

# Manual for solar cell optical simulation software: GENPRO4

---

Rudi Santbergen ([r.santbergen@tudelft.nl](mailto:r.santbergen@tudelft.nl))  
Photovoltaic Materials and Devices  
Delft University of Technology

Version: February 2016

## 0. Introduction

GENPRO4 is an optical model for solar cells that combines wave and ray-optics in an computationally efficient way. The model has been validated for a wide range of wafer based and thin-film solar cells. GENPRO4 represents the solar cell as a multilayer structure and calculates the fraction of incident light absorbed in each layer, taking into account scattering and trapping of light at the interfaces. From this, the implied photocurrent is calculated. Note that GENPRO4 is a purely *optical* model and does not consider the *electrical* characteristics of the solar cell. Also note that GENPRO4 is programmed in Matlab and requires Matlab to run.

Below the basic working principles of GENPRO4 are explained. Examples are provided to illustrate how to compose the input file, run the simulation and extract the output data.

## 1. Flat interfaces

Most types of solar cell can, to a good approximation, be represented as a multilayer structure as shown in Fig. 1. The layers and interfaces are numbered from top to bottom. The subscript  $i$  will be used to indicate the layer or interface numbers. The goal of the simulation is to determine the overall reflectance  $R$ , the transmittance  $T$  and the absorptance of each layer  $A_i$ . Each layer is characterized by thickness  $d_i$  and complex refractive index  $N_i(\lambda)$ , where  $\lambda$  is the wavelength. Because  $N_i$  is wavelength dependent,  $R$ ,  $T$  and  $A_i$  are wavelength dependent as well. Below it is explained how  $R$ ,  $T$  and  $A_i$  are calculated for a single wavelength. This calculation is then repeated for every wavelength in the relevant wavelength range.

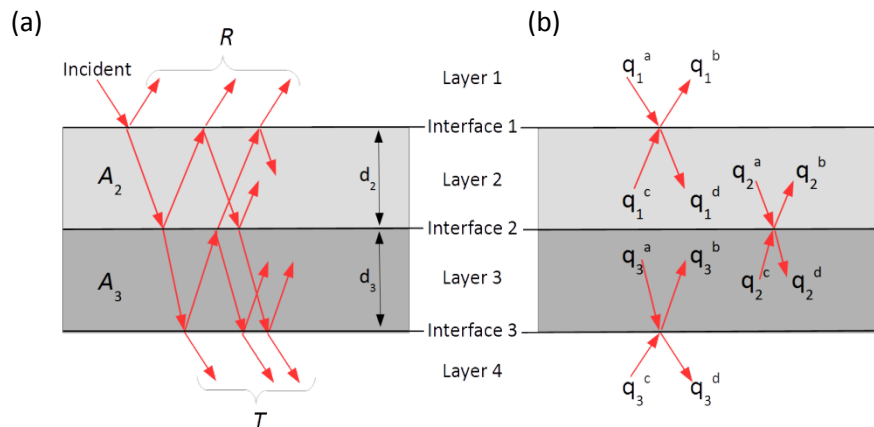


Fig. 1: Schematic representation of a multilayer structure showing the numbering convention for layers and interfaces. a) Various optical paths contributing to  $R$ ,  $T$  and  $A_i$ . b) Net-radiation fluxes.

When all interfaces are optically flat, the interface reflectances  $r_i$  and the layer transmittances  $\tau_i$  can be calculated from the Fresnel equation and Lambert-Beer law, respectively. Note that the values of  $r_i$  and  $\tau_i$  depend on the angle of incidence. An incident photon can reflect at the interfaces multiple times and travel a complicated path (see Fig. 1a). Therefore, to calculate  $R$ ,  $T$  and  $A_i$  from  $r_i$  and  $\tau_i$ , all possible

