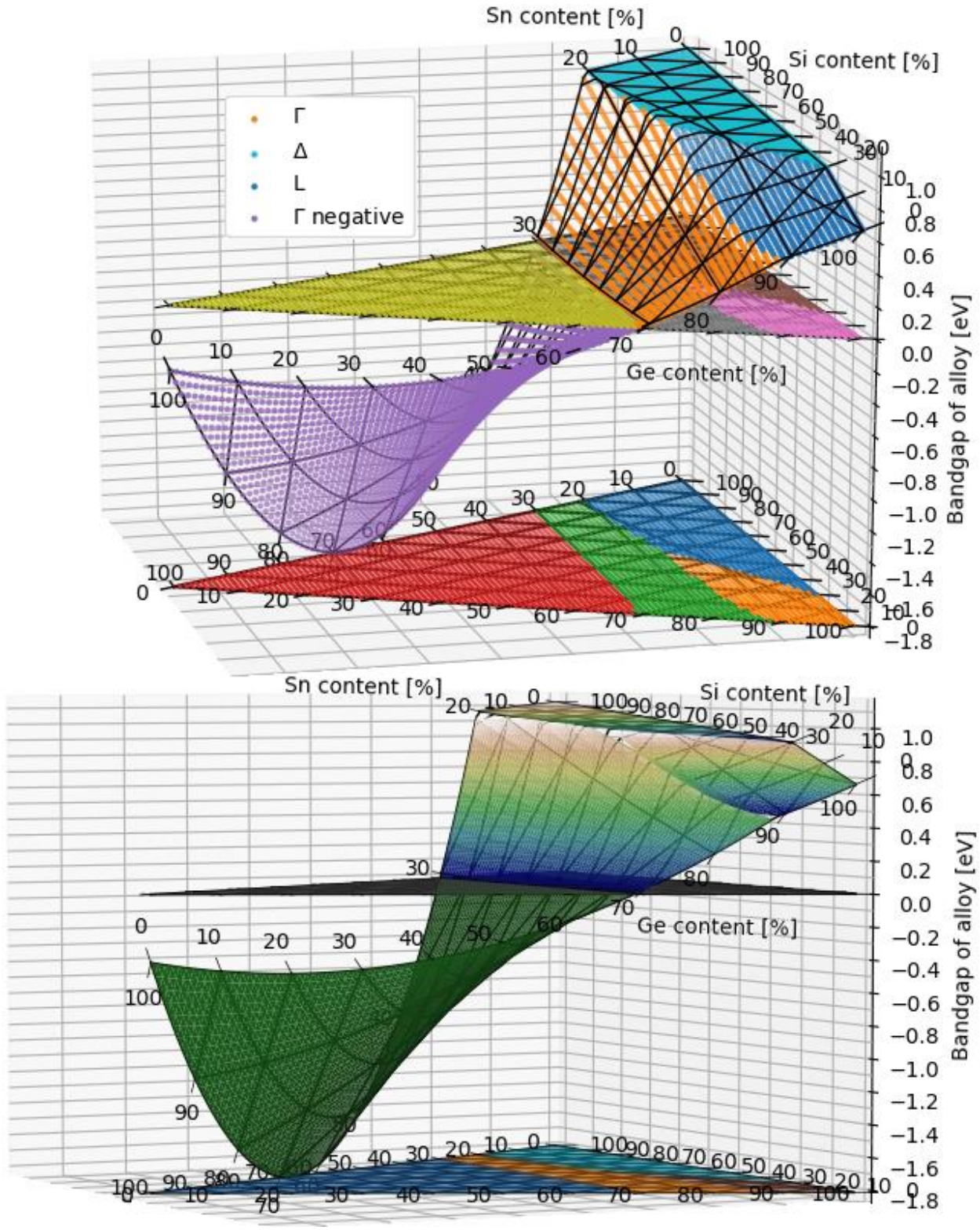

Figure 12 a), b): SiGeSn bandgap bowing formula output as a function of alloy composition. The term for strained GeSn [27] *is used.* b_{SiSn_Γ} *is a fit to data on the strain line with germanium content above 50%.* [39]

3-5: SiSn Bowing Parameter b_{SiSn} Values and Validity

While the very large (giant) [39] bowing parameter of 13.2 eV is widely invoked in bandgap modelling of SiGeSn, it is also well known that additional experimental data is necessary to solidify or modify the model in the extreme range of low germanium content material. [27]

The 13.2 eV value for b_{SiSn} originally proposed in [39] is based upon a fit to a set of direct bandgap measurements of SiGeSn compositions lattice-matched to germanium. This set of values indicates that the direct bandgap trend as the germanium composition decreases along this line must have a negative coefficient (B) to X^2, where X is the sum of the silicon and tin compositions. The valley reaching -1.8 eV on the SiSn side in Figure 12 is therefore a consequence of that fit, chosen to set the direct bandgap value according to experimental results on the line of lattice match, from data with high germanium content. These values are not shown as they are not chosen by the `min` function, but would correspond to the extrapolation of the visible part of the direct bandgap surface. Given the bowing terms of GeSn and SiGe, the SiSn bowing term must be large and positive to fit to the observed parabolic trend in the experimental results. The designs explored in this study consider SiGeSn lattice-matched to germanium, and therefore, the 13.2 eV value is appropriate.

D'Costa *et al.* [39] also indicate that this term is not appropriate for all compositions. They conjecture anomalous bowing, and a lower value of $b_{SiSn} = 3.26\ eV$ for compositions with tin content (y) near 50%. This lower value is much closer to prior SiSn supercell (4 eV) and empirical pseudopotential (3.915 eV) calculations of Moontragoon, Soref, and Ikonic [42]. [39] This conclusion was preferred over the supposition of another non-zero bowing coefficient proportional to the product of the three compositions, which would be most significant at the centre of the composition space. This idea, the relative magnitudes of the SiSn bowing parameter and another ternary coefficient proportional to $xy(1-x-y)$ was further explored in [38]. Many publications discuss these parameters and the reasons for the discrepancies, which are usually due to differing conditions or contexts, such as temperature, strain or composition. [27] Figure 13 plots the bandgap model when $b_{SiSn} = 3.26\ eV$ is employed. This value is proposed by D'Costa for tin concentration $y = 0.5$, and therefore preferred in the space farther from the Ge lattice match line.

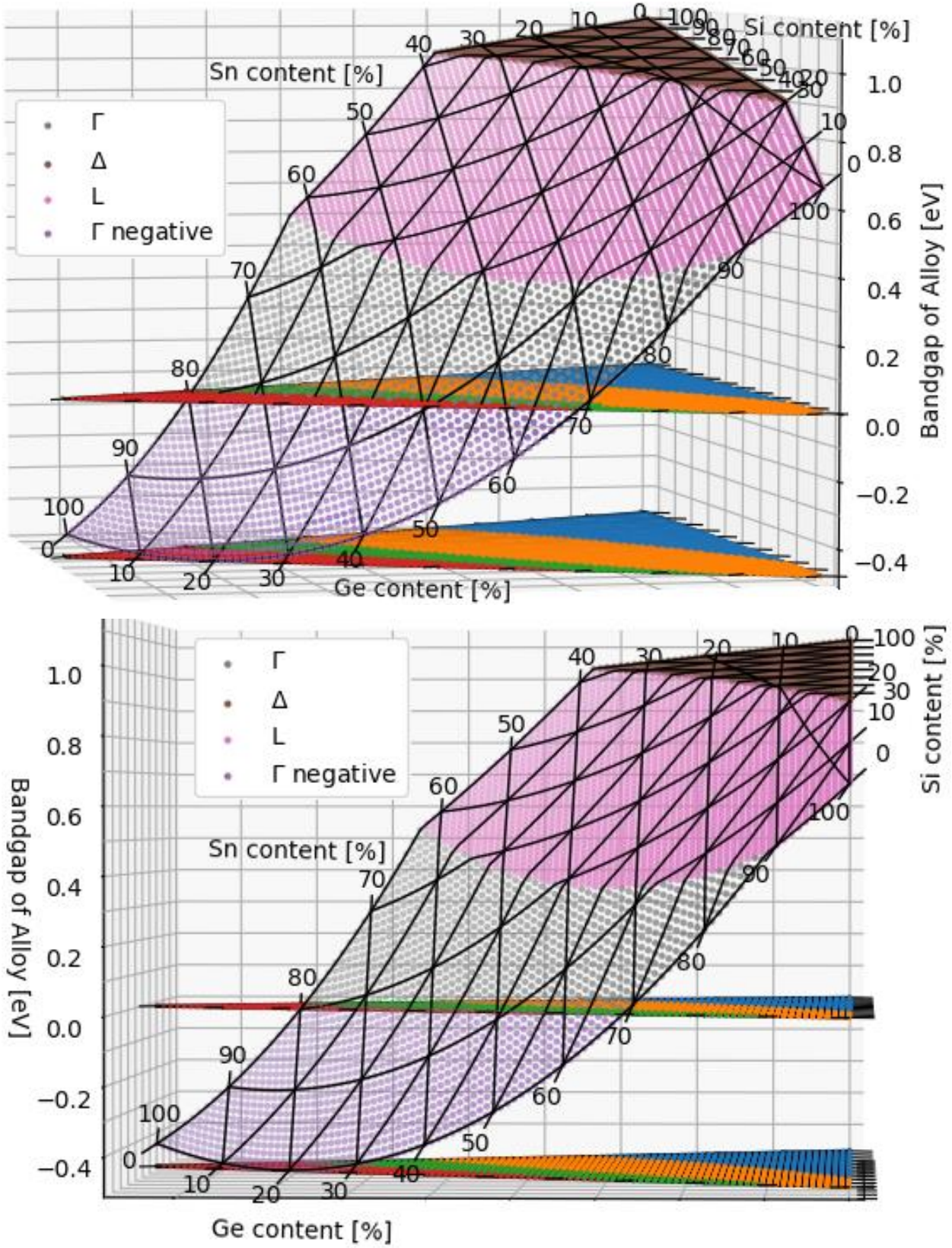

Figure 13 a, b: SiGeSn bandgap model with $b_{SiSn} = 3.26$ eV.

The empirical pseudopotential theory of [42] produce a SiSn 0 eV crossover result at 65% Sn, and the model with higher bowing parameter produces it at 30% Sn.

The primary conduction band minimum of germanium is in the L direction, which is a secondary minimum in silicon, creating the L-Δ transition along the SiGe edge. The value of 1.65 eV is tabulated in [36], citing [37] for this energy level in silicon. However, most other sources and calculations indicate a higher value nearer to 2.0 eV. [43] This value is important to the indirect bandgap slope with respect to composition in the region of interest, along the strain line near the pure germanium composition. The higher value would be preferable, keeping the indirect and direct states at a similar energy level, and making higher indirect bandgap more attainable at lower.

These visualizations can be applied to the popular bandgap versus lattice constant axes for semiconductors (Figures 14 and 15). The plot depends strongly however on the value of the mysterious b_{SiSn}. The higher value 13.2 eV should be valid at the lattice constant of germanium, but produces invalid information at higher lattice constant or tin concentrations. The condition $y > 50\%$ is depicted with red grid lines in Figure 15, and indicates an invalid result. This condition corresponds to ¼ of the composition space. The invalidity is expected to extend beyond the red region.

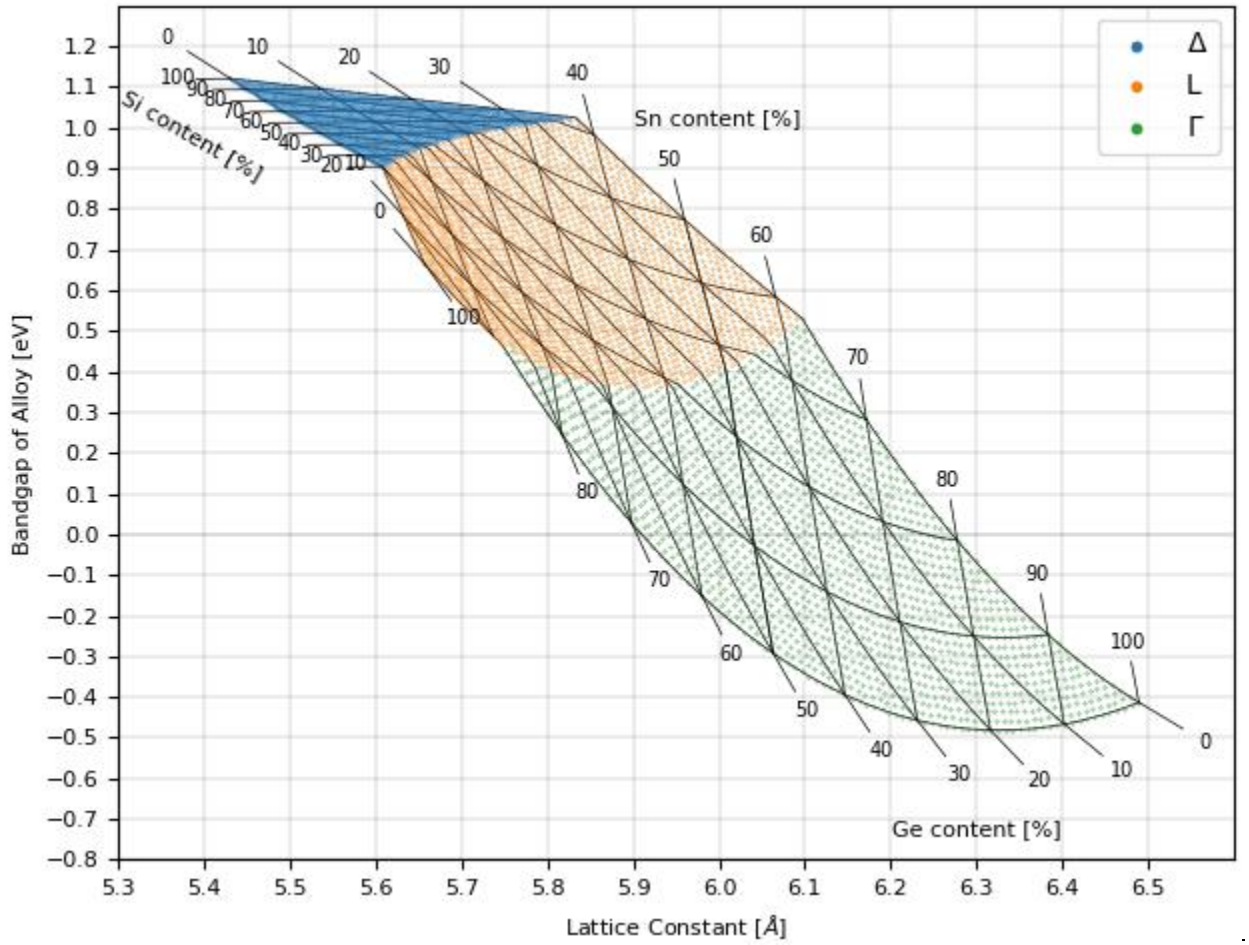

Figure 14: Bandgap vs Lattice Constant with b_{SiSn} = 3.26 eV.

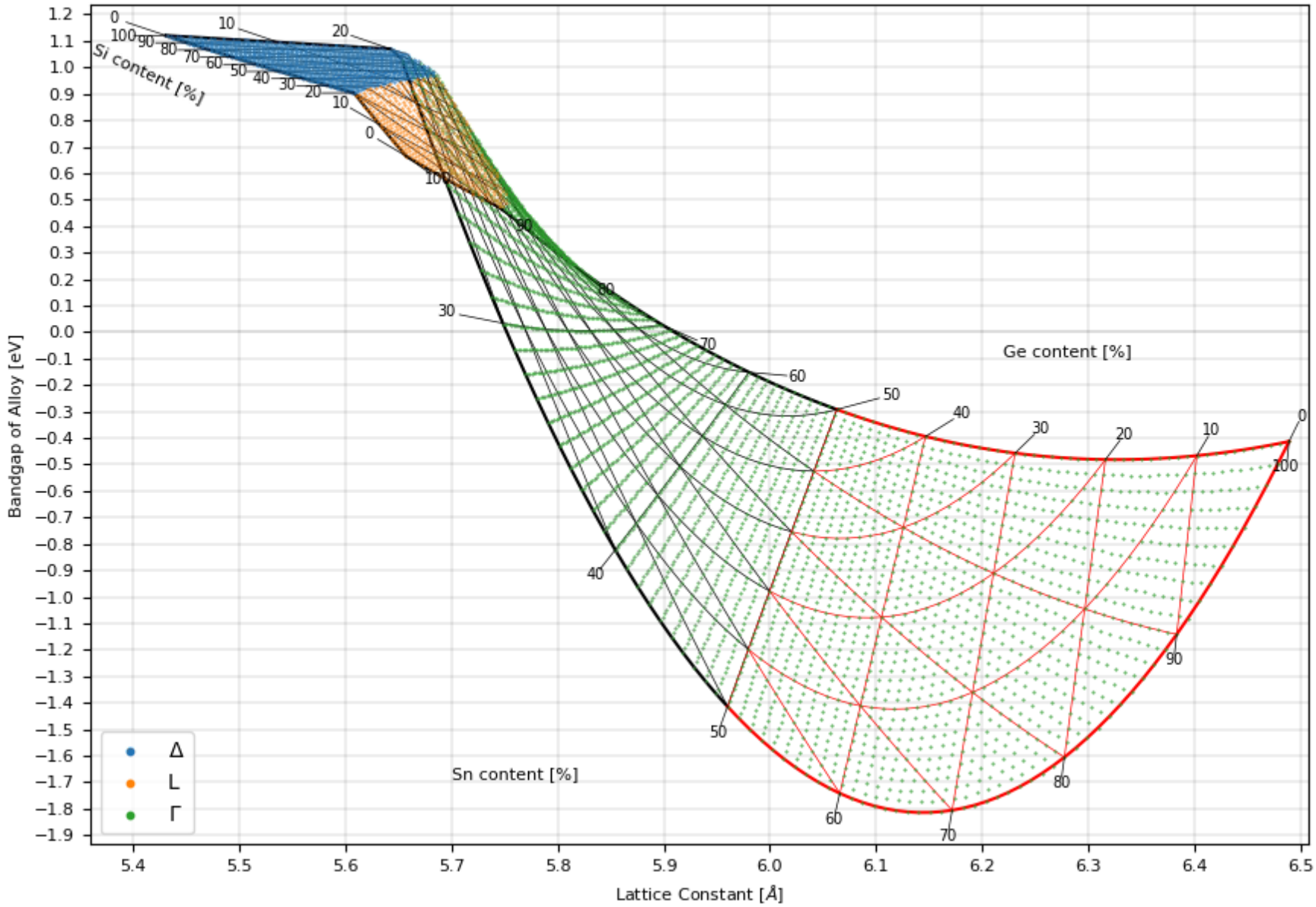

Figure 15: Bandgap vs Lattice Constant with b_{SiSn} = 13.2 eV.

3-6: General Approach To Electron Affinity

Bandgap model implementations in MatPar for Sentaurus Device imply a question about the band edge offsets with respect to other energy levels in the heterostructure, meanwhile, the effect of temperature is handled simultaneously. Valence and conduction band offsets in a heterostructure are best handled specifically, as transitivity does not always hold accurately with respect to the vacuum level, due in part to interface effects. [32]

The bulk electron affinity quantity dealt with here is different from the values commonly used in chemistry. The meaning is similar, only in that it is the energy required to remove an electron. In chemistry, the energy quantity of this process corresponds to atoms in the gaseous phase reacting in a multiplied individual process. The energy required here is different, rather, for an electron residing in a regular crystalline solid bulk. For example, the electron affinity of silicon in chemistry is $133.6\ \mathrm{kJ/mol}$, which, when converted by dividing by Avogadro's number, $6.022 \cdot 10^{23}\ \mathrm{mol}^{-1}$, and multiplying by the energy conversion factor $6.24 \cdot 10^{21}\ \mathrm{eV/kJ}$, equals 1.39 eV, while electron affinity for one electron within solid state silicon for this context at 300 K is 4.05 eV [34].

The symbol X (Chi) refers to the electron affinity, the energy level difference between the conduction band edge and the supposed vacuum energy level for an electron. Values of X are often publicized for bulk semiconductors at 300 K. The bandgap model files will typically refer to such a quantity by default to compute the energy level steps. If a band offset value is known at a different temperature, such as 0 K, an adjustment must be made to calculate the appropriate corresponding override value for X the appropriate temperature.

Normally, a bandgap's dependence on temperature is to decrease according to the Varshni model, which contains two coefficients, α and β. [31] As the bandgap decreases, the consequence on X supposes that half of the effect is allocated to conduction band level, which constitutes an increase in X, and the other half to an increase in the valence band level. Each material has different coefficients.

In MatPar, an important variable is the "substrate" variable. This variable does not refer necessarily to the physical base material of the growth. In this context, the term refers instead to the reference material to be used for band offset specifications. MatPar will execute model code for the material nominated as the substrate first, and any band offset specification should be made with respect to the material named as the substrate. In this project, there are many III-V materials whose band offsets are normally specified with respect to GaAs, so GaAs remains the ultimate reference for all band offset values. MatPar runs the material code for the material at the bottom of the structure next, which in this case is germanium. SiGeSn band offsets are modeled relative to germanium, and the band differences between germanium and GaAs will also be required in the SiGeSn code space to calculate X.

The default bandgap model files have code that accepts a relative conduction band offset as input, supplied at the same temperature as the bandgap. Since the bandgap may be known at one temperature, and a valence or conduction band offset at another, it is important to be able to translate all of these quantities with respect to temperature according to the Varshni model. The relative conduction band offset is not a value in eV, instead, it is a scalar factor of the difference

in bandgap between the given material and substrate material. If a valence band offset is known, the new X could be calculated manually, but a conversion function that calculates the relative conduction band offset is equally useful, since the default bandgap models accept that quantity as an input and will use it to write the appropriate X for the material in the heterostructure.

Sentaurus Device will require a bundle of these quantities all defined at the same temperature, T_{par}. X_0 is the electron affinity at T_{par}, and E_{g0} is the bandgap at T_{par}. The model file must define this bundle of values as namespace variables which will be read and printed by MatPar. The variable `::temp` is the temperature of the simulation. Sentaurus Device will implement the Varshni model to determine the physical quantities at temperature `::temp` based on the other inputs.

An issue here that is solved by flexible conversion functions is that there is only one T_{par}, while the bandgap, potentially band offsets, and default bulk X may rely upon source data at different temperatures. The default is not immediately applicable if T_{par} must be changed. The approach will be to determine E_{g0} first. If a heterostructure band offset is known, the relative conduction band offset is determined, then X at the current temperature, then at T_{par}. If it is not, then the bulk value at given temperature (usually 300 K) is used to determine X_0.

The Varshni temperature dependence of band gap is [31],

$$E_g(T) = E_g(0) - \frac{\alpha T^2}{T + \beta}$$

leading to a difference expression,

$$E_g(T_2) - E_g(T_1) = \frac{\alpha T_1^2}{T_1 + \beta} - \frac{\alpha T_2^2}{T_2 + \beta}$$

and an absolute expression, given any reference point, T_1:

$$E_g(T_2) = E_g(T_1) + \frac{\alpha T_1^2}{T_1 + \beta} - \frac{\alpha T_2^2}{T_2 + \beta}$$

The difference formula is less restrictive, not requiring a value to be specified at 0 K, and can be used as effectively as a definition of $E(T)$. This expression is implemented in TCL in a general manner that may be used in any material's bandgap model file:

```
proc Varshni_term {T} {
      return [expr {1.0*[_alpha]*$T*$T/($T+[_beta])}]
}
proc Eg_difference {T1 T2} {
      return [expr {[Varshni_term $T1] - [Varshni_term $T2]}]
}
```

The difference in valence band level can be estimated as negative one half of the difference in bandgap. [32] The difference in the conduction band level is positive one half of the difference in bandgap, and the electron affinity difference is negative one half of this difference. If these inferences seem inappropriate, the heterostructure band offset must be determined near or at the simulation temperature, in which case, no temperature effect needs to be estimated. Reports of

valence band offsets given at 0 K alongside values of α and β have implemented T dependence in this manner.

```
proc Chi_difference {T1 T2} {
      return [expr {-0.5*[Eg_difference $T1 $T2]}]
}
proc vb_difference {T1 T2} {
      return [expr {-0.5*[Eg_difference $T1 $T2]}]
}
```

This function implements the valence band difference, however, the offset depends on the reference material as well. The diagram depicts these changes, where α_{sub} and β_{sub} are the Varshni coefficients for the reference material (called the substrate,) and α_{mat} and β_{mat} are for the material in question. Numerical values are in eV. The horizontal lines indicate energy levels. The reference level of -0.8 eV is used here to correspond with Vurgaftman *et al.* [44], who report band offsets for many materials.

GaAs at 0 K	Other material at 0 K	GaAs at T	Other material at T
$-0.8 + 1.5$	$-0.8 + \mathrm{vbo}(0) + E_{g\_mat\_0}$	$-0.8 + 1.5 - \frac{\frac{1}{2}\alpha_{sub}T^2}{T + \beta_{sub}}$	$-0.8 + \mathrm{vbo}(0) + E_{g\_mat\_0} - \frac{\frac{1}{2}\alpha_{mat}T^2}{T + \beta_{mat}}$
-0.8 [44]	$-0.8 + \mathrm{vbo}(0)$	$-0.8 + \frac{\frac{1}{2}\alpha_{sub}T^2}{T + \beta_{sub}}$	$-0.8 + \mathrm{vbo}(0) + \frac{\frac{1}{2}\alpha_{mat}T^2}{T + \beta_{mat}}$

Figure 16: Diagram of energy band edge levels to calculate temperature dependence of valence band offset with respect to GaAs reference.

The purpose of the diagram is to show the calculation of the valence band offset dependence on temperature and to ensure that each term receives the correct sign. Bandgaps decrease with increasing temperature, conduction band levels decrease, electron affinities increase, and valence band levels increase.

The valence band offset as a function of temperature is the difference between the 7th and 8th cells, assuming that T_{par} is 0 K:

$$vbo(T) = vbo(0) + \frac{\frac{1}{2}\alpha_{mat}T^2}{T + \beta_{mat}} - \frac{\frac{1}{2}\alpha_{sub}T^2}{T + \beta_{sub}}$$

This expression can be extended to the case where the reference temperature, T_{par}, need not be 0 K, but can be any value:

$$vbo(T) = vbo(T_{par}) + \frac{\frac{1}{2}\alpha_{mat}T^2}{T+\beta_{mat}} - \frac{\frac{1}{2}\alpha_{mat}T_{par}^2}{T_{par}+\beta_{mat}} - \left(\frac{\frac{1}{2}\alpha_{sub}T^2}{T+\beta_{sub}} - \frac{\frac{1}{2}\alpha_{sub}T_{par}^2}{T_{par}+\beta_{sub}}\right)$$

In the code, the following is sufficient:

$$vbo(T) = vbo(T_{par}) + vb\_difference(T, T_{par}) - substrate\_vb\_difference(T, T_{par})$$

In TCL:

```
proc vbo_difference {T1 T2} {
    return [expr {\
        [vb_difference $T1 $T2] - [${::substrate}::Bandgap::vb_difference $T1 $T2]\
    }]
}
```

The TCL expressions above are implemented as small TCL procedures. They are implemented in a general manner, and are used without changes in different materials. This approach is similar to inheritance, however, at present, the general procedures are copied into every location that they are used. Any function that can be written to be independent of material is useful in the development and maintenance of the material parameter models, and is less likely to require adjustment when it performs a small task. The above functions are also shown in Appendix A.

3-7: SiGeSn Electron Affinity

The above functions require a material-specific implementation of functions that return the Varshni coefficients. Many of the coefficients important to SiGeSn are reported in [45]. The parameters of silicon are used to estimate α and β for the Δ bandgap. The underscores avoid a name collision with the namespace variables expected by MatPar.

```
proc is_direct {x y} {
    return [expr {[bandgap $x $y] == [Gamma_bandgap $x $y]}]
}
proc is_L_indirect {x y} {
    return [expr {[bandgap $x $y] == [L_bandgap $x $y]}]
}
proc is_X_indirect {x y} {
    return [expr {[bandgap $x $y] == [delta100_bandgap $x $y]}]
}
proc Gamma_or_L_param_bowing {pGeG pSiG pSnG pGeL pSiL pSnL pX} {
    set x 0.14
    set y [expr {$x/3.672}]
    if {[is_direct $x $y]} {
        return [SiGeSn_bowing_formula $x $y $pGeG $pSiG $pSnG]
    } elseif {[is_L_indirect $x $y]} {
        return [SiGeSn_bowing_formula $x $y $pGeL $pSiL $pSnL]
    } elseif {[is_X_indirect $x $y]} {
        return $pX
    } else {
        error "SiGeSn problem with is_direct, is_L_indirect, and is_X_indirect"
    }
}
proc _alpha {} {
    return [Gamma_or_L_param_bowing 5.82e-4 3.91e-4 0 4.774e-4 4.774e-4 0 3.91e-4]
}
proc _beta {} {
    return [Gamma_or_L_param_bowing   296   125     0    235    235     0   636]
}
```

The relative conduction band offset functions can also be written independently of the material, however, they depend on the bandgap, which requires that a bandgap procedure be defined.

```
proc _Tpar {} {
   return 300.0
}
proc _Eg0 {} {
   return [bandgap 0.14 [expr 0.14 / 3.672]]
}
proc _Eg {} {
   return [expr [_Eg0] + [Eg_difference [_Tpar] $::temp]]
}
proc delta_Eg {} {
   return [expr {[_Eg] - ["${::substrate}::Bandgap::_Eg"]}]
}
proc _cbo_absolute {vbo} {
   return [expr {[delta_Eg] + $vbo}]
}
proc _cbo_relative {cbo_absolute} {
   return [expr {$cbo_absolute/[delta_Eg]}]
}
proc convert_vbo_eV_to_cbo_relative {vbo_eV} {
   return [_cbo_relative [_cbo_absolute $vbo_eV]]
}
```

Wang *et al.* [46] work from a band offset model for SiGeSn developed initially by Moontragoon, Soref, and Ikonik. [42] It incorporates Jaros's model [47], which produces approximate data for most semiconductors. Additionally, it quantifies spin-orbit splitting as a function of the composition variables, and places the top of the valence band accordingly. Below, it is implemented to find the band offsets with respect to pure germanium. According to [48] the band offset between unstrained SiGeSn with respect to unstrained Ge can be expected to be -0.05 eV.

```
proc deltaEv_SiGeSn_Ge_av {Si Sn} {
   return [expr {-0.48*$Si + 0.69*$Sn}]
}
proc E_spin_orbit_splitting {Si Sn} {
   return [expr {0.295*(1-$Si-$Sn) + 0.043*$Si + 0.800*$Sn}]
}
proc deltaEv_SiGeSn_Ge {Si Sn} {
   return [expr {\
      [deltaEv_SiGeSn_Ge_av $Si $Sn] \
      + [E_spin_orbit_splitting $Si $Sn]/3.0 \
      - [E_spin_orbit_splitting 0 0]/3.0 \
   }]
}
# [42]
```

The offset of germanium band edges with respect to GaAs are found in the internal photoemission experiments involving Al_2O_3 by Afanas'ev [49]

```
proc deltaEv_Ge_GaAs {} {
   set deltaEc_Ge_GaAs [expr {-(2.2-2.0)}] ;# [49]... -0.2 eV
   return [expr {$GaAs::Bandgap::Eg + $deltaEc_Ge_GaAs - $Ge::Bandgap::Eg}] ;# 0.55
   # We've assumed that GaAs and Ge have been initialized
```

```
    # GaAs is the "substrate" and Ge is the bottom material,
    # which gets run after the "substrate"
}
proc deltaEv_SiGeSn_GaAs {Si Sn} {
    return [expr {[deltaEv_SiGeSn_Ge $Si $Sn] + [deltaEv_Ge_GaAs]}] ;# 0.50
}
```

The band offset related functions can be located alongside the bandgap functions in the bandgap namespace, or the code can be located in the top level tcl file, as is true for any material parameter code. However, the material parameter model infrastructure does not support the passing of function arguments. Therefore, in order to supply the code with the information it needs to set and print X_0, a global variable must be used unless the band offset is calculated within the bandgap namespace. If the global variable approach is used, it is very possible that a band offset variable intended for one material could be inadvertently read and used by another material. It is commonly known that global variables must be employed with care. One approach to reduce the risk of inadvertent effects is to manually implement the variable scoping infrastructure implied by the usage of functions or namespaces of nearly any programming language. As a preliminary action, the global variable can be considered and checked for existence, and if it does exist, its value can be saved on a stack, called the function call stack. Then, the same variable name can be used as needed to carry the band offset value into the bandgap model code. At the bottom of the top level tcl file, the global variable's value should be restored if it existed, or, otherwise, the variable should then be deleted with the `unset` command. If the whole project follows this pattern, the global variable containing a value from a previously executed material will not exist.

This general-purpose logical block supports many possible global variables as inputs, and converts the given quantity to the relative conduction band offset "`cbo`." Existing code follows through with the calculation of X_0.

```
if {[info exists ::0K_vbo]} {
  set ::cbo [_cbo_relative [_cbo_absolute [expr {$::0K_vbo + [vbo_difference $::0K_vbo $::temp]}]]]
} elseif {[info exists ::Tpar_vbo]} {
  set ::cbo [_cbo_relative [_cbo_absolute [expr {$::Tpar_vbo + [vbo_difference $::Tpar_vbo $::temp]}]]]
} elseif {[info exists ::vbo]} {
  set ::cbo [_cbo_relative [_cbo_absolute $::vbo]]
} else {
  puts stderr "$::material init did not find any heterostructure band discontinuity value - will use bulk
data"
}
```

The other approach, employed for SiGeSn, is to only deal with the band offset calculation at the pardb level, within the bandgap namespace. This requires checking the values of the global MatPar variable `::substrate` and acting accordingly. An effect of this is that the edge case of different projects that use the same substrate material but different band offset will need to implement another version of the model. The MatPar infrastructure is designed to allow many model files to be available on hand.

```
if {[info exists ::substrate]} {
  if {$::substrate == "GaAs"} {
    set cbo [convert_vbo_eV_to_cbo_relative [deltaEv_SiGeSn_GaAs $x $y] $Eg]
  } elseif {$::substrate == "Ge"} {
    set cbo [convert_vbo_eV_to_cbo_relative [deltaEv_SiGeSn_Ge $x $y] $Eg]
  } else {
    puts stderr "SiGeSn band offset code not being applied"
  }
```

}

These levels will affect the resultant electron and hole currents in the device.

3-8: Lattice Constant Matching Germanium

The lattice constant of $Si_xGe_{1-x-y}Sn_y$ alloys exhibits less bowing with respect to composition than the energy levels. Assuming no bowing, the compositions that lattice match to germanium can be solved for algebraically. [50]

$Ge = 5.6573$ Å; $Si = 5.4307$ Å; $Sn = 6.4892$ Å;

$a_{SiGeSn} = (1 - x - y)Ge + xSi + ySn$ [51]

$$x = \frac{\cancel{Ge} - yGe + ySn - \cancel{a_{SiGeSn}}}{Ge - Si}$$

$$\frac{x}{y} = \frac{Sn - Ge}{Ge - Si} = 3.67$$

However, non-zero bowing terms have been reported for SiGeSn. [45][52][53] Therefore, the above rule holds only as an approximation. The effect is solved for in Appendix B and shown in Figure 17.

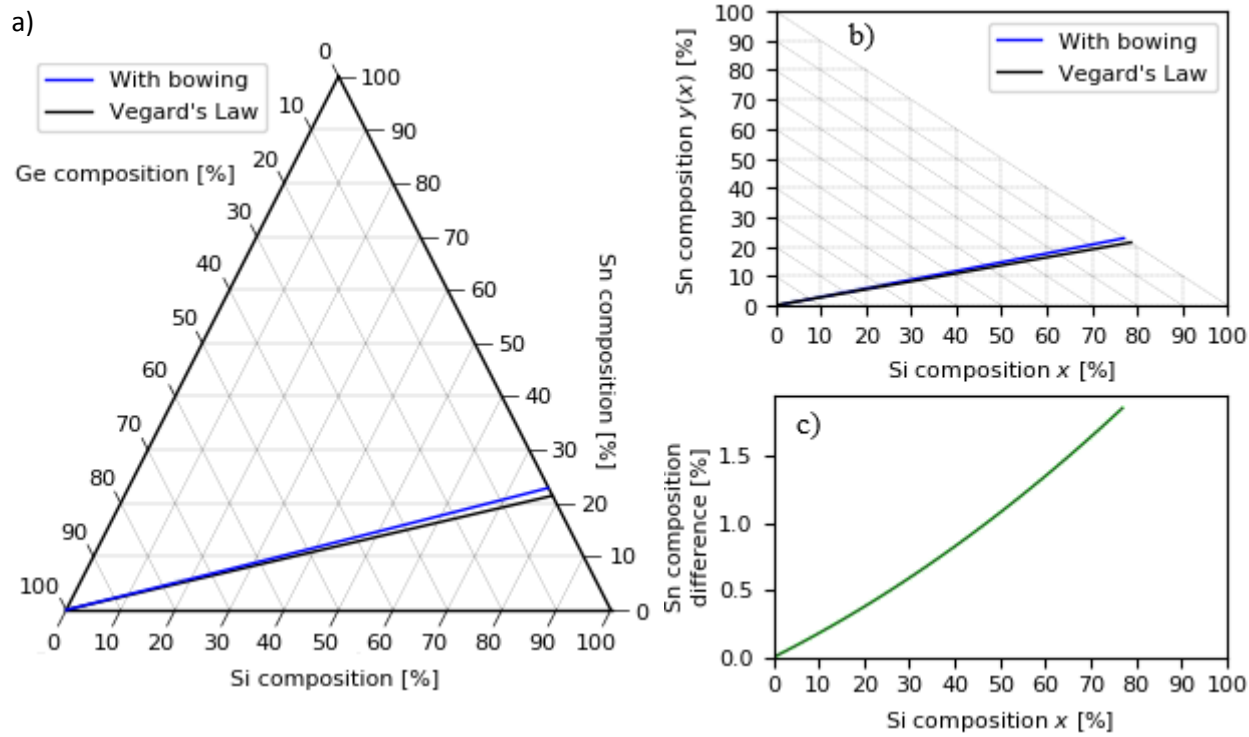

Figure 17: SiGeSn compositions that lattice match to germanium a) on triangular axes b) on rectilinear axes. c) Difference due to bowing term in tin composition needed to lattice match to germanium.

3-9: Permittivity

The permittivity material parameter, `epsilon` (ε,) of SiGeSn, in Sentaurus, is the wavelength-independent value that is applicable in the infrared range and larger. Lower wavelengths, in the

region of interest, will override this value, using the complex refractive index. It can be easily implemented in the material parameter database, but is not used in this simulation. It can be estimated with linear interpolation of the permittivities of silicon, tin, and germanium. [36] The '_' prefix in the procedure name prevents a name collision with the namespace variable "`epsilon`" that will be read by MatPar. If it seems unfortunate that a single file produces a single number, as is the case for the permittivity of germanium, the top-level TCL file can contain the parameter directly, instead of referring to subdirectories of *pardb*, such that the information is not hidden when reading the top-level file. The following code is placed within the `SiGeSn::Epsilon` namespace.

```
proc _epsilon {x y} {
      set Ge 16.2
      set Si 11.9
      set Sn 24.0
      return [SiGeSn_bowing_formula $x $y $Ge $Si $Sn]
}
proc init {} {
      variable epsilon [_epsilon 0.14 [expr {0.14/3.671227}]]
}
```

3-10: Electron Effective Mass

The effective mass in solid state physics is a proportionality factor relating the amount of kinetic energy possessed by the carrier to its "crystal momentum." The adjective "effective" is used to differentiate from the real physical mass of the electron. Due to the potentials within the crystalline environment, the kinetics of the electron are more easily described using this effective mass, while the real mass of the electron is sidelined. The effective mass is usually expressed as a factor of normal electron mass, so the unit becomes the normal electron mass m_0. The adjective "crystal" in "crystal momentum" is analogous. In this way, these kinetic quantities found within the crystal do not equal the normal mass or momentum in absolute value. The moving electrons within a semiconductor do have absolute momenta and similar physical masses, however their relations with respect to energy are adjusted by the crystalline environment, and may require for example less energy to reach a given velocity. These differences with respect to kinetics within a vacuum environment are demonstrated by models that use language such as crystal momentum, and effective mass.

The effective mass that an electron in the conduction band exhibits depends upon which energy state it occupies, because it corresponds with the curvature of a band edge of an $E(\vec{k})$ function. A large energy change with a small change of momentum (large curvature,) occurs when the carrier has small effective mass (inertia). Therefore, the effective mass of the conduction band electron is inversely proportional to the curvature of the $E(\vec{k})$ curve. Each of the L, Δ, and Γ $\vec{k}$-space locations have conduction band minima with different curvatures and therefore different electron effective masses. Additionally, the L and Δ iso-energy surfaces in a 3D $E(\vec{k})$ plot are anisotropic surfaces, with different curvatures along different axes, leading to transverse and longitudinal quantities for effective mass, and resulting in and from different responses to forces in different directions by carriers in those states.

These quantities are used to calculate the density of states, by way of an intermediate quantity called the "density of states effective mass," that depends on the transverse and longitudinal effective masses, and is also increased by the multiplicity of states at the given $\vec{k}$-space location. The Γ conduction band valley is symmetric in $\vec{k}$-space and has multiplicity of 1, therefore, the result of the equation is that the density of states effective mass equals the conduction band effective mass. This affects tabulated material parameters, such that the anisotropic effective masses only apply to indirect locations in $\vec{k}$-space, while the effective masses of in the isotropic direct regimes only have one value ($M = 1;\ m_c = m_l = m_t$).

$m_c = M^{2/3}(m_l m_t^2)^{1/3}$ [54]

Therefore, in the model implementation code, we have the choice of treating the effective mass in the Γ regime as a simpler special case that does not use the above equation, or we may implement the equation similarly for all cases, and setting $M = 1$ and $m_l = m_t$ in the Γ case. Either way, the equation applies to the L and Δ cases.

The model implementation is shown in Appendix C. A result of this model is to infer the effective mass, if 14% silicon and 3.8% tin are given, which corresponds to the germanium-like L-type bandgap region. The output is $0.567m_0$, an increase from the electron effective mass of germanium, $0.547m_0$.

3-11: COMPLEX REFRACTIVE INDEX

Ellipsometric experimental results for SiGeSn are reported by Pearce *et al.* [50]. The imaginary part of the complex refractive index, the extinction coefficient, K, describes the absorption characteristic of the material and is important to determine the thickness of the layers that will be required. It relates to the absorption according to $K = \alpha/2k_0 = \alpha\lambda_0/4\pi = hc\alpha/4\pi E$, where λ_0 is the incident wavelength. These quantities are all highly dependent upon the photon energy or wavelength of interest, and their onset relates to the band structure of the material. In the transparent region of low photon energy, ϵ_2 is zero, and K is also zero. At the onset of absorption, the measured ϵ_2 becomes non-zero, and this is carried in the output of the square root function as a non-zero value of K. It follows that the region near the onset of light absorption corresponds to complex numbers of both quantities in the first quadrant.

Instead of employing a complex square root function, the following approach can be used:

$$|\epsilon_{relative}| = \sqrt{\epsilon_r^2 + \epsilon_i^2}$$

$$n = |\epsilon_{relative}|^{1/2}\, cos\, tan^{-1}(\epsilon_1, \epsilon_2)$$

$$K = |\epsilon_{relative}|^{1/2}\, sin\, tan^{-1}(\epsilon_1, \epsilon_2)$$

where the two-argument $\tan^{-1}$ function supports the full output range of 2π radians, and the square root operators can be assumed to be positive.

The measurements of [50] result in a plot of α for the 0.7 to 1.1 eV range, and of the complex dielectric function $\epsilon_{\text{relative}}$ for the range up to 5 eV. The source data was read from the images with the *WebPlotDigitizer* tool. [55] This information must be converted into a table of n and K

for the Sentaurus Device `par` file. The values must be listed in increasing order with respect to the wavelength. Since the α plot only produces K, corresponding n values are filled in by interpolating the n result from the $\epsilon_{\text{relative}}$ plots.

The following plot shows how the calculated K values obtained from the α plot compare with the K data inferred from the $\epsilon_{\text{relative}}$ plots. The data from α is considered more accurate and is selected for energies below the crossover point. The resulting n and K values are shown.

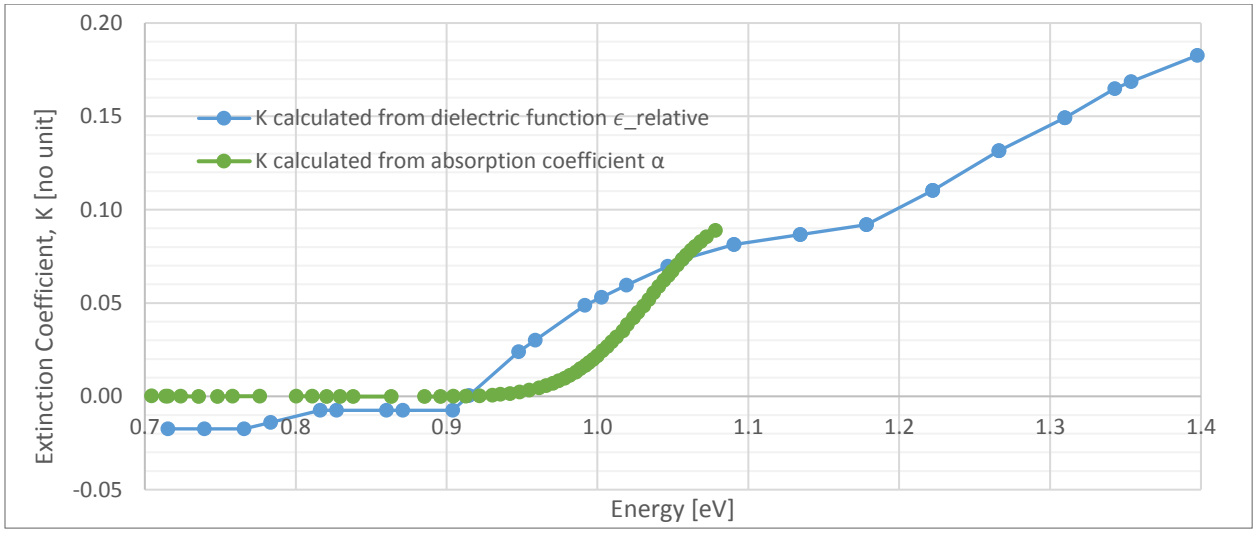

Figure 18: Calculated K result based on data from the two methods in [50] *versus energy.*

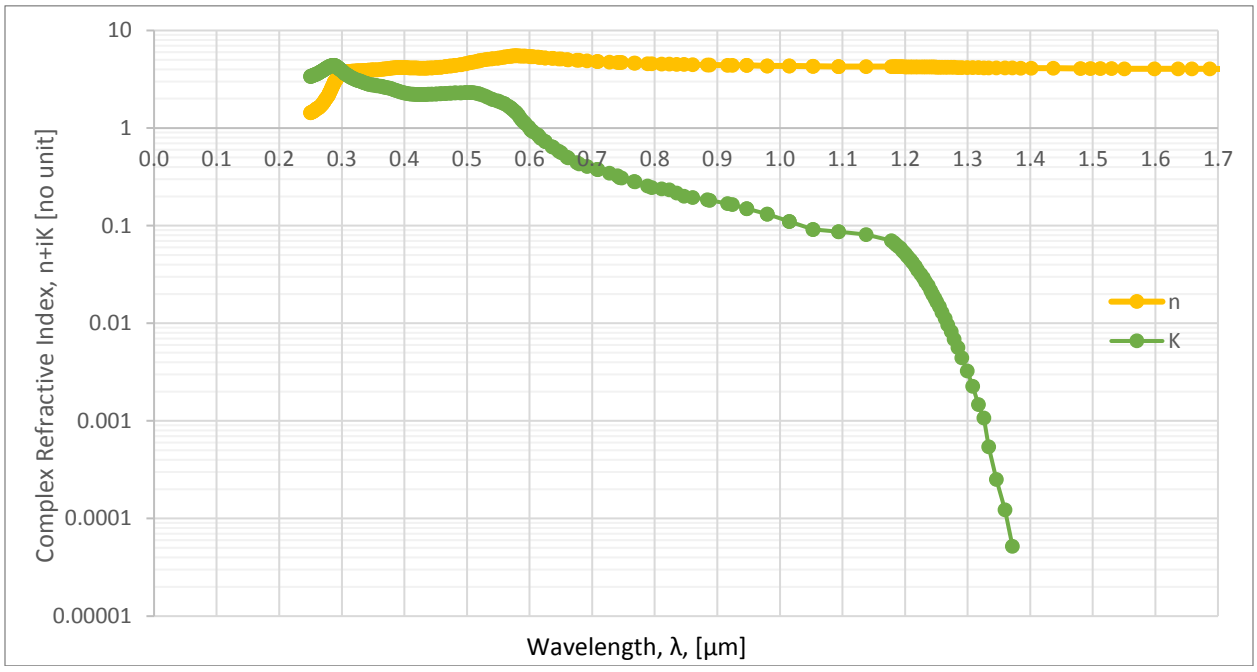

Figure 19: Calculated complex refractive index of SiGeSn versus wavelength. Composition is estimated in [50] *as 14 atomic % Si, lattice matched to germanium.*

4: SIMULATION TOOLFLOW

The previous sections dealt with SiGeSn material property model development, which falls within the domain of the tool called MatPar. The other tools in the chain are called Epi, the Sentaurus Structure Editor (also called SDE, as in device editor,), Sentaurus Device, and finally, for visualizations, Sentaurus Visual and python. The Sentaurus Workbench provides tools with which to configure the simulation. This includes the definition of experiment variables or parameter sweeps, which can be defined in a table in the workbench, and access to edit the programming input "command" files to the tools in the chain. This project organization can be called an *Epi/MatPar* style project. Possible project-level approaches are documented in the *Epi/MatPar* manual [32].

4-1: EPI

The Epi module can be used for devices that are made from planar layers. It streamlines the structure definition.

4-1-1: SIMULATION WIDTH

The variable `wtot` specifies for Epi the width of the layers. This is set to half of the grid pitch, the spacing between contacts, since the symmetry can be used to reduce the simulation space by half. The grid pitch is set to 120 μm, with a contact coverage factor of 3%, which is appropriate for a 7mm x 7mm device. [56] [57]

This width is required for calculations in post-processing, so it should be widely scoped, such that the same value can be referenced in the many places it is needed. A workbench parameter w is defined as the width between contacts, and then Epi's `wtot` is set to $w/2$. This arrangement allows the input to match the grid pitch, not half the grid pitch, and is a clear variable naming scheme that simplifies calculations and changes related to the device's dimensions. It implies whether the half-width is being used, as in the simulated structure, or the whole width, as in contexts other than the structure definition, such as the setting of input variables according to real device dimensions. This halving is also applied automatically to the contact width, and the contact width variables are named similarly, such that the quantity being used is obvious. The term "`tot`" in the subscript for total is only appropriate in the context of the layer creation.

The efficiency calculation in post-processing also requires the use of the device half-width, $w/2$, since the incident power, the denominator of the efficiency calculation, is related to the exposed surface, which is $w/2$ in the simulation. The output current, the numerator, is implicitly halved along with the simulated region. Current density, J, is then calculated by dividing the output current by the simulated area, including the 3rd dimensional length parameter, which can be input to Sentaurus Device via the `AreaFactor` parameter. A workbench variable for `AreaFactor` is therefore also created, such that it can be changed if necessary, and the change will be reflected as necessary in the coupled points of the code: the Sentaurus Device command file, Sentaurus Visual visualization and calculations, and the python visualization and calculations. Then, any change to any device dimension is automatically reflected wherever necessary. This facilitates testing as well as the redefinition of the device dimensions. The `AreaFactor` length dimension affects the output current, but has no effect on the total current density. The following line translates the workbench

variable *w* into an Epi variable. The `!( )!` braces call TCL code that executes the division by 2 and prints the result as the file is preprocessed for Epi.

```
$global wtot=!(puts [expr @w@ / 2.0])!
```

4-1-2: LAYER CSV TABLE

The Epi command file consists primarily of a table of layers written in a CSV (comma separated variable) text file format.

