

SiliconPV: March 25-27, 2013, Hamelin, Germany

A graphical user interface for multivariable analysis of silicon solar cells using scripted PC1D simulations

Halvard Haug^{a*}, Birger R. Olaisen^{a,b}, Ørnulf Nordseth^a, Erik S. Marstein^{a,c}

^a*Institute for Energy Technology, Instituttveien 18, 2007 Kjeller, Norway*

^b*Norwegian Defence Research Establishment, Instituttveien 20, 2007 Kjeller, Norway*

^c*Department of Physics, University of Oslo, 0316 Oslo, Norway*

Abstract

PC1D is an open source computer program for simulation of crystalline semiconductor devices, and is the most commonly used simulation software in the solar cell community today. In this work we present a modified version of PC1D which runs from a command line, thus allowing for scripted simulations using any programming language. In order to increase the accessibility of the new program, we have developed a free-for-use graphical user interface which can be run as a standalone executable. We demonstrate the potential of this program by presenting examples of different simulations. The new features that have been implemented in the new program include multivariable analysis, visualization of three-dimensional data sets, extraction of relevant parameters not directly available in PC1D, utilization of optimization algorithms, fitting to experimental data and implementation of additional features like contact shading, emitter series resistance and passivated rear surfaces with local contacts. Simulations can be performed in a simple and efficient manner, allowing for fast analysis and optimization of cell performance, as well as an enhanced fundamental understanding of the physics of solar cell devices.

© 2013 The Authors. Published by Elsevier Ltd.

Selection and/or peer-review under responsibility of the scientific committee of the SiliconPV 2013 conference

Keywords: Solar cells; simulation; software

1. Introduction

PC1D is one of the most commonly used modelling programs for simulation of crystalline semiconductor solar cells. The program uses a finite-element numerical method for solving the coupled nonlinear equations for carrier generation, recombination and transport in the device and can be used both

* Corresponding author. Tel.: +47 994 86 757.

E-mail address: halvard.haug@ife.no.

for simulation of device performance and as a tool for new users to understand the fundamentals of solar cell physics [1]. The main advantages of PC1D include high calculation speeds, an intuitive user interface and an extensive list of material and physical parameters. By varying the applied bias or the wavelength of the excitation light source, PC1D can calculate both current-voltage characteristics and spectral quantum efficiency of the solar cell, but it also has a large number of other options for output data, both in the spatial domain and in the time domain.

However, PC1D has some limitations in that it is somewhat cumbersome to do parameter variations, and additional calculations cannot be implemented directly by running it from within another program. The source code for the program is however freely available, allowing for user-defined modifications that have the potential to overcome these limitations. In this work we present a new version of the PC1D called *cmd-pc1d* which runs from a command line, thus allowing for scripted simulations using any programming language. In addition, we have also developed a free-for-use program which we have named *PC1D for Matlab*, which has many built-in possibilities for more advanced analysis, using the command line version of PC1D as the basis for the calculations. This program, which is written in Matlab, is modular and easy to use, giving the user the possibility of switching between the original PC1D interface and the new graphical user interface (GUI). The program is available free of charge and can either be run from Matlab or as a standalone executable.

2. Details of the simulation programs

2.1. The *cmd-pc1d* program

While PC1D is a great tool for users wanting to manually experiment with solar cell simulations, the original GUI makes it difficult to use for automated simulation or integration into other systems. The *cmd-pc1d* program was developed to allow access to all aspects of PC1D without having to launch the PC1D GUI. Since the PC1D source code has been made freely available, it has been possible to create a modified version of the program where all GUI code has been removed, and replaced with a simple command line interface called *cmd-pc1d.exe*. The program can be run in two different modes, (i) with a binary .prm parameter file, in which case the program runs the included simulation, or (ii) with both a .prm file and a batch file, which causes the program to run in batch mode. The simulation results are either written to an output file or directly to the command line window, which allows this version of PC1D to be scripted with common scripting tools like Matlab.

