Simulation of Iron Impurity Gettering in Crystalline Silicon Solar Cells

by

Douglas M. Powell

B.S. Mechanical Engineering
The Ohio State University, 2008

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author.....
Department of Mechanical Engineering
June 1, 2012

Certified by.....
Tonio Buonassisi
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Accepted by.....
David E. Hardt
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Theses

Simulation of Iron Impurity Gettering in Crystalline Silicon Solar Cells

by

Douglas M. Powell

Submitted to the Department of Mechanical Engineering

on June 1, 2012 in Partial Fulfillment of the

Requirements for the Degree of Master of Science in

Mechanical Engineering

ABSTRACT

This work discusses the Impurity-to-Efficiency (I2E) simulation tool and applet. The I2E simulator models the physics of iron impurity gettering in silicon solar cells during high temperature processing. The tool also includes a device simulator to calculate cell performance after processing. By linking input materials, processing, and cell performance, I2E enables accelerated solar cell optimization. Herein, background information on the economic drivers of solar cell installations and manufacturing are used to introduce the importance of iron impurity engineering. The fundamental physics of gettering and the development of the numerical methods employed by the tool are presented. The development, deployment, and use of the web applet are also discussed.

Thesis Supervisor: Tonio Buonassisi

Title: Assistant Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

It has been a remarkable journey from digging holes in the yard and tinkering in the garage to where I am today. I believe that what we are able achieve in life is largely a function of the opportunities presented to us and the encouragement provided by those around us. I am very blessed on both of these dimensions, and have a lot to be thankful for. First, I am thankful to God who has enabled me to live a life with nothing to be afraid of. I hope to live the example of love that Jesus has provided.

I am thankful for Ellen, your steadfast love recharges me daily. I am so grateful for your patience, courage, flexibility, and humor. You are so supportive. I am thankful for my Mom, you have nurtured me throughout the years. I was always quite a handful, and think you did a great job. I am thankful for my Dad, and to quote Toby Keith, you have dirty hands and a clean soul. You have taught me how to work hard and a mechanical intuition that I use every day. Finally, I am thankful for Joe, you have always been supportive of me. All of your proof-reading and grammatical consultations throughout the years have been a great service.

I have had excellent experiences in my academic/professional life as well. I thank Tonio for deciding to hire me and introducing me to a new field. Your passion for PV and the lab is amazing. I also thank Larry Willey and the team at GE for teaching me to take ownership of my work and expanding my thinking to the systems level. I thank Seppo Korpela and Gary Kinzel at OSU for introducing me to the role that engineers can serve in solving humanity's problems. I finally thank all my teachers and professors throughout the years. I stand on their shoulders.

I have wonderful colleagues. I thank for Sergio for putting up with my hording of junk in our cubical and always being able to lighten the mood. I thank David for teaching me pretty much everything I know about iron in silicon and generally being my go-to for any help I need in the lab. I also thank Jasmin for the development of I2E and Brianna Conrad for making the plotting functionality for the applet. I thank the authors of the Techpath and PVI papers, especially Mark, for introducing me to writing journal publications. Some of your words and input have inevitably made it in to this document. I thank Steve for the formatted word document that served as a template for this document. I finally thank all of the students and staff of the PV lab, fellow 35-135 and LMP students, the LMP staff, the MechE graduate office, my qualifying exam study groups, and anyone else who has impacted my life for the better.

CONTENTS

FIGURES

CHAPTER

1

MOTIVATION

1.1 Energy Sources in the United States

The primary energy consumption of the United States is predominantly fueled by petroleum, natural gas, and coal with less than 10% sourced from renewable resources.[1] The nation's use of fossil fuels harms the environment, naturally leads to energy shortages, and requires significant imports. The observation of anthropogenic global warming has reached scientific consensus and has been linked to the consumption of fossil fuels.[2, 3] Furthermore, the reliance on fossil fuels exposes the nation, and world, to a great risk of energy shortages upon inevitable (fossil fuels are a finite resource) reductions in well and mine capacity.[4] In petroleum liquids for example, the volume of 2P (proven plus probable reserves) conventional oil discovered has been less than that consumed since 1980.[5] Lastly, the United States imports approximately 30% of its primary energy consumption.[1] Renewable sources of energy can break free of these challenges[6] and could enable further expansions in energy accessibility throughout the developing world.

Electricity currently makes up more than 40% of the primary energy consumed in the United States[1] and is a prime candidate for renewable resource deployment because of its relatively low requirements for energy density. For renewable sources to achieve a significant fraction of generation capacity on the electrical grid, the focus of technologists and policymakers should be

to reduce the costs of disruptive renewable energy technology[7] rather than rely on regulatory measures that increase the costs of non-renewable technology.[8]

1.2 Unique Advantages of Silicon PV

Electricity from renewable energy technologies directly address the challenges discussed above, but solar photovoltaics (PV) benefit from the additional advantages of resource abundance and ease of implementation in a distributed grid. The solar energy resource is effectively unlimited, as the amount of capturable solar energy exceeds worldwide use by a factor of well over 1,000.[9] PV modules are relatively lightweight, benefit from standard racking and installation systems, and do not interact with their local environment. This enables the PV generation to be deployed on the electrical grid in both a distributed fashion at consumption points and in a traditional fashion at large scale generation facilities. The establishment of a distributed PV generation network reduces sensitivity to local weather fluctuations, which improves the consistency of power production.[10] Despite these advantages, the production of solar electricity within the renewable energy segment is far less than wind, wood, biological waste, and geo-thermal[1, 11, 12] because of its prohibitively high cost.[13, 14]

A variety of material systems, including crystalline silicon (c-Si) and several thin-film materials, can be used to fabricate a PV device. However, c-Si currently dominates the PV industry with a market share of 87%.[15] This success is largely influenced by advantageous material properties of silicon (Figure 1.1) and the long development history of the technology.[16] However, the current cost of manufacturing c-Si PV is greater than levels predicted to enable subsidy-free adoption (1.3.2.1).[16]

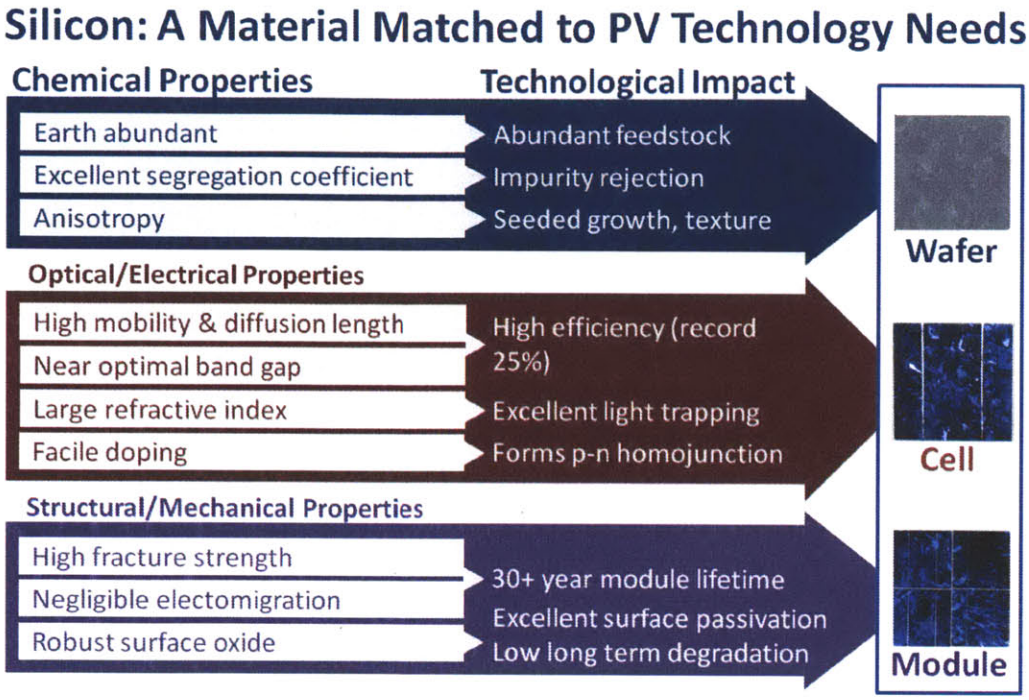

Figure 1.1: c-Si Material Advantages

Crystalline silicon's material properties align with the needs of PV. Image credit of mc-Si wafer: S. Castellanos. Figure reprinted from reference.[16]

1.3 Solar Economics

1.3.1 Historical Module Prices and Reduction Opportunities

Both in terms of price and scale, PV and traditional energy sources are on a convergence course. PV contributes a growing portion of new US electricity installations,[12, 13, 17] though the total electricity contribution to the grid in 2010 was less than 0.1% of total consumption.[1] The price of PV (Figure 1.2) modules in US$/$W_p$, US dollars per peak DC watt at the module level, has fallen significantly over time in constant 2011 dollars.[18, 19]

However, the levelized cost of electricity (LCOE)[20] of a PV installation is the true measure of grid competitiveness. In some niche markets, grid-competitive LCOEs have already been reached with solar installations, but the U.S. Department of Energy (DOE) SunShot program targets widespread subsidy-free utility scale adoption. The DOE estimates that subsidy free adoption can be achieved with module prices[a] of US\$0.50/W_p by 2020 (Figure 1.2).[21] New innovations that reduce cost are required to simply maintain the significant price decline that the industry has managed to demonstrate, but the SunShot target necessitates the deployment of advanced concepts.[22] The DOE estimates that approximately 25% of electricity would be generated by PV in the U.S. by 2050 if the SunShot price targets are achieved.[21]

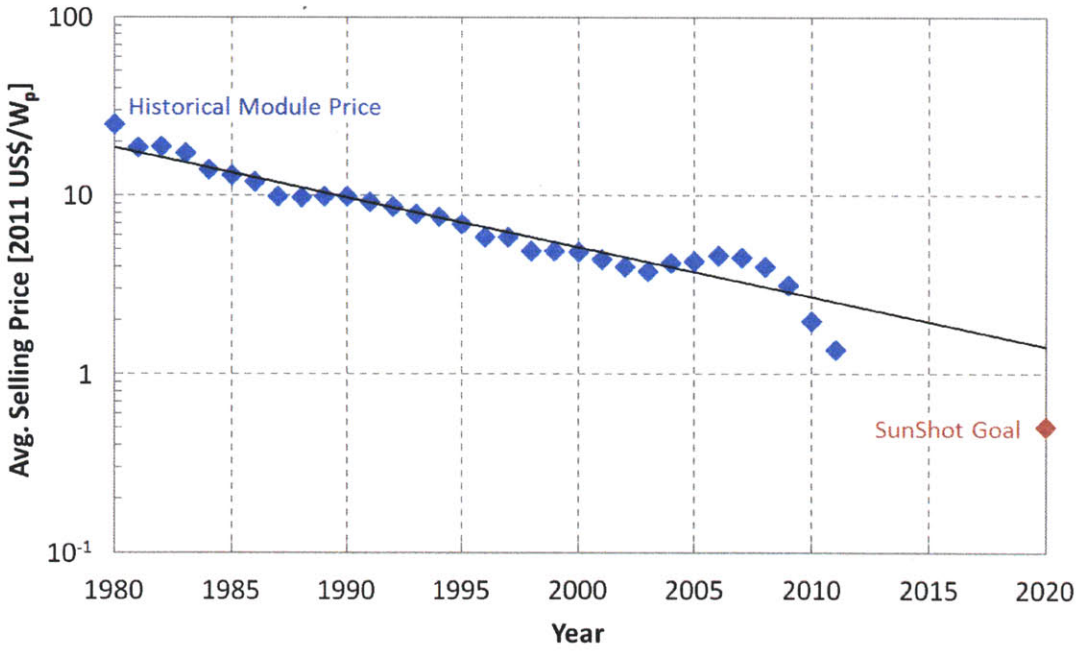

Figure 1.2: Module Price Learning Curve

Module prices have exponentially decreased over time and have generally followed a historical trend. Plot generated with data from references.[18, 19, 21]

---

[a] Note that US\$0.50/W_p is a module price target. Module manufacturing costs of approximately US\$0.40/W_p are required to sustainably support US\$0.50/W_p prices.

1.3.2 Bottom-up Cost Model for c-Si Module Manufacturing

Cost modeling is a critical tool for assessing the commercial impacts of new technology. A bottom-up cost model was developed to assess the current costs of manufacturing c-Si PV in the United States, as well as the cost reduction potential available by 2020 of both standard and advanced technology (Figure 1.3).[16] The model is freely available from the MIT Photovoltaic Research Laboratory, at http://pv.mit.edu/TMA.[16]

1.3.2.1 Cost Estimates for 2012 and 2020

The model estimated manufacturing cost for standard mc-Si technology in 2012[b] at 1.29 US$/W_p.[16] Line-of-sight innovations that are currently being fully developed in industry are estimated by the model to yield domestic manufacturing costs of 0.89 US$/W_p by 2020.[16] This rate of cost reduction is not sufficient to achieve the SunShot target.[16, 21] An advanced concept consisting of high-impact innovations that improve efficiency, silicon utilization, and manufacturing is estimated to provide domestic manufacturing costs of 0.52 US$/W_p by 2020.[16] This result showed that c-Si photovoltaics have the potential to reach the SunShot target, and that a variety of technology pathways exist to enable the cost structure of the advanced concept.[16]

[b] I note that the actual module manufacturing costs in 2012 were below (~US$1.00 – 1.10/W_p) those estimated by the model in the close of 2011. The 2020 line-of-site estimation is also expected to reduce with the inclusion of additional market changes observed in late 2011 and early 2012.

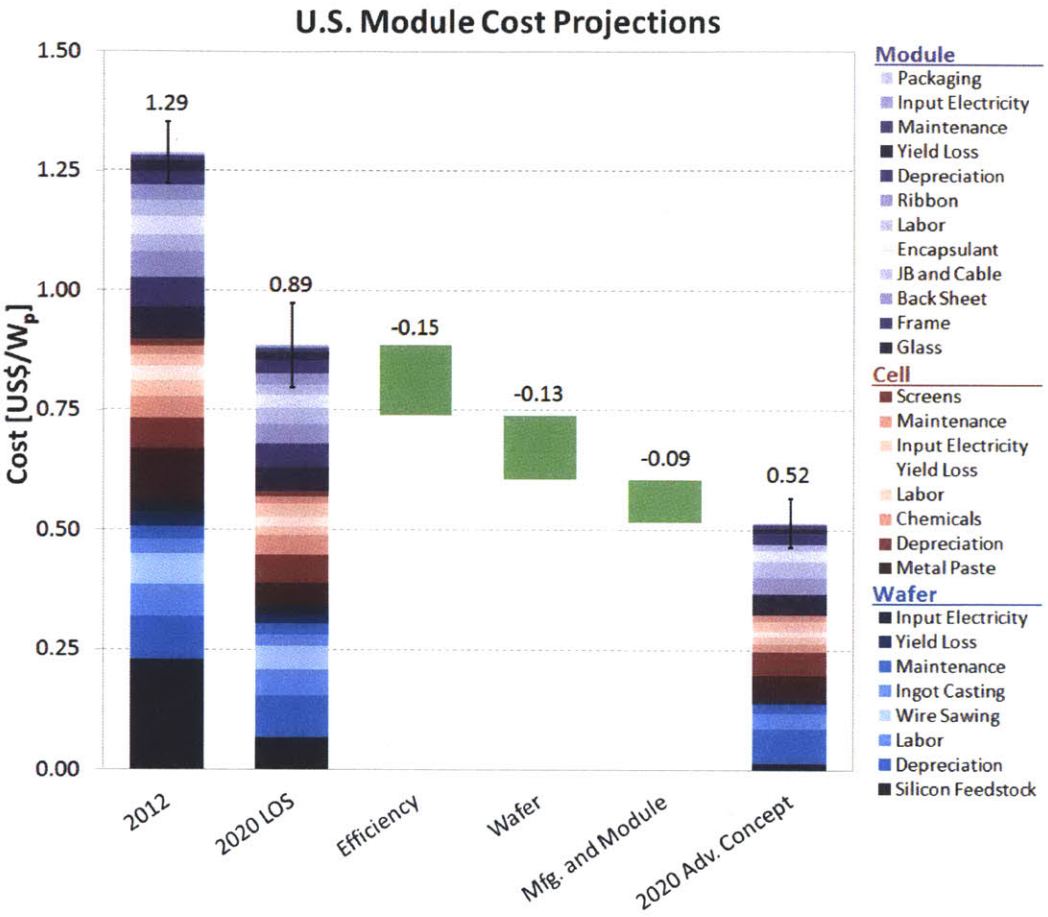

Figure 1.3: Bottom-Up Cost Estimates of US PV Module Manufacturing

Estimated cost structures of technology scenarios. The innovations in the green waterfalls provide significant opportunities for R&D. Figure reprinted from reference.[16]

1.3.2.2 Areas of Greatest Research Impact

The model was used to conduct a sensitivity analysis of the current mc-Si manufacturing cost structure in order to identify areas with greatest research impact.[16] A manufacturing cost sensitivity map was generated that assess both the sensitivity (% module cost change / % variable change) and the maximum theoretical cost savings available from a particular input variable (Figure 1.4).[16] Module efficiency has the largest impact on cost, as it affects all cost categories that scale with area.[16] Unsurprisingly, efficiency provides the largest cost saving category of the advanced technology concept (Figure 1.3) and is a valuable research direction.