Table 3: Example Epi CSV Table.

```
#_____________________________________________________________________________________________________
#|  LAYER    |           |             |           |          |          |      MESH REFINEMENT    |       |
#|  REGION   | MATERIAL  |    FILE     | THICKNESS | DOPING   |   MOLE   |(xref,yref,elsize,mbox)  | LAYER|
#|  LABEL    |           |             |   (um)    | (/cm^3)  | FRACTION |mbox melsize ratio direc|NUMBER|
#|___________|___________|_____________|___________|__________|__________|________________________|______|
     arc1    ,    MgF    ,    MgF.par  ,   0.100   ,          ,          ,                        ,;#01,
     arc2    ,    TiOx   ,   TiOx.par  ,   0.060   ,          ,          ,                        ,;#02,
   sc1_fsf   ,    AlInP  ,   AlInP.tcl ,   0.020   ,   6e18   ,   0.50   ,                        ,;#03,
   sc1_em    ,    GaInP  ,   GaInP.tcl ,   0.025   ,   5e18   ,   0.50   ,                        ,;#04,
   sc1_base  ,    GaInP  ,   GaInP.tcl ,   0.064   ,  -1e17   ,   0.50   ,                        ,;#05,
   sc1_bsf   ,  AlGaInP  , AlGaInP.tcl ,   0.020   ,  -3e19   ,   0.25   ,                        ,;#06,
   td1_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.022   ,  -2e20   ,   0.30   ,                        ,;#07,
   td1_nhigh ,    GaInP  ,   GaInP.tcl ,   0.022   ,   6e19   ,   0.50   ,                        ,;#08,
   sc2_fsf   ,    AlInP  ,   AlInP.tcl ,   0.020   ,   6e18   ,   0.50   ,                        ,;#09,
   sc2_em    ,    GaInP  ,   GaInP.tcl ,   0.10    ,   5e18   ,   0.50   ,                        ,;#10,
   sc2_base  ,    GaInP  ,   GaInP.tcl ,   1.90    ,  -1e17   ,   0.50   ,                        ,;#11,
   sc2_bsf   ,  AlGaInP  , AlGaInP.tcl ,   0.030   ,  -3e19   ,   0.25   ,                        ,;#12,
   td2_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -2e20   ,   0.30   ,                        ,;#13,
   td2_nhigh ,    GaInP  ,   GaInP.tcl ,   0.030   ,   6e19   ,   0.50   ,                        ,;#14,
   sc3_fsf   ,   AlGaAs  ,  AlGaAs.tcl ,   0.020   ,   1e19   ,   0.80   ,                        ,;#15,
   sc3_em    ,   InGaAs  ,  InGaAs.tcl ,   0.040   ,   3e18   , 0.01173  ,                        ,;#16,
   sc3_base  ,   InGaAs  ,  InGaAs.tcl ,   0.302   ,  -1e17   , 0.01173  ,                        ,;#17,
   sc3_bsf   ,   AlGaAs  ,  AlGaAs.tcl ,   0.020   ,  -1e19   ,   0.20   ,                        ,;#18,
   td3_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -2e20   ,   0.30   ,                        ,;#19,
   td3_nhigh ,    GaInP  ,   GaInP.tcl ,   0.030   ,   6e19   ,   0.50   ,                        ,;#20,
   sc4_fsf   ,   AlGaAs  ,  AlGaAs.tcl ,   0.020   ,   1e19   ,   0.80   ,                        ,;#21,
   sc4_em    ,   InGaAs  ,  InGaAs.tcl ,   0.10    ,   3e18   , 0.01173  ,                        ,;#22,
   sc4_base  ,   InGaAs  ,  InGaAs.tcl ,   1.15    ,  -1e17   , 0.01173  ,                        ,;#23,
   sc4_bsf   ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -1e19   ,   0.20   ,                        ,;#24,
   td4_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -1.2e20 ,   0.30   ,                        ,;#25,
   td4_nhigh ,     GaAs  ,    GaAs.tcl ,   0.030   ,   6e19   ,          ,                        ,;#26,
   sc5_fsf   ,   SiGeSn  ,  SiGeSn.tcl ,   0.015   ,   2e19   ,          ,                        ,;#27,
   sc5_em    ,   SiGeSn  ,  SiGeSn.tcl ,   0.040   ,   1e19   ,          ,                        ,;#28,
   sc5_base  ,   SiGeSn  ,  SiGeSn.tcl ,   0.298   ,  -1e18   ,          ,                        ,;#29,
   sc5_bsf   ,   SiGeSn  ,  SiGeSn.tcl ,   0.020   ,  -2e18   ,          ,                        ,;#30,
   td5_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -1.2e20 ,   0.30   ,                        ,;#31,
   td5_nhigh ,     GaAs  ,    GaAs.tcl ,   0.030   ,   6e19   ,          ,                        ,;#32,
   sc6_fsf   ,   SiGeSn  ,  SiGeSn.tcl ,   0.020   ,   2e19   ,          ,                        ,;#33,
   sc6_em    ,   SiGeSn  ,  SiGeSn.tcl ,   0.10    ,   1e19   ,          ,                        ,;#34,
   sc6_base  ,   SiGeSn  ,  SiGeSn.tcl ,   1.00    ,  -1e18   ,          ,                        ,;#35,
   sc6_bsf   ,   SiGeSn  ,  SiGeSn.tcl ,   0.100   ,  -2e18   ,          ,                        ,;#36,
   td6_phigh ,   AlGaAs  ,  AlGaAs.tcl ,   0.030   ,  -1.2e20 ,   0.30   ,                        ,;#37,
   td6_nhigh ,     GaAs  ,    GaAs.tcl ,   0.030   ,   6e19   ,          ,                        ,;#38,
   sc7_fsf   ,   SiGeSn  ,  SiGeSn.tcl ,   0.030   ,   2e19   ,          ,                        ,;#39,
   sc7_em    , Germanium ,     Ge.tcl  ,   0.10    ,   1e19   ,          ,                        ,;#40,
   sc7_base  , Germanium ,     Ge.tcl  ,   13.0    ,  -7e17   ,          ,                        ,;#41,
   sc7_bsf   , Germanium ,     Ge.tcl  ,    1.0    ,  -5e19   ,          ,                        ,;#42,
```

These layers define rectilinear material profiles. Epi does support an inline function to create an error function shape. However, this function is not sufficient, since a constant value term must also be added. Doping by ion implantation creates an error function plus constant profile if the material

has an initial level of doping. The rectangular doping profile implemented here is a sufficient approximation. If Epi's capabilities are ever found to be limited, the task may still be possible to do with SDE.

4-1-3: INTERFACES

In addition to the layer CSV table, there are some other related tasks that Epi handles. As the materials are specified in the table, their material parameter files must also be specified. This will inform MatPar which files to include as it compiles all of the material parameters into a single file for Sentaurus Device. It is at this point that many material parameter dependencies are resolved, such as temperature, doping, or mole-fraction dependent models. If two layers have different doping levels, or alloy component amounts, MatPar will calculate separate parameter sections for them and supply them to Sentaurus Device, such that it can solve the physics appropriately for the different regions. Epi links material parameter files within the layer table on a per-region basis in this way.

There is another option, below the table, where material parameter files can be linked in a material-wise manner instead of region-wise as is done in the table. This is most applicable if there are many layers with the same specifications, and has other uses, but is it not particularly applicable in this project, since the device is defined primarily in the table, and each region usually has different parameters. However, interface surface parameters are specified in the bottom section, in a manner similar to material-wise specifications, since they do not have a place in the table.

```
,      TiOx/AlInP     , TiOx_AlInP.par
,     AlInP/GaInP     , surface_recombination.par
,     GaInP/AlGaInP   , surface_recombination.par
,    AlGaAs/GaAs      , GaAs_AlGaAs.par
,    AlGaAs/InGaAs    , GaAs_AlGaAs.par
,      GaAs/SiGeSn    , surface_recombination.par
,    InGaAs/SiGeSn    , surface_recombination.par
,    SiGeSn/SiGeSn    , surface_recombination.par
,    SiGeSn/Germanium , surface_recombination.par
,     GaAs/Germanium  , surface_recombination.par
,   InGaAs/Germanium  , surface_recombination.par
,Germanium/Germanium  , surface_recombination.par
```

In this project, all the surfaces between a field layer and an active layer are linked to the same material parameter file that defines surface recombination parameters. The surface field layers alleviate surface recombination at the edge of the subcell due to their injection with higher dopant concentration that strongly repels the minority carriers, thereby reducing the possibility of any recombination. However, the interface between the active region and the surface field layer can then become the point of surface recombination. This is where the surface recombination is calculated in this project. Additionally, it is possible that there will be a high degree of surface recombination at the interface between the group IV growth and the III-V materials. This can be controlled in the simulation by creating a surface parameter file that specifies surface recombination coefficients, and linking it in the interface section of the Epi command file. The surface recombination coefficient is associated with a workbench parameter, such that its effect can be monitored at different values.

```
SurfaceRecombination {
   S0   = @srv@, @srv@  *[cm/s]    *Electrons, holes
   Sref = 0             *[no unit] *No doping dependence
   * Sref is the multiplier for doping dependence
   *        /           Ni  gamma \
   * s = s0 | 1 + sref(----)      | [58]
   *        \          Nref      /
}
```

4-1-4: DATA STRUCTURE SETUP

The ideal project organization has the Epi step set everything in motion. A primary aim of this particular project is to generalize the input commands of many of the modules such that they do not depend on the number of layers in the device. It would be useful to delete or add a layer and have all subsequent modules respond properly, and not overlook setting important physics, or calling an undefined identifier. Without such an approach, every tool is decoupled, and any small change to the device must be replicated in several locations. Such support is necessary to explore segmented designs with simulation, since they have a large number of layers. There are many tasks throughout the project that can be implemented in such a loop that operates on every layer or surface of the device, including setting up the meshing in SDE, creating virtual contacts within the device for performance measurement in SDE, turning on physics models like recombination in Sentaurus Device, or code that generates plots of calculated quantities like generation or recombination.

One approach is to follow a naming convention within each subcell, and to define a global variable that sets the number of subcells. Then, assume every subcell has the four normal elements, and that the tunnel layers also exist in the predictable locations and have predictable region names. This creates the ability to loop over the device according to the given number of subcells and naming assumptions. Often, it is more useful to supply a global TCL list containing all region labels. This is a proper use of a global variable, since many modules need access to that information. Then, the naming convention, and the added constraint of having the four normal layers in every subcell, is not required.

Epi and MatPar can supply a third alternative. Additional fields can be added into the Epi CSV table. MatPar can record these additional fields such that the subsequent modules can refer to them and implement code according to information set in the Epi table. The third approach is versatile, and applicable when some, but not all regions need one kind of physics, and a different set of regions might need another kind of physics, all to be described by the Epi table.

However, the first and second approaches applied together proved suitable here. Both the integer and the list are necessary, such that no assumptions need to be made about the layers in the device. The additional table column is not needed, since any process unique to any layers can be handled by using the region label.

These two approaches must use TCL code enclosed in `!( )!` brackets to write loops that print the necessary content for simulations. The TCL list of regions enables any task that must be done for every region. Other tasks applied on a per-subcell basis, must know the number of segments. TCL loops are general and powerful enough to handle the many specific tasks requiring this approach.

This requires printing the module command code from inside the `!( )!` brackets with the TCL `puts` (put string) command. Code that prints code is normal in software development, and also occurs in web programming, when server programs print web page code.

The global list variable called `epi_regions` containing all the region names is created alongside the table. Adjacent to the table, this list can easily remain synchronized with the table. It supports the reduction or elimination of the need to make changes throughout the project as different device designs are explored. Similarly, a list of important surface names is initialized here, which facilitates the definition of surface recombination.

```
!(
   set num_segments 7
   set epi_regions {                                        \
      sc1_fsf sc1_em sc1_base sc1_bsf td1_phigh td1_nhigh \
      sc2_fsf sc2_em sc2_base sc2_bsf td2_phigh td2_nhigh \
      sc3_fsf sc3_em sc3_base sc3_bsf td3_phigh td3_nhigh \
      sc4_fsf sc4_em sc4_base sc4_bsf td4_phigh td4_nhigh \
      sc5_fsf sc5_em sc5_base sc5_bsf td5_phigh td5_nhigh \
      sc6_fsf sc6_em sc6_base sc6_bsf td6_phigh td6_nhigh \
      sc7_fsf sc7_em sc7_base sc7_bsf                     \
      arc1 arc2                                           \
   }
   set epi_inner_subcell_surfaces {                                    \
      sc1_fsf/sc1_em sc1_base/sc1_bsf sc2_fsf/sc2_em sc2_base/sc2_bsf \
      sc3_fsf/sc3_em sc3_base/sc3_bsf sc4_fsf/sc4_em sc4_base/sc4_bsf \
      sc5_fsf/sc5_em sc5_base/sc5_bsf sc6_fsf/sc6_em sc6_base/sc6_bsf \
      sc7_fsf/sc7_em sc7_base/sc7_bsf arc2/sc1_fsf                    \
   }
)!
```

4-1-5: MESH REFINEMENT

Finally, the Epi command file also typically concerns itself with setting mesh variables.

The mesh is the network of points on which the differential equations are solved. A fine mesh, with a greater density of points, is required in regions of the device where the physical quantities are highly non-linear. The term "element" is used here and has the same meaning as it does in calculus. In calculus, limits are taken as the differential element size approaches zero. The device differential equations cannot be solved completely in this manner, however, the numerical methods offer a way to a solution. Instead of finding an analytical solution in the limit as the differential element size approaches 0, the approach of the simulation is to use small non-zero, non-infinitesimal element sizes, allowing a solution to be found. Having the differential element reach a size of 0 would lead to an infinite number of calculations. Some form of error is introduced by having non-zero sized differential elements. The mesh defines the element sizes throughout the device. The mesh spacing must be controlled, such that it is dense in areas where any derivative is highly dynamic, but not dense everywhere unnecessarily, because there is also a limit in the accessible amount of time the computation takes.

Epi has special control to set the mesh along the growth dimension y within the device layers, but has limited control in x, across the device. Epi supports many functions inline within the table rows, including mesh specification in a layer, as well as doping and mole fraction profile functions. A useful mesh function is called a *multibox*. It, along with the other mesh functions, are more appropriately called mesh refinement functions, since they only define constraints on the mesh. The final mesh definition is left to internal functions that work more specifically on the set of mesh nodes, while ensuring that the mesh is non-Delaunized. [59][58] The multibox can set mesh refinement very finely on one edge of a square, and increases mesh spacing away from the edge.

The multibox function asks for a specification of what the largest and smallest mesh spacings should be. There are variables to set global constraints as well, that apply to the entire device (`dXmax, dYmax, dXmin, dYmin`). It also asks which sides of the box should use the tighter values. Another set of arguments are `Xratio` and `Yratio`, that specify the factor by which the distance between each layer of mesh points increases, and must lie between 1 and 2. [32]

Rather than specifying these settings individually for every layer, the approach to meshing has been systematized. A use case that often occurs is that the mesh needs to be changed globally. It makes sense to run the simulation with a coarse mesh to obtain approximate results quickly, or a fine mesh to control sources of error. In a device with many layers, this means editing all the rows of the table, only to affect the y dimension mesh. This can be automated, since the multibox arguments can be set according to a rule or pattern. Another reason to change the approach is that it is difficult to properly specify the mesh near the contact using Epi. The cap is created adjacent to the ARC (anti-reflection coating) layers after the Epi step, invalidating the mesh definitions that were created for the ARC. The approach will be to define multibox rectangles on the device, and to define the mesh with fewer arguments according to an algorithm. Subsequently, the performance of the mesh can be more easily tracked, since it is fully defined by fewer parameters. Additionally, the global constraint variables become unnecessary, as they should simply equal the constraints on the region with the largest elements.

The systematic approach for defining the mesh refinement for y and x simultaneously falls within the scope of the next module in the tool chain.

4-2: SENTAURUS STRUCTURE EDITOR

The Sentaurus Structure Editor (SDE) is the second module in the project's toolflow. The acronym seems to be derived from a previous name, Sentaurus Device Editor. It is a very powerful tool that supports a GUI for defining device structures, in 3D if necessary, and every action can be scripted in its Scheme-based language. [60]

The role of SDE is to complete the definition of the structure. Note that Epi's role was to automate the generation of code for SDE. At a minimum, this second module must load and run that code. Afterwards, the tasks of the SDE command file are to finalize any physical definition of the device that Epi did not complete. SDE will output the structure definition in file formats familiar to MatPar and Sentaurus Device.

Epi can output SDE Scheme code that implements mesh refinement. It is possible to include Sentaurus Mesh as a module in the toolflow, but this is not necessary, since SDE also supports

meshing. Sentaurus Mesh has command file syntax similar to Sentaurus Device, and SDE will generate code of that syntax. The output of Epi can serve as an example of how the mesh is defined in SDE's Scheme language, with window definition, multibox definition, and multibox placement functions.

To implement the systematized mesh approach, the y-dimension mesh specification is removed from the Epi table, and TCL code is written to print the Scheme code that describes the desired mesh multiboxes. This code is shown in Appendix D.

`Define_1d_mesh_rectangle` is used in the 1D approximation of the simulation, which saves computation time by creating only two mesh points in x across the device, with the normal amount of points in y. A virtual contact will be placed on the front of the window layer. Alternatively, the 2D configuration of the simulation takes much more time, but is necessary to solve for any effects relating to the current moving laterally to reach the thin grid finger. The ARC layers are etched, and a small highly doped GaAs cap is placed on the window layer, with the virtual contact on the cap. This breaks the lateral symmetry, and requires much more meshing in the x-dimension and computation time. These two configurations are handled by the SDE module, since Epi only creates full-width layers.

In order to better resolve the optical generation, which changes greatly over a short distance in the front layers, an additional multiplier is applied to the front layer layers.

The rectangles are applied to create more meshing below the contact, where there is a step in the optical generation, from the shade. An additional rectangle the width of the half-contact is added to regulate the effect of the the exponential growth in mesh spacing implied by the `Xratio` control, which does not fit the changes in values as x increases away from the contact, regardless of the choice of value of `Xratio`.

4-2-1: CATHODE AND ANODE

The SDE module must create named virtual contacts. Epi has the ability to create contacts, but this is not done, since in the 2D mode, the cathode must be placed on the cap, which is created after Epi exits. Virtual contacts in SDE exist along a line or edge in a structure, or a surface in a 3D structure. A find-edge function is used to satisfy an argument of the define-contact function. The find-edge function simply requires coordinates, anywhere along the edge. To create a general-purpose create-contact function, an x-coordinate below the cap can be used, since this coordinate is common to all the contacts that must be made. Like the device width, the simulated cap is half the width of the actual cap, so the x-coordinate to use is anywhere exclusively between 0 and the cap half-width, or one quarter of the cap width. This approach supports the creation of contacts in a loop, which makes the code more robust if it is to be modified to simulate other device designs. Additionally, a global list of contacts should be maintained by the create-contact function. This is useful for post-processing, in support of similar algorithms that make no assumption of the number of segments or subcells in the device.

To determine the necessary Scheme commands, it can be useful to perform the task with the SDE GUI, which prints the Scheme of its actions, instead of working only from the functions in the manual.

```
(define half_w_contact (/ @w_contact@ 2.0))
!(
  global contact_list
  set contact_list [list]
  proc create_VContact {name yposition} {
    global contact_list
    lappend contact_list $name
    puts "(sdegeo:define-contact-set \"$name\" 4 (color:rgb 0 0 1))"
    puts "(sdegeo:set-current-contact-set \"$name\")"
   puts "(sdegeo:define-2d-contact (find-edge-id (position (/ half_w_contact 2.0) $yposition 0)) \"$name\")"
  }

  if {"@dimension@" == "1d"} {
    create_VContact "cathode" "Y0_sc1_fsf"
  } else {
    create_VContact "cathode" "(- Y0_sc1_fsf h_cap)" ;# 2d
  }
  create_VContact "anode" "Ybot"
)!
```

4-2-2: Additional Contacts

This code base is structured as in [3] and also includes other prior developments from within the University of Ottawa SUNLAB research group. [2][61] It is organized into three sub-projects, two of which will use the additional virtual contacts adjacent to every subcell to simulate the device, and one of which only has the normal two contacts on each side of the device, the cathode and anode. Along with the implementation of loops to enable the exploration of many designs, these three subprojects must also be coordinated. Having the three subprojects have copies of certain identical code blocks encumbers the code maintenance and development. The three subprojects use the same Epi file, and the same structure, with the exception of these contacts, and also the same material parameters.

The workbench's `#include` directive can be used to meet this requirement. The entire Epi command file is located at a common location and included, such that the three projects always simulate the same layers. There remain three individual Epi command files within the three different subprojects, each containing only one line that includes the common file, and the fact that there are three is never relevant. This is almost true for the SDE module. Most of the structure definition is located in the common directory and included in the same way, such as the mesh refinement, the cap creation, the create-contact function definition, and the function calls to create the cathode and anode. The creation of the additional contacts is therefore done within the sub-project, after the majority of the structure definition is done via the `#include`.

```
#include "../common/sde_dvs.cmd"
!(
  global num_segments
  for {set i 1} {$i < $num_segments} {incr i} {
    create_VContact "td${i}_VContact1" "Y0_td${i}_phigh"
    create_VContact "td${i}_VContact2" "Y1_td${i}_nhigh"
  }
  # Sort contact_list
  global contact_list
  proc lremove {L index} {
    return [lreplace $L $index $index]
  }
  set anode_index [lsearch $contact_list "anode"]
  set contact_list [lremove $contact_list $anode_index]
```

```
  lappend contact_list "anode"
)!
```

4-3: SENTAURUS DEVICE

More differences between subprojects arise in the Sentaurus Device step. The first subproject calculates subcell current-voltage characteristics under solar illumination by setting the voltages of the contacts. The second subproject makes a similar calculation for the entire device, and only creates two contacts. The third subproject holds the contacts at 0 Volts and ramps the wavelength of monochromatic illumination. Only voltage ramp function definitions are shared across the first two subprojects with the `#include` method, and they are called with different arguments afterwards.

The Sentaurus Device command file has several sections that are for settings, such as File, Electrode, Physics, Math, Plot, and sometimes Optics, then the Solve section specifies tasks that are executed sequentially. [58]

4-3-1: OPTICS

The first step is the optical calculation. This is done with the transfer matrix method (TMM,) which is specified in the Optics section. In the Solve section, the "Optics" command is issued. Internal to Sentaurus Device, a matrix is calculated for each layer, and these are used to determine the reflection and transmission coefficients for each interface in the device. The Beer-Lambert law is also used to determine the attenuation of the light as it travels through each medium. Light loses intensity as it penetrates according to the negative exponential function, and the absorption coefficient of the material, α. α is determined from the complex refractive index material parameter. All three subprojects use this method. The first two subprojects have nearly identical `optics_des.cmd` files. The `des` suffix is the suffix for Sentaurus Device.

The AM1.5D spectrum is used without any intensity scaling factors. Some processing of the spectrum data is necessary, since Sentaurus Device does not account for the spectral width of the wavelength bins, instead specifying that the data must be simple irradiance amounts at each wavelength, not spectral irradiance quantities that would have to be integrated to obtain irradiance. The AM1.5D standard data does not have constant wavelength bin size. This conversion of spectral irradiance to irradiance is done with Matlab in preprocessing.

The third subproject has much different simulated optics. It still uses TMM, but the irradiance is monochromatic, which is specified in the Optics section. The input is ramped in wavelength, described by different commands in the Solve section. The ramp solves the device at every point in the wavelength range. AM1.5D can be used in this subproject as well, after Sentaurus Device has finished. The wavelength dependent EQE determined by the current output can be multiplied with the spectral irradiance, divided by electron charge q, then integrated as another way to calculated the device's short-circuit current under AM1.5D.

4-3-2: POISSON, CONTINUITY, DRIFT AND DIFFUSION EQUATIONS

The optical solution leads to a generation term that contributes to conduction band electron and valence band hole (charge carrier) densities. However, first, the Poisson equation is solved for the electrostatic potential and electric field, $\vec{E}$.

$$\nabla \cdot \vec{E} = \rho/\epsilon \text{ [15]}$$

where ρ is charge density, ϵ is the permittivity of the material, as in $\epsilon = \epsilon_{\text{relative}}\epsilon_0$, where ϵ_0 is the permittivity of free space, and $\nabla \cdot$ is the vector divergence operator. The equation is specified in the code with the `Poisson` keyword, and the solver is invoked with the `Coupled` command. [58]

The keywords `Electron` and `Hole` are added in the following step, which add two equations, all to be solved together, where the solution of the first equation alone is used as the initial guess in the Newton algorithm. The electron and hole charge carrier concentrations and currents are found with the electron and hole continuity equations: [11]

$$\frac{\delta n}{\delta t} = \frac{1}{q}\,\nabla \cdot \overrightarrow{J_N} - r_N + g_N$$

$$\frac{\delta p}{\delta t} = \frac{1}{q}\,\nabla \cdot \overrightarrow{J_P} - r_P + g_P$$

At steady state, the temporal evolution of the carrier concentrations, $\delta n/\delta t$ and $\delta p/\delta t$, becomes zero, but prior to that, the other terms can contribute to or subtract from the transient carrier densities. In steady state, carrier generation g contributes to the current vector divergence, and recombination r reduces it. Additionally, $r_N = r_P$, since electrons and holes recombine together, and $g_N = g_P$, since they are generated in pairs.

The carriers are also subject to electric field forces (drift,) and diffusion-related movement, which are component terms of the electron and hole currents $\overrightarrow{J_N}$ and $\overrightarrow{J_P}$: [11]

$$\overrightarrow{J_N} = q\mu_N n\vec{E} + qD_N\nabla n$$
$$\overrightarrow{J_P} = q\mu_P p\vec{E} + qD_P\nabla p$$

The solver iterates, finding the carrier concentrations, current densities, and the electric fields.

These expressions are differential functions with respect to time and to space, and are solved in a sequence of steps.

The `Transient` command block is used to find the first solution of the coupled set, the Poisson and continuity equations, at the first point of the ramp. The ramp can set boundary conditions of the equations by defining a range of values of the electrostatic potential at the virtual contacts, or can define a range of incident monochromatic wavelengths, affecting g throughout the device. This solution is found by beginning from the solution to Poisson, the incident light is turned on, and the time evolution of the device is solved for, stepping through small time steps as the non-zero generation leads to non-zero time derivatives of carrier concentration. The small steps in time mean that each solution is a small deviation from the previous solution, enabling the Newton root-finding algorithm to succeed. Additionally, a gradual increase in the intensity of the incident light can also be input, to reduce sudden changes and assist in convergence. At the end of this solution, after one simulated second, the device can be assumed to be in steady state, and the time derivatives become zero.

Afterwards, the `Quasistationary` command uses the solution from the `Transient` command to inform the search of the solutions to the equations at the rest of the points in the ramp. The term

refers to the fact that the time-dependent terms are neutralized for the solution of the device, as in "stationary", except when the voltage or wavelength is changed for the next point in the ramp (quasi.) A small step in incident wavelength or voltage leads to a small difference in the generation function of space or boundary condition and a small difference in the device condition, facilitating the iterative algorithm in finding the solution, which is effective when the small difference means that the initial guess is not far from the solution of the next step.

This process is separated into two workbench modules, such that if there is an issue during the ramp, progress had been saved at the end of the initial Transient solution, and the ramp process can be easily restarted. The command files include a common header file that contains all but the `Solve` section.

4-3-3: Sentaurus Device Log File Tools

These commands have many settings to configure how to handle cases where the search for the solution diverges. The equations are rearranged such that the set of terms should equal zero, and as the process is iterated, and the terms that should equal zero are evaluated. This is called the right-hand-side (RHS) or the residual. [58] Some of the error control settings involve the RHS.

Many of the maximum error thresholds lead Sentaurus Device to abort the current step attempt and to try a smaller ramp step when exceeded. This is usually desirable, since the error threshold is configured to indicate that divergence is likely. The minimum error thresholds, like `Digits` or `RhsMin` will lead Sentaurus Device to continue iterating until they are met. If the number of iterations reaches 10, Sentaurus Device will retry with a smaller step. This helps recognize repetitions or looping situations where the numerical algorithm is not quickly finding the solution as it normally would; aborting and trying a smaller step should be beneficial. To specify these settings properly, the log information of simulation runs must be examined.

Tools have been developed to assist in observation of the Sentaurus Device log file. Different kinds of data are printed every step. Filtering the log file to sort out the data is necessary to adapt the mesh and solver preferences properly, without leading to excessive computation time. The following code is stored in `summarize_sdevice_log.sh` and is run from as a bash command with the log file name as an argument.

```
grep_counter=1
function print_and_grep {
   echo -e "\n---GREP EXPRESSION $grep_counter---\"$1\"---"
   grep_counter=`expr $grep_counter + 1`
   grep "$1" "$2"
}
```

This general-purpose function is used to apply many different filters to the log file. The `grep` standard command line program uses regular expressions to implement filters for any text. The first filter below keeps most of the information. It removes the summary of the optical transmission through every layer, which is a large table printed frequently. It also removes the `CNormPrint` data, which can be interpreted separately. It prints the row of information output on every iteration, and the reason that the algorithm terminated at each step.

```
print_and_grep "\
^Starting solve of next problem\|\
```

```
^Transient\|\
^Quasistationary \|\
^Wavelength: \|\
^   .[0-9]      [0-9]\|\
^   .[0-9]     [0-9]\|\
^   .[0-9]           inf \|\
^Error \|^\#iterations \|^|RHS|\|^Floating \| exception\|^Curve\|^Linear system
cannot " $1
```

This section more concisely prints only the information about iteration of the root finding algorithm.

```
echo "Iteration   |Rhs|      factor     |step|     error   #inner  #iterative  time"
print_and_grep "^   .[0-9]      [0-9]\|^   .[0-9]     [0-9]\|^   .[0-9]           inf " $1
```

Typically, the reason for concluding the step is clear from the last row of table. The `error` column might be below the threshold, or the `|RHS|` column might be above the threshold. The `Iteration` number between 1 and 10 implies when a new step has been taken. The time column will have extra digits if divergence was encountered. The "`inf`" text is an important part of the regular expression, such that the row is not filtered out if the number is greater than the maximum attainable by the binary format. A more concise summary of the run is printed by displaying only the last row of each step. This `-B` option of grep stands for "before" and pulls the line before the line that says "Finished".

```
echo -e "\n--- Line Before \"Finished, because\""
echo "Iteration   |Rhs|      factor     |step|     error   #inner  #iterative  time"
grep -B1 "^Finished, because" $1 | grep -v "Finished, because" | grep -v "^--$"
```

As example output, the following excerpt of 10 lines summarizes 10 steps of the algorithm by printing only the last line of the iteration table.

```
Iteration   |Rhs|      factor     |step|     error   #inner  #iterative  time
    3      9.64e+00   1.00e+00   2.04e-11   1.70e-01    0         1     33.13
    4      9.55e+00   1.00e+00   4.69e-10   1.21e-01    0         1     41.97
    3      8.37e+00   1.00e+00   1.02e-10   1.43e-01    0         1     33.10
    5      1.00e+01   1.00e+00   3.87e-01   9.91e-01    0         1     50.62
    3      1.01e+01   1.00e+00   1.76e+00   1.46e-01    0         1     33.06
    4      9.81e+00   1.00e+00   3.02e-01   1.53e-01    0         1     41.94
    7      8.09e+00   1.00e+00   2.28e-01   6.82e-01    0         1     68.56
    1           inf   1.00e+00   1.03e+25   1.36e+39    0         1   2233.62
    4      7.98e+00   1.00e+00   2.86e-11   1.33e-01    0         1     42.21
    3      9.79e+00   1.00e+00   4.87e-11   1.26e-01    0         1     32.99
```

In the original log file, the same events are covered in detail over 1171 lines. There are also other tools available for parsing the log file, making use of the available XML format.

The infinity shown is the sort of divergence that creates the time bottleneck of the device solution process. This was the only such event in this particular solution of the device.

This command prints only the reasons why a step was completed,

```
echo -e "--- Line after \"Finished, because\""
grep -A1 "^Finished, because" $1 | grep -v "Finished, because" | grep -v "^--$"
```

The summary script also separates and organizes output relating to the wavelength ramp, the time spent, and the total light absorption of the device.

```
print_and_grep "Stepsize\|^System\|^Quasistationary\|^Transient\|^Coupled\|^
Poisson\| Wavelength = \| grid points" $1
print_and_grep "^Total time:" $1
print_and_grep "^Wavelength\: " $1
print_and_grep "^A: [0-9]" $1
```

Instances where |RHS| was over maximum are found with

```
grep -B2 "^|RHS| is greater" $1 | grep -v "|RHS| is greater\|^--$\|Finished, because"
```

leading to output like this,

```
--- TWO LINES BEFORE |Rhs| is greater
    1           inf   1.00e+00   1.03e+25   1.36e+39    0         1   2233.62
    1           inf   1.00e+00   3.03e+23   2.31e+41    0         1     15.32
    2           inf   1.00e+00   1.18e+04   6.98e+20    0         1     24.24
    1           inf   1.00e+00  8.68e+101  8.61e+119    0         1     15.35
    1     3.78e+188   1.00e+00   1.94e+28   4.36e+48    0         1     19.64
    5     9.71e+210   1.00e+00   3.23e+01   1.26e+21    0         1     50.49
```

In this case, this important error condition occurred 5 times in the 76000-line log file.

This filter can be applied to find all the instances of the error for all simulation logs in the directory:

```
grep -B2 "^|RHS| is greater" *_des.out | grep -v "|RHS| is greater\|^--$\|Finished, because"
```

Finally, the following code is applicable when including the `CNormPrint` option. This concise summary includes all the interrupted steps, the preceding iteration's table entry, the corresponding wavelength point of the ramp, and the mesh vertices indicated by `CNormPrint`.

```
errors="^\#iterations \|^|RHS|\|^Floating \| exception\|^Linear system cannot "
grep -B 5 -A 21 "$errors" $filename | grep "$errors\|Wavelength\|^ poisson:\|^ electron:\|^ hole:\| ^
.[0-9]      [0-9]\|^   .[0-9]     [0-9]\|^   .[0-9]           inf " | grep -v "Finished, because" | grep -
v "^--$"
```

4-3-4: Solver Preferences

In order to balance the available convergence control options, the time bottleneck of the simulation is identified. By observation of the Sentaurus Device logs, when the initial guess has a high degree of error, it can lead Sentaurus Device to a state where the Newton algorithm diverges. In this situation, a single step can take 100 times or more the usual amount of time. If the average step time low enough, the algorithm is able to recover from this state. The `RhsMax` control helps to avoid this case. However, the RHS value is only checked against `RhsMax` after the first iteration, not before. This situation can often occur during the wavelength ramp in the two-dimensional configuration, unless the mesh and solver settings are set appropriately.

Having identified the time bottleneck, the strategy for configuring the minimum and maximum error control thresholds follows. It is a high priority to keep the time spent per iteration relatively low, and to tune tightly any settings that can reduce the likelihood of divergence of the root-finding algorithm.

The root-finding algorithm calculates an estimate of the error of the current finding at every step. The `Digits` control sets a threshold under which Sentaurus Device will accept a solution as converged based on the error estimate. It is set moderately high at 13. This control affects the number of iterations but does not affect the amount of time taken for individual iterations. Having moderately high precision will help the accuracy of the guess at the subsequent point of the ramp. Normal precision sets an absolute upper limit on `Digits` of 16, and extended precision sets that

limit to 19. [58] The level of 13 digits should be far enough removed from the absolute limit set by the precision of the floating-point binary format. The computation is unsuccessful with `Digits` as 14, but does complete with as low as 4.

`ExtendedPrecision` increases the time taken for the many necessary multiplications, and increases the step time, all to calculate unnecessary digits, so it is turned off. Computation time is better spent solving the device with finer mesh. This setting could reduce the number of floating-point exceptions, but they are rare, and are caught when they occur, leading simply to retrying the algorithm with a smaller step size.

`RhsMin` is set to 1, since the threshold of 0.1 is too low, and repeated iterations do not reach that level.

`RhsFactor` sets a limit on the amount the RHS is allowed to increase by in one step. This threshold can be set very low, since any increase in the RHS indicates a high risk of divergence, and that the algorithm is not approaching the root. Therefore, if the RHS increases significantly, the algorithm should be restarted with a smaller step size. By observation of the normal operation of the algorithm, this control is set to 10.

Sentaurus Device also offers a choice of solvers. The available solvers implement different algorithms and techniques to solve the huge linear system of equations that follow from the discretization of the device equations onto the mesh. Using the different mathematical techniques can result in slightly different convergence, accuracy, or parallelization properties, though the end result if converged can be similar. Three of the solvers are called *ParDiSo* (Parallel Direct Solver,) *Super*, (which uses a 'supernode' concept powerful on sparse matrices) and *ILS*, (Iterative Linear Solver,) which uses techniques that rearrange the sets of equations. There is also choice to use "*BE*" (Backward Euler) or *TBDRF*, (Composite Trapezoidal Rule Backward Differentiation) with the `Transient` keyword, which changes the Newton root-finding algorithm. The error control arguments exist independently of the solution method. A good approach is simply to run the simulation with each of the available solvers to determine which is the fastest for the given device geometry and mesh. It's also possible for one solver to converge where another diverges, which can produce output that would be informative in adapting the mesh.

The step size in the wavelength ramp subproject is another important setting with respect to convergence or divergence of the algorithm. With a large wavelength step, there is a chance to step over particular wavelength ranges that tend to lead to divergence. In this way, small wavelength steps can increase the probability of divergence due simply to the higher number of calculations, but, conversely, they can help resolve the detailed interference patterns and lead to better initial guesses, if sufficiently small. Ideally, the device should be solved with small wavelength steps, leading to higher resolution output.

The mesh definition over the device geometry is extremely important, and must be revisited based on the output of the simulation. Other settings, such as the precision of the binary data type, are not set high, such that more calculation resources can be allocated to solving the device with more detail by including more mesh points. Adding mesh leads to accuracy of the final calculation by reducing the error inherent to the approximation of curved trends by finding their value only on

the finite number mesh vertices. It is also an important part of managing the RHS error and avoiding divergence.

4-3-5: PLOT ERROR LOCATIONS

Sentaurus Device prints locations of the mesh vertices with the highest error to the log file when the `CNormPrint` option is specified. Since the device has many very thin layers, and there is a very large amount of `CNormPrint` data, the text output cannot be interpreted quickly by inspection. A method to plot this output has been developed, using bash and python. To interface with the bash tools, a python script is written to plot data given via `stdin`. This allows the bash pipe "`|`" tool to be used. This is implemented as `plot_coordinates_from_stdin.py`, shown in Appendix E. The low opacity specified by `alpha=0.1` creates an accent in locations that are printed repeatedly.

The regular expressions to pull the `CNormPrint` lines are given as arguments to `grep` below. The `cut` program is used to isolate the x and y coordinates, separated by a comma. The "`(`" and "`)`" brackets are used as delimiters. The pipe transmits the output of `cut` to the python script, which reads and plots the values. A "`sleep 1`" command can be added, to ensure that the plots are produced in the proper order.

```
plot_script="./bash_python_plot/plot_coordinates_from_stdin.py"
grep "^ poisson:"  $filename | cut -d "(" -f 2 | cut -d ")" -f 1 | gpythonsh $plot_script &
grep "^ electron:" $filename | cut -d "(" -f 2 | cut -d ")" -f 1 | gpythonsh $plot_script &
grep "^ hole:"     $filename | cut -d "(" -f 2 | cut -d ")" -f 1 | gpythonsh $plot_script &
```

This plots all points from the file; an example is shown in Figure 20.

However, this includes maximum measured error in instances where the solution has converged, and the information is not always important. The locations provided in instances where errors were encountered are isolated and considered in this context. The following code plots reported locations of maximum error in the mesh shortly before an `RhsMax`-exceeded event.

```
grep -B 5 "|RHS| is greater" $filename | grep "^ poisson:"  | cut -d \( -f 2 | cut -d \) -f 1 | gpythonsh $plot_script &
sleep 1
grep -B 5 "|RHS| is greater" $filename | grep "^ electron:" | cut -d \( -f 2 | cut -d \) -f 1 | gpythonsh $plot_script &
sleep 1
grep -B 5 "|RHS| is greater" $filename | grep "^ hole:"     | cut -d \( -f 2 | cut -d \) -f 1 | gpythonsh $plot_script &
```

Here, the error maximum locations of any rejected attempt are plotted:

```
cut1="cut -d ( -f 2"
cut2="cut -d ) -f 1"
grep -B 6 "^|RHS| is greater\|^|RHS| increased\|^\#iterations" $filename | \
    grep "^ poisson:\|^ electron:\|^ hole:" |$cut1|$cut2| \
    gpythonsh $plot_script "Any divergence (^|RHS| is greater\|^|RHS| increased\|^\#iterations) - Poisson,
Electron or Hole" "- $filename" &
```

This line is applicable when a simulation has been abandoned, or is lost in long-term divergence, and the latest recorded lines contain printed coordinates of maximum error.

```
tail -n 3 | grep "^ poisson:\|^ electron:\|^ hole:" | cut -d \( -f 2| cut -d \) -f 1 | gpythonsh $plot_script &
```

This line plots the point related to the latest error of a completed log file.

```
grep -B 5 "$errors" $filename | tail -n 6 | \
    grep "^ poisson:\|^ electron:\|^ hole:" | cut -d \( -f 2| cut -d \) -f 1 | gpythonsh $plot_script &
```

Note that the error of these points is quantified, and this quantity can also be considered to weigh the importance of the plotted points. The error calculation is related to the `Digits` setting. If `Digits` is set low, the error is only printed, while if `Digits` is set high, the error is scaled up and

the solution may not be accepted, and the number of iterations may reach the configured maximum. In this way, the error information can be obtained whether `Digits` is set high or low. Typically, the points specified for the Poisson equation have low error. The error quantities are printed at the end of the summarization script.

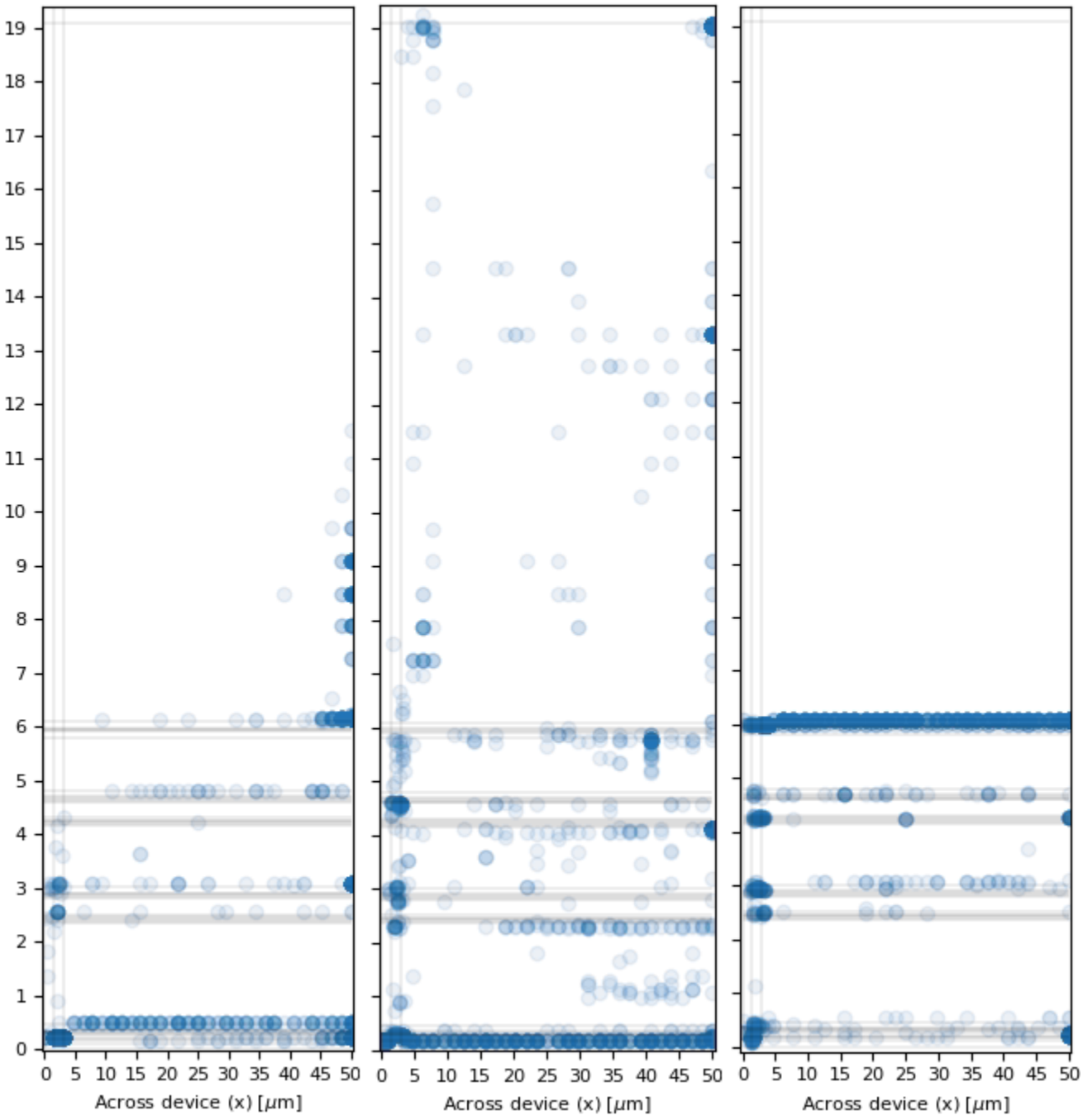

Figure 20: Mesh vertices indicated by `CNormPrint` during a full wavelength ramp to be locations of maximum estimated error found from a) the Poisson equation, b) the electron continuity equation, and c) the hole continuity equation.

4-3-6: MESH ADAPTATION

The mesh is adapted according to the following procedure. The mesh divisors are set at 4 and 6, and the simulation is run, and it can converge due to the light meshing and fast stepping. The simulation takes about 25 minutes, but is abandoned if it ever encounters divergence. The log is checked for any instances of `RhsMax` or `RhsFactor` being exceeded, or greater than 10 iterations being taken on a step. The locations of maximum error in a step indicated by `CNormPrint` are considered, in case an obvious pattern occurs, however, those points corresponding to the error cases are particularly important. For example, one case of RHS reported as infinity, (above the maximum possible value that can be represented by the binary format,) is observed, and one case of 10 iterations exceeded is observed. The location of the infinity was within the region labeled `td2_phigh`, with an x-value slightly above the contact edge. When one step met the 10-iterations threshold, it in fact appeared to be slowly converging. However, the location of maximum error from those iterations all lay within the base of the second segment, with a similar x-coordinate. Common to these two regions is that they lie within the region with x-values near to but not below the edge of the contact. Therefore, the mesh divisors in that area are multiplied by 1.5. In this way, the mesh is increased particularly in the areas where it is appropriate, and computation resources are allocated appropriately.

Divergence is less likely in locations where the accuracy of the solution is improved by additional mesh, reducing the error of the solution related to the mesh's approximation of the curves of the physical parameters. `CNormPrint` locations corresponding to convergent iterations might indicate areas where there are simply more mesh vertices, because these prints will have more randomness, independently increasing the likelihood that these areas will be indicated. In this plot of all points, there are some irrelevant error location prints from the electron equation in p-type areas, and from the hole equation in n-type areas.

The "`multiplier`" argument to the `mesh_refinement_rectangle` is used for these purposes. The `mesh_divisors` are used for global control, and set the amount of mesh throughout the entire device. This approach takes much of the guesswork out of improving the mesh, and allows for improvement of the accuracy of the solution, by reducing the likelihood that divergence is encountered. Following the above procedure, the multipliers are increased especially on the window layer and front emitter layer, and also on every FSF and emitter layer.

4-4: WAVELENGTH RAMP CIRCUIT – CALCULATION OF J_{sc}

Two of the three subprojects make use of the circuit that is implied by the creation of virtual contacts. In the wavelength ramp subproject, all the contacts are held at $0\,V$. Therefore, all the currents that follow are short-circuit currents. These currents are caused by the input monochromatic illumination. Every segment of the device has contacts on both sides. Sentaurus Device calculates electron and hole current density vector fields throughout the device, and this leads to scalar electron and hole current output data at each contact.

The simulation is done with a specific width, but it is valid to generalize the result with respect to device area, and to assume that the device will produce current proportional to its area. This is done by dividing the output current by the simulation width, and by the `AreaFactor` input

argument appropriately, to achieve a unit of [A/cm^2], or a current density J for the device, with the cross-section defined as the device area, including the width of the contact. That is, Sentaurus Device outputs a current in [A], but supplies an optional argument `AreaFactor` that can be set to

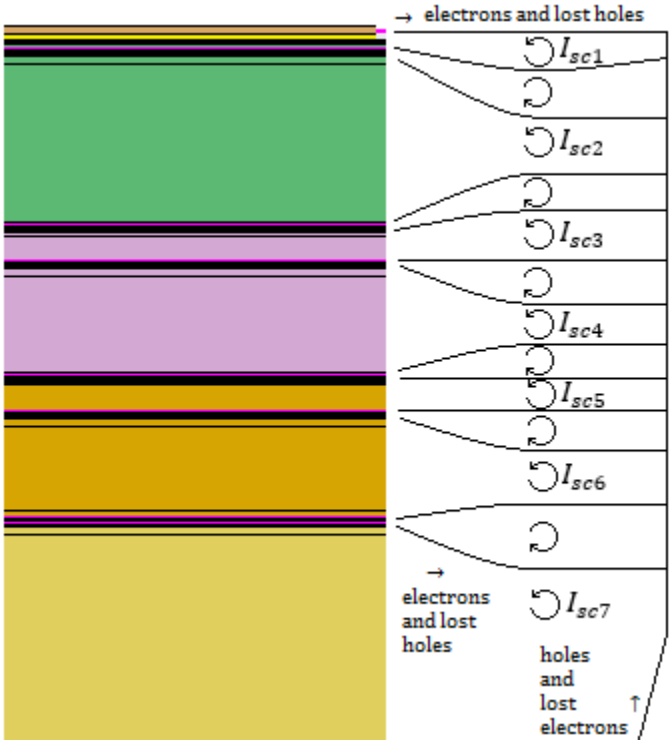

Figure 21: Schematic of external circuit showing current loops for Kirchhoff's Current Law.

convert the output to current density. The necessary conversion factor will depend on the simulated width. Since the approach affects the calculations in Sentaurus Visual and python, that processing is made more robust by creating a more global space for `AreaFactor` as a workbench variable, and the current I is converted to current density J simply by dividing by the simulated area, `AreaFactor*@w@/2`. At that point, J can be multiplied by any particular device area to determine the expected current. This allows the simulated width to be changed without also having to change `AreaFactor` in the Sentaurus Device command file, as well as the calculations in the Sentaurus Visual and python modules.

The scalar current output by Sentaurus Device is called `TotalCurrent`, and electron and hole currents `eCurrent` and `hCurrent` are also available. In the python post-processing, these outputs become arrays indexed by the terminal number. When Sentaurus Visual is used for post-processing, the dataset is loaded in and referred to with the same format as provided by Sentaurus Device. For the seven-segment device, there are 14 terminals, so the terminal indices lie from `0` to `13` inclusive. This allows the total current produced by each individual subcell to be calculated by applying Kirchhoff's Current Law. This leads to an array `J_sc` of length `7`. Each entry of the array contains a dataset parallel with the wavelength ramp. This data structure allows a loop to be implemented that is generalized with respect to the number of segments or contacts. $J_{sc1}(\lambda)$ is therefore `J_sc[0]` in the array.

The sign convention of `eCurrent` and `hCurrent` follows the generalized device convention where usually a device consumes current, so current going into the device is defined as positive. Additionally, `TotalCurrent[i] = eCurrent[i]+hCurrent[i]`, which implies the sign of `eCurrent`. It is the contribution to the total current from electrons. This ignores the electron particle movement, which is opposite, due to the negative electron charge. For power production, electrons come out the top of the device, since it is n-type there, and holes come out of the p-side at the bottom. Minority carriers can also touch the contact, which leads to recombination, and this is reflected in the `eCurrent` or `hCurrent` data at the terminals. Electron particles depart the cathode, and holes departing the device via the cathode represent energy lost to recombination. The short circuit current of each subcell is found from the terminal currents by calculating the loop current of a portion of the circuit. The arrows in Figure 21 indicate actual particle and current directions.

The particle directions are not relevant to the calculation of short circuit current, with given contact currents. According to the definition, current is positive when it flows into the device, and this true of both contributions of `eCurrent` and `hCurrent`. This direction is in line with the loop direction of I_{sc1} defined in the figure, so it follows that `I_sc[0] = eCurrent[0] + hCurrent[0]`. In this way, the loop direction is chosen such that I_{sc} is positive. In the result, at the first terminal, `eCurrent` is dominant and positive, and `hCurrent` is small and negative, in wavelengths where the first GaInP subcell produces current. Kirchoff's Current Law does not hold for each of `eCurrent` and `hCurrent`, however, it is true that `sum(eCurrent)` equals `sum(hCurrent)`, such that the law holds for `TotalCurrent`.