reflections have to be taken into account. There are different mathematically equivalent methods to do this. A convenient method, the net-radiation method [1,2], is illustrated in Fig. 1b. At every interface four fluxes are defined:  $q_i^x$ . Here subscript  $i$  is the interface number and the superscript  $x$  ( $= a, b, c$  or  $d$ ) indicates whether the light is approaching/leaving the interface from the top/bottom, as defined in Fig. 1b. Each flux represents the total intensity (in  $W/m^2$ ) due to all possible photon paths. All fluxes are related by a set of linear equations

$$\begin{cases} q_i^a = \tau_i \cdot q_{i-1}^d \\ q_i^b = r_i \cdot q_i^a + t_i \cdot q_i^c \\ q_i^c = \tau_{i+1} \cdot q_{i+1}^b \\ q_i^d = t_i \cdot q_i^a + r_i \cdot q_i^c \end{cases} \quad (1)$$

where  $t_i = 1 - r_i$ . In total there are  $4 \cdot l$  equations, where  $l$  is the total number of interfaces. It is convenient to normalize all fluxes to the incident power, such that they can be written in non-dimensional form. The assumption that all light is incident from the top and none from the bottom then implies that  $q_1^a = 1$  and  $q_l^c = 0$ . Eq. (1) represents a set of linear equations that can be solved using standard numerical techniques to obtain the values of every flux. From this  $R$ ,  $T$  and  $A_i$  are obtained

$$R = q_1^b \quad (2)$$

$$T = q_l^d \quad (3)$$

$$A_i = q_{i-1}^d - q_{i-1}^c + q_i^b - q_i^a \quad (4)$$

Note that  $A_i$  is simply the sum of fluxes entering minus the fluxes leaving layer  $i$ .

### Example 1: polished c-Si wafer

The goal of this example is to simulate  $R$ ,  $T$  and  $A_i$  of a 200  $\mu\text{m}$  thick double side polished c-Si wafer. Fig. 2a shows the corresponding input file 'example1.m', provided with GENPRO4. For every GENPRO4 simulation three inputs are required:

1. A structure with layer information 'Lay', specifying the medium 'med' and thickness 'thi' of each layer in  $\mu\text{m}$ . This optical structure consists of 3 layers (air/c-Si/air), of which the first and final are infinitely thick. Note that the name of the materials 'air' and 'c-Si' corresponds with the files 'air.nk' and 'c-Si.nk' in the nk-folder (see appendix A).
2. A structure with interface information 'Int'. In this case both interfaces are optically flat so the 'flat' model is used and no further information is required.
3. A vector 'wav' indicating the wavelengths in  $\mu\text{m}$ . In this example the simulation will be performed for the wavelengths: 300, 320, ..., 1200 nm.

The final line starts the simulation with the input 'Lay', 'Int' and 'wav'.

This file can be run by Matlab as a script by pressing the F5 key or the 'run' button in Matlab. This simulation should finish in a few seconds and the output is shown in Fig. 2b. The results show  $R$  (white),  $T$  (black) and  $A$  (orange) as a function of wavelength. As expected, shorter wavelengths ( $\lambda < 900$  nm) are either absorbed or reflected but not transmitted. The largest wavelengths ( $\lambda \approx 1200$  nm) are reflected and transmitted but not absorbed. It can be

checked that energy is conserved because for every wavelength  $R + T + \sum A_i = 1$ . GENPRO4 automatically integrates  $R$ ,  $T$  and  $A_i$  over the AM1.5 spectrum to calculate the corresponding implied photocurrents (in mA/cm<sup>2</sup>). These are shown in the legend of Fig. 2b.

(a)

```
Lay(1).med = 'air';      Lay(1).thi = inf;
Lay(2).med = 'c-Si';    Lay(2).thi = 200;
Lay(3).med = 'air';    Lay(3).thi = inf;

Int(1).model = 'flat';
Int(2).model = 'flat';

wav = 0.300:0.20:1.200;

GENPRO4(Lay, Int, wav)
```

(b)

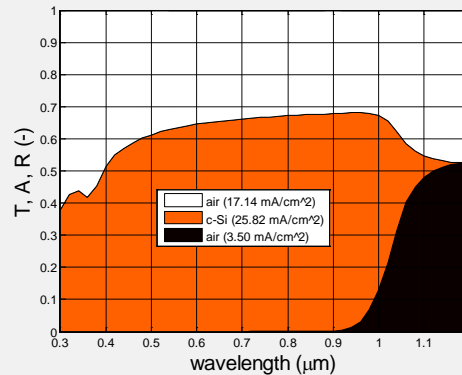


Fig. 2: a) Input file 'example1.m' for simulating a double side polished c-Si wafer. b) Simulation output: an area plot of  $R$ ,  $T$  and  $A_i$  versus wavelength.