The redistribution or mitigation of harmful defects in the solar cell, known as defect engineering,[23] has the potential to improve efficiency (2.2). This has motivated investigations into defect engineering methods (2.1.1) for crystalline silicon solar cells as discussed herein.

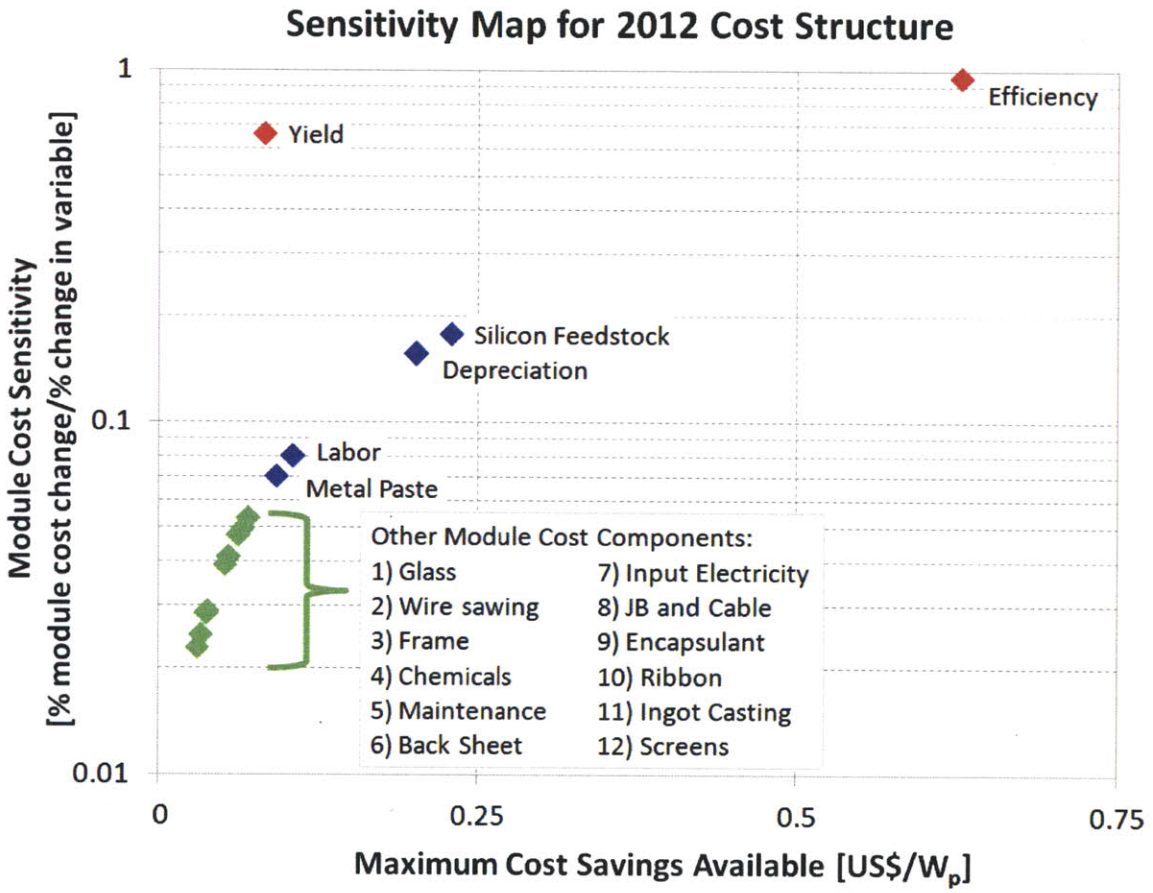

Figure 1.4: Sensitivity of 2012 US c-Si Cost Structure

Efficiency and manufacturing yield have the largest impact on module cost. Figure reprinted from reference.[16]

CHAPTER

2

SOLAR CELL OPERATION AND DEFECT ENGINEERING

2.1 Fundamentals of Solar Cell Operation

A schematic illustration of a traditional c-Si solar cell, as used in the model (1.3.2), is shown in Figure 2.1 below.[24] A 180 μm *p*-type silicon wafer forms the bulk of the device.[25] The wafer is textured to improve light-trapping and then doped with phosphorus to form an *n*-type emitter with a thickness of approximately 0.1 μm (Figure 3.2). An anti-reflective coating is applied to the front of the solar cell to limit light reflection.[26] Front contacts are screen printed onto the device with silver paste with an optimized spacing to reduce resistive losses while minimizing shading.[27] The rear contacts are then printed from aluminum onto the device to collect charge and a low-recombination back surface field is formed.[28, 29] A DC load circuit, *e.g.* batteries, inverter, joins the front and read contacts.

In a actual PV module, solar cells are connected by metal ribbon.[30] The strings of cells are encapsulated between ethylene vinyl acetate (EVA), which is covered by glass on the front side and a polymer back sheet on the rear.[30] The glass-EVA-cell-EVA-back sheet laminate is then surrounded by an aluminum frame for handling and installation, and a junction box is applied to allow the wiring of the module in an installation.[30]

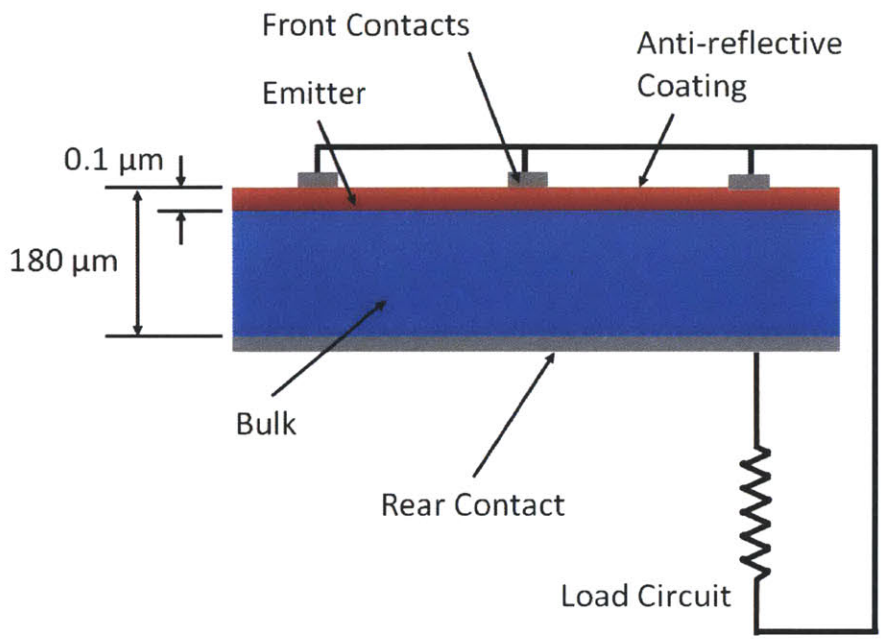

Figure 2.1: Solar Cell Schematic Illustration

Representative structure of solar cell and load circuit.

During the operation of the typical device described above, photons, with an energy equal to or greater than the bandgap, excite electrons from the valence band to the conduction band of the silicon and create an electron-hole pair (Figure 2.2).[31] Incoming photons with energy less than the bandgap are not absorbed and lost, while the excess energy above the bandgap for higher energy photons is lost as heat. [31] The electron-hole pair is separated by the electrical field in the junction between the emitter and base, with electrons flowing towards the front contact and holes towards the rear contact.[32] The electrons flow out of the front contact and contribute to useful current through the load circuit. Upon returning from the load, electrons recombine with the holes at the rear contact.[32]

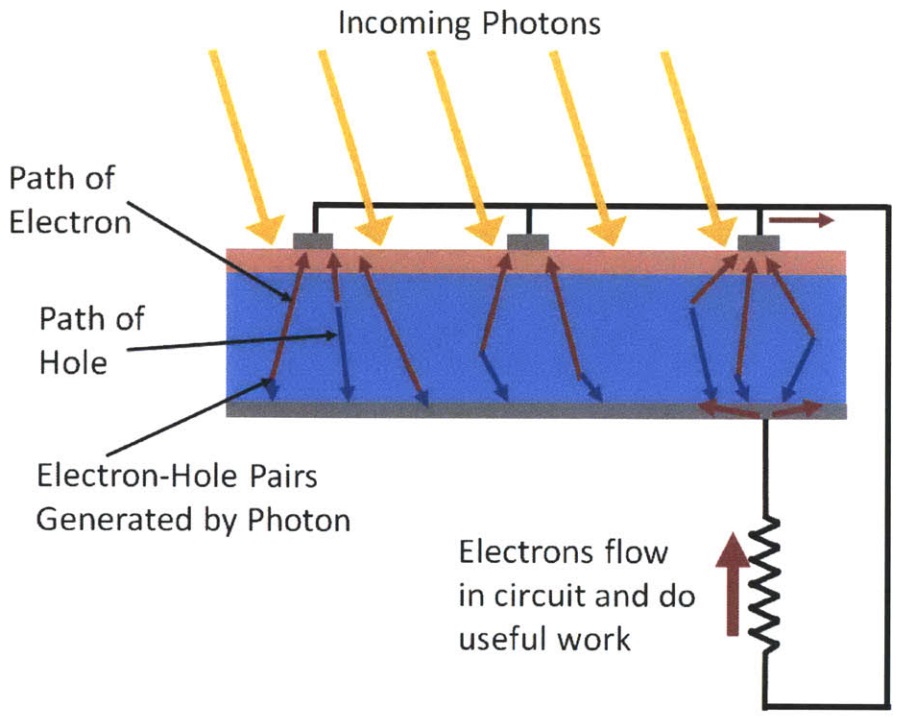

Figure 2.2: Solar Cell Operation Schematic Illustration

Incoming photons create electron-hole pairs that conduct useful work in the load circuit.

2.1.1 Dependence of Impurity Concentration and Cell Performance

In an ideal device (Figure 2.2), all electron-hole pairs contribute to useful work. In an actual device, several mechanisms force the electron-hole pair to recombine before being collected (Figure 2.3). This reduces the output of the device. The effective minority carrier diffusion length L_e, given in equation (2.1) below, measures the average distance a minority carrier can travel in a semiconductor before recombining.[33] The minority carrier diffusivity is given by D_e and the effective minority carrier lifetime by τ_e. To approach ideal carrier flow (Figure 2.2), the minority carrier diffusion length should be much greater than the thickness of the device.

$$L_e = \sqrt{D_e \tau_e} \tag{2.1}$$

$$\frac{1}{\tau_e} = \frac{1}{\tau_{rad}} + \frac{1}{\tau_{Auger}} + \frac{1}{\tau_{SRH}} \tag{2.2}$$

The effective minority carrier lifetime is the harmonic sum of the lifetime of multiple recombination mechanisms (2.2).[31, 34] The trap lifetime (τ_{SRH}) is the dominant lifetime limiting term in standard silicon solar cells and is the result of the effects of multiple defects, such as metal impurities, dislocations and grain boundaries, in the bulk material of the solar cell.[31, 35] The efficiency penalty caused by metal impurities has been evaluated through the intentional contamination of ingots, where it was found that concentrations < 1 ppba for some impurities are sufficient to significantly reduce performance.[36, 37] In silicon solar cells, iron impurities are typically a dominant performance-limiting defect in as-grown materials,[38, 39] with concentrations as low as 10 ppba (10^{14} atoms/cm^3) that are sufficient to impede performance.[36, 37] The mitigation of these performance limiting impurities is critical to reach the cost reduction targets presented above (1.3.2), especially given the importance of efficiency in determining module manufacturing cost (Figure 1.4).

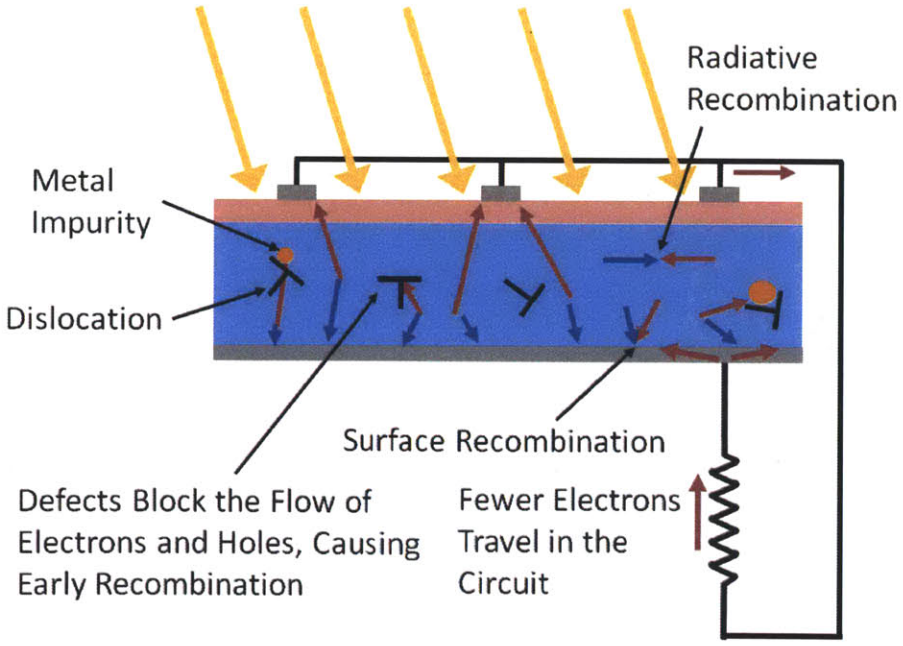

Figure 2.3: Recombination Limited Solar Cell Operation Schematic Illustration

Defects such as dislocations and metal impurities block the flow of charge, and reduce the amount of useful work from the cell.

2.2 Phosphorus Diffusion Gettering

The harmful effects of iron impurities can be reduced with a manufacturing process called phosphorus diffusion gettering (PDG) (Figure 2.4).[40, 41] During PDG, iron is gettered (transported to) to the phosphorus rich emitter layer which has a higher solubility for interstitial iron than the *p*-type bulk.[42] The process generally increases cell performance, but must be optimized for both the concentration and distribution of impurities for maximum impact.[23, 39, 41-43]

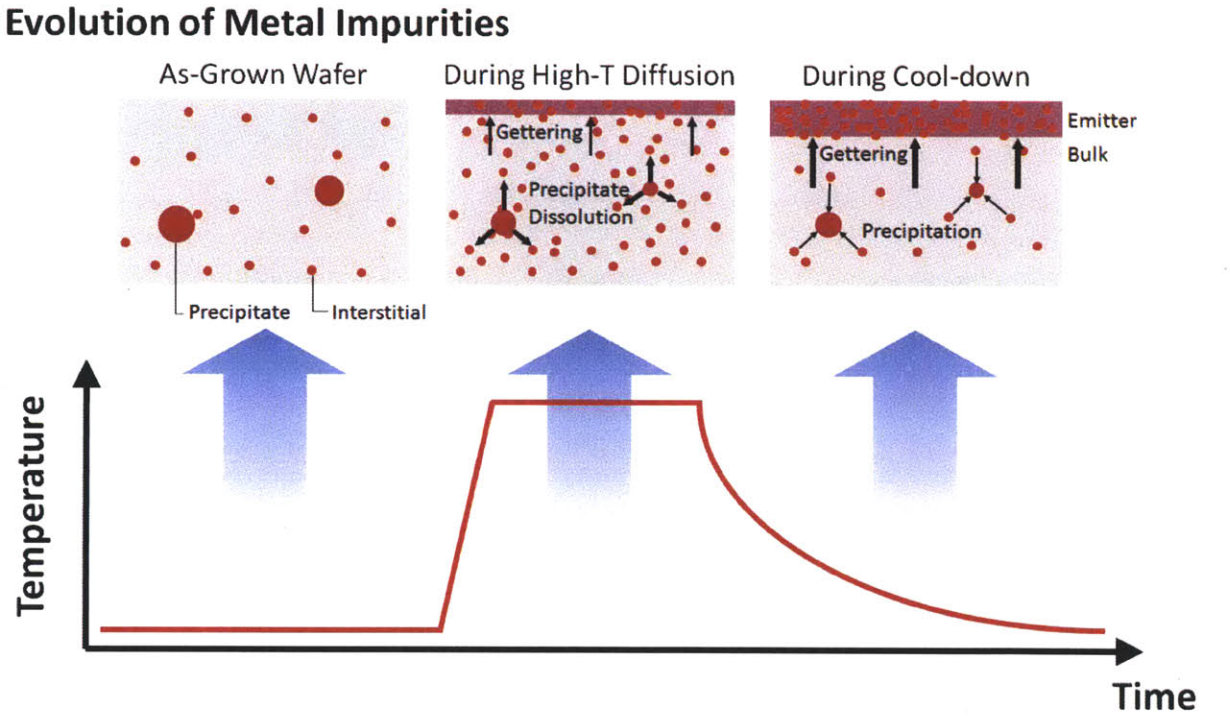

Figure 2.4: Phosphorus Diffusion Gettering

The kinetics of the PDG process are complex, but generally result in a reduced interstitial iron concentration. Figure reprinted from reference.[41]