The Epi module code maintains a variable `num_segments` such that loops can be written to process each subcell generally. This supports the exploration of other designs, and also makes the code manageable within the scope of one project, given the high number of layers. The structure editor module creates list of contact names, `contact_list`, based on `num_segments`, such that it need not be assumed at this point in the project. These two variables are naturally coupled. They are referenced from the TCL space by printing them within quotes and they become a python string after preprocessing. The split function creates a list from the string. A library function is used to create a dictionary containing the data from Sentaurus Device. The contact names become parts of the keys for this data structure. There are two possible methods to generalize the calculation for all subcells: one by starting at the first terminal, and one by starting at the last terminal. The following list comprehension takes steps of 2, skipping the loops adjacent to the tunnels. This is done with the expression `2*i+1`, defining the end of the range of terminals to sum as the set of odd numbers.

```
eCurrent = [sim.data[s + ' eCurrent'] for s in "!(puts -nonewline
$contact_list)!".split()]
hCurrent = [sim.data[s + ' hCurrent'] for s in "!(puts -nonewline
$contact_list)!".split()]
J_sc = [sum(eCurrent[0:2*i+1]+hCurrent[0:2*i+1]) for i in range(0,num_segments)]
```

Additional loops are present in the external circuit. These are named `tunnel_Isc`, as in the short circuit current being produced by the tunnel junction layers. Tunnel junctions are engineered to be transparent, because absorption in the tunnel layers is undesirable. Similarly, generation too close

to these virtual contacts in the BSF or FSF layers is also undesirable. These layers are necessary to manage the electrical processes in the device, and absorption within them is a side-effect to be minimized. This kind of absorption is called parasitic. These losses are reflected in the output currents in the circuit. The `tunnel_Isc` quantity is primarily a consequence of this sort of absorption. Similarly, absorption in the FSF or BSF layers can promote minority carrier loss. The tunnel component used here between subcells 1 and 2 is not transparent to wavelengths to be collected by the second GaInP subcell. Engineering this tunnel to be performant in a segmented solar cell design remains as work yet to be done. In this design, the layer thicknesses of the first tunnel junction were set low at 22 nm to reduce the parasitic absorption. The parasitic currents produced by the tunnel junctions are much less than the desired currents produced by the subcells.

The currents in the loops adjacent to the tunnels are solved in a similar manner. The negative sign results in positive quantities that match the loop directions in the figure.

```
tunnel_Jscs = [-sum(eCurrent[0:2*i]+hCurrent[0:2*i]) for i in range(1,num_segments)]
```

4-4-1: Minority Carrier Loss at Contacts

There is energy lost when minority carriers have traveled the undesirable direction, opposite that which could create power, and recombine at the contact. The surface field layers are designed to prevent minority carriers from diffusing into this direction from inside the active layers of the subcell, however, when electron-hole pairs are created immediately near to contact, within the surface field layer, on the wrong side of the barrier, they can easily and very significantly arrive at the wrong contact and create current in the wrong direction, dissipating energy. The first case of this loss quantity is available as the hole current out of the cathode at the top of the device. This is significant for short, ultraviolet wavelengths, which are absorbed immediately within the first FSF (AlInP window) layer, on the wrong side of the hole barrier. This kind of loss is not significant for most of the spectral range, which will be transmitted through the window.

In Figure 22, holes near the top surface of the window layer and cap travel the wrong direction to recombine at the top contact. The surface SRH process is losing less carriers than the effect at the contact on the cap in this data, with surface recombination velocity of 10^3 cm/s.

There is a similar loss mechanism within the device, at every subcell. To plot this loss, virtual contacts on both sides of every tunnel component are required. With only one contact beside each tunnel, majority holes, producing desired current, coming from above, are indiscernible from minority holes, coming from below, the incorrect direction, and the loss cannot be quantified, even by considering summations or continuity of electrons and hole currents from elsewhere in the device. In this case, there is slightly significant recombination in the tunnel layers, compared to when both contacts are present and there is less current traveling through the tunnel layers. With the additional contacts, minority electrons traveling down, undesirably, are detected and quantified at the first internal virtual contact, hole loss from the top of the second segment are quantified and plotted, and so on, until the electron current at the bottom of the last subcell is detected at the anode. These energy losses corresponding to minority carriers that have circumvented barriers and diffused the wrong direction are shown in shades of cyan for electrons and yellow for holes in the fill plots in the results section.

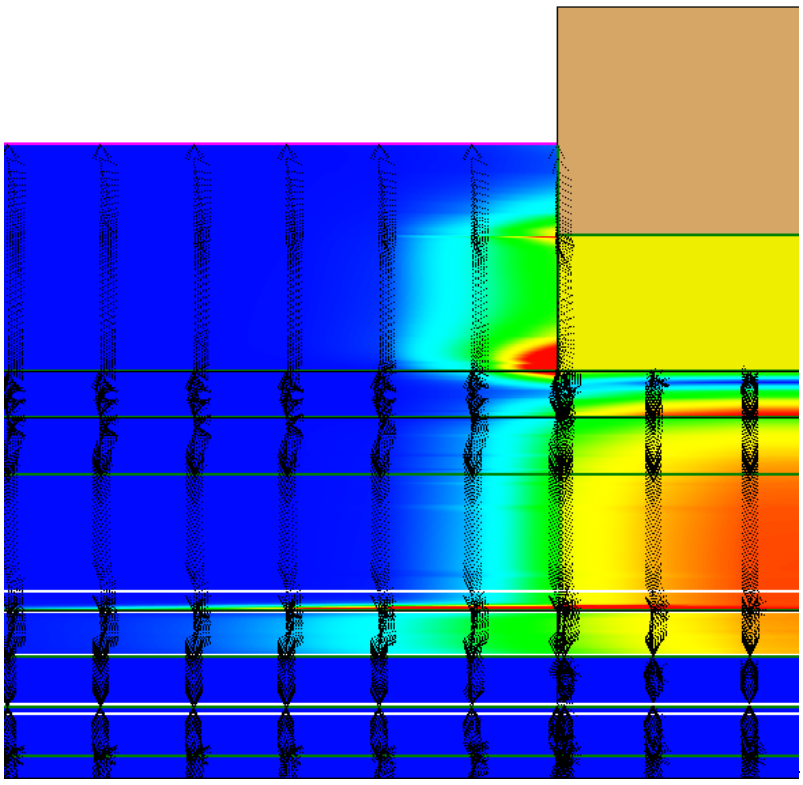

Figure 22: Vector and colour contour plot of hole current density near the top contact with 300 nm incident light.

The presence of two virtual contacts at every tunnel, maintained at 0 V, does have more important effects on the outcome of the experiment, and the relevance of the result to the design process. For lower-order multi-junction solar cells, the generation within the tunnel layers is very little. In this design, for each segment to produce similar current, as is required, the first segment is extremely thin. Therefore, very short wavelengths with very high absorption coefficients are now incident on the first tunnel, and generation there becomes significant. The effect will be different near the maximum power operating point of the device. In this experiment, when there are two 0 V contacts adjacent to the tunnel, that p-n junction operates in the regime of a shorted solar cell, an unusual regime for the component, as there is no voltage present that would correspond to the tunneling regime for the layers. A short-circuit current opposite to the desired direction is observable at the contacts, which corresponds to an important energy loss in this experiment. The current corresponds to less than the total energy absorbed in these layers, since not all carriers arrive at the virtual contacts. The loss is similar, but cannot be quantified, when only one contact is present adjacent to the tunnel junction layers.

The minority electrons from the *p* side of subcell 1 are detected at the second terminal, and the majority electrons from the *n* side of subcell 2 are detected at the third terminal. If only one terminal is placed adjacent to each tunnel, another method would be required to detect the minority carrier losses, since the `eCurrent` of that terminal would be the sum of both sources, and some information would be lost. Therefore, with one contact beside each tunnel, only the minority carrier

loss at the cathode and anode can be inferred. With all the necessary contacts present, the minority carrier loss expressions are below.

```
hole_losses     = [-hCurrent[2*i]  for i in range(0,num_segments)]
electron_losses = [eCurrent[2*i+1] for i in range(0,num_segments)]
```

4-4-2: External Quantum Efficiency

It can be informative to interpret the output in units of quantum efficiency instead of typical current units. This is a calculation of the probability that an incident photon of given wavelength is converted into current. It is found by calculating the output number of electrons and dividing by the input number of photons. Dividing a typical current by the charge of an individual electron produces a rate of the number of electrons. As an input to Sentaurus Device, the power of the monochromatic light was specified in [W/m^2]. This is divided by the energy of individual photons, which is a function of wavelength, $E = hc/\lambda$, where h is Planck's constant, and c is the speed of light. Since the incident power is constant, the number of photons is less in the short wavelengths and more in the long wavelengths. Normalizing the output according to this trend produces the external quantum efficiency (EQE).

```
E             = hc/(sim.wavelengths/1e6) # [J] array
w_cm          = @w@/1e4
areaFactor_cm = @areaFactor@/1e4
I_to_EQE      = E / @intensity@ / (w_cm/2) / q / areaFactor_cm
e = [I_to_EQE*e_i for e_i in eCurrent]
h = [I_to_EQE*h_i for h_i in hCurrent]
```

4-4-3: Internal Quantum Efficiency

Internal quantum efficiency is another useful quantity to help the designer interpret the performance of the device. The EQE result has strong dependence on reflectivity, in addition to the many electrical processes within the device, such as the carrier dynamics and recombination. Ruling out the possibility of reflection, one might consider only what may happen to photons that are not reflected. This evens out the reflectivity and interference patterns that affect the EQE plot and distort other processes. [59] Additionally, the contact shading is not considered to affect IQE. Transmission is not generally ruled out in this manner. The IQE can be calculated from the EQE by dividing by a factor of $1 - R$, where R is the probability of reflection, to remove the effect of reflection, and also by dividing by the relative coverage of the contact.

```
eqe_to_iqe = @w@/(@w@-@w_contact@)/(1-sim.data["LayerStack(Window1) R_Total"])
```

All of the above currents can be converted to EQE and IQE with these prefactors.

4-4-4: Generation and Recombination Rates

Generation and recombination rates can be integrated throughout layers of the device to produce a similar unit. The typical unit for these processes is number of particles per second per volume, and when integrated, becomes number of particles per second, which can then be divided by the incident number of photons to produce a dimensionless unit akin to probability and EQE. Surface recombination can be integrated similarly, though the unit before integration has one less dimensional factor. Sentaurus Device's "`IntegrationUnit`" argument in the Math section must be set to "`cm`" to produce a unit of number of photons/s when integrating over a surface. Sentaurus

Device only considers AreaFactor when supplying output terminal currents; generation and recombination rates are only integrated over the 2D rectangle in the simulation space and must be multiplied by AreaFactor to be comparable with output currents.

```
rate_to_EQE = E / @intensity@ / (w_cm/2) * device.areaFactor_cm
```

To reduce the number of arguments required by functions that create plots, it is useful to bundle these device variables and conversion factors into one object. This is shown in Appendix F.

4-5: VOLTAGE RAMP FUNCTIONS

In the subprojects that calculate device *J-V* characteristics, the AM1.5D spectral data is applied as illumination, and voltages of the contacts are ramped with the `Quasistationary` command of Sentaurus Device. Each segment is ramped individually by keeping the terminals on the cathode side at $0\ V$ and ramping the terminals below it. This directly biases each segment, one at a time, keeping the others shorted. The shorted segments produce their respective short circuit currents while the portion of the device under test is being measured. The expressions following from Kirchhoff's Current Law are identical in the voltage ramp and in the wavelength ramp, isolating the current produced from any segment.

Two `Quasistationary` code blocks are used, such that the ramp step size can be decreased near the maximum power point. The voltages define boundary conditions for the device equations. This sets the electrostatic potential at the contacts, and the Poisson equation is solved for the electrostatic potential throughout the device, coupled with the carrier concentrations that follow from the continuity equations.

Given that there are 7 segments in the device explored currently, and potentially many more in other designs to be explored, the `Quasistationary` code blocks must be produced systematically. A TCL function is created that can print the two ramp code blocks. It is defined in a common directory and included, such that it can be called by both subprojects that can use it. The first subproject produces a *J-V* curve for each segment and is useful for current matching. The second subproject simulates the entire device without virtual contacts within it. The TCL function that prints the ramp code accepts intuitive arguments to specify the parameters of the ramp.

First, the simple voltage ramp function that requires the goal voltage to be specified and the terminals to ramp. Unspecified terminals remain at $0\ V$. The terminal set can be specified with a regular expression supported within Sentaurus Device, including string fragments like "`[1-3]`", that will match any integer in the range.

```
#sdevice_quasistationary_ramp_functions.tcl
proc voltage_ramp {goal num_steps terminals {DoZero 0}} {
    # These steps are in a parametric variable t that ramps from 0 up to Goal
    puts "  Quasistationary("
    puts "      InitialStep   = [expr 1.0 / $num_steps]"
    puts "      MinStep       = [expr 1.0 / $num_steps / 1e9]"
    puts "      MaxStep       = [expr 1.0 / $num_steps]"
    puts "      Increment     = 2"
    puts "      Decrement     = 2"
    if {"$DoZero" == "DoZero" || $DoZero} {
        puts "      DoZero"
    }
    foreach terminal $terminals {
        puts "      Goal{ Voltage = $goal Name = Regexp(\"$terminal\") }"
```

```
    }
    puts "  ) {"
    puts "      Coupled("
    puts "          Iterations = 40"
    puts "      ) {"
    puts "          Poisson Electron Hole"
    puts "      }"
    puts "  }"
}
```

This function does not define the beginning of the ramp. The `Solve` section of Sentaurus Device is sequential. First, the `Coupled` command invokes the solution of Poisson. Second, the `Transient` command couples in the two continuity equations. These steps are so far identical to the wavelength ramp subproject. However, in the *J-V* ramp subprojects, the AM1.5D illumination is programmed in the `Optics` section to turn on during the `Transient` step. As such, the `Transient` command first produces a solution in the dark, with the initial solution of Poisson providing a starting point, then the light is turned on, and the corresponding steady-state solution is found by the conclusion of the `Transient` block. The project could also be configured to find the *J-V* characteristics in the dark. The state of the device is saved at this point - solved with all contacts shorted and the illumination on. This device configuration must be loaded before every ramp. The `Quasistationary` code block simply causes Sentaurus Device to walk the solution toward the `Goal` from the current configuration.

```
proc reset_ramp {reset_plot_file_name next_plot_file_name} {
    puts "  Load(FilePrefix  = \"$reset_plot_file_name\")"
    puts "  NewCurrentPrefix = \"$next_plot_file_name\""
}
```

The `IV_ramp` function executes two voltage ramps, such that a smaller step size is used to more accurately resolve the maximum power point. It requires the name of plot file saved at the end of the previous `Transient` step, so that it can reload the starting device condition prior to every ramp.

```
proc IV_ramp {reset_plot_file_name next_plot_file_name V_turn_on V_stop num_points_1 num_points_2
terminals {DoZero 0}} {
    reset_ramp $reset_plot_file_name $next_plot_file_name
    voltage_ramp $V_turn_on [expr 1.0*@points_multiplier@*$num_points_1] $terminals $DoZero
    voltage_ramp $V_stop    [expr 1.0*@points_multiplier@*$num_points_2] $terminals
}
```

The following code block defines the number of points and range of the voltage ramps for the subcell *J-V* curves as the `IV_ramp` function is called. A rough expression increases the range of the ramp when the light concentration is increased and the subcells are expected to have higher open-circuit voltage V_{oc}, such that the code need not be edited when the concentration changes. The base number of data points in each section of the curve is set as function arguments. This number is multiplied in an intuitive manner by a workbench variable, then converted to step constraints for the ramp block in the format required by Sentaurus Device. This arrangement allows the choice between speed or precision from the workbench, making a coarse setting available for testing purposes. Not shown here is another control that ramps only one subcell, to produce a test result more quickly. This block of code does have to be maintained to simulate different devices or numbers of segments, as it is not written as a loop, but this arrangement makes that possible. Without this TCL function, these 14 ramps would comprise 225 lines of code. It may be possible to improve this implementation with the Sentaurus Device "`Continuation`"

command, which is designed to ramp I-V curves and resolve the maximum power point. However, that automatic process has no expectation of the result and could take additional time to execute.

```
!(
    #include "../common/sdevice_quasistationary_ramp_functions.tcl"
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc1" \
        @<1.2+0.0003*concentration>@   @<1.5+0.0003*concentration>@  2 7 {"td[1-6]_VContact[1-2]" "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc2" \
        @<1.2+0.00025*concentration>@  @<1.5+0.00025*concentration>@ 2 7 {"td[2-6]_VContact[1-2]" "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc3" \
        @<0.7+0.00025*concentration>@  @<1.0+0.00025*concentration>@ 2 4 {"td[3-6]_VContact[1-2]" "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc4" \
        @<0.7+0.00025*concentration>@  @<1.0+0.00025*concentration>@ 2 7 {"td[4-6]_VContact[1-2]" "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc5" \
        @<0.15+0.00025*concentration>@ @<0.4+0.00025*concentration>@ 2 7 {"td[5-6]_VContact[1-2]" "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc6" \
        @<0.15+0.00025*concentration>@ @<0.4+0.00025*concentration>@ 2 7 {"td6_VContact[1-2]"     "anode"}
    IV_ramp "n@node@_BanddgmJsc" "n@node@_sc7" 0.1 @<0.25+0.00025*concentration>@ 2 7 {"anode"}
)!
```

The full device voltage ramp is defined in the other subproject as follows:

```
#include "../common/sdevice_quasistationary_ramp_functions.tcl"
voltage_ramp @<5.0+0.015*concentration>@ @<5*points_multiplier>@ {"anode"} "DoZero"
voltage_ramp @<6.2+0.013*concentration>@ @<5*points_multiplier>@ {"anode"}
```

The endpoint of the voltage ramps need not be set accurately, since a break condition for the ramp is set in the `Math` section relating to the power of the device. Since all but one of the segments are shorted, the device power is the power of the segment experiencing the non-zero Volt bias. Power, following $P = IV$, is zero for the segments biased at $0\,V$. The power follows the normal device convention, where generated power is negative, and consumed power is positive. Therefore, for the entire ramp, until the photocurrent is overcome by the diode current, the device power is negative. For the bias to surpass V_{oc} of the segment, the net power consumption of the device becomes positive. This occurs when the J-V curve crosses the horizontal axis, and ends the ramp at the appropriate time. This ramp break condition is configured in the `Math` section of the Sentaurus Device command file with the following code. This is included in the subcell and device J-V projects.

```
#if @concentration@ > 0
   BreakCriteria{DevicePower(MaxVal=0)}
#endif
```

4-6: Sentaurus Visual Plot Development

The post-processing of this data is done in Sentaurus Visual. This is a plotting tool that interacts with the data structures used by the other modules. When a data file is loaded, its details can be navigated with the sidebar of the GUI. The file can contain multiple datasets to explore, and each dataset can contain multiple variables. Variables can be named in two levels. The first level can be the name of the contact, and the second level is any physical quantity associated with the contact, such as Voltage, electron current, hole current, or charge. In the code, this is as if there can be a space in the variable name, and in the GUI, the dataset variables can be displayed in a more organized manner.

Each pair of voltage ramps from Sentaurus Device creates a different dataset, and the variables of the dataset contain data sufficient to plot the J-V curve. It is a pair of ramps that control the step

size for two portions of the curve, but the resultant data is not separated in that manner. As part of resetting the state of the device, prior to beginning the *J-V* ramp for the next segment, a new dataset was initiated. The dataset contains the voltages of all contacts, but there are only two voltages involved, and many of the contacts have the same voltage. The Sentaurus Visual code refers to the anode for the ramp voltage of any segment, since all terminals below the segment under test are ramped with the same voltage.

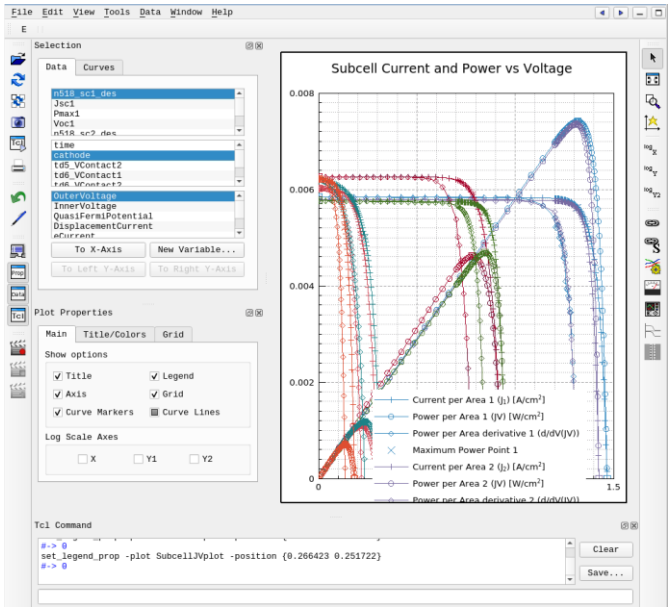

Figure 23: Sentaurus Visual GUI.

This structure is reflected in the TCL commands available in Sentaurus Visual. The `create_curve` function expects a dataset name and a dataset variable as input. To apply general arithmetic operations to the data, the `create_variable` function is also necessary, since the `create_curve` function has limited arithmetic function support. Curves can be created that are sums or functions of other curves, but the use of dataset variables is required for more complex functions. The `create_variable` function can add data to the dataset, and has better functional support. This TCL function pairs the two operations, and can be used to create a curve based on more complex functions. Creating variables in the dataset to enable them to be plotted is a generally effective pattern in this context.

```
proc create_variable_function_and_curve_vs_V {dataset name function} {
    create_variable -dataset $dataset -name $name -function $function
    create_curve    -dataset $dataset -name $name -axisX "anode OuterVoltage" -axisY $name
}
```

To structure the code, a function named `plot_JV_data` is created, according to the top-down programming style. A function such as this can be called for every *J-V* curve as part of a loop. This function is defined in the common area, included, and also called by the full device subproject. Sentaurus Device created a different data file for each voltage ramp. The file is loaded and an identifier is passed into the plot function. The full device subproject calls the same function, and specifies a letter “T” in the place of the segment number, which is used as a subscript. Whenever possible, functions such as those that process a *J-V* curve, are placed in a common directory, and included with an `#include` directive, such that they need only be updated in one place.

```
for {set i 1} {$i <= $loopmax} {incr i} {
    set curve_colour [getColor]
    plot_JV_data [load_file n@node|sdevice@_sc${i}_des.plt] $i $num_segments $curve_colour
    do_metrics $i $curve_colour
}
```

The following function call calculates the current density and plots it. The current density is converted from the current in the data by dividing by the area, interpreting the `AreaFactor`

variable as a horizontal dimension perpendicular to the simulated width. The simulated width was specified by Epi as `@w@/2`, to be clear that half of a device is simulated, exploiting symmetry, and making use of a globally available workbench variable.

```
create_variable_function_and_curve_vs_V $dataset J$i "([getKCLterms $i $ds])/(@areaFactor@*1e-4)/(@w@*1e-4/2.0)"
```

The current of a segment is obtained from the currents of the terminals by applying Kirchhoff's Current Law. The expression must be written as a string in the format expected by the `create_variable` function. This approach allows the current density to be calculated without maintenance if the simulated width or the value of `AreaFactor` is changed. The syntax of the `create_variable -function` argument requires the dataset variable name, specifying the contact and the data, and the dataset name, separated by a colon. The `-function` option of the `create_curve` command does not have this level of capability, instead, curves can only be created by processing other curves. This is sometimes possible, by plotting and deleting curves of intermediate terms. The loop implements a generalized application of Kirchhoff's Current Law, where the current generated by any segment is solvable from the terminals above it.

```
proc getKCLterms {subcell_number ds} {
    set terms "<cathode TotalCurrent:$ds>"
    for {set j 1} {$j < $subcell_number} {incr j} {
        set terms "$terms+<td${j}_VContact1 TotalCurrent:$ds>+<td${j}_VContact2 TotalCurrent:$ds>"
    }
    return $terms
}
```

The power density is calculated and plotted with similar methodology, and applying $P = JV$. Similar to current density, this is not a localized power density. Higher or lower localized power densities exist within different areas of the device. This power density is an average with respect to the device area, that, when multiplied by the device area, produces the device's output power.

```
create_variable_function_and_curve_vs_V $ds P$i "<J$i:$ds>*<anode OuterVoltage:$ds>"
```

The derivative of the power curve will be used to find the maximum power point. The `create_curve -function` option is sufficient, providing a derivative function that can be applied to another curve.

```
create_curve -name dPdV$i -function "diff(<P$i>)"
```

The `plot_JV_data` function returns, and the `do_metrics` function is called.

There are a few interpolation functions available within Sentaurus Visual. When applied to the shape of the *J-V* curve, the cubic splines interpolation can create some error. The arrangement of functions below implements the structure required by the Sentaurus Visual functions, compartmentalizing the algorithms, and allowing them to be interchanged for testing. The output of this data processing can be subject to noise error, if the step size is very small and the output is not smooth.

```
package require math::interpolate
proc interpolate {curve_name x} {
    return [cubic_interpolation $curve_name [expr 1.0*$x]]
}
proc cubic_interpolation {curve_name x} {
    set spline_coeffs [::math::interpolate::prepare-cubic-splines [get_curve_data $curve_name -axisX] [get_curve_data $curve_name -axisY]]
    return            [::math::interpolate::interp-cubic-splines $spline_coeffs $x]
```

```
}
proc linear_interpolation {curve_name x} {
    return [::math::interpolate::interp-linear [get_list_of_pairs $curve_name] $x]
}
proc lagrange_interpolation {curve_name x} {
    return [::math::interpolate::interp-lagrange [get_list_of_pairs $curve_name] $x]
}
proc get_list_of_pairs {curve_name} {
    set list_of_pairs [list]
    foreach x [get_curve_data $curve_name -axisX] y [get_curve_data $curve_name -axisY] {
        lappend list_of_pairs $x $y
    }
    return $list_of_pairs
}
```

The maximum power voltage V_{mpp} is found by interpolating where $dP/dV = 0$ with the `::ext::ExtractValue` interpolation function, then the maximum power is found by interpolating the power curve. V_{oc} is found similarly from the J-V curve. Subsequently, the current at the maximum power point J_{mpp} is found with $J_{mpp} = P_{max}/V_{mpp}$. This avoids a significant source of error related to interpolating the J-V curve.

The characteristic points are highlighted on the plot and printed as results. The points are placed in datasets such that they can be referred to by the `create_curve` function, which reads strictly from datasets, and also, they remain available outside the main loop such that the output can be sorted before printing to the workbench.

```
create_variable -dataset Jsc$i  -name V -values 0.0
create_variable -dataset Jsc$i  -name J -values [interpolate J$i 0]
create_variable -dataset Voc$i  -name V -values $Voc
create_variable -dataset Voc$i  -name J -values 0.0
create_variable -dataset Pmax$i -name V -values $Vmpp
create_variable -dataset Pmax$i -name P -values [interpolate P$i $Vmpp]
create_variable -dataset Pmax$i -name J -values [expr [get_Pmax $i] / $Vmpp]
create_curve    -dataset Jsc$i  -name Jsc$i  -axisX V -axisY J
create_curve    -dataset Voc$i  -name Voc$i  -axisX V -axisY J
create_curve    -dataset Pmax$i -name Jmpp$i -axisX V -axisY J
create_curve    -dataset Pmax$i -name Pmax$i -axisX V -axisY P

proc get_Jsc {i} {
    return [get_variable_data -dataset Jsc$i J]
}
proc get_Voc {i} {
    return [get_variable_data -dataset Voc$i V]
}
proc get_Pmax {i} {
    return [get_variable_data -dataset Pmax$i P]
}
proc get_Jmpp {i} {
    return [get_variable_data -dataset Pmax$i J]
}
proc get_Vmpp {i} {
    return [get_variable_data -dataset Pmax$i V]
}
proc get_FF {i} {
    return [expr 100.0 * [get_Pmax $i] / [get_Jsc $i] / [get_Voc $i]]
}
proc get_efficiency {i incident_power_W_cm2} {
    return [expr 100.0 * [get_Pmax $i] / $incident_power_W_cm2]
}
```

The argument `incident_power_W_cm2` is the full integration of the AM1.5D spectrum, 0.09 $\mathrm{W/cm^2}$, multiplied by the concentration at the input.

4-7: PLOT DEVELOPMENT WITH PYTHON MATPLOTLIB

While many `matplotlib` and `pyplot` tutorials are available [41], a concise revisitation of the subset of functions and approaches available that have been employed in this particular project is appropriate here.

It is common to import the library with the style import `matplotlib.pyplot as plt`. However, to adopt a style that limits acronyms and shortened variable names, it is imported with `from matplotlib import pyplot`. Then, all interactions with the module are done with `pyplot`, and are very clearly, easily, and quickly read and understood, without the potential to confuse `plt` even briefly with some individual plot.

When multiple plots are worked with simultaneously, in a very long python code file, and the number of simultaneously available axes and figure variables compound as the code is developed, the code can become complicated and disorganized. Here, every plot has its own function or code block and is started and ended before the next. There is some code common to different plots that can be factored out, that is in fact, primarily, the physics calculations. Therefore, processing the data produced by Sentaurus, including calculating the short circuit EQE from the current outputs, is done at the top level. Then, completed or nearly completed datasets and factors are passed as arguments to the plot functions, which can be universally tasked with the intuitive plotting job only, such that the developer need only read the details within the plot functions when necessary.

One process that may be common to multiple plots, such as to set the *x*-axis limits, could all be done at once in a loop. However, it is just as effective to finish each plot before starting the next one. The function, `configure_axes`, can be called by any plot with common axis limits. It is just as acceptable and concise to use `pyplot.xlim(wl[0],wl[-1])` where `wl` is the wavelength array variable, within every individual plot function. In python, the array index `-1` is used here to refer to the last element of the list, the last wavelength of the ramp. The *y*-axis has less commonality, which bolsters the stylistic preference to configure every plot individually and separately.

The Matplotlib `pyplot` module returns axes and figure objects that can be stored as variables, however, the module itself does supply references and functions that render this practice optional if plots are done one at a time. Therefore, unnecessary variables aren't created for these objects. Then, the code is simpler to read and more concise. The `pyplot.gcf()` function returns the current figure, and a figure can be focused on by name with `pyplot.figure("Figure name")`, which is not necessary when the python code is organized such that plots are completed one at a time.

Commonly employed functions include `pyplot.figure` to create a figure, `pyplot.plot(wavelengths, data)` or `pyplot.plot(wavelengths, factor * data)` to create a line, where `wavelengths` and `data` are `numpy` arrays of the same length. `factor` is also a `numpy` array of that length. `Numpy` is a commonly invoked python library that supplies arrays that can be manipulated similarly to Matlab arrays. Often, it is abbreviated to `np`, though that convention is not followed here. The multiplication operator `*` used on `numpy` arrays will multiply the data point by point, which is not the behaviour of regular python lists. Alternatively to `numpy`, point-by-point multiplication can be accomplished in python with the list comprehension syntax. `pyplot.ion()` turns on behavior for interactive sessions, updating figures without the need for

`pyplot.show(block=False)`. The `block` argument sets whether the function returns while the figures remain open, and the interactive mode changes the default to `False`, such that the command prompt remains available. The `block=True` option is appropriate if the python interpreter would otherwise exit before the could be viewed. The `pyplot.autoscale(tight=True)` scales the axes while showing all data, where manual scaling with `ylim` could hide data at an unexpected *y* level.

Plot code files or functions shouldn't become too long, so they remain readable and workable. If plots are described one-by-one, and calculations are also contained in their own area, with results passed as arguments to the plot functions ,whose responsibility is solely plotting, the workflow is manageable.

There are over 40 plots produced by this code. It has been helpful to run the post-processing outside of the workbench, though it is also useful that it can be run from the workbench as a Sentaurus module occasionally. One workflow has been to skip showing the plots in open windows, and to automatically save them as files, and view them with Microsoft Windows Explorer's "Very Large Icons" setting. This produces a convenient overview of one simulation's entire outcome. Otherwise, in Microsoft Windows, after undesired windows are minimized, the "Show Windows Side-by-side" option in the taskbar context menu is useful to organize many plots, potentially saving nearly hundreds of mouse-clicks.

Perhaps the best strategy has been to write the python code such that it can be copy and pasted into an interactive Python session. Rather than typing commands into the prompt, commands are entered into the text editor, and saved, and run in the prompt. The session started after using the workbench to preprocess the module with `gpythonsh -i ppNNN_pyt.cmd`, where `NNN` is the node number corresponding to the job in the workbench table. In this manner, plot functions or small code blocks can be developed more quickly. Having many plot windows open in Microsoft Windows Remote Desktop can put many of them out of reach, so finding and running the code block for a desired plot can be more convenient than having all plots simultaneously available in open windows. This is only possible when each figure's code is begun and ended in a concise manner. Writing code blocks in this style can invoke the `pyplot.cla()` (clear axes) function, such that a repeated execution of the same code block reproduces the same plot. Initializing the session in this manner, and organizing the programming object roles consistently, ensures that the top-level file is concise, readable, and maintainable, and that all the plots are quickly available by running the appropriate function.

To save viewing space, `pyplot.title()` can be neglected in favour of using a string argument to `pyplot.figure`. Then, plots are later identified by name. This has been more manageable than working with the sequentially assigned integer figure numbers, which are subject to change as the code is developed. The figure name appears in the Microsoft Windows title bar and task bar.

At the end of the top-level file, there is a loop through all figures that uses `pyplot.get_fignums()` and saves a copy using `pyplot.gcf().savefig()`, since this processing is indeed common to all figures. Simulation experiment parameters, such as 1d or 2d mode, the concentration level, and surface recombination coefficient, are shown in the title.

Functions that automate grid and tick mark spacing are available in Appendix H.

4-8: WORKBENCH PROJECT SETTINGS WITH GTOOLDB.TCL

The project `tooldb` file describes file types and suffixes that determine how file input and output is done along the tool chain. The "g" stands for "global," which refers to settings that apply to all projects, though here it is used for project-level settings. There are rules for how the Sentaurus Workbench projects should define these settings to avoid overwriting global-level settings with project-level settings. These configurations are done in the TCL programming language. There is a global site-wide dictionary variable called `WB_tool`, that projects can access and should not edit. Instead, projects can create another `WB_tool` variable, and the workbench will consider both variables, using the project level variable contents to overwrite the global level settings for the current project only, without editing the global variable. The variable is an array, keyed by a string, and can be called the tool database. Each entry in the database is a list. In TCL, the list is not strongly typed, and an isolated string is similar to a list with one element.

The global database must be checked, and the project settings code must then be configured depending on the contents of the global database. Done manually, the potential for changes at the global level is not accounted for. This function helps to follow the various rules [62] about when to append to a database entry, and when and how to create a new database entry, depending on the contents of the global database.

```
array set WB_tool {}
proc append_WB_tool_field {field setting} {
   upvar 1 WB_tool WB_tool

   if {[info exists WB_tool($field)]} {
      lappend WB_tool($field) $setting
   } elseif {[globalexists WB_tool($field)]} {
      # no need for [getactual] because of 1st case
      set WB_tool($field) [getglobal WB_tool($field)]
      lappend WB_tool($field) $setting
   } else {
      set WB_tool($field) $setting
   }
}
```

If the field does not exist, it is created. If the field exists in the global database, those contents are copied, and the new field is appended. In this way, the global settings are not changed, but the project settings maintain and add to or overwrite the global settings as necessary. All tool database configuration can be implemented properly by using this function.

4-9: OTHER TOOLS

The simulations are executed on a remote server. The files are uploaded using WinSCP's automatic synchronization function, that immediately uploads when a file is changed, such that it can be tested immediately, and files can be edited locally.

Sublime Text 3 is used to edit the tool command files. It supports code syntax highlighting and *vim*-style text editing. The 'Highlight' plugin is used to copy the code with syntax highlights.

Mercurial and its HgWorkbench GUI is used for version control. This is useful for debugging, since it displays all differences since the previous commit, and provides a *revert* function. It also supports code sharing within the group.

The *PDF Property Extension* software [63] is assistive in organizing PDF files by displaying metadata within Windows File Explorer.

4-10: DEVELOPMENT PRACTICES

The Sentaurus Workbench view usually displays simulation parameter values. The execution time of all nodes can be shown with `ctrl+4`, the execution date is displayed with `ctrl+5`, and parameter values are restored with `ctrl+1`. These functions and others are also accessible from the `View->Tree Options` menu. Typically, the "Add new Experiment" function is used for new developments, adding a new column to the table. Old simulation results are usually saved, since they generally take time to complete, and likely remain relevant. This practice supports the function of the version control tool Mercurial, since it saves preprocessed versions of the source files, which also helps to track code developments. Parameter variables are used for purposes excessive of their intent – each new experiment is given a title using a parameter variable, that has no bearing on the simulation, to help organize the experiment table. The workbench supports a command line argument in which the path of the project can be specified to be opened automatically, which can be preset with the `alias` command in the bash session setup file, `.bash_profile`.

Visualization code preference files include the 'interactive' workbench execution mode, so that results plots are automatically opened for viewing. If a device calculation in progress, dependencies may not allow the visualization code to be started from the GUI. The visualization scripts can contain "@" substitutions, so they must be preprocessed before they are run. This is possible from the context menu, creating a ppNNN_suffix.cmd file that can be run separately. Sentaurus Visual can be run with `svisual -s` to load a preprocessed command file script from the command line. Python can be run with `gpythonsh`, where the preprocessed source file is given as an argument, and the `-i` option can be used to create an environment for interacting with the plots. Version L of Sentaurus Visual is used to open mesh files.

It is possible that updates to the Sentaurus version invalidate command code, especially SDE schema function signatures. To debug this kind of code, it is useful to run the structure editor GUI (with the `sde` command) which prints the up-to-date schema code as operations are conducted.

The output of running processes is visible from the workbench node explorer, or by a feed of the output file viewable with `tail -f`.

Since the simulation flow produces many files, it is too many to consider with a simple `ls` command. The file suffixes can be used to narrow down the output, as in the command `ls *des.out`, which selects only output Sentaurus Device files. Alternatively, `ls -ltr` views recently changed files, `ls *<node_number>*` filters by node number, which is useful when transitioning from the workbench, or `ls | grep -v ^n | grep -v ^pp` filters output files and shows only source code files.

5: RESULTS

5-1: FOUR-JUNCTION CELL WITH FRONT THREE SUBCELLS CURRENT-MATCHED

The first step in the consideration of results is to execute the subcell voltage ramp code to ensure that each segment produces similar current. Supposing base thicknesses of 0.8, 1.9, 1.9, and 170 μm, and 1000x concentrated AM1.5D incident spectrum, the resultant current density, power density, and power density derivative curves are shown in Figure 24.

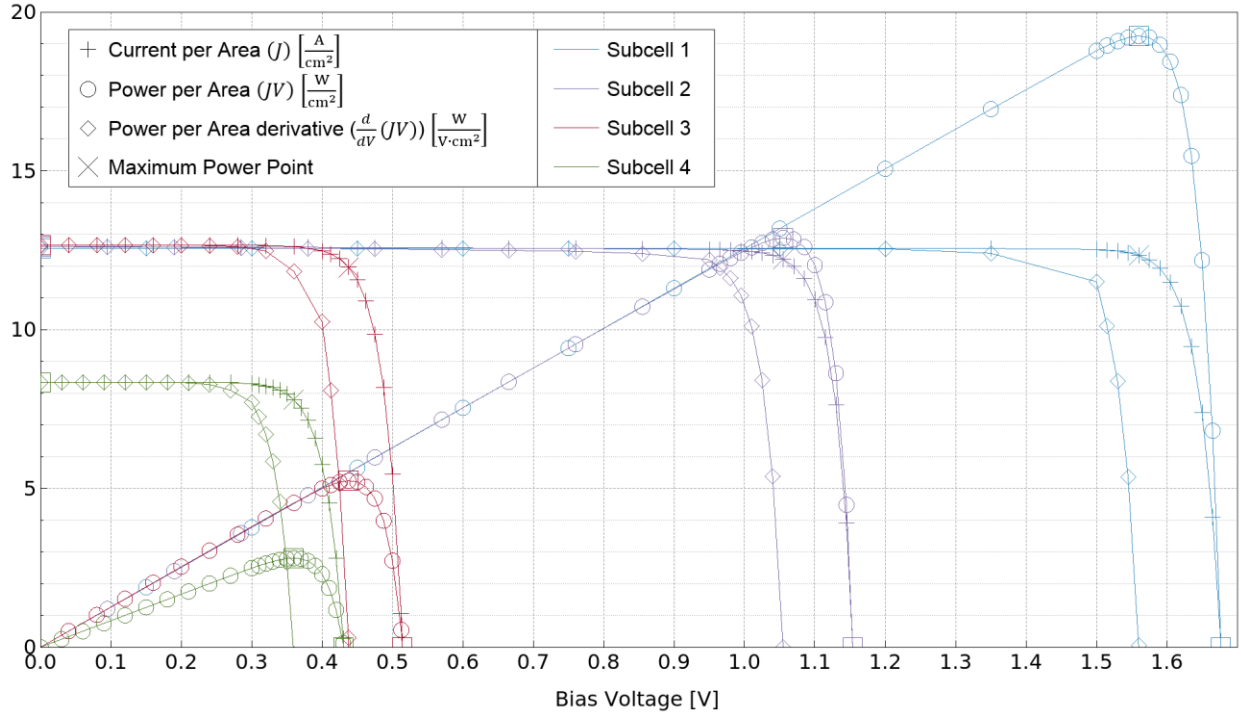

Figure 24: Current density J(V), power density, and power density derivative curves for device with current-matched front three subcells.

The GaInP subcell base is thinned to share current with the second subcell. This assumes a GaInP bandgap of 1.9 eV. The GaInP bandgap can be engineered by adjusting the growth conditions and atomic ordering. The disordered arrangements have higher bandgap, in the range of approximately 1.8 eV to 1.9 eV. [64][65] The highest achievable bandgap would likely be best in this situation, since increasing the bandgap would be a preferable method of sharing photons than thinning the subcell. Both adjustments lead to a higher output voltage, but increasing the bandgap is more effective.

The second subcell can be made thick enough to absorb nearly all photons with energy above the bandgap of $In_{0.01}Ga_{0.99}As$ with a base thickness of 2.5 μm. This matches the current of the GaInP subcell when its base is thinned to 0.8 μm. However, the SiGeSn subcell requires a challenging 2 μm base thickness to match the front subcells in this case. Matching these front three subcells is applicable to the consideration of a triple-junction device without germanium. The optimization problem requires the thickest possible SiGeSn subcell, and the front two subcells can be made to match. In the design shown, the second and third subcells are matching, with both bases at 1.9 μm, and there is extra current for the GaInP subcell. Extra current in the front subcell makes the design

less sensitive to spectral changes due to daily and seasonal changes in air mass. The curves appear to match at the short circuit level because of the additional current available to the GaInP subcell. In precisely this condition, the SiGeSn subcell will operate at a voltage slightly lower than the maximum power point. Slight deviations from the operating voltage, especially below the maximum operating point, in the least power-producing subcell, have only a slight effect on the output power.

The power produced by a triple junction device with these three front subcells would be approximately the sum of the three maximum power points shown in the figure. These are printed by the software as 19.2, 12.9, and $5.2\ \mathrm{W/cm^2}$, resulting in $37.3\ \mathrm{W/cm^2}$ for the triple-junction device. The incident power is $90\ \mathrm{W/cm^2}$, therefore the efficiency is $(37.3\ \mathrm{W \cdot cm^{-2}})/(90\ \mathrm{W \cdot cm^{-2}}) = 41.4\%$.

There is insufficient current remaining for the germanium subcell to match. For more detail, Figure 25 shows the EQE of the device multiplied with the input spectrum. It shows that photons are present but are not all absorbed effectively in the range between the germanium direct bandgap of 0.8 eV (1.55 μm) and indirect bandgap of 0.67 eV (1.85 μm). The integrated area of the portions labeled "segment current" in Table 4 are output by the program as $12.6\ \mathrm{mA/cm^2}$, $12.6\ \mathrm{mA/cm^2}$, $12.7\mathrm{mA/cm^2}$, and $9.0\ \mathrm{mA/cm^2}$, which approximately equal the J_{sc} results from the voltage ramp subproject graphed previously in Figure 24. Many photons in germanium subcell's range are lost to reflection, Auger recombination, and electron loss current as shown. Auger recombination occurs in germanium because radiative recombination is a much lower probability process in indirect materials, being the inverse process to optical generation of electron-hole pairs. The minority carrier loss process, such as the electron loss of the germanium subcell, also occurs similarly elsewhere in the device. Many of these losses can be expected. Due to the indirect bandgap of germanium, the absorption does not take place in the desired location, near the p-n junction. Carriers must travel a large distance in the proper direction from within the substrate bulk by diffusion to be collected. The backing is set as gold, simulating a gold surface contact, providing good reflection of the long wavelengths in the TMM optical calculation. In the simulation, this only affects the optics, as all contacts are considered to be Ohmic. In reality, a gold contact does produce a good Ohmic contact as well as the mirror effect, though it does add to the cost of the device, which is justifiable in these scenarios. [66] A textured back reflector, or other light trapping strategies, could improve upon this design by increasing the germanium absorption in these wavelengths where it is indirect. These strategies would benefit the four-junction devices, which appear from these results to require significant thinning in the front three subcells to achieve current matching. Such strategies can improve the minority carrier loss by leading to a reduction of subcell thickness, increasing the likelihood of electron hole pairs being created where the electric field exists near the p-n junction, and reducing the volume of the bulk region where minority carriers can diffuse in the undesirable direction. The Auger recombination is also improved by reducing the distance through which majority carriers must travel. This design does include a large absorbing distance, and the gold back reflector, and yet a high degree of thinning must still be employed for the fourth subcell to produce suitable current due to the inefficacy of the germanium in absorbing photons by the indirect absorption process. This shows that SiGeSn engineered with direct bandgap of 0.685 eV could be used to design a preferable fourth subcell.

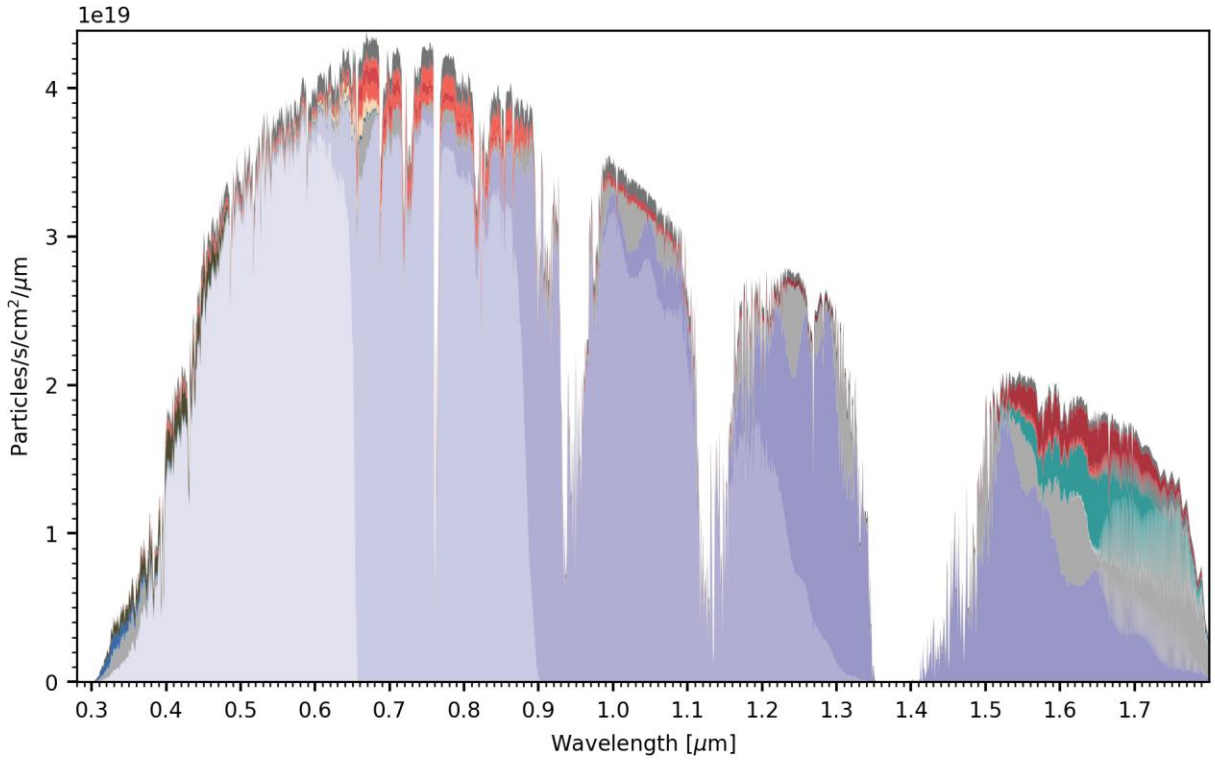

Figure 25: Device EQE multiplied by AM1.5D spectrum converted to unit of $photons/s/cm^2/\mu m$. Legend is Table 4. Vertical axis unit "Particles" can refer to a photons of incident light (reflected, transmitted, or the sum total equalling the incident,) or to electron or hole particles of electric current as in the EQE or loss currents.

Table 4: Legend for Quantum Efficiency Plots.

Segment 1 current	Segment 1 hole loss current
Segment 2 current	Segment 2 hole loss current
Segment 3 current	Segment 3 hole loss current
Segment 4 current	Segment 4 hole loss current
Grid finger shade	Segment 1 electron loss current
Reflection	Segment 2 electron loss current
Transmission	Segment 3 electron loss current
ARC 1 AbsorbedPhotonDensity	Segment 4 electron loss current
ARC 2 AbsorbedPhotonDensity	Radiative recombination
Tunnel 1 loss current	SRH recombination
Tunnel 2 loss current	Auger recombination
Tunnel 3 loss current	Surface recombination

The EQE result prior to the multiplication with the AM1.5D spectrum is shown in Figure 26. This shows several device characteristics more clearly, such as the loss of energy to reflection, and recombination trends. The plot does not specify in which layer the recombination takes place, though the fill plot does stack the recombination sources in order by layer from front to back. The EQE figure can be cross-referenced with Figure 27 to identify the location of the recombination. Radiative recombination in the GaInP base layer is significant, radiative and SRH recombination from the GaAs emitter are apparent, and radiative recombination from the GaAs base is also

prominent. In SiGeSn, the SRH recombination is strongly influenced by the lifetime input parameter set to 1 μs, discussed later, which appears next as a loss in the EQE figure. Undesired absorption in the tunnel layers leads to a loss current in the terminals adjacent to the tunnel, which is only apparent from the first tunnel junction, since it is subject to irradiance of shorter wavelengths of light. The gold backing produces a detailed interference pattern [67] that causes the Sentaurus Device solver to reduce the step-size as it finds convergence. This part of the result assumes such precise wavelengths and interfaces that it would be more valid to average the detailed output over small wavelength ranges to obtain more realistic results. Note also that the spectral data is not published in such small intervals. Care should be taken in averaging this data, perhaps by resampling, in case the inconsistent step size would create a systematic error.

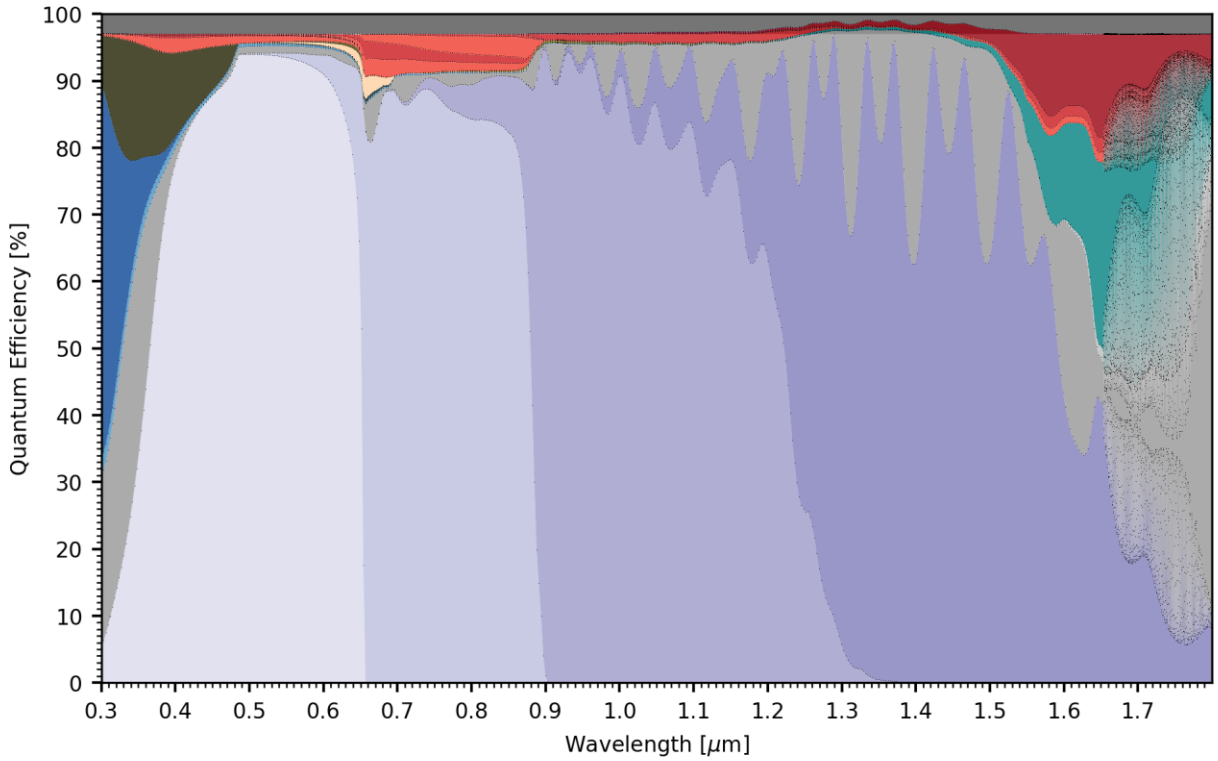

Figure 26: External quantum efficiency.

