



# Solcore: a multi-scale, Python-based library for modelling solar cells and semiconductor materials

D. Alonso-Álvarez<sup>1</sup> · T. Wilson<sup>1</sup> · P. Pearce<sup>1</sup> · M. Führer<sup>1</sup> · D. Farrell<sup>1</sup> · N. Ekins-Daukes<sup>1,2</sup>

Published online: 12 April 2018  
© The Author(s) 2018

## Abstract

Computational models can provide significant insight into the operation mechanisms and deficiencies of photovoltaic solar cells. Solcore is a modular set of computational tools, written in Python 3, for the design and simulation of photovoltaic solar cells. Calculations can be performed on ideal, thermodynamic limiting behaviour, through to fitting experimentally accessible parameters such as dark and light IV curves and luminescence. Uniquely, it combines a complete semiconductor solver capable of modelling the optical and electrical properties of a wide range of solar cells, from quantum well devices to multi-junction solar cells. The model is a multi-scale simulation accounting for nanoscale phenomena such as the quantum confinement effects of semiconductor nanostructures, to micron level propagation of light through to the overall performance of solar arrays, including the modelling of the spectral irradiance based on atmospheric conditions. In this article, we summarize the capabilities in addition to providing the physical insight and mathematical formulation behind the software with the purpose of serving as both a research and teaching tool.

**Keywords** Solar cell modelling · Quantum solvers · Semiconductor properties · Solar irradiance · Optical modelling

## 1 Introduction

Computer-aided design and device models are valuable tools when developing and evaluating photovoltaic solar cells. Laboratory scale tests can be usefully compared against detailed models that account for all relevant processes or with ideal, thermodynamically limited behaviour. Over the years, and with different degrees of sophistication, many pieces of software have been developed and published to tackle different aspects of solar energy research. For example, to calculate the solar spectrum as a function of the atmospheric conditions a traditional solution is to use SMARTS [1]; the light absorption profile in the solar cell or even at module level could be

addressed by OPTOS [2] or OPAL2 [3]; while to solve the transport equations of a solar cell one could use PC1D [4], SCAPS [5] or Quokka [6]. Several free and commercial programs, not specifically designed for solar energy research, have also been used historically, including AFORS-HET [7], Nextnano [8], ATLAS [9] or SENTAURUS [10], with the first two focused on the device and semiconductor properties and the latter two also solving the optics of the solar cells, among other properties. An extensive list of software for solar energy research—both online calculators and downloadable programs—has been compiled by PV Lighthouse [11]. In general, programs like ATLAS and SENTAURUS provide a general purpose, easy to use interface—often solving multi-physics problems, such as electrical transport coupled with thermal transport—to the detriment of performance. On the contrary, specific programs like AFORS-HET or PC1D are extremely fast and efficient, but limited in the problems they can solve, in this case 1D heterostructures and solar cells.

Apart from a few exceptions, such as PVlib [12], all these solvers are high-level, self-contained applications. While users can provide their own inputs and, in some cases, access the source code of the programs and customize some aspects of them, they are not designed with that purpose in mind.

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s10825-018-1171-3>) contains supplementary material, which is available to authorized users.

---

✉ D. Alonso-Álvarez  
d.alonso-alvarez@imperial.ac.uk

<sup>1</sup> Department of Physics, Imperial College London, London, UK

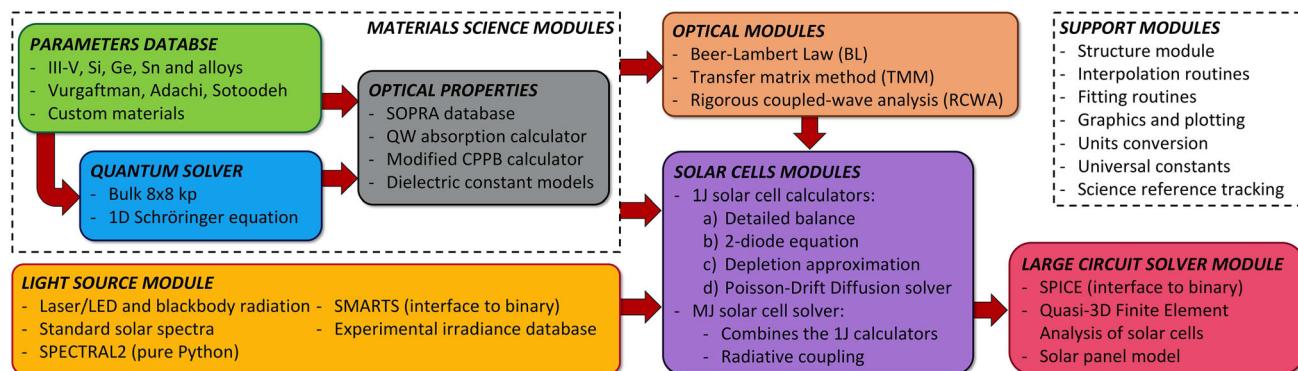
<sup>2</sup> Present Address: School of Photovoltaic and Renewable Energy Engineering, University of New South Wales, Sydney, Australia

Solcore is a multi-scale, modular simulation framework for solar energy research, written mostly in Python. Solcore evolved from SOL, a Fortran-based, quantum well solar cell solver developed by Nelson and Connolly [13], with the explicit purpose of simplifying its integration in other programs, its expansion with custom routines and algorithms, and being didactic and informative. It is a teaching and learning tool as much as a rigorous research tool. Solcore is also extremely flexible. It integrates several algorithms for the multi-scale modelling of semiconductors and solar cells, allowing a user without access to other methods, but with some experience with Python, to solve many different problems out of the box. On the other hand, most of its functionality can be interfaced with external tools, optimized to solve a specific problem, which are more advanced and accurate or that use an approach not considered in Solcore. The most recent version has been released under the GNU Lesser General Public License (GNU-LGPL) and can be found on GitHub [14].

Solcore's capabilities can be grouped into four categories: materials science (Sect. 2), light sources (Sect. 3), solar cells (Sects. 4–6) and large-scale calculators (Sect. 7), each of them tackling a different area and scale relevant for research in solar energy. Figure 1 shows how these parts relate to each other and summarizes some of their content.

## 2 Materials science

The materials science modules in Solcore deal with the retrieval and calculation of material properties as well as those of quantum nanostructures, in particular quantum wells. They form the building blocks necessary to create the structures and calculate the performance of full solar cell devices. While focused on its application for solar cells, this part of Solcore is widely applicable to any research area related to semiconductor materials, as a way of managing the material properties, customizing them and using them in other calculations.



**Fig. 1** General structure and workflow of Solcore

It should be noted that, in reality, electronic and optical properties are not independent but related to each other via the material band structure. As the case in most programs, Solcore uses a non-consistent approach, with independent electronic and optical parameters obtained from different sources. The reader should consider full band structure methods, like pseudopotential or tight-binding, for a consistent set of electronic and optical properties.

### 2.1 Parameters database

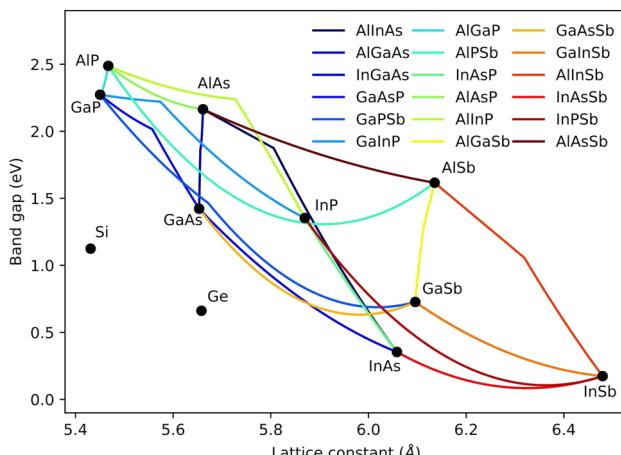
The parameters database contains the basic properties of many semiconductor materials, including silicon, germanium and many III–V semiconductor binary and ternary alloys. Among other parameters, it includes the energy bandgap, the electron and hole effective masses, the lattice constants and the elastic constants.

The main sources of data are the article by Vurgaftman on *Band parameters for III–V semiconductors* [15] and the *Handbook Series on Semiconductor Parameters* by Levinstein et al. [16]. The carrier mobility calculator is based on the empirical low-field mobility model by Sotoodeh et al. [17], and it is available only for some materials where the inputs for the model are available.

There are two methods for retrieving parameters from the database. The first one consists simply of getting the data using the `get_parameter` function with the required inputs. For example:

```
get_parameter("GaAsP", "band_gap", P = 0.45, T = 300)
```

will return the bandgap of GaAsP for a phosphorus concentration of 45% at a temperature of 300 K, equal to 1.988 eV. This method only uses the existing data. Another method is to create a material object which will contain all the properties existing in the database for that material, as well as those included as input, which will override the value of the database parameter, if it exists. The following example cre-



**Fig. 2** Materials with most parameters included in Solcore’s database (excluding the optical properties)

ates a GaAs object and an AlGaAs object, using a custom electron effective mass in the latter:

```
GaAs = material("GaAs") (T=300, Na=1e24)
AlGaAs = material("AlGaAs") (T=300, Al
    = 0.3, Nd=1e23, eff_mass_electron
    = 0.1)
```

Now, any parameter—including the custom ones—are attributes that can be easily accessed and used anywhere in the program. For example, `GaAs.band_gap` is the GaAs bandgap and `AlGaAs.lattice_constant` is the AlGaAs lattice constant, both at the composition and temperature chosen when creating the objects.

Figure 2 shows the well-known bandgap versus lattice constant map of all semiconductor materials and alloys (only ternary compounds) currently implemented into Solcore. However, any other material can be used in all of the Solcore functions, as long as the necessary input parameters are provided. This can be done by overriding all the properties of an existing material during the creation as above, or adding it as an external material in the configuration files.

## 2.2 Optical properties database

In order to calculate and model the optical response of potential solar cell architectures and material systems, access to a library of accurate optical constant data is essential. Therefore, Solcore incorporates a resource of freely available optical constant data measured by Sopra S. A. and provided by Software Spectra Inc. [18]. The refractive index  $n$  and extinction coefficient  $k$  are provided for over 200 materials, including many III–V, II–VI and group IV compounds in addition to a range of common metals, glasses and dielectrics.

Any material within the Sopra S. A. optical constant database can be used with the “material” function described above, but they will have only the optical parameters  $n$

and  $k$ . In the case of materials that are in both databases, the keyword “sopra” will need to be set to “True” when creating the material. Once a material is loaded its  $n$ ,  $k$  and absorption coefficient data are returned by calling the appropriate method, for example `SiO2.n(wavelength)` and `SiO2.k(wavelength)`. For certain materials in the database, the optical constants are provided for a range of alloy compositions. In these cases, any desired composition within the range can be specified and the interpolated  $n$  and  $k$  data are returned. Several examples of materials created from the Sopra database are shown in Listing 1.

```
# Normal GaAs material with all the
# parameters
GaAs = material("GaAs")()

# Sopra version, with only optical
# constants
GaAs_sopra = material("GaAs", sopra=
    True)()

# Ni, Au and SiO2 are only in the Sopra
# database, so there is no need to
# include a flag
Ni = material("Ni")()
Au = material("Au")()
SiO2 = material("SiO2")()

# Creating materials with different
# alloy compositions
AlGaAs_sopra = material("AlGaAs", sopra
    =True)(Al=0.4)
HgCdTe = material("HgCdTe")(Cd=0.25)

# Relaxed SiGe alloys
SiGe25 = material("ReSiGe")(Si=0.25)
SiGe75 = material("ReSiGe")(Si=0.75)
```

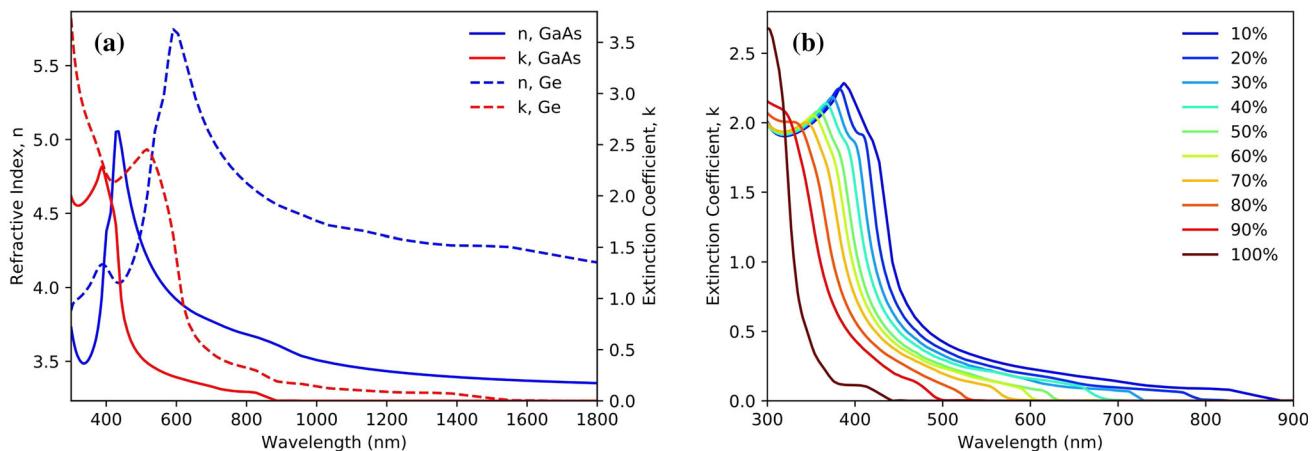
**Listing 1** Creating Solcore materials from the Sopra database.

Figure 3 highlights example output from the Sopra materials library with Fig. 3a showing GaAs and Ge optical constant data and Fig. 3b showing interpolated AlGaAs extinction coefficients for a range of aluminium fractions.

## 2.3 Quantum solver

The electronic band structure of semiconductor materials is responsible for their light absorption and emission properties as well as for many of their transport properties, ultimately depending on the carriers’ effective masses. These properties are not intrinsic to the material, but depend on external factors, too, most notably the strain and the quantum confinement.

Given the crystalline nature of most semiconductor materials, there will be strain whenever two materials with different crystal lattice constants are grown on top of each other pseudomorphically. Even those with the same lattice constant might be under stress due to other effects such as the presence of impurities or if used at different temperatures



**Fig. 3** Example output from the Sopra S. A. optical constant database accessed from Solcore. **a** Refractive index and extinction coefficient data for GaAs (solid lines) and Ge (dashed lines), **b** interpolated AlGaAs extinction coefficient data with aluminium fractions ranging from 10 to 100%

having dissimilar thermal expansion coefficients. Quantum confinement, in turn, takes place when the size of the semiconductor material in one or more dimensions is reduced to a few nanometres. In that situation, the energy levels available to the carriers become quantized in the direction of confinement, also changing the density of states. Both conditions take place simultaneously when dealing with strain-balanced quantum wells (QW).

Quantum wells—and more recently quantum wires—have been employed to tune the absorption properties of high efficiency solar cells for the past two decades. The need for appropriate tools to study them in the context of photovoltaics led to the development of the simulation models that were the seed of Solcore [19–22]. As strained materials with quantum confinement, special care must be taken to obtain a sensible set of parameters for the QW structures, including the band edges with confined energy levels, the effective masses and the absorption coefficient.

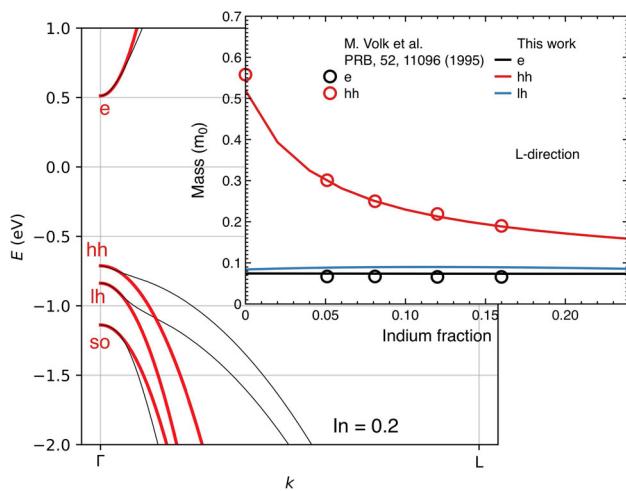
Solcore's approach for evaluating the properties of QWs involves calculating first the effect of strain using a 8-band

Pikus–Bir Hamiltonian (Sect. 2.3.1), treating each material in the structure as bulk, and then using the shifted bands and effective masses to solve the 1D Schrödinger equation, after a proper alignment between layers (Sect. 2.3.2) [23]. Finally, the absorption coefficient is calculated based on the 2D density of states, including the effect of excitons (Sect. 2.5).

### 2.3.1 Bulk 8-band $\mathbf{k}\cdot\mathbf{p}$ calculator

There are many numerical methods to calculate the band structure of a material with a varied degree of sophistication and complexity, such as the tight binding, pseudopotential, Green's function or  $\mathbf{k}\cdot\mathbf{p}$  methods. Solcore includes a modified 8-band Pikus–Bir Hamiltonian to calculate the band structure of bulk materials under biaxial strain [24], considering the coupling between the conduction, heavy hole, light hole and split-off bands. The eigenfunctions  $\Psi$  and eigenstates  $E$  are the solutions of the following equation, where  $\mathcal{H}$  is the Pikus–Bir Hamiltonian:

$$\mathcal{H}\Psi = \begin{pmatrix} E_{cb} & -\sqrt{3}T & \sqrt{2}U & -U & 0 & 0 & -T^* & -\sqrt{2}T^* \\ E_{hh} & \sqrt{2}S & -S & 0 & 0 & 0 & -R & -\sqrt{2}R \\ E_{lh} & -\sqrt{2}Q & T^* & R & -\sqrt{3}S & 0 & \sqrt{3}S & 0 \\ E_{so} & \sqrt{2}T^* & \sqrt{2}R & -U & \sqrt{2}U & -U & -\sqrt{3}S & \sqrt{2}S^* \\ E_{cb} & -\sqrt{3}T^* & E_{hh} & \sqrt{2}S^* & E_{lh} & -\sqrt{2}Q & -S^* & -S \\ E_{hh} & & E_{so} & & E_{so} & & & \end{pmatrix} \Psi = E\Psi \quad (1)$$



**Fig. 4** Band structure of  $\text{In}_{0.2}\text{Ga}_{0.8}\text{As}$  calculated with the bulk  $\mathbf{k}\cdot\mathbf{p}$  solver. The inset shows the effective mass determined for a range of indium fractions and a comparison with the experimental data from Volk et al. [25]

Here, the sub-diagonal elements are just the complex conjugate of the corresponding upper elements. Diagonal terms have three components: the information about the unstrained band edges, a kinematic term and a strain term. As an example, the term  $E_{cb}$  is given by:

$$E_{cb} = E_{c0} + O_k + O_\epsilon \quad (2)$$

$$O_k = \frac{\hbar^2}{2m_0} \gamma_c (k_x^2 + k_y^2 + k_z^2) \quad (3)$$

$$O_\epsilon = a_c (\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) \quad (4)$$

where  $E_{c0}$  is the position of the unstrained conduction band edge,  $\gamma_c$  a modified Luttinger parameter, the  $k_i$  the momenta in the different directions of the reciprocal space,  $a_c$  the conduction band hydrostatic deformation potential and the  $\epsilon_{ij}$ s the strain tensor components. Off-diagonal terms have similar expressions, not involving the unstrained band edges. A detailed description of all these terms and their origin can be found in Tomic [24].