CHAPTER

3

THE IMPURITIES TO EFFICIENCY SIMULATOR

3.1 I2E Overview

The MIT PV Lab and our collaborators have developed the I2E simulation tool to quickly predict the efficiency of solar cell devices as a function of the initial iron concentration and distribution in as-grown wafers and cell processing parameters (Figure 3.1).[44] It is a compactly packaged deployment of previous simulation efforts[41, 45-50] and has been validated through comparisons to experimental data for gettering in mono-Si and mc-Si wafers.[41, 44] Three main components make up the tool:

1. A kinetic model for phosphorus doping, interstitial diffusion, and precipitate dissolution and growth.[41]
2. A minority carrier lifetime model that considers both interstitial and precipitate recombination mechanisms.[41]
3. A device simulator, PC1D, to calculate cell performance.[41, 51, 52]

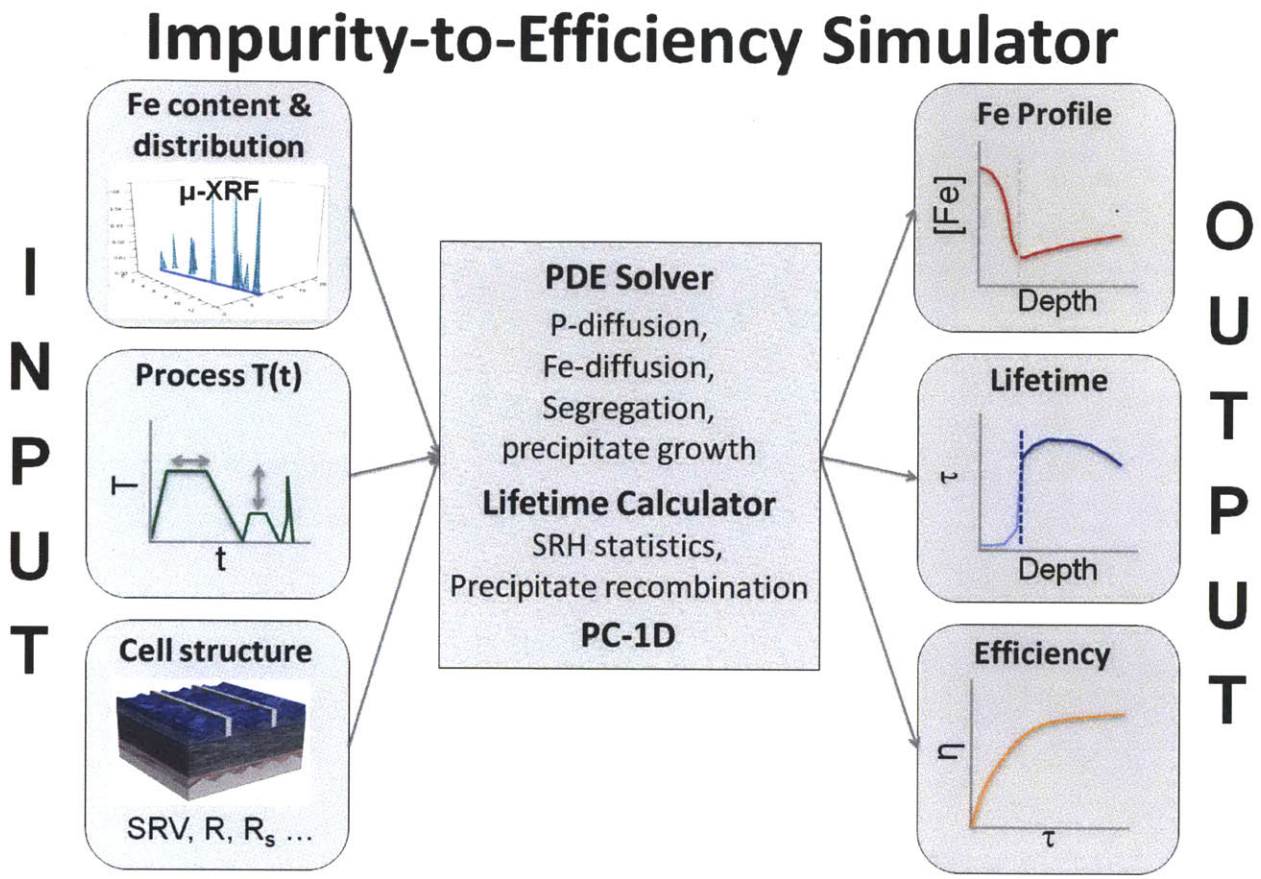

Figure 3.1: I2E Simulator Schematic

The I2E simulator links input impurity concentration, processing, and device performance. Figure reprinted from reference.[41]

3.2 System of Equations

The kinetic simulator requires the solution of three coupled non-linear partial differential equations (PDE) in one dimension as described in the reference.[44] First, the concentration of phosphorus is given by the diffusion equation (3.1) with a concentration dependent diffusitivity $D_P[P]$.[44, 53] Next, the concentration of interstitial iron is given by the diffusion-segregation equation for iron (3.2).[44, 54] This relies on the semi-empirical segregation coefficient for iron in phosphorus doped Si $m[P,T]$[44, 55] The diffusion coefficient of iron is given by $D_i[T]$, and the

derivative of the segregation coefficient normalized by dP/dx is given by $H[P,T]$.[44] Lastly, the precipitated iron concentration is given by the precipitation equation (3.3).[44, 56] Here the equilibrium concentration of interstitial iron is given as a function of temperature in C_{ieq}, while constants define the volume of an iron atom in an iron precipitate V_{Fe}, and the density of precipitates N_p.[44]

These equations are subject to the initial conditions below (3.4 – 6). The initial phosphorus concentration is a Gaussian distribution, while the interstitial and precipitate iron are constant through the thickness of the wafer.[44] The equations are also subject to the boundary conditions below (3.7 – 9). A constant surface concentration of phosphorus is assumed during diffusion, while all other concentrations and surfaces are constrained to have zero flux. The kinetic equations result in the formation of a phosphorus doped emitter region and the redistribution of precipitate and interstitial iron (Figure 3.2) following the physics discussed above (2.2).

$$\frac{\partial P}{\partial t} = \frac{\partial}{\partial x}\left(D_P[P]\frac{\partial P}{\partial x}\right) \tag{3.1}$$

$$\frac{\partial C_i}{\partial t} = \frac{\partial}{\partial x}\left(D_i[T]\frac{\partial C_i}{\partial x} - D_i[T]\frac{C_i}{m[P,T]}H[P,T]\frac{\partial P}{\partial x}\right) - \frac{\partial C_p}{\partial t} \tag{3.2}$$

$$\frac{\partial C_p}{\partial t} = 4\pi N_p\left(\frac{C_p}{N_p}\frac{V_{Fe}}{\pi}\frac{3}{4}\right)^{1/3} D_i[T]\left(C_i - C_{ieq}[T]\right) \tag{3.3}$$

$$P_0 = P_0[x] \tag{3.4}$$

$$C_{i0} = C_i[x] \tag{3.5}$$

$$C_{p0} = C_p[x] \tag{3.6}$$

$$P[0,t] = P_{00} \quad \frac{\partial P[x\max,t]}{\partial x} = 0 \tag{3.7}$$

$$\frac{\partial C_i[0,t]}{\partial x} = 0 \quad \frac{\partial C_i[x\max,t]}{\partial x} = 0 \tag{3.8}$$

$$\frac{\partial C_p[0,t]}{\partial x} = 0 \quad \frac{\partial C_p[x\max,t]}{\partial x} = 0 \tag{3.9}$$

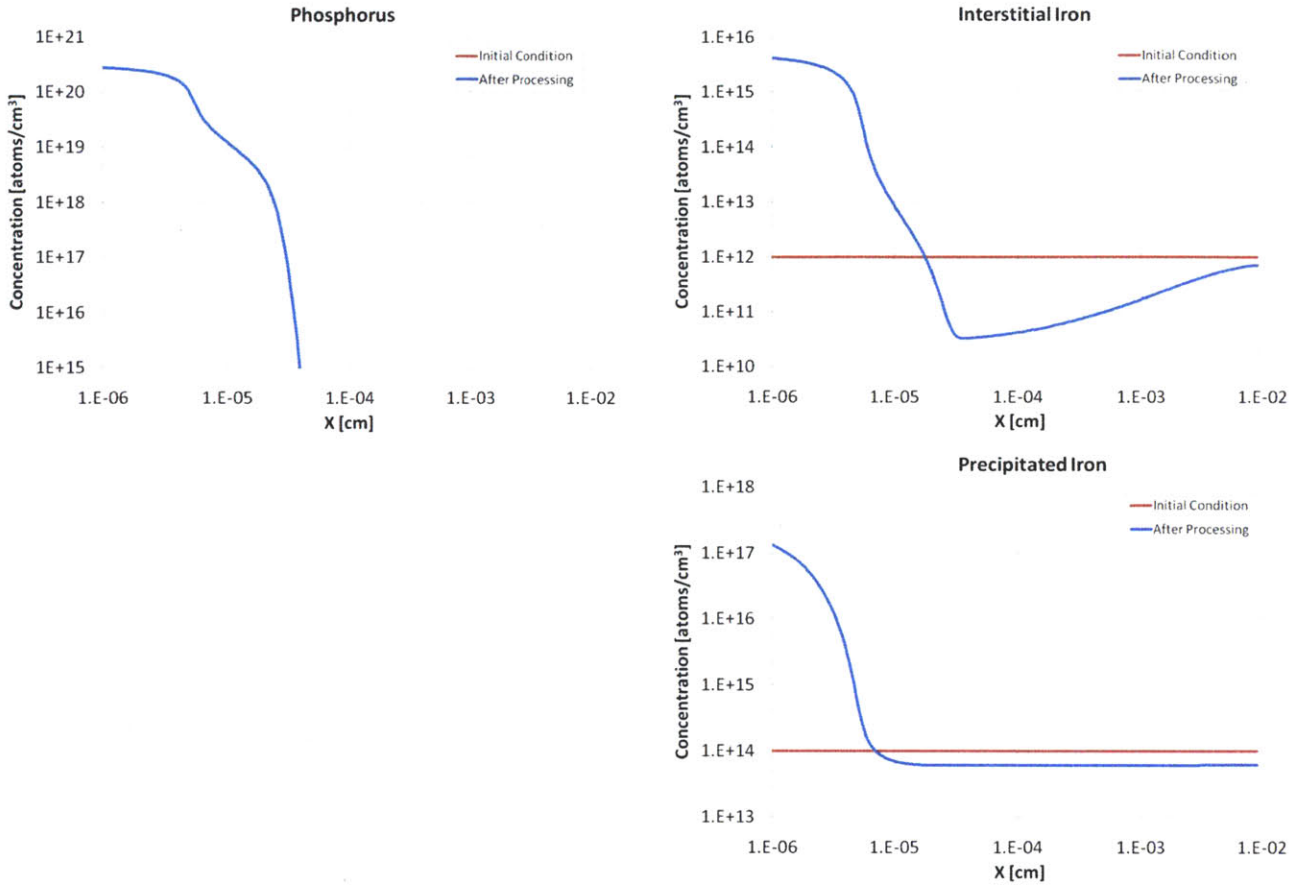

Figure 3.2: Results of Kinetic Equations

The kinetic equations result in a defined phosphorus diffusion profile, and the resulting distributions of interstitial and precipitated iron.

The lifetime calculator considers recombination through both interstitial and precipitate mechanisms.[41, 44] The effective lifetime is given by (3.10) and is a harmonic sum of the lifetime contributions from the interstitial, τ_i, and precipitate components, τ_p.

$$\tau_{eff} = \frac{1}{\frac{1}{\tau_i} + \frac{1}{\tau_p}} \tag{3.10}$$

3.3 Numerical Methods

3.3.1 System Stability

The system of equations governing phosphorus diffusion gettering proved to be unstable during initial solution attempts. Instabilities were observed in two manifestations. First, globally unsettled behavior could be observed that would result in the violation of the iron boundary conditions and the loss of physically meaningful concentrations (Figure 3.3). The second instability that could be observed were aggressive local oscillations that followed the general trend of a stable solution. The development of the numerical methods explained below was completed to reduce these effects and improve the computational penalty of the simulator. Unstable solutions can still be observed with the simulator after the modifications presented herein but can generally be eliminated through applying a larger number of nodes to the solution (3.3.3).

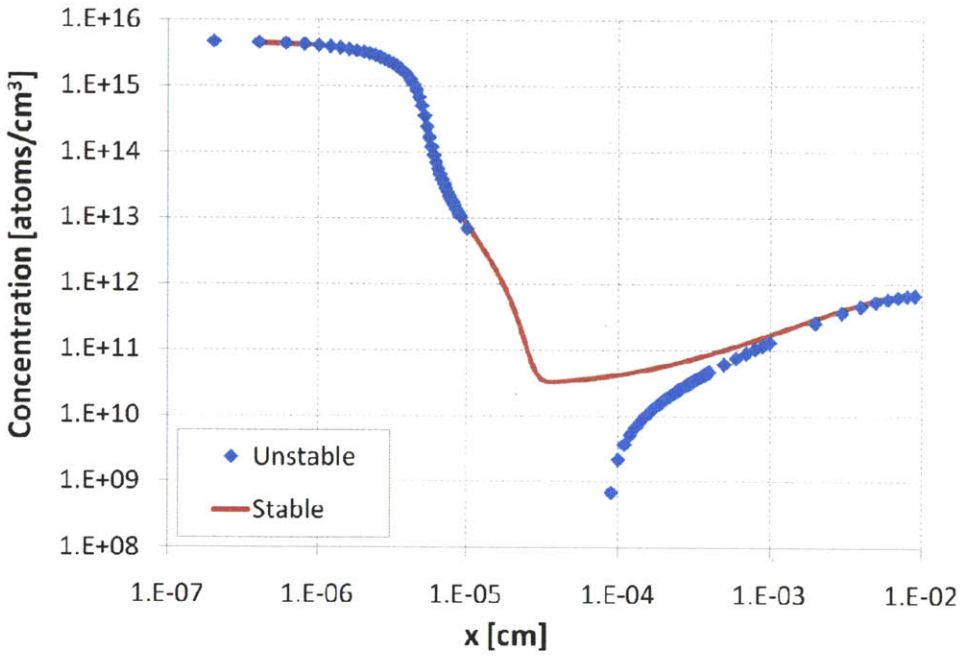

Figure 3.3: Globally Unstable Solution

Example of unstable solution from globally unsettled behavior.

3.3.2 Method of Lines and PDEPE

Two solution schemes were investigated to solve the system of nonlinear PDE's defining the kinetics of the gettering process. First, a built-in MATLAB® function, pdepe, for solving initial value boundary value PDEs in one dimension was employed.[57] Pdepe, discretizes the spatial coordinate across the spatial node points of the system of PDE's using a piecewise nonlinear Galerkin/Petrov-Galerkin method[58] and then solves the resulting system of ordinary differential equations (ODE) in parallel with ode15s.[59] Ode15s is well suited to solve a stiff system of ODE's that are characterized by unstable behavior[60, 61] as observed with the kinetic simulator. The default deployment of pdepe initially yielded unsatisfactory results with the system of equations of interest, as the solution was highly unstable and demanded significant computational time.

To improve stability and execution time, a custom solution script was developed that employed the Method of Lines (MOL) algorithm and was based on the open source MATMOL toolbox.[62, 63] Similarly to pdepe, the MOL algorithm discretizes the spatial coordinate and then solves the resulting system in parallel. Unlike the pdepe formulation however, this method provided greater transparency and the flexibility to implement a variety of finite difference methods for spatial derivatives and ODE solving routines. The two methods were first compared with a simple heat equation (3.11) subject to the initial and boundary conditions below (3.12). Both solution schemes match well with the analytical solution (3.13) (Figure 3.4), but the MOL scheme completed the calculation in approximately half the time as pdepe.