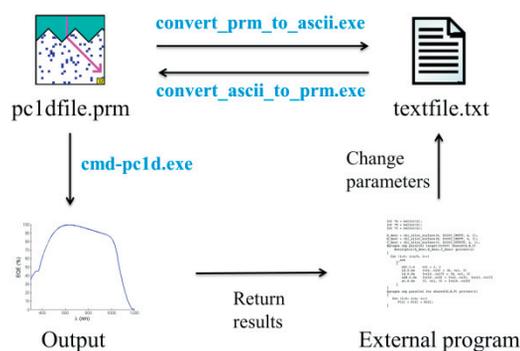


Fig.1. Schematic illustration of a typical workflow using the command-line version of PC1D and the two converter programs (marked in blue text).

In order to control *cmd-pc1d* from another program, the other program needs to create a binary .prm parameter file with specifications and commands that PC1D understands. This can be done with the original PC1D program, but in order to be flexible, two small tools (called *convert_prm_to_ascii.exe* and *convert_ascii_to_prm.exe*) were made to convert the PC1D .prm parameter files between binary format and text format. Fig. 1 shows a typical workflow when using *cmd-pc1d* from an external program. Starting with a base .prm file, which can be made using the original PC1D program, the file is converted to text format. Then individual parameters are modified, as required by the problem to be solved, and the file is converted back again to binary format. Next, *cmd-pc1d* is run with the modified .prm file, and the results are either saved in a text file or output directly into the calling program. Depending on the type of program, the results could be displayed to the user, or a different modification could be done to the parameter file used to run the next iteration.

2.2. The PC1D for Matlab program

Even though many types of calculations are possible with the *cmd-pc1d* program, it requires a high effort for users without knowledge of programming. In order to increase the accessibility of the program to a wider audience, we have created a graphical user interface using the Mathworks Matlab software [2], which in addition to setting up simulations using the *cmd-pc1d* program also implements many of Matlab's features for data analysis, visualization and optimization. The program can be run from Matlab, but has also been compiled as a standalone executable which does not require a Matlab license.

An image of the *PC1D for Matlab* user interface is shown in Fig. 2. A group of menu items and buttons for different actions like loading and saving .prm files, handling simulation results and changing plotting options are placed in the top part of the screen. The user can choose between five different tabs for access to various features of the program, including a complete list of input parameters, single and batch simulation setup, selection of output data, plotting options for batch results and options for importing and fitting simulations to experimental data. In the first tab a subset of the parameters used to define the PC1D prm-file is shown in a scrollable list with descriptive parameters names that can be defined in a separate text file. The user has the option to change to the original PC1D interface at any time. The figure window placed on the right part of screen can be used for plotting PC1D output data, data sets calculated from the output data (e.g. current density vs. applied voltage) or results from batch simulations (e.g. open circuit voltage as a function of bulk carrier lifetime).

The *PC1D for Matlab* program has implemented a range of calculations used both to calculate additional results from PC1D output data and for determining PC1D input parameters based on a list of secondary parameters. A few simulation examples are given in section 3 below, but the rest of calculations are not described in detail in this paper. A list of the different calculations and secondary parameters that are currently included is however shown in Table 1. For a complete and updated description of the different calculations that are included in the program and the necessary settings for each of them, please see the documentation file which is provided at the download page.

In addition to this, the *PC1D for Matlab* program has capabilities for importing external (experimental) data for direct comparison with the simulated results. We have also implemented an Nelder-Mead optimization algorithm available through the *fminsearch()* function in the Mathworks Optimization Toolbox [3], which can be used to fit simulated data to an external curve or to optimize the results with respect to e.g. the cell efficiency. Any set of primary or secondary parameters may be set as fitting variables.