This system is readily solved for the given  $k_i$  using Numpy's `linalg.eig` function, which provides as output the eigenfunctions and the corresponding eigenvalues. Typically, we are interested in the new band edges due to the effect of strain and the resulting effective masses, given by the curvature of the bands near  $k_i = 0$ . Figure 4 shows an example of the bands calculated in this way for the case of a strained InGaAs layer grown pseudomorphically on GaAs, and the resulting dependence of the effective mass with the indium content of the layer. Notice that, due to the effect of strain, the heavy and light hole bands are no longer degenerate at the gamma point  $\Gamma$  ( $k = 0$ ).

### 2.3.2 1D Schrödinger equation

Once the new band edges and effective masses for each of the materials forming the quantum well structure are known, the quantum properties can be calculated by solving the 1-dimensional Schrödinger's equation. Solcore uses the method described by Frenley [26], which allows calculation of the eigenvalues and eigenvectors of an arbitrary potential. However, only closed and periodic boundary conditions are implemented. Solcore does not incorporate the quantum transmitting boundary method (QTBM) described by Frenley, meaning that unbounded states will be, in general, not correct. A tridiagonal matrix is constructed by writing the variable effective mass Schrödinger's equation over a series of mesh points. The eigenvalues of the matrix correspond to the allowed energy levels of the system. Thus, the system of equations to solve over the mesh points is given by:

$$\mathcal{H}\Psi_i = -s_i\Psi_{i-1} + d_i\Psi_i - s_{i+1}\Psi_{i+1} = E\Psi_i \quad (5)$$

where  $s_i$  and  $d_i$  depend on the mesh spacing  $\Delta$  and the position-dependent potential  $V_i$  and effective masses  $m_i$  as:

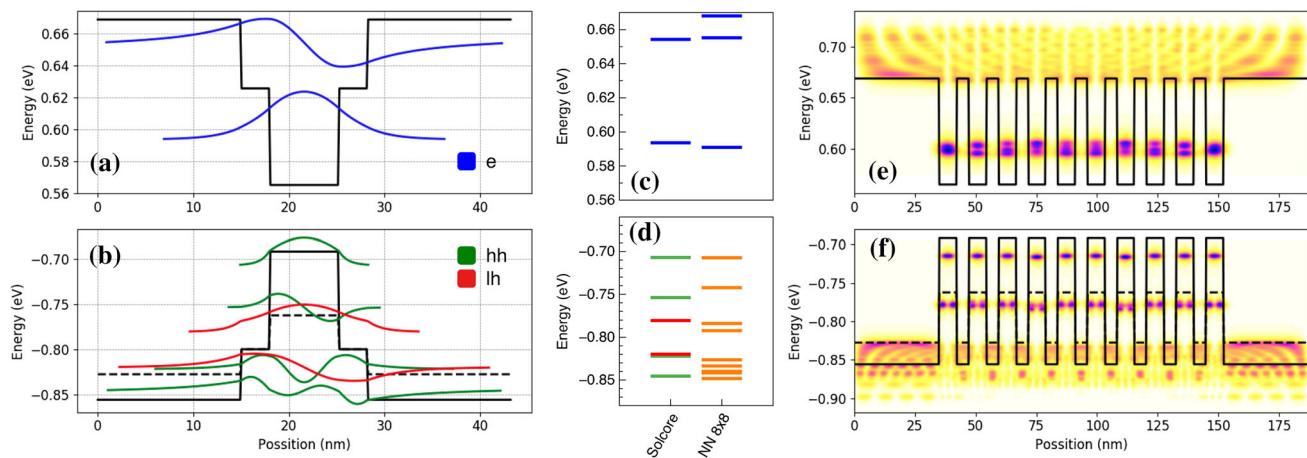
$$d_i = \frac{\hbar^2}{4\Delta^2 m_0} \left( \frac{1}{m_{i-1}} + \frac{2}{m_i} + \frac{1}{m_{i+1}} \right) + V_i \quad (6)$$

$$s_i = \frac{\hbar^2}{4\Delta^2 m_0} \left( \frac{1}{m_{i-1}} + \frac{1}{m_{i+1}} \right). \quad (7)$$

This system is solved using the tools available in the Scipy package for solving sparse linear systems of equations, in particular `sparse.linalg.eigs`.

Figure 5 shows two examples of the band profile and wavefunctions calculated by Solcore. The first one (Fig. 5a, b) is a single InGaAs QW with GaAs interlayers and GaAsP barriers. The strain and quantum confinement shift the light hole valence band (dashed line) with respect to the heavy hole valence band (continuous line). In the GaAsP barriers, under tensile strain, this shift is in the opposite direction to the shift inside the QW, which is under compressive strain and experiences the effects of confinement. Figure 5c, d compares the position of the energy levels predicted by Solcore in this structure with the more rigorous treatment of the 1-dimensional 8-band  $\mathbf{k}\cdot\mathbf{p}$  solver implemented by Nextnano++. As shown, the electron energy levels are barely modified, but hole levels are shifted due to the coupling between the bands [8]. The in-plane dispersion when using a 8-band solver will no longer be parabolic, and we would expect this to have an impact into the absorption coefficient and especially the selection rules for the transitions due to the band mixing effects (see Sect. 2.5).

Finally, Fig. 5e, f shows 1D the local density of states (LDOS) of a multi-QW structure with thin barriers, including a Lorentzian broadening of 5 meV. For the electrons, there



**Fig. 5** **a, b** Band profile of a single 7.2 nm-thick  $\text{In}_{0.15}\text{Ga}_{0.85}\text{As}$  QW with 3 nm GaAs interlayers and  $\text{GaAs}_{0.9}\text{P}_{0.1}$  barriers. **c, d** Comparison of the energy levels calculated by Solcore and using the more rigorous

1D 8-band kp solver implemented in Nextnano [8]. **e, f** 1D-LDOS of a multi-QW structure with 10 coupled QWs, as before but without interlayers

is strong coupling between the QWs, resulting in a range of energies for the ground state. The heavy hole ground states are too deep, resulting in lower coupling between wells. This figure also shows the artefacts in the LDOS of unbound states due to the use of closed boundary conditions rather than QTBM.

## 2.4 Critical-point parabolic-band optical constant model

Understanding the optical response of both established and novel materials is crucial to effective solar cell design. To efficiently model the complex dielectric function of a material, Solcore incorporates an optical constant calculator based on the well-known critical-point parabolic-band (CPPB) formalism popularized by Adachi [27–29]. In this model, contributions to  $\epsilon_2(\omega)$  from critical points in the Brillouin Zone at which the probability for optical transitions is large (van Hove singularities) are considered. The transition probability for such transitions is proportional to the joint density of states (JDOS)  $\mathbf{J}_{cv}(\omega)$ , which describes the number of available electronic states between the valence and conduction bands at given photon energy. The imaginary part of the complex dielectric function is related to the JDOS by:

$$\epsilon_2(\omega) = \frac{4\hbar^2 e^2}{\pi \mu_0^2 \omega^2} |\langle c|p|v\rangle|^2 \mathbf{J}_{cv}(\omega) \quad (8)$$

where  $|\langle c|p|v\rangle|$  is the momentum matrix element for transitions from the valence band ( $v$ ) to the conduction band ( $c$ ). Critical point transitions are considered at the following points of symmetry in the band structure:  $E_0$  corresponds to the optical transition at the  $\Gamma$  point and  $E_0 + \Delta_0$  to the transition from the spin-orbit split off band to the conduction

band at the  $\Gamma$  point.  $E_1$  and  $E_1 + \Delta_1$  denote the transitions from the valence heavy-hole (HH) band and the valence light-hole (LH) band, respectively, to the conduction band at the  $L$  point. The  $E'_0$  triplet and  $E_2$  transitions occur at higher energies, between the HH band and the split conduction bands at the  $\Gamma$  point as well as across the wide gap  $X$  valley. The model also includes contributions from the lowest energy indirect bandgap transition and the exciton absorption at the  $E_0$  critical point. The contributions listed above are summed to compute the overall value of  $\epsilon_2(\omega)$ . The real and imaginary components of the overall complex dielectric function  $\epsilon(\omega) = \epsilon_1(\omega) - i\epsilon_2(\omega)$  are then related via the Kramers–Kronig relations;

$$\epsilon_1(\omega) = 1 + \frac{2}{\pi} \int_0^\infty \frac{\omega' \epsilon_2(\omega')}{(\omega')^2 - \omega^2} d\omega' \quad (9)$$

$$\epsilon_2(\omega) = -\frac{2}{\pi} \int_0^\infty \frac{\epsilon_1(\omega')}{(\omega')^2 - \omega^2} d\omega'. \quad (10)$$

The CPPB model included with Solcore also incorporates a modification to the critical point broadening present in Adachi's description, which is shown to produce a poor fit to experimental data in the vicinity of the  $E_0$  and  $E_1$  critical points [30]. To give a more accurate description of the broadening of the optical dielectric function, Kim et al. [31,32] proposed that a frequency-dependent damping parameter be used to replace the damping constant given by Adachi at each critical point;

$$\Gamma'(\omega) = \Gamma \exp \left[ -\alpha \left( \frac{\hbar\omega - E_0}{\Gamma} \right)^2 \right] \quad (11)$$

where  $\Gamma$  is the damping constant used by Adachi and  $\alpha$  describes the shape of the lineshape broadening with  $\alpha = 0$

producing purely Lorentzian character and  $\alpha = 0.3$  producing a good approximation to Gaussian broadening.

The Solcore module absorption\_calculator contains the CPPB model within the Custom\_CPPB class. The class offers a flexible way to build up the optical constant model by adding individual critical point contributions through the *Oscillator* structure type within Solcore. In addition to the oscillator functions described by Adachi, the Custom\_CPPB class also provides additional oscillator models and the Sellmeier equation for describing the real part of the dielectric function for non-absorbing materials [33]. For example, the code in Listing 2 calculates the complex dielectric function of GaAs.

```
from solcore.absorption_calculator.
Custom_CPPB import Custom_CPPB
import numpy as np

# Generate a list of energies over
# which to calculate the model
# dielectric function.
E = np.linspace(0.2, 5, 1000)

# Class object is created,
CPPB_Model = Custom_CPPB()

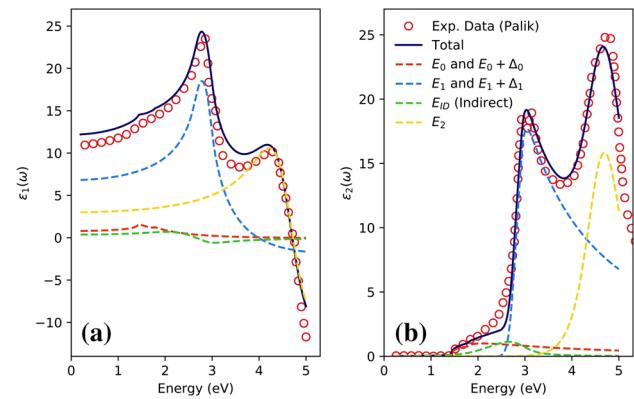
# The MatParams method loads in the
# desired material parameters as a
# dictionary variable.
MatParams = CPPB_Model.
Material_Params("GaAs")

# The oscillator type and material
# parameters are both passed to
# individual oscillators in the
# 'Oscillator' structure.
Adachi_GaAs = Structure([
    Oscillator(oscillator_type="E0andE0_d0", material_parameters=MatParams),
    Oscillator(oscillator_type="E1andE1_d1", material_parameters=MatParams),
    Oscillator(oscillator_type="E_ID", material_parameters=MatParams),
    Oscillator(oscillator_type="E2", material_parameters=MatParams)
])

Output = CPPB_Model.eps_calc(
    Adachi_GaAs, E)
```

**Listing 2** Modelling of the  $n$  and  $k$  based on the CPPB model.

Figure 6 shows the real and imaginary components of the complex dielectric function of GaAs as calculated by the Custom\_CPPB class. The model, using a set of parameters for GaAs similar to those specified in [28], shows excellent agreement with the experimental data taken from Palik [34].



**Fig. 6** Output of the CPPB model provided by Solcore fit to existing experimental data for GaAs, from Palik [34]

For a recent demonstration of Solcore's CPPB model, please refer to the discussion on Wilson et al. [35].

## 2.5 QW absorption calculator

For modelling the optical properties of QWs, we use the method described by Chuang [36]. The absorption coefficient at thermal equilibrium in a QW is given by:

$$\alpha_0(E) = C_0(E) \sum_{n,m} |I_{hm}^{en}|^2 |\hat{e} \cdot \mathbf{p}|^2 \rho_{rmn}^{2D} \times [H(E - E^{en} + E_{hm}) + F_{nm}(E)] \quad (12)$$

where  $|I_{hm}^{en}|^2$  is the overlap integral between the holes in level  $m$  and the electrons in level  $n$ ;  $H$  is a step function,  $H(x) = 1$  for  $x > 0$ , 0 and 0 for  $x < 0$ ,  $\rho_{rmn}^{2D}$  is the 2D joint density of states,  $C_0$  a proportionality constant dependent on the energy, and  $F$  the excitonic contribution, which will be discussed later.

$$C_0(E) = \frac{\pi q^2 \hbar}{n_r c \epsilon_0 m_0^2 E} \quad (13)$$

$$\rho_r^{2D} = \frac{m_{rmn}^*}{\pi \hbar L}. \quad (14)$$

Here,  $n_r$  is the refractive index of the material,  $m_{rmn} = m_{en}m_{hm}/(m_{en} + m_{hm})$  the reduced, in-plane, effective mass and  $L$  an effective period of the quantum wells. The in-plane effective mass of each type of carriers is calculated for each level, accounting for the spread of the wavefunction into the barriers as [37]:

$$m_{\perp} = \int_0^L m(z) |\psi(z)|^2 dz. \quad (15)$$

This in-plane effective mass is also used to calculate the local density of states shown in Fig. 5b. In Eq. 12,  $|\hat{e} \cdot \mathbf{p}|^2$  is

**Table 1** Momentum matrix elements for transitions in QWs

	TE	TM
$c - hh$	$3/2M_b^2$	0
$c - lh$	$1/2M_b^2$	$2M_b^2$
$M_b^2 = m_0 E_p / 6$ is the bulk matrix element		

the momentum matrix element, which depends on the polarization of the light and on the Kane's energy  $E_p$ , specific to each material and determined experimentally. For band edge absorption, where  $k = 0$ , the matrix elements for the absorption of TE and TM polarized light for the transitions involving the conduction band and the heavy and light holes bands are given in Table 1. As can be deduced from this table, transitions involving heavy holes cannot absorb TM polarized light.

In addition to the band-to-band transitions, QWs usually have strong excitonic absorption, included in Eq. 13 in the term  $F_{nm}$ . This term is a Lorentzian (or Gaussian) defined by an energy  $E_{nmx,j}$  and oscillator strength  $f_{ex,j}$ . It is zero except for  $m = n \equiv j$  where it is given by Klipstein et al. [38]:

$$F_{nm} = f_{ex,j} \mathcal{L}(E - E_{nmx,j}, \sigma) \quad (16)$$

$$E_{nmx,j} = E^{en} - E_{hm} - \frac{R}{(j - v)^2} \quad (17)$$

$$f_{ex,j} = \frac{2R}{(j - v)^3} \quad (18)$$

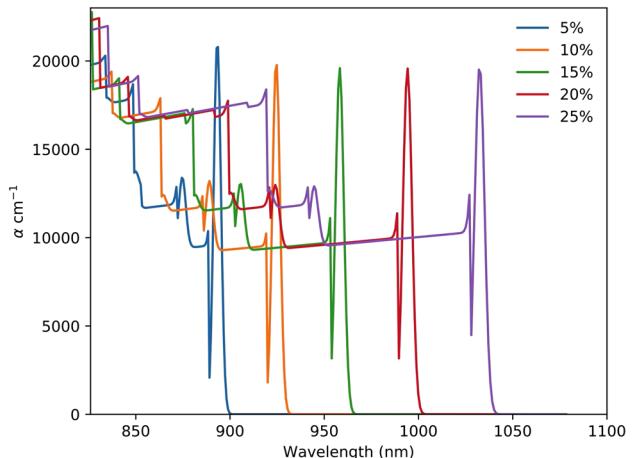
$$R = \frac{m_r q^4}{2(4\pi\epsilon_r\epsilon_0)^2\hbar^2}. \quad (19)$$

Here,  $v$  is a constant with a value between 0 and 0.5 and  $\sigma$  is the width of the Lorentzian, both often adjusted to fit some experimental data. In Solcore, they have default values of  $v = 0.15$  and  $\sigma = 6$  meV.  $R$  is the exciton Rydberg energy [36].

Figure 7 shows the absorption coefficient of a range of InGaAs/GaAsP QWs with a GaAs interlayer and different indium content. Higher indium content increases the depth of the well, allowing the absorption of less energetic light and more transitions.

### 3 Light sources

Transforming sunlight into electricity is the final goal of any solar cell, and it is therefore necessary to have a convenient way of creating, manipulating, and modifying the properties of the spectrum of the light. Ideally, solar cells will be designed and evaluated under a standard solar spectrum—e.g. the air mass 1.5 direct solar spectrum, AM1.5D—but practical light sources are not standard. More often than not,



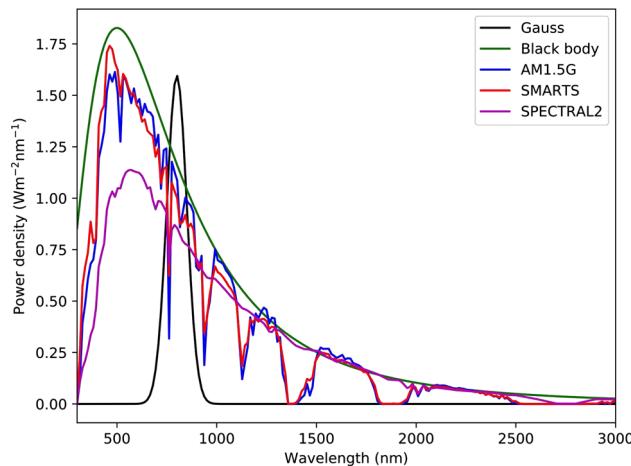
**Fig. 7** Absorption coefficient of a single 7.2 nm-thick InGaAs QW with 3 nm GaAs interlayers and GaAs<sub>0.9</sub>P<sub>0.1</sub> barriers as a function of the indium content

we are interested in modelling the performance of a solar cell under the experimental spectrum of a solar simulator or lamp in a laboratory, simulated data calculated from atmospheric conditions (temperature, humidity, aerosol content, etc.) or even under real irradiance data measured at different locations worldwide. This can then be compared with the experimental performance or tailored to work best under certain conditions.