Table 5: Legend for Figure 27 a-d and Figure 33.

arc1	td1_phigh	td2_phigh	td3_phigh
arc2	td1_nhigh	td2_nhigh	td3_nhigh
sc1_fsf	sc2_fsf	sc3_fsf	sc4_fsf
sc1_em	sc2_em	sc3_em	sc4_em
sc1_base	sc2_base	sc3_base	sc4_base
sc1_bsf	sc2_bsf	sc3_bsf	sc4_bsf

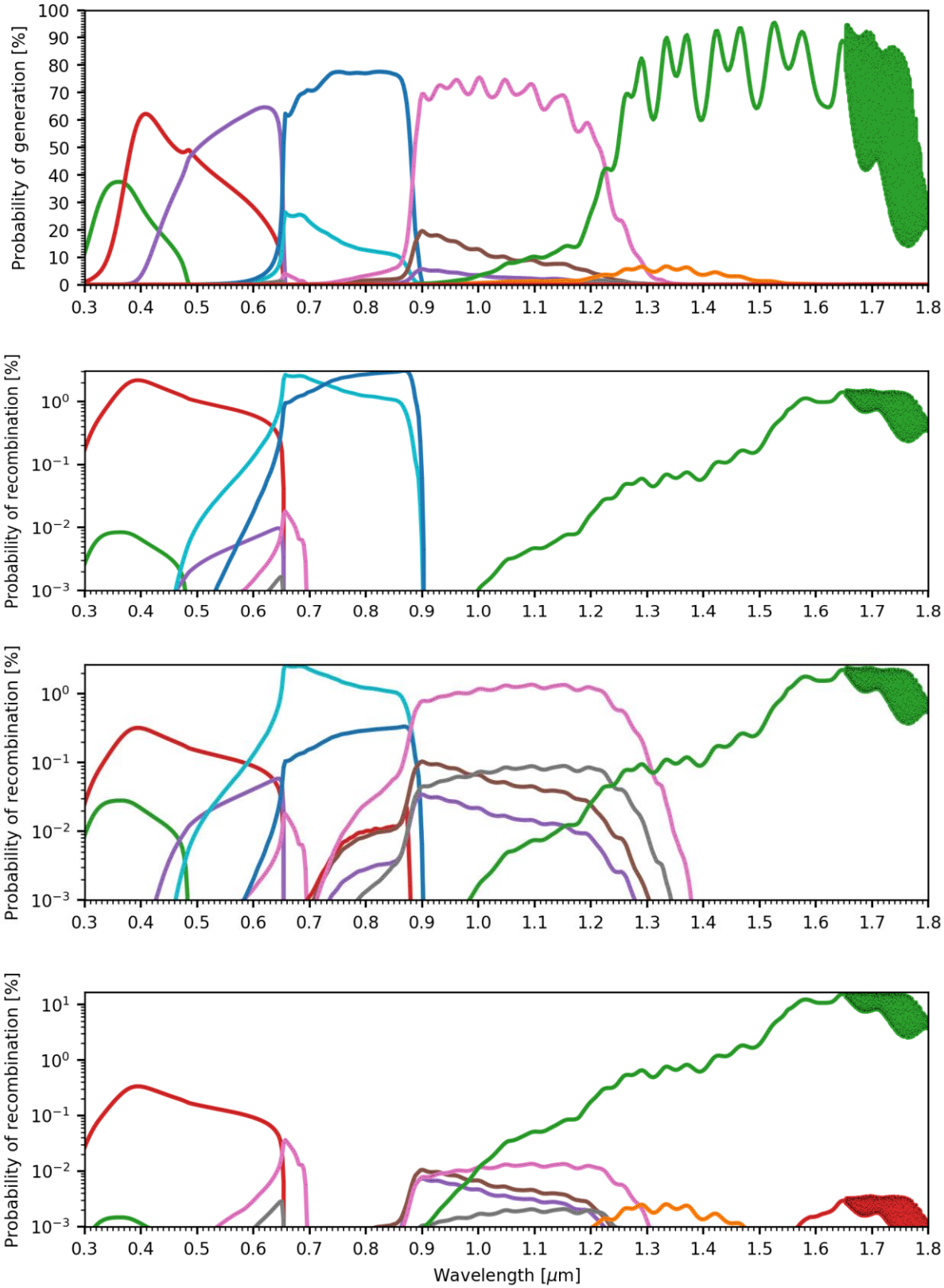

Figure 27 a-d: Integrated a) optical generation, b) radiative, c) SRH, and d) Auger recombination in each layer. Legend is Table 5.

The IQE result is shown in Figure 28. This removes the effect of reflection out of the device, and shows more clearly the response internal to the device [59].

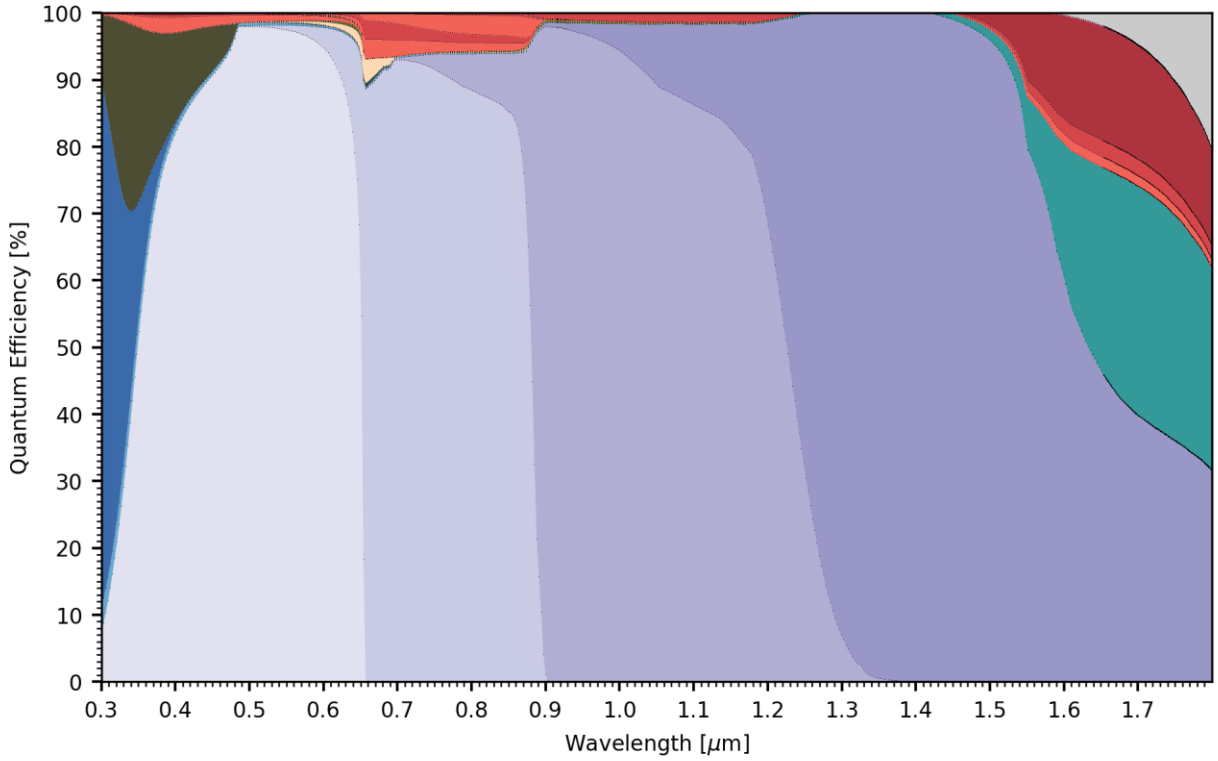

Figure 28: Internal quantum efficiency.

The EQE and IQE plots have been formed by converting the loop currents of Kirchhoff's current law to quantum efficiency, dividing by the electron charge, and comparing the number of flowing electrons to the incident number of photons. Additionally, recombination rates output by Sentaurus Device are converted to a comparable unit. Summing up these output effects, and comparing the sum of the output to the input, is a form of black box testing. Without investigating closely the detailed calculations, we can observe an error in the result, appearing as a clear discrepancy between the incident number of particles and the sum of reported effects. Of course, Sentaurus Device solves the equations accurately according to the set of input information, including the mesh.

The electrical current plotted as EQE and IQE follows from the optical generation within the device. This quantity can also be converted to a unit comparable to EQE for comparison, by comparing the integrated generation rate to the rate of arrival of photons. In the following plot, Figure 29, the sum of all integrated optical generation is compared with the reflection, transmission, and grid finger shading.

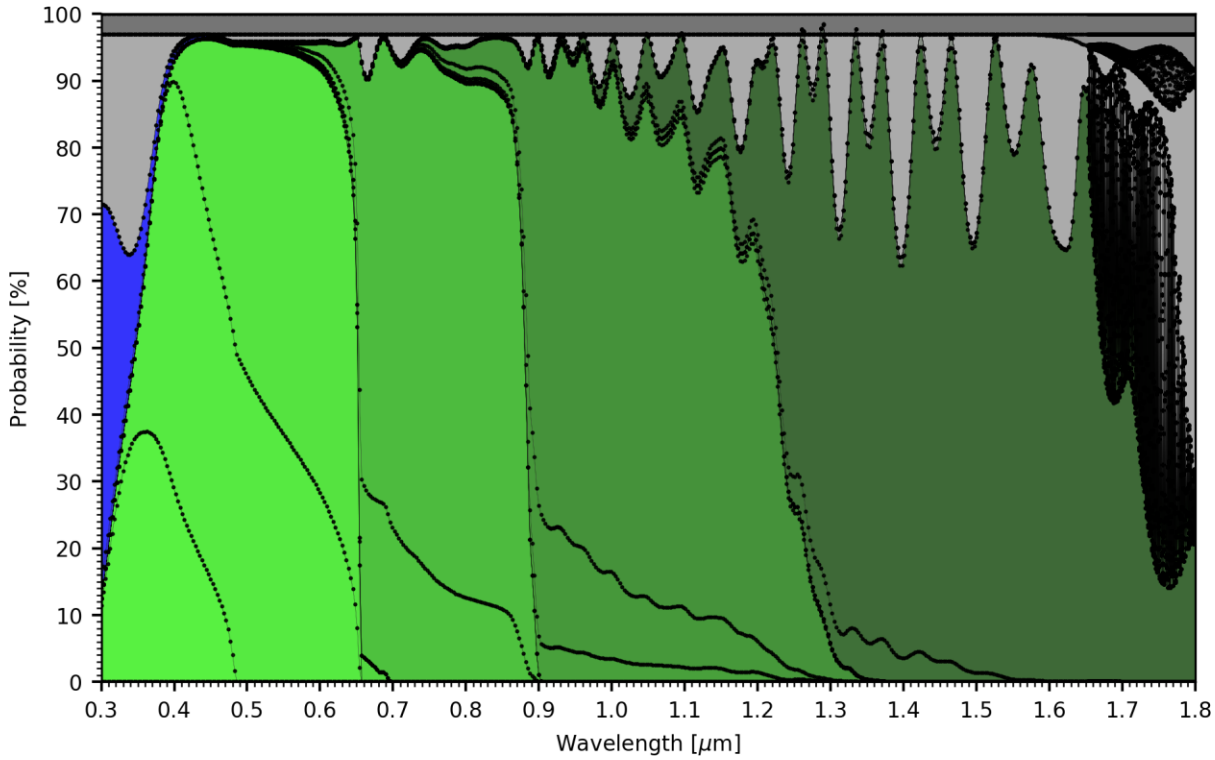

Figure 29: Summation of optical generation by layer.

A similar error is present. The difference of the sum of all quantities from 100% is shown in Figure 30. The quantities shown in this plot are calculated prior to the application of Poisson or the continuity equations. Hence, this is the original source of the error, which propagates through and also produces error in the EQE.

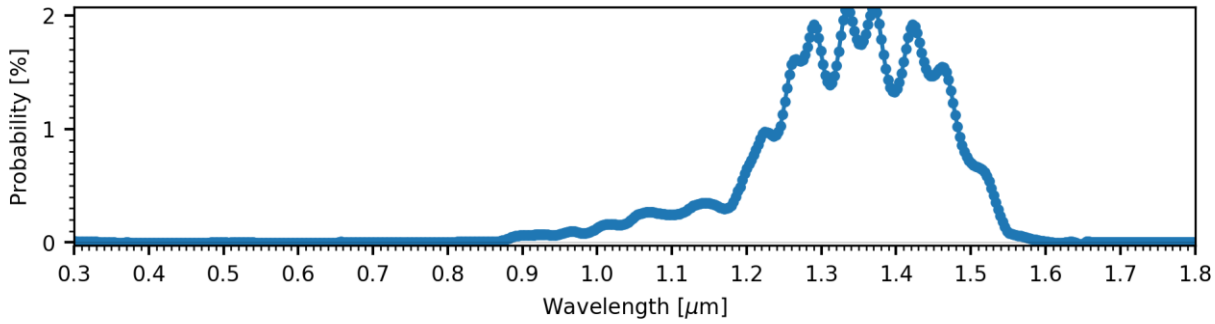

Figure 30: Difference between the total integrated optical generation and the probability of light absorption according to the TMM optics result.

A similar conversion can be made to remove the effect of reflection, which distorts the effects within the device, by dividing by $1 - R$ and by the grid finger coverage factor. This is similar to supposing that there is no light reflection, since the physical quantities are linearly scalable in this regime.

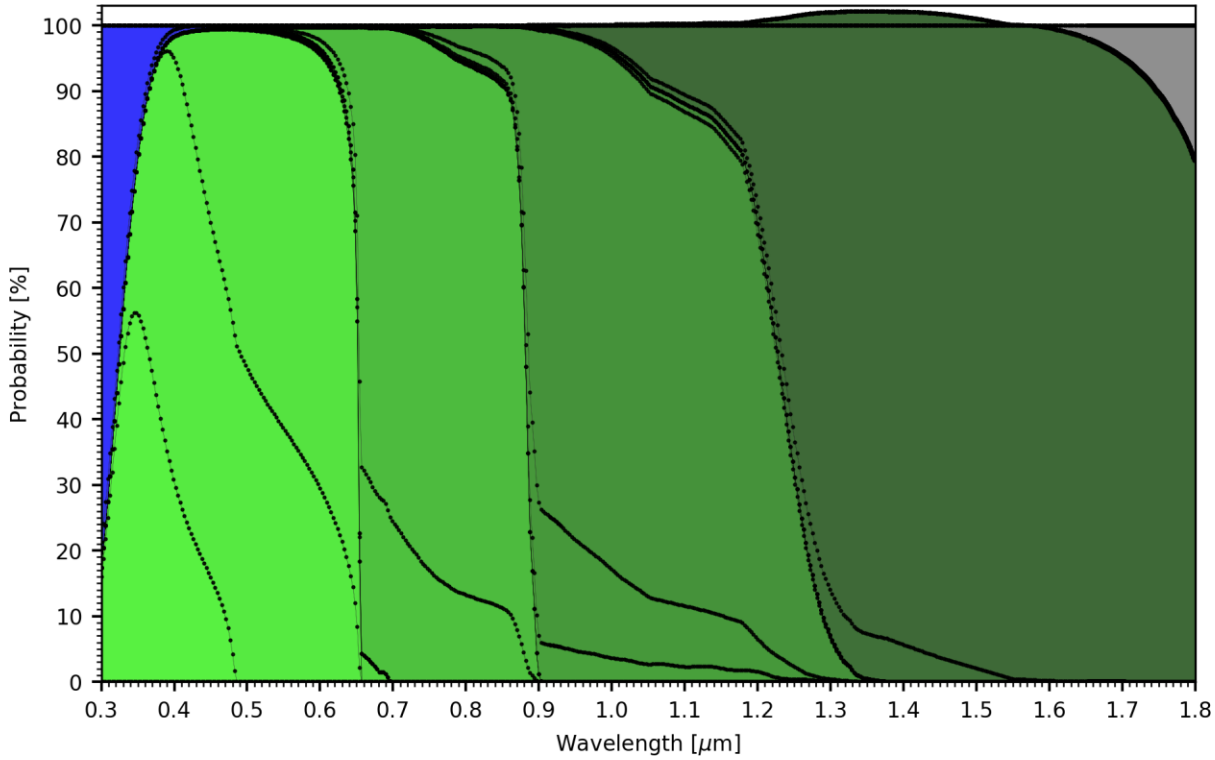

Figure 31: Summation of optical generation renormalized by reflectivity and grid shade.

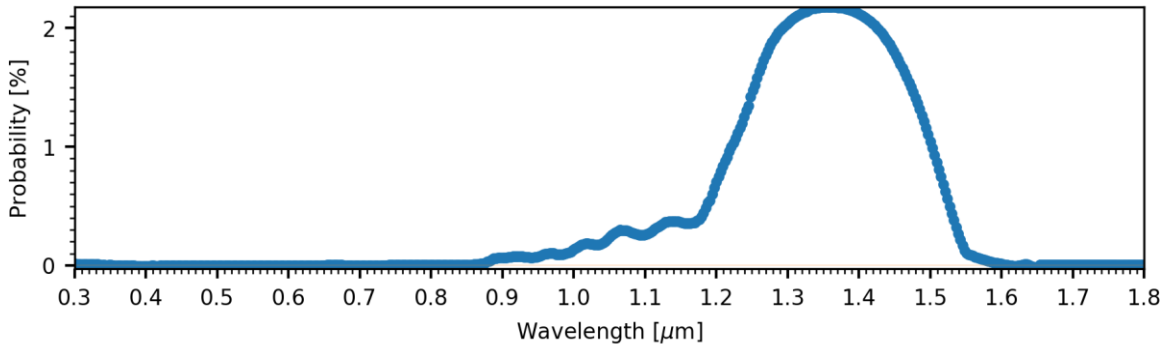

Figure 32: Discrepancy in optical generation total, renormalized to IQE unit.

The optical generation in the germanium is an exponentially decreasing function in the y-dimension. The calculus error as the mesh samples the optical solution and assumes linearity between the samples is to overestimate the total generation as the numerical integration is performed. This error can only be decreased asymptotically when simply increasing the number of mesh points. However, it is very useful to precisely determine the location of the error.

To locate mesh-related errors in optical generation, the integrated optical generation in each layer and the output discrepancy of the summation are normalized with respect to reflectivity, with the factor `eqe_to_iqe` of the `physics` object and compared in log scale. These figures show that the error occurs at wavelengths where there is absorption in the germanium emitter or base.

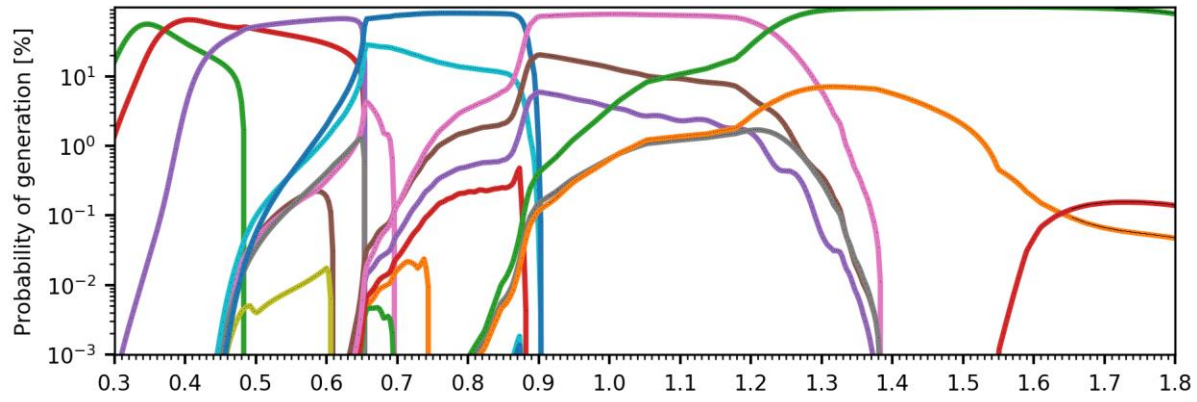

Figure 33: Optical generation by layer, normalized with respect to reflectivity, log scale. Legend is Table 5.

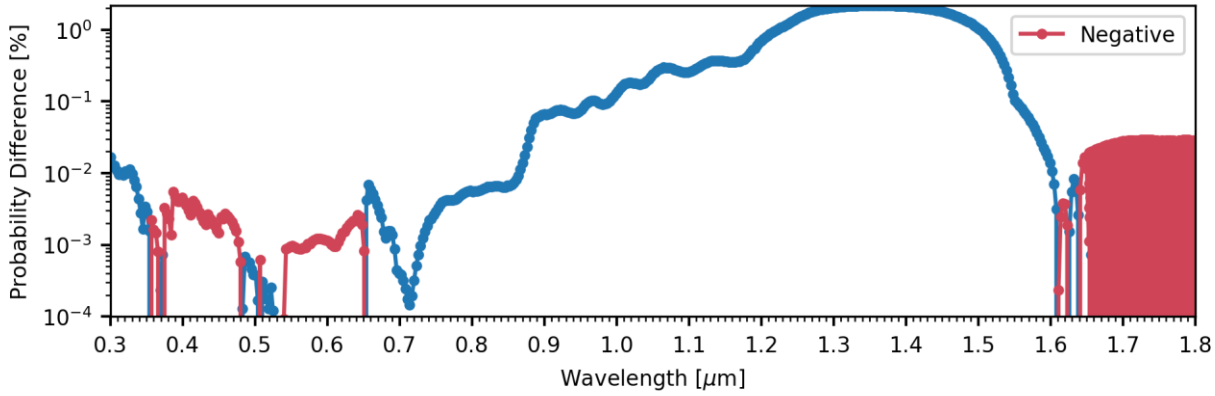

Figure 34: Discrepancy in optical generation total (overlap in Figure 31) in IQE unit, log scale.

As shown in the IQE discrepancy plot, Figure 35, the solution of the electrical equations contributes mesh error subsequently, and the net error can be dominated by the initial optical meshing inaccuracy, or by the subsequent mesh error in the electric solution. The difference occurs in the 0.55 to 0.9 um range. The optical error dominates again beyond 0.9 µm, (below 1.4 eV, which is the bandgap of (In)GaAs); this is due to the error in approximating the optical generation curve in the SiGeSn layers in the wavelengths where (In)GaAs becomes transparent.

Error of this magnitude is not necessary. Additionally, allocating mesh points well improves the completion time of the simulation, and can help lead to convergence of the solution algorithm, avoiding unsuccessful solution attempts. Improving the speed of the solution can enable the use of a genetic algorithm to aid in the determination of optimal layer thicknesses.

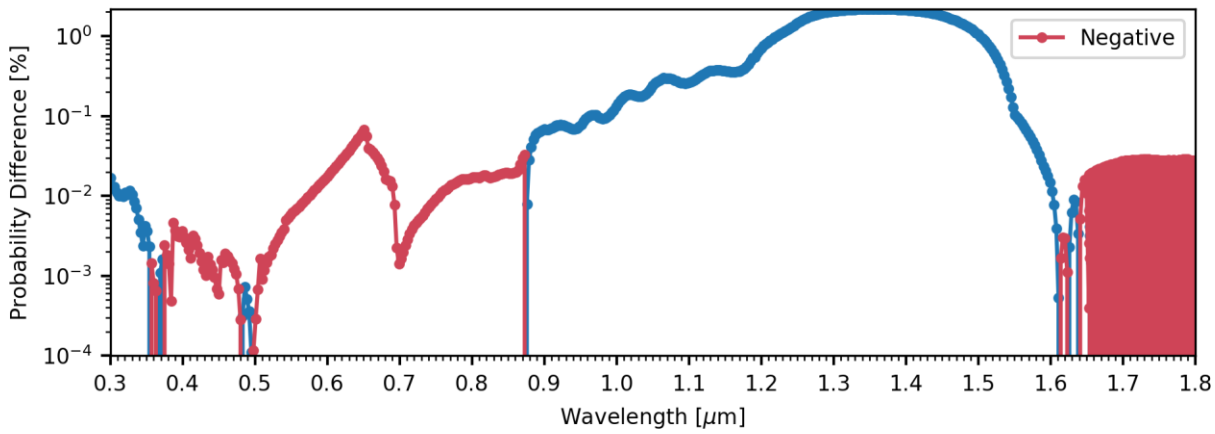

Figure 35: IQE discrepancy plot (overlap in Figure 28), log scale.

5-1-1: Error Reduction by Mesh Adaptation

To investigate the mesh-related errors, it is best to start with the '1d' project configuration, which sets dX_{min} and dX_{max} to equal $w/2$, such that there are only two columns of mesh points in the lateral dimension. This configuration removes the GaAs cap from the structure and only defines the components of the device that are full plane layers. By creating symmetry in the *x*-dimension, a much faster solution to the device is possible, though it is an approximation that does not require the electron and hole currents to have a lateral component. At the front, an unrealistic transparent virtual contact is placed on the front window surface instead of on a grid finger. Adaptations can be made in this configuration before working with the much more computationally expensive '2d' mesh configuration. In the code, the *y*-dimension mesh functions can be defined and called first, independently, and the *x*-dimension functions can be added after if necessary.

The meshing can be adapted in the Sentaurus Structure Editor command file. Increasing the mesh in the germanium emitter layer has no effect. The overestimation of optical generation occurs at the front of the base layer, which has larger distances between mesh points. It is apparent that the error related to optical generation is dominant at this point. This rules out quantities that first appear in the electrical equations, such as the electric field and electron and hole currents. Mesh related error in those quantities should be addressed subsequently. It can be appropriate to organize mesh constraints by physical quantity in the code, since the mesher will implement the strongest constraint. Increasing the mesh at the front of the germanium base does have impact and reduces the error in the region above 1.2 µm (Figure 36). Defining a mesh rectangle of higher mesh point density in the first 8th of the germanium base reduces the error for a large portion of the observed error, but then leads to another place behind that refinement rectangle where error is produced for light of wavelength relatively likely to absorb in that area.

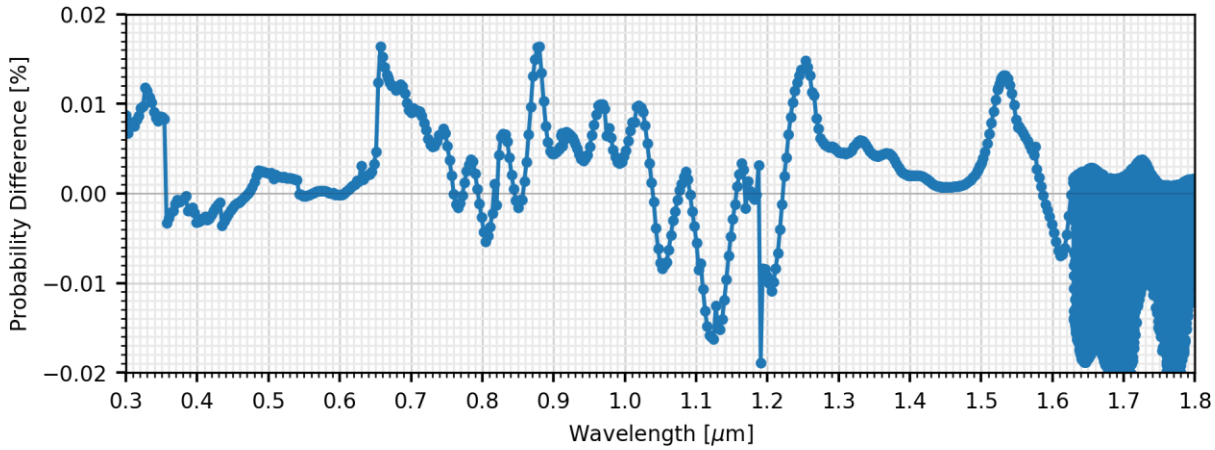

Figure 36: Discrepancy in optical generation integration with a mesh multibox added in the front eighth of the germanium base. The error in the range of 1.3 to 1.5 μm is reduced, but a peak then appears at 1.55 um, requiring a different approach. Nearly all features of this curve are due to the mesh approximation of the optical generation curves within the device.

The Sentaurus mesh tools take three types of arguments as input, a minimum mesh element size, a maximum element size, and a ratio. The ratio is a growth factor that is multiplied to the element size at each step, and defines the growth of the element size constraint through the distance into the bulk in an exponential manner. This is how the mesh multibox functions, as well as the interface mesh function, called "`MaxLengthInterface`" or "`MaxLenInt`". To evenly allocate mesh points to reduce the error from optical generation in the large substrate layer, and in the other base layers, the element size should be determined using the ratio input to the Sentaurus mesh tools. This produces a growth trend of element sizes that matches the decay of optical generation. The maximum element size argument need not be used, such that the element size constraint can be determined with the ratio. If the base mesh is defined with the maximum element size argument, the mesh spacing will be constant for most of the bulk, leading to an error profile like Figure 32, with too few points at the front of the layer. Patching that location only moves the error source. The ratio approach to meshing for Beer-Lambert optical generation can be used to allocate points appropriately throughout the large base layers.

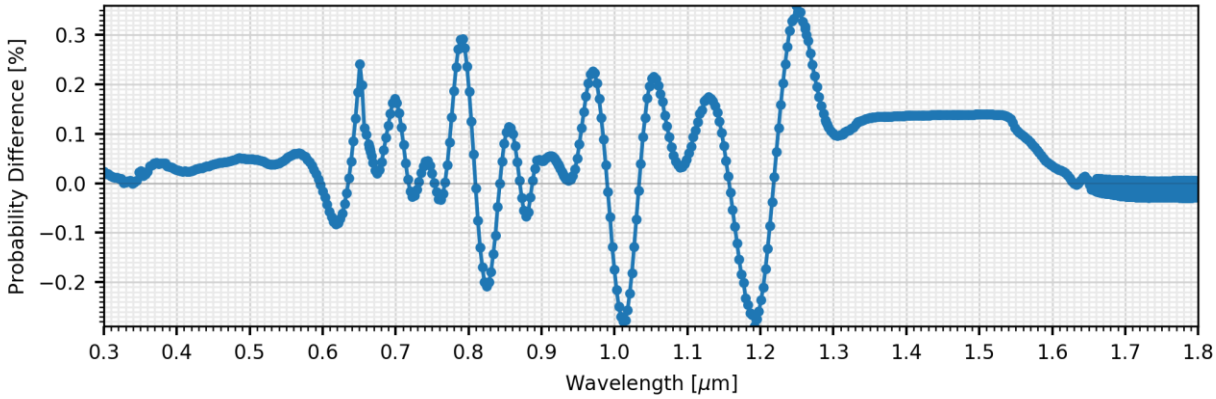

Figure 37: Discrepancy in integrated generation normalized by reflectivity for device meshed entirely with ratio method, 50 pm minimum element size, and 1.15 ratio.

In this result, the germanium base was meshed with the ratio approach. The initial element size is 50 pm, which is very small and not relevant to most of the depth of the layer. The ratios applied are 1.15 and -1.15. The negative ratio indicates that it is applied to the bottom of the region. The flat portion from 1.34 μm to 1.54 μm occurs since the mesh element sizing is growing appropriately compared to the decreasing optical generation and the error through that region is constant. However, the error is strongly reduced at longer wavelengths. This error reduction is not necessary, given that there is much larger error contribution elsewhere in the device. The reduction occurs because the -1.15 ratio produces mesh points at the back of the 170 μm germanium base. This -1.15 ratio input is very expensive in terms of computation time, since it leads to many unnecessary mesh nodes in the back half of the germanium base. The layer need not be meshed symmetrically to control the dominant error, that of optical generation. A more efficient approach would be to delete the -1.15 ratio input, and allow the plateau to continue beyond 1.54 μm. Then, decreasing the positive 1.15 ratio will reduce the height of the plateau uniformly as necessary. Given that the germanium base is the largest layer, it is the region in which it is most expensive to control the optical generation error in terms of computation time, making it important to control that error efficiently.

Optical generation from light that was reflected from deeper in the device occurs primarily at the back of a base layer in the front subcells, which has the opposite trend, requiring relatively smaller points at the back surface of the base instead of the front. However, this light always has wavelength that was long enough to penetrate the entire base from the front. Therefore, the approach of constraining the mesh element sizes with a ratio-based development throughout the thickness of the base starting at the front without including a negative ratio is sufficient to also control the error from reflecting light incident on the back. The mesh refinement constraint at the back of the base layers can be defined only in consideration of the electrical quantities.

The error waves in the region from 0.6 to 0.87 μm are produced by the insufficiency of the mesh spacing to resolve the optical generation differences caused by interference patterns in the InGaAs layers, and the waves in the range of 0.87 to 1.3 μm are similarly caused in the SiGeSn layers. The Sentaurus Device optical solver has the option "`IntensityPattern=Envelope`", which produces a solution that removes the interference patterns. This offers a way to control the error produced when the mesh spacing is too large to resolve the interference pattern, and that option is on in the germanium base. Alternatively, the mesh spacing can be reduced in these regions by decreasing the ratio argument; Decreasing the initial element size argument has negligible effect in the central regions of the base layers where these errors are produced.

The error produced in the germanium base is strictly positive, because the error produced by finite differential elements on a decreasing function is positive. The error when there are interference patterns can become negative, since the optical generation is no longer a strictly decreasing function. Another phenomenon that can cause a negative error to occur is related to the initial element size at the boundary of a transparent layer and an absorbing layer. This initial element should be much smaller than elements along the decreasing optical generation curve, since it must resolve the large step from zero optical generation to the non-zero onset at the interface of the absorbing layer. If this element is very large, a negative error is produced for the range of wavelengths likely to absorb near the front of that layer.

This function implements a method that is used to reduce ratio arguments systematically. A ratio must remain above 1. This is required to maintain the capability of the mesh to be scaled for fast, approximate results, or slower, more accurate results. Using a dividend argument of 100 to this function transforms a ratio of 1.15 to 1.0015. This value is of proper magnitude to control the slow increase of mesh element sizes required to efficiently manage the error introduced by sampling the optical generation function with respect to distance.

```
proc R_divide {R divider} {
    if {$R>0} {
        return [expr  1.0+($R-1.0)/(1.0*$divider)]
    } else {
        return [expr -1.0+($R+1.0)/(1.0*$divider)]
    }
}
```

To visualize the mesh that is produced by these inputs, plotting code is developed in Sentaurus Visual TCL. The mesh determines the values in the dataset of the independent variable produced by a Sentaurus Visual cutline. The independent variable is the depth into the device, y, and the mesh spacing is reflected in the step size between entries in the y data. To execute the necessary subtraction, the y data array is copied once with a zero added to the front, and once with a zero added to the back, and the two arrays are subtracted. This method is required because the Sentaurus Visual arrays in a dataset must have the same length. This is an excerpt of the code that produces the mesh visualization.

```
set sdevice n@node|ramp@_des
load_file ${sdevice}.tdr
create_plot -name structure_plot -dataset ${sdevice}
set xcut [expr 0.5*@w@/2.0]
create_cutline -name structure_cutline -plot structure_plot -type free -points {$xcut 0 $xcut 180}
remove_plots structure_plot
create_variable -dataset mesh_spacing -name Y_padded1 -values "0 [get_variable_data -dataset \
    structure_cutline Y]"
create_variable -dataset mesh_spacing -name Y_padded2 -values "[get_variable_data -dataset \
    structure_cutline Y] [lindex [get_variable_data -dataset structure_cutline Y] end]"
create_variable -dataset mesh_spacing -name Y_spacing -function \
    "1000*(<Y_padded2:mesh_spacing>-<Y_padded1:mesh_spacing>)"
create_plot -1d -dataset mesh_spacing -name mesh_spacing_plot
create_curve -dataset mesh_spacing -name Y_spacing -axisX Y_padded1 -axisY Y_spacing
```

An example of the mesh spacing plot produced by this code is shown in Figure 38.

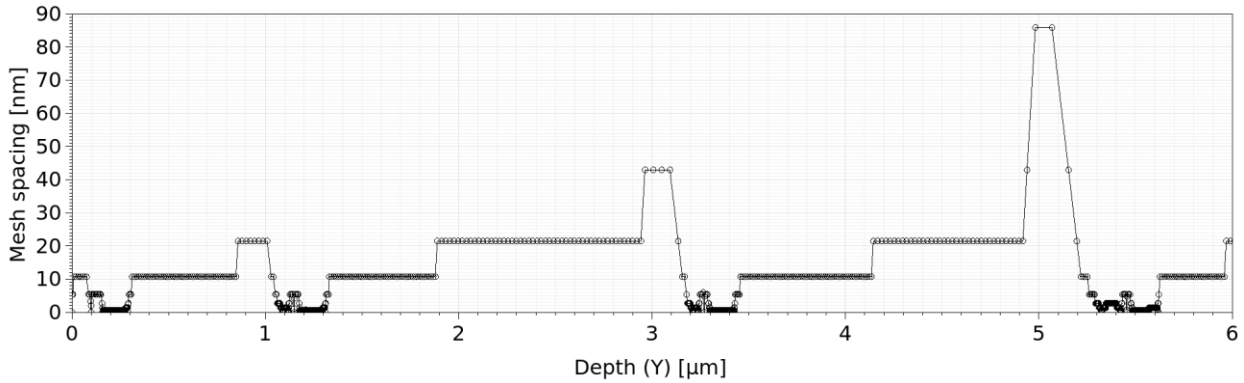

Figure 38: Visualization of mesh point spacing through the first three subcells of the device.

Another plot useful in improving the mesh is a plot that shows all important quantities throughout the device. In the wavelength ramp, the entire device is not active at once. The subproject that

simulates the solar spectrum incident on the device saves the state at 0 V, which can be used to obtain a convenient view of the trends of the physical quantities that the mesh must resolve. Since all quantities have much different magnitudes and units, they are all normalized with respect to their maximum value, and that value is printed in the legend, such that all the quantities can be viewed on the same scale. The code for this plot is in Appendix I.

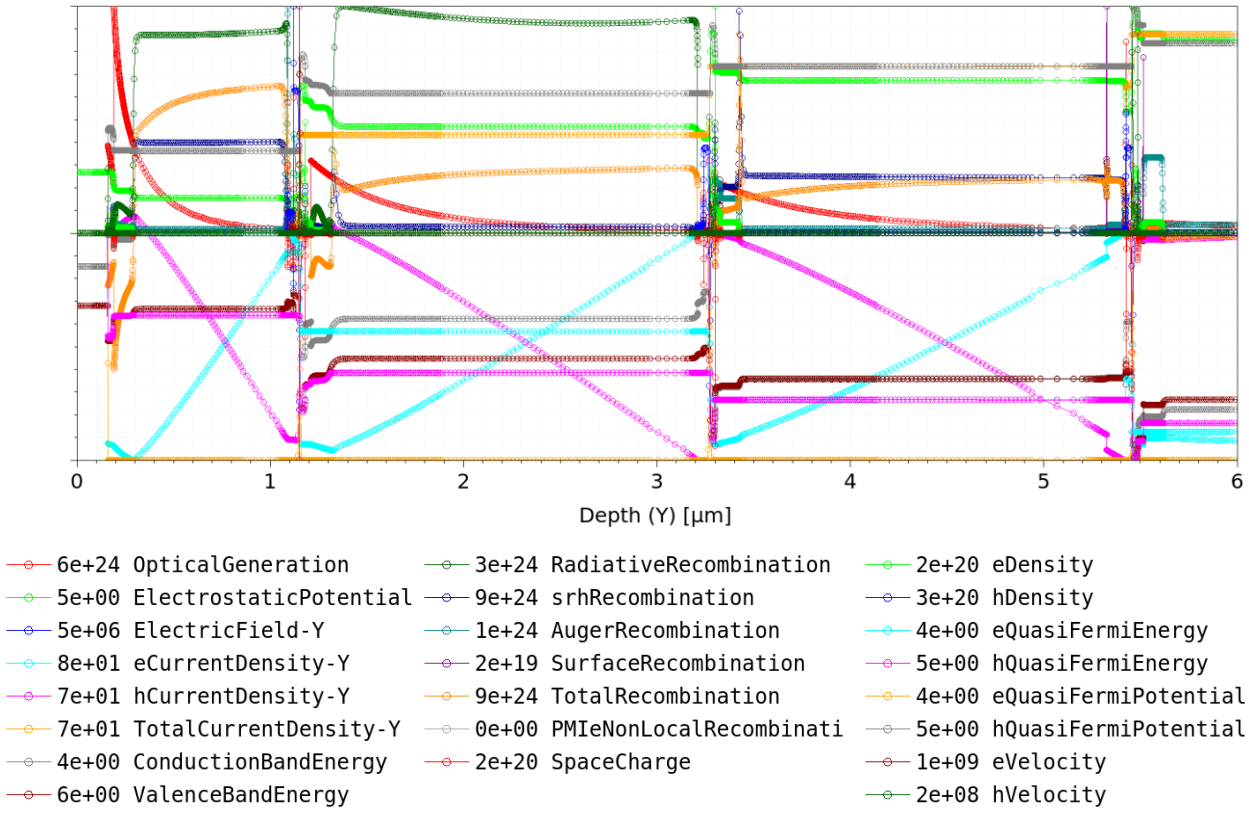

Figure 39: Plot of many physical quantities to inform mesh adaptation.

The optical generation curve is of primary importance with respect to initial mesh point spacing requirements. This plot is convenient because the location in the device is readily apparent from the family of curves. It can be difficult to determine the exact location of features of a singular curve based only upon the depth value.

This mesh produces the following error profile:

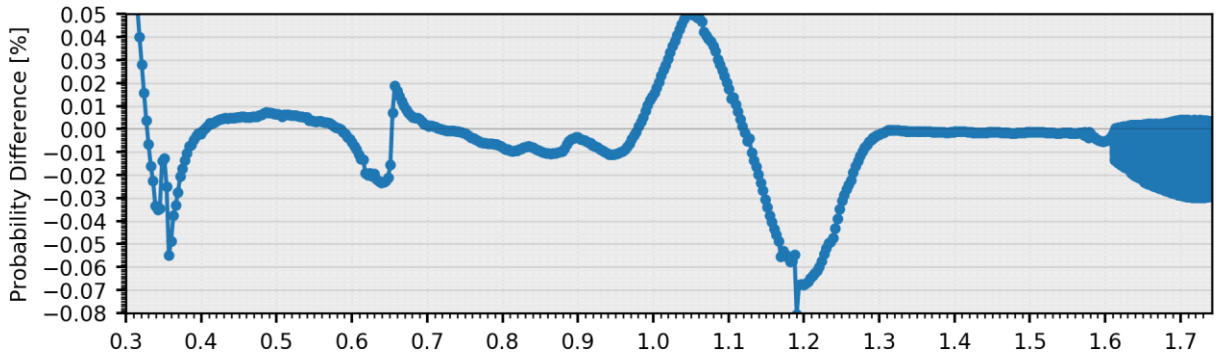

Figure 40: Discrepancy in generation normalized by reflection produced with mesh of Figure 38.

These plots are informative about how insufficient meshing in one location of the device is reflected in the wavelength ramp. This error is produced because the mesh must resolve interference patterns in the SiGeSn base. This mesh spacing is almost sufficient to resolve the interference patterns. The required spacing is related to the wavelength of the interference pattern, and therefore, can be related to the wavelength of the incident light. The mesh was produced with two multiboxes, and the low ratio front (1.016) multibox did not extend through the entire SiGeSn base. This produced a small area of large mesh spacing, which resulted in the observed error pattern. The interference pattern within the SiGeSn base was aliased by this mesh in a way that produced one wave, that clearly depicts the range of wavelengths affected by that probing mesh irregularity. The large error source can affect wavelengths that are more likely to be absorbed in the germanium subcell, because they are still producing some generation in the SiGeSn base. Extending the front multibox further does remove this effect and improves the error in the result.

To mesh the emitter and base for optical generation, an initial element size is chosen for the front of the emitter, and as well as ratio. It's clear from the discrepancy plots when the ratio is sufficiently low (~1.002). This depends on the reflection and interference in the device. The multibox can be applied to a window encompassing both layers.

The ratio methods are most suitable for adjusting the mesh at a surface. A very small element can be placed, and with a ratio around 1.2, the element growth is fast, so such a refinement function will only create points near the surface. This can be used in tandem with another constraint that sets the element size required in the bulk. Having a very small element near the surface can be important to resolve surface effects without introducing the error of a differential element too large to resolve the effect. These surface functions can be applied with the "`MaxLenInt`" function that requires naming both regions of the interface, and supports a "`DoubleSide`" option, or with a multibox with an edge on the surface.

The ratio method can be applied to create gradually increasing mesh throughout a larger bulk distance. The calculation of what the element size will be after some distance for a given initial element and ratio is necessary. One approximation is that, for any sufficiently small initial element, if the ratio is 1.1, the final element will be $1/10^{th}$ of the distance covered, or a ratio of 1.001 can lead to a final element about $1/1000^{th}$ of the distance covered, etc.. Another approach is to choose the desired element size at some given distance, and to use this recursive Scheme function as a

calculator to determine the necessary ratio. A function like this can also be used within the structure editor code, but using an extended multibox region can serve the same purpose.

```
(define (element_size T min_element_size R counter)
  (if (<= T min_element_size)
    (begin
      (display counter)
      (display " elements.")
      (newline)
      min_element_size
    )
    (element_size (- T min_element_size) (* R min_element_size) R (+ counter 1))
  )
)
(define (print_element_size T min_element_size R)
  (display "element_size ")
  (display T) (display " ")
  (display min_element_size) (display " ")
  (display R) (display ": ")
  (display (element_size T min_element_size R 1))
  (newline)
)
(print_element_size 10 .1 1.04)
;; 10 nm distance, 1 Å initial element, ratio 1.4 creates:
;; 42 elements.
;; 0.5 nm element size at 10 nm depth
```

A good alternative is to use regular mesh refinement boxes. They can be used to specify point spacing in the simplest imaginable manner: with a region specification and element size specification to be applied within that region. By referencing an initial solution of the physical quantities, the distance over which small elements are required can be very apparent. This information can be used to create regular mesh refinement boxes. The ratio functions, "`MaxLenInt`" and the multibox, create steps in unpredictable locations. They usually do not create smooth grading. The regular boxes are well-suited for depletion region mesh refinement. Another strength of regular multiboxes is that they scale well. The distance of a small element multibox can be increased, or the element sizes of all regular refinement boxes can be decreased globally by adding a scalar variable to the argument. The ratio functions would require both arguments to be scaled down in an unclear proportion.

To work freely with the mesh functions, they are implemented as TCL functions that print the Scheme code, shown in Appendix J. These functions implement the window, definition, refinement, and placement infrastructure required by the structure editor and mesher, and offer fewer arguments, for usability, since *x*-dimension meshing can be done separately. The output is formatted and readable, so it is possible to see how the device was meshed from the code input TCL, output Scheme, or from the mesh spacing plot.

Appendix K describes a one-dimensional mesh effective at controlling the error introduced by the discretization of optical generation. The ratio-controlled meshing from the front controls the error, without the addition of unnecessary points and computation in the back half of the large base layers. For segmented devices, one multibox can be defined over a group of segmented subcells, which matches the decreasing trend of optical generation over the set of components that have similar absorption properties and therefore similar mesh element size requirements. The use of procedures and loops makes the definition more concise, however, some precision is lost in the

generalization. It can be more effective to mesh the device layer-by-layer for a more verbose mesh definition, in which each region can be specified independently.

5-2: Four-Junction Device Current-Matched by Thinning Front Subcells

The four-junction device is current matched with thicknesses shown in

Table 6. The front three subcells are much thinner, with base thicknesses of 0.41, 0.80, and 0.68 μm. The GaInP emitter thickness has also been reduced to 0.03 μm. This large degree of thinning allows all four subcells to produce similar current.

Table 6: Four-Junction Device Design with Thinned Front Subcells.

LAYER REGION LABEL	MATERIAL	FILE	THICKNESS (um)	DOPING (/cm^3)	MOLE FRACTION	LAYER NUMBER
arc1	MgF	MgF.par	0.100			;#01
arc2	TiOx	TiOx.par	0.060			;#02
sc1_fsf	AlInP	AlInP.tcl	0.030	6e18	0.50	;#03
sc1_em	GaInP	GaInP.tcl	***0.030***	5e18	0.50	;#04
sc1_base	GaInP	GaInP.tcl	***0.41***	-8e16	0.50	;#05
sc1_bsf	AlGaInP	AlGaInP.tcl	0.030	-3e19	0.25	;#06
td1_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30	;#07
td1_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50	;#08
sc2_fsf	AlGaAs	AlGaAs.tcl	0.030	1e19	0.80	;#09
sc2_em	InGaAs	InGaAs.tcl	0.1	3e18	0.01173	;#10
sc2_base	InGaAs	InGaAs.tcl	***0.8***	-1e17	0.01173	;#11
sc2_bsf	AlGaAs	AlGaAs.tcl	0.030	-1e19	0.20	;#12
td2_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30	;#13
td2_nhigh	GaAs	GaAs.tcl	0.030	6e19		;#14
sc3_fsf	SiGeSn	SiGeSn.tcl	0.025	2e19		;#15
sc3_em	SiGeSn	SiGeSn.tcl	0.1	1e19		;#16
sc3_base	SiGeSn	SiGeSn.tcl	***0.68***	-1e18		;#17
sc3_bsf	SiGeSn	SiGeSn.tcl	0.1	-2e18		;#18
td3_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30	;#19
td3_nhigh	GaAs	GaAs.tcl	0.030	6e19		;#20
sc4_fsf	GaAs	GaAs.tcl	0.030	7e17		;#21
sc4_em	Germanium	Ge.tcl	0.100	1e19		;#22
sc4_base	Germanium	Ge.tcl	***170***	-7e17		;#23
sc4_bsf	Germanium	Ge.tcl	0.100	-3e18		;#24

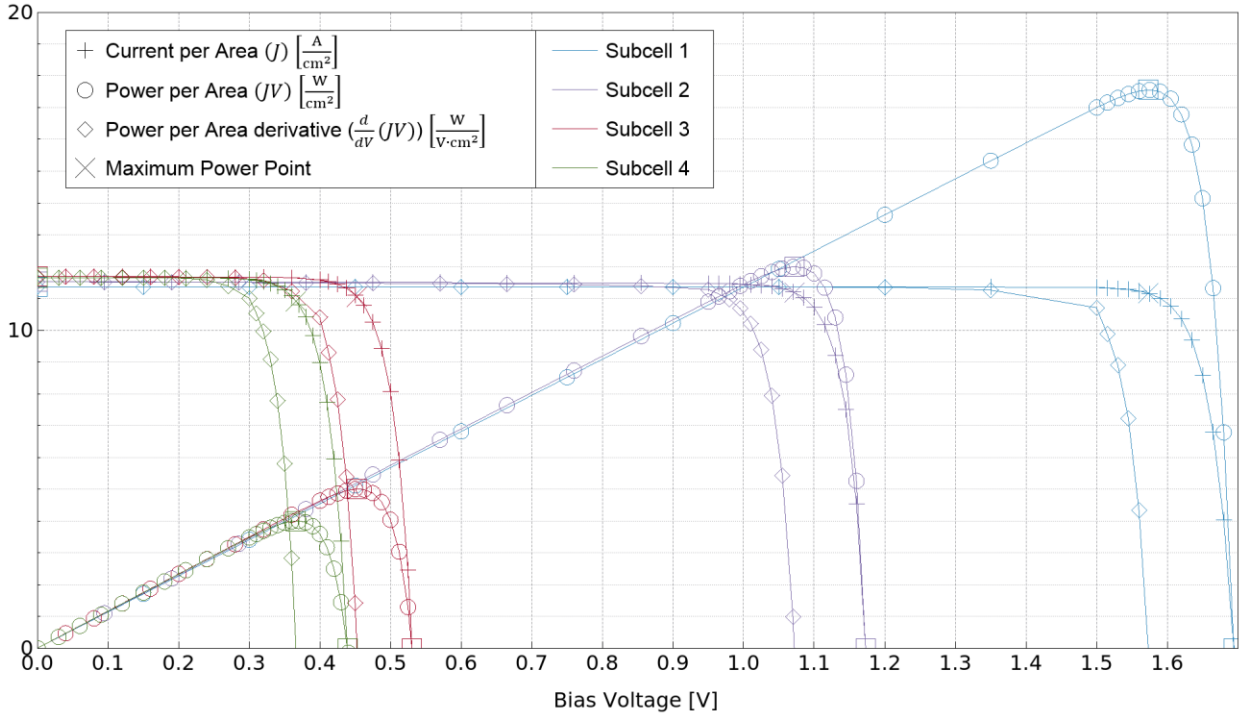

Figure 41: Subcell current and power density characteristics current-matched by layer thinning.

The triple junction does not require a SiGeSn/germanium tunnel junction, which has not yet been engineered. The tunnel used in the simulation is made of AlGaAs and GaAs, which is impractical because the Group IV growth chamber is not equipped to grow III-V materials, and vice-versa.