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{3.11}$$

$$u(x,t=0) = \sin(\pi x/2)$$

$$\frac{\partial u(x=x_f,t)}{\partial x} = 0 \tag{3.12}$$

$$u(x,t) = e^{-(\pi^2/4)t}\sin(\pi x/2) \tag{3.13}$$

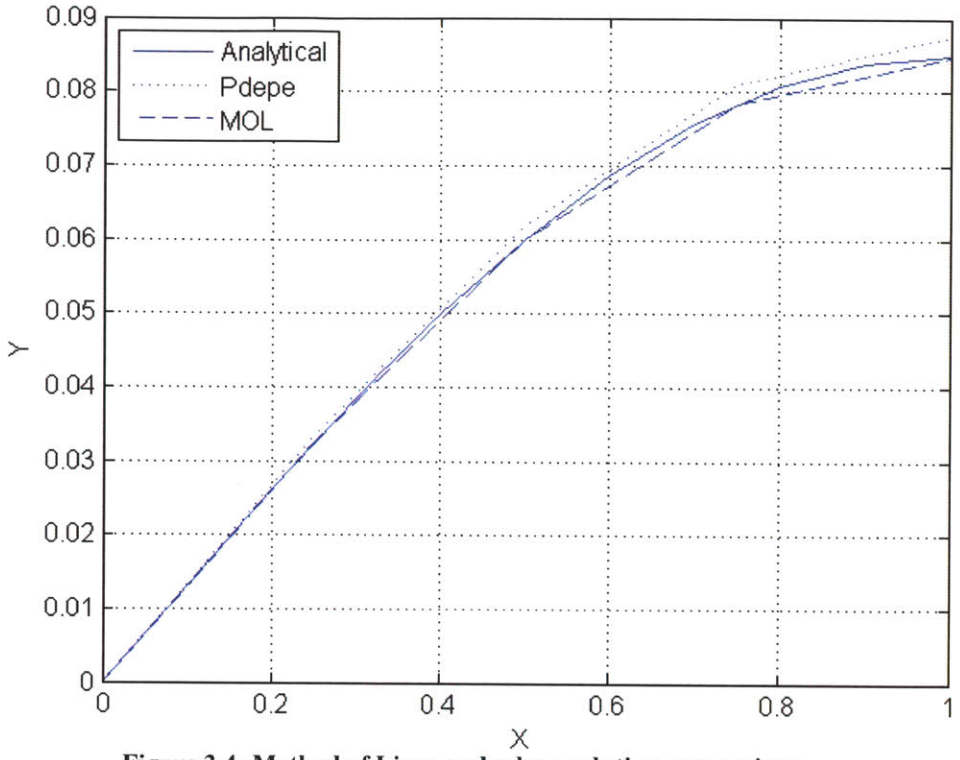

Figure 3.4: Method of Lines and pdepe solution comparison

Both the Method of Lines and pdepe solution schemes match the analytical solution well.

Both schemes were implemented to solve the governing system of PDE's (3.2) with the modifications described below. The final MOL implementation relied on two point upwind and three point centered finite differences for spatial discretization. Higher order spatial derivatives were attempted but tended to increase the instability of the solution. Ode15s was also used in the final MOL implementation to solve the system of parallel ODEs that resulted from the discretization procedure.

The pdepe solution scheme performed better than the Method of Lines implementation while solving the kinetic equations. Both schemes were compared, and are summarized for one simulation case below (850 °C, 20 min anneal, 10^{14} atom/cm^3 total iron, 10^{12} atom/cm^3 interstitial iron). The simulator was used to calculate the final bulk minority carrier lifetime at

multiple numbers of nodes. Increasing the number of nodes increases the accuracy of the simulation at the expense of greatly increased execution time (Section 3.3.4). The MOL solution reduced computational time by an average of 60% relative to pdepe (Figure 3.5). However, MOL suffered from poorer accuracy than the pdepe algorithm (Figure 3.6). The final results of the pdepe solution scheme proved to be much less sensitive to the number of node points than MOL, in this case acceptable results for pdepe found at 100 node points while 400 were required for MOL. Additionally, the MOL algorithm provided an outlier at 300 node points. Therefore, the pdepe scheme was utilized for the simulator, though the majority of the improvements gained through the modifications discussed below were applied to both solution schemes before the final determination was made.

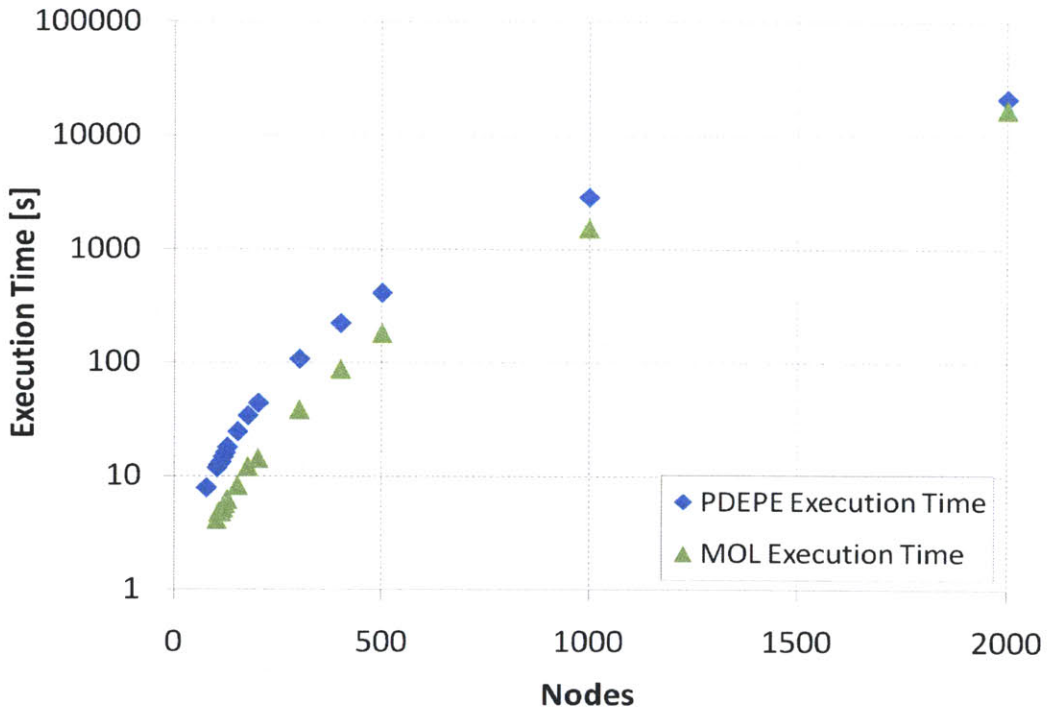

Figure 3.5: Execution Time Comparison between MOL and pdepe

The MOL scheme reduces execution time in comparison to pdepe at multiple numbers of mesh points.

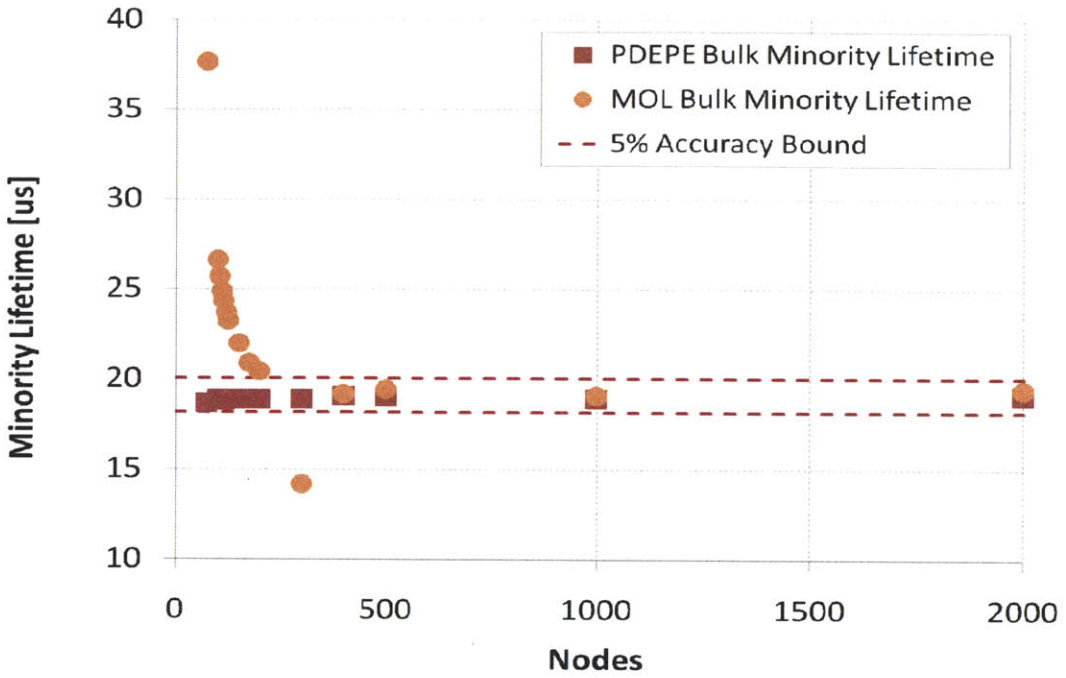

Figure 3.6: Accuracy Comparison between MOL and pdepe

The pdepe scheme is less sensitive to the number of node points and provides much greater accuracy at lower node counts.

3.3.3 Mesh Optimization

An algorithm was implemented to optimize the distribution of spatial node points used in the simulation. The number of spatial node points greatly affects the execution time of the simulation, which provides strong motivation to distribute them in an optimal fashion. The most significant kinetic activity in the simulation occurs in approximately the first 0.5 μm of an approximately 180 μm thick wafer (Figure 3.2). Therefore, poor results were obtained if a constant spatial mesh was applied to the entire thickness of the device. Mesh densities were too low in the region of interest near the front surface of the device, while much higher than necessary in the bulk of the device. If the mesh density in the region of interest was made suitably small for an accurate solution, the total node count would approach 10^6, which is too high for reasonable execution times (Figure 3.5).

A spatially varying mesh density was employed to overcome this shortcoming, where a higher density was employed in the highly active region, and lower densities were employed in the bulk of the device. First, a predictive junction depth based algorithm was implemented that defined a single boundary between the region of interest and the bulk of the device. This algorithm resulted in significant instabilities (Figure 3.3) near the discontinuity in node densities between the two regions. Additional attempts to smooth the transition region did not provide adequate stability. Upon the suggestion from a colleague that the solution of the kinetic equations naturally falls in a logarithmic spatial coordinate, a logarithmically based method was implemented. In the method, the requested node points are equally distributed in logarithmic space starting at 10^{-8} cm, and then are transformed to x space ($x=e^{z}$). The logarithmic mesh resulted in reduced accuracy sensitivity to the number of node points (Figure 3.6) and improved stability.

3.3.4 Maximum Time Step

The specification of the time step of the PDE solution is dependent on the spatial coordinate as discussed above (3.3.3). Both the MOL and pdepe solution schemes utilize a variable time step algorithm that modulates the time step as needed while solving the system of equations. However, a maximum allowable time step was defined following the stability condition of an explicate finite difference solution for the heat equation.[61] The maximum allowable time step (3.14) scales inversely with the square of the logarithmic spatial step (3.15). This relationship creates a strong computational penalty for increasing the number of mesh points (Figure 3.5). The execution time as a function of the time ratio constant for two characteristic problems was investigated (Figure 3.7). The incomplete data set for the 950 °C simulations originates from a lack of stability at higher time ratios for this case. The time ratio constant, $TR_{constant}$, was tuned empirically to a default value of 1000 for accuracy and stability using the same analysis methods

used for the solution algorithm. Higher time ratio constants, up to 1500, are stable, accurate, and provide a worthwhile reduction in execution time for lower temperature simulations that do not fully dissolve precipitates (3.3.5).

$$\frac{\Delta t}{\Delta z^2} = TR_{\text{constant}} \tag{3.14}$$

$$z = \log_{10}(x) \tag{3.15}$$

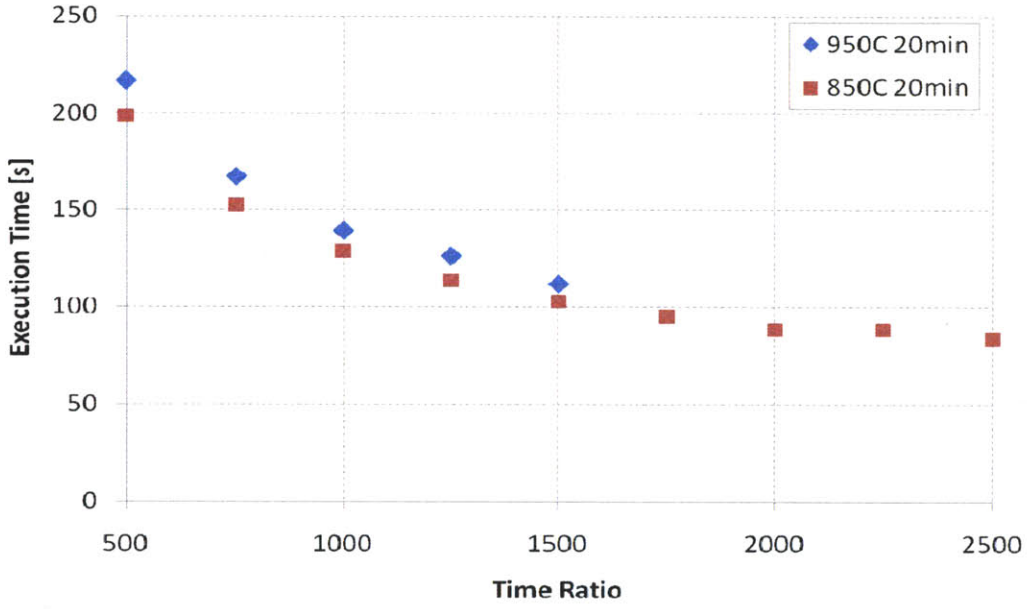

Figure 3.7: Dependence of Execution Time on Time Ratio

Execution time is highly sensitive to the time ratio. The time ratio was tuned empirically to a default value of 1000.

3.3.5 Minimum Precipitate Concentration

At high temperatures, the rate of change in precipitate concentration is rapid (3.3). This trend is further exacerbated by small precipitate sizes, with very high surface area to volume ratios. Because of this, the precipitate concentration could become negative as the ODE solvers stepped through time, a violation of physical principles. An algorithm was enacted in the pdepe solution scheme that checked the initial calculation for precipitate concentration against a minimum limit.

If the solution scheme attempts arrive at a precipitate concentration that is below the minimum limit, the algorithm enforces the limit while accounting for the conservation of interstitial and precipitate iron at the node point (Figure 3.8). The algorithm required adding a global variable within ode15s to report the current time step of the solution.[59] The time step is used with the current time derivative of the precipitate concentration and the solved value at time *t-1* to predict the solution at time *t* before exiting the precipitate calculations in DSE_Pdepe.m. If the limit needs to be enforced, the time derivative is set in DSE_Pdepe.m that will result in the desired solution with the global time step variable.

The minimum concentration limit was calculated for a precipitate radius of 2 nm, corresponding to a concentration of approximately 10^{10} atoms/cm^3. In order to improve execution time, precipitate concentrations are not recalculated for node points that have been forced to the minimum allowable concentration until the temperature is reduced below the temperature where the limit was first enforced. A post solution smoothing function was also implemented to reduce oscillations in the final precipitate concentration that resulted from the algorithm.

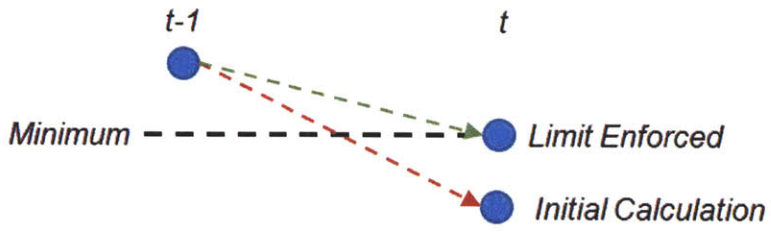

Figure 3.8: Minimum Precipitate Concentration Algorithm

The minimum precipitate concentration algorithm prevents calculated values from falling below an enforceable minimum limit.

3.3.6 Removal of Trapped Iron Term

The initial formulation of the system of equations for the kinetic equations included a term to account for the concentration of interstitial iron that was trapped within other defects in the wafer.[44] This term required the subtraction of two terms spanning approximately 10^{19}, which is greater than the number of significant digits, 15- 17, that the double precision data type can support.[64] The removal of the trapped iron term modified the simulation result by 10^{-5} % and reduced execution time by 22% for a characteristic problem.

3.4 Example Application

The I2E simulator has optimized cell processing conditions for industrially relevant problems.[41] As an example, the tool was used to optimize the hold temperature of a low temperature annealing (LTA) step (Figure 3.9),[65] which has been shown to increase the performance of some mc-Si solar cells.[66, 67] I2E simulations were conducted with a high initial total iron concentration of 10^{15} atoms/cm^3, an initial interstitial iron concentration of 10^{13} atoms/cm^3, and LTA duration of 45 min after 30 min P-diffusion at 850 °C.[65] The LTA temperature was varied from 400 – 700 °C, and resulted in the identification of an optimum temperature (Figure 3.10).[65] In this simulation, final lifetimes were observed to be quite low because of the very high initial total iron concentration.[41]

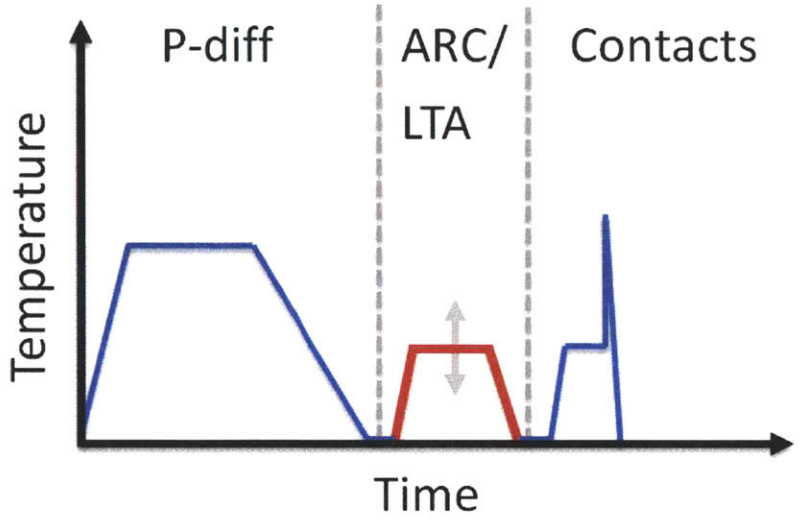

Figure 3.9: Time Temperature Profile for LTA Optimization

The hold temperature of a low temperature anneal was optimized using the I2E simulator.