The Solcore module `light_source` is designed to deal easily with different light sources. It has direct support for:

- Gaussian emission, typical of lasers and light-emitting diodes.
- Black-body radiation, characteristic of halogen lamps defined by a temperature, but also used very often to simulate the spectrum of the Sun, very close to a black body source at 5800 K.
- Standard solar spectra: the extraterrestrial spectrum AM0 and the two terrestrial ones, AM1.5D and AM1.5G as defined by the ASTM G173-03(2008) standard.
- Irradiance models, using location, time and atmospheric parameters to calculate a synthetic solar spectrum. Solcore includes two models: SPECTRAL2, fully implemented in Python, and an interface to SMARTS binaries (which need to be installed separately), which greatly simplifies its use in batch mode.
- User-defined irradiances, provided externally from a database or any other source, allowing for maximum flexibility.

The syntax in all cases is simple and intuitive considering the type of source that needs to be created. In the case of the irradiance models, which often have a large number of inputs, Solcore defines a set of default values, so only those



**Fig. 8** Plot of the spectra of different light sources

that are different need to be provided. The code in Listing 3 illustrates the creation of several light sources using the minimum required input in each case. A plot of those light sources is shown in Fig. 8.

```
import numpy as np
from solcore.light_source import
    LightSource

#The wavelength range of the spectra
wl = np.linspace(300,3000,200)

gauss = LightSource(source_type='laser',
    x=wl,center=800,linewidth=50,power
    =200)
bb = LightSource(source_type='black
    body',x=wl,T=5800,entendue='Sun')
am15g = LightSource(source_type='
    standard',x=wl,version='AM1.5g')
smarts = LightSource(source_type='
    SMARTS',x=wl)
spectral = LightSource(source_type='
    SPECTRAL2',x=wl)
```

**Listing 3** Example of the use of the LightSource class.

Once created, specific parameters of the light sources can be easily modified without the need for creating the source from scratch. That is particularly useful for the irradiance models, where we might be interested in getting the spectrum as a function of a certain parameter (e.g. the hour of the day, or the humidity) without changing the others. For example, `smarts.spectrum(HOUR=11)` and `smarts.spectrum(HOUR=17)` will provide the spectrum of the SMART light source defined above calculated at 11 h and at 17 h, respectively; all additional parameters have the default values. This method has been used to model experimental solar irradiances measured by different spectroradiometers based on the local atmospheric conditions [39].

A final, very convenient feature of the LightSource class is the ability to request the spectrum in a range of different

units. The default is power density per nanometre, but other common units are power density per eV or photon flux per nanometre, among others. While these unit conversions are straightforward, it is often an initial source of errors due to missing constants or incompatible magnitudes.

The `light_source` module has been described in the context of the solar spectrum, but it can be applied broadly where there is spectral data involved, such as the fitting of photoluminescence, electroluminescence, or Raman spectra.

## 4 Optical solvers

The purpose of the optical solvers is to obtain the fraction of incoming light reflected, absorbed, and transmitted in a solar cell as a function of the wavelength of the light and the position inside the structure. Solcore includes three models to tackle this problem: Beer–Lambert law (BL), transfer matrix method (TMM), and rigorous coupled-wave analysis (RCWA). At the moment, Solcore does not have explicit support for light trapping effects using general textured surfaces, which are usually present in silicon solar cells. However, this can be implemented to a large extent using the RCWA method, although not very efficiently. Additionally, the reflected, absorbed, and transmitted light can be calculated externally and then provided as input to Solcore to obtain the electrical properties of a solar cell structure, giving it full flexibility.

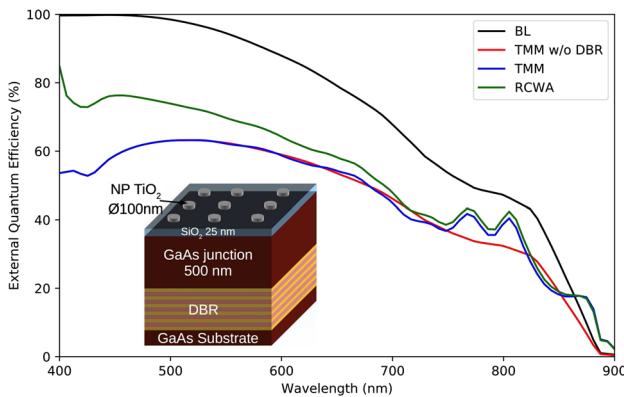
All the optical solvers apply to the solar cell structure as a whole, providing as output the fraction of light reflected ( $R(\lambda)$ ), transmitted ( $T(\lambda)$ ), and absorbed per unit length at a depth  $z$  from the front surface ( $A(\lambda, z)$ ).

Figure 9 shows a comparison of the quantum efficiency of a thin GaAs solar cell with a distributed Bragg reflector (DBR) and an array of  $\text{TiO}_2$  nanoparticles (NP) on top calculated using the three optical solvers described in the following sections. Supplementary Information shows the full code needed to produce these curves [40].

### 4.1 Beer–Lambert law (BL)

This is the simplest model to calculate the absorption in a multi-layer structure. It ignores all reflection at the interfaces—the front surface reflection can be provided externally, and is zero otherwise at all wavelengths—and the absorption per unit length as a function of the wavelength  $\lambda$  and the position  $z$  in layer  $n$  is given by:

$$A_n(\lambda, z) = \alpha_n(\lambda) \exp \left( - \sum_{i=1}^{n-1} \alpha_i(\lambda) w_i - \alpha_n(\lambda)(z - z_n) \right) \quad (20)$$



**Fig. 9** Comparison of the quantum efficiency of a thin GaAs solar cell with a distributed Bragg reflector (DBR) and an array of  $\text{TiO}_2$  nanoparticles (NP) on top calculated using the three optical solvers. The BL model ignores the NP and the DBR and does not include the front surface reflection, overestimating the result at all wavelengths. TMM correctly accounts for the reflection and the DBR but cannot model the effect of diffraction due to the NP layer. RCWA takes into account scattering from the NP, although it is significantly more time consuming

where  $\alpha_n$  is the absorption coefficient of layer  $n$ ,  $w_n$  its thickness and  $z_n$  the position of the beginning of the layer. Due to its simplicity, the BL law is used widely in photovoltaics but in reality it is only applicable when the contrast in the refractive index between layers can be ignored and when there is strong absorption, reducing the effects of light reflection at the interfaces.

#### 4.2 Transfer matrix method (TMM)

In order to evaluate the realistic optical behaviour of a solar cell design, it is important to consider the interaction of incident electromagnetic (EM) radiation with a succession of both absorbing and non-absorbing planar layers. The combined optical response of such a layered structure is crucial when considering the minimization of extrinsic front surface reflection losses [41], the emissivity in the mid-IR of low emissivity coatings for hybrid PV-thermal applications [42] and also when studying the optical constants and layer thicknesses of material using the experimental technique of spectroscopic ellipsometry. Therefore, Solcore evaluates the interaction of incident EM radiation through a layered structure using the TMM. The incident light radiation takes the form of homogeneous, electromagnetic plane-polarized waves and is represented by components describing the electric,  $\mathbf{E}$  and magnetic field strengths,  $\mathbf{H}$ :

$$\mathbf{E} = \mathcal{E} \exp \left[ i\omega t - \left( \frac{2\pi N}{\lambda} \right) z + \varphi \right] \quad (21)$$

$$\mathbf{H} = \mathcal{H} \exp \left[ i\omega t - \left( \frac{2\pi N}{\lambda} \right) z + \varphi' \right] \quad (22)$$

where  $\mathcal{E}$  and  $\mathcal{H}$  denote the electric and magnetic field amplitudes, respectively,  $N$  the complex refractive index,  $z$  the distance in the direction of propagation,  $\omega$  the angular frequency and  $\lambda$  the wavelength of radiation.  $\varphi$  and  $\varphi'$  represent arbitrary phase angles for both the electric and magnetic travelling wave components and are not independent of each other. The characteristic transfer matrix for evaluating the interaction between planar electric and magnetic waves at the interface of  $n$  thin films on a semi-infinite substrate is derived in detail elsewhere [43] and given by the equation:

$$\begin{bmatrix} \mathbf{E} \\ \mathbf{H} \end{bmatrix} = \left\{ \prod_{q=1}^n \begin{bmatrix} \cos(\delta_q) & [i\sin(\delta_q)]/\eta_q \\ i\eta_q \sin(\delta_q) & \cos(\delta_q) \end{bmatrix} \right\} \begin{bmatrix} 1 \\ \eta_m \end{bmatrix} \quad (23)$$

where  $\delta_q$  is defined as the phase factor of the  $q$ th planar layer,  $\eta_q$  the optical admittance of the  $q$ th layer, and  $\eta_m$  the optical admittance of the substrate. The layer closest to the incident medium is evaluated first before working through the  $n$  layer structure in order. The term  $\delta_q$  describes the phase shift required to translate the  $z$  coordinate of the  $E$  and  $H$  interactions by the thickness of each layer,  $q$ . The spectrally varying Fresnel coefficients describing reflection, transmission, and absorption of the multi-layer structure can be calculated from the solutions to Eq. 23 at discrete wavelengths:

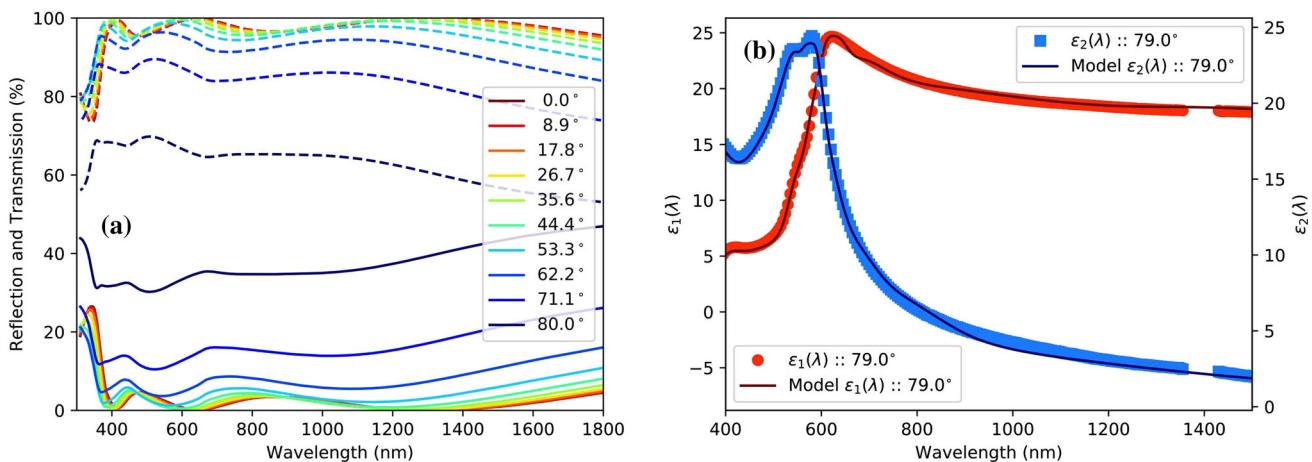
$$R = \left( \frac{\eta_0 \mathbf{E} - \mathbf{H}}{\eta_0 \mathbf{E} + \mathbf{H}} \right) \left( \frac{\eta_0 \mathbf{E} - \mathbf{H}}{\eta_0 \mathbf{E} + \mathbf{H}} \right)^* \quad (24)$$

$$T = \frac{4\eta_0 \operatorname{Re}(\eta_m)}{(\eta_0 \mathbf{E} + \mathbf{H})(\eta_0 \mathbf{E} + \mathbf{H})^*} \quad (25)$$

$$A = \frac{4\eta_0 \operatorname{Re}(\mathbf{E}\mathbf{H}^* - \eta_m)}{(\eta_0 \mathbf{E} + \mathbf{H})(\eta_0 \mathbf{E} + \mathbf{H})^*}. \quad (26)$$

The implementation of the TMM in Solcore uses the freely available `tmm` module developed by Byrnes [44]. The multi-layer optical stack is built up using Solcore's `Structure` object. Some example code evaluating the TMM for a triple layer anti-reflection coating (ARC) on top of conventional multi-junction solar cell materials AlInP and GaInP is included in Listing 4.

```
# The optical stack is built
# defining layer thickness,
# wavelength range and material
# n and k data.
OptiStack = Structure([
    [117, 1240/E_eV, mgf_nk[1],
     mgf_nk[2]],
    [80, 1240/E_eV, sic_nk[1],
     sic_nk[2]],
    [61, 1240/E_eV, zns_nk[1],
     zns_nk[2]],
    [25, 1240/E_eV, alinp_nk[1],
     alinp_nk[2]],
    [350000, 1240/E_eV, gainp_nk
     [1], gainp_nk[2]]])
```



**Fig. 10** Example solutions from the TMM module in Solcore calculating Reflection, Transmission and complex dielectric function. **a** Spectral reflection (solid lines) and transmission (dashed lines) over a range of angles for the optimized triple layer ARC reported in [41], **b**

the calculated real and imaginary parts of the complex dielectric function obtained from spectroscopic ellipsometry of a Ge substrate at 79° (points), and the dielectric constant calculated using TMM for a model consisting of a semi-infinite Ge substrate and 4.5 nm GeO<sub>2</sub> (lines)

```
# The Reflection, Transmission and
# Absorption is evaluated for a range
# of incident
# angles (in degrees).
angles = np.linspace(0, 80, 10)
RAT_angles = []

for theta in angles:
    rat_data = []
    # Calculate RAT data...
    rat_data = calculate_rat(
        OptiStack, angle=theta, wavelength
        =1240 / E_eV)

    RAT_angles.append((theta,
        rat_data["R"], rat_data["A"],
        rat_data["T"]))
```

**Listing 4** Calculation of the reflection, absorption and transmission of a structure.

Figure 10a depicts calculated reflection and transmission from the solutions to the characteristic TMM equation over a range of incident angles for an optimized triple layer ARC design reported in [41]. The solid lines indicate the optical reflection at the front surface, while the dashed lines correspond to the transmitted light into the substrate, in this case taken to be the optically thick top sub-cell material of GaInP.

In addition to using the TMM for calculating the reflection, transmission, and absorption in a multi-layer optical stack, it can also be applied to the popular spectroscopic technique of ellipsometry. This measures a change in the polarization of incident light reflected at the surface of a sample. A more detailed description of ellipsometry and its uses can be found elsewhere [33]. The measured values are expressed as the angles Psi ( $\Psi$ ) and Delta ( $\Delta$ ), which are related to  $\mathbf{R}_s$  and

$\mathbf{R}_p$  (the Fresnel reflection coefficients for *s* and *p*-polarized light, respectively) by<sup>1</sup>:

$$\rho = \frac{\mathbf{R}_p}{\mathbf{R}_s} = \tan(\Psi)e^{i\Delta}. \quad (27)$$

As the ratio between  $\mathbf{R}_s$  and  $\mathbf{R}_p$  is a complex quantity, phase change information is contained within  $\Delta$  in Eq. 27. The change in phase of the reflected electric and magnetic plane waves can be evaluated from the solution of Eq. 23 and is given by:

$$\varphi = \arctan \left( \frac{\text{Im} [\eta_m (\mathbf{E}\mathbf{H}^* - \mathbf{H}\mathbf{E}^*)]}{(\eta_m^2 \mathbf{E}\mathbf{E}^* - \mathbf{H}\mathbf{H}^*)} \right). \quad (28)$$

The complex dielectric function of a sample can be calculated from the experimental ellipsometry results with the solutions to Eq. 27:

$$\langle \epsilon \rangle = \epsilon_1 - i\epsilon_2 = \sin^2(\theta) \left[ 1 + \tan^2(\theta) \left( \frac{1-\rho}{1+\rho} \right)^2 \right]. \quad (29)$$

Some example code calculating the complex dielectric function from the ellipsometric response of a sample of Ge substrate is shown in Listing 5. The output of the model is shown in Fig. 10b and is compared with experimentally obtained data at an incident angle of 79°. Good agreement with the experimental data is observed when a thin 4.5 nm

<sup>1</sup> The definition of the phase  $\Delta$  in the definition of  $\rho$  (Eq. 27), which affects the sign of  $\epsilon_2$  in Eq. 29, is the same here as in [33]. However, other conventions are in use; for instance,  $\Delta$  is defined with the opposite sign in [44].

layer of germanium oxide ( $\text{GeO}_2$ ) is included in the layer model.

```
from solcore.absorption_calculator
    import calculate_ellipsometry
from solcore.structure import Structure

# Input array of energies (in eV).
E_eV = np.linspace(0.7, 4.2, 1000)

Define the optical stack structure, a
piece of Ge substrate with a thin
Ge oxide layer.
# Layer 1 :: GeO2 native oxide layer
# Substrate :: Bulk Ge
OptiStack = Structure([
    [4.5, 1240/E_eV, GeO2_nk["n"],
     GeO2_nk["k"]],
    [350000, 1240/E_eV, Ge_nk["n"],
     Ge_nk["k"]]
])

# Calculate ellipsometry variables, Psi
# and Delta
Exp_Angles = [75, 77, 79]
Out = calculate_ellipsometry(OptiStack,
    1240/E_eV, angle=Exp_Angles)

# From calculated ellipsometry
# variables, Psi and Delta, the
# complex dielectric function
# is computed using equations 25 and
# 27.
rho = lambda psi, delta: np.tan(psi) *
    np.exp(1j * delta)
eps = lambda r, theta: np.sin(theta)**2 *
    (1 + np.tan(theta)**2 * ((1 - r) /
    (1 + r))**2)

# Modelled data...
Mod_rho = rho(np.radians(Out["psi"][:, i]),
    np.radians(Out["Delta"][:, i]))
Mod_eps = eps(Mod_rho, np.radians(
    Exp_Angles[i]))
```

**Listing 5** Example of the calculation of the ellipsometric  $\Psi$  and  $\Delta$ .

### 4.3 Rigorous coupled-wave analysis (RCWA)

Finally, Solcore includes an interface to the  $S^4$  solver (which must be installed separately), developed at Stanford University, in order to model solar cells with nanophotonic designs [45].  $S^4$  is an implementation of RCWA, also sometimes referred to as the Fourier Model Method (FMM), which solves the linear Maxwell's equations in structures containing 2D periodicity. Structures with 2D periodicity can be found in advanced solar cell designs aiming, for example, to reduce the solar cell thickness by scattering the incoming light using a periodic diffraction grating at the front or rear of the absorbing layer [46,47].