## 2. Interference

The fluxes  $q$  introduced in section 1 represent the light intensities in W/m<sup>2</sup>. In eq. (1) and (4) these fluxes are simply being added without taking into account interference effects. This approach is only accurate for so-called incoherent layers that are *thick* compared to the coherence length of the incident sunlight ( $\approx 1 \mu\text{m}$ ). For *thin* (coherent) layers, interference *does* play a role and a different calculation method should be used in which the fluxes represent the complex amplitudes of electromagnetic waves [3]. In GENPRO4 these two distinct approaches can be combined by treating the thin (coherent) layers as a 'coating', which is part of the interface between two thick (coherent) 'layers'. In GENPRO4 'layers' are treated incoherently and do not give rise to interference while 'coatings' are treated coherently and do give rise to interference (see example 2).

### Example 2: flat silicon heterojunction cell

The goal of this example is to simulate a silicon heterojunction (SHJ) cell with optically flat interfaces. The SHJ cell is based on the polished c-Si wafer of example 1 with several thin layers deposited on both sides. It has an about 80 nm thin ITO coating that works as an anti-reflective coating due to interference. To include the interference effect in the simulation, the ITO and other thin layers are *not* treated as a 'layer' but as a 'coating' at the air/c-Si interface. Therefore the interface structure 'Int' has a new field 'coat'. On both front and rear there is a multilayer coating. At the front: ITO (80 nm)/a-Si(p) (5 nm)/a-Si(i) (5 nm) and at the rear a-Si(i) (5 nm)/a-Si(p) (5 nm)/silver (200 nm). Note that the first line clears any 'Lay' and 'Int' structures that may have been left in Matlab's workspace from previous simulations

The result of the simulation is shown in Fig. 3b. As expected, anti-reflection effect of the ITO reduces  $R$  (white area) considerably and as a result the absorption in c-Si increased (orange area). As indicated in the legend, the implied photocurrent of the c-Si wafer increased

to  $33.5 \text{ mA/cm}^2$  (was  $25.8 \text{ mA/cm}^2$ ). The other colours indicate that there is a small amount of absorption in the other layer as well. Because the silver coating at the rear is opaque, no light is transmitted.

(a)

```
clear Lay Int
Lay(1).med = 'air';           Lay(1).thi = inf;
Lay(2).med = 'c-Si';         Lay(2).thi = 200;
Lay(3).med = 'air';           Lay(3).thi = inf;

Int(1).model = 'flat';       %use ray-optics model
Int(1).coat(1).med = 'ITO';   Int(1).coat(1).thi = 0.080;
Int(1).coat(2).med = 'a-Si(p)'; Int(1).coat(2).thi = 0.005;
Int(1).coat(3).med = 'a-Si(i)'; Int(1).coat(3).thi = 0.005;

Int(2).model = 'flat';       %use ray-optics model
Int(2).coat(1).med = 'a-Si(i)'; Int(2).coat(1).thi = 0.005;
Int(2).coat(2).med = 'a-Si(n)'; Int(2).coat(2).thi = 0.005;
Int(2).coat(3).med = 'silver'; Int(2).coat(3).thi = 0.200;

wav = 0.300:0.020:1.200;

GENPRO4(Lay,Int,wav);
```

(b)