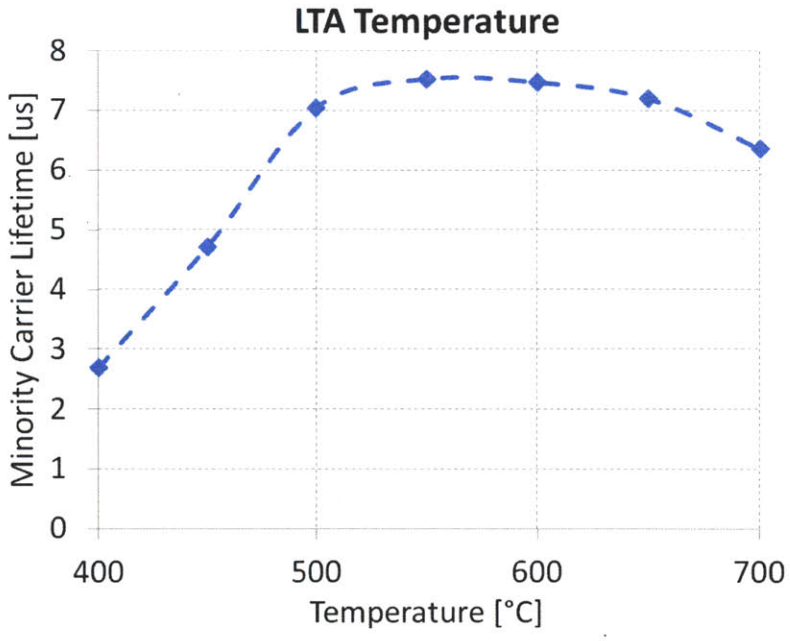

Figure 3.10: Optimization of LTA Temperature

The I2E simulator predicts an optimum low temperature annealing temperature. Figure reprinted from reference.[65]

CHAPTER

4

ONLINE IMPLEMENTATION

4.1 Web Applet

The initial implementation of the I2E simulator was created in MATLAB® for internal use. In order to increase the impact of the effort expended to develop the tool, the simulator was deployed for free use by the academic and industrial communities. The utilization of MATLAB® as the computational engine, and the modifications to ode15s, were confirmed to be allowable under the MATLAB® license (Thread ID: 1-I0NERQ with MathWorks customer service). A Java based web applet was chosen for distribution because the core code of the simulator could be protected on a local calculation server and the language is inherently cross-platform. Users are able to configure simulations (Figure 4.1) and interpret results (Figure 4.2) within a graphical interface. The applet and interface were developed using the Netbeans Integrated Development Environment version 7.0.1 with the Java™ SE 1.6, build 26. The applet is signed[68] for security purposes and requires users to approve the applet in order to allow it to access their local file system. This is required if users would like to save and load files on the applet, including PC1D inputs, from their local file system. The applet was coded with over 5,800 lines of code that relied heavily on helpful resource materials.[69-74]

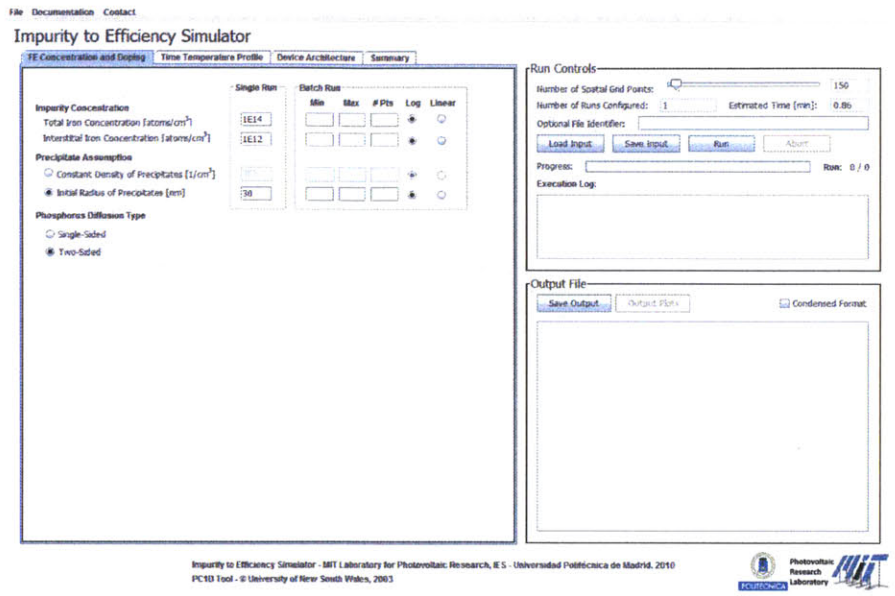

Figure 4.1: I2E Applet Screenshot before Submitting a Calculation

Users define the initial iron concentration, time temperature profile, device architecture, and numerical parameters.

Figure reprinted from reference.[65]

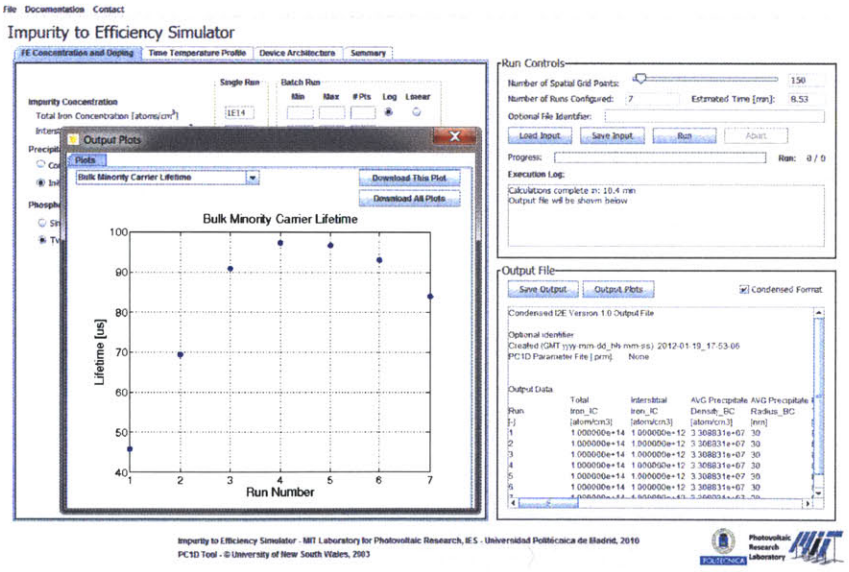

Figure 4.2: I2E Applet Screenshot after Completing Calculation

The I2E applet allows users to interpret data with plots or download text file outputs. Figure reprinted from reference.[65]

Input parameters to the simulation are configured in the GUI. When the user selects the run button (Figure 4.3), a new calculation thread is initiated on the users CPU. This thread creates an input file in the java applet and saves it on a calculation server using common gateway interface (CGI) scripts (4.2). Additionally, if the use of PC1D is requested from the user, an additional file directory with a modified copy of the PC1D code is created on the server. The CGI scripts then call the MATLAB® instance of the kinetic simulator and lifetime calculator (3.3) that is installed on the server. During execution, the MATLAB® simulation updates the applet GUI with its progress for the purpose of updating the user with a progress bar. Upon completion of the calculations, the MATLAB® program writes output data files, creates plots to display in the GUI, and creates a log entry for the program's execution. A timestamp and random number are used to identity files and directories from one another on the server.

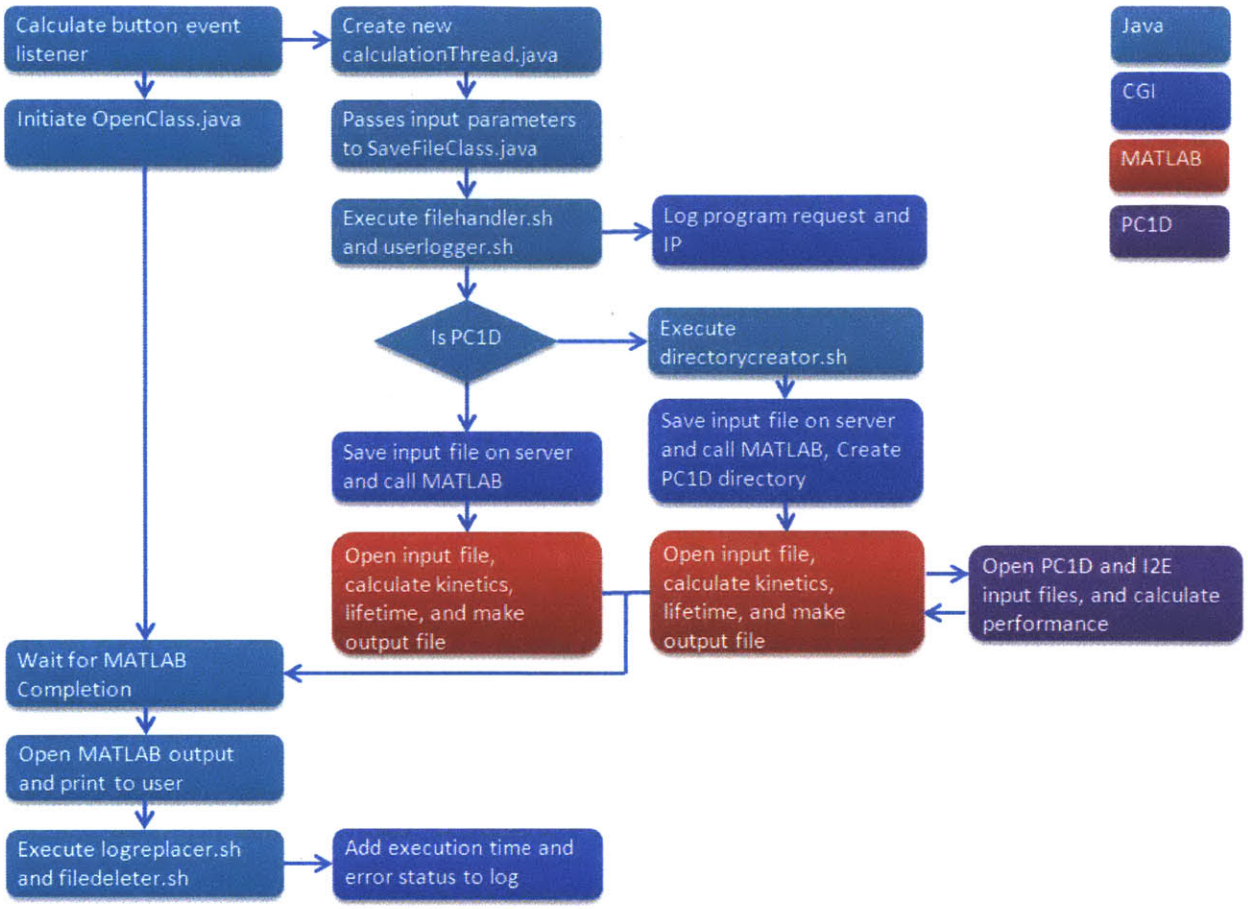

Figure 4.3: I2E Communication Diagram

Running an I2E simulation requires the coordination of multiple pieces of software

4.2 CGI Scripts

The server also hosts seven CGI scripts that complete the handshakes between the applet, MATLAB®, and PC1D (Table 4.1). The scripting relied on several useful reference materials.[75-84] The majority of the scripts are placed in the cgi-bin directory on the server.

Table 4.1: CGI Scripts

Script	Description
directorycreator	Creates a directory for PC1D calculations, if requested, and copies the modified PC1D executable to the directory. The run's timestamp and random index are used to identify the directory.
filedeleter	Deletes input files after calculations are complete. The directory path is hard-coded to the tempdata directory to avoid the deletion of system files. This function was replaced with the cleanuptempdata function after multiple output file type formats began to be supported.
filehandler	Saves the standard input/output file from the applet onto the server and initiates MATLAB® calculations.
filehander_condensed	Creates condensed version of the input/output file.
filesaver	Saves PC1D input files into directory created by directorycreator.
userlogger	Creates log file entry when user executes program, which includes: user IP address, random index, Date and time, number of runs requested, and if PC1D was requested.
logreplacer	Modifies the original log entry when a run is completed to indicate: execution time, and the success of the run; no-errors, errors, user canceled, or calculations timed-out.
cleanuptempdata	Deletes all files in the tempdata directory daily that are more than one day old. Placed in the /etc/cron.daily

4.3 Calculation Server

A dedicated high performance calculation server (Table 4.2) is used to host the MATLAB® based calculations (3.2) and PC1D. The same server also hosts the website and applet with a static IP address defined from MIT Information Services and Technology (IST). The server was

configured with input from several resources while following the best practices of internet security.[85-89] The server runs a terminal emulator[90] as PC1D requires a display to be set.

Table 4.2: Calculation Server Specifications

Model:	Dell T7500
Processors:	Dual Six-Core Intel® Xeon® E5645 with 2.4 GHz per core
Memory:	12GB of 1333 MHz DDR3
Battery Backup:	Tripp Lite SmartPro 750SLT UPS
Operating System:	Ubtuntu 9.10
Address:	http://pv-i2e.mit.edu

The original calculation server used on the project utilized a single four-core Xeon® processor and 4GB of ram. The current server was deployed to allow up to ten users to simultaneously use the tool online with no performance degradation. To test the capabilities the servers, execution time was logged for a standard simulation with a varying level of active calculation threads (Figure 4.4). The current server maintains low execution time for 10 users, while the original server's performance begins to degrade as the number of active threads surpasses the available number of computational cores. The current server also reduced execution time relative to the original server in non-computational core limited cases.

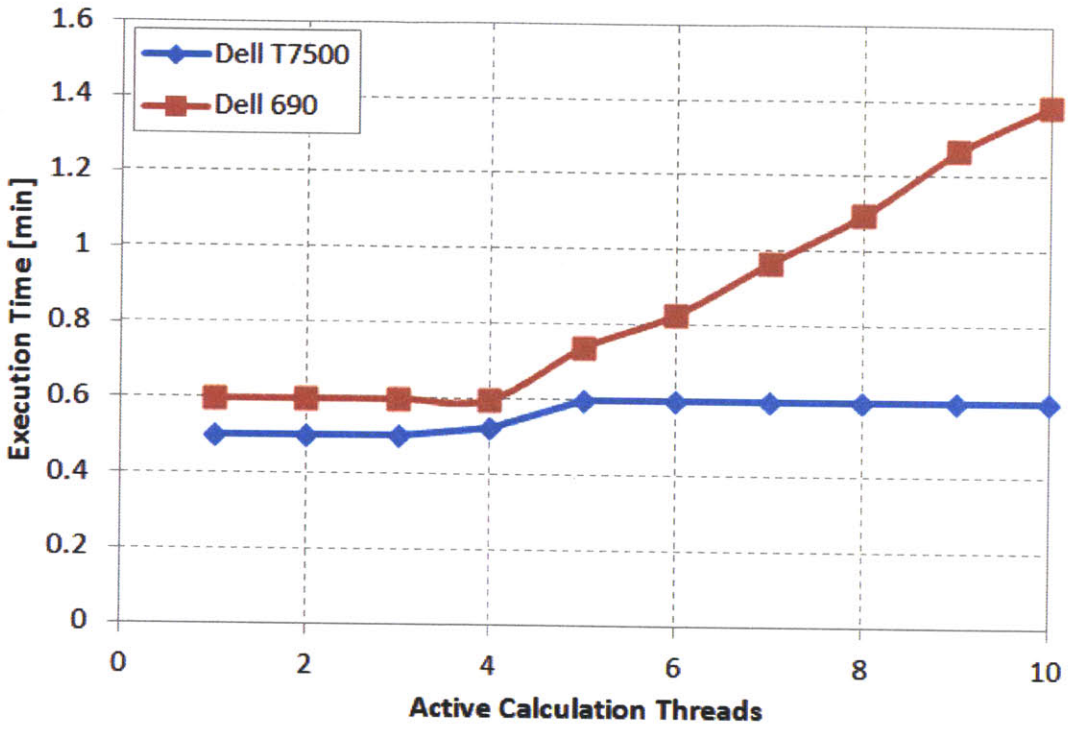

Figure 4.4: Execution Time of Servers

The current web server maintains performance with 10 simultaneous calculation threads, while the original server's performance degrades when the number of active threads surpasses the number of computational cores available.