$S^4$  defines structures by creating a layer stack of the desired materials using Solcore's *Layer* and *Junction* classes,

in which each layer's composition can be modified by adding circles, rectangles, ellipses, or a generalized polygon made of a specified Solcore material. Each layer is assumed to be infinitely periodic in the  $x$  and  $y$  direction, and uniform in the  $z$  direction. A unit cell must be defined for the whole structure (using the *size* attribute in the user options), and each shape is placed at a specified location (and, where relevant, angular orientation) in the unit cell; as many shapes as necessary can be added, supplied as a list of dictionaries with the relevant parameters for each shape. An example for each type of shape supported by  $S^4$  is shown in Listing 6. The size of the unit cell in the  $x$  and  $y$  directions and the base size of the unit cell must be given in nm. The number of Fourier components used in the calculation [45] must also be specified; this is done using the *orders* attribute in the user options passed to the solver of choice.

```
from solcore import si, material
from solcore.structure import Layer

Air = material('Air')(T=298)
TiO2 = material('TiO2', sopra=True)(T
    =298)

# Define four geometry types, as a list
# of dictionaries: circles,
# rectangles, a general polygon
# (in this case, a triangle) and
# ellipses. All the shapes are made
# of TiO2.
geometry_circles = [{type': 'circle',
    'mat': TiO2, 'center': (100, 100),
    'radius': 50}]
geometry_rectangles = [{type': 'rectangle',
    'mat': TiO2, 'center': (200, 200),
    'angle': 30,
    'halfwidths': (50, 75)}] # rotation
# angle in degrees
geometry_polygons = [{type': 'rectangle',
    'mat': TiO2, 'center': (200, 200),
    'angle': 0,
    'vertices': ((-100, -100), (100,
    -100), (0, 100))}]
# vertices must be defined counter-
# clockwise around the center (origin
# ) for the unrotated polygon
geometry_ellipses = [{type': 'rectangle',
    'mat': TiO2, 'center': (200, 200),
    'angle': 0, 'halfwidths':
    (50, 75)}]

# Define a layer of air with TiO2
# nanopillars 50nm high.
cylinder_layer = Layer(width=si('50nm'),
    material=Air, geometry=
    geometry_circles)
```

**Listing 6** Creating geometry objects for use with the  $S^4$  RCWA solver, which are added to Solcore *Layer* objects using the *geometry* attribute.

## 5 Single-junction solar cells

Solcore includes four solvers to calculate the electrical properties of a single-junction device. In order of increasing accuracy, these are: detailed balance, 2-diode equation, depletion approximation, and Poisson–drift–diffusion.

### 5.1 Detailed balance (DB)

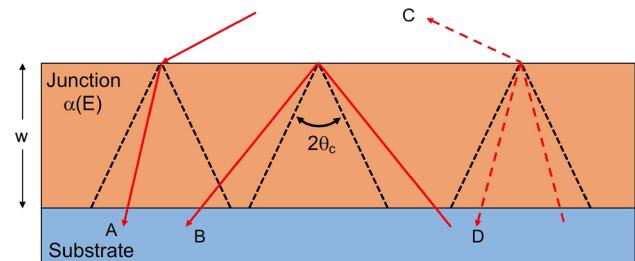
This solver calculates the electrical properties of the junction by balancing the elementary processes taking place in the solar cell, carrier generation, and radiative recombination, using the formalism described by Araújo and Martí [48]. The method is widely used by the photovoltaic community to calculate the limiting conversion efficiencies of the different solar cell architectures or materials. The simplest DB formulation only needs an absorption edge energy and an absorptivity value above that edge. Out of this, the carrier generation and radiative recombination are calculated for different internal chemical potentials, equal to the external electrical bias, in the ideal case. Solcore includes this basic model, but also allows the user to provide a more complex absorption profile.

The radiative recombination or thermal generation current  $J_{\text{rad}}$  from the solar cell is calculated following the formalism described by Nelson et al. [20], considering all the possible paths of the light absorbed by the cell and the reciprocity relation between emission and absorption. The total radiative current, using the generalized Planck equation, is given by:

$$\begin{aligned} J_{\text{rad}}(V, T) &= q \frac{2n^2}{h^3 c^2} \int_0^\infty \frac{E^2}{e^{\frac{E-qV}{k_b T}} - 1} \\ &\quad \times \left[ \int_S A(E, \theta, \mathbf{s}) d\Omega d\mathbf{S} \right] dE \quad (30) \\ &= q \frac{2n^2}{h^3 c^2} \int_0^\infty \frac{E^2}{e^{\frac{E-qV}{k_b T}} - 1} \\ &\quad \times [A_{\text{front}}(E) + A_{\text{back}}(E)] dE \end{aligned}$$

where  $A(E, \theta, \mathbf{s})$  is the probability that a photon of energy  $E$  will be emitted (absorbed) from the point  $\mathbf{s}$  on the surface at an *internal* angle  $\theta$ , and  $A_{\text{front}}(E)$  and  $A_{\text{back}}(E)$  are the combined probability of the photon to be emitted (absorbed) by the front and the back of the cell, respectively.

The different paths of the absorbed light are depicted in Fig. 11. Path A represents light that reaches the front surface within the escape cone ( $\theta < \theta_c$ ) and that crosses the structure. Path B is the light that reaches the back surface outside the escape cone of the front surface ( $\theta > \theta_c$ ), being totally internally reflected and crossing the structure twice. Light reaching the back surface within the escape cone ( $\theta < \theta_c$ ) can either escape through the front (path C) or be reflected



**Fig. 11** All the paths that radiation can follow within the cell, used to calculate the absorptivity $\equiv$ emissivity as a function of the angle

(path D). With these considerations, the contribution to the surface integral of the four terms will be [20]:

$$\begin{aligned} A &= 2\pi S_{\text{front}} \int_{\cos \theta_c}^1 [1 - r(E, \theta)] (1 - e^{-\alpha w / \cos \theta}) \cos \theta d(\cos \theta) \\ B &= 2\pi S_{\text{back}} \int_0^{\cos \theta_c} (1 - e^{-2\alpha w / \cos \theta}) \cos \theta d(\cos \theta) \\ C &= 2\pi S_{\text{back}} \int_{\cos \theta_c}^1 [1 - r(E, \theta)] (1 - e^{-\alpha w / \cos \theta}) \cos \theta d(\cos \theta) \\ D &= 2\pi S_{\text{back}} \int_{\cos \theta_c}^1 r(E, \theta) (1 - e^{-2\alpha w / \cos \theta}) \cos \theta d(\cos \theta). \end{aligned} \quad (31)$$

These equations can be written in a more compact form by noting that  $r = 1$  for  $\theta > \theta_c$ . In this situation,  $B$ ,  $C$ , and  $D$  can be combined and the integral extended from 0 to 1, resulting simply in:

$$\begin{aligned} A_{\text{back}}(E) &= 2\pi \int_0^1 [1 + r(E, \theta) e^{-\alpha w / \cos \theta}] \\ &\quad \times (1 - e^{-\alpha w / \cos \theta}) \cos \theta d(\cos \theta). \end{aligned} \quad (32)$$

The factor  $S_{\text{back}}$  representing the area of the back of the cell has been omitted here as we are interested in the current density, and therefore independent of the area. Likewise,  $A_{\text{front}}(E)$  will be given simply by the component  $A$  in Eq. 31:

$$\begin{aligned} A_{\text{front}}(E) &= 2\pi \int_{\cos \theta_c}^1 [1 - r(E, \theta)] \\ &\quad \times (1 - e^{-\alpha w / \cos \theta}) \cos \theta d(\cos \theta). \end{aligned} \quad (33)$$

$J_{\text{rad}}(V, T_{\text{cell}})$  will represent the radiative recombination of the cell at a bias  $V$  and temperature  $T_{\text{cell}}$ , while  $J_{\text{rad}}(0, T_a)$  will be the carrier generation due to thermal radiation from the ambient at a temperature  $T_a$ . Typically,  $T_{\text{cell}} = T_a$ .

Carrier generation in the solar cell due to the absorption of the solar irradiance  $H(E)$  can be written simply as:

$$J_{\text{sc}} = q \int_0^\infty [1 - R(E)] A_n(E) H(E) dE \quad (34)$$

where  $R = r(E, 0)$  is the normal incidence reflection and  $A_n$  is the normal incidence absorptivity of the cell, given by  $A(E) = 1 - \exp(-\alpha(E)w)$  where  $w$  is the thickness of the junction and  $\alpha$  is the absorption coefficient.

Combining all these equations, the total current as a function of the bias calculated with the DB model will be given by:

$$J = J_{sc} + J_{rad}(0, T_a) - J_{rad}(V, T_{cell}). \quad (35)$$

If  $T_a = T_{cell}$  and  $E \gg k_b T$ , Eq. 35 simplifies, resulting in:

$$J = J_{sc} - J_{01} \left( e^{\frac{qV}{k_b T_{cell}}} - 1 \right) \quad (36)$$

with  $J_{01}$  the reverse saturation current, given by:

$$J_{01} = q \frac{2n^2}{h^3 c^2} \int_0^\infty E^2 e^{-\frac{E}{k_b T_{cell}}} [A_{front}(E) + A_{back}(E)] dE. \quad (37)$$

While in these equations the term  $\exp(-\alpha w)$  is used, it should be noted that none of the two parameters  $\alpha$  and  $w$  are needed as the product is calculated internally by Solcore from the normal incidence absorptivity  $A_n(E)$ , which is the value given as input:

$$e^{-\alpha w} = 1 - A_n(E). \quad (38)$$

## 5.2 2-diode model (2D)

This is the simplest method for simulating the behaviour of a solar cell, using electrical components to model the different transport and recombination mechanisms of the device. The 2D model is widely applied when modelling solar cells at the most engineering end of the topic, when a detailed knowledge of the solar cell structure (layers, absorption coefficient, etc.) are not known or sought. It is often used to fit experimental IV curves and find approximate, general information on the solar cell quality without entering on the fundamental processes. It can provide valuable information to engineers, when designing solar modules for example, or for diagnostic purposes. The complete form of the equation is:

$$\begin{aligned} J &= J_{sc} - J_{01} \left( e^{\frac{q(V-R_s J)}{n_1 k_b T}} - 1 \right) \\ &\quad - J_{02} \left( e^{\frac{q(V-R_s J)}{n_2 k_b T}} - 1 \right) \\ &\quad - \frac{V - R_s J}{R_{sh}}. \end{aligned} \quad (39)$$

Generally, the photocurrent is modelled as a current source ( $J_{sc}$ ), with radiative and non-radiative recombination modelled as two diodes with reverse saturation currents  $J_{01}$  and  $J_{02}$ , and ideality factors  $n_1 \approx 1$  and  $n_2 \approx 2$ , respectively. The shunt resistance  $R_{sh}$  accounts for alternative current paths between the contacts of the solar cell, being infinite in the ideal case, and the series resistance  $R_s$  accounts for the other transport losses. The values of the saturation currents and ideality factors can, ultimately, be calculated from the material properties and device structure, as is done in the depletion approximation model (Sect. 5.4), but the 2D model allows them to be provided directly as input, obtained from a fit to experimental data, for example. They can also be calculated internally, using the DB solver to obtain  $J_{01}$  and  $J_{sc}$ , and then using a radiative efficiency coefficient to obtain  $J_{02}$ . The radiative efficiency  $\eta$  is defined as the fraction of radiative current  $J_{rad}$  at a given reference total current  $J_{ref}$ :

$$\eta = \frac{J_{rad}}{J_{ref}} = \frac{J_{01}}{J_{ref}} \left( e^{\frac{qV_{ref}}{n_1 k_b T}} - 1 \right). \quad (40)$$

The reference voltage  $V_{ref}$  can be written as a function of  $\eta$  and  $J_{ref}$  as:

$$V_{ref} = \frac{n_1 k_b T}{q} \log \left( \frac{\eta J_{ref}}{J_{01}} + 1 \right). \quad (41)$$

On the other hand, the radiative coefficient can also be written as:

$$\eta = \frac{J_{ref} - J_{nrad} - V_{ref}/R_{sh}}{J_{ref}}. \quad (42)$$

Combining Eqs. 41 and 42 and using the expression for the diode with ideality factor  $n_2$ ,  $J_{02}$  can be written as:

$$J_{02} = \frac{(1 - \eta) J_{ref} - V_{ref}/R_{sh}}{e^{\frac{qV_{ref}}{n_2 k_b T}} - 1}. \quad (43)$$

In the common situation of very large shunt resistance and  $V_{ref} \gg k_b T/q$ , this equation further simplifies to:

$$J_{02} = (1 - \eta) J_{ref} \left( \frac{J_{01}}{J_{ref} \eta} \right)^{n_1/n_2}. \quad (44)$$

This process can, of course, be reversed to use knowledge of  $J_{01}$  and  $J_{02}$  at a given reference current to calculate the radiative efficiency of a solar cell, which is useful to compare different materials, technologies or processing methods. This was done by Chan et al. using  $J_{ref} = 30 \text{ mA/cm}^2$ , obtaining  $\eta$  values of 20% for InGaP, 22% for GaAs, and 27% for InGaAs devices [49]. It should be pointed out that this method is only valid under the assumption that  $J_{01}$  corresponds only to

radiative recombination and  $J_{02}$  only to non-radiative recombination, which is generally true for QW solar cells and some III–V solar cells, like those made of GaAs or InGaP, but not for Si or Ge, for example. Other definitions of the radiative efficiency are based on the external quantum efficiency, the  $I_{sc}$  and  $V_{oc}$  of the cell, as described by Green [50].

Despite the simplicity of the 2-diode model, it is very useful to guide the design of new solar cells and explore the performance of new materials, such as dilute bismuth alloys [51], or to assess the performance of large arrays of solar cells, as will be shown in Sect. 7 [52].

### 5.3 Poisson-drift-diffusion (PDD)

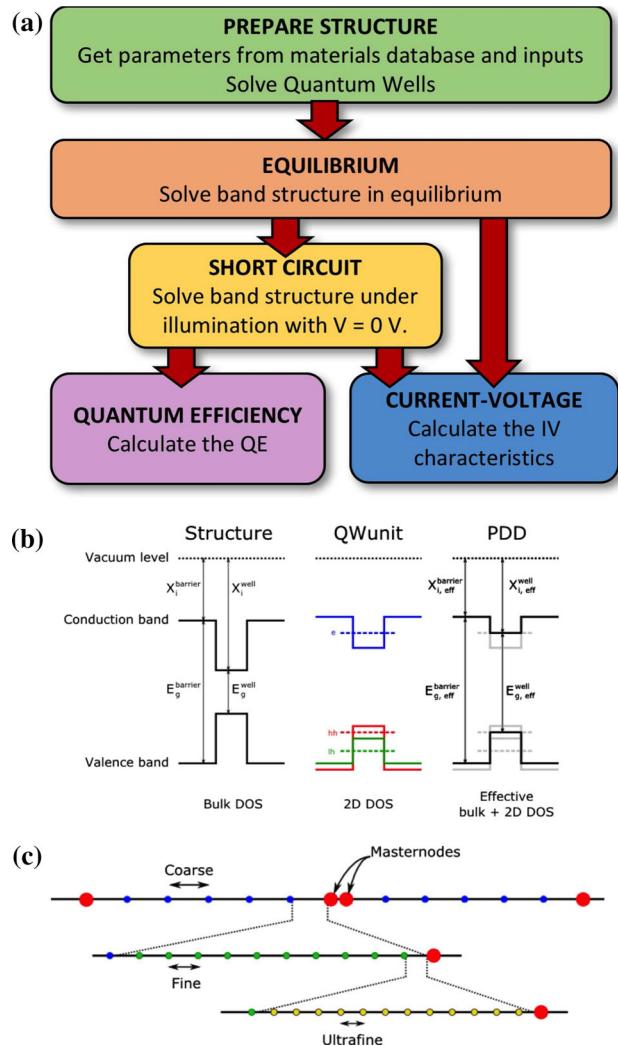
This method solves the Poisson equation for the electrostatic potential coupled with the transport equations for electrons and holes and suitable boundary conditions in the steady state. It is the standard method for calculating the electrical properties of most semiconductor devices, including solar cells, transistors, or light-emitting diodes. It is also the only method included in most software packages for simulating semiconductor devices, such as PC-1D, Nextnano, SCAPS and AFORS-HET.

Figure 12a shows the flow chart of Solcore's PDD solver, which currently only solves the time-independent PDD equations (steady state). Any simulation starts by calculating the band structure under equilibrium conditions (no illumination or bias). If the simulation includes illumination, the photogeneration as a function of the position in the structure is calculated externally to the PDD solver using any of the models described in Sect. 4. To aid convergence, the solution at short circuit conditions is calculated by increasing the light intensity from zero to the nominal value in small steps. Similarly, the solution at any bias is obtained by solving the problem first at zero bias and then increasing it in small steps, using the previous solution as the initial condition for the next one. Re-meshing is performed several times during the simulation of the current-voltage characteristics (see Sect. 5.3.3).

To calculate the internal quantum efficiency (IQE), a small differential increase is included in the photogeneration profile as a function of wavelength. The IQE is then calculated as the ratio between the resulting increase in the photocurrent and the increase in the photogeneration at that wavelength. This procedure is comparable to the actual experimental measurement of the quantum efficiency.

#### 5.3.1 Solver assumptions and formulation

The Poisson's and drift–diffusion equations relate the electrostatic potential created by the free and fixed charges with the carrier densities and their variation across the structure due to generation, recombination and externally applied bias.



**Fig. 12** **a** Work flow of Solcore's PDD solver, **b** process of obtaining the effective band structure of QWs to use in the PDD solver. From left to right: simple sequence of layers; band profile and energy levels after considering the strain and quantum confinement; effective band structure, **c** description of the inhomogeneous mesh scheme used in Solcore

The reader is referred to any semiconductor textbook for a detailed description of their derivation (see for example [53] or [54]). The solver uses the Boltzmann approximation for the carrier distribution with the following assumptions:

- All carrier populations are in quasi-thermal equilibrium.
- The mobility of carriers is independent of the electric field.
- Temperature is uniform.
- There are no magnetic fields.

As a consequence of the field-independent mobility, Solcore's PDD solver will be valid only in situations where electric field is not very strong. Poisson's equation relates

the electrostatic potential  $\phi$  and the electrical charges in the structure. In one dimension, it can be written as:

$$\frac{d}{dx} \left( \epsilon \frac{d\phi}{dx} \right) + q(p - n + N_D - N_A) = 0 \quad (45)$$

where  $N_A$ ,  $N_D$ ,  $n$ , and  $p$  are the density of ionized acceptors, donors, the density of free electrons and holes, respectively, and  $\epsilon$  the dielectric constant. The current density equations account for the movement of carriers due to the electric field (drift component) and the carrier concentration gradient (diffusion component):

$$J_n = q\mu_n \left( nF + \frac{k_b T}{q} \frac{dn}{dx} \right) \quad (46)$$

$$J_p = q\mu_p \left( pF + \frac{k_b T}{q} \frac{dp}{dx} \right) \quad (47)$$

where  $\mu$  is the carrier mobility and  $F = -d\phi/dx$  the electric field. Finally, the continuity equations ensure particle conservation, balancing the carriers that enter and leave any point of the structure. Under steady state, this means that the variation of the current must be equal to the generation  $G$  and recombination  $R$  processes, which are equal for electrons and holes since they are created and annihilated in pairs:

$$\frac{dJ_n}{dx} + qG - qR = 0 \quad (48)$$

$$-\frac{dJ_p}{dx} + qG - qR = 0. \quad (49)$$

Combining Eqs. 46–49 gives:

$$q\mu_n \left( \frac{k_b T}{q} \frac{d^2n}{dx^2} + F \frac{dn}{dx} + n \frac{dF}{dx} \right) + qG - qR = 0 \quad (50)$$

$$q\mu_p \left( \frac{k_b T}{q} \frac{d^2p}{dx^2} - F \frac{dp}{dx} - p \frac{dF}{dx} \right) + qG - qR = 0. \quad (51)$$

Poisson's equation (Eq. 45) and the continuity equations (Eqs. 50, 51), together with the definitions for  $n$ ,  $p$  and  $R$ , represent the complete system that needs to be solved in order to obtain the performance of the solar cell. The PDD solver included in Solcore uses the same discretization scheme used by PC-1D [4,55] taking as independent variables the electrostatic potential  $\phi$  and the quasi-Fermi potentials for electrons and holes,  $\phi_n$  and  $\phi_p$ , respectively. These three variables are continuous across the whole structure and have comparable magnitudes in the voltage range. Details of the discretization process are included in [4,55], but they are based on minimizing the total electrostatic energy in the case of the Poisson's equation, and in the Scharfetter–Gummel discretization scheme for the drift–diffusion equations [56].