The layers are much thinner than required by the three-junction device. Current matching was achieved by thinning the GaInP, (In)GaAs, and SiGeSn subcells. However, the three-junction device did not involve the germanium substrate, which is an advantage. This affects the design's potential markets. Increasing the weight of the device increases the cost to include it in a space launch. The three-junction design considered previously uses layers totalling under 5 µm thick, while this design uses the 170 µm substrate. Additionally, a device design using this large amount of bulk germanium becomes highly dependent on the price of the material. It is likely possible to redesign this cell using direct compositions of SiGeSn for the bottom subcell instead of germanium. Additionally, these are likely to have better collection of the photons in the indirect range of germanium.

An atmospheric absorption band exists in terrestrial solar spectra in the range of 1800 to 1925 nm or 0.64 to 0.69 eV. [13] Therefore, 0.69 eV would be an appropriate target absorption step and direct bandgap value for SiGeSn. Response in the range of 0.64 to 0.69 eV is not useful and would unnecessarily decrease the voltage of the subcell.

Direct bandgaps in this range have been demonstrated of $Ge_{1-y}Sn_y$. The direct conduction band energy valley of germanium and of GeSn is brought down more significantly than the indirect bandgap under tensile strain. [68][69][40] In these contexts, tensile strain occurs when the crystalline arrangement is stretched in the two axes of the growth plane (biaxial strain). Strain refers to a difference in the geometry of crystalline matrix, and stress refers to the forces within

the structure. As the matrix is stretched in the two aforementioned dimensions, the third dimension, the growth dimension, experiences a reduction in length. Tensile strain of this sort brings the direct bandgap to a more attainable level, and the material becomes direct at an earlier point with respect to tin composition. For SiGeSn, the material can be called more direct, or the region where the material is "almost direct" [48] shifts with respect to composition under tensile strain. To achieve tensile strain, the material could be grown on a substrate of higher lattice constant. Note that tin content increases the natural lattice constant of the material, given that tin has larger lattice constant relative to germanium according to its atomic number. Growth of GeSn at the lattice constant of germanium substrate does not produce the necessary strain, it produces compressive strain.

To implement a device using strained SiGeSn or germanium, an option is to invert the growth direction, such that the III-V devices need not be grown on top of the strain-managed SiGeSn and germanium layers. This requires the ability to remove the inverted III-V growth from its substrate. Since the optimized four-junction device does not require extreme thicknesses and diffusion lengths in the III-V subcells, the original growth direction may also be possible and viable.

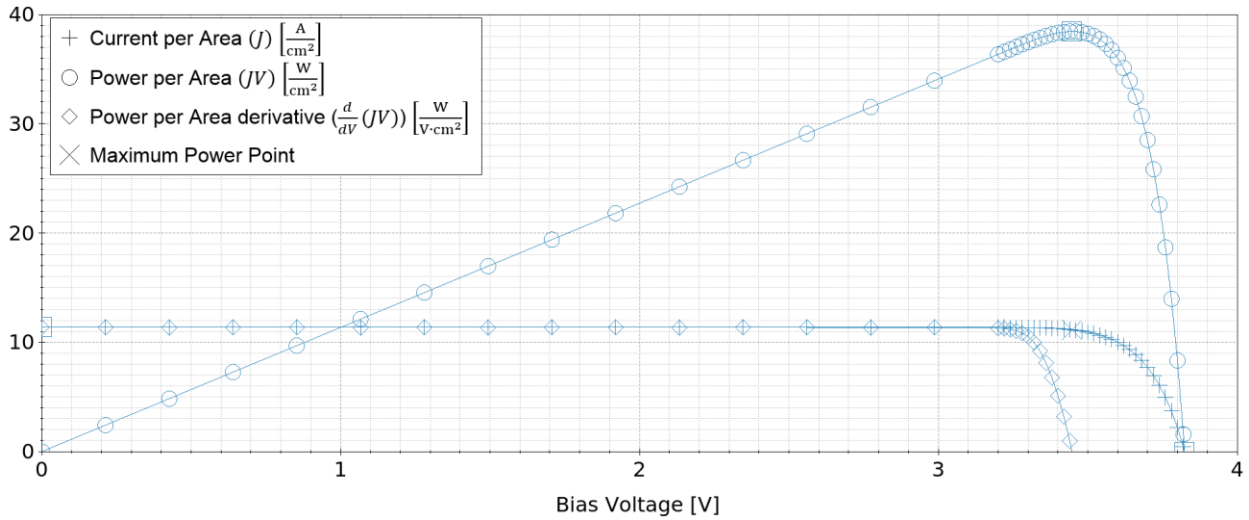

Figure 42: Current-matched four-junction device J-V characteristic.

The result of the full device simulation is that under 1000x concentration AM1.5D illumination, this device can produce 38.4 $\mathrm{W/cm^2}$, which is 42.7% conversion efficiency. This is higher than the performance of the triple junction considered above (41.4%). The growth layer thicknesses are much lower in this device, which makes the growth less challenging, however, it does require the use of the germanium substrate.

This is the only subproject in which there are no virtual contacts held at 0 V adjacent to the tunnel junctions. These components work by quantum mechanical tunneling. The layers are very thin, increasing the possibility of electrons to traverse the barrier. The electrons tunnel through the barrier between the conduction band of the n-side of the junction and the valence band of the p-side. The tunneling probability and current depends on the potential experienced by the electron or hole at distances beyond the location of the barrier. The quantum mechanical wavefunction of the particle can penetrate the thin barrier, and its magnitude on the opposite side of the barrier enables the non-zero tunneling probability. This is why Sentaurus Device, requires definition of a

"non-local" mesh. This is another mesh, that is independent of the original mesh. It includes the wavefunction of the electron at a locations other than the mesh point being solved. The tunneling model is called the non-local model. [58] [59]

The calculated EQE of the subcells of the device make the degree of thinning and photon sharing apparent in Figure 43. The 4J device design scenario would benefit from improved collection efficiency on the part of the bottom two subcells in the energy range below the (In)GaAs bandgap. This is addressed in the segmented design to follow by segmenting both of the front two subcells and only one of the bottom subcells, which rebalances the photocurrent such that less thinning is necessary. Other avenues for addressing the issue include improved light trapping in the bottom subcells, the utilization of direct compositions of SiGeSn in the bottom subcells instead for improved absorption, or increasing the bandgaps of the front subcells, such as by including aluminum in the front subcell alloy.

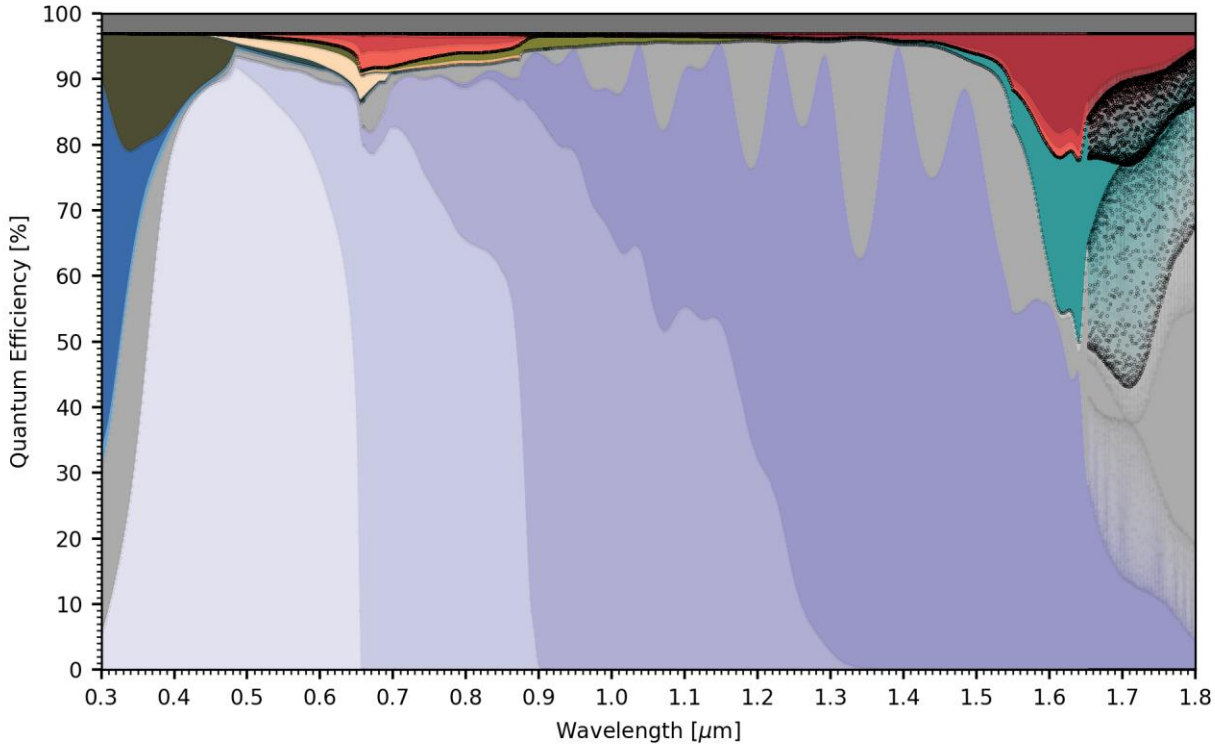

Figure 43: Calculated EQE of four-junction design with thinned front subcells and tunneling model disabled.

Figure 44 includes the standard solar spectrum and shows in comparison with Figure 43 that many of the reflectivity maxima occur in preferred locations, at the wavelengths where the atmosphere is absorbing. It also shows that the electron minority carrier loss at the back contact is a significant loss mechanism of the limiting back subcell.

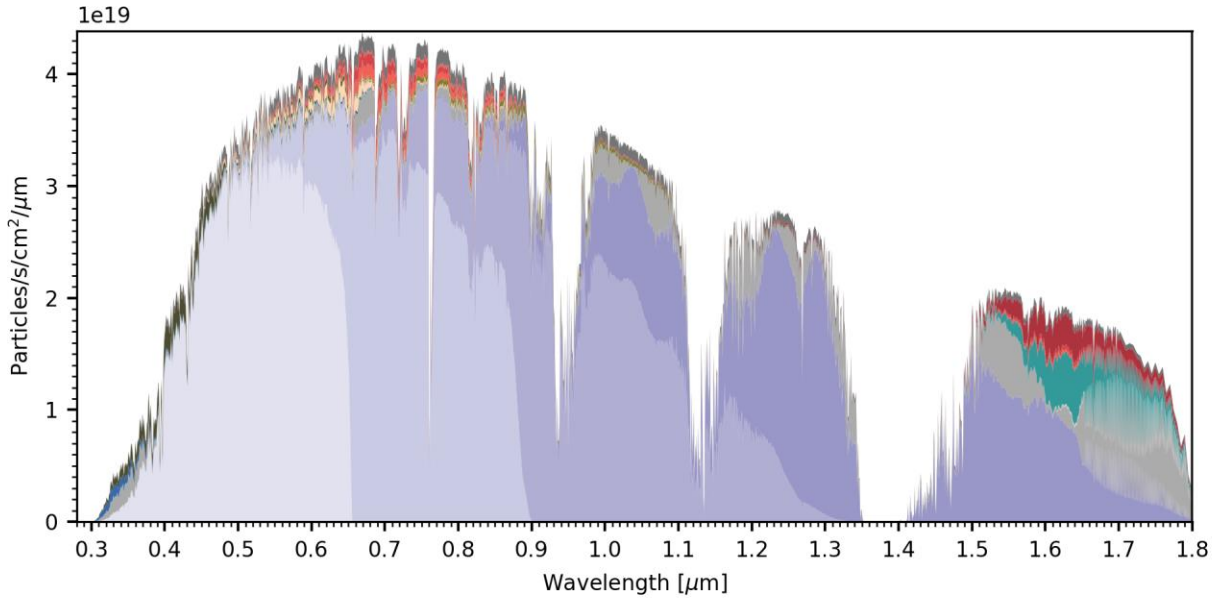

Figure 44: Four-*junction device EQE multiplied with the AM1.5D standard spectrum.*

Figure 45 can be used to determine the location of the recombination processes within the device. The optical generation is tallied by layer, and at the location that the optical generation is added, the recombination in that layer is overlaid. The plot versus device depth (Figure 39) also shows the locations of many physical processes. The SRH recombination occurs in the (In)GaAs subcell depletion region. Radiative recombination occurs throughout the (In)GaAs base. It also occurs in the GaInP base, however, that subcell is very thin in the four-junction design, so recombination in that base is less significant than in the previous design where only the front three subcells are current matched. Auger recombination occurs strongly in the germanium emitter, according to the plot versus device depth, Figure 39, however, the larger contribution is from the integration of the Auger recombination throughout the germanium base.

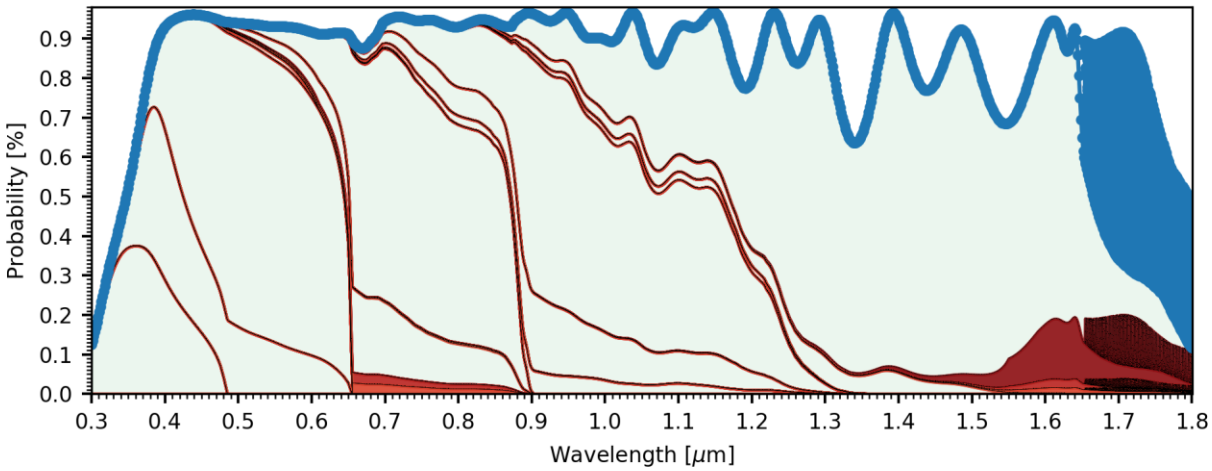

Figure 45: Recombination overlaid against optical generation by device layer.

These calculations are done with the tunneling physics model off at the tunnel junctions. Given that there are 0 V virtual contacts on both sides of the tunnel junctions in this subproject, the tunneling physics is not necessary, since the component is shorted. With tunneling physics on, the process counteracts the current loss appearing at the contacts. Therefore, data from the tunneling model would have to be added to the plot. Without this addition, the effect of the tunnel physics appears as a discrepancy in the EQE and IQE summations.

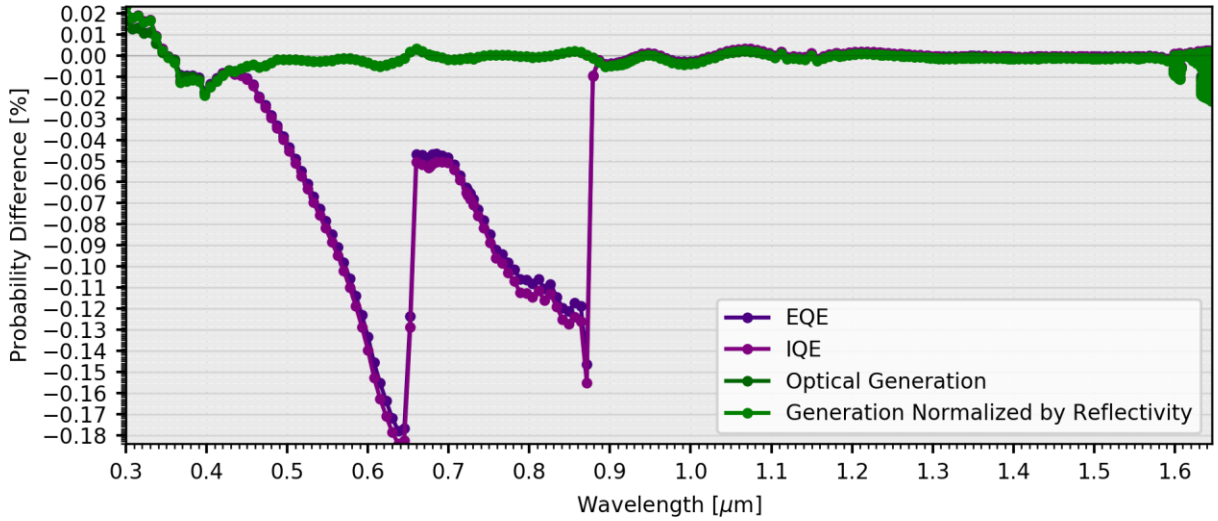

Figure 46: Discrepancy in four-junction device EQE with tunneling model enabled and not plotted.

The effect of the tunneling model increases as the wavelength increases until the bandgap of the absorbing tunnel layer is reached, and then there is a step as the effect from the second tunnel becomes significant. The third tunnel is completely transparent, and therefore a negligible amount of carriers are present there for all wavelengths, and the tunneling physics of that tunnel is insignificant in this configuration.

The overestimate of total optical generation at 0.66 μm occurs not from error introduced by the finite mesh, but due to a reflection producing additional generation in a non-transparent tunnel layer. This phenomenon is counted twice in the tally, because it contributes to the total reflection, as well as to additional generation in the tunnel layer. This is true of the IQE tally as well, since that is produced by dividing quantities by $1 - R$. Additionally, finite mesh related calculus error at the front face of a layer is maximized at photon energy slightly below the bandgap of a preceding subcell, where a relatively large proportion of the optical generation is sampled by fewer mesh points at the front of a layer. The error contribution of the first finite element is negative, and the contribution from all other elements is positive, except when the error from an interference pattern is more significant.

Figure 47 shows the discrepancy in the tally when the tunneling effect is not included. The error is then dominated by optical generation effects. Many of these errors could be reduced further by increasing the number of mesh elements, however, they are at an acceptably low level.

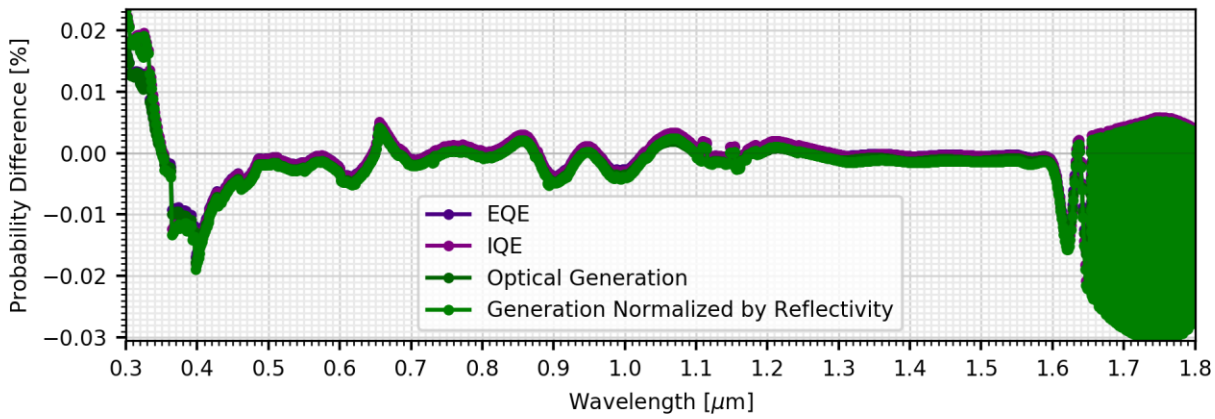

Figure 47: Discrepancy in four-junction device EQE with tunneling model disabled.

5-2-1: Germanium Subcell Back Surface Field

Based on these results, the germanium BSF design is revisited. A much stronger barrier to conduction band electron passage prevents them from reaching the back contact where there are more states available to them and can discharge their potential energy. This BSF is simply more highly doped germanium, but a higher bandgap material with the same valence band level to allow holes to pass could create a better barrier. However, surface recombination at the edge of the barrier can defeat the purpose if the material cannot create a high quality interface with the germanium.

Since the electron loss is not shown to be as significant in the third subcell, its back can be adjusted, to reduce the electrical barrier of the BSF while improving its transmission of photons to the fourth subcell. The thickness is reduced from 0.1 μm to 0.05 μm, the *p*-type doping is increased to $4*10^{18}$ cm^{-3}, and the base region is lengthened from 0.68 μm to 0.73 μm, improving absorption in the base, and reducing absorption in the BSF.

Increasing the BSF doping from 3e18 to 8e19 and the BSF layer thickness from 0.1 μm to 2 μm can improve the 4^{th} subcell short circuit current from 12.4 to 16.1 A/cm^2, however, this result is subject to error from the mirror reflection calculation. In the short circuit EQE plot, the subcell 4 electron loss current is almost completely removed. It is also necessary to ensure that the highly doped, thick BSF does not affect the subcell fill factor too strongly. Another consideration is that it should not be unnecessarily thick to maintain transparency to the reflection from the back mirror. Given that the shortage of photons is shared among the four subcells, the germanium subcell can be operated at a current above its own maximum power point, because it would require more photocurrent to be shared by the front subcells to bring the germanium subcell short-circuit current high enough for its maximum power point current to match the other subcells'. This BSF is not as strongly coupled to the rest of the device as some other components; Being at the back, its transparency only affects its own subcell, while changes to other BSFs affect the absorption requirement of the adjacent subcell.

Simulation trials are run over a limited wavelength range to determine the thickness and doping required of the BSF to prevent the electron minority carrier loss. A 1.5 μm $4*10^{19}$ cm^{-3} *p*-type BSF layer nearly completely prevents this energy loss process.

5-2-2: Germanium Subcell Front Surface Field

Improving the germanium subcell BSF increases the hole loss at the virtual contact on the other side of the subcell. The front surface field component to the germanium subcell can be improved.

The valence band model of SiGeSn indicates a decrease of the valence band level in SiGeSn lattice-matched to germanium. The effective valence band level is the highest available state, so valence band splitting increases the valence band level. The decrease of the valence band level of SiGeSn with respect to germanium is due in part to having less valence band splitting. This is compounded by the effect of silicon content to reduce the average valence band level independently of splitting. This is excerpted output of the SiGeSn bandgap and band offset model, quantifying these effects.

Table 7: $Si_{14}Ge_{82.2}Sn_{3.8}$ Valence Band Offset Calculation.

```
Ge_spin_orbit_splitting:      0.295 eV
SiGeSn_spin_orbit_splitting:  0.278 eV
deltaEv_SiGeSn_Ge_av_state:  -0.041 eV
deltaEv_SiGeSn_Ge_top_state: -0.046 eV
deltaEv_Ge_GaAs:              0.560 eV
deltaEv_SiGeSn_GaAs:          0.514 eV
SiGeSn bandgap:               0.748 eV
GaAs_bandgap:                 1.424 eV
deltaEg:                     -0.675 eV
absolute_cbo:                -0.161 eV
```

Figure 48 is an adaptation of figures from [46] to illustrate an overview the band offset trends of SiGeSn lattice-matched to $Ge_{1-x}Sn_x$ including pure germanium (x=0). It depicts the band offset of SiGeSn grown on compositions of $Ge_{1-x}Sn_x$, where x is the set of 0, 5, 8, 10, and 12%, and the SiGeSn is grown lattice matched to the GeSn. The x=0 data is added here. The orange lines show the silicon composition as a function of tin composition in the SiGeSn to lattice match to the GeSn, and has a slope near to the 3.67 ratio required to maintain lattice match. Generally, the valence band level decreases and the conduction band level increases at first, resulting in a type-I interface.

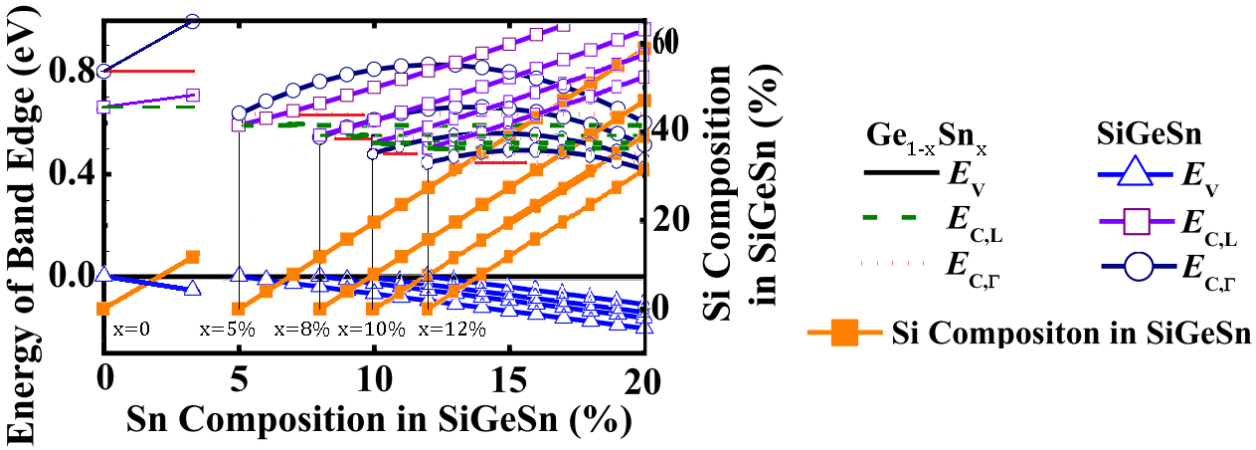

Figure 48: Energy band edge levels of SiGeSn and $Ge_{1-x}Sn_x$, lattice-matched, for x in 0, 5, 8, 10, and 12%. Merged data of figures from [46] *and added x=0.*

The decreasing valence band level is appropriate for the FSF layer against germanium, and the conduction band level can be reduced to equal the germanium emitter conduction band level with

doping. The energy levels produced by Sentaurus Device with the output of the above calculations present in the material parameter files, FSF doping set to $2 \cdot 10^{19}$ cm^{-3}, and germanium emitter doping set to $1 \cdot 10^{19}$ cm^{-3} are shown in Figure *49*. The holes are blocked at the FSF/emitter interface while the electrons are allowed to pass.

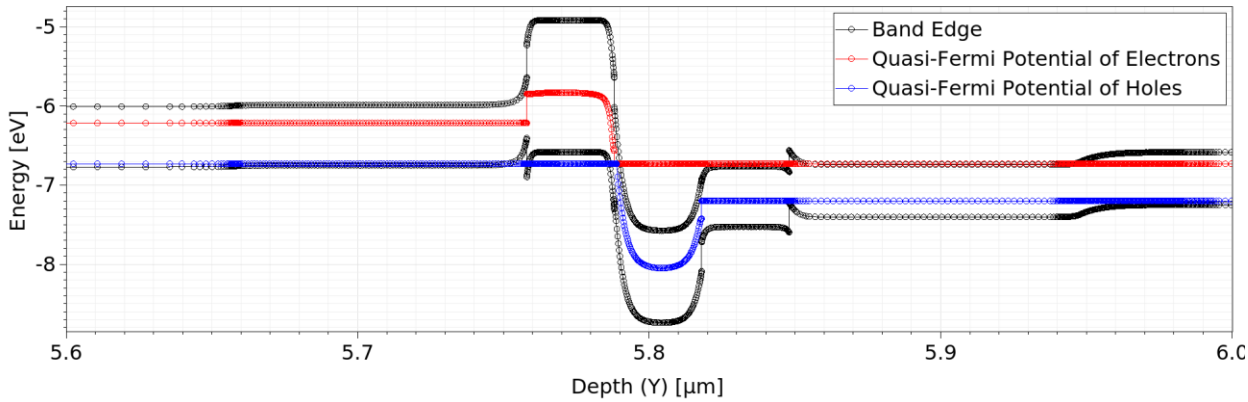

Figure 49: Band diagram of SiGeSn BSF, tunnel junction, SiGeSn FSF in germanium subcell, and germanium emitter.

5-2-3: SiGeSn SRH Lifetime Parameter Sweep

The time constant of SRH recombination in SiGeSn can be adjusted to simulate higher defect density and defect-related recombination. This can determine a range of validity of the presented designs, and determine a requirement for material quality. In high-quality germanium, the conduction electron lifetime is normally 1 ms and of holes is 0.1 ms. [34] Supposing potentially lower lifetimes in SiGeSn, the $J(V)$ curves show the modelled effect. Electron and hole lifetimes are set to equal to 100, 10, 1 µs, and 100 ns in the following simulation results, where the voltages of virtual electrodes adjacent to each subcell are ramped under 1000x AM1.5D illumination.

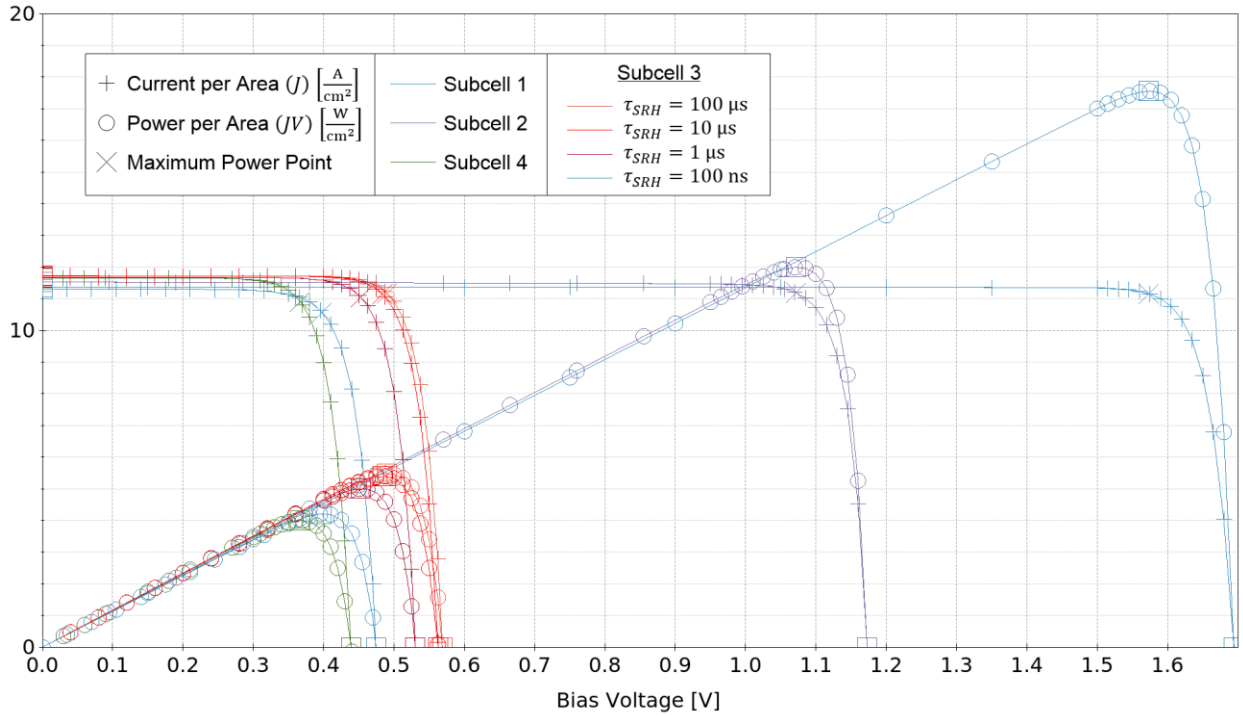

Figure 50: Calculated effect of SRH lifetime parameter on SiGeSn subcell current and power versus voltage curves.

At 100 ns lifetime, the maximum power point of the SiGeSn subcell is 1 W less. Even at that small level, the SiGeSn subcell appears to serve its purpose. The subcell voltage remains greater than that of germanium. Additionally, the effect of SRH recombination is more pronounced on the subcell output voltage, and less on the subcell current. Since the primary constraint on the design is that the subcells must match current, the required layer thicknesses are fairly independent of material quality in this range, and this design appears valid for a large window of possible defect densities.

The lifetime of the minority carriers can be related to its defect density N_T through the capture coefficient, B, for the recombination process, which, for SRH recombination, depends on the nature of the defects, via [70]

$$\tau_{SRH} = \frac{1}{BN_T}$$

5-3: Seven-Junction Design by 2:2:2:1 Segmentation

Many of the instructions in the command files throughout the project have been implemented in loops, such that the number of segments can be increased arbitrarily without requiring changes to these many areas of the code. A few global variables are used for this, `num_segments`, `epi_regions`, and `epi_inner_subcell_surfaces`. These variables control loops that create the virtual contacts, turn on the usual optical generation and recombination physics in the bulk, set up the tunneling physics as appropriate, and also control the loops in the visualization code. This arrangement allows the same code to be run on different Epi CSV input files without changes to the remainder of the project. The meshing code, for optimal performance, cannot be completely generalized. A

workbench variable `@epi_label@` is used to branch to different mesh code when different devices are simulated. The other reason for this project architecture is to enable the code to run with an arbitrary number of segments.

Given that there is a significant degree of thinning of the front two subcells in the four-junction design, the segmented device explored is segmented in a 2:2:2:1 pattern. This rebalances the current-collection requirements in the device, allowing the front III-V subcells to operate at maximum thickness and power, and budgeting additional photons for the newer group IV technology, to lighten some of the absorption requirements on the indirect materials.

The front segment in the pairs is very thin compared to the second segment. Higher-order segmentation is not explored here, which would have additional thin segments in the front. In the four-junction device, ~100 nm emitter layers are suitable for various base layers approaching 2 μm, however, in a thin segment, the emitter layer thicknesses must also be reduced. Many layers must be reduced below 30 nm, including the FSF, BSF, and tunnel layers. The layers of the first tunnel junction within the GaInP subcell have increased transparency requirements in the segmented device, and this tunnel has not yet been re-engineered for this increased requirement. Currently, containing GaInP of the same composition as the first subcell base, the first tunnel junction is showing significant parasitic absorption. The tunnel layers here are 22 nm, and lower thickness tunnel layers would be beneficial according to the simulation. Most appropriate would be a tunnel junction redesigned with higher bandgap materials. Additionally, a group IV tunnel junction is also required to realize this device.

Table 8: Seven-Junction Epi CSV

```
#_____________________________________________________________________________________
#|  LAYER    |           |              |           |           |           |        |
#|  REGION   | MATERIAL  |    FILE      | THICKNESS | DOPING    |   MOLE    | LAYER  |
#|  LABEL    |           |              |   (um)    | (/cm^3)   | FRACTION  | NUMBER |
#|___________|___________|______________|___________|___________|___________|________|
     arc1    ,    MgF    ,    MgF.par   ,   0.100   ,           ,           ,  ;#01  ,
     arc2    ,    TiOx   ,    TiOx.par  ,   0.060   ,           ,           ,  ;#02  ,
   sc1_fsf   ,    AlInP  ,   AlInP.tcl  ,   0.020   ,   6e18    ,   0.50    ,  ;#03  , window
   sc1_em    ,    GaInP  ,   GaInP.tcl  ,   0.025   ,   5e18    ,   0.50    ,  ;#04  ,
   sc1_base  ,    GaInP  ,   GaInP.tcl  ,   0.063   ,  -1e17    ,   0.50    ,  ;#05  ,
   sc1_bsf   ,  AlGaInP  , AlGaInP.tcl  ,   0.020   ,  -3e19    ,   0.25    ,  ;#06  ,
   td1_phigh ,   AlGaAs  ,  AlGaAs.tcl  ,   0.022   ,  -2e20    ,   0.30    ,  ;#07  ,
   td1_nhigh ,    GaInP  ,   GaInP.tcl  ,   0.022   ,   6e19    ,   0.50    ,  ;#08  ,
   sc2_fsf   ,    AlInP  ,   AlInP.tcl  ,   0.020   ,   6e18    ,   0.50    ,  ;#09  , 2nd AlInP window
   sc2_em    ,    GaInP  ,   GaInP.tcl  ,   0.10    ,   5e18    ,   0.50    ,  ;#10  ,
   sc2_base  ,    GaInP  ,   GaInP.tcl  ,   1.90    ,  -1e17    ,   0.50    ,  ;#11  ,
   sc2_bsf   ,  AlGaInP  , AlGaInP.tcl  ,   0.030   ,  -3e19    ,   0.25    ,  ;#12  ,
   td2_phigh ,   AlGaAs  ,  AlGaAs.tcl  ,   0.030   ,  -2e20    ,   0.30    ,  ;#13  ,
   td2_nhigh ,    GaInP  ,   GaInP.tcl  ,   0.030   ,   6e19    ,   0.50    ,  ;#14  ,
   sc3_fsf   ,   AlGaAs  ,  AlGaAs.tcl  ,   0.020   ,   1e19    ,   0.80    ,  ;#15  ,
   sc3_em    ,   InGaAs  ,  InGaAs.tcl  ,   0.040   ,   3e18    , 0.01173   ,  ;#16  ,
   sc3_base  ,   InGaAs  ,  InGaAs.tcl  ,   0.269   ,  -1e17    , 0.01173   ,  ;#17  ,
   sc3_bsf   ,   AlGaAs  ,  AlGaAs.tcl  ,   0.020   ,  -1e19    ,   0.20    ,  ;#18  ,
   td3_phigh ,   AlGaAs  ,  AlGaAs.tcl  ,   0.030   ,  -2e20    ,   0.30    ,  ;#19  ,
   td3_nhigh ,    GaInP  ,   GaInP.tcl  ,   0.030   ,   6e19    ,   0.50    ,  ;#20  ,
   sc4_fsf   ,   AlGaAs  ,  AlGaAs.tcl  ,   0.020   ,   1e19    ,   0.80    ,  ;#21  ,
   sc4_em    ,   InGaAs  ,  InGaAs.tcl  ,   0.10    ,   3e18    , 0.01173   ,  ;#22  ,
   sc4_base  ,   InGaAs  ,  InGaAs.tcl  ,   1.05    ,  -1e17    , 0.01173   ,  ;#23  ,
   sc4_bsf   ,   AlGaAs  ,  AlGaAs.tcl  ,   0.030   ,  -1e19    ,   0.20    ,  ;#24  ,
   td4_phigh ,   AlGaAs  ,  AlGaAs.tcl  ,   0.030   ,  -1.2e20  ,   0.30    ,  ;#25  ,
   td4_nhigh ,     GaAs  ,    GaAs.tcl  ,   0.030   ,   6e19    ,           ,  ;#26  ,
   sc5_fsf   ,   SiGeSn  ,  SiGeSn.tcl  ,   0.015   ,   2e19    ,           ,  ;#27  ,
   sc5_em    ,   SiGeSn  ,  SiGeSn.tcl  ,   0.040   ,   1e19    ,           ,  ;#28  ,
```

```
sc5_base  ,   SiGeSn  , SiGeSn.tcl ,   0.30    ,  -1e18   ,          ,  ;#29  ,
sc5_bsf   ,   SiGeSn  , SiGeSn.tcl ,   0.020   ,  -2e18   ,          ,  ;#30  ,
td5_phigh ,   AlGaAs  , AlGaAs.tcl ,   0.030   ,  -1.2e20 ,   0.30   ,  ;#31  ,
td5_nhigh ,     GaAs  ,   GaAs.tcl ,   0.030   ,   6e19   ,          ,  ;#32  ,
sc6_fsf   ,   SiGeSn  , SiGeSn.tcl ,   0.020   ,   2e19   ,          ,  ;#33  ,
sc6_em    ,   SiGeSn  , SiGeSn.tcl ,   0.10    ,   1e19   ,          ,  ;#34  ,
sc6_base  ,   SiGeSn  , SiGeSn.tcl ,   1.00    ,  -1e18   ,          ,  ;#35  ,
sc6_bsf   ,   SiGeSn  , SiGeSn.tcl ,   0.100   ,  -2e18   ,          ,  ;#36  ,
td6_phigh ,   AlGaAs  , AlGaAs.tcl ,   0.030   ,  -1.2e20 ,   0.30   ,  ;#37  ,
td6_nhigh ,     GaAs  ,   GaAs.tcl ,   0.030   ,   6e19   ,          ,  ;#38  ,
sc7_fsf   ,   SiGeSn  , SiGeSn.tcl ,   0.030   ,   2e19   ,          ,  ;#39  ,
sc7_em    , Germanium ,     Ge.tcl ,   0.10    ,   1e19   ,          ,  ;#40  ,
sc7_base  , Germanium ,     Ge.tcl ,   10.0    ,  -7e17   ,          ,  ;#41  ,
sc7_bsf   , Germanium ,     Ge.tcl ,    2.0    ,  -8e19   ,          ,  ;#42  ,
```

The subcell current-voltage characteristics are produced by setting 1000 times concentrated AM1.5D solar illumination to be incident, using the virtual contacts adjacent to each subcell to bias each one individually, and solving for the contact currents under each set of boundary conditions. Figure 51 shows the bottom subcell is receiving more photocurrent than the other subcells. This current can be made available to the SiGeSn subcells by thickening them if necessary. Alternatively, the thickness of the germanium subcell can be reduced. To increase the operating current, a lower bandgap configuration of GaInP can be used. In this result, it is near the maximum available bandgap of 1.9 eV. The bandgap is adjustable with growth techniques that alter the randomness in the ordering of the elements in the crystal. In this design, the maximum available value is used, maximizing the voltage of the GaInP subcells, while also maximizing the photon sharing. The additional photocurrent available to the bottom subcells can be used to address challenging aspects of the material growth or other differences from the simulated output.

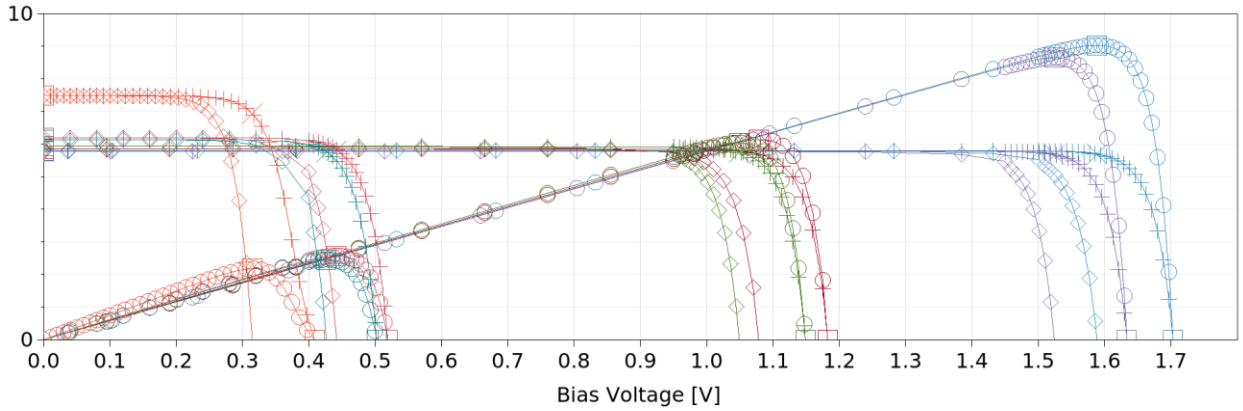

Figure 51: Current density [A/cm^2], power density [W/cm^2], and power density derivative [W·cm^{-2}·V^{-1}] characteristics with respect to controlled voltage ramping of virtual contacts across the seven subcells under 1000x the AM1.5D spectrum with one-dimensional mesh.

This design incorporates thinning in the (In)GaAs subcell. The way it is currently modeled, there is a clear path for improvement. The GaInP bandgap can be decreased, improving the absorption of the front subcell, sharing less photocurrent, and increasing the device's operating current. The (In)GaAs thickness can be increased in this case. This scenario would require relatively strong performance from the bottom three subcells, but is much more attainable than the 2:2:2:2 configuration that would evenly divide all available photons according to the material bandgaps. This design budgets additional photons for the SiGeSn subcells until their performance has been

shown to be sufficiently optimal. The approach is apparent from the depicted subcell EQE, which shows some effects of the photon sharing by thinning.

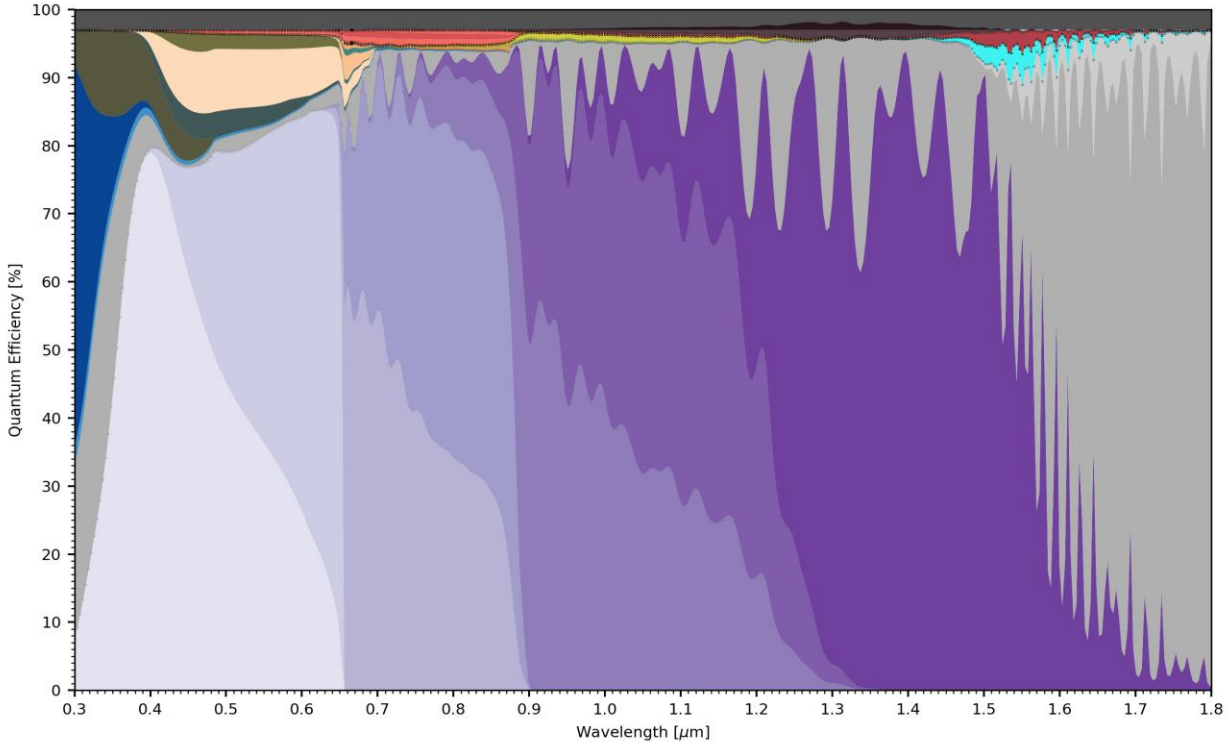

Figure 52: Seven-junction device EQE.

The full device simulation of this design shows output power of 37.4 W/cm^2 under 1000x illumination, and an efficiency of 41.5%. This is slightly less than the four-junction result, the bulk of which can be reclaimed or surpassed with thinner or higher bandgap tunnel junction layers. The short-circuit current is 5.75 A/cm^2, the open-circuit voltage is 7.150 V, and the fill factor is 89.8%.

The error is related to surface recombination at the germanium FSF/emitter interface. The surface recombination velocity is set to 10^3 cm/s at all FSF/emitter and BSF/base interfaces, and the germanium subcell is showing the most surface recombination after increasing the doping and thickness of the germanium BSF. The surface recombination model appears to create error in the result. The error is reduced by reducing the mesh element size adjacent to the surface but not consistently and cannot be completely removed. The error could be introduced by the approximation of the finite elements adjacent to the singular vertex containing the surface recombination velocity, but spot-surface recombination also introduces gradients and error in the electron and hole current outputs integrated within the device. The automatic grid adaptation feature is an option available to rule out mesh approximation error caused by the surface recombination, however, the error was persistent in manual grid-adaptation trials of a one-junction device using high y-dimension mesh densities.

To confirm the validity of the post-processing calculations, the summation can be done differently. Majority and minority carriers are created in pairs in the optical generation process, and lead to current at the terminals unless they recombine. The summation of one kind of carrier, electrons, plus the total of all recombination processes, should equal the incident number of photons that

interact within the device. This leads to a similar error seen in the EQE error calculation. This summation can also be done with the hole output currents, since the sum of hole currents equals the sum of electron currents.

In Figure 53, the electron currents from each terminal area plotted in a cumulative manner, and the hole currents are also plotted in a cumulative manner, each reaching the same total. To that total, all sources of recombination are added. The incident number of photons is plotted, which is normally the divisor in the quantum efficiency calculations, and the loss effects from the shade of the front contact, the reflectivity, and the transmissivity are subtracted. This other method of tallying effects rules out unit conversion and algebraic error, localizing it to calculations within Sentaurus Device, such as TMM, or mesh-related integration error, or the residuals in the numerical methods. The error is mainly from surface recombination at the front of the germanium subcell, and is an overshoot of 1.2% EQE.

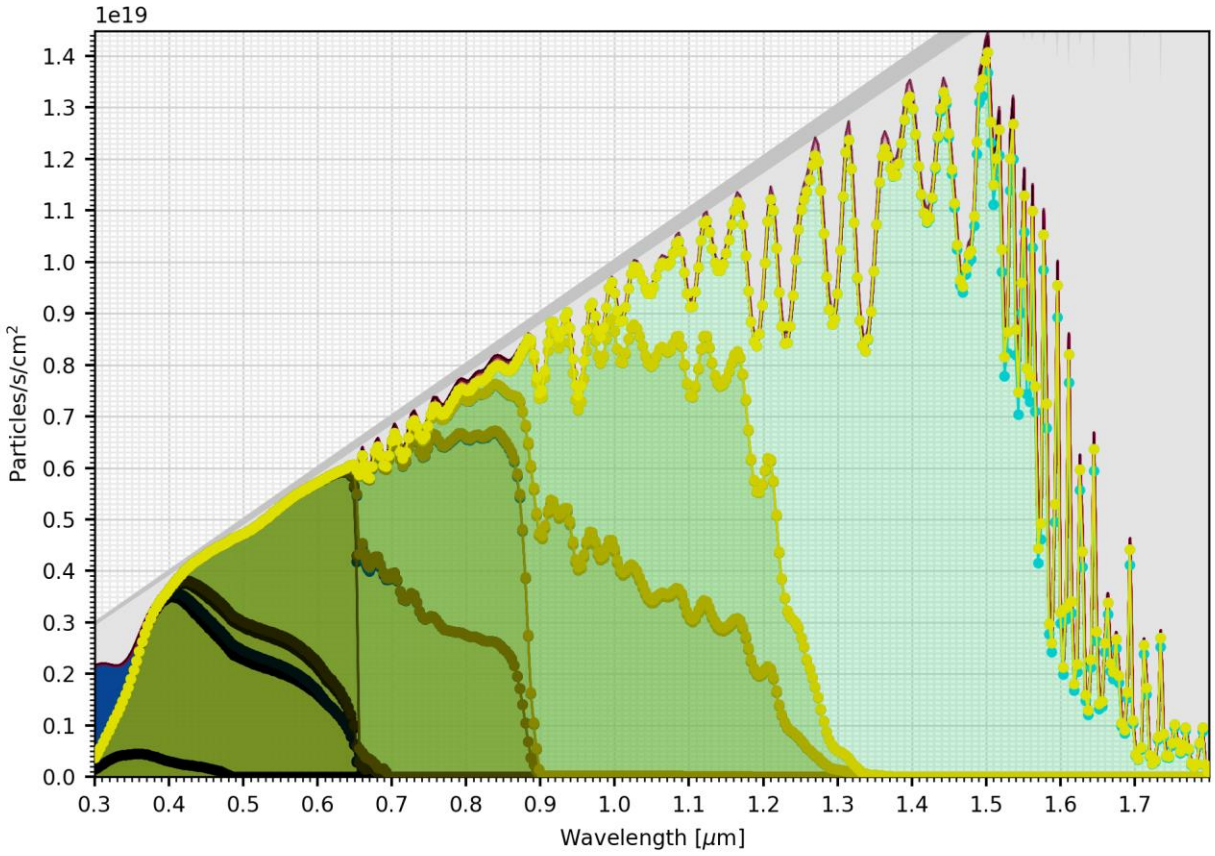

Figure 53: Tally of carriers reaching contacts, recombining, with non-generative absorption.