4.4 PC1D Modifications

PC1D is an industry-standard, one-dimensional device simulator.[51, 52] The I2E model uses PC1D to calculate device performance using the final metal distribution and minority carrier lifetime calculated from the kinetic simulator. The PC1D program's executable and source code are available from SourceForge.[91] The I2E simulator uses a modified version of the PC1D tool that allows it to function in a command line environment (Figure 4.5). These changes have not affected calculation algorithms. PC1D is a Windows application, but runs on the Linux based server with the Wine application.[92-94] The PC1D source code was modified in Microsoft® Visual

Studio® with helpful references.[95-97] Five key modifications were made to the PC1D source code (Table 4.3). An example functional call is:

```
>> C:\PC1D>pc1d_MIT -g pvcell.prm -C:\PC1D\TauInput.txt –C:\PC1D\Pdiff.txt -C:\OutputFolder\DeviceOutput.txt -
C:\OutputFolder\OutputFile.txt - -C:\OutputFolder\ErrorFile.txt
```

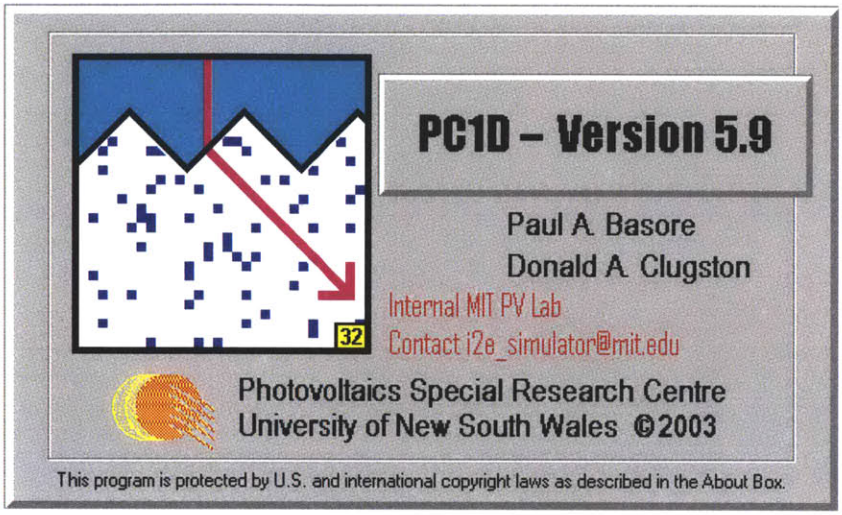

Figure 4.5: Modified PC1D Splash Screen

The I2E Simulator uses a modified version of PC1D

Table 4.3: PC1D Code Modifications

Modification	Description
Output file creation	A standard output file was created in ParmSection.cpp. ParmSection.cpp is called during the program initialization before the .prm file is read, so a default file name is provided that will be overwritten at each program execution.
Command line execution	The existing "-g" command line functionality was modified in PC1D.cpp. A class "GlobalFileNames" was added in the useful.h header file to store file names passed to the program via the command line. The variables are initialized in useful.cpp and given default values to avoid errors, as parmselect.cpp runs before pc1d.cpp. As an example, global variables are called with GlobalFileNames::tauInput.
Device thickness and number of region output	Users do not input the total device thickness when PC1D functionality is requested. "-d" command line functionality was added in PC1D.cpp to output only total device thickness and number of regions for the purpose of informing the applet

	what the total device thickness and number of regions.
Import carrier lifetime profile from kinetic simulator	Over-writes the carrier lifetime from the original PC1D .prm file with that calculated from the I2E kinetic simulator. Because PC1D allows multiple regions, the total thickness of the device from I2E must be segmented for the purpose of the PC1D calculations. In Device.cpp in void CRegion::Serialize(CArchive & ar) importing is completed one region at a time. This function gets data from the lifetime input file and computes a weighted average through the region's thickness. The input file format contains 3 columns with no header information separated by colons (distance in from front service [cm]:m_TauN[s]:m_TauP[s] with no spaces). The weighted average neglects the first 10 um when the when user has selected one region device to appropriately report the bulk lifetime.
Import phosphorus diffusion profile	Over-writes the phosphorus profile, Pdiffusion, from the original parameter files with that calculated from the I2E kinetic simulator. External diffusion profiles are standard ASCII files with a filename suffix .DOP. The input file format contains 3 columns separated by a tab or space (distance in from front surface[μm] donor doping density[cm^{-3}] acceptor doping density[cm^{-3}]). The position values must start at 0 and increase monotonically. The maximum number of positions that may be defined in the file is 500.

4.5 Web Applet Utilization

The applet was made available for use on June 29, 2011 after being announced at the 37th *IEEE Photovoltaic Specialists Conference*.[65] Since being introduced, the I2E applet has been used over 3,000 times (Figure 4.6). Users span the globe, but are concentrated in the United States, Germany, and China (Figure 4.7). How to use the I2E applet is described in the User Manual as attached in Appendix A.[98]

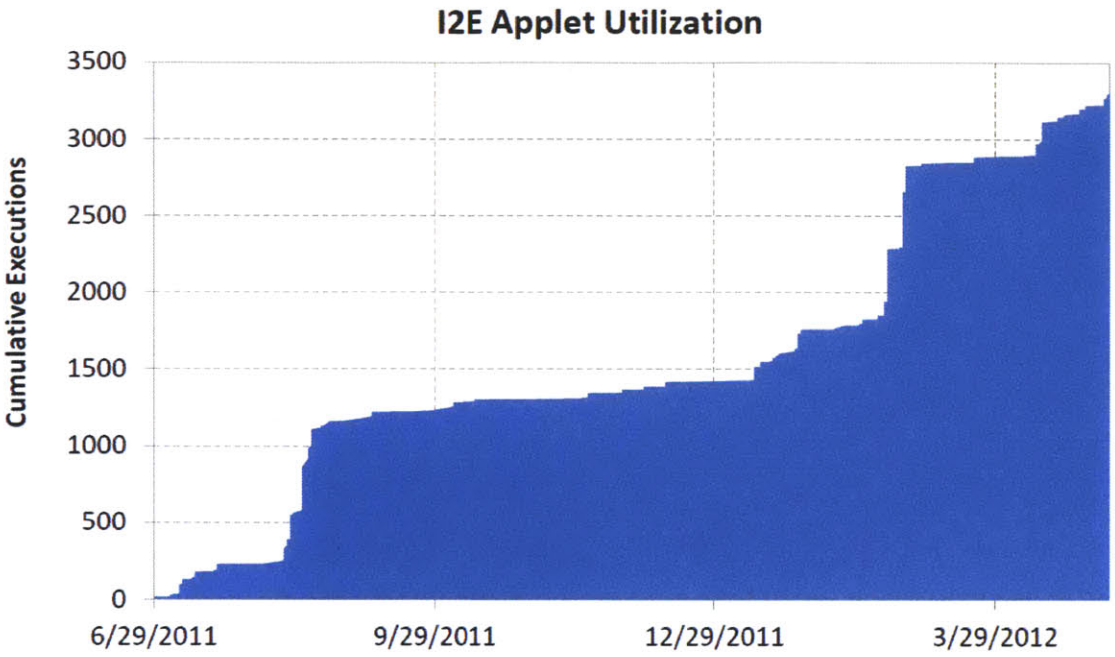

Figure 4.6: Cumulative I2E Applet Executions

The I2E Applet has been ran over 3,000 times since its release

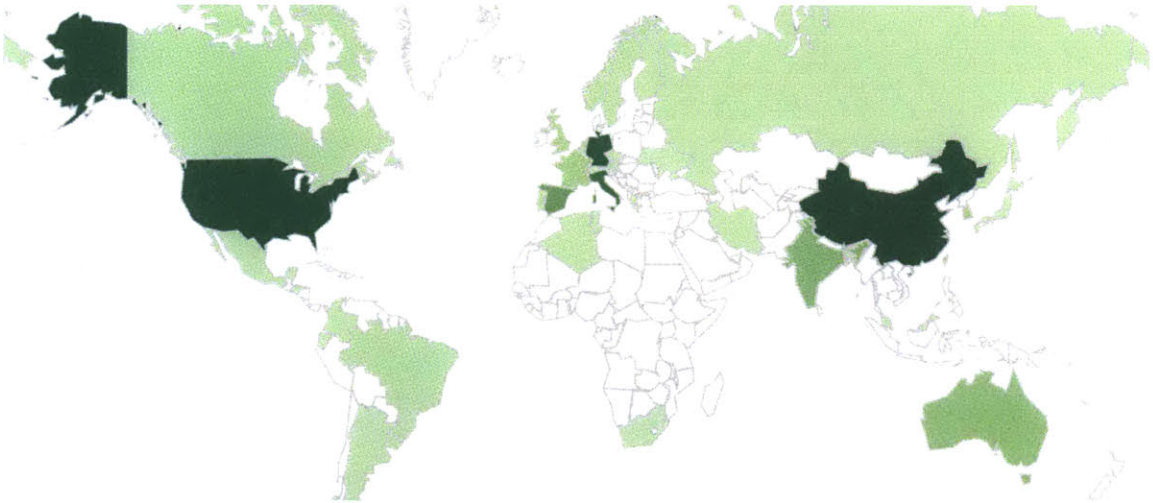

Figure 4.7: Concentration Map of I2E Executions

The I2E Applet has been utilized primarily in the United States, Germany, and China.

CHAPTER

5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

This work presented the Impurity-to-Efficiency (I2E) simulation tool and applet, and provided background on the economic drivers of solar cell installations and manufacturing. The adoption of renewable energy is a critical challenge facing this generation. Crystalline silicon photovoltaics are a particularly well suited technology to overcome this challenge. Manufacturing cost is very sensitive to changes in module efficiency, which is very sensitive to metal impurity contamination, especially iron. Metal contaminants can be mitigated with optimized manufacturing processes to increase module efficiency.

The I2E simulator captures the essential physics of iron impurity gettering during high temperature silicon solar cell processing. The tool also includes a device simulator. I2E required numerical methods to be developed that improved the reliability of solving the equations that govern iron impurity kinetics. To increase the impact of the tool, a web applet was developed to distribute the findings of the model to the academic and industrial communities. The I2E tool has been used to complete over 3,000 simulations to date, and has demonstrated high-impact improvements in industry.[41]

5.2 Future Work

The I2E simulator operates quickly because of its 1D design and simplified impurity kinetics. However, extending the simulation to 3D would allow back-contacted cell architectures that rely on lateral charge transport to be simulated by the tool. Additionally, many 3D process and device simulators are available, *ex.* Synopsys Sentaurus, that are rooted in the integrated circuit industry but gaining prevalence in PV.[99] The addition of the capability of I2E to simulate precipitate dissolution and growth to an industrial tool could increase its relevance in the PV industry. I2E also currently considers only iron-related defects, a dominant performance-limiting defect in most silicon solar cells,[38, 39] but other impurities contribute to cell efficiency losses. I2E could be expanded to consider other metallic elements.

The I2E simulator also demonstrates the paradigm of predictive defect engineering. This model could be applied to other defects, *i.e.* structural, in silicon or extended to other material systems. I2E could also be used in-line in a manufacturing facility to tailor processing to specific variations in input material quality.

REFERENCES

1. U.S. Energy Information Administration, *Annual Energy Review 2010*, U.S. Energy Information Administration,, 2011.
2. W. S. Broecker, *Science*, 1975, **189**, 460-463.
3. L. Bernstein, P. Bosch, O. Canziani, Z. Chen, R. Christ, O. Davidson, W. Hare, S. Huq, D. Karoly, V. Kattsov, Z. Kundzewicz, J. Liu, U. Lohmann, M. Manning, T. Matsuno, B. Menne, B. Metz, M. Mirza, N. Nicholls, L. Nurse, R. Pachauri, J. Palutikof, M. Parry, D. Qin, N. Ravindranath, A. Reisinger, J. Ren, K. Riahi, C. Rosenzweig, M. Rusticucci, S. Schneider, Y. Sokona, S. Solomon, P. Stott, R. Stouffer, T. Sugiyama, R. Swart, D. Tirpak, C. Vogel and G. Yohe, *Climate Change 2007: Sythesis Report*, Intergovernmental Panel on Climate Change, 2007.
4. K. Aleklett, M. Höök, K. Jakobsson, M. Lardelli, S. Snowden and B. Söderbergh, *Energy Policy*, 2010, **38**, 1398-1414.
5. N. A. Owen, O. R. Inderwildi and D. A. King, *Energy Policy*, 2010, **38**, 4743-4749.
6. International Energy Agency, *World Energy Outlook 2008*, International Energy Agency, Paris, France, 2008.
7. J. L. Bower and C. M. Christensen, *Disruptive Technologies: Catching the Wave*, Harvard Business Review, 1995.
8. The Breakthrough Institute, *Energy and Climate Program*, http://thebreakthrough.org/energy.shtml, Accessed May 11, 2012.
9. H. Weston A, *Energy*, 2006, **31**, 1685-1702.
10. A. Mills and R. Wiser, *Implications of Wide-Area Geographic Diversity for Short-Term Variability of Solar Power*, Ernest Orlando Lawrence Berkeley National Laboratory, 2010.
11. Global Wind Energy Council, *Global Wind Report - Annual market update 2010*, Global Wind Energy Council, 2011.
12. International Energy Agency, *Trends in Photovoltaic Applications - Survey report of selected IEA countries between 1992 and 2010*, International Energy Agency (IEA) Photovoltaic Power Systems Programme (PVPS), 2011.
13. U.S. Energy Information Administration, *Annual Energy Outlook 2011*, U.S. Energy Information Administration, 2011.
14. J. Nelson, J. Johnston, A. Mileva, M. Fripp, I. Hoffman, A. Petros-Good, C. Blanco and D. M. Kammen, *Energy Policy*, 2012, **43**, 436-447.
15. S. Mehta, *28th Annual Cell & Module Data Collection Results* in PVNews, 2012, vol **31**, issue 5.
16. D. M. Powell, M. T. Winkler, H. J. Choi, C. B. Simmons, D. B. Needleman and T. Buonassisi, *Energy and Environmental Science*, 2012, **5**, 5874-5883.
17. P. Maycock, *PV insider's report* in, 2001, vol **2**, issue.
18. US Department of Energy, *The Prospect for $1/Watt Electricity from Solar*, $1/W Workshop, 2010.
19. P. Mints, *DoE*, SunShot Silicon PV Roundtable Discussion, 2012.
20. S. B. Darling, F. You, T. Veselka and A. Velosa, *Energy & Environmental Science*, 2011, **4**, 3133-3139.

21. U.S. Department of Energy, *SunShot Vision Study*, 2012.
22. E. Sachs, *Photon International*, 2010, **6**, 272.
23. T. Buonassisi, A. A. Istratov, M. A. Marcus, B. Lai, Z. Cai, S. M. Heald and E. R. Weber, *Nature Materials*, 2005, **4**, 676-679.
24. C. Honsberg and S. Bowden, *Solar Cell Structure*, http://www.pveducation.org/pvcdrom/solar-cell-operation/solar-cell-structure, Accessed May 16, 2012.
25. S. Chunduri, *Photon International*, 2011, **5**, 222-257.
26. C. Honsberg and S. Bowden, *Anti-Reflection Coatings*, http://www.pveducation.org/pvcdrom/design/anti-reflection-coatings, Accessed May 16, 2012.
27. C. Honsberg and S. Bowden, *Optimisation of Finger Spacing*, http://www.pveducation.org/pvcdrom/design/optimisation-of-finger-spacing, Accessed May 16, 2012.
28. C. Honsberg and S. Bowden, *Surface Recombination*, http://pveducation.org/pvcdrom/design/surface-recombination, Accessed May 16, 2012.
29. C. Khadilkar, S. Kim, A. Shaikh, S. Sridharan and T. Pham, *International PVSEC-15*, Shanghai, China, 2005.
30. F. Faller, *PV Panles for the US: Make - not Buy!*, Centrotherm Photovoltaics USA Inc., 2010.
31. S. R. Wenham, M. A. Green, M. E. Watt and R. Corkish, *Applied Photovoltaics: Second Edition*, Earthscan, 2007.
32. C. Honsberg and S. Bowden, *Light Generated Current*, http://pveducation.org/pvcdrom/solar-cell-operation/light-generated-current, Accessed Jan 30, 2012.
33. C. Honsberg and S. Bowden, *Diffusion Length*, http://pveducation.org/pvcdrom/pn-junction/diffusion-length, Accessed Feb 2, 2012.
34. C. Honsberg and S. Bowden, *Bulk Lifetime*, http://pveducation.org/pvcdrom/characterisation/bulk-lifetime, Accessed Feb 3, 2012.
35. I. Tobías, C. d. Cañizo and J. Alonso, in *Handbook of Photovoltaic Science and Engieering*, eds. A. Luque and S. Hegedus, 2011.
36. J. R. Davis, A. Rohatgi, R. H. Hopkins, P. D. Blais, P. Rai-Choudjury, J. R. McCormick and H. C. Mollenkopf, *IEEE Transactions on Electronic Devices*, 1980, **27**, 677-687.
37. R. H. Hopkins, J. R. Davis, A. Rohatgi, R. B. Campbell, P. D. Blais, P. Rai-Choudhury, R. E. Stapleton, H. C. Mollenkopf and J. R. McCormick, *Effect of Impurities and Processing on Silicon Solar Cells, Phase III Summary and Seventeenth Quarterly Report*, Westinghouse R&D center, 1980.
38. B. L. Sopori, *Journal of Crystal Growth*, 1987, **82**, 228-236.
39. G. Coletti, P. C. P. Bronsveld, G. Hahn, W. Warta, D. Macdonald, B. Ceccaroli, K. Wambach, N. L. Quang and J. M. Fernandez, *Advanced Functional Materials*, 2011, **21**, 879–890.
40. A. Goetzberger and W. Shockley, *Journal of Applied Physics*, 1960, **31**, 1821-1824.
41. D. M. Powell, D. P. Fenning, J. Hofstetter, J. F. Lelièvre, C. d. Cañizo and T. Buonassisi, *Photovoltaics International*, 2012, **15**, 91-100.
42. J. Kang, *Journal of Applied Physics*, 1989, **65**, 2974.
43. R. H. Hopkins and A. Rohatgi, *Journal of Crystal Growth*, 1986, **75**, 67–79.