The bulk recombination models included in Solcore are Shockley–Read–Hall recombination, radiative recombination, and Auger recombination, as well as a surface recombination velocity model for the recombination at the contacts. The three recombination models are given by the following equations, respectively:

$$R_{SRH} = \frac{pn - n_i^2}{\tau_n(p + n_i) + \tau_p(n + n_i)} \quad (52)$$

$$R_{RAD} = B(pn - n_i^2) \quad (53)$$

$$R_{AUG} = (C_n n + C_p p)(pn - n_i^2) \quad (54)$$

with  $\tau_n$  and  $\tau_p$  the non-radiative lifetimes of electrons and holes,  $B$  the radiative recombination coefficient and  $C_n$  and  $C_p$  the Auger recombination coefficient for electrons and holes. By default, the radiative recombination coefficient is calculated internally by Solcore based on the absorption coefficient, as described by Nelson [54], and given by:

$$B = \frac{1}{n_i^2} \frac{2\pi}{h^3 c^2} \int_0^\infty n_s(E)^2 \alpha(E) E^2 e^{-\frac{E}{k_b T}} dE \quad (55)$$

where  $n_i$  is the intrinsic carrier concentration,  $n_s$  the refractive index and  $\alpha$  the absorption coefficient.

**At the moment, Solcore's PDD solver cannot include interface charges or bandgap narrowing due to heavy doping and only implements Ohmic contacts.** Additionally, it only includes local carrier recombination processes and therefore cannot deal with tunnel transport, which relates remote nodes.

This is the only part of Solcore implemented in Fortran using quadruple precision variables in order to increase the numerical accuracy and improve convergence.

### 5.3.2 QWs in the PDD solver

Quantum wells have been developed in the context of solar cells mainly to tailor the absorption edge of the sub-cells in multi-junction devices to their optimum values [57]. Typically, achieving the proper performance requires a delicate trade-off between carrier collection and light absorption [58, 59]. Solcore **includes a simplified QW structure in the PDD solver in order to calculate the performance of solar cells containing them**. Contrary to other programs like Nextnano, Solcore **does not solve the Schrödinger equation and the PDD equations self-consistently**: first, the energy levels of the quantum wells are solved using a flat-band condition, considering also the strain in the materials, and then an effective band structure is used to solve the transport equations in a bulk-like fashion. This is illustrated in Fig. 12b.

From the perspective of the PDD solver, the actual bandgap and electron affinity of each layer in a quantum

well depend on the energy levels, i.e. the minimum energy for electrons is not the band edge of the conduction band, but the ground confined level. The same applies to holes, with the actual band edge being the maximum between the ground states of light holes and heavy holes. The resulting band profiles used in the PDD solver are shown in the right picture of Fig. 12b.

To use QWs in the PDD solver, we create an effective electron affinity and bandgaps for all layers in the QW. For the barriers, the electron affinity and band gap are the same as they are in bulk, modified by the strain, if necessary. For interlayers, if present, it depends on what is higher, the band edges of the interlayer or the confined carrier levels.

The density of states and the absorption profile need to be modified in a similar way. For the density of states:

- *Barriers* have the bulk density of states and absorption profile.
- *Interlayers* only have the bulk density of states above the barrier and the bulk absorption from the barrier energy and zero below that.
- *Wells* have all the density of states associated with the confined states and the bulk density of states above the barrier, while they have the absorption of the confined levels below the barrier energy and of the bulk above it.

These simplifications are similar to those in Nelson et al. [54] and in Cabrera et al. [60] and allow us to keep the bulk-like form of the carrier densities in the drift diffusion equations under the Boltzmann approximation. A more rigorous treatment will be necessary in the presence of tunnel transport across a superlattice, tunnel escape from the QWs to the barriers—possible in the presence of high electric fields—and in the case of very deep QWs, when carrier escape from the less confined levels might be possible but not from the deeper ones. In these situations, a set of rate equations linking the different levels, as well as a self-consistent solution of the transport and Schrödinger equations would be required, besides using more advanced methods such as a non-equilibrium Green's functions (NEGF) formalism [61].

### 5.3.3 Mesh creation and dynamic meshing

The PDD solver discretizes the device into a finite number of mesh points at which to calculate the band structure, carrier densities, the generation and recombination. The mesh can be static and homogeneous, but the default configuration and the one that results in the least number of mesh points, most accurate result and best convergence uses an inhomogeneous mesh with dynamic re-meshing.

There are two types of nodes in the mesh: masternodes and normal nodes. There are two masternodes at the ends of the device and two more at each side of an interface separated by

0.1 nm. These nodes are static and not affected by re-meshing. The rest of the nodes are automatically distributed depending on the distance to the closest masternode, as illustrated in Fig. 12c. During re-meshing, nodes can be added or removed according to the following rules:

An element will be divided into smaller elements by adding new nodes if any of the following statements is true:

- The variation of the potentials or the carrier densities across the element is large.
- The element is too close to the masternodes limiting the layer.
- The element is too big for the region.

A node will be removed if it fulfils all the following conditions:

- It is not a masternode.
- The variation of the potentials or the carrier densities with respect the previous and next nodes is small.
- It is not too close to the masternodes limiting the layer.
- Removing it does not create an element too big for the region.

After the initial meshing and every time there is a re-meshing, the position of the nodes (except that of the masternodes) is smoothed to avoid having adjacent elements too different in size. This re-meshing process is controlled by a *growth* parameter, which can be adjusted by the user.

Using the inhomogeneous mesh in addition to the dynamic re-meshing ensures that those regions where material properties change abruptly are modelled with more detail, aiding the convergence. It also allows the modelling of devices which have layers with very different thickness, such as QWs a few nanometres thick and bulk absorbers of several microns, without increasing the number of nodes significantly.

### 5.4 Depletion approximation

The depletion approximation provides an analytical—or semi-analytical—solution to the Poisson–drift–diffusion equations described in the previous section applied to simple PN homojunction solar cells. Historically, it has been used extensively to model solar cells and it is still valid, to a large extent, for traditional PN junctions. More importantly, it requires less input parameters than the PDD solver and these can be easily related to macroscopic measurable quantities, like mobility or diffusion lengths. The DA model is based on the assumption that around the junction between the P and N regions, there are no free carriers and therefore all the electric field is due to the fixed, ionized dopants. This “depletion” of free carriers reaches a certain depth towards

the N and P sides; beyond this region, free and fixed carriers of opposite charges balance and the regions are neutral. Under these conditions, Poisson's equation decouples from the drift and diffusion equations and it can be solved analytically for each region. For example, for a PN junction with the interface between the two regions at  $z = 0$ , the solution to Eq. 45 will be:

$$\phi(z) = \begin{cases} 0 & \text{if } z < -w_p \\ \frac{qN_a}{2\epsilon_s}(z + w_p)^2 & \text{if } -w_p < z < 0 \\ -\frac{qN_d}{2\epsilon_s}(z - w_n)^2 + V_{bi} & \text{if } 0 < z < w_n \\ V_{bi} & \text{if } w_n < z \end{cases} \quad (56)$$

where  $w_n$  and  $w_p$  are the extensions of the depletion region towards the N and P sides, respectively, and can be found by the requirement that the electric field  $F$  and the potential  $\phi$  need to be continuous at  $z = 0$ .  $V_{bi}$  is the built-in voltage, which can be expressed in terms of the doping concentration on each side,  $N_d$  and  $N_a$ , and the intrinsic carrier concentration in the material,  $n_i^2$ :

$$V_{bi} = \frac{k_b T}{q} \ln \left( \frac{N_d N_a}{n_i^2} \right). \quad (57)$$

Another consequence of the depletion approximation is that the quasi-Fermi level energies are constant throughout the corresponding neutral regions and also constant in the depletion region, where their separation is equal to the external bias  $qV$ . Based on these assumptions, Eqs. 46 and 47 simplify and an analytical expression can be found for the dependence of the recombination and generation currents on the applied voltage. A full derivation of these expressions is included in Nelson [54].

Solcore's implementation of the depletion approximation includes two modifications to the basic equations. The first one is allowing for an intrinsic region to be included between the P and N regions to form a PIN junction. For low injection conditions (low illumination or low bias) this situation can be treated as described before, simply considering that the depletion region is now widened by the thickness of the intrinsic region. Correctly, no low doping level is allowed for this region.

The second modification is related to the generation profile, which in the equations provided by Nelson is given by the BL law (Eq. 20) which has an explicit dependence on  $z$  and results in analytic expressions for the current densities. In Solcore, we integrate the expressions for the drift-diffusion equations under the depletion approximation numerically to allow for an arbitrary generation profile calculated with any of the methods described in Sect. 4. It should be noted that although the equations are integrated numerically this will not be a self-consistent solution of the Poisson-

drift-diffusion equations, as is achieved by the PDD solver (Sect. 5.3).

## 6 Multi-junction solar cells

A complete photovoltaic solar cell can include one or more junctions, metal contacts, optical layers (including anti-reflective coatings and nanophotonic structures) and tunnel junctions. The junctions, in turn, might range from simple PN homojunctions to complex heterojunctions, including multi-quantum well structures. The solvers described in Sect. 5 only calculate the properties of single junction devices. To combine them into a multi-junction device, it is necessary to consider that the individual junctions are electrically connected in series and the potential coupling of light emitted by the wider bandgap junctions into those with smaller bandgap. The Supplementary Information includes a full step-by-step example of the modelling of a dual junction solar cell with QWs, anti-reflecting coating and a tunnel junction, calculating the external quantum efficiency, the IV characteristics under illumination and the performance of the solar cell as a function of light concentration.

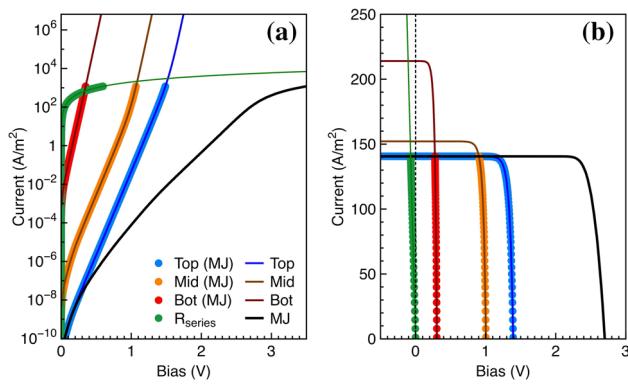
### 6.1 No radiative coupling

We first consider the case of no radiative coupling between junctions. This is a good approximation for materials which do not radiate efficiently or radiative materials working at low concentration, when the fraction of radiative recombination compared to non-radiative recombination is low. In this case, the IV curve of each junction can be calculated independently of each other and the current flowing through the MJ structure is limited by the junction with the lowest current at any given voltage. Series resistances defined for each junction are now added together and included as a single term. The operating voltage of each of the junctions is finally back-calculated and added together to get the voltage of the MJ device.

The pseudocode for this solver is:

1. Calculate the  $I_j(V)$  of each junction  $j$  in the structure.
2. Find the current flowing through the MJ device as  $I_{MJ}(V) = I_j(V)$ , if  $|I_j(V)| = \min(|I_1(V)| \dots |I_N(V)|)$ .
3. Calculate the voltage of each junction by interpolating its IV curve at the new current values,  $V_j(I_{MJ})$ , and the voltage dropped due to the series resistances,  $V_{RS} = R_s I_{MJ}$ .
4. Calculate the total voltage at a given current as  $V_{MJ} = V_{RS} + \sum_j V_j$ .
5. Interpolate the  $I_{MJ}(V_{MJ})$  and the  $I_{MJ}(V_j)$  to the desired output voltage values.

Figure 13 shows the simulated IV curve of a 3J solar cell made of Ge/InGaAs/GaAsP. The electrical properties of the



**Fig. 13** **a** Dark IV curve of a MJ solar cell, including the IV of the individual junctions separately (continuous lines) and the junctions as part of the MJ structure, **b** light IV curve of the same MJ solar cell

three junctions were calculated using the depletion approximation solver. In the dark (Fig. 13a) the voltages of each of the junctions at a given current add together, resulting in the total voltage of the MJ structure. The  $R_s$  contribution to the voltage goes in the same direction as those of the junctions. Under illumination (Fig. 13b) the junction producing the lower current (the top junction in this case) limits the overall current of the MJ device. At zero bias, or even at some negative bias, the non-limiting junctions are positively biased, recombining all the photocurrent that cannot be extracted because of the limiting top cell. The contribution of the  $R_s$  to the voltage of the MJ device is negative, resulting in a reduction of the fill factor and the overall efficiency of the solar cell.

## 6.2 With radiative coupling

Radiative coupling takes place when the light produced by a high bandgap junction due to radiative recombination is absorbed by a lower bandgap junction, contributing to its photocurrent and changing the operating point. It has been identified in numerous highly radiative materials such as quantum well solar cells and III–V MJ solar cells [62–64]. It appears as an artefact during the QE measurements of MJ solar cells [65], but it is also an effect that can be exploited to increase the performance of MJ devices [57] and their tolerance to spectral changes, resulting in superior annual energy yield [66].

The radiative coupling formalism included in Solcore is based on the works by Chan et al. [66] and Nelson et al. [20]. It is implemented only for the DB junction model and for the 2D model when it is defined in terms of a radiative efficiency and the parameters calculated from the DB model. The current of a junction  $j$  including radiative coupling from the junction immediately above it  $j - 1$  is given by:

$$J_j^{\text{total}} = J_j^{\text{nc}} + J_{j-1 \rightarrow j}^{\text{coupled}}. \quad (58)$$

This current depends on two factors: the amount of radiation effectively emitted downwards, towards the lower junction, and the fraction of it that is absorbed and converted into electricity. If we ignore the possible reflection of light at the interface between both junctions, this current can be written by using a modified version of Eq. 30 that considers only the radiation emitted towards the back:

$$J_{j-1 \rightarrow j}^{\text{coupled}}(V, T) = q \frac{2n^2}{h^3 c^2} \int_0^\infty \frac{E^2}{e^{\frac{E-qV}{k_b T}} - 1} A_{j-1 \rightarrow j}(E) dE \quad (59)$$

with  $A_{j-1 \rightarrow j}(E)$  given by:

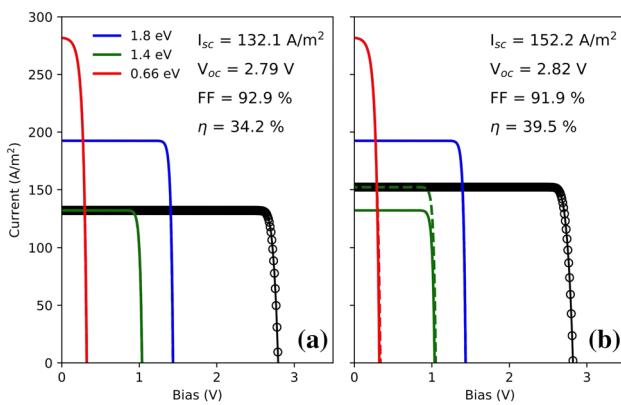
$$A_{j-1 \rightarrow j}(E) = 2\pi \int_0^1 \left[ 1 + r(E, \theta) e^{-\alpha_{j-1} w_{j-1} / \cos \theta} \right] \times \left( 1 - e^{-\alpha_{j-1} w_{j-1} / \cos \theta} \right) \times \left( 1 - e^{-\alpha_j w_j / \cos \theta} \right) \cos \theta d(\cos \theta). \quad (60)$$

As discussed previously, any information related to total internal reflection will be contained in the  $r(E, \theta)$  term, and therefore the integral over  $\cos \theta$  can be done from 0 to 1. In the case of thin junctions, some radiation could reach the next junction  $j + 1$ . The coupled current in that case can be easily calculated by modifying Eq. 60 to account for the fraction of light absorbed by the junctions between the emitting junction and the junction of interest. In the general case, the current coupled into junction  $j$  will be given by:

$$J_j^{\text{coupled}} = \sum_{i=1}^{j-1} J_{i \rightarrow j}^{\text{coupled}}. \quad (61)$$

Radiative coupling might change the junction that is current limiting the MJ device, so the process to obtain the IV curves in this case proceeds in two steps. First, the IV of the junctions and the total IV are calculated without coupling. The resulting IV curves are then used as the initial conditions for the numerical solver that will calculate the correct voltage of each junction including the radiative coupling.

Figure 14 shows the IV curve under the AM1.5G solar spectrum of a three junction solar cell (a) without and (b) with radiative coupling. Without coupling, the middle junction severely limits the current of the MJ solar cell. When coupling is enabled, the middle junction is still the limiting one but part of the excess current of the top junction is transferred to it, increasing its photocurrent by around  $20 \text{ A/m}^2$ . Part of the radiative recombination is also transferred to the bottom cell, increasing slightly its photocurrent. In this case, given that the junction was overproducing current already, such coupling is only visible as an increase in the voltage. Altogether, the radiative coupling results in an enhancement



**Fig. 14** Light IV curve of a 3J solar cell **a** without and **b** with radiative coupling. Continuous lines represent the individual IV curves of the junctions isolated and the dash lines when they are inside the MJ device, illustrating the effect of the coupling

of the  $V_{oc}$  of 30 mV and of the efficiency  $\eta$  of 5.3%. This example uses junctions with 100% radiative efficiency to illustrate the effect, but this phenomenon is always present to some extent, becoming especially important under concentration [57,66].

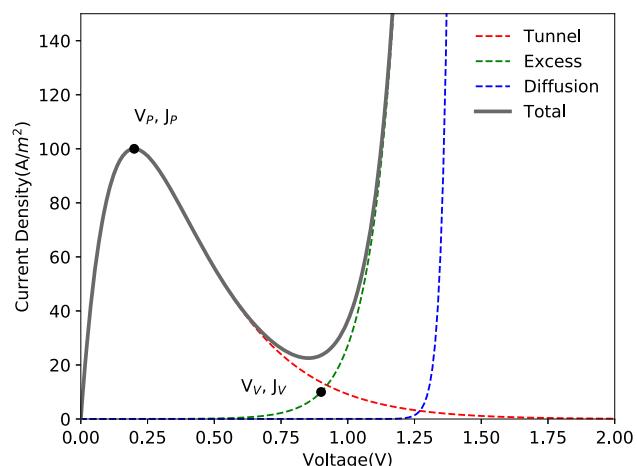
### 6.3 Restrictions in the junction definitions

Having multiple methods for modelling the junctions gives a lot of freedom and flexibility but it also imposes some restrictions in how and when they can be combined in order to create a MJ solar cell. The following compatibility rules apply:

- When there is no radiative coupling and we are interested only in the dark IV characteristics, all junction models can be combined with each other. This allows a MJ device where the top junction is defined using the DB model, the middle junction is defined with the 2D model and the bottom junction uses a more accurate PDD model, for example.
- The same applies for light IV and quantum efficiency simulations as long as the optical model chosen is the BL law. In this case, any junction defined using the 2D model needs to include an absorptivity value.
- The TMM and RCWA optical models are supported only by the PDD and DA junction models.
- In the presence of radiative coupling, the only junction models that can be used are DB and 2D, as long as the latter includes an absorptivity value.