The simulation with two-dimensional mesh is possible in the segmented design due to the reduced germanium layer thickness. The following mesh produces a result with less error from surface recombination. It contains 10^6 mesh vertices, `ErrRef(electron)` and `ErrRef(hole)` are set to 10^9, `Digits` and `TransientDigits` are set to 13, `RhsMax` is 10^8, `RhsFactor` is 10^5, `RhsMin` is 1, Sentaurus Device version *P*, and the surface recombination error is contained. These tolerances and the number of mesh points can be walked back to observe whether similar error performance can be produced more quickly. Apparently, the manner in which the surface recombination parameter file

is written is most compatible with Sentaurus Device version *P*, and using this version solves the sort of error visible in Figure 52.

```
proc make_multibox_with_ratio_lists {window maxX maxY minX minY {Xratios {}} {Yratios {}}} {
    foreach xr $Xratios {
        make_multibox $window $maxX $maxY $minX $minY $xr 0
    }
    foreach yr $Yratios {
        make_multibox $window $maxX $maxY $minX $minY 0 $yr
    }
}
proc full_x_multibox_for_shadow {y0 y1 xmaxel xminel {R 1.2}} {
    make_multibox_with_ratio_lists [make_window "Xmin" $y0 "(+ Xmin half_w_contact)" $y1] $xmaxel "(- $y1
$y0)" $xminel "(- $y1 $y0)" [list $R -$R] {}
    make_multibox_with_ratio_lists [make_window "(+ Xmin half_w_contact)" $y0 "Xmax" $y1] $xmaxel "(- $y1
$y0)" $xminel "(- $y1 $y0)" [list $R -$R] {}
}
proc concise_full_wrap {} {
    global epi_regions
    foreach region $epi_regions {
        y_region_multibox $region 0.001 {1.1 -1.1}
    }
    make_multibox_with_ratio_lists [make_window "Xmin" "(- Y0_sc1_fsf h_cap)" "(+ Xmin half_w_contact)"
"Y0_sc1_fsf"] 1 1 0.001 0.001 {1.2 -1.2} {1.2 -1.2}
    full_x_multibox_for_shadow "Y0_sc1_fsf" "Y1_sc1_em" 1 0.001 1.2
    full_x_multibox_for_shadow "Ytop" "Ybot" 10 0.001 1.2
}
```

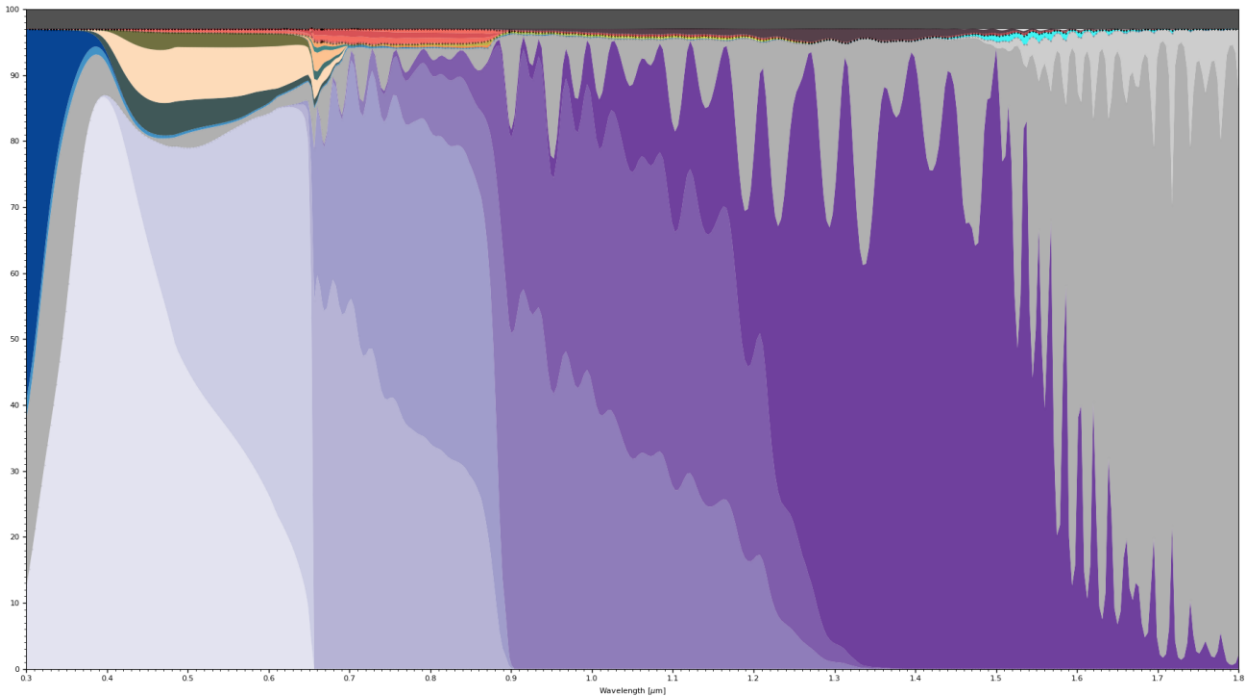

Figure 54: EQE result from two-dimensional mesh, surface recombination coefficient set to $5*10^4$ cm/s, surface recombination model error acceptable from Sentaurus Device version P.

The result shows that there is much less hole loss from the first subcell in the two-dimensional simulation. The hole loss found previously is an artifact of the one-dimensional structure configuration. The two-dimensional mesh produces the subcell *J-V* characteristics shown in . Some electrostatic potential is lost as carriers move across, especially in the front segment. This produces

a reduced fill factor, but current is improved with the removal of the virtual contact across the front window, showing that the front segment could be thinned slightly.

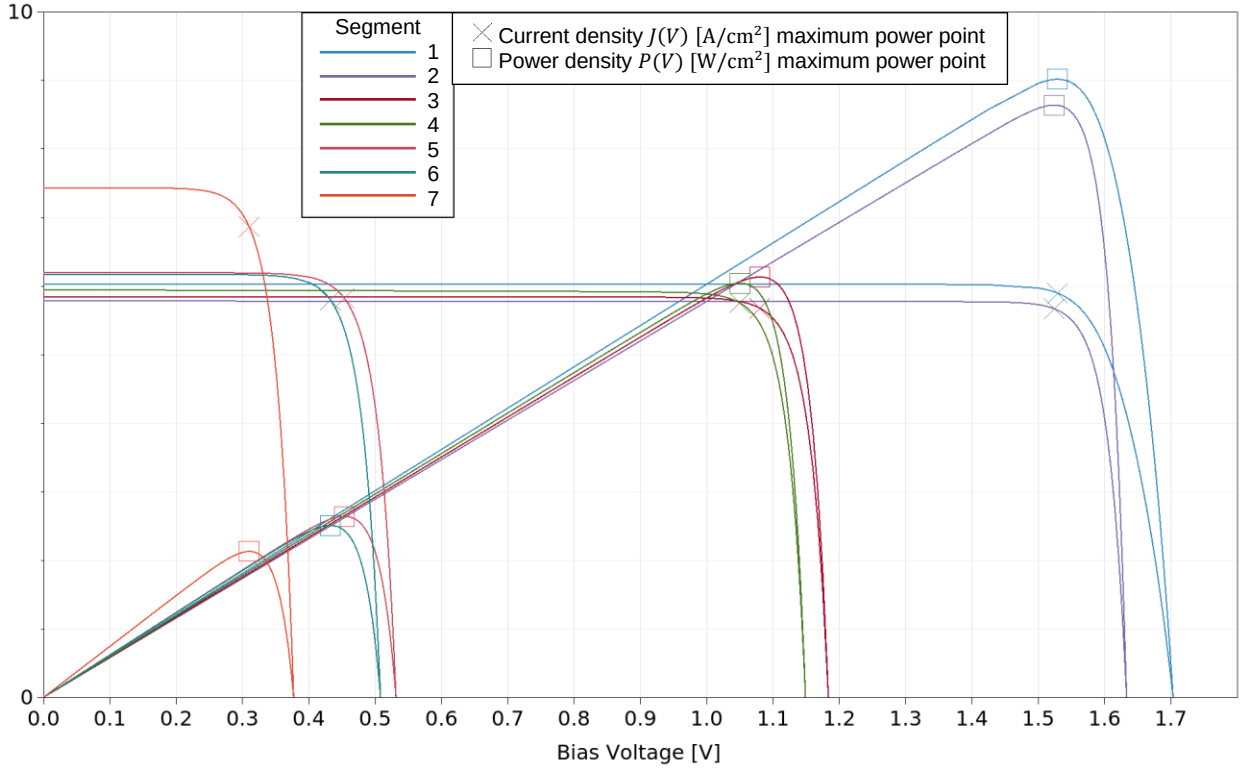

*Figure 55: Subcell J(V), P(V) under 1000x AM1.5D with two-dimensional mesh. Surface recombination velocity is set to 5*10^4 cm/s.*

The following figures show the total, electron, and hole current densities within the device, at short circuit voltage, 1000 times AM1.5D illumination, and 300 Kelvin. This solution can be useful for mesh adaptation. This mesh likely has too many elements at the bottom of base layers, and does not exploit symmetry, so there is room for improvement in computation time, though this mesh was effective in controlling the surface recombination error. The color scale is chosen such that the profiles are visible in the larger areas of the device. For electrons, this emphasizes their role as minority carriers in the bases, and the electron current in the emitter layers is generally above the scale limit. In the electron current density figure, the BSF effect to block the electrons can be seen, while holes are allowed to pass.

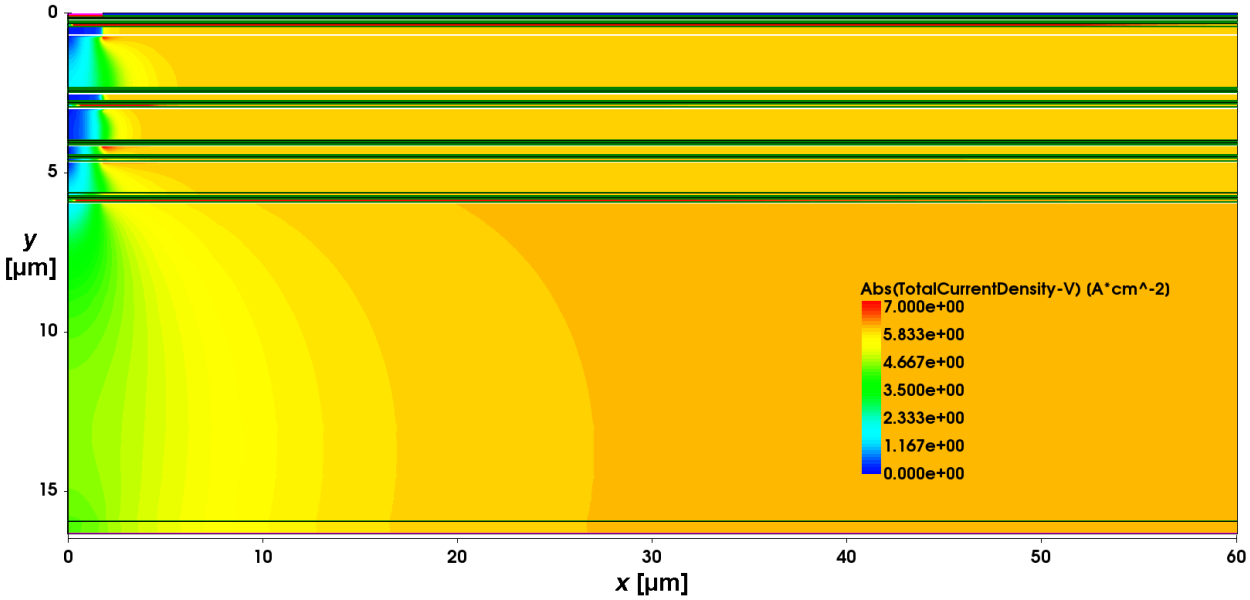

Figure 56: Current density magnitude.

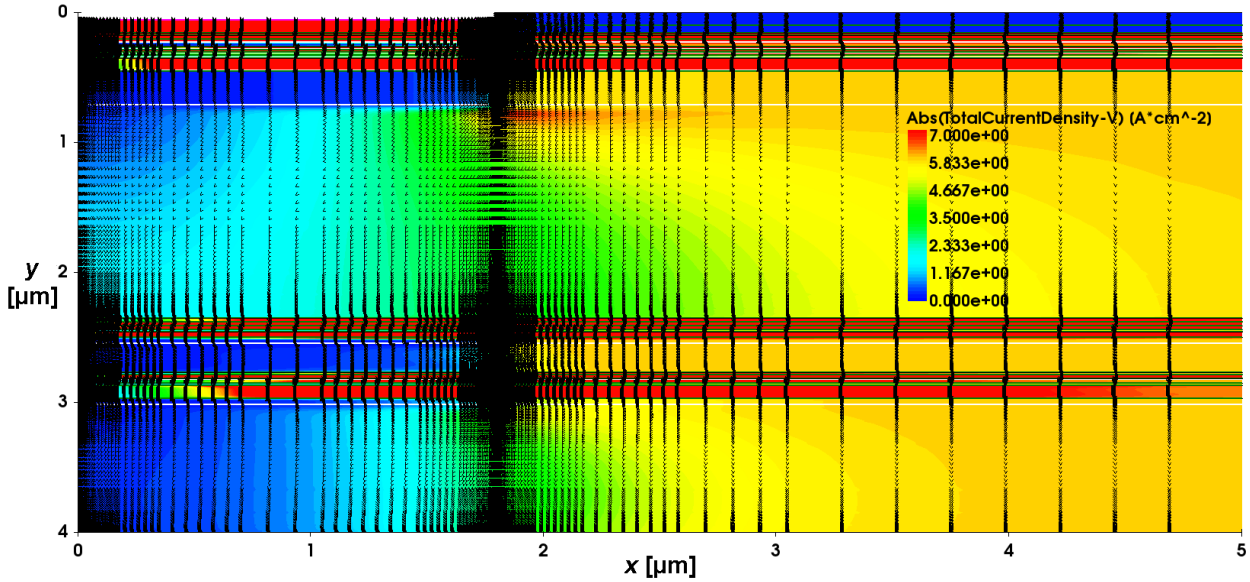

Figure 57: Current density magnitude and direction. Arrow-head points are located on mesh vertices.

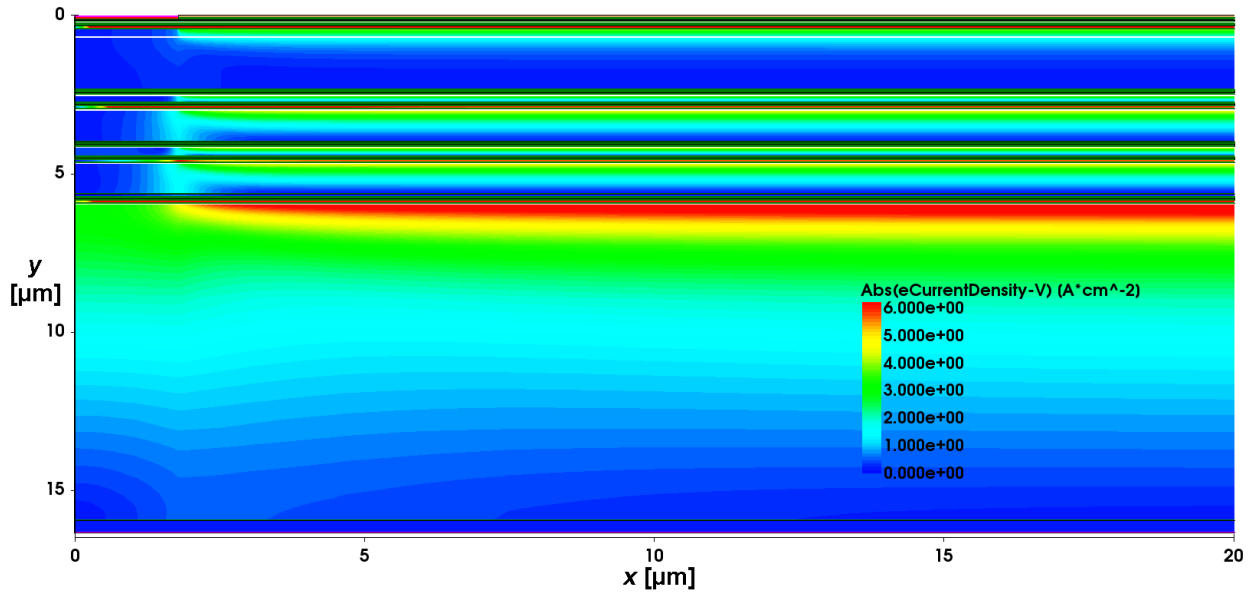

Figure 58: Electron current density.

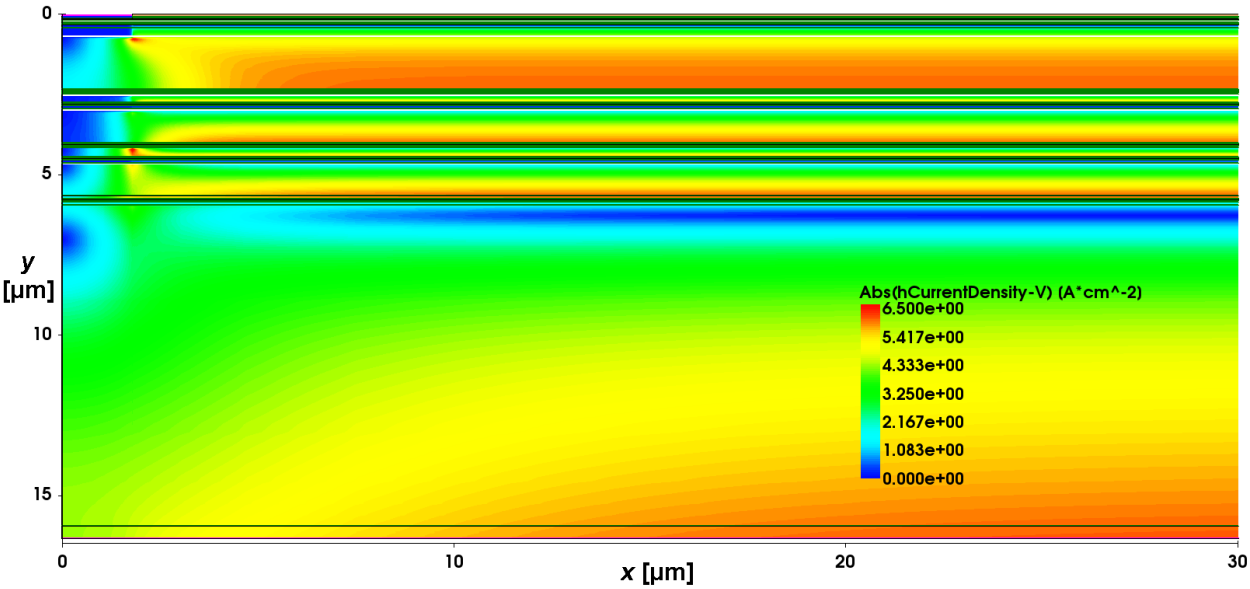

Figure 59: Hole current density.

5-4: Optimization

5-4-1: Procedure

After controlling sources of error related to the mesh and surface recombination model, a series of devices can be optimized. The previous section optimized devices using the one-dimensional mesh. Upon simulating the device with a two-dimensional mesh, the main observable differences are a) that the fill factor of the first segment suffers, and only that segment, and b) there is much more hole loss at the cathode in the one-dimensional structure. Much of the optimization work can be done with the one-dimensional mesh. This is necessary to keep execution time reasonable. If the simulation can run in a few minutes, then the process can be iterated. Two-dimensional iterations can be expected to take several hours, however, some such iterations are necessary. The

current-matching must be rebalanced with the two-dimensional structure because of the different hole loss at the cathode compared to the one-dimensional structure.

Additional hole loss occurs at the cathode in the one-dimensional mesh, because in that configuration, the small cap and contact are not made with the appropriate width in the appropriate location, as in Figure 4. Instead, the contact is made to virtually exist across the entire front face of the device, between the ARC and the FSF. Since contacts in the simulation simply provide a boundary condition on the structure, and the size and location of the metal is not input into the structure editor, they are completely transparent to the incident light. To allow for fast solution with 1-dimensional mesh, the virtual contact is placed in this manner. This alternative, wide location of the front contact results in more minority carriers being collected at the wrong contact, counteracting the desired collection effect of the device. This loss mechanism can be seen in the EQE results, such as Figure 26, and reduces the power output of the device. The difference is also observable in , where the device current-matched with one-dimensional mesh is simulated with two-dimensional mesh, and the $J_s$of the first segment is then higher than the others. This is a source of error present in the faster-executing one-dimensional arrangement.

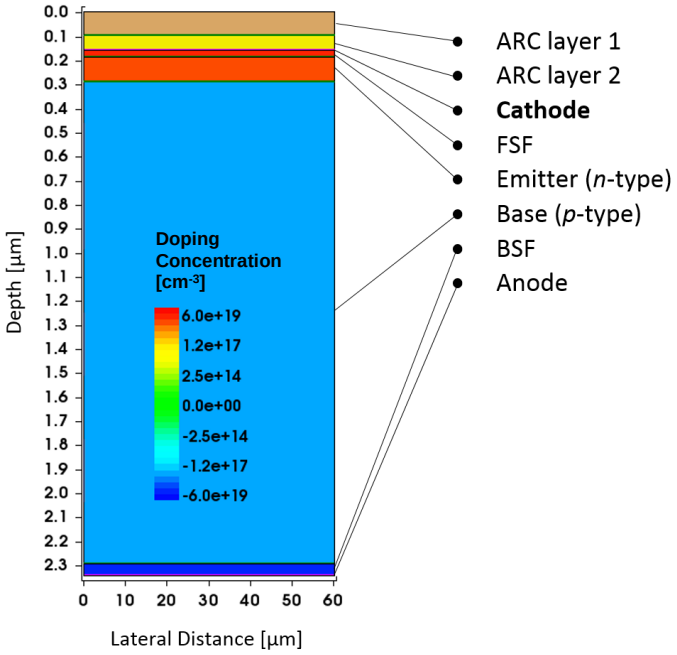

Figure 60: Cathode location in one-dimensional structure, collecting more minority carriers.

Another principal difference between the one- and two- dimensional meshes is that energy loss in the lateral movement of carriers is only calculated in the two-dimensional configuration. also the loss of fill factor that occurs in the front segment. By inspection of the current density magnitude plot in the two-dimensional solution in Figure 61, it is apparent that most of the lateral current is passing through the front emitter layer. By running trials on the emitter layer thickness, improvements in the front segment fill factor are observable when the thickness of this layer is increased. However, changing only the emitter thickness upsets any current matching that has been done. It is possible to optimize the emitter thickness of the front segment by coupling it with the

base thickness, trading material from one layer to the other, such that total absorbing distance remains constant.

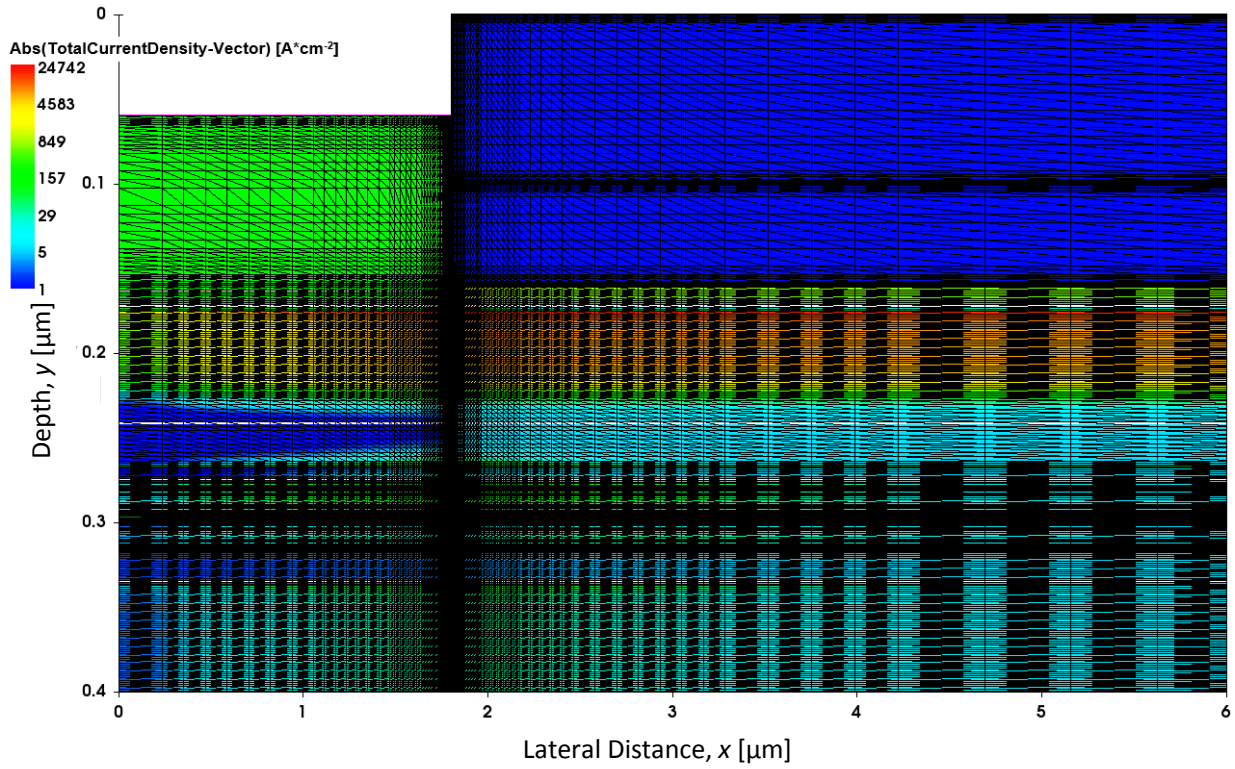

Figure 61: Current density at the front of the seven-junction device.

Most FSF and BSF layers can be optimized with the one-dimensional mesh, showing negligible difference in the one- and two- dimensional structures. According to EQE plots, these components are capable of completely blocking minority carriers. A very highly doped FSF or BSF can nearly completely eliminate this loss mechanism by creating a strong barrier to the minority carriers in the way of the contact where they would otherwise connect to the circuit and recombine. Instead, they are held away by the potential energy barrier of the surface field layer and only allowed to travel the proper direction. For this purpose, the EQE loss plot has been implemented in Sentaurus Visual, which can be run in interactive mode from the Sentaurus Workbench, allowing the plot to be created rapidly after the completion of a wavelength ramp. This arrangement allows experimental trials to be run on the doping and thickness of an FSF or BSF.

The four-junction device employs a simulation of a germanium substrate layer to be actively included as part of the device. This layer has 170 µm of depth, while all of the other layers are less than 2 µm. It has been observed that this affects the requirements of the BSF layer. If the BSF is too thin and has too low of doping, the minority carriers (electrons) can reach the anode at the bottom. They can drain there as if a swimming pool had a wall that was too low. With no BSF, the electrons are free to go in either direction. The field of the *p*-*n* junction does not penetrate far into the large-scale substrate. Therefore, the mechanism of collection of carriers is by diffusion. The BSF must be strong enough to hold away the larger volume of minority carriers present in the substrate-style device as compared to a thinner grown portion of the device. By observation of the

1-dimensional simulation results, a BSF with thickness of 1.0-1.5 um and doping of 3-8 * 10^19 cm^{-3} is capable of preventing most minority carrier electrons from recombining at the anode.

The other extreme situation for a back surface field is in the front segment of the seven-junction device. This tiny subcell is only 128 nm thick, while the second subcell of the same material is 2.05 um. This part of the device is a manifestation of the segmented design idea, where a subcell is partitioned, increasing the number of *p-n* junctions in the device. A very thin subcell is placed in front, and another larger subcell is placed behind it, following a calculation of the absorption properties of the material, such that both *p-n* junctions will separate a similar number of electron-hole pairs. These two components are series-connected with a tunnel junction component. The BSF in this position has much more need to be transparent than the BSF discussed above. Fortunately, the BSF in this position is made of AlGaInP, which has a very high bandgap. The AlGaInP BSF was designed originally for the InGaP/InGaAs/Ge device. AlGaInP has higher bandgap and is more transparent than InGaP. However, in the case of the very thin front segment, most of the spectrum has not yet been absorbed, and even the AlGaInP is not totally transparent, and has some undesirable absorption. This is apparent in the minority carrier loss plotted in the EQE. The optimization procedure of this BSF has shown that it should be extremely thin. In the simulation, 10 μm is chosen as what may be the thinnest possible to achieve. The increased bandgap of AlGaInP does make it effective at blocking minority carriers, so it should remain present, however, this is an important location of the device, where parasitic absorption is detrimental to the remainder of the device, since those photons are needed by the subsequent components.

The FSFs have transparency requirements similar to the BSF of the first segment. The equations of the simulation calculate the strength of the of the barrier compared to the diffusion characteristics based on the concentration of carriers in the device. This is used to find the thickness and doping necessary, while also in consideration of the absorption characteristics of the layer. These optimizations of doping and thickness of the BSFs and FSFs can be done by trial and error with the one-dimensional mesh.

In summary, the procedure for optimization is roughly as follows.

Preliminary current matching in 1D with subcell $J(V)$ curves

- Let first segment have slightly lower J_{sc}, since there is an incorrect source of error introduced by the contact in 1D

Optimize BSF and FSF doping levels and thicknesses with EQE of 1D mesh

- Except front window and emitter layer are dependant on 2D effects

With 2D mesh,

- Expand segment 1 emitter to help fill factor, likewise reducing base thickness to maintain current match
- Final current-matching with proper 1st segment behavior

The initial current-matching task is primarily done by changing only the base thickness. Generally, a 0.1 μm emitter thickness is assumed, and the base is on the order of 1 μm. However, when segmentation is applied, base thicknesses are then smaller, on the order of 0.3 μm, so the emitter should be thinned, such that the *p-n* junction is placed centrally in the optical generation profile.

The findings of the optimization procedure are observable in the thicknesses appearing in the seven-junction Epi table, *Table 11*. The front segment has an emitter of 40 nm and base of 52 nm. Typically, in a solar cell, the ratio of the emitter to the base is much smaller, and the *p*-*n* junction is much more asymmetric. However, this emitter is nearly the thickness of its adjacent base, since its thickness has been increased to help maintain the fill factor of the front segment. In this way, the output of the calculation can be taken as more informative than certain rules of thumb of solar cell design: It has shown that the front segment of the seven-junction device is an exception to the asymmetry rule, because of the lateral current losses within in the device.

5-4-2: General Mesh for Two-Dimensional Structure

The mesh function `2d_mesh_many_devices` in Appendix L defines the mesh in a way that can be used for every device. The small number of differences in mesh required by the different devices are encoded with `if` statements. The triple-junction device requires a smaller growth ratio in the third subcell, because the mirror is more significant.

These mesh functions guide the Sentaurus Mesh tool by indicating to the tool constraints on the mesh element sizing. The tool also implements additional constraints, such as Delaunization, necessary for efficient and effective solution of the device. This mesh constraint definition is also called mesh refinement, as the TCL functions employed here print Scheme mesh refinement functions that the Sentaurus Structure Editor interprets. The mesh refinement is a necessary part of the mesh definition, since without this expected guidance, the mesh tool defaults to large element sizes unsuitable for resolving the physical processes in the device.

First, the `arc_region_quintuple_y_refdef` function sets up three mesh spacing values within the ARC layers. The five sections per layer include smaller spacing near the edges and larger spacing in the middle. This provides a controlled balance between discretization error suppression and the amount of computation that would be added by additional mesh in the middle of the ARC layers. Often, controlling the mesh explicitly in this manner with normal refinement boxes is preferable than using a multibox, because a multibox can create mesh spacing steps in unpredictable locations.

The `y_region_refdef` function sets up a consistent mesh spacing in the depth dimension throughout a given region. This is used for the FSFs and emitters. The `y_multibox` function is used in the base region of the subcells. This creates a Sentaurus Mesh multibox with a ratio argument to describe a mesh where the element sizes are allowed to increase. A positive ratio argument sets the increasing sizing to being on the top surface of the region, and a negative ratio argument signifies that mesh sizing growth begins on the bottom surface, and a list containing two ratio arguments can be used. This is used to create an asymmetric mesh in the base region, which efficiently controls the error related to optical generation. The tunnel layer mesh is described with `region_triple_y_refdef_with_distance_divider`. This function is used to divide a tunnel junction layer into three parts, such that less mesh points are employed in the central region. The approach is similar to the ARC mesh function, except with three sections instead of five. Another difference is that the ARC mesh functions create a window that does not extend into the cap.

In the *x*-dimension, surfaces are meshed with an `x_line_multibox`. This specifies a small `x` sizing near the edge of the contact, and applies a ratio argument growth term. Then, a normal multibox

is applied to the bulk region, and a larger growth term is allowed. This ensures that the surfaces have a sufficient number of points to resolve the surface recombination, which depends on the carrier concentrations along the length of the surface, without applying the same number of points throughout the bulk.

To mesh the base, the `y_multibox` function is used instead of the `y_region_multibox` function. The arguments extend the multibox over the base and the BSF. The reason is that this multibox is used to control the discretization error with respect to the optical absorption and electron-hole pair generation. This constraint is similar in the BSF and the end of the base. An alternative is to calculate the appropriate mesh sizing after penetration through the base, however, extending the multibox is more efficient and appropriate. It is possible to mesh two segments with one multibox, but this approach is more concise, and given that front segments are thin, the difference is minor.

This function produces the cap region mesh depicted in Figure 61, and the substrate mesh depicted in Figure 62. This substrate mesh is highly efficient, reducing the number of mesh points in the large substrate bulk while maintaining coverage across important surfaces. This is especially effective in solving the large substrate layer of the four-junction device in a timely manner.

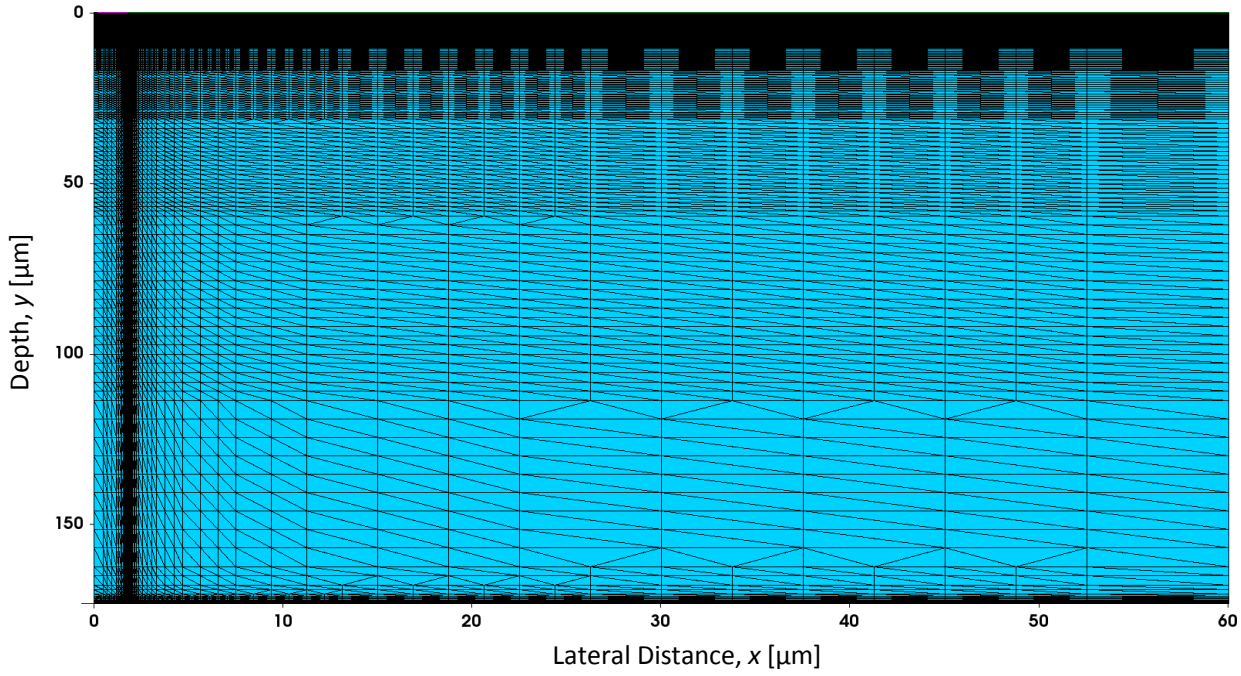

Figure 62: Substrate layer mesh with more x-dimension coverage on surfaces than bulk, specified by using line and window refinement multiboxes.

5-4-3: THREE-JUNCTION DEVICE

This device aims to improve upon InGaP/InGaAs/Ge by substituting germanium with SiGeSn. SiGeSn has a higher bandgap than germanium, and there are sufficient photons to support a higher bandgap material in that position.

If the SiGeSn can be shown to supply sufficient current in this scenario, then there is room for further improvement by reducing the bandgap of the GaInP. This simulation assumes a GaInP

bandgap of 1.9 eV, while the bandgap of GaInP can be adjusted in the range of 1.8-1.9 eV by controlling the atomic ordering with growth conditions.

In this device, the reflection provided by a gold contact underneath the SiGeSn BSF affects the design of the BSF. The mirror is significant in that it reduces the required thickness of the SiGeSn subcell. However, for the mirror to be effective, the BSF must be sufficiently thin and only absorb a minimized portion of the reflection, while still sufficiently repelling minority carriers. This trade-off can be solved by trial-and-error with the one-dimensional configuration. A large doping and small thickness are ideal. There is additional current available in this design if certain components cannot be grown as specified.

Table 9: Triple-Junction Device Epi Layer Specifications

LAYER REGION LABEL	MATERIAL	FILE	THICKNESS (um)	DOPING (/cm^3)	MOLE FRACTION	LAYER NUMBER
arc1	MgF	MgF.par	0.100			;#01
arc2	TiOx	TiOx.par	0.060			;#02
sc1_fsf	AlInP	AlInP.tcl	0.020	6e18	0.50	;#03
sc1_em	GaInP	GaInP.tcl	0.1	5e18	0.50	;#04
sc1_base	GaInP	GaInP.tcl	0.65	-8e16	0.50	;#05
sc1_bsf	AlGaInP	AlGaInP.tcl	0.030	-3e19	0.25	;#06
td1_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30	;#07
td1_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50	;#08
sc2_fsf	AlGaAs	AlGaAs.tcl	0.030	1e19	0.80	;#09
sc2_em	InGaAs	InGaAs.tcl	0.1	3e18	0.01173	;#10
sc2_base	InGaAs	InGaAs.tcl	1.8	-1e17	0.01173	;#11
sc2_bsf	AlGaAs	AlGaAs.tcl	0.030	-1e19	0.20	;#12
td2_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30	;#13
td2_nhigh	GaAs	GaAs.tcl	0.030	6e19		;#14
sc3_fsf	InGaAs	InGaAs.tcl	0.030	7e17	0.01173	;#15
sc3_em	SiGeSn	SiGeSn.tcl	0.1	6e18		;#16
sc3_base	SiGeSn	SiGeSn.tcl	2.0	-2e17		;#17
sc3_bsf	SiGeSn	SiGeSn.tcl	0.03	-5e19		;#18

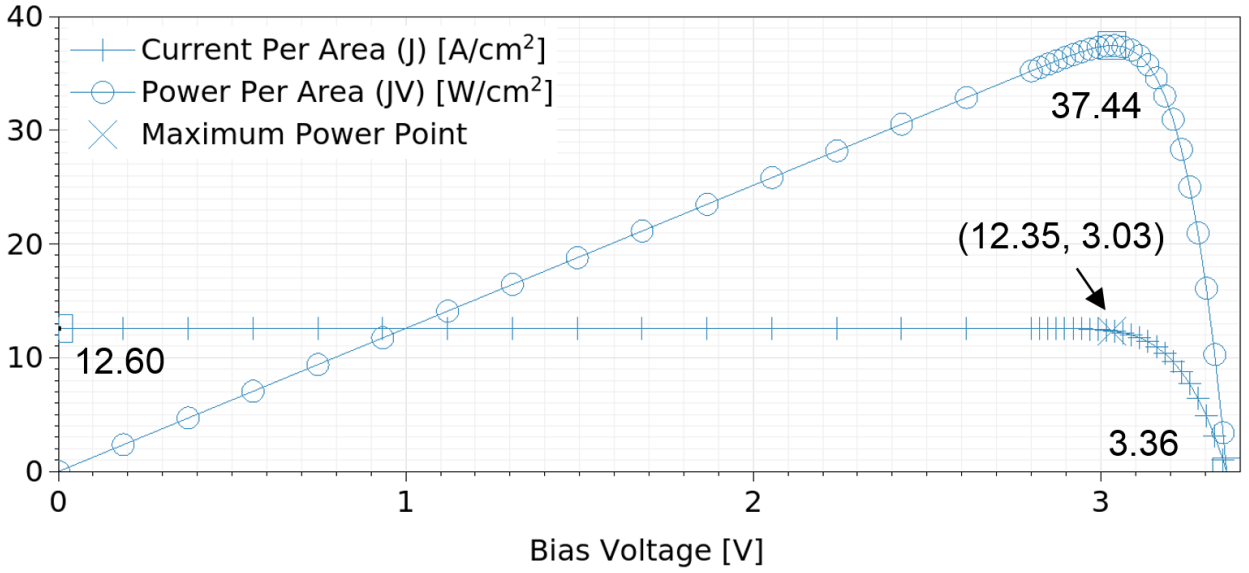

Figure 63: Device J-V and P-V characteristics.

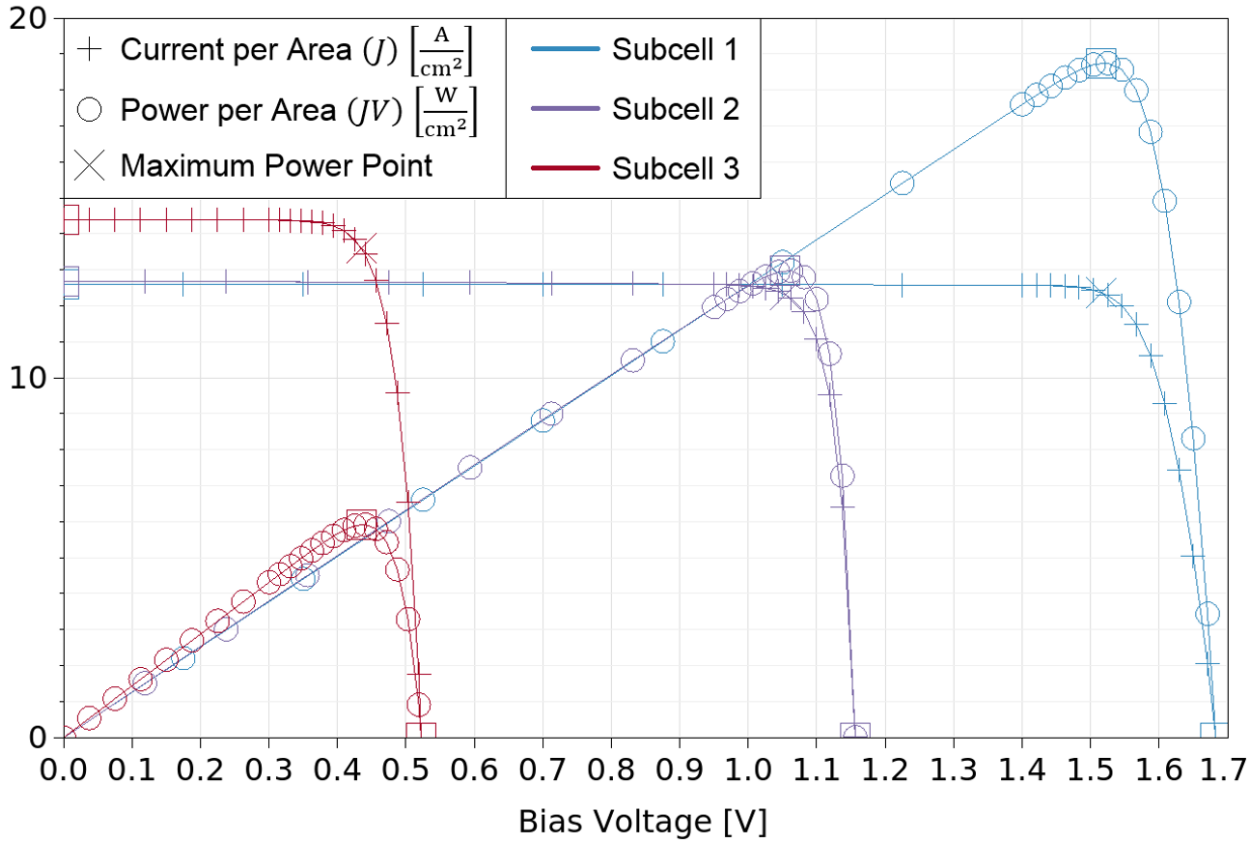

Figure 64: Subcell J-V and P-V characteristics.

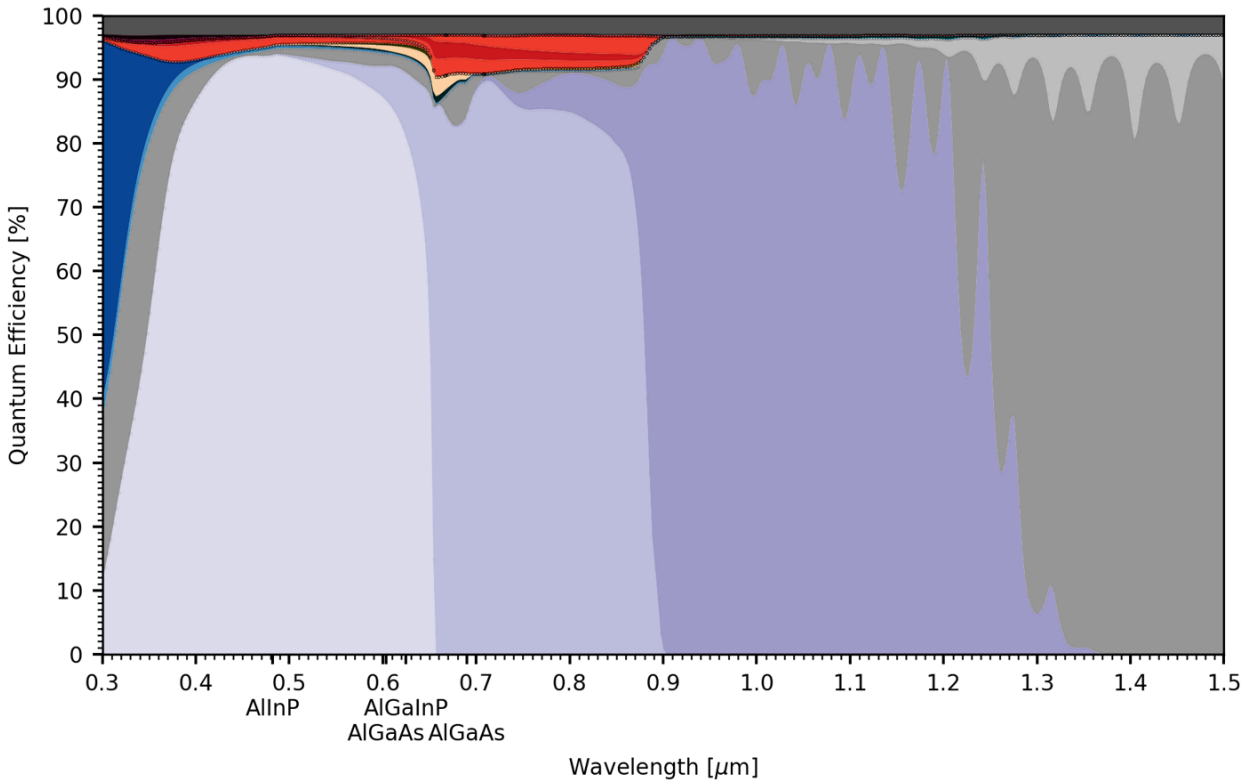

Figure 65: EQE response with losses also shown, SRV=10^5 cm/s.

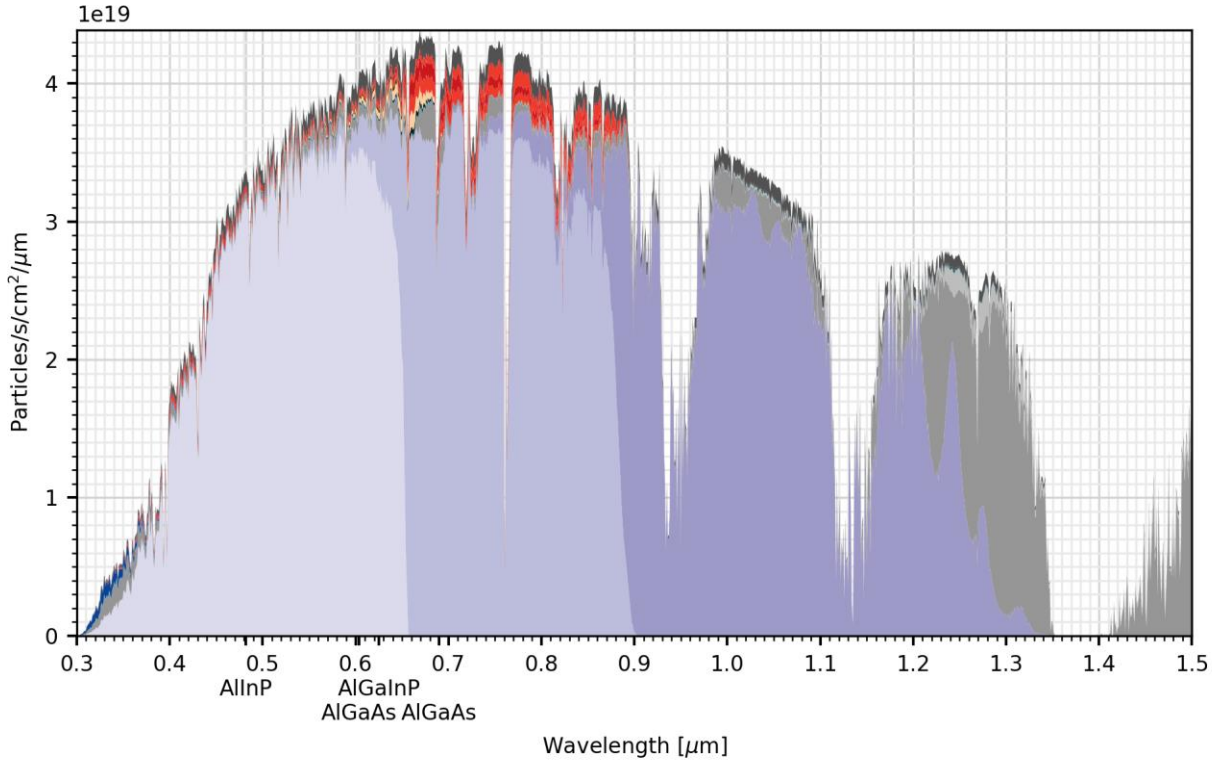

Figure 66: Spectral response of triple-junction device with SiGeSn bottom subcell.

5-4-4: FOUR-JUNCTION DEVICE

The two-dimensional simulation trials of the four-junction scenario show that the emitter thickness must be increased from 30 nm to 80 nm and the base thickness decreased in turn from 380 nm to 340 nm. In the simulation, a very strong BSF responds very well. However, a BSF doped by ion implantation can degrade the quality of the crystal. Therefore, further optimization of this layer depends on the extent of the crystal degradation introduced by the doping method. The doping and thickness suggested here is expected to be within an acceptable range.

The current matching of this device does not leave extra photocurrent available for any additional constraints like the triple-junction above or the six-junction below. The photons are very tightly rationed in this device. This makes the nuances of the current matching condition more important. The front subcells should be prioritized, since they convert the high energy photons more efficiently. They must be thinned to the extent that the bottom subcells can produce sufficient current and do not limit the performance of the higher bandgap subcells. In the choice of whether to thin the front subcells such that the maximum power points of all subcells match, or to thin the front subcells less such that the bottom subcells do not operate at their maximum power point, the latter is preferred. In this way, neither is it optimal for the maximum power points of all the subcells to match (this requires too much thinning,) nor is it optimal for the subcells to have their short-circuit currents match. The bottom subcells are allocated a very small amount of extra current, since the parameters of the novel SiGeSn are subject to more potential error.

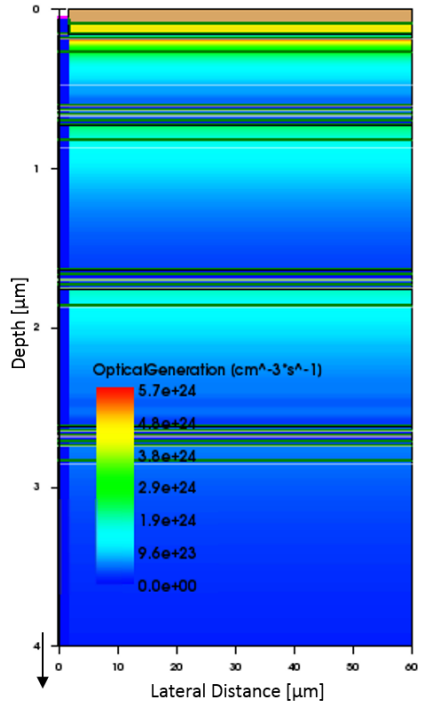

Figure 67: Depiction of the growth portion of the device. Optical generation is shown.

