Sesame: A Numerical Simulation Tool for Polycrystalline Photovoltaics

Benoit Gaury[1,2], Yubo Sun[3], Peter Bermel[3], and Paul Haney[1]

Center for Nanoscale Science and Technology, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA)

Maryland NanoCenter, University of Maryland, College Park, MD 20742, USA

School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN, USA

***Abstract* — We present a new software simulation tool "Sesame", which solves the drift-diffusion-Poisson equations in 1 and 2-dimensions. Sesame is distributed both as an open source Python package and as a standalone executable for Windows. The software is designed to enable easy construction of systems with extended defects such as grain boundaries and sample surfaces. In this paper, we present an overview of Sesame's capabilities along with results benchmarking Sesame against commercial software packages. The source code, executable, and full documentation with tutorials is available at https://pages.nist.gov/sesame**

***Index Terms* — numerical simulation, polycrystalline solar cell, software.**

I. Introduction

Numerical simulations play a central role in photovoltaic development and research. Freely available software packages for describing solar cells in 1-dimension have been widely used for decades. Among these include AMPS [1], PC-1D [2], SCAPS [3], wxAMPS [4]. A 1-dimensional description is adequate for systems which are translationally invariant along the directions perpendicular to the *p-n* junction interface, such as single crystal Si or GaAs. However, there are a number of photovoltaic materials with lateral inhomogeneities, such as thin film polycrystalline materials including CdTe and CIGS. The grain boundaries in these materials result in complex geometries that require 2 or 3-dimensional modeling, and which play a key role in device performance [5]-[6]. Free simulation software for 2-dimensional systems is relatively less commonly available, which motivates the development and release of a new software package "Sesame". Sesame is an open source Python package recently developed by the authors (B. G and P. M. H.), and includes documentation and tutorials. In this paper we describe some details of Sesame's usage and benchmark its results relative to commercial software packages. The source code, executable, and full documentation with tutorials is available at https://pages.nist.gov/sesame.

II. PROGRAM DESCRIPTION

Sesame solves the drift-diffusion-Poisson equation in 1 and 2-dimensions, and includes Shockley-Read-Hall, radiative, and Auger recombination mechanisms. Sesame supports variable electronic structure and can describe heterojunctions or graded materials. Sesame is currently limited to describing non-degenerate semiconductors with Boltzmann statistics, and does not include thermionic emission and quantum tunneling at interfaces. These can be important contributions to the transport in heterojunctions [7], so care should be exercised when using Sesame to simulate such systems. Sesame includes Ohmic and Schottky contact boundary conditions, and periodic or hardwall (infinite potential) transverse boundary conditions. Sesame uses finite differences to solve the continuity and Poisson equations, and the standard Scharfetter-Gummel scheme for discretizing the current [8]. Sesame is designed to easily construct systems with planar defects, such as grain boundaries or sample surfaces, which may contain both discrete or a continuum of gap state defects. This enables a description of grain boundaries in polycrystalline solar cells, and the ability to model experiments for which sample surface effects are important (such as Electron Beam Induced Current or Scanning Kelvin Probe Microscopy). Sesame is open source and is distributed under the BSD license.

A. Python scripting

As a Python package, Sesame can be called within any Python script. This mode of use enables users to generate complex systems (e.g. materials with graded electronic properties) and to perform large scale "batch" calculations, where many simulations are distributed on a computing cluster. The number of simulations increases exponentially with number of varied parameters, but access to a cluster enables the efficient computation of many system configurations. Access to a large number of solutions in parameter space is particularly useful for determining the important nonlinearities which control the system behavior. In the distribution, we provide several example scripts which describe standard PV simulations (e.g. J-V, IQE calculations), along with in-depth tutorials. Sesame also includes tools for efficiently saving and loading the results of simulations, and tools for data analysis and plotting.

B. Graphical User Interface

To facilitate Sesame's use by those without Python distributions, we've compiled Sesame into a standalone executable for Windows with a graphical user interface using the PyQt library. Use of the standalone GUI as an alternative to scripting can be more convenient for small-scale calculations. Simulation settings can be saved and loaded, and the standalone GUI also includes a full Python distribution which can be accessed with an interactive Python console.

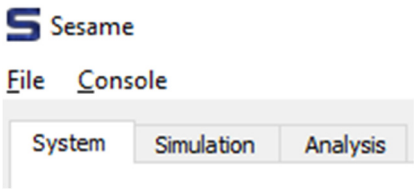

Fig. 1. Menu and table structure of the Sesame GUI.

The GUI is divided into three tabs: 1. The "System" tab contains fields to define the system geometry and material parameters. Fig. 2 shows a portion of the system tab where planar defects are defined. 2. The "Simulation" tab lets the user specify which parameter is varied: either the voltage is swept, or a user-defined variable related to the generation rate density is swept. The boundary conditions and output file information are also set here, and the simulation is launched from this tab. A window on this tab provides details on the calculation progress. 3. The "Analysis" tab enables the user to plot the output of the simulation, and to save and export plotted data. Sesame is distributed with several sample input files for setting up standard PV simulations in the GUI. More detailed documentation for the GUI is included in the distribution.