### 6.4 Tunnel junctions

Solcore includes partial support for tunnel junctions. They represent an optical loss due to parasitic absorption in the



**Fig. 15** IV curve of a tunnel junction defined according to the parametric model

layers, but also an electrical loss. At the moment, there are two models for tunnel junctions. The first one is a simple resistive model, where the tunnel junction is simply modelled as a series resistance. This approximation should be valid in most cases, but will break down if the current is close to or higher than the peak current density of the junction.

The second model is a parametric model, based on the simple formalism described by Sze [53]. In this model, the total current of the tunnel junction will have three components: the tunnel current  $J_T$  accounting for band-to-band transport, the excess current  $J_{ex}$  related to transport across states inside the forbidden gap, and the diffusion current  $J_D$ , which is the usual minority-carrier injection current in PN junctions. The following equations summarize all these components.

$$J_{TJ} = J_T + J_{ex} + J_D \quad (62)$$

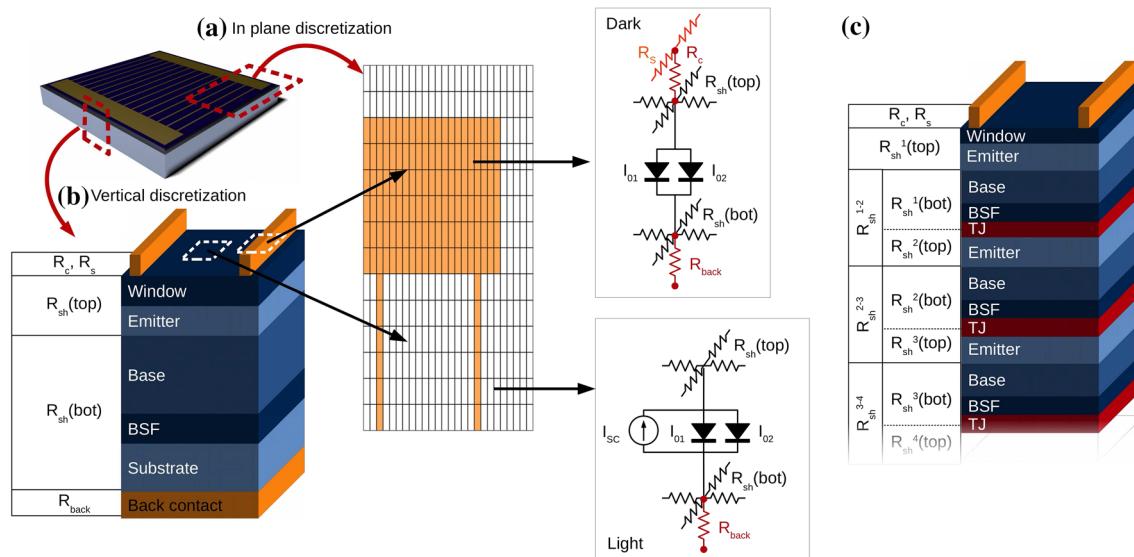
$$J_T = \frac{J_P V}{V_p} \exp\left(1 - \frac{V}{V_p}\right) \quad (63)$$

$$J_{ex} = J_V \exp[C(V - V_V)] \quad (64)$$

$$J_D = J_0 \left[ \exp\left(\frac{qV}{k_b T}\right) - 1 \right]. \quad (65)$$

As illustrated in Fig. 15,  $J_p$  and  $V_p$  are the peak current and voltage,  $J_V$  and  $V_V$  are the valley current and voltages,  $C$  is a prefactor of the exponent and  $J_0$  the reverse saturation current. In this simple implementation, these six parameters need to be provided as inputs, and can be used as fitting parameters to reproduce experimental data. This allows to correctly account for the breakdown of the tunnel junction in situations when the current is above the peak current.

Solcore can also accept external IV data for the tunnel junctions and the implementation of the more rigorous, but still analytic model, described by Louarn et al. [67] is currently under way in order to relate the tunnel currents with



**Fig. 16** Schema of the quasi-3D solar cell modelling included in Solcore. The solar cell is discretized **a** in the plane and **b** in the vertical direction. Illuminated and dark regions are then modelled using electrical components that, when combined, form a 3D electrical mesh giving the voltages and currents at any point of the structure. **c** An example of the vertical discretization of a N-junction solar cell

the actual materials and layer structure used in the tunnel junction definition.

## 7 Large circuit solver

When the two-diode model is used to define the junctions in a MJ solar cell, then larger scale circuits can be constructed. Solcore includes two levels of large-scale equivalent circuits: quasi-3D solar cell modelling and solar array modelling. Both solvers are based on the interface between Solcore and SPICE, allowing for a fast calculation of complex structures with many elements.

### 7.1 Quasi-3D solar cell model

The quasi-3D solar cell model included in Solcore uses a SPICE-based electrical network to model the flow of injected current through the solar cell, as depicted in Fig. 16. The plane of the cell is discretized into many elements, each of them representing a small portion of the cell. Depending on the location of the element—exposed to the sunlight or underneath a metal finger—the IV curve of the cell will be the light IV or the dark IV. Each element is linked to their neighbours with resistors, representing the lateral current flow and dependent on the sheet resistance of the cells. This method can be applied to any number of junctions.

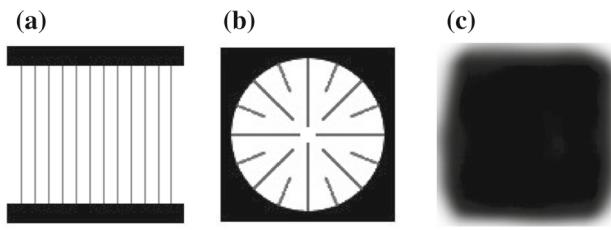
This type of formalism is widely used to simulate the performance of solar cells when the effect of a spatial variable needs to be incorporated in the model. This variable can be the

design of the front metal grid, in order to minimize the effect of series resistances [68]; the inhomogeneous illumination profile in concentrator devices; the impact of such inhomogeneity on the transport through the tunnel junctions [69,70]; or the distribution of defects or inhomogeneities [71,72]. Recently, this formalism was used to model the photoluminescence and the electroluminescence-based IV curves of MJ devices, accounting for the limited lateral carrier transport [73].

Specifically for the modelling and optimization of the front grid of solar cells in order to minimize shading losses and series resistance, there are two packages already available: PVMOS, developed by Pieters in C and released as open source [74,75], and Griddler, developed by Wong using Matlab and available at PV Lighthouse [11,76].

#### 7.1.1 In-plane discretization

As shown in Fig. 16, there are two regions in the plane: the metal and the aperture. These two are provided to Solcore as grey scale images that will work as masks. The resolution of the images, in pixels, will define the in-plane discretization. By default, the aspect ratio of the pixels in the image will be 1:1, but this can be set to a different value in order to reduce the number of elements and improve speed. For example, the in-plane discretization of Fig. 16a has an aspect ratio  $A_r = L_y/L_x = 4$ , with  $L_x$  and  $L_y$  the pixel size in each direction. The values of the pixels in the metal mask are 0 where there is no metal (the aperture area), 255 where there is metal and the external electrical contacts (the boundaries



**Fig. 17** **a, b** Two examples of metal masks and **c** an illumination mask. The thin metal fingers in **(a)** and **(b)** are grey, indicating that there is metal in those pixels but that their bias is not set to be equal to the external bias

with fixed, externally set voltage values) and any other value in between to represent regions with metal but not fixed voltage. The pixels of the illumination mask—which become the aperture mask after removing the areas shadowed by the metal—can have any value between 0 and 255. These values divided by 255 will indicate the intensity of the sunlight at that pixel relative to the maximum intensity. Figure 17 illustrates two examples of metal masks (a and b) and an illumination mask (c) with  $120 \times 120$  pixels. As it can be seen, while rectangular metal fingers are well reproduced, diagonal fingers are less accurate and could require a finer discretization. The illumination mask is mostly homogeneous except around the edges and in the corners, where intensity is much lower. This pattern could be produced, for example, by the secondary optics of a concentration system.

The minimum total number of nodes where SPICE will need to calculate the voltages will be  $N \times M \times 2 \times Q$ , with  $N$  and  $M$  the number of pixels in both in-plane directions and  $Q$  the number of junctions, which require 2 nodes each. To this, the front and back metal contacts could add a maximum of  $2M \times M$  nodes. Exploiting symmetries of the problem as well as choosing an appropriate pixel aspect ratio will significantly reduce the number of nodes and therefore the time required for the computation of the problem.

### 7.1.2 Vertical discretization

First, the solar cell is solved as described in Sects. 5 and 6 in order to obtain the parameters for the 2-diode model at the given illumination conditions. These parameters are then used to replicate the 2-diode model in SPICE. The  $I_{SC}$  is scaled in each pixel by the intensity of the illumination given by the illumination mask. Sheet resistances above and below each junction,  $R_{sh}(\text{top})$  and  $R_{sh}(\text{bot})$ , account for the lateral transport. Beneath the metal, there is no current source, as the region is in the dark, and there are extra resistances accounting for the contact between the metal and the semiconductor  $R_c$  and the transport along the metal finger  $R_s$  [68]. Given that the pixels can be asymmetric, these resistances need to be defined in both in-plane directions,  $x$  and  $y$ :

$$R_{sh}^x = \frac{1}{A_r} R_{sh} \quad (66)$$

$$R_{sh}^y = A_r R_{sh} \quad (67)$$

$$R_s^x = \frac{1}{h A_r} \rho_m \quad (68)$$

$$R_s^y = \frac{A_r}{h} \rho_m \quad (69)$$

$$R_c = R_{\text{back}} = \frac{1}{L_x^2 A_r} \rho_c \quad (70)$$

where  $h$  is the height of the metal,  $\rho_m$  their linear resistivity, and  $\rho_c$  the contact resistivity between metal and semiconductor. The sheet resistance of a stack of semiconductor layers  $R_{sh}$  is equal to the combination in parallel of the individual sheet resistances. Using the single junction example of Fig. 16,  $R_{sh}(\text{top})$  will be given by:

$$\frac{1}{R_{sh}(\text{top})} = \frac{1}{R_{sh}(\text{window})} + \frac{1}{R_{sh}(\text{emitter})}. \quad (71)$$

Each of these can be estimated from the thickness of the layer  $d$ , the majority carrier mobility  $\mu$  and the doping  $N$  as [70]:

$$\frac{1}{R_{sh}} = q d \mu N. \quad (72)$$

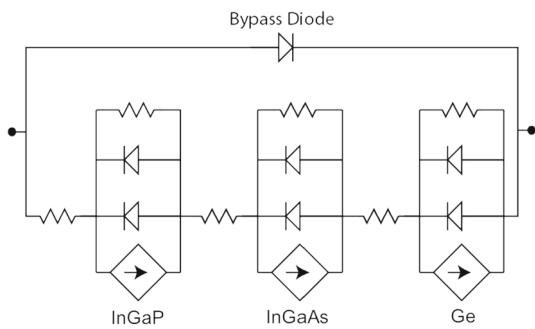
If the solar cell has been defined using only the DA and PDD junction models, this information is already available for all the layers of the structure. For junctions using the DB and two diode models,  $R_{sh}$  will need to be provided for the top and bottom regions of each junction. Intrinsic layers will be ignored as they do not contribute to the lateral current transport.

### 7.2 Solar array model

The ability to use Solcore to build a SPICE equivalent circuit allows entire PV systems to be simulated from the bottom up [77]. Each photovoltaic solar cell is described using an equivalent circuit which can then be arranged in strings of series and parallel cells to represent the entire system. An example for a triple junction solar cell, complete with a bypass diode is shown in Fig. 18; this unit is the basic building block for a concentrator PV module [52].

The diode and resistance values for the equivalent circuit are determined from solar cell testing, while the current source is evaluated by integrating the product of the spectral irradiance (estimated using an appropriate radiative transfer code, e.g. SPCTRL2 or SMARTS) and the quantum efficiency which in turn can be calculated dynamically as a function of temperature by Solcore [78].

Since the entire module (and subsequently the system) is assembled from individual solar cell components, it is



**Fig. 18** Equivalent circuit for a triple junction solar cell

possible (and indeed, necessary) to distribute the component values to accommodate for manufacturing tolerances. This enables a close match between the modelled output power and that measured experimentally and has been used to determine how both aerosols and precipitable water affect the electricity yield from concentrator PV systems [79,80]. Where system IV data are available, the emergence of electrical faults, (e.g. shunts or shading) can also be accounted for [81].

## 8 Closing remarks

In this article, we have described the capabilities of Solcore, a multi-scale, Python-based, modular simulation framework for semiconductor materials and solar cells. Its main strengths are:

- *Flexibility* Provides a variety of tools, rather than a single solution, for the study of traditional and novel semiconductor materials and devices.
- *Modularity* Can be expanded with new capabilities, innovative solvers and tools.
- *Accessibility* Not only is it open source, but it is also designed to be easy to learn and to use, serving as a teaching tool as much as a research tool.
- *Rigour* The physics behind every functionality are well understood and supported by numerous references, as are the approximations made in order to simplify the implementation of the problem or the interpretation of the results.
- *Integrated* All of Solcore’s features are designed to be compatible with one another to allow for truly multi-scale modelling in an integrated way.

**Acknowledgements** Solcore is not the creation of a single person but the combined effort of present and past members of the Quantum Photovoltaics Group at Imperial College London. Therefore, the authors would like to thank T. Thomas, N. L. A. Chan, J. Nelson and J. Connolly, who also contributed significantly to its development. The authors will also like to acknowledge the financial support of the Engineering

and Physical Science Research council (EPSRC) through the research Grant EP/M025012/1.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Gueymard, C.: “SMARTS2” [Online] (1995). <https://www.nrel.gov/redc/smarts/>
2. Eisenlohr, J., Tucher, N., Höhn, O., Hauser, H., Peters, M., Kiefel, P., Goldschmidt, J.C., Bläsi, B.: Matrix formalism for light propagation and absorption in thick textured optical sheets. Opt. Express **23**(11), A502–A518 (2015)
3. Baker-Finch, S.C., McIntosh, K.R.: A freeware program for precise optical analysis of the front surface of a solar cell. In: 2014 IEEE 40th Photovoltaic Specialist Conference (PVSC), pp. 002 184–002 187. IEEE (2010)
4. Basore, P.A.: PC-1D version 3: improved speed and convergence. In IEEE 20nd Photovoltaic Specialists Conference, pp. 299–302 (1991)
5. Burgelman, M., Nollet, P., Degrave, S.: Modelling polycrystalline semiconductor solar cells. Thin Solid Films **361–362**, 527–532 (2000)
6. Fell, A.: A free and fast three-dimensional/two-dimensional solar cell simulator featuring conductive boundary and quasi-neutrality approximations. IEEE Trans. Electron Devices **60**(2), 733–738 (2013)
7. Varache, R., Leendertz, C., Gueunier-Farret, M., Haschke, J., Muñoz, D., Korte, L.: Investigation of selective junctions using a newly developed tunnel current model for solar cell applications. Sol. Energy Mater. Sol. Cells **141**(Supplement C), 14–23 (2015)
8. Birner, S., Zibold, T., Andlauer, T., Kubis, T., Sabathil, M., Trelakis, A., Vogl, P.: nextnano: general purpose 3-D simulations. IEEE Trans. Electron Devices **54**(9), 2137–2142 (2007)
9. Silvaco, “Atlas: Device Simulation Framework” (2017)
10. Synopsis, “Sentaurus Device” (2017)
11. P. L. team and contributors. PV Lightouse. [Online] (2018). <http://www.pvlighthouse.com.au>
12. Andrews, R.W., Stein, J.S., Hansen, C.: Introduction to the open source PV LIB for python Photovoltaic system modelling package. In: 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC), pp. 0170–0174 (2014)
13. Connolly, J.P.: Analytical models of bulk and quantum well solar cells and relevance of the radiative limit. In: Fara, L., Yamaguchi, M. (eds.) Advanced Solar Cell Materials, Technology, Modelling, and Simulation, chap. 5, pp. 59–77. IGI Global, Hershey, PA (2013). <https://doi.org/10.4018/978-1-4666-1927-2>
14. Alonso-Álvarez, D., Wilson, T., Pearce, P., Führer, M., Farrell, D., Ekins-Daukes, N.J.: Solcore Code Repository. [Online] (2018). <http://doi.org/10.5281/zenodo.1185316>
15. Vurgaftman, I., Meyer, J.R., Ram-Mohan, L.R.: Band parameters for III–V compound semiconductors and their alloys. J. Appl. Phys. **89**(11), 5815–5875 (2001)
16. Levinstein, M., Rumyantsev, S., Shur, M., Levinstein, M., Rumyantsev, S., Shur, M.: Handbook Series on Semiconductor Parameters, ser. Volume 2: Ternary and Quaternary III-V Compounds, vol. 2. World Scientific, Singapore (2012)
17. Sotoodeh, M., Khalid, A.H., Rezazadeh, A.A.: Empirical low-field mobility model for III–V compounds applicable in device simulation codes. J. Appl. Phys. **87**(6), 2890–12 (2000)