Table 10: Layer Specifications for Four-Junction Solar Cell

LAYER REGION LABEL	MATERIAL	FILE	THICKNESS (um)	DOPING (/cm^3)	MOLE FRACTION
arc1	MgF	MgF.par	0.100		
arc2	TiOx	TiOx.par	0.060		
sc1_fsf	AlInP	AlInP.tcl	0.030	6e18	0.50
sc1_em	GaInP	GaInP.tcl	0.080	5e18	0.50
sc1_base	GaInP	GaInP.tcl	0.33	-8e16	0.50
sc1_bsf	AlGaInP	AlGaInP.tcl	0.030	-3e19	0.25
td1_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30
td1_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50
sc2_fsf	AlGaAs	AlGaAs.tcl	0.030	1e19	0.80
sc2_em	InGaAs	InGaAs.tcl	0.1	3e18	0.01173
sc2_base	InGaAs	InGaAs.tcl	0.8	-1e17	0.01173
sc2_bsf	AlGaAs	AlGaAs.tcl	0.030	-1e19	0.20
td2_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td2_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc3_fsf	InGaAs	InGaAs.tcl	0.030	7e17	0.01173
sc3_em	SiGeSn	SiGeSn.tcl	0.1	1e19	
sc3_base	SiGeSn	SiGeSn.tcl	0.76	-1e18	
sc3_bsf	SiGeSn	SiGeSn.tcl	0.03	-5e19	
td3_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td3_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc4_fsf	SiGeSn	SiGeSn.tcl	0.030	8e19	
sc4_em	Germanium	Ge.tcl	0.100	1e19	
sc4_base	Germanium	Ge.tcl	170	-7e17	
sc4_bsf	Germanium	Ge.tcl	1.5	-8e19	

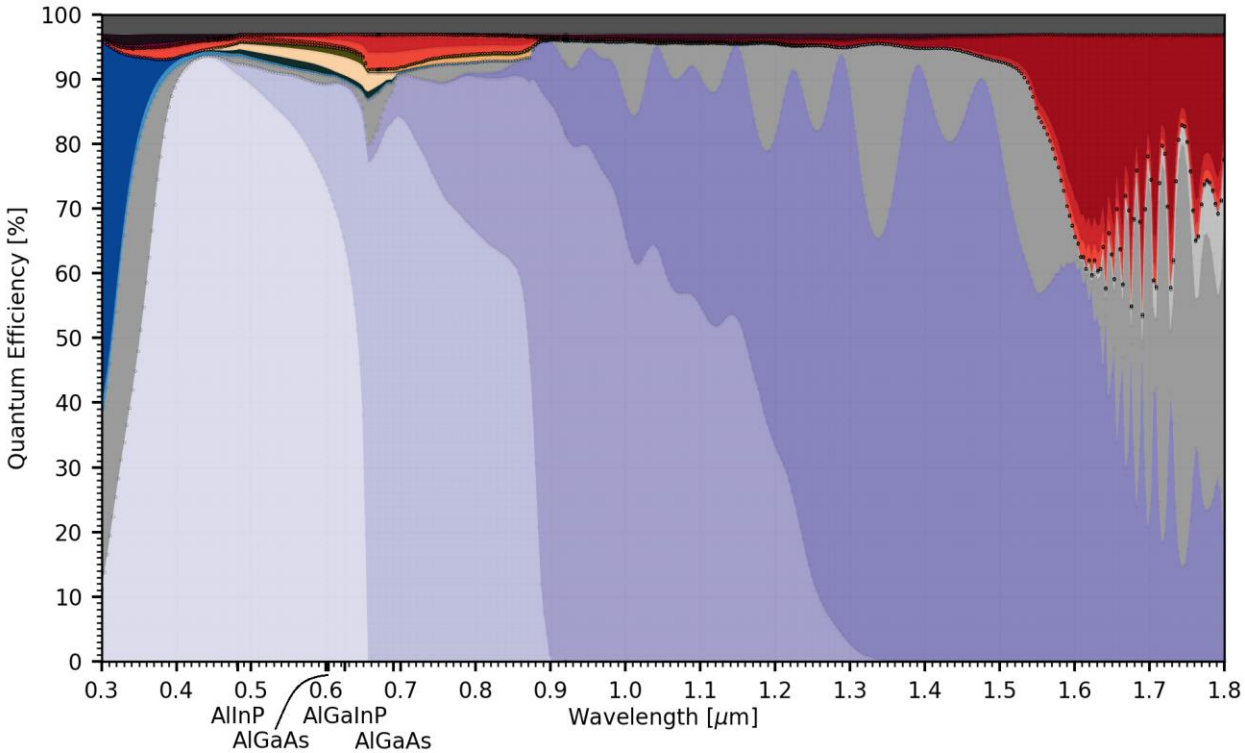

Figure 68: EQE of four-junction device with SRV=10^3 cm/s at short-circuit condition.

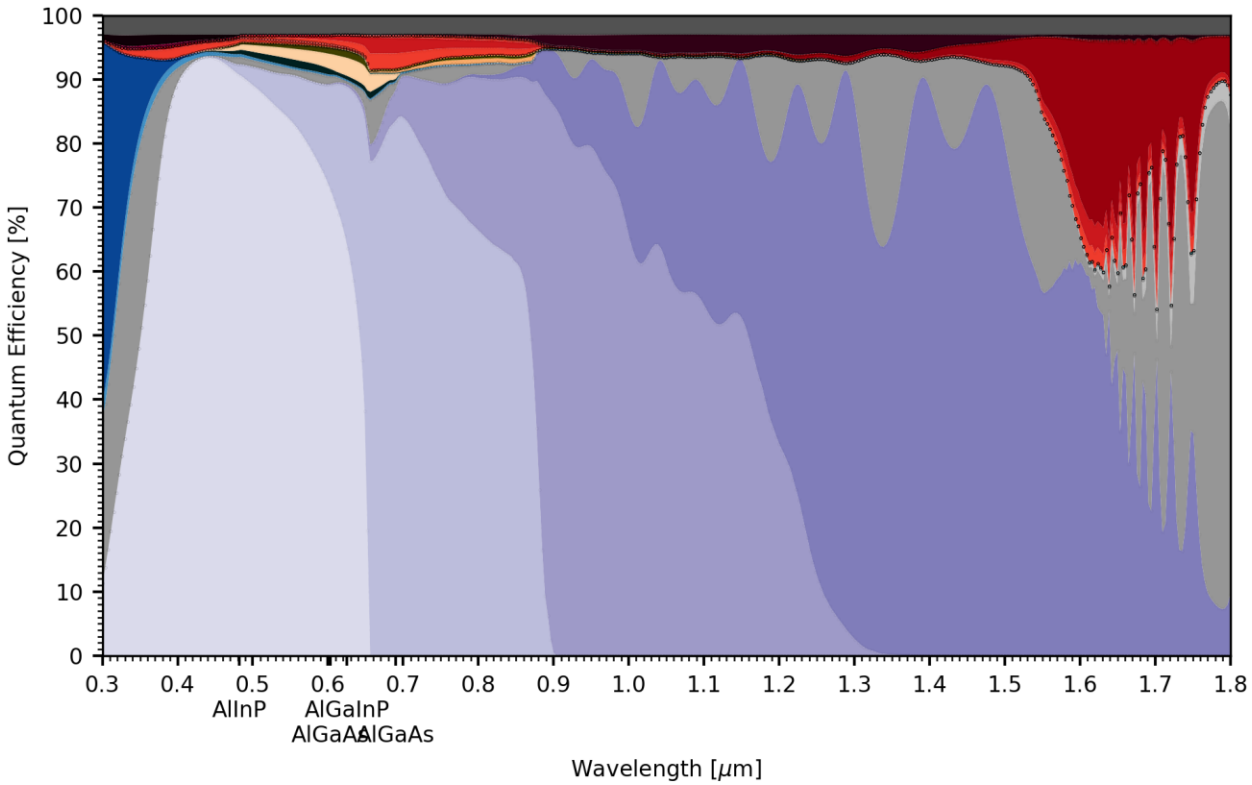

*Figure 69: EQE of four-junction device with SRV=5*10^4 cm/s at short-circuit condition.*

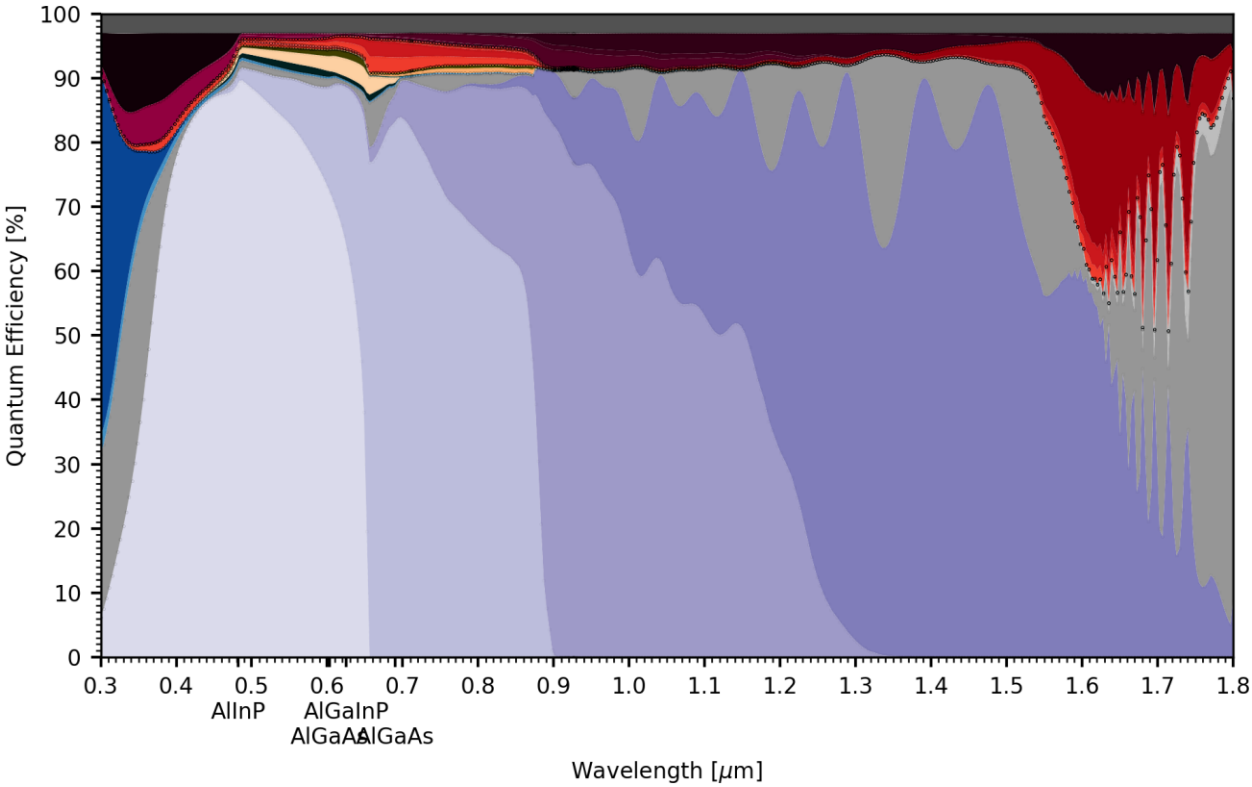

Figure 70: EQE of four-junction device with SRV=10^6 cm/s, showing that the surface recombination model and mesh sources of error continue to pass the self-consistency test for a high degree of surface recombination.

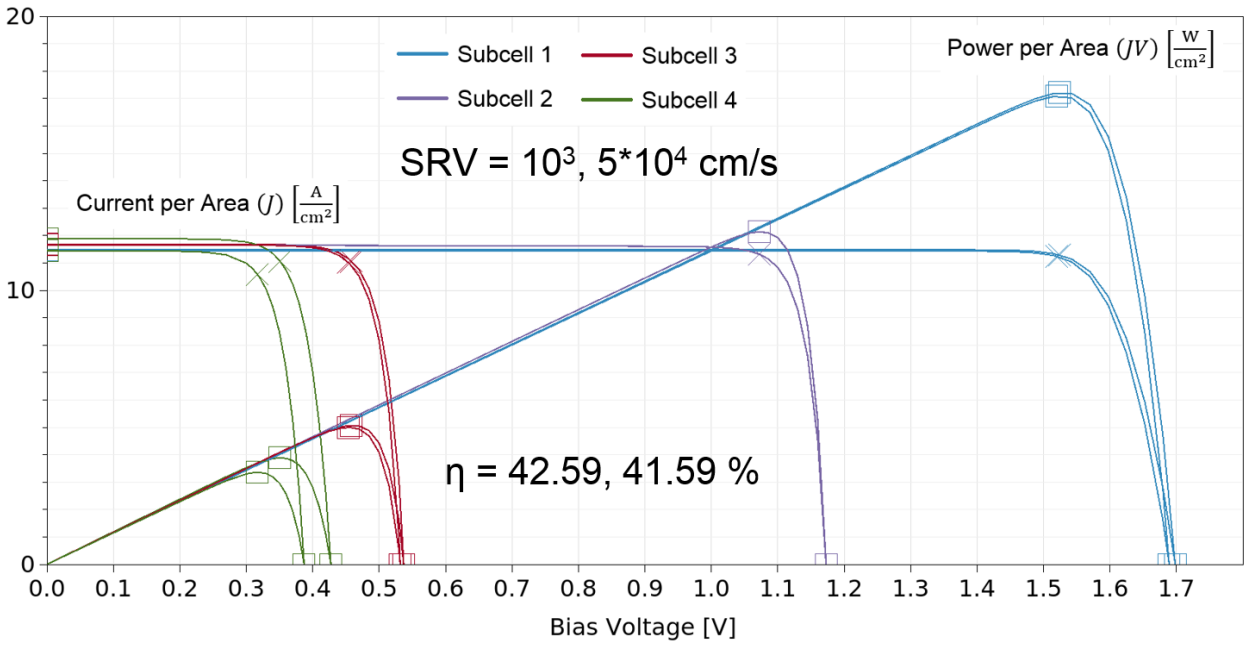

Figure 71: Four-junction subcell J-V and P-V characteristics with two SRV values.

5-4-5: SEVEN-JUNCTION DEVICE

In consideration of two-dimensional effects, the seven-junction device's window layer is thinned to 15 nm, the front segment emitter is expanded to 40 nm to improve the fill factor of the first segment, and the front segment BSF is thinned to 10 nm to promote transmission. The germanium base has been thinned to 4 µm from 12 µm with the previously shown excess current. If 4 µm of grown germanium cannot be grown, the six-junction to follow can be considered. The six-junction is not current matched as tightly, and leaves more tolerance for any source of error that may arise.

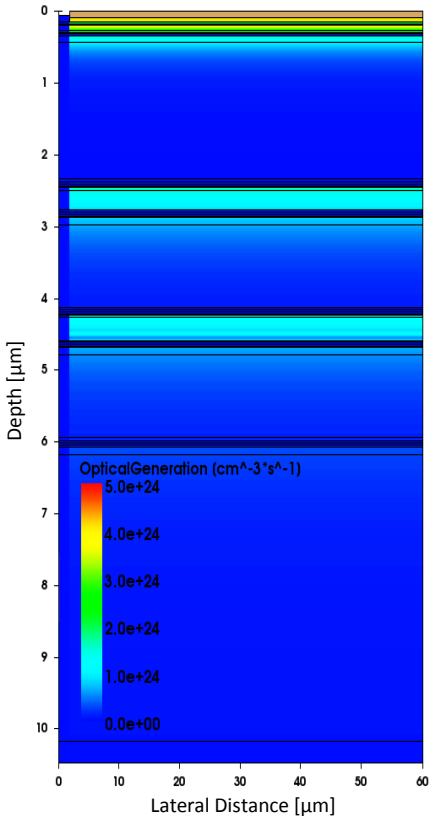

Figure 72: Depiction of seven-junction device.

Table 11: Layer Specifications for Seven-Junction Device

LAYER REGION LABEL	MATERIAL	FILE	THICKNESS (µm)	DOPING (cm^-3)	MOLE FRACTION
arc1	MgF	MgF.par	0.100		
arc2	TiOx	TiOx.par	0.060		
sc1_fsf	AlInP	AlInP.tcl	0.015	6e18	0.50
sc1_em	GaInP	GaInP.tcl	0.040	5e18	0.50
sc1_base	GaInP	GaInP.tcl	0.052	-1e17	0.50
sc1_bsf	AlGaInP	AlGaInP.tcl	0.010	-1e19	0.25
td1_phigh	AlGaAs	AlGaAs.tcl	0.020	-2e20	0.30
td1_nhigh	GaInP	GaInP.tcl	0.020	6e19	0.50
sc2_fsf	AlInP	AlInP.tcl	0.020	6e18	0.50
sc2_em	GaInP	GaInP.tcl	0.10	5e18	0.50
sc2_base	GaInP	GaInP.tcl	1.90	-1e17	0.50
sc2_bsf	AlGaInP	AlGaInP.tcl	0.04	-3e19	0.25
td2_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30
td2_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50
sc3_fsf	AlGaAs	AlGaAs.tcl	0.020	1e19	0.80
sc3_em	InGaAs	InGaAs.tcl	0.040	3e18	0.01173
sc3_base	InGaAs	InGaAs.tcl	0.278	-1e17	0.01173
sc3_bsf	AlGaAs	AlGaAs.tcl	0.020	-1e19	0.20
td3_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30
td3_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50
sc4_fsf	AlGaAs	AlGaAs.tcl	0.020	1e19	0.80
sc4_em	InGaAs	InGaAs.tcl	0.10	3e18	0.01173
sc4_base	InGaAs	InGaAs.tcl	1.15	-1e17	0.01173
sc4_bsf	AlGaAs	AlGaAs.tcl	0.030	-1e19	0.20
td4_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td4_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc5_fsf	InGaAs	InGaAs.tcl	0.015	7e17	0.01173
sc5_em	SiGeSn	SiGeSn.tcl	0.040	1e19	
sc5_base	SiGeSn	SiGeSn.tcl	0.325	-1e18	
sc5_bsf	SiGeSn	SiGeSn.tcl	0.020	-2e18	
td5_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td5_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc6_fsf	SiGeSn	SiGeSn.tcl	0.020	2e19	
sc6_em	SiGeSn	SiGeSn.tcl	0.10	1e19	
sc6_base	SiGeSn	SiGeSn.tcl	1.15	-1e18	
sc6_bsf	SiGeSn	SiGeSn.tcl	0.050	-4e18	
td6_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td6_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc7_fsf	SiGeSn	SiGeSn.tcl	0.030	2e19	
sc7_em	Germanium	Ge.tcl	0.10	1e19	
sc7_base	Germanium	Ge.tcl	4.0	-7e17	
sc7_bsf	Germanium	Ge.tcl	0.3	- 5e19	

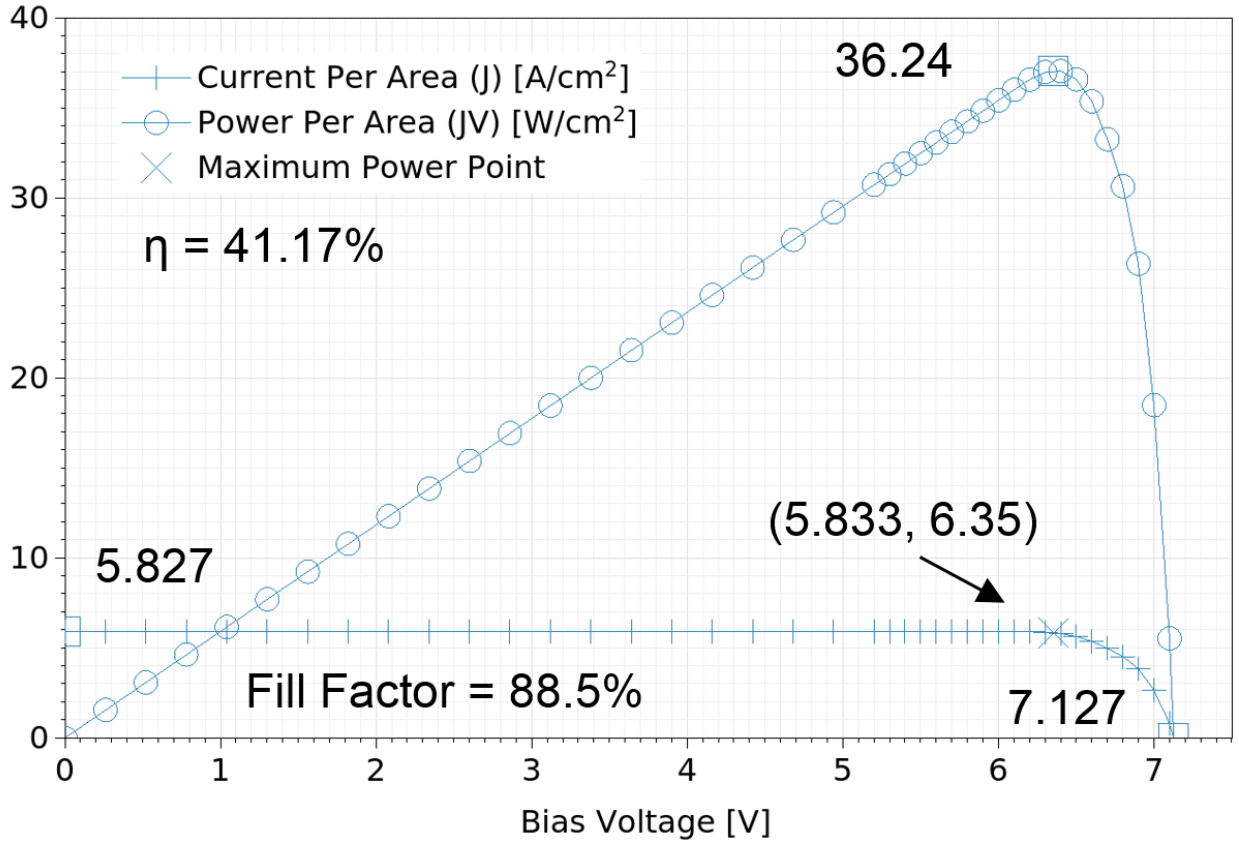

Figure 73: Seven-junction device J(V) and P(V) characteristics.

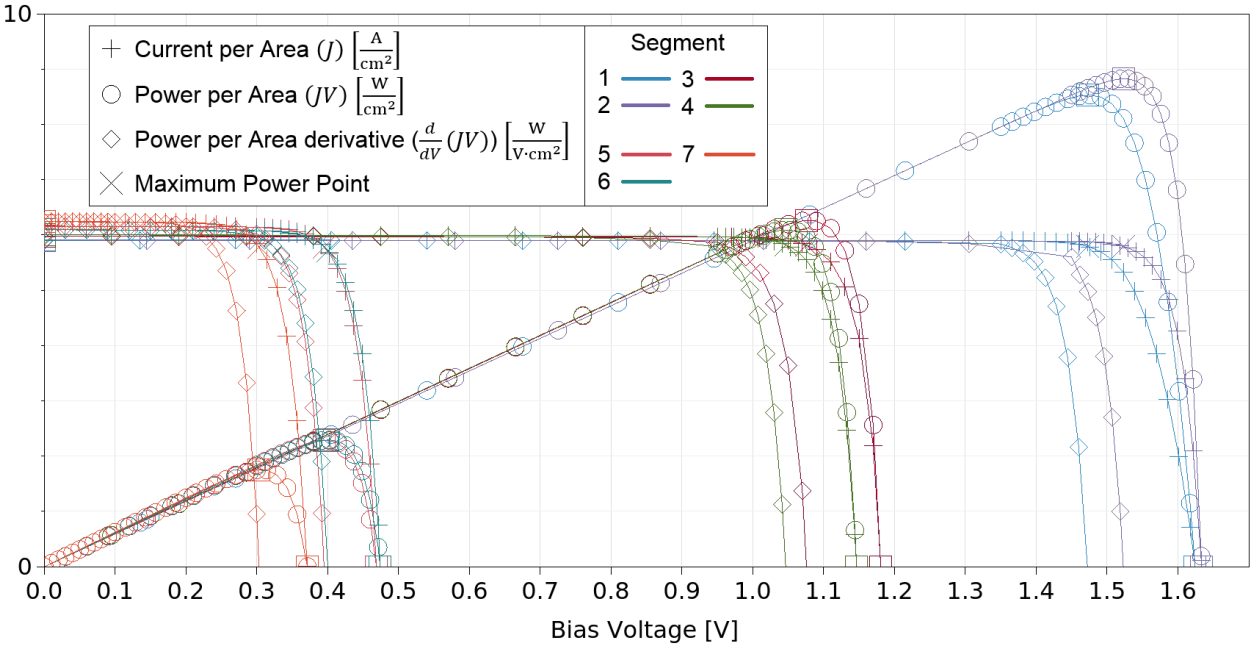

Figure 74: Seven-junction subcell J(V), P(V), and $dP(V)/dV$.

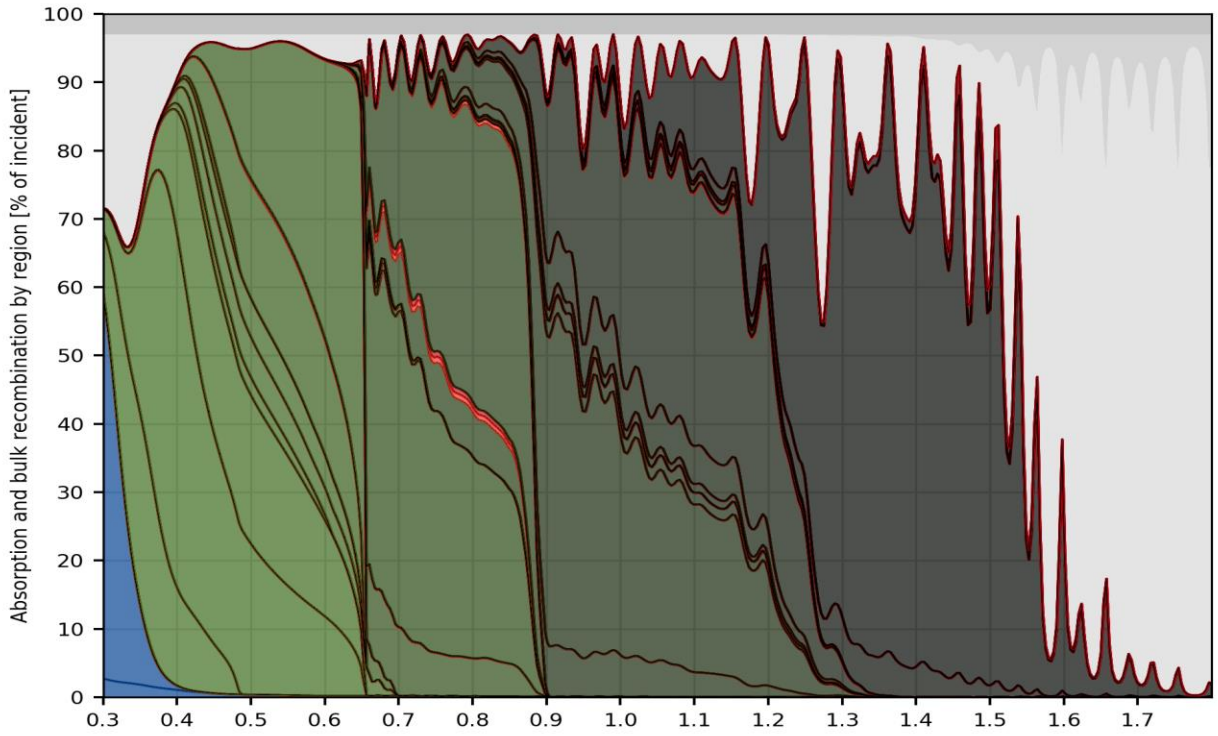

Figure 75: Optical generation in each layer of seven-junction device, with bulk recombination integrated by region and superimposed. Surface recombination does not appear on this plot.

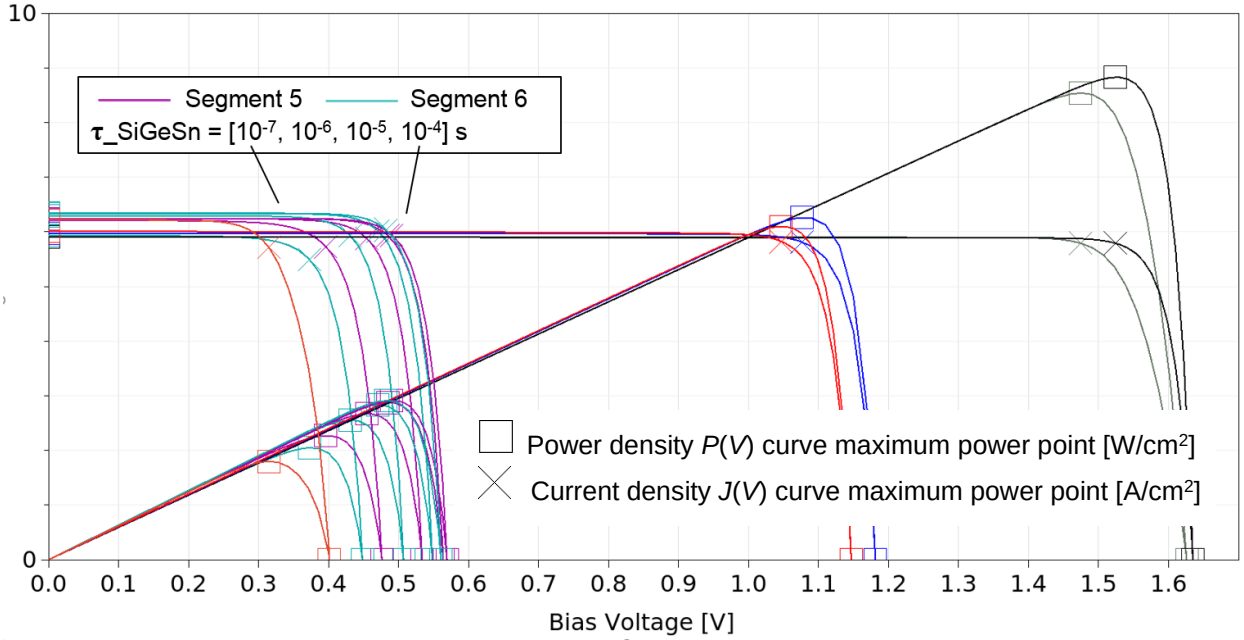

Figure 76: Seven-junction segment J(V) and P(V) characteristics with SiGeSn lifetime parameter sweep.

Figure 76 shows that a thin segment is more resilient than a traditional subcell. The thin segment's current would be acceptable at τ_{SRH_SiGeSn} = 100 ns, but the larger traditional segment would limit the current of the device substantially.

5-4-6: SIX-JUNCTION DEVICE

This six-junction device may alleviate certain problems associated with developing a prototype device. By removing one of the bottom subcells, many requirements on the growth are relaxed, and there is additional tolerance for imperfections in the novel material. The simulated efficiency of the device is slightly lower, though the simulation, does not consider the resistive losses in the grid fingers. There are significant efficiency gains to be made by implementing a segmented device, and pairing it with a high concentration factor, since the segmented device's current-voltage characteristics are very different and significantly reduce resistive loss in the grid fingers. This device is a contender because it will be simpler to realize than the seven-junction device: the SiGeSn and germanium subcells both require only ~ 500 nm of thickness.

Table 12: Layer Specifications for Six-Junction Device

LAYER REGION LABEL	MATERIAL	FILE	THICKNESS (um)	DOPING (/cm^3)	MOLE FRACTION
arc1	MgF	MgF.par	0.100		
arc2	TiOx	TiOx.par	0.060		
sc1_fsf	AlInP	AlInP.tcl	0.020	6e18	0.50
sc1_em	GaInP	GaInP.tcl	0.040	5e18	0.50
sc1_base	GaInP	GaInP.tcl	0.053	-1e17	0.50
sc1_bsf	AlGaInP	AlGaInP.tcl	0.010	-1e19	0.25
td1_phigh	AlGaAs	AlGaAs.tcl	0.020	-2e20	0.30
td1_nhigh	GaInP	GaInP.tcl	0.020	6e19	0.50
sc2_fsf	AlInP	AlInP.tcl	0.020	6e18	0.50
sc2_em	GaInP	GaInP.tcl	0.10	5e18	0.50
sc2_base	GaInP	GaInP.tcl	2.00	-1e17	0.50
sc2_bsf	AlGaInP	AlGaInP.tcl	0.04	-3e19	0.25
td2_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30
td2_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50
sc3_fsf	AlGaAs	AlGaAs.tcl	0.020	1e19	0.80
sc3_em	InGaAs	InGaAs.tcl	0.080	3e18	0.01173
sc3_base	InGaAs	InGaAs.tcl	0.27	-1e17	0.01173
sc3_bsf	AlGaAs	AlGaAs.tcl	0.020	-1e19	0.20
td3_phigh	AlGaAs	AlGaAs.tcl	0.030	-2e20	0.30
td3_nhigh	GaInP	GaInP.tcl	0.030	6e19	0.50
sc4_fsf	AlGaAs	AlGaAs.tcl	0.020	1e19	0.80
sc4_em	InGaAs	InGaAs.tcl	0.10	3e18	0.01173
sc4_base	InGaAs	InGaAs.tcl	1.8	-1e17	0.01173
sc4_bsf	AlGaAs	AlGaAs.tcl	0.030	-1e19	0.20
td4_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td4_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc5_fsf	InGaAs	InGaAs.tcl	0.040	1e19	0.01173
sc5_em	SiGeSn	SiGeSn.tcl	0.1	5e18	
sc5_base	SiGeSn	SiGeSn.tcl	0.4	-3e18	
sc5_bsf	SiGeSn	SiGeSn.tcl	0.050	-1e19	
td5_phigh	AlGaAs	AlGaAs.tcl	0.030	-1.2e20	0.30
td5_nhigh	GaAs	GaAs.tcl	0.030	6e19	
sc6_fsf	Germanium	Ge.tcl	0.030	3e19	
sc6_em	Germanium	Ge.tcl	0.10	3e18	
sc6_base	Germanium	Ge.tcl	0.5	-3e17	
sc6_bsf	Germanium	Ge.tcl	0.02	-3e19	

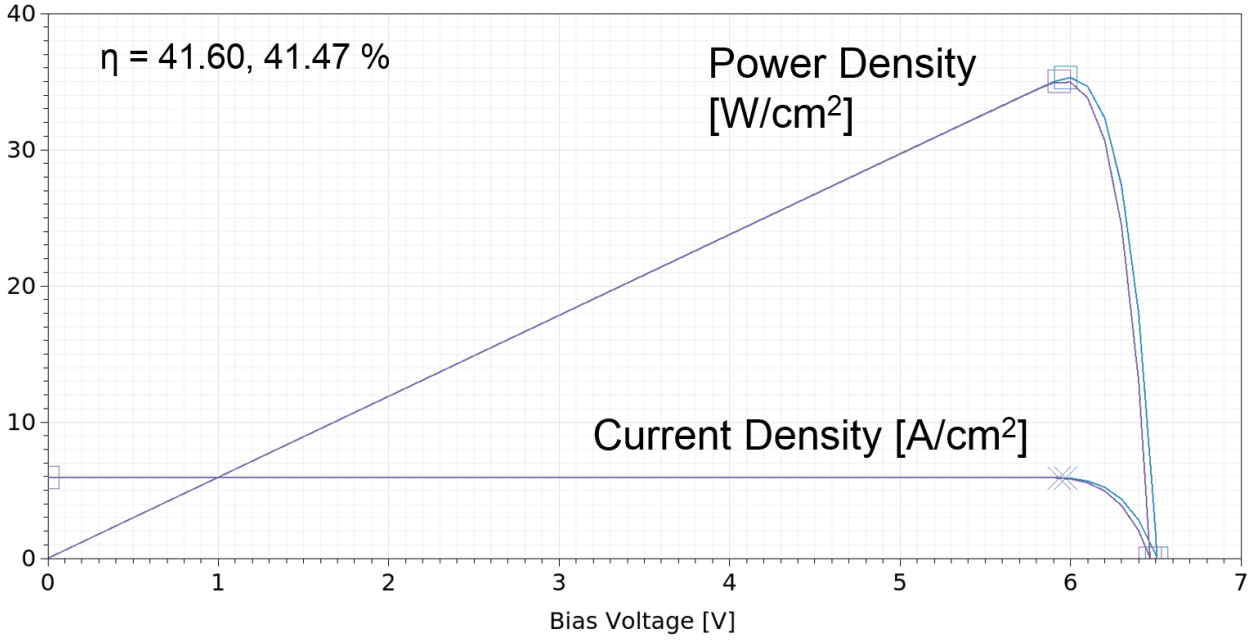

Figure 77: Six-junction device illuminated current and power density characteristics with respect to bias voltage for SRV=10^3 and SRV=$5*10^4$.

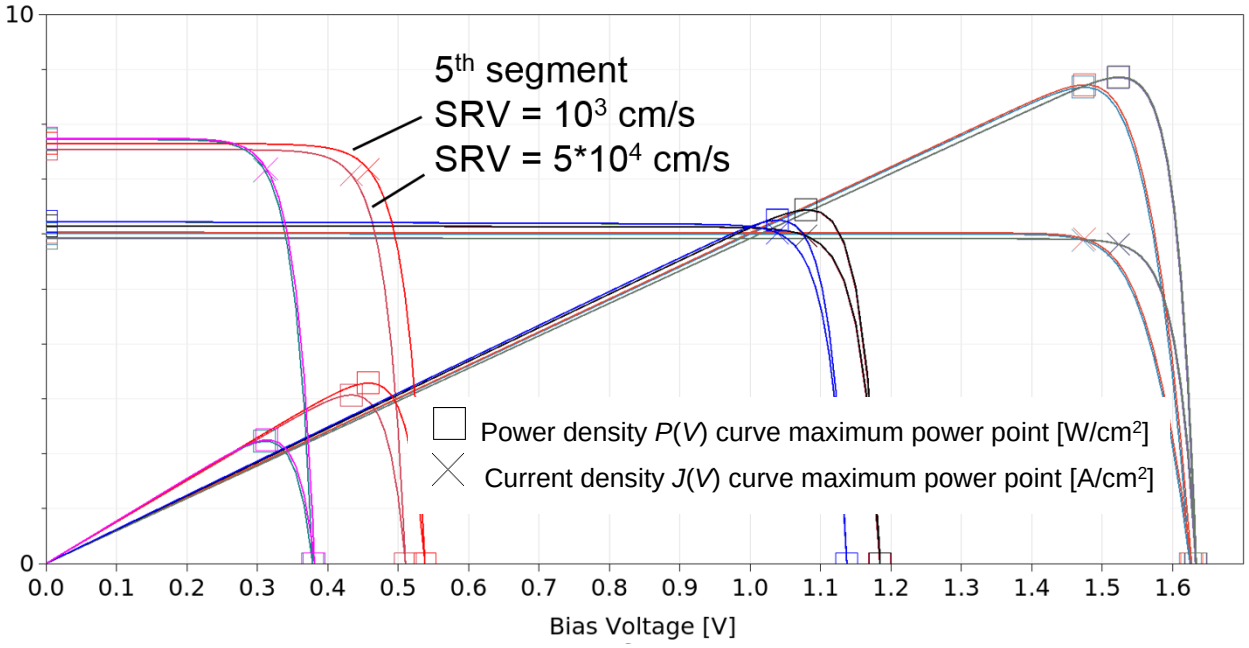

Figure 78: Six-junction device subcell current and power density characteristics for two SRV values, showing excess current available to the bottom subcells.

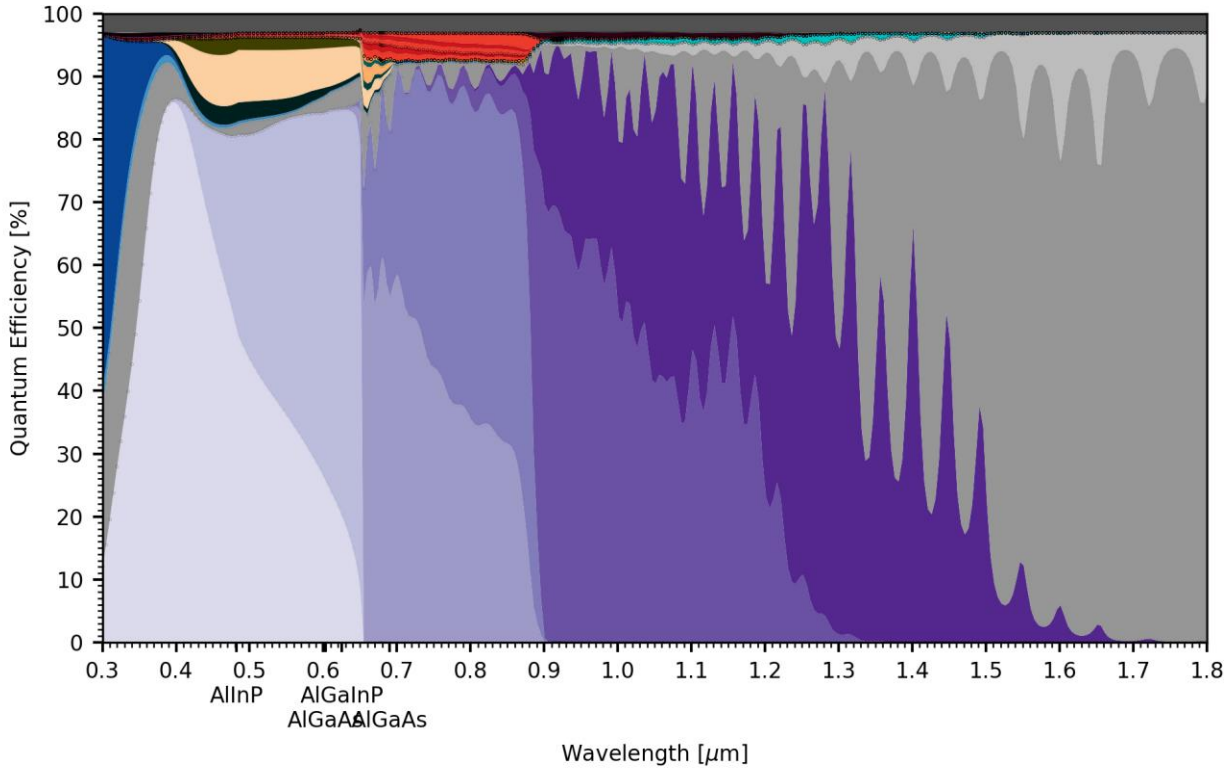

Figure 79: Six-junction device EQE with loss mechanisms.

6: CONCLUSION

In these simulations, triple-junction, quadruple-junction, sextuple-junction, and septuple-junction devices containing silicon-germanium-tin (SiGeSn) have been designed and modeled with the drift-diffusion theory of carrier dynamics. The layer stack specifications are to be found herein. The efficiencies are found to be 41.6%, 42.6%, and 39.2%, and 41.2%, respectively, under 1000x AM1.5D standard illumination. The septuple- and sextuple-junction devices have room for improvement due to parasitic absorption of the first tunnel junction, which would be reduced with the development of a tunnel junction made of higher bandgap materials. Regardless, seven-junction device appears competitive in comparison as is, considering especially that efficiency benefits are available to it at high incident light concentration factors on the order of 1000 times, where the devices with a lower number of junctions would incur significant resistive losses in the grid fingers on the top of the device.

Table 13: Summary of Simulated Devices' Subcell Total Thicknesses

Device	$Ga_{0.5}In_{0.5}P$	$In_{0.012}Ga_{0.988}As$	$Si_{0.14}Ge_{0.822}Sn_{0.038}$	Germanium
3-Junction	820 nm	2.16 μm	2.16 μm	-
4-Junction	480 nm	970 nm	890 nm	170 μm
7-Junction	117 nm, 2.08 μm	358 nm, 1.30 μm	385 nm, 1.32 μm	4.16 μm
6-Junction	123 nm. 2.16 μm	300 nm, 1.95 μm	550 nm	650 nm

Table 14: Summary of Device Characteristics for two SRV values, with 1 μs SiGeSn SRH lifetime.

		3-Junction		**4-Junction**		**6-Junction**		**7-Junction**	
SRV	[cm/s]	10^3	$5*10^4$	10^3	$5*10^4$	10^3	$5*10^4$	10^3	$5*10^4$
J_{sc}	[A/cm^2]	12.60	12.57	11.49	11.44	5.96	5.95	5.92	5.92
J_{mpp}	[A/cm^2]	12.35	12.34	11.25	11.15	5.91	5.90	5.83	5.82
V_{oc}	[V]	3.36	3.35	3.84	3.78	6.51	6.48	7.13	6.91
V_{mpp}	[V]	3.03	3.02	3.41	3.36	5.97	5.93	6.35	6.22
Fill Factor	[%]	88.4	88.6	86.9	86.5	91.0	91.0	87.8	88.5
Power	[W/cm^2]	37.4	37.3	38.3	37.4	35.3	35.0	37.1	36.2
Efficiency	[%]	**41.6**	**41.5**	**42.6**	**41.6**	**39.2**	**38.9**	**41.2**	**40.2**

Modularization and abstraction of the device data structure was appropriate in the command files to enable the definition of larger numbers of layers. This was done in the TCL environment available within the command files by creating a list of layers within the first module to be used by looping constructs throughout the other modules of the project as necessary. Development process human errors in this approach such as calling an undefined command or failing to close parentheses can lead to odd side-effects in the Sentaurus Workbench, but these were minor and no data was lost. In post-processing, a python object holding all device characteristics, simulation data, and related vector prefactors was part of an effective architecture; a useful object to be passed as an argument to various plotting functions.

The radiative, SRH, and Auger recombination throughout the device were calculated and visualized. It was found that a better method than integrating the parasitic absorption within tunnel

junction layers was to calculate the resultant current out of virtual terminals placed adjacent to the tunnel layers. Extracting all loss information in the ramped-laser EQE simulation leads to additional useful information related to discretization error. The primary source of error was noticeable in data from an early step of the computation, in the numerical integration of optical generation. This was controllable to a high degree by refining the mesh in a way that matched the optical generation profile. Some error is related to the Beer-Lambert exponential decay in intensity at the front of each subcell of the device, and some can be related to interference patterns. Meshing to reduce error must be balanced with the simulation's execution time. In the device with an active substrate, which is a very thick layer, it was important to apply a different mesh sizing growth definition to the front of the substrate layer than to the back. A small growth ratio argument of 1.003 was appropriate to apply to a small mesh element at the front of each subcell, while a large growth ratio of 1.2 was suitable at the back of each subcell. Sentaurus Device supports an envelope option to suppress interference in which case a larger growth ratio can be used at the front.

SiGeSn continues to provide a promising landscape of compositions and band structures, and its demonstrated material parameters were instrumental in achieving these results. If they can be grown with high quality and low defect density, and exhibit properties similar to pure germanium, these alloys will greatly benefit the set of conventional lower-order (dual, triple, and quadruple) junction solar cell. The segmentation design strategy can be applied, which leads to very thin layer requirements for some of the components, creating a cost advantage. This can synergize with SiGeSn, which can supply a larger direct bandgap when strained than when unstrained. This work implemented material parameters of an unstrained variety of SiGeSn lattice-matched to germanium. It was found that a significant degree of thinning was required anyway to achieve current matching in the four-junction device. Due to the indirect bandgaps of the bottom two semiconductors, it would be appropriate to consider higher bandgap III-V materials in this structure. It's possible that a direct-bandgap SiGeSn subcell could outperform a germanium subcell, and better convert the 0.69 to 0.84 eV photons of the solar spectrum. This documentation has been sufficiently detailed to show the implementation of the SiGeSn bandgap and complex refractive index models within the software, to enable future work to implement other compositions of SiGeSn. Additionally, the bandgap, electron effective mass, and band offset models use compositional interpolation and bowing expressions from the literature that accept the stoichiometric arguments, which can facilitate the development of models of other compositions of SiGeSn. The next steps include the addition of complex refractive index data, implying the absorption characteristics, for other compositions of SiGeSn, to enable the design of devices incorporating them. Another avenue to consider is a conventional high-order solar cell using a few SiGeSn subcells of differing bandgaps for the accessible range below 1.0 eV.

The SiGeSn is expected to have defect-related recombination in its early stages. Trials were run with adjustments to the SRH lifetime in order to simulate different defect densities within the growth. It was found that a lifetime as low as 1 μs would be acceptable, having a small impact on the *J-V* characteristic of the subcell. By affecting primarily the fill factor and subcell voltage, and less-so the short-circuit current, SRH lifetime at this level should not ruin the performance of the entire device, retaining its ability to current-match with the other subcells, albeit at reduced power. The seven-junction device is slightly more sensitive to the SiGeSn SRH lifetime and could see the

SiGeSn subcell begin to limit the current of the device with a 100 ns lifetime. The trap densities corresponding with these lifetimes can be calculated if the capture coefficient is determined.

WORKS CITED

[1] IPCC 2021, "Summary for Policymakers," in *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, V. Masson-Delmotte, P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. B. R. Matthews, T. K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, Eds. Cambridge University Press. In Press, 2021.

[2] M. M. Wilkins, A. M. Gabr, P. Sharma, H. P. Schriemer, S. Fafard, and K. Hinzer, "4-Junction Solar Cells With Dilute Nitrides: Optimization With Luminescent Coupling," *Proc. 29th EU PVSEC*, no. November, pp. 10–13, 2014.

[3] "Simulation of Triple-Junction GaInP/GaAs/InGaAs Solar Cells with Transfer Matrix Method," TCAD Sentaurus application note. Synopsys SolvNet. [Online]. Available: https://solvnet.synopsys.com.

[4] C. E. Valdivia and K. Hinzer, "Subcell Segmentation for Current Matching and Design Flexibility in Multijunction Solar Cells," *IEEE J. Photovoltaics*, vol. 10, no. 5, pp. 1329–1339, 2020.

[5] R. A. Serway, C. J. Moses, and C. A. Moyer, *Modern Physics*, 3rd ed. Belmont, California, 94002: Thomson Learning Brooks/Cole, 2005.

[6] S. Sekula, "Seeing the Quantum Universe," The Adventures of My Pet Hamster, WordPress, September 21, 2019. [Blog]. Available: https://steve.cooleysekula.net/blog/2019/09/21/seeing-the-quantum-universe/. [Accessed: Jan. 11 2022].

[7] C. Kittel, *Introduction to Solid State Physics*, 8th ed. Hoboken, New Jersey, 07030-5774: John Wiley & Sons, Inc., 2005.

[8] R. A. Serway and J. Jewett, *Physics for Scientits and Engineers*, 6th ed. Belmont, California, 94002: Thomson Learning Brooks/Cole, 2004.

[9] S. Hubbard, "Absorption and Generation," in *Photovoltaic Solar Energy: From Fundamentals to Applications*, 1st ed., A. Reinders, P. Verlinden, W. van Sark, and A. Freundlich, Eds. John Wiley & Sons, Ltd., 2017.

[10] S. O. Kasap, *Optoelectronics and Photonics: Principles and Practices*, 2nd ed. Upper Saddle River, New Jersey, 07458: Pearson Education, Inc., 2013.

[11] R. F. Pierret, *Modular Series on Solid State Devices: Advanced Semiconductor Fundamentals*, 2nd ed., vol. 6. Upper Saddle River, New Jersey, 07458: Pearson Education, Inc., 2003.

[12] "Simulation of Triple-Junction GaInP / GaAs / InGaAs Solar Cells with Transfer Matrix Method," *Structure*.

[13] Fraunhofer Institute for Solar Energy Systems ISE. "III-V Epitaxy and Solar Cells." [Online]. Available: https://www.ise.fraunhofer.de/en/business-areas/photovoltaics/iii-v-and-concentrator-photovoltaics/iii-v-epitaxy-and-solar-cells.html. [Accessed: 15-Jan-2022].

[14] M. M. Wilkins, C. E. Valdivia, A. M. Gabr, D. Masson, S. Fafard, and K. Hinzer, "Luminescent coupling in planar opto-electronic devices," *J. Appl. Phys.*, vol. 118, no. 14, 2015.

[15] C. Hu, *Modern Semiconductor Devices for Integrated Circuits*, vol. 2. Upper Saddle River, New Jersey, 07458: Prentice Hall, 2010.

[16] H. Kampwerth, "Measurement of Carrier Lifetime, Surface Recombination Velocity, and Emitter Recombination Parameters," in *Photovoltaic Solar Energy: From Fundamentals to Applications*, 1st ed., A. Reinders, P. Verlinden, W. van Sark, and A. Freundlich, Eds. John Wiley & Sons, Ltd., 2017.

[17] R. Garrison and R. Kleiman, "Higher efficiency tandem solar cells through composite-cell current matching," *Opt. Express*, vol. 27, no. 8, p. A543, 2019.

[18] C. B. Honsberg and S. G. Bowden, "Photovoltaics Education Website," 2020. [Online]. Available: www.pveducation.org.

[19] ASTM G173-03(2020), "Standard Tables for Reference Solar Spectral Irradiances: Direct Normal and Hemispherical on 37° Tilted Surface", ASTM International, West Conshohocken, PA, 2020. [Online]. Available: www.astm.org

[20] National Renewable Energy Laboratory, "Grid Modernization: Reference Air Mass 1.5 Spectra." [Online]. Available: nrel.gov/grid/solar-resource/spectra-am1.5.html.