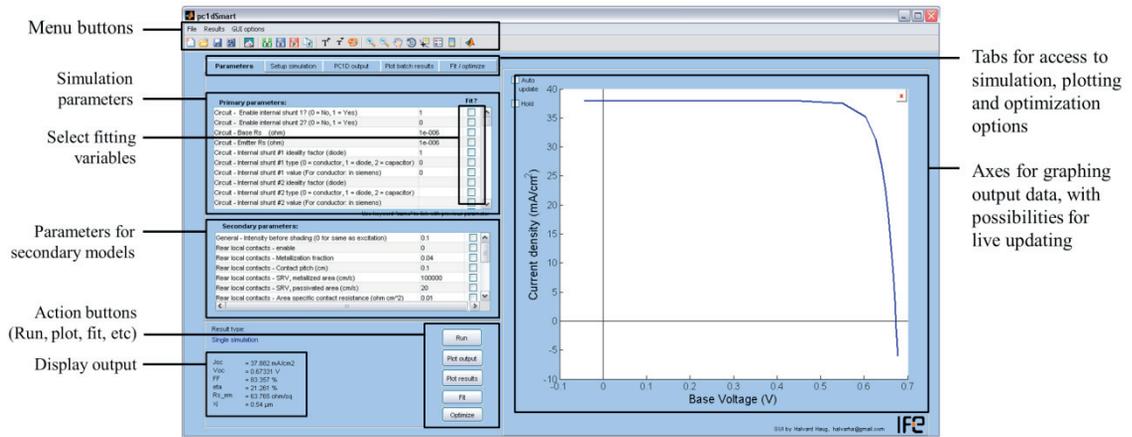


Fig.2. Illustration of the graphical user interface at the time of writing. Several relevant features of the program are indicated.

Table 1. List of different result values that are calculated in the *PCID for Matlab* program and the parameters of the secondary models used to set input values for PCID simulations. Note that changes and additions may occur in future updates of the program.

Results calculated from PCID output data		Secondary parameters used for setting PCID input data
Scalar values	Vector data	
Short circuit current	Current density	Front side metallization grid:
Open circuit voltage	Power density	Contact finger spacing
Fill factor	Global AM1.5 solar spectrum	Contact finger width
Efficiency	Reflectance times solar spectrum	Contact finger thickness
Emitter sheet resistance	EQE times solar spectrum	Contact finger length
Emitter junction depth	IQE times solar spectrum	Busbar width
Steady state carrier lifetime	Excess minority carrier density	Busbar thickness
Steady state generation rate	QSS carrier lifetime	Busbar length
Steady state injection level	Transient carrier lifetime	Metal resistivity
Generalized lifetime at $\Delta n = 10^{15} \text{ cm}^{-3}$	Generalized carrier lifetime	Area specific contact resistance
Sinton optical constant		Local rear contacts:
Integrated weighted reflectance		SRV of metallized area
		SRV of passivated area
		Metallization fraction
		Contact pitch
		Area specific contact resistance

3. Simulation examples

In this section we present different examples of simulations that are possible with the command-line version of PC1D and the accompanying graphical user interface. Note that all these simulations can be performed in a both fast and understandable manner on a standard personal computer. The estimated time used for setting up the simulation in the user interface by a trained user and the approximate simulation time are indicated in the figure captions for each case.

3.1. Optimization of a double layer antireflection coating

In order to illustrate the potential for large scale batch simulations in the *PC1D for Matlab* program, we have simulated a planar Si solar cell while changing the layer thicknesses in a double layer antireflection coating (ARC) placed on a planar Si surface. The two thicknesses were varied between 0 and 150 nm in steps of 2 nm, giving a 76x76 array of result data, with all the characteristics of the I-V curve saved at each point. A constant refractive index of 2.5 and 1.5 was used for the innermost and outermost layer, respectively. Fig. 3 shows a colour map of the short circuit current of the simulated cell, plotted within the *PC1D for Matlab* program. To illustrate the use of the scripted PC1D simulations in an optimization process, we tracked the steps of the optimization algorithm described above and plotted it over the colour map in Fig. 3. The optimization function was used to find the maximum short circuit current with the two layer thicknesses set as free fitting variables.