18. Software Spectra Inc., “Optical Data from Sopra S. A.” (2008)
19. Paxman, M., Nelson, J., Braun, B., Connolly, J., Barnham, K.W.J., Foxon, C.T., Roberts, J.S.: Modeling the spectral response of the quantum well solar cell. *J. Appl. Phys.* **74**(1), 614–621 (1993)
20. Nelson, J., Barnes, J., Ekins-Daukes, N., Kluftinger, B., Tsui, E., Barnham, K., Foxon, C.T., Cheng, T., Roberts, J.S.: Observation of suppressed radiative recombination in single quantum well p-i-n photodiodes. *J. Appl. Phys.* **82**(12), 6240–6246 (1997)
21. Nelson, J., Ballard, I., Barnham, K., Connolly, J.P., Roberts, J.S., Pate, M.: Effect of quantum well location on single quantum well p-i-n photodiode dark currents. *J. Appl. Phys.* **86**(10), 5898–5905 (1999)
22. Fuhrer, M.F., Adams, J.G.J., Barnham, K.W.J., Browne, B.C., Chan, N.L.A., Farrell, D.J., Hirst, L., Lee, K.-H., Ekins-Daukes, N.J., Ogura, A., Yoshida, K., Okada, Y.: Extensible modelling framework for nanostructured III-V solar cells. In: 2011 37th IEEE Photovoltaic Specialists Conference (PVSC), pp. 002615–002618 (2011)
23. Führer, M., Farrell, D., Ekins-Daukes, N.: CPV modelling with Solcore: an extensible modelling framework for the rapid computational simulation and evaluation of solar cell designs and concepts. In: 9th International Conference on Concentrator Photovoltaic Systems: CPV-9, pp. 34–37. AIP (2014)
24. Tomic, S.: Electronic structure of  $\text{InyGa}_{1-y}\text{As}_1\text{-xN}_x/\text{GaAs}(N)$  quantum dots by ten-band kp theory. *Phys. Rev. B* **73**(12), 125348–8 (2006)
25. Volk, M., Lutgen, S., Marschner, T., Stolz, W., Göbel, E.O., Christianen, P.C.M., Maan, J.C.: Carrier effective masses in symmetrically strained (GaIn)As/Ga(PAs) multiple-quantum-well structures. *Phys. Rev. B* **52**(15), 11096–11104 (1995)
26. Frensel, W.R.: Numerical evaluation of resonant states. *Superlattices Microstruct.* **11**(3), 347–350 (1992)
27. Adachi, S.: Model dielectric constants of GaP, GaAs, GaSb, InP, InAs, and InSb. *Phys. Rev. B* **35**(14), 7454–7463 (1987)
28. Adachi, S.: Optical dispersion relations for GaP, GaAs, GaSb, InP, InAs, InSb,  $\text{Al}_x\text{Ga}_{1-x}\text{As}$ , and  $\text{In}_{1-x}\text{Ga}_x\text{As}_y\text{P}_{1-y}$ . *J. Appl. Phys.* **66**(12), 6030–6040 (1989)
29. Adachi, S.: Optical dispersion relations for Si and Ge. *J. Appl. Phys.* **66**(7), 3224–3231 (1989)
30. Rakić, A.D., Majewski, M.L.: Modeling the optical dielectric function of GaAs and AlAs: extension of Adachi's model. *J. Appl. Phys.* **80**(10), 5909–5914 (1996)
31. Kim, C.C., Garland, J.W., Abad, H., Raccah, P.M.: Modeling the optical dielectric function of semiconductors: extension of the critical-point parabolic-band approximation. *Phys. Rev. B* **45**(20), 11749–11767 (1992)
32. Kim, C.C., Garland, J.W., Raccah, P.M.: Modeling the optical dielectric function of the alloy system  $\text{Al}_x\text{Ga}_{1-x}\text{As}$ . *Phys. Rev. B* **47**(4), 1876–1888 (1993)
33. Woollam, J.A.: Guide to using WVASE 32: spectroscopic ellipsometry data acquisition and analysis software. J. A. Woollam Company (2008). <https://books.google.co.uk/books?id=xOupYgEACAAJ>
34. Palik, E.D.: Gallium Arsenide (GaAs). In: *Handbook of Optical Constants of Solids*, pp. 429–443. Elsevier, Amsterdam (1997)
35. Wilson, T., Mellor, A., Hylton, N.P., Ekins-Daukes, N.J.: The effects of short-range alloy disorder on the potential voltage performance in GaAsBi based solar cells. In: Proceedings of the 33rd European Photovoltaic Solar Energy Conference and Exhibition (2017)
36. Chuang, S.L.: *Physics of Optoelectronic Devices*. Wiley-Interscience, New York (1995)
37. Barnham, K., Vvedensky, D. (eds.): *Low-Dimensional Semiconductor Structures: Fundamentals and Device Applications*. Cambridge University Press, Cambridge (2001)
38. Klipstein, P.C., Apsley, N.: A theory for the electroreflectance spectra of quantum well structures. *J. Phys. C Solid State Phys.* **19**(32), 6461–6478 (2000)
39. Galleano, R., Zaaiman, W., Alonso-Álvarez, D., Minuto, A., Ferretti, N., Fucci, R., Pravettoni, M., Halwachs, M., Friederichs, M., Plag, F., Friedrich, D., Haverkamp, E.: Results of the fifth international spectroradiometer comparison for improved solar spectral irradiance measurements and related impact on reference solar cell calibration. *IEEE J. Photovolt.* **6**(6), 1587–1597 (2016)
40. Alonso-Álvarez, D., Wilso, D., Pearce, P., Führer, M., Farrell, D., Ekins-Daukes, N.J.: Supplementary Information—Solcore: a multi-scale, Python-based library for modelling solar cells and semiconductor materials. *J. Comput. Electron.*, vol. Supplementary Information (2018)
41. Wilson, T., Thomas, T., Führer, M., Ekins-Daukes, N.J.: Addressing reflectivity losses in multi-junction solar cells to achieve 50 conversion efficiency. *Mater. Res. Innov.* **19**(7), 503–507 (2016)
42. Alonso-Álvarez, D., Ferre Llin, L., Mellor, A., Paul, D.J., Ekins-Daukes, N.J.: ITO and AZO films for low emissivity coatings in hybrid photovoltaic-thermal applications. *Sol. Energy* **155**, 82–92 (2017)
43. Macleod, H.A.: *Thin-Film Optical Filters*. Institute of Physics Publishing, Bristol (2001)
44. Byrnes, S.J.: Multilayer Optical Calculations. <https://arxiv.org/abs/1603.02720v2> (2016)
45. Liu, V., Fan, S.: S4: a free electromagnetic solver for layered periodic structures. *Comput. Phys. Commun.* **183**(10), 2233–2244 (2012)
46. Hylton, N.P., Li, X.F., Giannini, V., Lee, K.H., Ekins-Daukes, N.J., Loo, J., Vercruyse, D., Van Dorpe, P., Sodabanlu, H., Sugiyama, M., Maier, S.A.: Loss mitigation in plasmonic solar cells: aluminum nanoparticles for broadband photocurrent enhancements in GaAs photodiodes. *Sci. Rep.* **3**(1), 2874–6 (2013)
47. Mellor, A., Hylton, N.P., Maier, S.A., Ekins-Daukes, N.: Interstitial light-trapping design for multi-junction solar cells. *Sol. Energy Mater. Solar Cells* **159**, 212–218 (2017). <https://doi.org/10.1016/j.solmat.2016.09.005>
48. Martí, A., Araújo, G.L.: Limiting efficiencies for photovoltaic energy conversion in multigap systems. *Sol. Energy Mater. Sol. Cells* **43**(2), 203–222 (1996)
49. Chan, N.L.A., Ekins-Daukes, N.J., Adams, J.G.J., Lumb, M.P., Gonzalez, M., Jenkins, P.P., Vurgaftman, I., Meyer, J.R., Walters, R.J.: Optimal bandgap combinations—does material quality matter? *IEEE J. Photovolt.* **2**(2), 202–208 (2012)
50. Green, M.A.: Radiative efficiency of state-of-the-art photovoltaic cells. *Prog. Photovolt. Res. Appl.* **20**(4), 472–476 (2011)
51. Thomas, T., Mellor, A., Hylton, N.P., Führer, M., Alonso-Álvarez, D., Braun, A., Ekins-Daukes, N.J., David, J.P.R., Sweeney, S.J.: Requirements for a GaAsBi 1 eV sub-cell in a GaAs-based multi-junction solar cell. *Semicond. Sci. Technol.* **30**(9), 094010-6 (2015)
52. Ekins-Daukes, N.J., Kemmoku, Y., Araki, K., Betts, T.R., Gottschalg, R., Infield, D.G., Yamaguchi, M.: The design specification for Syracuse: a multi-junction concentrator system computer model. In: Proceedings of the 19th European Photovoltaic Solar Energy Conference, pp. 1–4 (2004)
53. Sze, S.M.: *Physics of Semiconductor Devices*, 2nd edn. Wiley, New York, NY (1981)
54. Nelson, J.: *The Physics of Solar Cells*. Imperial College Press, London; River Edge, NJ: Distributed by World Scientific Pub. Co, (2003)
55. Basore, P.A.: PC-1D Installation Manual and User's Guide Version 3.1, pp. 1–96 (2009)
56. Farrell, P.A., Gartland, E.C.: On the Sharfetter-Gummel discretization for drift-diffusion continuity equations. In: Miller, J.J.H. (ed.) *Computational Methods for Boundary and Interior Layers in Several Dimensions*, pp. 51–79 (1991)

57. Thomas, T., Wilson, T., Führer, M., Alonso-Álvarez, D., Ekins-Daukes, N.J., Lackner, D., Kailuweit, P., Philipps, S.P., Bett, A.W., Toprasertpong, K., Sugiyama, M., Okada, Y.: Potential for reaching 50% power conversion efficiency using quantum heterostructures. In: 6th World Conference on Photovoltaic Energy Conversion, pp. 1–2 (2014)
58. Alonso-Álvarez, D., Führer, M., Thomas, T., Ekins-Daukes, N.: Elements of modelling and design of multi-quantum well solar cells. In: 2014 IEEE 40th Photovoltaic Specialists Conference (PVSC), pp. 2865–2870 (2014)
59. Alonso-Álvarez, D., Ekins-Daukes, N.J.: Quantum wells for high-efficiency photovoltaics. In: Freundlich, A., Lombez, L., Sugiyama, M. (eds.) Physics, Simulation, and Photonic Engineering of Photovoltaic Devices V, vol. 9743, p. 974311. SPIE OPTO, San Francisco, CA (2016). <https://doi.org/10.1117/12.2217590>
60. Cabrera, C.I., Rimada, J.C., Connolly, J.P., Hernandez, L.: Modelling of GaAsP/InGaAs/GaAs strain-balanced multiple-quantum well solar cells. *J. Appl. Phys.* **113**(2), 024512–7 (2013)
61. Aeberhard, U.: Quantum-kinetic perspective on photovoltaic device operation in nanostructure-based solar cells. *J. Mater. Res.* **33**, 373–386 (2018)
62. Lee, K.-H., Barnham, K.W.J., Connolly, J.P., Browne, B.C., Airey, R.J., Roberts, J.S., Führer, M., Tibbits, T.N.D., Ekins-Daukes, N.J.: Demonstration of photon coupling in dual multiple-quantum-well solar cells. *IEEE J. Photovolt.* **2**(1), 68–74 (2012)
63. Steiner, M.A., Geisz, J.F., Moriarty, T.E., France, R.M., McMahon, W.E., Olson, J.M., Kurtz, S.R., Friedman, D.J.: Measuring IV curves and subcell photocurrents in the presence of luminescent coupling. *IEEE J. Photovolt.* **3**(2), 879–887 (2013)
64. Steiner, M.A., Geisz, J.F.: Non-linear luminescent coupling in series-connected multijunction solar cells. *Appl. Phys. Lett.* **100**(25), 251106–5 (2012)
65. Steiner, M.A., Kurtz, S.R., Geisz, J.F., McMahon, W.E., Olson, J.M.: Using phase effects to understand measurements of the quantum efficiency and related luminescent coupling in a multijunction solar cell. *IEEE J. Photovolt.* **2**(4), 424–433 (2012)
66. Chan, N.L.A., Thomas, T., Führer, M., Ekins-Daukes, N.J.: Practical limits of multijunction solar cell performance enhancement from radiative coupling considering realistic spectral conditions. *IEEE J. Photovolt.* **4**(5), 1306–1313 (2014)
67. Louarn, K., Fontaine, C., Arnoult, A., Olivie, F., Lacoste, G., Piquemal, F., Bouchnouh, A., Almuneau, G.: Modelling of interband transitions in GaAs tunnel diode. *Semicond. Sci. Technol.* **31**(6), 06LT01–6 (2016)
68. Steiner, M., Philipps, S.P., Hermle, M., Bett, A.W., Dimroth, F.: Validated front contact grid simulation for GaAs solar cells under concentrated sunlight. *Prog. Photovolt. Res. Appl.* **19**(1), 73–83 (2010)
69. Nishioka, K., Takamoto, T., Agui, T., Kaneiwa, M., Uraoka, Y., Fuyuki, T.: Evaluation of InGaP/InGaAs/Ge triple-junction solar cell under concentrated light by simulation program with integrated circuit emphasis. *Jpn. J. Appl. Phys.* **43**(3), 882–889 (2004)
70. Steiner, M., Guter, W., Peharz, G., Philipps, S.P., Dimroth, F., Bett, A.W.: A validated SPICE network simulation study on improving tunnel diodes by introducing lateral conduction layers. *Prog. Photovolt. Res. Appl.* **20**(3), 274–283 (2011)
71. Jürgens, T., Güntay, L., Bauer, G.H.: Photoluminescence, open circuit voltage, and photocurrents in Cu(In, Ga)Se<sub>2</sub> solar cells with lateral submicron resolution. *Thin Solid Films* **511–512**, 678–683 (2006)
72. Paire, M., Lombez, L., Guillemoles, J.-F., Lincot, D.: Measuring sheet resistance of CIGS solar cell's window layer by spatially resolved electroluminescence imaging. *Thin Solid Films* **519**(21), 7493–7496 (2011)
73. Alonso-Álvarez, D., Ekins-Daukes, N.: SPICE modelling of photoluminescence and electroluminescence based current-voltage curves of solar cells for concentration applications. *J. Green Eng.* **5**(4), 33–48 (2016)
74. Pieters, B.E.: A free and open source finite-difference simulation tool for solar modules. In: 2014 IEEE 40th Photovoltaic Specialist Conference. PVSC 2014, pp. 1370–1375 (2014)
75. Pieters, B.E.: PVMOS Code Repository. [Online] (2018). <https://github.com/IEK-5/PVMOS>
76. Wong, J.: Griddler: intelligent computer aided design of complex solar cell metallization patterns. In: 2013 IEEE 39th Photovoltaic Specialists Conference (PVSC), pp. 0933–0938 (2013)
77. Castaner, L., Silvestre, S.: Modelling PV-Systems Using PSPice. Wiley, New York (2002)
78. Ekins-Daukes, N.J., Betts, T.R., Kemmoku, Y., Araki, K., Lee, H.S., Gottschalg, R., Boreland, M.B., Infield, D.G., Yamaguchi, M.: Syracuse—a multi-junction concentrator system computer model. In: Proceedings of the 31st IEEE Photovoltaic Specialists Conference, pp. 651–654 (2005)
79. Chan, N.L.A., Young, T.B., Brindley, H.E., Ekins-Daukes, N.J., Araki, K., Kemmoku, Y., Yamaguchi, M.: Validation of energy prediction method for a concentrator photovoltaic module in Toyohashi Japan. *Prog. Photovolt. Res. Appl.* **21**(8), 1598–1610 (2012)
80. Chan, N.L.A., Brindley, H.E., Ekins-Daukes, N.J.: Impact of individual atmospheric parameters on CPV system power, energy yield and cost of energy. *Prog. Photovolt. Res. Appl.* **22**(10), 1080–1095 (2014)
81. Kamath, H.G., Araki, K., Ekins-Daukes, N.J., Ramasesha, S.K.: Performance analysis and fault detection method for concentrator photovoltaic modules. Submitted to Journal of Photovoltaics (2018)

# Supplementary Information

Solcore: A multi-scale, Python-based library for modelling solar cells and semiconductor materials

D. Alonso-Álvarez      T. Wilson      P. Pearce      M. Führer      D. Farrell  
N. Ekins-Daukes

Department of Physics, Imperial College London, London, United Kingdom

March 30, 2018

## 1 Comparison of optical solvers

This section contains the complete code needed to reproduce Figure 9, using different optical models to calculate the quantum efficiency of a thin GaAs solar cell.

```
from solcore import si, material
from solcore.structure import Junction, Layer
from solcore.solar_cell import SolarCell
from solcore.solar_cell_solver import solar_cell_solver, default_options
from solcore.light_source import LightSource
import numpy as np
import matplotlib.pyplot as plt

# user options
T = 298
wl = si(np.linspace(400, 900, 80), 'nm')
light_source = LightSource(source_type='standard', version='AM1.5g', x=wl,
output_units='photon_flux_per_m', concentration=1)
opts = default_options
opts.wavelength, opts.no_back_reflexion, opts.size, opts.light_source, opts.
    T_ambient = \
wl, False, [400, 400], light_source, T
# The size of the unit cell for the RCWA structure is 400 x 400 nm

# Defining all the materials we need
Air = material('Air')(T=T)
p_GaAs = material('GaAs')(T=T, Na=si('4e18cm-3')) # for the GaAs cell
emitter
n_GaAs = material('GaAs')(T=T, Nd=si('2e17cm-3')) # for the GaAs cell
base
AlAs, GaAs = material('AlAs')(T=T), material('GaAs')(T=T) # for the DBR
SiO2 = material('SiO2', sopra=True)(T=T) # for the spacer layer
TiO2 = material('TiO2', sopra=True)(T=T) # for the nanoparticles

# some parameters for the QE solver
for mat in [n_GaAs, p_GaAs]:
    mat.hole_mobility, mat.electron_mobility, mat.permittivity = 3.4e-3, 5e-2, 9
    n_GaAs.hole_diffusion_length, p_GaAs.electron_diffusion_length = si("500nm"), si(
        "5um")

# Define the different parts of the structure we will use
GaAs_junction = [Junction([Layer(width=si('100nm'), material=p_GaAs, role="emitter"
    ), Layer(width=si('400nm'), material=n_GaAs, role="base")], T=T, kind='DA')]
DBR = 10 * [Layer(width=si("73nm"), material=AlAs), Layer(width=si("60nm"),
    material=GaAs)]
```

```

# this creates 10 repetitions of the AlAs and GaAs layers, to make the DBR
structure
NP_layer = [Layer(si('50nm'), Air, geometry=[{'type': 'circle', 'mat': TiO2, 'center': (200, 200), 'radius': 50}])]
substrate = [Layer(width=si('50um'), material=GaAs)]
spacer = [Layer(width=si('25nm'), material=SiO2)]

solar_cell = SolarCell(spacer + GaAs_junction + substrate)      # solar cell with
SiO2 coating
opts.optics_method = 'TMM'
solar_cell_solver(solar_cell, 'qe', opts)
TMM_EQE = solar_cell[1].eqe(opts.wavelength)
opts.optics_method = 'BL'
solar_cell_solver(solar_cell, 'qe', opts)
BL_EQE = solar_cell[1].eqe(opts.wavelength)

solar_cell = SolarCell(spacer + GaAs_junction + DBR + substrate)  # as above, with
a DBR on the back
opts.optics_method = 'TMM'
solar_cell_solver(solar_cell, 'qe', opts)
TMM_EQE_DBR = solar_cell[1].eqe(opts.wavelength)

solar_cell = SolarCell(NP_layer + spacer + GaAs_junction + DBR + substrate)
# cell with TiO2 nanocylinder array on the front
opts.optics_method = 'RCWA'
opts.orders = 49          # number of diffraction orders to keep in the RCWA solver
solar_cell_solver(solar_cell, 'qe', opts)
RCWA_EQE_NP = solar_cell[2].eqe(opts.wavelength)
opts.optics_method = 'TMM'
solar_cell_solver(solar_cell, 'qe', opts)
TMM_EQE_NP = solar_cell[2].eqe(opts.wavelength)
opts.optics_method = 'BL'
solar_cell_solver(solar_cell, 'qe', opts)
BL_EQE_NP = solar_cell[2].eqe(opts.wavelength)

plt.figure()
plt.plot(wl*1e9, BL_EQE_NP, wl*1e9, TMM_EQE, wl*1e9, TMM_EQE_DBR, wl*1e9,
RCWA_EQE_NP)
plt.legend(labels=['Beer-Lambert law (all structures)', 'TMM (GaAs SC)', 'TMM (GaAs
SC + DBR)', 'RCWA (GaAs SC + NP array + DBR)'])
plt.xlabel("Wavelength (nm)")
plt.ylabel("Quantum efficiency")
plt.show()

a = np.asarray([opts.wavelength*1e9, BL_EQE, TMM_EQE, TMM_EQE_DBR, RCWA_EQE_NP])
np.savetxt("optcalc_results.csv", np.transpose(a), delimiter=",")

```

Listing 1: Script which defines a thin GaAs solar cell and calculates the QE performance (using the depletion approximation) of a bare cell, and the same cell with a distributed Bragg reflector (DBR) on the rear and a layer of TiO<sub>2</sub> nanocylinders on the front. The optical calculations are carried out using the three optical methods available: BL, TMM and RCWA, in order to compare their suitability for investigating different types of structures.