[21] M. Kondow *et al.*, "GaInNAs: A novel material for long-wavelength semiconductor lasers," *IEEE J. Sel. Top. Quantum Electron.*, vol. 3, no. 3, pp. 719–729, 1997.

[22] J.-H. Fournier-Lupien *et al.*, "In Situ Studies of Germanium-Tin and Silicon-Germanium-Tin Thermal Stability," *ECS Trans.*, vol. 64, no. 6, pp. 903–911, 2014.

[23] C. A. Stephenson *et al.*, "Band structure of germanium carbides for direct bandgap silicon photonics," *J. Appl. Phys.*, vol. 120, no. 5, 2016.

[24] A. Attiaoui and O. Moutanabbir, "Indirect-to-direct band gap transition in relaxed and strained Ge 1-x-ySixSny ternary alloys," *J. Appl. Phys.*, vol. 116, no. 6, pp. 0–15, 2014.

[25] P. Moontragoon, Z. Ikonić, and P. Harrison, "Band structure calculations of Si-Ge-Sn alloys: Achieving direct band gap materials," *Semicond. Sci. Technol.*, vol. 22, no. 7, pp. 742–748, 2007.

[26] T. Wilson *et al.*, "Single and multi-junction solar cells utilizing a 1.0 eV SiGeSn junction," *AIP Conf. Proc.*, vol. 1766, pp. 0–6, 2016.

[27] Q. Chen, L. Zhang, H. Zhou, W. Li, B. K. Son, and C. S. Tan, "The effects of strain and composition on the conduction-band offset of direct band gap type-I GeSn/GeSnSi quantum dots for CMOS compatible mid-IR light source," *Semicond. Sci. Technol.*, vol. 35, no. 2, 2020.

[28] J. D. Lorentzen *et al.*, "Photoluminescence in Si1-x-yGexCy alloys," vol. 70, no. 18, pp. 2353–2355, 1996.

[29] V. R. D'Costa *et al.*, "Optical critical points of thin-film Ge1-y Sny alloys: A comparative Ge1-y Sny Ge1-x six study," *Phys. Rev. B - Condens. Matter Mater. Phys.*, vol. 73, no. 12, pp. 1–16, 2006.

[30] R. Soref. “Silicon-based silicon-germanium-tin heterostructure photonics.” *Phil. Trans. R. Soc. A.*, vol. 372, no. 2021: 20130113, 2014, doi: 10.1098/rsta.2013.0113

[31] Y. P. Varshni, “Temperature dependence of the energy gap in semiconductors,” *Physica*, vol. 34, no. 1, pp. 149–154, 1967.

[32] *Epi / MatPar User Guide*, April 2008. Mountain View, California: Synopsys, Inc., 2008.

[33] Ioeffe Institute. “New Semiconductor Materials: Characteristics and Properties.” [Online]. Available: http://matprop.ru/semicond (accessed Oct. 15 2021).

[34] M. E. Levinstein, S. Rumyantsev, and M. Shur, Eds., *Handbook series on Semiconductor Parameters*, vol. 2. London: World Scientific, 1999.

[35] O. Madelung, Ed., *Semiconductors: Group IV Elements and III-V Compounds (Data in Science and Technology)*. Berlin/Heidelberg/New York: Springer-Verlag, 1991.

[36] G. Chang, S. Chang, and S. L. Chuang, “Strain-Balanced GezSn1−z-SixGeySn1 − x − y Multiple-Quantum-Well Lasers,” *IEEE J. Quantum Electron.*, vol. 46, no. 12, pp. 1813–1820, 2010.

[37] O. Madelung, M. Schultz, and H. Weiss, *Physics of Group IV Elements and III-V Compounds*, 1st ed., vol. 17a. New York: Springer-Verlag, 1982.

[38] V. R. D’Costa, C. S. Cook, J. Menéndez, J. Tolle, J. Kouvetakis, and S. Zollner, “Transferability of optical bowing parameters between binary and ternary group-IV alloys,” *Solid State Commun.*, vol. 138, no. 6, pp. 309–313, 2006.

[39] V. R. D’Costa, Y. Y. Fang, J. Tolle, J. Kouvetakis, and J. Menéndez, “Tunable optical gap at a fixed lattice constant in group-IV semiconductor alloys,” *Phys. Rev. Lett.*, vol. 102, no. 10, pp. 1–4, 2009.

[40] D. Stange *et al.*, “Quantum Confinement Effects in GeSn/SiGeSn Heterostructure Lasers,” *IEEE Int. Electron Devices Meet.*, pp. 24.2.1-24.2.4, 2017.

[41] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[42] P. Moontragoon, R. A. Soref, and Z. Ikonic, “The direct and indirect bandgaps of unstrained Si xGe 1-x-ySn y and their photonic device applications,” *J. Appl. Phys.*, vol. 112, no. 7, 2012.

[43] N. Mori, “Electronic band structures of silicon–germanium (SiGe) alloys,” in *Silicon–germanium (SiGe) nanostructures: Production, properties and applications in electronics*, Y. Shiraki and U. Noritaka, Eds. Philadelphia, PA: Woodhead Publishing Limited, 2011, pp. 26–42.

[44] I. Vurgaftman, J. R. Meyer, and L. R. Ram-Mohan, “Band parameters for III-V compound semiconductors and their alloys,” *J. Appl. Phys.*, vol. 89, no. 11 I, pp. 5815–5875, 2001.

[45] D. Rainko *et al.*, “Investigation of carrier confinement in direct bandgap GeSn/SiGeSn 2D and 0D heterostructures,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–13, 2018.

[46] H. Wang *et al.*, “Theoretical Investigation of performance enhancement in GeSn/SiGeSn Type-II staggered heterojunction tunneling FET,” *IEEE Trans. Electron Devices*, vol. 63, no. 1, pp. 303–310, 2016.

[47] M. Jaros, “Simple analytic model for heterojnnction band offsets,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[48] R. Soref, J. Kouvetakis, and J. Menendez, “Advances in SiGeSn/Ge technology,” *Mater. Res. Soc. Symp. Proc.*, vol. 958, no. January 2006, pp. 13–24, 2007.

[49] V. V. Afanas’ev, “Electron Band Alignment at Interfaces of Semiconductors with Insulating Oxides: An Internal Photoemission Study,” *Adv. Condens. Matter Phys.*, vol. 2014, 2014.

[50] P. Pearce, T. Wilson, A. Johnson, and N. Ekins-Daukes, “Characterization of SiGeSn for Use as a 1 eV Sub-Cell in Multi-Junction Solar Cells,” *2018 IEEE 7th World Conf. Photovolt. Energy Conversion, WCPEC 2018 - A Jt. Conf. 45th IEEE PVSC, 28th PVSEC 34th EU PVSEC*, pp. 943–948, 2018.

[51] L. Vegard, “Die Konstitution der Mischkristalle und die Raumfüllung der Atome,” *Zeitschrift für Phys.*, vol. 5, no. 1, pp. 17–26, 1921.

[52] M. P. Polak, P. Scharoch, and R. Kudrawiec, “The electronic band structure of Ge1-xSnx in the full composition range: Indirect, direct, and inverted gaps regimes, band offsets, and the Burstein-Moss effect,” *J. Phys. D. Appl. Phys.*, vol. 50, no. 19, 2017.

[53] P. Aella, C. Cook, J. Tolle, S. Zollner, A. V. G. Chizmeshya, and J. Kouvetakis, “Optical and structural properties of Si xSn yGe 1-x-y alloys,” *Appl. Phys. Lett.*, vol. 84, no. 6, pp. 888–890, 2004.

[54] J. Piprek, *Semiconductor Optoelectronic Devices: Introduction to PHysics and Simulation*. San Diego, California: Academic Press, 2003.

[55] A. Rohatgi, “WebPlotDigitizer,” 2019. [Online]. Available: https://automeris.io/WebPlotDigitizer.

[56] K. Nishioka, T. Takamoto, T. Agui, M. Kaneiwa, Y. Uraoka, and T. Fuyuki, “Evaluation of InGaP/InGaAs/Ge triple-junction solar cell and optimization of solar cell’s structure focusing on series resistance for high-efficiency concentrator photovoltaic systems,” *Sol. Energy Mater. Sol. Cells*, vol. 90, no. 9, pp. 1308–1321, 2006.

[57] M. Steiner, G. Siefer, T. Schmidt, M. Wiesenfarth, F. Dimroth, and A. W. Bett, “43% Sunlight to Electricity Conversion Efficiency Using CPV,” *IEEE J. Photovoltaics*, vol. 6, no. 4, pp. 1020–1024, 2016.

[58] *Sentaurus Device User Guide*, L-2016.03. Mountain View, California: Synopsys, Inc., 2016.

[59] A. W. Walker, “Bandgap Engineering of Multi-Junction Solar Cells Using Nanostructures for Enhanced Performance Under Concentrated Illumination,” *uOttawa Thesis - PhD*, 2013.

[60] *Sentaurus Structure Editor User Guide*, L-2016.03. Mountain View, California: Synopsys, Inc., 2016.

[61] A. W. Walker, O. Theriault, J. F. Wheeldon, and K. Hinzer, “The effects of absorption and recombination on quantum dot multijunction solar cell efficiency,” *IEEE J. Photovoltaics*, vol. 3, no. 3, pp. 1118–1124, 2013.

[62] *Sentaurus Workbench User Guide*, L-2016.03. Mountain View, California: Synopsys, Inc., 2016.

[63] CoolSoft, "PDF Property Extension," 2021. [Online]. Available: https://coolsoft.altervista.org/en/pdfpropertyextension.

[64] M. M. Wilkins *et al.*, "Multijunction solar cell designs using silicon bottom subcell and porous silicon compliant membrane," *IEEE J. Photovoltaics*, vol. 3, no. 3, pp. 1125–1131, 2013.

[65] E. E. Perl *et al.*, "(Al)GaInP/GaAs Tandem Solar Cells for Power Conversion at Elevated Temperature and High Concentration," *IEEE J. Photovoltaics*, vol. 8, no. 2, pp. 640–645, 2018.

[66] E. W. Williams, "Gold in Solar Cells: A Review of Current Work," *Solid State Technol.*, vol. 24, no. 1, pp. 80–86, 1981.

[67] Z. H. Mohammed, "The Fresnel Coefficient of Thin Film Multilayer Using Transfer Matrix Method TMM," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 518, no. 3, 2019.

[68] A. Attiaoui and O. Moutanabbir, "Indirect-to-direct band gap transition in relaxed and strained Ge 1-x-ySixSny ternary alloys," *J. Appl. Phys.*, vol. 116, no. 6, 2014.

[69] H. Ye and J. Yu, "Germanium epitaxy on silicon," *Sci. Technol. Adv. Mater.*, vol. 15, no. 2, 2014.

[70] S. Hubbard, "Recombination," in *Photovoltaic Solar Energy: From Fundamentals to Applications*, 1st ed., A. Reinders, P. Verlinden, W. van Sark, and A. Freundlich, Eds. John Wiley & Sons, Ltd., 2017.

[71] C. Tserbak, H. M. Polatoglou, and G. Theodorou, "Unified approach to the electronic structure of strained Si/Ge superlattices," *Phys. Rev. B*, vol. 47, no. 12, pp. 7104–7124, 1993.

[72] M. M. Rieger and P. Vogl, "Electronic-band parameters in strained Si_{1-x}Ge_x alloys on Si_{1-y}Ge_y substrates," *Phys. Rev. B - Condens. Matter Mater. Phys.*, vol. 48, no. 19, pp. 14 276-14 287, 1993.

[73] T. Brudevoll, D. S. Citrin, M. Cardona, and N. E. Christensen, "Electronic structure of alpha-Sn and its dependence on hydrostatic strain," *Phys. Rev. B*, vol. 48, no. 12, pp. 8629–8635.

APPENDIX A: CONDUCTION BAND OFFSET CALCULATION

Each material must implement the following functions independently: `_alpha`, `_beta`, `_Eg0`, and `_Tpar`. Documented in section 3-6.

```
proc Varshni_term {T} {
  return [expr {1.0*[_alpha]*$T*$T/($T+[_beta])}]
}
proc Eg_difference {T1 T2} {
  return [expr {[Varshni_term $T1] - [Varshni_term $T2]}]
}
proc _Eg {} {
  return [expr {[_Eg0] + [Eg_difference [_Tpar] $::temp]}]
}
proc vb_difference {T1 T2} {
  return [expr {-0.5*[Eg_difference $T1 $T2]}]
}
proc vbo_difference {T1 T2} {
  return [expr {[vb_difference $T1 $T2] - [${::substrate}::Bandgap::vb_difference $T1 $T2]}]
}
proc delta_Eg {} {
  return [expr {[_Eg] - [evalVar "${::substrate}::Bandgap::Eg"]}]
}
proc _cbo_absolute {vbo} {
  return [expr {[delta_Eg] + $vbo}]
}
proc _cbo_relative {cbo_absolute} {
  return [expr {$cbo_absolute/[delta_Eg]}]
}
if {[info exists ::0K_vbo]} {
    set ::cbo [_cbo_relative [_cbo_absolute [expr {$::0K_vbo + [vbo_difference $::0K_vbo $::temp]}]]]
} elseif {[info exists ::Tpar_vbo]} {
    set ::cbo [_cbo_relative [_cbo_absolute [expr {$::Tpar_vbo + [vbo_difference $::Tpar_vbo $::temp]}]]]
} elseif {[info exists ::vbo]} {
    set ::cbo [_cbo_relative [_cbo_absolute $::vbo]]
} else {
    puts stderr "$::material init did not find any heterostructure band discontinuity input - using bulk data"
}
```

APPENDIX B: EFFECT OF LATTICE CONSTANT BOWING ON GERMANIUM LATTICE MATCHING SIGESN COMPOSITIONS

The set of compositions that lattice match to germanium in consideration of bowing can be determined by solving for $y(x)$ on a point-by-point basis.

$$a_{SiGeSn} = (1-x-y)Ge + xSi + ySn - b_{GeSn}y(1-x-y) - b_{GeSn}x(1-x-y) - b_{SiSn}xy$$

$$x^2 b_{GeSi} + y^2 b_{GeSn} + xy(b_{GeSn} + b_{GeSn} - b_{SiSn}) + x(Si - Ge - b_{GeSi}) + y(Sn - Ge - b_{GeSn}) - a_{SiGeSn} + Ge = 0$$

When x is given, a quadratic polynomial of the form $ay^2 + by + c = 0$ remains. This code implements the solution and shows the effect on the resultant compositions that lattice match to germanium in Figure 17.

```
from math import sqrt
Ge = 5.6573; Si = 5.4307; Sn = 6.4892
bSiGe=0.026; bGeSn=-0.041; bSiSn=0    # [45]
bGeSi=bSiGe

def y(x):
        a_SiGeSn = Ge
        a        = bGeSn
        b        = x*(bGeSn+bGeSi-bSiSn) + (Sn-Ge-bGeSn)
        c        = x*(Si-Ge-bGeSi) - a_SiGeSn + Ge
        return (-b + sqrt(b*b-4*a*c))/(2*a)

import numpy
from matplotlib import pyplot

pyplot.rc('xtick',labelsize=8)
pyplot.rc('ytick',labelsize=8)
pyplot.rc('axes', labelsize=8)
pyplot.rc('legend',fontsize=8)
pyplot.figure("b) Si$_x$Ge$_{1-x-y}$Sn$_y$ compositions\nthat lattice match to Ge", figsize=(3,2))
for tick in range(0,11):
    T=10*tick
    pyplot.plot([0,T],[T,0],         linestyle='dotted', linewidth=.2, color='black', zorder=1) # Ge
    pyplot.plot([T,T],[0,100-T],     linestyle='dotted', linewidth=.2, color='black', zorder=1) # Si
    pyplot.plot([0,T],[100-T,100-T], linestyle='dotted', linewidth=.2, color='black', zorder=1) # Sn
pyplot.plot([100*_x for _x in x if y(_x)+_x<1], [100*y(_x) for _x in x if y(_x)+_x<1], zorder=2, \
  'marker='', linewidth=1, color='blue',  label='With bowing')
pyplot.plot(100*x, [100*Vegard_y(_x) for _x in x], zorder=3, marker='', linewidth=1, color='black', \
  label='Vegard\'s Law')
pyplot.legend().set_zorder(4)
pyplot.xlim([0,100]); pyplot.ylim([0,100])
pyplot.xlabel("Si composition $x$ [%]"); pyplot.ylabel("Sn composition $y(x)$ [%]")
pyplot.xticks([r*10 for r in range(0,11)]); pyplot.yticks([r*10 for r in range(0,11)])
pyplot.tight_layout()

pyplot.figure("c) Difference in Ge lattice match composition\ndue to bowing term", figsize=(3,2))
pyplot.plot([100*_x for _x in x if y(_x)+_x<1], [100*(y(_x)-Vegard_y(_x)) for _x in x if y(_x)+_x<1], \
  marker='', linewidth=1, color='green')
pyplot.xlim([0,100]); pyplot.ylim([0,2])
pyplot.xlabel("Si composition $x$ [%]"); pyplot.ylabel("Sn composition\ndifference [%]")
pyplot.xticks([r*10 for r in range(0,11)])
pyplot.tight_layout()

from math import sin,cos
pi = 3.141592653589793
def transform_x(x_list, y_list):
    return [_x + _y*sin(30*pi/180) for _x,_y in zip(x_list, y_list)] # [42]
def transform_y(y_list):
    return [_y*cos(30*pi/180) for _y in y_list] # [42]
def plot_gridline(x,y,**kwargs):
```

```
    return pyplot.plot(transform_x([x[0],x[-1]],[y[0],y[-1]]), transform_y([y[0],y[-1]]), color='black',
**kwargs)[0]
def plot_grid_triangle(**kwargs):
    for tick_level in [10*r for r in range(1,11)]:
        plot_tick_triangle(tick_level, linewidth=.1, **kwargs)
    plot_tick_triangle(0, linewidth=1, **kwargs)
def plot_tick_triangle(T, **kwargs):
    return [ \
        plot_gridline([0,100-T],[100-T,0], **kwargs),\
        plot_gridline([T,T],[0,100-T],     **kwargs),\
        plot_gridline([0,100-T],[T,T],     **kwargs),\
    ]
def plot_tick_marks(**kwargs):
    slope=cos(30*pi/180)/.5
    length=3
    for T in [10*r for r in range(0,11)]:
        plot_gridline([T,T], [0,-length], linewidth=.5)
        plot_gridline([T,T+length], [100-T,100-T], linewidth=.5)
        plot_gridline([0,-length/sqrt(2)], [T,T+length/sqrt(2)], linewidth=.5)
pyplot.figure("a) Si$_x$Ge$_{1-x-y}$Sn$_y$ compositions that lattice match to Ge", figsize=(4,4))
plot_grid_triangle(zorder=1)
pyplot.plot(100*numpy.array(transform_x(x,[y(_x) for _x in x if y(_x)+_x<1])), \
  100*numpy.array(transform_y([y(_x) for _x in x if y(_x)+_x<1])), zorder=3, marker='', linewidth=1, \
  color='blue',  label='With bowing')
pyplot.plot(100*numpy.array(transform_x(x,[Vegard_y(_x) for _x in x])), \
  100*numpy.array(transform_y([Vegard_y(_x) for _x in x])), zorder=2, marker='', linewidth=1, \
  color='black', label='Vegard\'s Law')
pyplot.xlabel("Si composition [%]  ( Ge composition [%]  label added after )")
pyplot.ylabel("Sn composition [%] (use triangle grid/ticks)")
pyplot.xticks([r*10 for r in range(0,11)]); pyplot.yticks([r*10 for r in range(0,11)])
plot_tick_marks()
pyplot.legend().set_zorder(4)
pyplot.show()
```

APPENDIX C: SIGESN ELECTRON EFFECTIVE MASS

```
proc _mc_gamma {x y} {
        set  mc_Ge  0.0038  ;# [36] [37]
        set  mc_Si  0.528   ;# [36] [71]
        set  mc_Sn  0.058   ;# [36] [37]
        return [SiGeSn_bowing_formula $x $y $mc_Ge $mc_Si $mc_Sn]
}
proc _mt_L {x y} {
        set  mt_L_Ge  0.0807  ;# [36] [37]
        set  mt_L_Si  0.113   ;# [36] [72]
        set  mt_L_Sn  0.075   ;# [36] [73]
        return [SiGeSn_bowing_formula $x $y $mt_L_Ge $mt_L_Si $mt_L_Sn]
}
proc _ml_L {x y} {
        set  ml_L_Ge  1.57   ;# [36] [37]
        set  ml_L_Si  1.659  ;# [36] [72]
        set  ml_L_Sn  1.478  ;# [36] [73]
        return [SiGeSn_bowing_formula $x $y $ml_L_Ge $ml_L_Si $ml_L_Sn]
}
# On Ioffe (possibly based on Levinshtein) there is ml=1.59m0 (corresponds to mll); mt=0.0815m0
# [Kittel] mentions Ge ml (mll) = 1.59m and mt=0.082m (close)
proc Calculate_electron_DOS_effective_mass {x y} {
        if {[::SiGeSn::Bandgap::is_direct $x $y]} {
                set mt  [_mc_gamma $x $y]
                set mll $mt
                set M   1.0
        } elseif {[::SiGeSn::Bandgap::is_L_indirect $x $y]} {
                set mt  [_mt_L $x $y]
                set mll [_ml_L $x $y]
                set M   4.0 ;# Multiplicity of L indirect states
        } elseif {[::SiGeSn::Bandgap::is_X_indirect $x $y]} {
                set mt  [_mt_X $x $y]
                set mll [_ml_X $x $y]
                set M   6.0 ;# Multiplicity of X indirect states (Si like)
        } else {
                error "SiGeSn bandgap type error"
        }
        return [expr {(pow($M, 2.0/3.0))*(pow($mll*$mt*$mt,1.0/3.0))}]
        # We have neglected higher multiplicity at crossover points
        # where more states are available at the same energy level
}
proc Calculate_electron_DOS {mm} {
        return [expr {2*pow(2.0*$::pi*$::kB*$::temp*$::m0*$mm/($::h*$::h*1.0e4), 3.0/2.0)}]
}
proc init {} {
        variable  Formula  1
        variable  a        0.0
        variable  ml       0.0
        variable  mm

        set  Si_Sn_ratio  3.671227
        set  x            0.14
        set  y            [expr {$x/$Si_Sn_ratio}]

        set mm [Calculate_electron_DOS_effective_mass $x $y]
        variable Nc [Calculate_electron_DOS $mm]
```

APPENDIX D: DIVIDING MULTIBOX MESH CUSTOMIZABLE WITH WORKBENCH VARIABLES

This mesh is configurable with variables called dividers. It draws rectangles over sections of the device, and the small and large mesh spacings are determined by applying the divider inputs. It can lead to discretization error in a large device region (Section 5-1-1). A similar approach involves using `R_divide` to set the upper bound of mesh element sizing with ratio-defined mesh spacing growth instead of bounding it with maximum element size. Documentation in section 4-2.

```
set my_window_counter 0
proc make_window {x1 y1 x2 y2} {
    global my_window_counter
    incr my_window_counter
    puts "(sdedr:define-refeval-window \"win$my_window_counter\" \"Rectangle\" \n\t\
        (position $x1 $y1 0) \n\t\
        (position $x2 $y2 0))"
    return "win$my_window_counter"
}
set my_multibox_counter 0
set my_placement_counter 0
proc make_multibox {window maxX maxY minX minY Xratio Yratio} {
    global my_multibox_counter my_placement_counter
    puts "(sdedr:define-multibox-size \"box$my_multibox_counter\" \n\t\
        $maxX $maxY \n\t$minX $minY \n\t$Xratio $Yratio)"
    puts "(sdedr:define-multibox-placement \
        \"place$my_placement_counter\" \"box$my_multibox_counter\" \"$window\")"
    incr my_multibox_counter
    incr my_placement_counter
}
proc define_directional_multibox {window_name {directions {"right" "left" "down" "up"}} maxX maxY minX minY {Xratio 1.02} {Yratio 1.15}} {
    # The "left" and "right" terms aren't always intuitive
    if {[llength $directions] == 0} {
        make_multibox $window_name $maxX $maxY $maxX $maxY 1.0 1.0
    }
    if {[lsearch -exact $directions "right"] >= 0} {
        make_multibox $window_name $maxX $maxY $minX $maxY -$Xratio 1.0
    }
    if {[lsearch -exact $directions "left"] >= 0} {
        make_multibox $window_name $maxX $maxY $minX $maxY $Xratio 1.0
    }
    if {[lsearch -exact $directions "down"] >= 0} {
        make_multibox $window_name $maxX $maxY $maxX $minY 1.0 -$Yratio
    }
    if {[lsearch -exact $directions "up"] >= 0} {
        make_multibox $window_name $maxX $maxY $maxX $minY 1.0 $Yratio
    }
}
```

```
proc define_mesh_rectangle {x0 x1 y0 y1 {directions {"right" "left" "down" "up"}} {mindivider 100.0} {maxdivider 30.0} {divider_multiplier 1}} {
      set minY      "(/ (- $y1 $y0) (* $mindivider $divider_multiplier))"
      set maxY      "(/ (- $y1 $y0) (* $maxdivider $divider_multiplier))"
      set minX      "(/ (- $x1 $x0) (* $mindivider $divider_multiplier))"
      set maxX      "(/ (- $x1 $x0) (* $maxdivider $divider_multiplier))"
      define_directional_multibox [make_window $x0 $y0 $x1 $y1] $directions $maxX $maxY $minX $minY
}
proc define_layer_mesh {region mindivider maxdivider multiplier} {
      define_mesh_rectangle 0 "half_w_contact" "Y0_$region" "Y1_$region" {"up" "down" "right"} $mindivider $maxdivider $multiplier
      define_mesh_rectangle "half_w_contact" "(* 2 half_w_contact)" "Y0_$region" "Y1_$region" {"up" "down" "left"} $mindivider $maxdivider $multiplier
      define_mesh_rectangle "(* 2 half_w_contact)" "Xmax" "Y0_$region" "Y1_$region" {"up" "down" "left"} $mindivider $maxdivider $multiplier
}
proc define_1d_mesh_rectangle {x0 x1 y0 y1 {directions {"right" "left" "down" "up"}} {mindivider 100.0} {maxdivider 30.0} {divider_multiplier 1}} {
    set minY      "(/ (- $y1 $y0) (* $mindivider $divider_multiplier))"
    set maxY      "(/ (- $y1 $y0) (* $maxdivider $divider_multiplier))"
    set minX      "(/ Xmax 2.0)"
    set maxX      "(/ Xmax 2.0)"
    define_directional_multibox [make_window $x0 $y0 $x1 $y1] $directions $maxX $maxY $minX $minY
}
global epi_regions
set maxdivider [expr 1.0 * @mesh_divisor_1@]
set mindivider [expr 1.0 * @mesh_divisor_2@]
# maxdivider is the divider for the big elements, and mindivider is the divider for the little elements
# maxdivider is less than mindivider, because max refers to the size of the larger elements, and corresponds to the smaller divisor value (LB)
if {"@dimension@" == "1d"} {
    set number_of_tighter_regions 10
    foreach region $epi_regions {
        if {$number_of_tighter_regions > 0} {
            set number_of_tighter_regions [expr $number_of_tighter_regions-1]
            define_1d_mesh_rectangle 0 "Xmax" "Y0_$region" "Y1_$region" {"up" "down"} $mindivider $maxdivider 5
        } else {
            define_1d_mesh_rectangle 0 "Xmax" "Y0_$region" "Y1_$region" {"up" "down"} $mindivider $maxdivider 1
}}} else {
    define_mesh_rectangle 0 "half_w_contact" "(- Y0_sc1_fsf h_cap)" "Y0_sc1_fsf"      {"right" "down"}     $mindivider $maxdivider 2
    define_mesh_rectangle "half_w_contact" "(* 2 half_w_contact)" "Y0_arc1" "Y1_arc1" {"left" "down"}      $mindivider $maxdivider 2
    define_mesh_rectangle "(* 2 half_w_contact)" "Xmax" "Y0_arc1" "Y1_arc1"           {"left" "down"}      $mindivider $maxdivider 2
    define_mesh_rectangle "half_w_contact" "(* 2 half_w_contact)" "Y0_arc2" "Y1_arc2" {"left" "down" "up"} $mindivider $maxdivider 2
    define_mesh_rectangle "(* 2 half_w_contact)" "Xmax" "Y0_arc2" "Y1_arc2"           {"left" "down" "up"} $mindivider $maxdivider 2
    set number_of_tighter_regions 8
    foreach region [my_lremove $epi_regions {arc1 arc2}] {
        if {$number_of_tighter_regions > 0} {
            define_layer_mesh $region $mindivider $maxdivider 5
            set number_of_tighter_regions [expr $number_of_tighter_regions-1]
        } else {
            define_layer_mesh $region $mindivider $maxdivider 1
}}}
```

APPENDIX E: PLOT COORDINATES FROM STANDARD INPUT

Accepts two rows of CSV input via Unix pipe and plots the points.

```
#plot_coordinates_from_stdin.py
import sys
from math import floor, log, ceil

def print_leading_spaces(data_value):
    if data_value < 10:
        print(" ", end="")

def print_with_leading_spaces(data_value, **kwargs):
    print_leading_spaces(data_value)
    print("%2.3f" % data_value, **kwargs)

def print_xy_data_line(x,y):
   print_with_leading_spaces(x, end='')
   print(', ', end="")
   print_with_leading_spaces(y)

x=[]; y=[]
for line in sys.stdin:
    mysplit=line.split(',')
    x.append(float(mysplit[0]))
    y.append(float(mysplit[1]))
    print_xy_data_line((x[-1],y[-1]))

epi_4J = "\ …
# (Epi tables copied as a large strings)

epi          = epi_4J
device_depth = 175.9 #19 # 7j
w_sim        = 120/2
wc_sim       = 3.6/2
cap_Y        = 0.1+0.06-0.1 # To draw cap

from matplotlib import pyplot
from sys import argv

if len(argv) > 1:
   pyplot.figure(' '.join(argv[1:]))
pyplot.xlabel("Across device (x) [$\mu$m]", fontsize=8);
pyplot.ylabel("Into device (y) [$\mu$m]", fontsize=8)
pyplot.plot([0,wc_sim],[cap_Y,cap_Y], linewidth=.1, color='black')
location = 0
for thickness in [float(col[3]) for col in [row.split(',') for row in epi.split('\n')]]:
    pyplot.plot([0,w_sim],[location,location], linewidth=.1, color='black')
    location+=thickness
Ybot=location; del location
pyplot.plot([0,0],           [0,Ybot], linewidth=.1, color='black')
pyplot.plot([wc_sim,wc_sim], [0,Ybot], linewidth=.1, color='black')
pyplot.xticks([r*10 for r in range(0,6)], fontsize=8)
pyplot.gca().set_xticks(range(0,ceil(w_sim)+1), minor=True)
pyplot.yticks([r for r in range(0,ceil(device_depth)+1)], fontsize=8)
pyplot.gca().set_yticks([r/10 for r in range(0,10*ceil(device_depth)+1)], minor=True)
pyplot.scatter(x,y, alpha=0.1)
pyplot.xlim(left=0,right=max(x+[8]))
pyplot.ylim(top=0,bottom=max(y+[1]))
pyplot.tight_layout()
pyplot.show(block=False)
```

APPENDIX F: OBJECT WITH DATA AND CONVERSION FACTORS

The physics object can passed to functions that make plots, reducing the number of arguments to such functions.

```
#device.py in Python directory
class Device(object):
    def __init__(self, w, areaFactor, w_contact, num_segments, regions, contact_list, surfaces):
        self.w_cm              = w/1e4
        self.areaFactor_cm     = areaFactor/1e4
        self.grid_shading      = w_contact/w
        self.grid_shade_factor = 1-w_contact/w
        self.num_segments      = num_segments
        self.regions           = regions
        self.contact_list      = contact_list
        self.surfaces          = surfaces

class Physics(object):
    def __init__(self, device, intensity, sim):
        print("---- device.py Processing electron currents, hole currents, Jsc, EQE, tunnel loss current")
        self.device      = device
        self.intensity   = intensity
        self.sim         = sim
        self.q    = q    = 1.60217662e-19     # C
        self.h    = h    = 6.62607015e-34     # J*s
        self.c    = c    = 299792458          # m/s
        self.hc   = hc   = h*c
        self.wl   = wl   = sim.wavelengths
        self.E    = E    = hc/(wl/1e6)        # [J] array
        self.N_ph = N_ph = intensity/E*device.w_cm/2*device.areaFactor_cm
        self.rate_to_EQE = rate_to_EQE   = 1/N_ph*device.areaFactor_cm
        # Sentaurus Device does not multiply integrated rates by areaFactor, only output currents
        self.I_to_EQE                    = 1/N_ph/q
        self.integration_unit_conversion = 1 # It's 1e-8 if Math(CurrentPlot(IntegrationUnit=um))
        self.surface_rate_to_EQE = rate_to_EQE*self.integration_unit_conversion
        self.eqe_to_iqe          = 1/(1-sim.data["LayerStack(Window1) R_Total"])/device.grid_shade_factor
        self.rate_to_IQE         = rate_to_EQE*self.eqe_to_iqe
        self.I_to_IQE            = self.I_to_EQE   *self.eqe_to_iqe
        self.eCurrent = eCurrent = [sim.data[c + ' eCurrent'] for c in device.contact_list]
        self.hCurrent = hCurrent = [sim.data[c + ' hCurrent'] for c in device.contact_list]
        self.Jsc        = [sum(eCurrent[0:2*i+1]+hCurrent[0:2*i+1]) for i in range(0,device.num_segments)]
        self.eCurrent_EQE = e  = [self.I_to_EQE*e_i  for e_i in eCurrent]
        self.hCurrent_EQE = h  = [self.I_to_EQE*h_i  for h_i in hCurrent]
        self.EQE               = [self.I_to_EQE*Jsci for Jsci in self.Jsc]
        self.tunnel_Jsc_EQE    = [-sum(e[0:2*i]+h[0:2*i]) for i in range(1,device.num_segments)]
        self.hole_loss_EQE     = [-_h for _h in h[0::2]] ;# even-numbered terminals (Stride notation)
        self.electron_loss_EQE = e[1::2]  ;# odd-numbered contacts

#python_pyt.cmd
from Python.device import Device, Physics
def get_physics(sim):
    return Physics(
        Device(
            @w@,
            @areaFactor@,
            @w_contact@,
            !(puts -nonewline $num_segments)!,
            "!(puts -nonewline $epi_regions)!".split(),
            "!(puts -nonewline $contact_list)!".split(),
            "!(puts -nonewline $epi_inner_subcell_surfaces)!".split()),
        @intensity@,
        sim
    )
```

APPENDIX G: MERCURIAL REPOSITORY IGNORE FILE .HGIGNORE

The `.hgignore` file is used such that only source files, and no node, preprocessed, or other output files are visible from the Mercurial GUI TortoiseHg. Workbench project files `.database` and `gtree.dat` files are updated automatically by the workbench, and these are committed and pushed routinely.

```
syntax: glob
#*.pdf
#*.prf
*.tdr
*.msh
*.bnd
*.out
*.log
*.pyc
*.pyo
*.so.*
pp*_*
n*_*
*.log.BAK
# Sentaurus Workbench Manual Appendix C
#gtree.dat
gtree.dat.backup
gvars.dat
#gtooldb.tcl
glog.txt
greadme.txt
#greadme.pdf
preprocessor.log
gexec.cmd
#gopt.cmd
gopt.log
#gqueues.dat
gsummary.txt
#gcomments.dat
#runlimits.txt
#.project
.status
.history
#.database
usrdefvars.dat
cmlog.txt
gsummary.txt
gsub.err # (Manual Section 11 Schedulers) "gsub: For every project you run, Sentaurus Workbench launches
the gsub process that manages project scheduling. ""
*~ # All files ending in ~ (left behind by editor)
scrap # Directory
```

APPENDIX H: AUTOMATIC SCALE AND GRID WITH MATPLOTLIB

```
from matplotlib import pyplot
from math import log10, floor, ceil

def do_ticks_current_figure():
    do_y_ticks_current_figure()
    do_x_ticks_current_figure()

def do_y_ticks_current_figure():
    do_y_ticks(
        min([min(L.get_ydata()) for L in pyplot.gca().get_lines()]),
        max([max(L.get_ydata()) for L in pyplot.gca().get_lines()]))
def do_x_ticks_current_figure():
    do_x_ticks(
        min([min(L.get_xdata()) for L in pyplot.gca().get_lines()]),
        max([max(L.get_xdata()) for L in pyplot.gca().get_lines()]))

def do_y_ticks(ymin,ymax):
    do_ticks(ymin,ymax,pyplot.gca().set_yticks, pyplot.gca().set_ylim)
def do_x_ticks(xmin,xmax):
    do_ticks(xmin,xmax,pyplot.gca().set_xticks, pyplot.gca().set_xlim)

def do_ticks(amin, amax, set_tick_function, set_limits_function, amount=‘’):
    print('do_ticks amin and amax are %f and %f' % (amin,amax))
    try:
        if amount=='more’:
            order = floor(log10(amax-amin))
        elif amount=='less’:
            order = ceil(log10(amax-amin))
        else:
            order = round(log10(amax-amin))
    except ValueError as e:
        print("do_ticks() failed, returning, maybe range too small, error printed:")
        print(str(e))
        return
    major_step   = pow(10,order-1)
    minor_step   = pow(10,order-2)
    bottom       = floor(amin/major_step)*major_step
    bottom_minor = floor(amin/minor_step)*minor_step
    set_tick_function(
        [r*major_step+bottom for r in range(0,ceil((amax-amin)/major_step)+1)])
    set_tick_function(
        [r*minor_step+bottom_minor for r in range(0,ceil((amax-amin)/minor_step)+1)],
        minor=True)
    set_limits_function(amin,amax)

def do_grid():
    pyplot.grid(which='minor',color='#eaeaea', zorder=0)
    pyplot.grid(which='major',color='#d4d4d4', zorder=0)
```

APPENDIX I: PLOTTING ENTIRE DATASET TO INSPECT DISCRETIZATION

```
proc everything_plot {} {
    create_variable -dataset Normalized -name Y -function "<Y:structure_cutline>"
    plot_list_vs_y "Everything_plot" {
        "OpticalGeneration" "ElectrostaticPotential" "ElectricField-Y" "eCurrentDensity-Y" "hCurrentDensity-Y" "TotalCurrentDensity-Y"\
        "RadiativeRecombination" "srhRecombination" "AugerRecombination" "SurfaceRecombination" "TotalRecombination" \
        "ConductionBandEnergy" "ValenceBandEnergy" "SpaceCharge" "eDensity" "hDensity" "eVelocity" "hVelocity"\
        "eQuasiFermiEnergy" "hQuasiFermiEnergy "eQuasiFermiPotential" "hQuasiFermiPotential"\
    }
}
proc plot_list_vs_y {name data_variable_list} {
    create_plot -1d -dataset "Normalized" -name "$name"
    set_axis_prop -axis x -title "Depth (Y) \[<greek>m</greek>m\]" -title_font_size 8 -ticks_position out\
        -min 0 -min_fixed -anchor 0 -max [lindex [get_variable_data -dataset structure_cutline Y] end] -max_fixed\
        -auto_precision -scale_format fixed -scale_font_size 8 -auto_spacing -nof_minor_ticks 10
    set_axis_prop -axis y -min -1 -max 1 -min_fixed -max_fixed -ticks_position out -hide_title  -title ""\
        -manual_spacing -spacing 1 -nof_minor_ticks 10 -anchor 0 -ticks_position out \
        -hide_scale -manual_precision -scale_precision 1 -scale_font_size 8
    set_legend_prop -label_font_family courier -label_font_size 6 -location "bottom_right"
    set_plot_prop -show_grid -title_font_size 8 -hide_title
    set_grid_prop -line1_color "#f4f4f4" -line1_style "solid" -line2_color "#f9f9f9" -line2_style "solid"
    foreach var $data_variable_list {
        plot_normalized $var
    }
}
proc plot_normalized {var} {
    set maxabs [max_abs $var]
    create_variable_function_and_curve Normalized $var "<$var:structure_cutline>/$maxabs" "[format %.0e $maxabs] "
}
proc create_variable_function_and_curve {dataset name function {label_tag ""}} {
    create_variable -dataset $dataset -name $name -function $function
    create_curve    -dataset $dataset -name $name -axisX "Y" -axisY $name
    set_curve_prop $name -label "$label_tag$name" -show_markers -markers_type circle -markers_size 3 -show_line
}
proc max_abs {data_variable} {
    ext::ExtractExtremum -type max -name noprint -out themax -x Y -y [get_variable_data -dataset structure_cutline $data_variable]
    ext::ExtractExtremum -type min -name noprint -out themin -x Y -y [get_variable_data -dataset structure_cutline $data_variable]
    ext::ExtractExtremum -type max -name noprint -out maxabs -x [list 1 2] -y [list [expr abs($themax)] [expr abs($themin)]]
    return $maxabs
}
set sdevice n@node|ramp@_des
load_file ${sdevice}.tdr
create_plot -name structure_plot -dataset ${sdevice}
set xcut [expr 0.5*@w@/2.0]
create_cutline -name structure_cutline -plot structure_plot -type free -points {$xcut 0 $xcut 180}
everything_plot
```

APPENDIX J: TCL INTERFACE TO SDE SCHEME FOR 1D MESH REFINEMENT

```
(sdeaxisaligned:set-parameters "maxAspectRatio" 1e99)
(sdedr:define-refinement-size "my_global_refdef" (- Xmax Xmin) (- Ybot Ytop) (- Xmax Xmin) (- Ybot Ytop))
(sdedr:define-refinement-placement "my_global_placement" "my_global_refdef" (list "window" "Ref_global"))
!(
    proc interface_mesh_function {region1 region2 minelement ratio {DoubleSide False}} {
       if {$DoubleSide=="DoubleSide"} {
          puts "(sdedr:define-refinement-function \"my_global_refdef\" \"MaxLenInt\" \"$region1\" \"$region2\" $minelement $ratio \"UseRegionNames\" \"DoubleSide\")"
       } else {
          puts "(sdedr:define-refinement-function \"my_global_refdef\" \"MaxLenInt\" \"$region1\" \"$region2\" $minelement $ratio \"UseRegionNames\")"
       }
    }
    set my_refdef_counter 0
    set my_refplace_counter 0
    proc make_mesh_refinement_definition {window dxmax dymax dxmin dymin} {
        global my_refdef_counter my_refplace_counter
        puts "(sdedr:define-refinement-size \"refdef$my_refdef_counter\" \n\t$dxmax $dymax $dxmin $dymin)"
        puts "(sdedr:define-refinement-placement \"refplace$my_refplace_counter\" \"refdef$my_refdef_counter\" \"$window\")"
        incr my_refdef_counter
        incr my_refplace_counter
    }
    proc y_refdef {y0 y1 {max_element_size 0.0001}} {
        make_mesh_refinement_definition [make_window "Xmin" $y0 "Xmax" $y1] "(- Xmax Xmin)" $max_element_size "(- Xmax Xmin)" $max_element_size
    }
    proc y_thickness_refdef {start T el} {
        if {$T>0} {
            y_refdef $start "(+ $start $T)" $el
        } else {
            y_refdef "(+ $start $T)" $start $el
        }
    }
    proc y_region_refdef {region el} {
        y_refdef "Y0_$region" "Y1_$region" $el
    }
    proc y_multibox {y0 y1 {element 0.0001} {ratios {1.2 -1.2}} {maxelement "(- Ybot Ytop)"}} {
        foreach R $ratios {
            make_multibox [make_window "Xmin" $y0 "Xmax" $y1] 0 $maxelement 0 $element 0 $R
        }
    }
    proc y_region_multibox {region minelement ratios {maxelement "(- Ybot Ytop)"}} {
        y_multibox "Y0_$region" "Y1_$region" $minelement $ratios $maxelement
    }
    proc R_divide_list {Rs divider} {
        set R_divided_list [list]
        foreach R $Rs {
            lappend R_divided_list [R_divide $R $divider]
        }
        return $R_divided_list
    }
)!
```

APPENDIX K: ONE-DIMENSIONAL MESH FOR LOW ERROR

```
proc quintuple_y_refdef {y0 y1 d1 el1 d2 el2 el3 b1 b2} {
    set inside1 "(+ $y0 $d1)"
    set inside2 "(+ $y0 $d2)"
    set inside3 "(- $y1 $b1)"
    set inside4 "(- $y1 $b2)"
    y_refdef $y0      $inside1 $el1
    y_refdef $inside1 $inside2 $el2
    y_refdef $inside2 $inside3 $el3
    y_refdef $inside3 $inside4 $el2
    y_refdef $inside4 $y1      $el1
}
proc region_quintuple_y_refdef {region d1 el1 d2 el2 el3 b1 b2} {
    quintuple_y_refdef "Y0_$region" "Y1_$region" $d1 $el1 $d2 $el2 $el3 $b1 $b2
}
proc manual_3box {y0 y1 el1} {
    puts "(define _step (/ (- $y1 $y0) 3.0))"
    y_refdef $y0 $y1 [expr 2.0*$el1]
    y_thickness_refdef $y0 "_step"  $el1
    y_thickness_refdef $y1 "(- 0 _step)" $el1
}
proc region_3box {region {el1 0.0001}} {
    manual_3box "Y0_$region" "Y1_$region" $el1
}
proc mesh_layer {region {el1 0.0001} {max 0.002}} {
    y_region_multibox $region $el1 {1.2 -1.2} $max
}
proc mesh_em_base_surfaces {em_region base_region} {
    y_multibox "Y0_$em_region"   "Y1_$em_region"   0.0001 {1.2 -1.2} ;# surfaces
    y_multibox "Y0_$base_region" "Y1_$base_region" 0.0001 {1.2 -1.2} ;# surfaces
    y_thickness_refdef "Y1_$em_region" -0.007 0.0003 ;# general depletion region
    y_thickness_refdef "Y1_$em_region"  0.035 0.0003
    y_thickness_refdef "Y1_$em_region"  0.045 0.0006
}
proc low_error_1D_mesh {} {
    global num_segments
    region_quintuple_y_refdef "arc1" 0.002 0.0001 0.004 0.0002 0.0004 0.004 0.002
    region_quintuple_y_refdef "arc2" 0.002 0.0001 0.004 0.0002 0.0004 0.004 0.002
    for {set i 1} {$i <= $num_segments} {incr i} {
        mesh_layer "sc${i}_fsf" 0.0001 0.001
        mesh_layer "sc${i}_bsf"
        mesh_em_base_surfaces "sc${i}_em" "sc${i}_base"
    }
    for {set i 1} {$i < $num_segments} {incr i} {
        region_3box "td${i}_phigh" 0.0001
        region_3box "td${i}_nhigh" 0.0001
    }
    if {"@epi_label@" == "full_epi"} {
        y_multibox "Y0_sc1_em" "Y1_sc2_base" 0.001 {1.003}
        y_multibox "Y0_sc3_em" "Y1_sc4_base" 0.001 {1.003}
        y_multibox "Y0_sc5_em" "Y1_sc6_base" 0.001 {1.003}
        y_multibox "Y0_sc7_em" "Y1_sc7_base" 0.001 {1.006}
    } else {
        y_multibox "Y0_sc1_em" "Y1_sc1_base" 0.001 {1.003}
        y_multibox "Y0_sc2_em" "Y1_sc2_base" 0.001 {1.003}
        y_multibox "Y0_sc3_em" "Y1_sc3_base" 0.001 {1.003}
        y_multibox "Y0_sc4_em" "Y1_sc4_base" 0.001 {1.006}
    }
}
low_error_1D_mesh
```

APPENDIX L: GENERAL 2D MESH REFINEMENT

```
# First, more tools to define mesh, used by 2D_mesh_many_devices, described in section 5-4-2.

proc triple_y_refdef {y0 y1 d1 d2 el1 el2} {
    set inside1 "(+ $y0 $d1)"
    set inside2 "(- $y1 $d2)"
    y_refdef $y0      $inside1 $el1
    y_refdef $inside1 $inside2 $el2
    y_refdef $inside2 $y1      $el1
}
proc triple_y_refdef_with_distance_divider {y0 y1 dd1 dd2 el1 el2} {
    triple_y_refdef $y0 $y1 "(/ (- $y1 $y0) $dd1)" "(/ (- $y1 $y0) $dd2)" $el1 $el2
}
proc region_triple_y_refdef_with_distance_divider {region dd1 dd2 el1 el2} {
    triple_y_refdef_with_distance_divider "Y0_$region" "Y1_$region" $dd1 $dd2 $el1 $el2
}
proc y_region_list_refdef {region_list element} {
    foreach region $region_list {
        y_region_refdef $region $element
    }
}
proc make_multibox_with_ratio_lists {window maxX maxY minX minY {Xratios {}} {Yratios {}}} {
    foreach xr $Xratios {
        make_multibox $window $maxX $maxY $minX $minY $xr 0
    }
    foreach yr $Yratios {
        make_multibox $window $maxX $maxY $minX $minY 0 $yr
    }
}
proc make_x_multibox_with_ratio_lists {window minel ratios {maxel "(- Xmax Xmin)"}}  {
    make_multibox_with_ratio_lists $window $maxel "(- Ybot Ytop)" $minel "(- Ybot Ytop)" $ratios {}
}
proc make_x_line_x_multibox {y x0 x1 minel ratios {maxel "(- Xmax Xmin)"}} {
    make_x_multibox_with_ratio_lists [make_line $x0 $y $x1 $y] $minel $ratios $maxel
}
proc make_x_line_x_multibox_under_contact {y minel ratios {maxel "(- Xmax Xmin)"}} {
    make_x_line_x_multibox $y "Xmin" "(+ Xmin half_w_contact)" $minel $ratios $maxel
}
proc make_x_line_x_multibox_under_sun {y minel ratios {maxel "(- Xmax Xmin)"}} {
    make_x_line_x_multibox $y "(+ Xmin half_w_contact)" "Xmax" $minel $ratios $maxel
}
proc x_line_multiboxes_with_ylist {ylist minel R {maxel "(- Xmax Xmin)"}} {
    foreach y $ylist {
        make_x_line_x_multibox_under_contact $y $minel -$R $maxel
        make_x_line_x_multibox_under_sun     $y $minel $R $maxel
    }
}
```

```
proc 2D_mesh_many_devices {} {
    #region_quintuple_y_refdef   {region   d1   el1    d2   el2   el3    b1    b2 }
    arc_region_quintuple_y_refdef "arc1" 0.002 0.001 0.004 0.002 0.008 0.004 0.002
    arc_region_quintuple_y_refdef "arc2" 0.002 0.001 0.004 0.002 0.008 0.004 0.002
    interface_mesh_function "arc2" "sc1_fsf" 0.001 1.1 "DoubleSide"
    ;# This spot could use the 1 nm element (small numerical discrepancies)
    global num_segments
    for {set i 1} {$i <= $num_segments} {incr i} {
        y_region_refdef   "sc${i}_fsf"  0.002
        y_region_refdef   "sc${i}_em"   0.002
        if {"@epi_label@" == "3j" && $i == 3} {
            y_multibox "Y0_sc3_base" "Y1_sc3_bsf" 0.002 1.02 ;# x1.06 has ~1.5%
        } else {
            y_multibox "Y0_sc${i}_base" "Y1_sc${i}_bsf" 0.002 1.06 ;# x1.1 has ~0.2% error, x1.05 has 0.05%
        }
        y_region_multibox "sc${i}_base" 0.002 -2
        y_region_multibox "sc${i}_bsf"  0.002 {1.2 -1.2}
    }
    for {set i 1} {$i < $num_segments} {incr i} {
        if {("@epi_label@" == "full_epi" || "@epi_label@" == "6j") && $i == 1} {
            y_region_list_refdef {"td1_phigh" "td1_nhigh"} 0.001
        } else {
            region_triple_y_refdef_with_distance_divider "td${i}_phigh" 3.0 3.0 0.001 0.003
            region_triple_y_refdef_with_distance_divider "td${i}_nhigh" 3.0 3.0 0.001 0.003
        }
    }
    if {"@dimension@" == "2d"} {
        make_multibox_with_ratio_lists [make_window "Xmin" "(- Y0_sc1_fsf h_cap)" "(+ Xmin half_w_contact)" "Y0_sc1_fsf"] 1 1 0.002 0.002 {-1.2} {1.2 -1.2}
        x_line_multiboxes_with_ylist {"Y0_sc1_fsf" "Y1_sc1_fsf" "Y1_sc1_em"} 0.002 1.2
        global num_segments
        for {set i 1} {$i <= $num_segments} {incr i} {
            x_line_multiboxes_with_ylist [list "Y1_sc${i}_fsf" "Y1_sc${i}_em" "Y1_sc${i}_base" "Y1_sc${i}_bsf"] 0.002 1.2 10
        }
        for {set i 1} {$i < $num_segments} {incr i} {
            x_line_multiboxes_with_ylist [list "Y1_td${i}_phigh" "Y1_td${i}_nhigh"] 0.002 1.2 10
        }
        x_multibox_for_shadow "Ytop" "Ybot" 0.002 {1.4}
        ;# Reduced from 1.5 since some corners were visible in the contour plots
    }
}
```