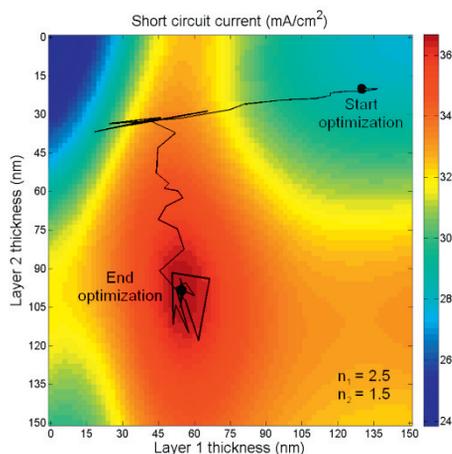


Fig. 3. 2D colour map showing the short circuit current density of a Si solar cell as a function of the two layer thicknesses in a double layer ARC. (5776 points. Setup time: ~30 s. Simulation time: ~10 min). The black lines illustrate the steps of an optimization algorithm used to find the optimal parameters. (Setup time: ~30 s. Simulation time: ~45 s).

3.2. Simulation of a quasi-steady state photoconductance measurement

Quasi-steady state photoconductance (QSS-PC) is a technique commonly used to measure the effective minority carrier lifetime of a Si wafer [4]. The measurement process can be simulated using PC1D by calculating the time dependency of the excess carrier density Δn and optical generation rate G in a passivated Si wafer. For the calculations shown in this paper, the spectral distribution and time-varying intensity of the flash that is mounted in the Sinton WCT-100 measurement setup were used as the

excitation source. The generalized expression for the effective carrier lifetime τ_{eff} can then be calculated as [5]

$$\tau_{eff} = \frac{\Delta n(t)}{G(t) - \frac{d\Delta n(t)}{dt}} \quad (1)$$

When the time-dependent cumulative excess conductivity and cumulative photogeneration have been chosen as output values, the program automatically calculates the effective lifetime following both the quasi-steady state, transient and generalized equations described in Ref. [5]. It thereafter lets the user plot them as a function of the excess carrier density, resulting in the familiar curves of the effective lifetime vs injection level that are obtained from a QSS-PC measurement. The program will also store the effective lifetime value corresponding to an injection level of $\Delta n = 10^{15} \text{ cm}^{-3}$, which can for instance be plotted as a function of one of the variable input parameters in a batch simulation. Fig. 4 (a) shows the effective lifetime vs excess minority carrier density with varying bulk doping concentration N_A .

Because the effective carrier lifetime is calculated from the photogeneration rate in the bulk of the sample, it is important to know the sample thickness and the optical properties of the sample surfaces when performing a QSS-PC measurement. These properties are described in the spreadsheet controlling the measurement as a so-called optical constant, which is defined as the ratio between the photogenerated current in the sample and a reference value of 38 mA/cm^2 . This value depends both on the wafer thickness and the front side reflectance integrated over the spectral distribution of the excitation source, and is generally not straight-forward to calculate. The optical constant is another example of a useful additional output value that is automatically calculated in the *PCID for Matlab* program when the correct output data is chosen. Fig. 4 (b) shows the optical constant for a polished Si wafer as function of the thickness of a coating layer with constant refractive index of 2.1 for different wafer thicknesses. The results were obtained by running a single, two-parameter variation from the user interface.