## 2 Example of a complete solar cell simulation

This tutorial is a step-by-step creation of a solar cell in Solcore and the calculation of their properties using Solcore. The solar cell will include:

- Dual junction GaInP/GaAs solar cell, lattice matched to GaAs.
- The bottom cell has 30 strained-balanced quantum wells (QW), made of GaAsP/InGaAs.

- There is a tunnel junction in between the subcells.
- There is a dual layer anti-reflecting coating on the front made of MgF-ZnS

## 2.1 Defining the structure

First we need to create the solar cell structure. It is made of several bits and pieces: the QWs, 2xJunctions, 1x tunnel junction and the ARC. We start with the first.

### 2.1.1 Defining the QWs

```
from solcore import material
from solcore.structure import Layer
import solcore.poisson_drift_diffusion as PDD
import numpy as np

T = 300
wl = np.linspace(350, 1050, 1001) * 1e-9

# First, we create the materials of the QW
QWmat = material('InGaAs')(T=T, In=0.2, strained=True)
Bmat = material('GaAsP')(T=T, P=0.1, strained=True)
i_GaAs = material('GaAs')(T=T)

# The QW is 7 nm wide, with GaAs interlayers 2 nm thick at each side and GaAsP
# barriers 10 nm thick.
# The final device will have 30 of these QWs.
QW = PDD.CreateDeviceStructure('QW', T=T, repeat=30, substrate=i_GaAs, layers=[
    Layer(width=10e-9, material=Bmat, role="barrier"),
    Layer(width=2e-9, material=i_GaAs, role="interlayer"),
    Layer(width=7e-9, material=QWmat, role="well"),
    Layer(width=2e-9, material=i_GaAs, role="interlayer"),
    Layer(width=10e-9, material=Bmat, role="barrier") ])

# We solve the quantum properties of the QW, leaving the default values of all
# parameters
QW_list = PDD.SolveQWproperties(QW, wavelengths=wl)
```

Listing 2: Initial definitions of the QW structure

The first few lines in Listing 2 import the Solcore utilities needed to define the structure, including the *material* function, the *Layer* class and the Poisson-Drift-Diffusion solver. After that, we create the materials that will make the QWs and create a *Device structure*. We will use these structures just to be able to solve the QWs afterwards and transform it into a sequence of layers with effective properties that the PDD solver understands. The call to *CreateDeviceStructure* has several inputs, including the temperature, the substrate, the number of repetition of the QWs and the structure of the layers. The call to *SolveQWproperties* will, indeed, use the utilities within the quantum mechanics module to calculate the band structure of the QWs, their absorption coefficient and, finally, will calculate effective bandgap, density of states, etc. that the PDD solver will use. Although the device will have 30 quantum wells, only one unit (the one indicated in *Layers*) will be modelled as an isolated QW.

While Solcore can solve the Schrödinger equation in a structure with any number of layers, the absorption calculator for QWs can only deal properly with single QWs. That is the reason of modelling only 1 QW despite having 30 in the structure. This will clearly represent a limitation when modelling the absorption of superlattices, where there is a strong coupling between neighbouring QWs.

In the code above, we have used the *PDD.SolveQWproperties* function with the default values, but it can have a number of optional input parameters to define the number of confined states to calculate, the energy of quasiconfined states, electric field, boundary conditions, etc.

### 2.1.2 Defining the junctions

In order to calculate the properties of a solar junction using the PDD solver, we need to give all the layers and materials the junction is made of, in a similar way we have done for the QWs. One thing

to note is that if Solcore cannot find a property it needs to solve the PDD equations, it will take the corresponding property for GaAs as a default value.

The different code snippets are additive in order to get a final, complete script. Normally, all the import statements would be packed together at the beginning.

```
from solcore.structure import Junction

## Materials for the BOTTOM junction
window_bottom = material('GaInP')(T=T, Nd=5e24, In=0.49)
n_GaAs = material('GaAs')(T=T, Nd=1e24)
p_GaAs = material('GaAs')(T=T, Na=8e22)
bsf_bottom = material('GaInP')(T=T, Na=5e24, In=0.49)

GaAs_junction = Junction([Layer(width=10e-9, material=window_bottom, role="Window"),
                           Layer(width=150e-9, material=n_GaAs, role="Emitter")] +
                           QW_list +
                           [Layer(width=2000e-9, material=p_GaAs, role="Base"),
                            Layer(width=200e-9, material=bsf_bottom, role="BSF")], sn=1
                           e6, sp=1e6, T=T, kind='PDD')

## Materials for the TOP junction
window_top = material('AlInP')(T=T, Nd=5e23, Al=0.53)
n_GaInP = material('GaInP')(T=T, Nd=5e23, In=0.49)
p_GaInP = material('GaInP')(T=T, Na=8e22, In=0.49)
bsf_top = material('AlInP')(T=T, Na=5e24, Al=0.53)

GaInP_junction = Junction([Layer(width=120e-9, material=n_GaInP, role="Emitter"),
                           Layer(width=800e-9, material=p_GaInP, role="Base")], sn=1
                           e3, sp=1e3, T=T, kind='PDD')
```

Listing 3: Definition of the junctions

As it can be seen in Listing 3, while we have defined the window and back surface field layer (BSF) for the TOP junction, we have not included it into the *Junction* definition. The reason for this is that very wide bandgap materials cause convergence problems when doing calculations under illumination, specially when working as the front-most window layers or in distributed Bragg reflectors, where there are very abrupt changes of properties in very short distances. In order to account for their presence, two things are done: (1) the surface recombination velocity of the top junction is set to a low value to mimic the passivating effect of the window and BSF layers, and (2) the missing layers are added outside the *Junction* object when creating the full solar cell in order to consider their optical properties (see below).

The only tunnel junction of this solar cell will be defined according to the parametric model and we will assume it is made of GaInP layers, 40 nm-thick in total, that will block part of the light reaching the bottom junction. The code is in Listing 4. Since the top junction is also made of GaInP, most of the light should already be absorbed and therefore it should not represent a very important loss. We will use a relatively low peak current to demonstrate the effect of tunnel junction breakdown when working at high concentration.

```
from solcore.structure import TunnelJunction

tunnel = TunnelJunction([Layer(width=40e-9, material=n_GaInP, role="TJ")],
                        v_peak=0.2, j_peak=7.5e4, v_valley=1, j_valley=4e4, prefactor
                        =5, j01=1e-23, kind='parametric', pn=True)
```

Listing 4: Definition of the tunnel junction

### 2.1.3 Defining the AR coating

The AR coating will reduce the front surface reflection and, therefore, increase the photocurrent of the solar cell. We use a simple dual layer coating made of MgF<sub>2</sub> and ZnS. Both materials are available in the SOPRA database of optical constants ("MgF2" and "ZnScub", respectively). The data for MgF<sub>2</sub> only extends to 900 nm, but we will assume its optical properties will be similar at longer wavelengths,

extrapolating the available data. The code simply consists on importing those materials, as shown in Listing 5.

```
MgF2 = material('MgF2')()
ZnS = material('ZnScub')()
```

Listing 5: Definition of the anti reflecting coating

### 2.1.4 Creating the solar cell

With all the materials and structures defined, we just need to put everything together, including the front window layer and the BSF layer of the top junction that we had left outside (Listing 6).

```
from solcore.solar_cell import SolarCell

my_solar_cell = SolarCell([Layer(width=110e-9, material=MgF2, role="ARC1"),
                          Layer(width=60 - 9, material=ZnS, role="ARC2"),
                          Layer(width=30e-9, material=window_top, role="window"),
                          GaInP_junction,
                          Layer(width=100e-9, material=bsf_top, role="BSF"),
                          tunnel,
                          GaAs_junction],
                          T=T, substrate=n_GaAs)
```

Listing 6: Creation of the MJ solar cell

## 2.2 Calculating the external quantum efficiency

With the structure fully defined, now it is possible to calculate its behavior as solar cell. We first calculate the external quantum efficiency at 1 Sun of intensity using the AM1.5g solar spectrum. The code that defines the spectrum to use and lunch the calculation will be:

```
from solcore.solar_cell_solver import solar_cell_solver
from solcore.light_source import LightSource

light_source = LightSource(source_type='standard', version='AM1.5g', x=w1,
                           output_units='photon_flux_per_m', concentration=1)

solar_cell_solver(my_solar_cell, 'qe',
                  user_options={'light_source': light_source, 'wavelength': wl,
                                'optics_method': 'TMM'})
```

Listing 7: Calculation of the EQE

The calculation will proceed in several stages: (1) The structure is processed and discretized, (2) the problem is solved under equilibrium, (2) the problem is solved under short circuit conditions increasing in small steps the intensity of the illumination, and (3), the quantum efficiency is calculated, one wavelength at a time. This is done for the two junctions. Figure 1 shows the electrons and holes densities at equilibrium (dashed line) and at short circuit (continuous line), as well as the EQE of the two junctions and the total absorption in the layers.

As it can be seen, the minority carrier population increases significantly under illumination, specially in the QW region of the bottom cell, which is a relatively thick, undoped region. The EQE of the bottom junction, in comparison with the total absorbed light, indicates excellent carrier collection with only some missing carriers in the region of the spectrum where light is absorbed by the QWs. The top junction is not performing that well, the reason being a combination of low mobility of carriers in GaInP (between 3 and 6 times smaller than that of p-GaAs), insufficient thickness to absorb all light (part of it reaching the GaAs junction), parasitic absorption of the AlGaP window layer and surface recombination.

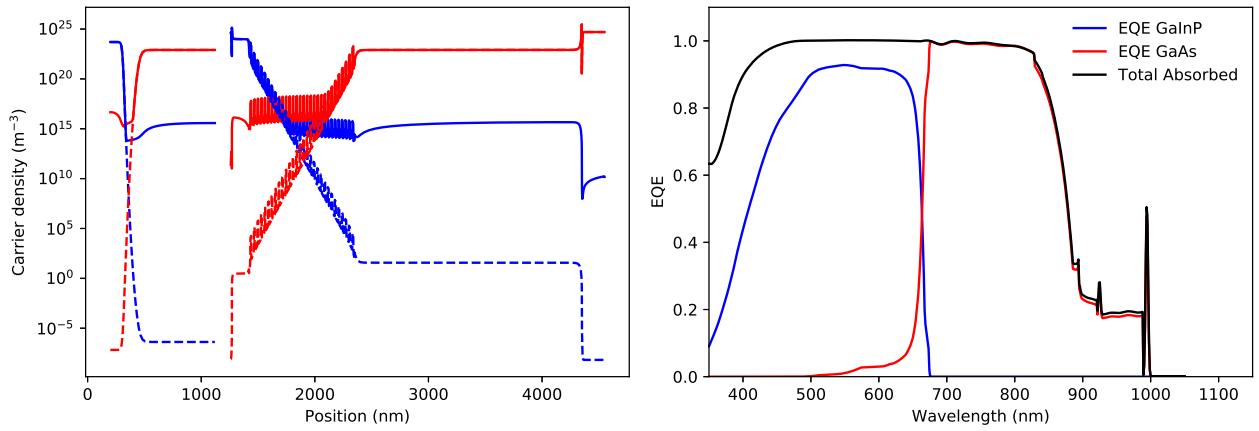


Figure 1: (a) Carrier densities in equilibrium (dashed lines) and at short circuit (continuous lines) as a function of the position inside the cell. The left part is the GaInP junction, separated from the GaAs subcell on the right by a gap left by the tunnel junction. (b) EQE of the two junctions and the total absorption in the layers.

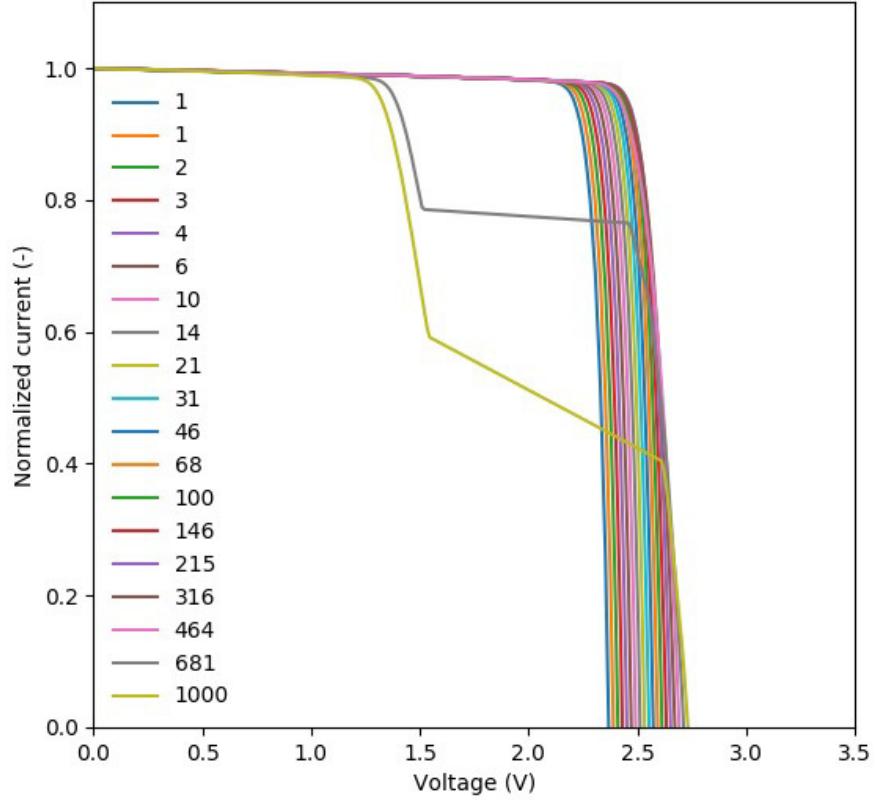


Figure 2: All IV curves normalized by their corresponding  $I_{SC}$ .

## 2.3 Calculating the IV characteristics

The calculation of the current-voltage characteristics proceed in a similar way. In this case, care must be taken to use the correct polarity for the (internal) voltage sweep, and the polarity of the tunnel junction, already defined above as  $pn$ . As the optical properties have already been calculated, we set the optics method to None, so they are not calculated again. Listing 8 shows the definition of the voltages, the sunlight concentration factor (from 1 to 1000 suns) and the execution of the calculation within a for loop.

Note that this example will take several hours to run due to the presence of QWs, which increase the number of mesh points in the bottom cell to several thousand. To have an example of the results faster, reduce the number of concentrations (19 in the example) and comment the QWs related line in the definition of the GaAs junction in Listing 3.

```
con = np.logspace(0, 3, 19)
vint = np.linspace(-3.5, 4, 600)
V = np.linspace(-3.5, 0, 300)

allI = []
isc = []
voc = []
FF = []
pmpp = []

for c in con:
    light_source.options['concentration'] = c

    solar_cell_solver(my_solar_cell, 'iv',
                      user_options={'light_source': light_source, 'wavelength': wl,
                                    'optics_method': None,
                                    'light_iv': True, 'mpp': True, 'voltages': V,
                                    'internal_voltages': vint})

    isc.append(my_solar_cell.iv['Isc'])
    voc.append(my_solar_cell.iv['Voc'])
    FF.append(my_solar_cell.iv['FF'])
    pmpp.append(my_solar_cell.iv['Pmpp'])
    allI.append(my_solar_cell.iv['IV'][1])
```

Listing 8: Calculation of the IV curves

Figure 3 shows the summary of the efficiency,  $I_{SC}$ ,  $V_{OC}$  and fill factor (FF) versus concentration. The efficiency of the cell is just above 30% at 1 sun, and increases with concentration , peaking at around 200-300 suns. Further increasing the concentration rapidly decreases the efficiency. Both,  $I_{SC}$  and  $V_{OC}$  follow the expected linear trend in log-log and semilog plots, respectively, while the fill factor increases for low concentrations and then continuously drops from 20 suns and, specially, from 200 suns. The initial smooth drop of the fill factor can be related with a limited carrier transport across the QW region, that under concentration, suffers from charge accumulation and damping of the electric field. However, the abrupt drop at higher concentrations is related to the failure of the tunnel junction which, by design, had a low peak current.

This effect can be better seen in Figure 2 where, for the two highest concentrations, the IV curve exhibit steps due to the tunnel junction limiting the current.

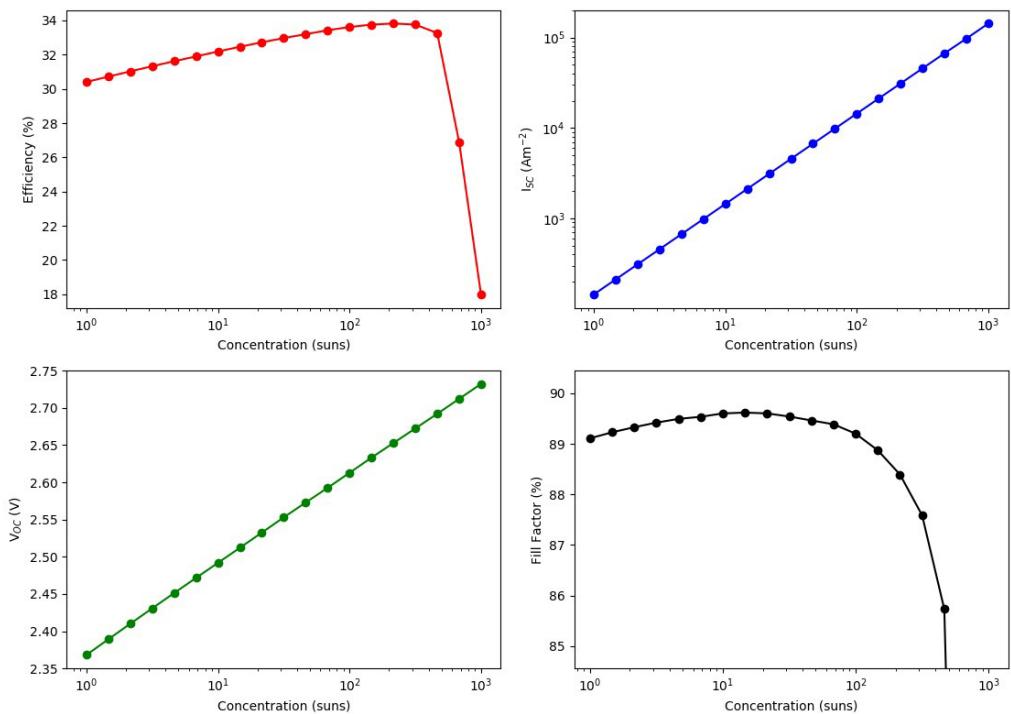


Figure 3: Summary of the efficiency,  $I_{SC}$ ,  $V_{OC}$  and fill factor versus concentration.