44. J. Hofstetter, D. P. Fenning, M. I. Bertoni, J. F. Lelièvre, C. d. Cañizo and T. Buonassisi, *Progress in Photovoltaics: Research and Applications*, 2011, **19**, 487-497.
45. A. Haarahiltunen, H. Vainola, O. Anttila, E. Saarnilehto, M. Yli-Koski, J. Storgards and J. Sinkkonen, *Applied Physics Letters*, 2005, **87**, 151908.
46. J. Schon, H. Habenicht, M. C. Schubert and W. Warta, *Journal of Applied Physics*, 2011, **109**, 063717.
47. A. L. Smith, S. T. Dunham and L. C. Kimerling, *Physica B: Condensed Matter*, 1999, **273–274**, 358-362.
48. C. d. Cañizo and A. Luque, *Journal of The Electrochemical Society*, 2000, **147**, 2685-2692.
49. P. S. Plekhanov, R. Gafiteanu, U. M. Gosele and T. Y. Tan, *Journal of Applied Physics*, 1999, **86**, 2453-2458.
50. H. Hieslmair, A. A. Istratov, S. A. McHugo, C. Flink and E. R. Weber, *Journal of The Electrochemical Society*, 1998, **145**, 4259-4264.
51. D. A. Clugston and P. A. Basore, *26th IEEE PVSC*, Anaheim, USA, 1997, 207-210.
52. P. A. Basore, *IEEE Transactions on Electronic Devices*, 1990, **37**, 337-343.
53. A. Bentzen, *J. Appl. Phys.*, 2006, **99**, 093509.
54. T. Y. Tan, R. Gafiteanu, S. M. Joshi and U. Gosele, *Silicon Materials Science and Technology*, 1998, **2**, 1050-1063.
55. A. Haarahiltunen, H. Savin, M. Yli-Koski, H. Talvitie and J. Sinkkonen, *Journal of Applied Physics*, 2009, **105**, 023510.
56. F. S. Ham, *Journal of Physics and Chemistry of Solids*, 1958, **6**, 335-351.
57. The MathWorks, *R2012a Documentation: pdepe*, http://www.mathworks.com/help/techdoc/ref/pdepe.html, Accessed April 10, 2012.
58. R. D. Skeel and M. Berzins, *SIAM Journal on Scientific and Statistical Computing*, 1990, **11**, 1-32.
59. The MathWorks, *R2012a Documentation: ode23,ode45, ode113, ode15s, ode23s, ode23t, ode23tb*, http://www.mathworks.com/help/techdoc/ref/ode15s.html, Accessed April 10, 2012.
60. J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley, 1992.
61. G. Strang, *Computational Science and Engineering*, Wellesley-Cambridge Press, 2007.
62. A. Vande Wouwer, P. Saucez and W. E. Schiesser, *Industrial & Engineering Chemistry Research*, 2003, **43**, 3469-3477.
63. P. Saucez, W. E. Schiesser and A. V. Wouwer, *MATMOL: a Matlab-bases method of lines toolbox with engineering applications*, presented in part at the 5th International Congress on Industrial and Applied Mathematics Sydney, Australia, 2003.
64. W. Kahan, *Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic*, http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF, Accessed April 12, 2012.
65. D. M. Powell, D. P. Fenning, B. S. Conrad, J. Hofstetter, J. F. Lelievre, C. del Canizo and T. Buonassisi, *Photovoltaic Specialists Conference (PVSC), 2011 37th IEEE*, 19-24 June 2011, 2011, 001124-001126.
66. M. Rinio, A. Yodyunyong, M. Pirker, S. Keipert, P. Wang, T. Buonassisi and D. Borchert, *23th European Photovoltiac Solar Energy Conference*, Valencia, Spain, 2008, 1014.

67. M. Rinio, A. Yodyunyong, S. Keipert-Colberg, Y. P. B. Mouafi, D. Borchert and A. Montesdeoca-Santana, *Progress in Photovoltaics: Research and Applications*, 2011, **19**, 165-169.
68. Oracle, *Chapter 10: Signed Applets*, http://java.sun.com/developer/onlineTraining/Programming/JDCBook/signed.html, Accessed May 3, 2012.
69. D. Lowe and B. Burd, *Java All-in-One Desk Reference for Dummies*, Wiley, 2007.
70. S. Clark, *Java Jive: File I/O with Java: It Can be Done!*, http://www.webdeveloper.com/java/java_jj_read_write.html, Accessed May 3, 2012.
71. Oracle, *Reading from and Writing to a URLConnection*, http://docs.oracle.com/javase/tutorial/networking/urls/readingWriting.html, Accessed May 3, 2012.
72. Alvin, *Java HTTP example - Reading and writing to an HTTP server*, http://www.devdaily.com/java/edu/pj/pj010023, Accessed May 3, 2012.
73. Rose India, *Java - Opening a url in new window from an applet*, http://www.roseindia.net/java/example/java/applet/Opening-new-browser-window-from-Applet.shtml, Accessed May 3, 2012.
74. *HOWTO: use Netbeans and ANT to auto-run jarsigner*, http://stoken-tips-and-tricks.blogspot.com/2007/03/howto-use-netbeans-and-ant-to-auto-run.html, Accessed May 3, 2012.
75. The MathWorks, *R2012a Documentation: Startup Options*, http://www.mathworks.com/help/techdoc/matlab_env/f8-4994.html;jsessionid=3030c1df12b3efb5fe2e8b3cf142, Accessed May 3, 2012.
76. M. Hall, *CGI Programming in Java*, http://www.apl.jhu.edu/~hall/java/CGI-with-Java.html, Accessed May 3, 2012.
77. V. G. Gite, *Linux Shell Scripting Tutorial v1.05r3 A Beginner's handbook*, http://www.freeos.com/guides/lsst/index.html, Accessed May 3, 2012.
78. A. Arensburger, *Bourne Shell Programming*, http://www.ooblick.com/text/sh/, Accessed May 3, 2012.
79. W. Books, *Bourne Shell Scripting/Appendix D: Cookbook*, http://en.wikibooks.org/wiki/Bourne_Shell_Scripting/Appendix_D:_Cookbook, Accessed May 3, 2012.
80. Scriptsearch, *Your Programming Directory*, http://www.scriptsearch.com/, Accessed May 3, 2012.
81. *References From Chapter 9 The HTML Sourcebook, 3ed*, http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed/chap9/refs9.html, Accessed May 3, 2012.
82. jGuru, *How can I run a remote program using Runtime.exec()? For example, I'd like to launch a remote program on a Linux box from Windows*, http://www.jguru.com/faq/view.jsp?EID=464743, Accessed May 3, 2012.
83. *Newbie: Intro to cron*, http://www.unixgeeks.org/security/newbie/unix/cron-1.html, Accessed May 3, 2012.
84. H.-T. Geek, *Delete Files Older Than x Days on Linux*, http://www.howtogeek.com/howto/ubuntu/delete-files-older-than-x-days-on-linux/, Accessed May 3, 2012.

85. S. Davis, *Ubuntu LAMP Server – Setup Guide with Desktop GUI*, http://www.zaphu.com/2007/08/07/ubuntu-lamp-server-setup-guide-with-desktop-gui/, Accessed May 3, 2012.
86. S. Davis, *Ubuntu LAMP Server Guide – Configure Apache, mySQL, and cgi-bin*, http://www.zaphu.com/2007/08/21/ubuntu-lamp-server-guide-configure-apache-mysql-and-cgi-bin/, Accessed May 3, 2012.
87. A. Villmann, *How to Setup a Dedicated Web Server for Free*, http://net.tutsplus.com/tutorials/php/how-to-setup-a-dedicated-web-server-for-free/, Accessed May 3, 2012.
88. Ubuntu, *Using The Terminal*, https://help.ubuntu.com/community/UsingTheTerminal, Accessed May 3, 2012.
89. J. Peddicord, *Ubuntu Cheat Sheet*, http://fosswire.com/post/2008/04/ubuntu-cheat-sheet/, Accessed May 3, 2012.
90. Ubuntu, *Fluxbox*, https://help.ubuntu.com/community/Fluxbox, Accessed May 3, 2012.
91. SourceForge, *PC1D*, http://sourceforge.net/projects/pc1d/, Accessed May 3, 2012.
92. W. Wiki, *HowTo*, http://wiki.winehq.org/HowTo, Accessed May 3, 2012.
93. W. Wiki, *Ubuntu and Kubuntu*, http://wiki.jswindle.com/index.php/Ubuntu_and_Kubuntu, Accessed May 3, 2012.
94. Ubuntu, *Wine*, https://help.ubuntu.com/community/Wine, Accessed May 3, 2012.
95. Cplusplus, *C++ Lanugage Tutorial*, http://www.cplusplus.com/doc/tutorial/, Accessed May 3, 2012.
96. Mircosoft Developer Network, *MFC Reference*, http://msdn.microsoft.com/en-us/library/d06h2x6e(v=vs.100).aspx, Accessed May 3, 2012.
97. *CStdioFile Sample*, http://simplesamples.info/Beginners/CStdioFile.php, Accessed May 3, 2012.
98. MIT Photovoltaic Research Laboratory, *Impurity to Efficiency Simulator Version 1.0 User Manual*, 2011.
99. N. E. B. Cowen and C. Ahn, *Photovoltaics International*, 2011, **12**, 72-80.

Appendix

A

IMPURITY TO EFFICIENCY SIMULATOR VERSION 1.0 USER MANUAL

This appendix is reprinted from the I2E Users Manual.[98]

A.1 Quick Start

Open the applet and accept the signed applet warning. Specify the initial conditions for the as-grown concentration of iron throughout the sample. Users can specify a decimal using either (.) or (,) delimiters, and an exponential with either (e) or (E). The user can study multiple parameters by using the Batch Run input areas. The simulator calculates every perturbation of the input variables. Specify the time and temperature profile with a maximum of three steps. Each temperature step has a defined hold temperature, ramp-to time, and hold time. This page also defines the cooling profile after the final temperature step. Temperature profiles with ramp times of 0 minutes are suggested for initial calculations as they execute quickly with high reliability. Define the device architecture on the Device Architecture tab with PC1D input files. All files referenced from the parameter file must also be specified on the tab. If the user prefers, the simulator is able to run without PC1D. A summary of the input file is found in the summary tab. Input data can be saved and loaded using the *Save Input* and *Load Input* buttons.

Select the number of spatial grid points for the solution of the kinetic simulator and lifetime calculator. A larger number of grid points increases accuracy but slows execution time. An optional identifier can be placed in the input file to help a user organize his or her studies. Submit the input file by selecting the *Run* button in the Run Control input panel. After the simulation completes, the results are reported in the Output File panel. The output data can be saved as a text document or copied to the Clipboard. Output plots can be viewed and saved with the *Output Plots* button. Any output data in the Output File area is overwritten when another simulation is submitted. The ability to submit runs will return after a run is either aborted or completed.

A.2 I2E Inputs

The interface is separated into three areas: inputs, run control, and outputs (Figure A.1). The input area is divided into 4 tabs listed below. Users can specify a decimal using either (.) or (,) delimiters, and an exponential with either (e) or (E). Thousands place delimiters, such as 1,000.0 are not allowed.

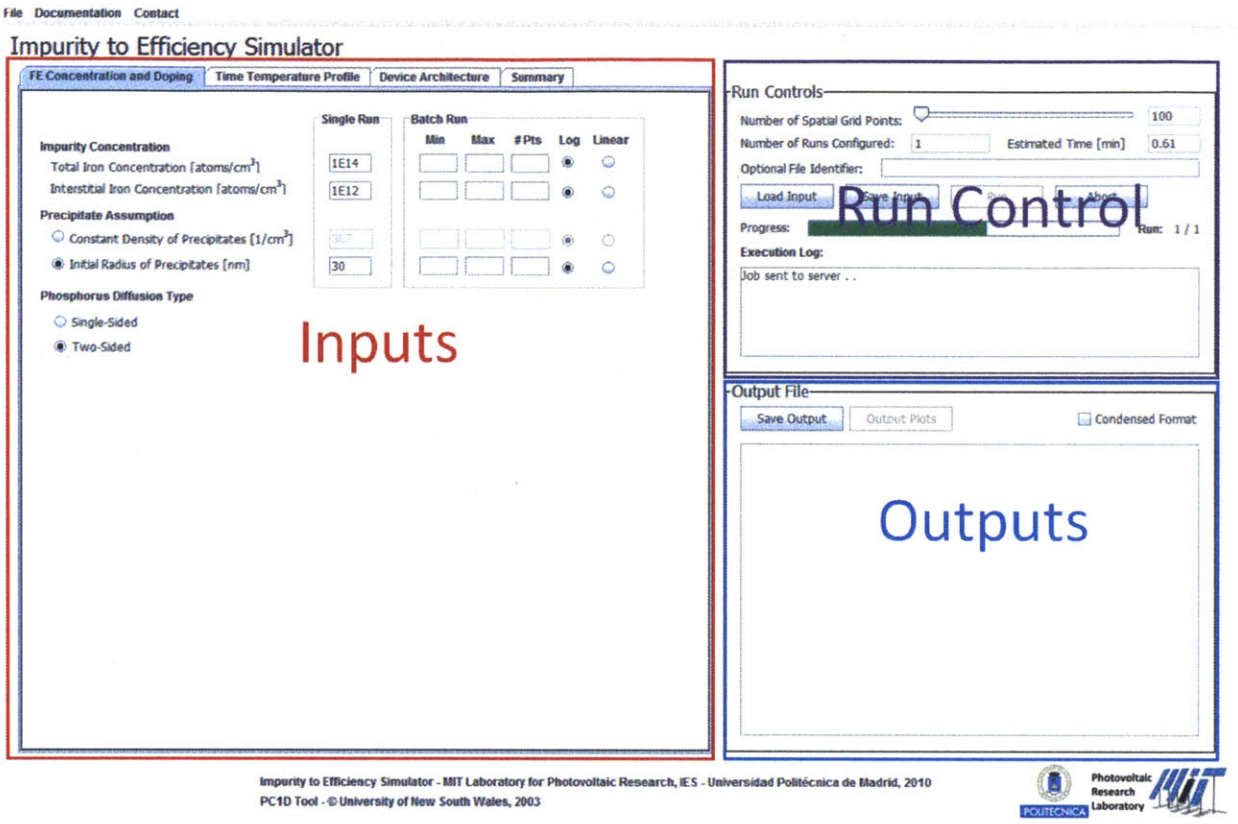

Figure A.1: I2E Applet User Interface Areas

The I2E Applet user interface is divided between the input, run control, and output areas. Figure reprinted from reference.[98]

A.2.1 FE Concentration and Distribution

Users are able to specify the as-grown iron distribution on the FE Concentration and Distribution tab. The total iron concentration must be greater than the interstitial concentration. The kinetic simulator also requires the initial radius of the iron-silicide precipitates to be defined, or an initial density of precipitates. Both single and double sided phosphorus diffusion configurations are allowed by the tool.

Each input variable available to the user can be studied over a wide design space using the Batch Run input areas. A variable can be batched through specifying a minimum, maximum and

a number of points within that range. The mesh points are generated with a linear of logarithmic profile. In this mode, multiple simulation runs are produced and listed as separate line items in the output file. If more than one variable is placed into the batch configuration, then every combination of batched parameters is calculated. The number of simulations executed is limited to 30 with one selection of the *Run* button.

A.2.2 Time Temperature Profile

The time temperature profile used for cell processing is defined in the second tab. Up to three temperature steps, consisting of hold temperature, duration, and ramp time, can be configured (Figure A.2). Discontinuous temperature steps with 0 minute ramp times are permissible and reduce simulation time. The *Temperature* parameter is the plateau temperature of the step. *Ramp Time* is the time taken to ramp to the plateau temperature from the previous temperature. When configuring the first temperature step, room temperature is used as the initial temperature at the beginning of the ramp. The last parameter, *Hold Time*, specifies the duration of the plateau.