The standalone executable contains a full Python distribution, and the scripting capability can be accessed by selecting the "Console" option on the main menu (see Fig. 1). This opens an interactive Python console, from which Python scripts can be executed. In addition, the numerical output generated by scripts can be loaded into the GUI for plotting and analysis. The capability of the GUI to run scripts and analyze/plot the output of scripts allows for a flexible usage of the Sesame package, which can be optimized according to the user's needs and experience.

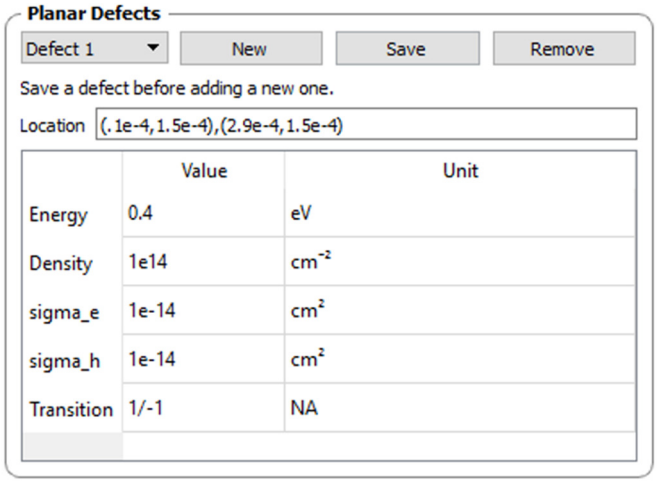

Fig. 2. Portion of the "System" tab of Sesame GUI for specifying planar defects.

III. BENCHMARKS

To verify the accuracy of Sesame, we compared its results with that of Sentaurus [9] and COMSOL's Semiconductor module [10]-[11]. We first consider a 2-dimensional *p-n* homojunction with a single columnar grain boundary (see inset of Fig. 3(a) for a depiction of the geometry). The bulk material parameters are given in Table 1. The *n*-type region is taken to have a thickness of 100 nm with doping 10^{17} cm^{-3}, while the p-type region has thickness 2.9 μm with doping 10^{15} cm^{-3}. The grain boundary parameters are given in Table 1. The grain boundary contains both a donor and an acceptor defect at the given energy level. The contacts are taken to be Ohmic for both majority and minority carries, and we use hardwall transverse boundary conditions. The generation rate density is given as $G(x) = \phi_0 \exp(-\alpha x)$, where $\phi_0 = 10^{17}$ (cm$^2 \cdot$ s)$^{-1}$, $\alpha = 2.3 \times 10^4$ cm^{-1}.

Fig. 3(a) shows the comparison of the illuminated J-V curve computed with Sesame and Sentaurus. To quantify the comparison, we define the relative difference between two computed currents J_1 and J_2 as $|J_1 - J_2|/((J_1 + J_2)/2)$. We find excellent agreement between software packages, with a relative difference of less than 1 % in the computed current. Fig. 3(b) shows the computed band structure along the grain boundary core under short circuit conditions, where we find that the two solutions coincide.

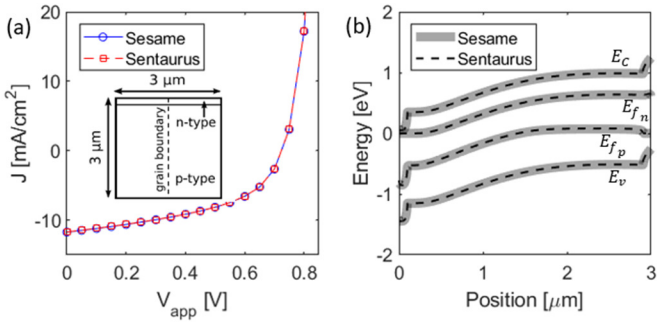

Fig. 3. Comparison between Sesame and Sentaurus for a 2-dimensional homojunction system. (a) is the J-V curve obtained with given software packages under nonuniform (see inset for geometry). (b) is the band diagram along the grain boundary core at short-circuit conditions obtained by the given software packages.

We next consider a 2-d system with an additional grain boundary, which is oriented at an oblique angle with respect to the *p*-*n* junction, and which intersects the initial columnar grain boundary (see inset of Fig. 4(a) for geometry, and the "GB2" row of Table 1 for exact position of the additional grain boundary). We use this system to compare the output of Sesame and COMSOL Semiconductor Module. For this system, we find a less than 2 % difference between Sesame and COMSOL Semiconductor Module in the computed current values. Fig. 4(b) shows the J-V curve on a log scale.