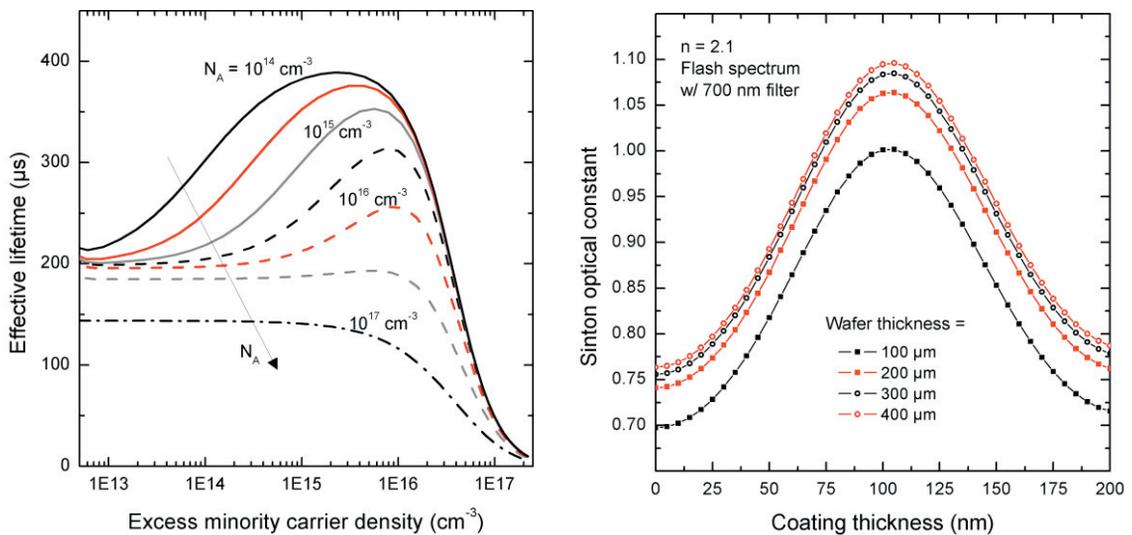


Fig. 4. (a) Simulated QSS-PC curves (effective carrier lifetime as a function of excess carrier density) for different acceptor doping levels. Parameters used for the simulations: Bulk lifetime SRH parameters: $\tau_{0n} = \tau_{0p} = 200 \text{ } \mu\text{s}$, $S_{0n} = S_{0p} = 1 \text{ cm/s}$ on both surfaces, midgap defect levels. The effective lifetime was calculated using the generalized formula described in Ref. [5]. (Setup time: $\sim 30 \text{ s}$. Simulation time: $\sim 30 \text{ s}$). (b) The optical constant (used in a Sinton QSS-PC measurement [3]) for a polished wafer as a function of the thickness of the coating layer for different wafer thicknesses. (Setup time: $\sim 30 \text{ s}$. Simulation time: $\sim 1 \text{ min}$).

3.3. Si solar cell with passivated rear surface and local contacts.

In the last section we showed examples of how additional calculations based on PC1D output data have been implemented into the *PC1D for Matlab* program. Another possibility is to include calculations in the beginning of the process, determining parameters which are then used as input for the PC1D calculations. One example of a case where this approach is relevant is when calculating the rear side surface recombination velocity (SRV) of a passivated Si solar cell with local point contacts. This is normally achieved with Fischer's equation, giving the rear side SRV as [6]

$$S_{eff} = \frac{D}{W} \left(\frac{p}{2W\sqrt{\pi f}} \arctan \left(\frac{2W}{p} \sqrt{\frac{\pi}{f}} \right) - e^{-\frac{W}{p}} + \frac{D}{fWS_{met}} \right)^{-1} + \frac{S_{pass}}{1-f} \quad (2)$$

where D is the minority carrier diffusion constant, W is the wafer thickness, p is the pitch between the contacts, f is the metallization fraction and S_{met} and S_{pass} are the SRVs of the metallized and passivated regions, respectively. These and other parameters used for such secondary models are shown in a separate list in the GUI (see Fig. 2), and can be edited and used for parameter variations and fitting in the same way as the normal PC1D parameters. The GUI also allows for setting a value for the specific contact resistance, which together with the metallization fraction and the cell area is used to determine the total internal series resistance of the base contact. Fig. 5 shows the simulated cell efficiency of a Si solar cell as a function of the rear side metallization fraction for different values of the SRV of the metallized area. For the lowest metallization fractions the cell efficiency drops quickly because of series resistance in the contacts. The efficiency is also reduced at higher metallization fractions for large values of S_{met} caused by the increase in SRV at the rear surface, resulting in a peak in efficiency at approximately 3 % for this particular case.