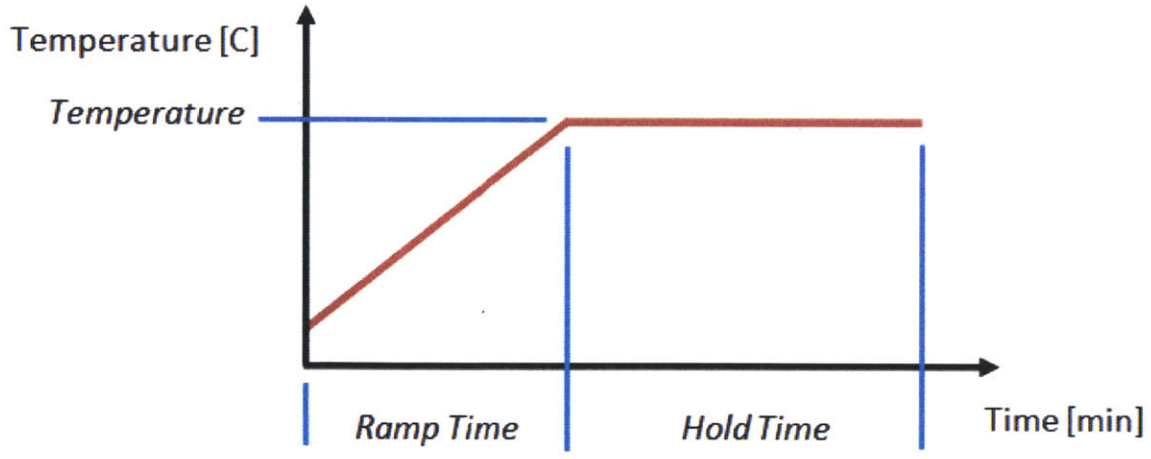

Figure A.2: Time Temperature Profile Nomenclature

The time temperature profile is defined by ramp time, hold time, and temperature. Figure reprinted from reference.[98]

Sample cooling after the plateau of the final temperature step is also defined. If *Air Cool to Room Temperature* is not selected, then the temperature profile will terminate at the final value. The *Exponential Cooling After Final Temperature Step* option provides users with the ability to gradually cool using an exponential profile. The time to cool and the convective cooling rate constant are defined. The resulting temperature profile follows the below equation and is dependent on the time constant (A.1). The time to cool parameter defines the duration of cooling. Therefore, the final temperature experienced by the sample will only be room temperature if the time to cool is greater than approximately 5 time constants. As with the previous high temperature processing steps, execution time is reduced with the utilization of discontinuous temperature profile. All inputs on the tab are rounded to the nearest integer value for the purpose of computation speed.

$$T(t) = (T_{final} - T_{ambient}) e^{-t/\tau} + T_{ambient} \quad (A.1)$$

A.2.3 PC1D Inputs

The I2E simulator can be executed with or without PC1D. If PC1D is not used, the I2E simulator uses the thickness defined by the user to calculate the final iron and minority carrier lifetimes. When using PC1D, the user must specify the file path on their local computer to the parameter file defining the cell architecture and environment under study (Figure A.3). These files will be copied to the server when the *Run* button is selected.

Figure A.3: Completed PC1D Input Example

Users must browse to location of PC1D input files on their local file system. Figure reprinted from reference.[98]

A.2.4 Summary Tab

The summary tab contains a copy of the data file that is sent to the server for computation. The number of values to be simulated for a given parameter is specified as the first number after the parameter name. The remaining values are the value of the parameter to be simulated. Any input errors will be shown on this page with instruction on how to rectify the input.

A.3 Executing the I2E Applet

An I2E simulation is submitted for calculation in the Run Control panel (Figure A.4). The number of spatial grid points determines the accuracy and execution time of the simulation. The execution time scales nonlinearly with the spatial grid. To guarantee numerical stability, the maximum time step used by the kinetic simulator is limited by the density of the spatial mesh (3.14).

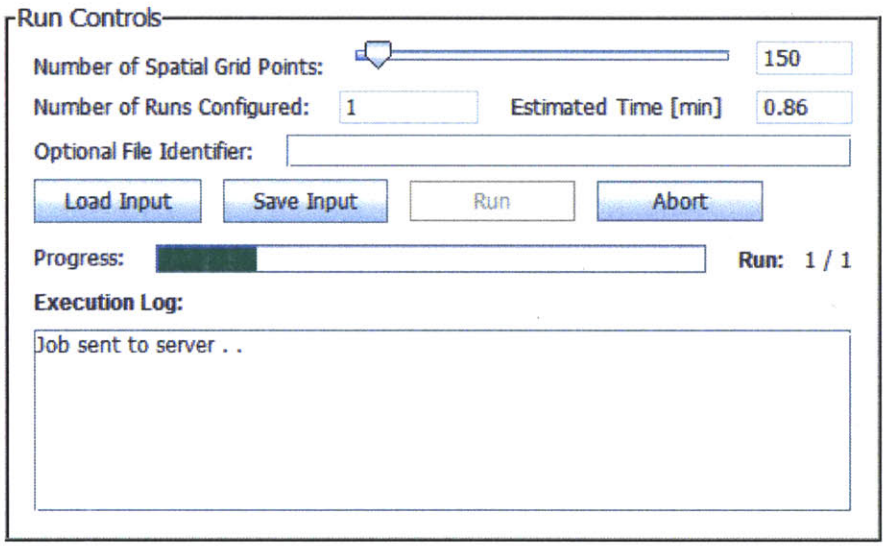

Figure A.4: Run Control Panel

Users are able to control the execution of the simulator in the Run Control Panel. Figure reprinted from reference.[98]

Run time is related to the accuracy of the simulation through the selection of the number of node points. Figure A.5 shows a run parameter curve for low temperature conditions (approximately < 900 °C) when iron precipitates will not begin significantly dissolving to the minimum allowable concentration (3.3.5). The number of nodes can be quite low with acceptable accuracy and short runtimes. Therefore, 100 - 150 nodes are suggested for this type of operation. Figure A.6 shows a convergence curve when for high temperature processing with heavily

dissolved precipitates. These simulations are more computationally intensive and require larger numbers of grid points for successful completion. For acceptable accuracy, approximately 300 - 400 nodes are suggested for this temperature regime. Temperature ramps can also cause difficulties with solution convergence. It is usually best to simulate without temperature ramps with a small number of node points before adding complexity to the run. If a solution fails to converge, a good first step is changing the number of grid points up or down by approximately five nodes. This will likely have the affect of avoiding a singularity at a particular node that caused the solution to fail. Alternatively, substantially more node points might be required to converge to a solution.

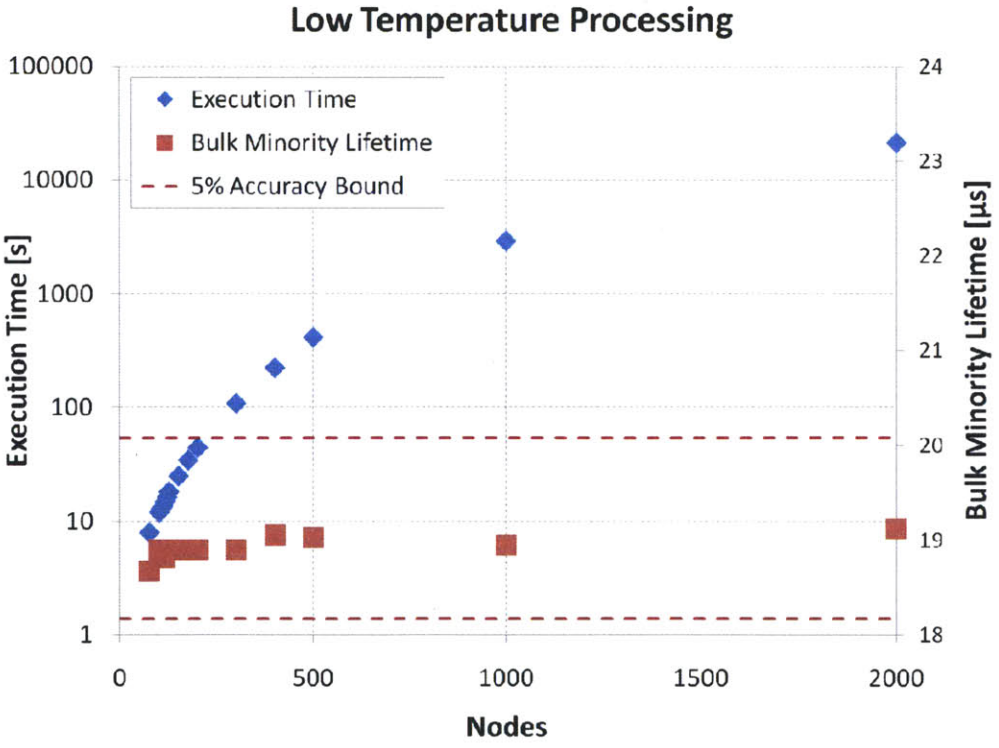

Figure A.5: Run Parameter Plot for Low Temperature (~ <900°C) Operation

A smaller number of nodes can be used when temperatures are not high enough to fully dissolve precipitates. Figure reprinted from reference.[98]

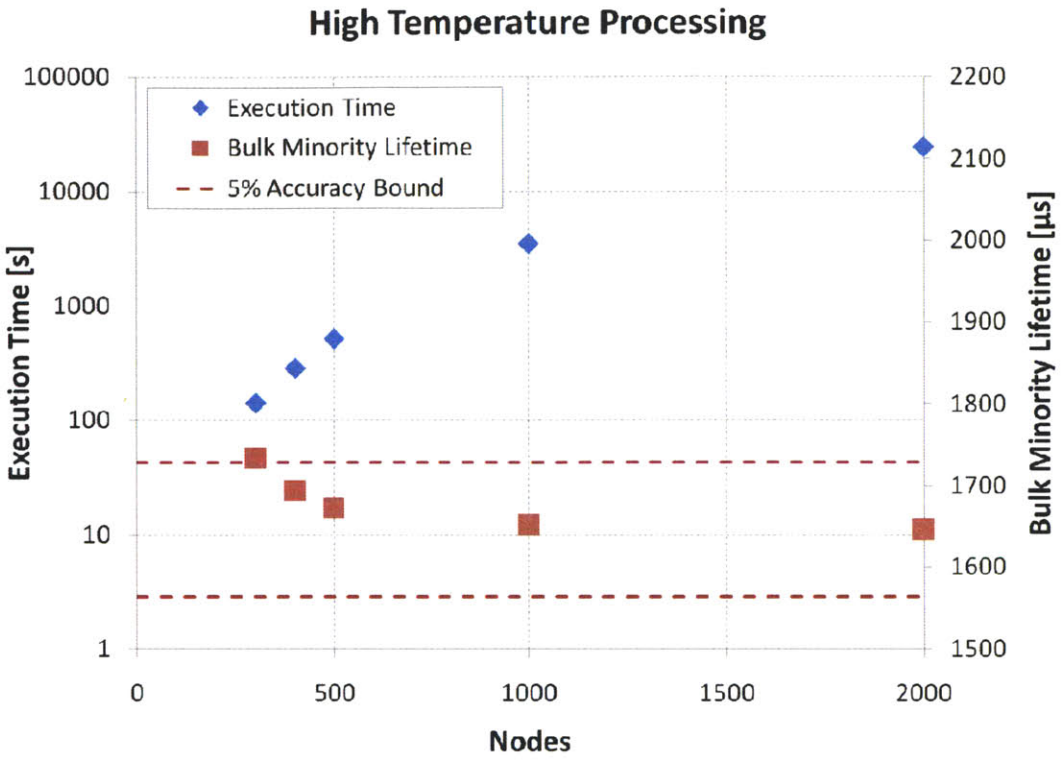

Figure A.6: Run Parameter Plot for High Temperature (~ >900°C) Operation

A greater number of nodes is needed when temperatures are high enough to fully dissolve precipitates. Figure reprinted from reference.[98]

One way to assess the quality of a result is to observe the interstitial iron distribution (Figure A.7). This figure is a good example of a solution that converged without difficulty. If the iron distribution is missing any sections or contains oscillatory behavior the solution is suspect and should be recalculated with a greater number of node points.

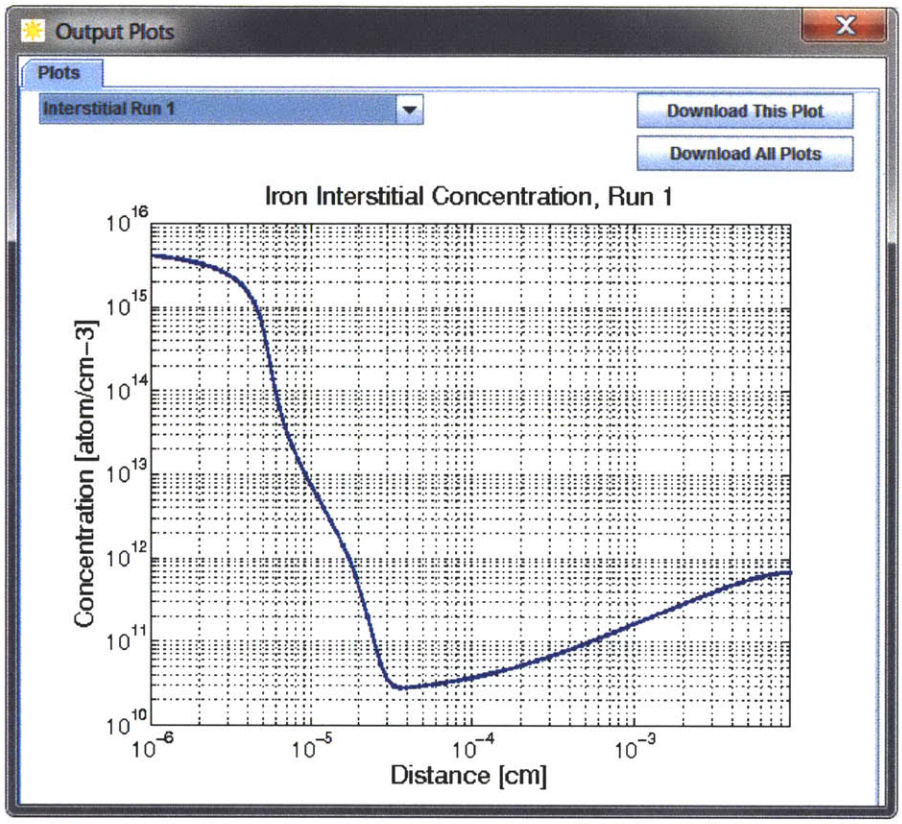

Figure A.7: Output Plots Window

The interstitial iron profile shown displays no evidence of instabilities. Figure reprinted from reference.[98]

The optional identifier text area gives users the ability to place text within the program output files to help organize program runs. The *Save Input* button allows users to save the selections made on the input panels. The *Load Input* button loads previously saved input files. The text area on the Run Control panel provides users with information about what the simulator is currently doing. The panel also contains a progress bar which updates the user with the progress of each simulation. When a run is in the process of being calculated, the user can select the *Abort* button to regain the ability to submit a new run.

A.4 I2E Applet Outputs

When a run is complete, the output file is placed in the text area overwriting any previous data (Figure A.8). The standard output file begins with a reflection of the input variables. The baseline cell performance is listed if PC1D simulations were requested and is followed by the results of each permutation of the input variables. The *Condensed Outputs* option gives the user the ability to suppress the spatially varying data generated from the simulation and only look at the resulting effective aggregate values in tabular form. With *Condensed Outputs* selected, the simulation parameter inputs are organized in columns with the identification tags below.

IC – Initial condition

BC – Boundary condition

R – Result after phosphorus diffusion and thermal cycle

The user can save the text of either file to a local machine by selecting the *Save Output* button or copying it into a text editor. Microsoft Excel can be used to easily interpret the output data by importing the text file into a spreadsheet with tab and colon (:) delimiters. The user interface also allows users to view and save plots created by the simulator (Figure A.7). An individual plot can be saved by selecting the *Download This Plot* button. All plots can be saved with the *Download All Plots* button. If an error occurs with calculation, the user is notified in the Run Control panel and an error file is generated in the Output File panel. Users can use this information to rectify an input file or modify the spatial mesh.

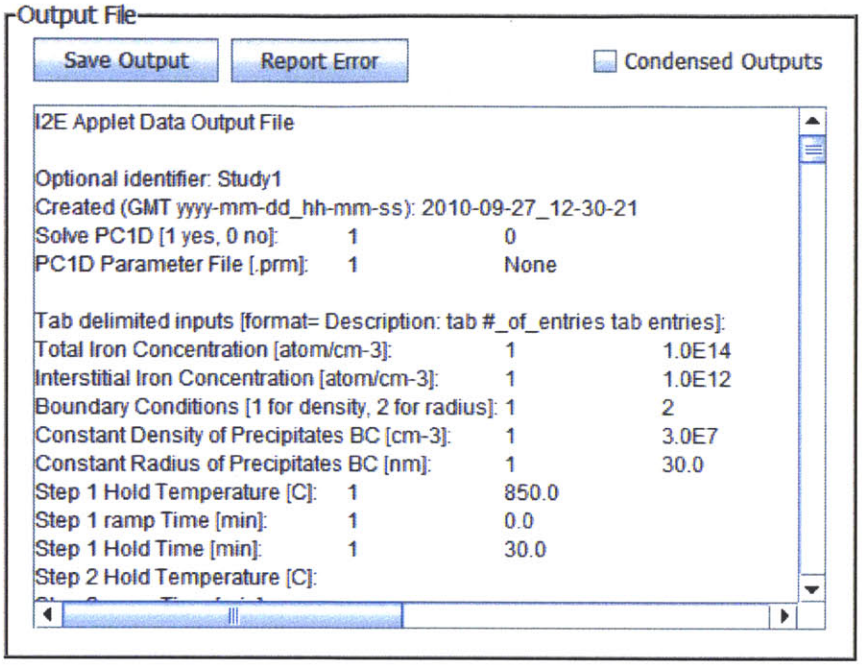

Figure A.8: Output File Panel

The output file can be viewed in the panel and saved to the user's local file system. Figure reprinted from reference.[98]