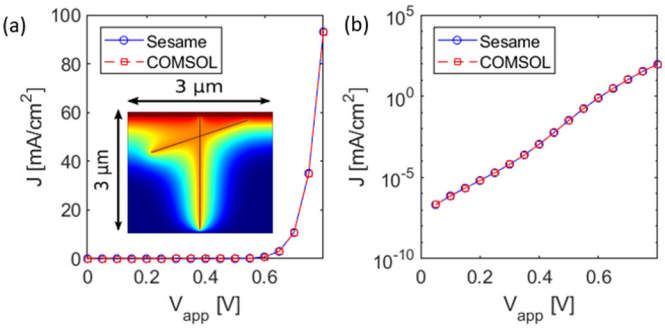

Fig. 4. Comparison between Sesame and COMSOL Semicoductor Model for a 2-dimensional homojunction system, with two grain boundaries. The inset of (a) shows a colormap representation of the electrostatic potential of the system in equilibrium, together with lines indicated the grain boundary positions. (a) is the J-V curve obtained with given software packages under dark conditions. (b) shows the J-V curve on a log scale.

The two examples given here are intended to indicate the veracity and functionality of Sesame. We emphasize that there are several important questions that can be addressed with this tool, such as determining the impact of grain boundary networks on the device efficiency of polycrystalline PV. Previous work has considered the role of grain boundaries on the open-circuit voltage [13], but has not addressed the short-circuit current density or fill factor. It is likely that understanding the impact of grain boundaries on the overall efficiency will require substantial additional analysis.

TABLE I
DEFAULT SIMULATION PARAMETERS

Parameter [units]	Value
ϵ	9.4
τ_n [ns]	10
τ_p [ns]	10
N_C [cm^{-3}]	8×10^{17}
N_V [cm^{-3}]	1.8×10^{19}
E_g [eV]	1.5
χ [eV]	3.9
μ_n [cm^2/(V · s)]	320
μ_p [cm^2/(V · s)]	40
Grain Boundary Parameters	
Defect density [cm^{-2}]	10^{14}
Defect energy [eV]	0.4 (above midgap)
σ_e [cm^2]	10^{-14}
σ_h [cm^2]	10^{-14}
GB1 endpoints (x,y) [μm]	(1.5, 0.1), (1.5, 2.9)
GB2 endpoints (x,y) [μm]	(2.5, 0.2), (0.5, 1.0)

IV. CONCLUSION

We present a free, open source distribution of photovoltaic device modeling software. It is available freely as a standalone executable and as a Python package. We believe that the study of complex geometries of defects within *p*-*n* junctions contains more physics to uncover, and hope that Sesame is a tool that opens this field to a wide class of researchers. The source code is open and documented, and can be easily modified and supplemented according the terms of the BSD license.

REFERENCES

[1] S. Fonash, et al., "A Manual for AMPS-1D for Windows 95/NT", The Pennsylvania State University, 1997.

[2] P. A. Basore and D. A. Clugston, "PC1D Version 4 for Windows: From Analysis to Design", *25th IEEE Photovoltaic Specialists Conference*, Washington, May 1996, pp.377- 381.

[3] M. Burgelman, P. Nollet and S. Degrave, "Modelling polycrystalline semiconductor solar cells", *Thin Solid Films*, vol. 361-362, pp. 527-532, 2000.

[4] Liu, Yiming, Yun Sun, and Angus Rockett, "A new simulation software of solar cells—wxAMPS." *Solar Energy Materials and Solar Cells*, vol. 98, pp. 124-128, 2012.

[5] S. Edmiston, G. Heiser, A. Sproul, and M. Green, "Improved modeling of grain boundary recombination in bulk and p - n junction regions of polycrystalline silicon solar cells." *Journal of Applied Physics*, vol. 80, pp. 6783-6795, 1996.

[6] M. Gloeckler, J. R. Sites, and W. K. Metzger, " Grain-boundary recombination in Cu(In,Ga)Se2 solar cells." *Journal of Applied Physics,* vol. 98, pp. 113704 1-10, 2005.

[7] K. Horio and H. Yanai, " Numerical modeling of heterojunctions including the thermionic emission mechanism at the

heterojunction interface." *IEEE Transactions on Electron Devices*, vol. 37, pp. 1093-1098, 1990.
[8] H. K. Gummel, "A self-consistent iterative scheme for one-dimensional steady state transistor calculations." *IEEE Transactions on Electron Devices*, vol. 11, pp. 455-465, 1964.
[9] S. D. U. Guide and E. Version, Mountain View, CA, USA, 2013.
[10] C. Multiphysics, COMSOL AB, Stockholm, Sweden, 2017.
[11] The full description of the procedures used in this paper requires the identification of certain commercial products. The inclusion of such information should in no way be construed as indicating that such products are endorsed by NIST or are recommended by NIST or that they are necessarily the best software for the purposes described.
[12] J. D. Major, "Grain boundaries in CdTe thin film solar cells: a review." *Semiconductor Science and Technology,* vol. 31, p. 093001, 2016.
[13] B. Gaury and P. M. Haney, "*Charged grain boundaries reduce the open-circuit voltage of polycrystalline solar cells—An analytical description*", Journal of Applied Physics vol. 120, p. 234503, 2016.