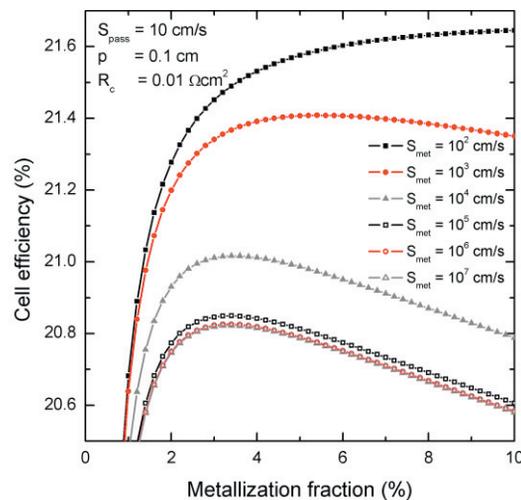


Fig. 5. Simulated cell efficiency of a PERC-type Si solar cell as a function of the rear side metallization fraction for different values of the SRV of the metallized area S_{met} . Values for the SRV of the passivated area S_{pass} , the contact pitch p and the area-specific contact resistance R_c are indicated. (Setup time: ~30 s. Simulation time: ~1 min).

4. Program availability and documentation

The files for the graphical user interface, the command line version of PC1D and the associated conversion programs can be downloaded free of charge from the following URL:

www.ife.no/en/ife/departments/solar_energy/downloads.

The *cmd-pc1d* program runs on 32-bit and 64-bit windows, but versions for the Mac and Linux operating systems may be added in the future. The standalone version of *PC1D for Matlab* program is currently only compiled for 64-bit Windows, and requires the installation of a library of Matlab files called *Matlab Compiler Runtime*, which can be downloaded and installed free of charge (see download page for details).

For more information regarding the various options and calculations implemented in the *PC1D for Matlab* program, readers are referred to the documentation file shown on the download page. New updates will be documented in a separate changelog which can be found at the same location. Please cite this conference paper in addition to the original PC1D publications if you plan to use the software in published material.

5. Summary

In this paper we have presented a modification of PC1D to create a version called *cmd-pc1d* which can be run from a command line. We also present a new graphical user interface for the program, which includes many new features for multivariable analysis, advanced plotting, optimization, fitting to experimental data and implementing additional models and parameters not available in the original PC1D program. The new GUI can be run as a standalone executable without the need for any paid license files, and can be downloaded free of charge.

Acknowledgements

This work has been funded by the Research Council of Norway through the project "Thin and highly efficient silicon-based solar cells incorporating nanostructures," NFR Project No. 181884/S1.

References

- [1] D. A. Clugston and P. A. Basore, "PC1D version 5: 32-bit solar cell modeling on personal computers", Conference Record of the 26th IEEE Photovoltaic Specialists Conference, 207–210, (1997).
- [2] The Mathworks, Inc, "MATLAB 2012b", Natick, Massachusetts, United States.
- [3] The Mathworks, Inc, "Optimization Toolbox User's Guide, 2009". [Online].
<http://www.mathworks.se/help/toolbox/optim/>
- [4] R. A. Sinton, A. Cuevas, and M. Stuckings, "Quasi-Steady-State PhotoConductance, A new method for solar cell material and device characterization", Conference Record of the 25th IEEE Photovoltaic Specialists Conference, p. 457 (1996).
- [5] H. Nagel, C. Berge, and A. G. Aberle. "Generalized analysis of quasi-steady-state and quasi-transient measurements of carrier lifetimes in semiconductors", Journal of Applied Physics 86 (11), 6218–6221 (1999).
- [6] B. Fischer, "Loss analysis of crystalline silicon solar cells using photoconductance and quantum efficiency measurements", Dissertation, Universitaät Konstanz, Konstanz, (2003).