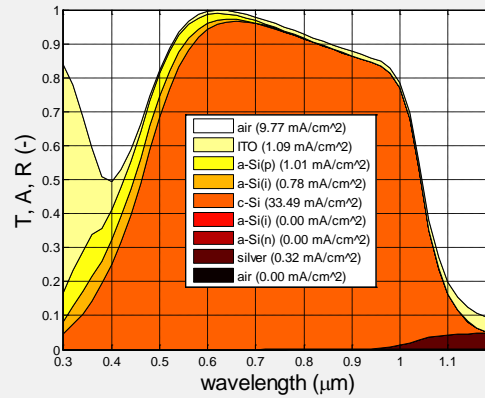


Fig. 3: a) Input file 'example2.m' for simulating a flat SHJ cell. b) Simulation output: an area plot of  $R$ ,  $T$  and  $A$ , versus wavelength.

### 3. Textured interfaces

Most interfaces in the solar cell have a surface texture to reduce reflection losses and to scatter incident light. In that case, instead of having a well-defined propagation direction, the reflected and transmitted light is distributed over a range of angles. The calculation method described in section 1 can be extended by dividing the range of angles into angular intervals. The propagation angle is measured relative to the surface normal direction and ranges from  $0^\circ$  (perpendicular to the interface) to  $90^\circ$  (parallel to the interface). Typically the number of angular intervals is set to 30, resulting in an angular width of the intervals of  $3^\circ$ . Each flux can then be divided into 30 sub-fluxes. All relations between these sub-fluxes can still be written as a set of linear equations. Note that this set now contains  $4 \cdot I \cdot V$  equations, where  $I$  and  $V$  are the number of interfaces and the number of intervals, respectively. Typically this results in a large set of hundreds of equations. However, using standard numerical techniques a computer can solve such a set of linear equations within a fraction of a second.

It is convenient to indicate each sub-flux in the following way:  $q_i^x(v)$ , where  $x$  and  $i$  have the same meaning as explained in section 1, and  $v$  is the interval number. For example  $q_2^a(5)$  is the flux approaching interface 2 from above in the 5<sup>th</sup> angular interval (i.e. the interval ranging from  $12^\circ$  to  $15^\circ$ ). The fluxes can then be grouped into vectors  $\mathbf{q}_i^x = [q_i^x(1), q_i^x(2), \dots, q_i^x(V)]$ , where the bold font indicates the vector.  $V$  is the number of intervals, which typically is 30. This vector notation allows the large set of equations to be written in a compact way using matrix multiplication

$$\begin{cases} \mathbf{q}_i^a = \boldsymbol{\tau}_i \cdot \mathbf{q}_{i-1}^d \\ \mathbf{q}_i^b = \mathbf{r}_i^+ \cdot \mathbf{q}_i^a + \mathbf{t}_i^- \cdot \mathbf{q}_i^c \\ \mathbf{q}_i^c = \boldsymbol{\tau}_{i+1} \cdot \mathbf{q}_{i+1}^b \\ \mathbf{q}_i^d = \mathbf{t}_i^+ \cdot \mathbf{q}_i^a + \mathbf{r}_i^- \cdot \mathbf{q}_i^c \end{cases} \quad (6)$$

Note that the coefficients  $\mathbf{r}_i$ ,  $\mathbf{t}_i$  and  $\boldsymbol{\tau}_i$  are now also in bold to indicate that these are now *matrices* of size  $V \times V$ . The matrices  $\mathbf{r}_i$  and  $\mathbf{t}_i$ , to which we will refer as *scatter matrices*, contain the angular intensity distribution of scattered light for every angle of incidence (also known as the bi-directional scatter distribution function). The  $\mathbf{r}_i$  and  $\mathbf{t}_i$  matrices for light incident on the interface from the top and bottom are different and the superscripts '+' and '-' are used to distinguish them. The scatter matrices are calculated from the surface morphology of the interface using either the ray-optics model or wave-optics model. The ray-optics model uses a standard ray-tracing technique. This model is suitable for surface morphologies with features larger than the wavelength. The wave-optics model is based on the scalar scattering model developed by Jäger et al. [4]. This model is more suitable for surface morphologies with features smaller than the wavelength. Once the matrices are calculated, they can be substituted into eq. (6) and the set of equations can be solved.

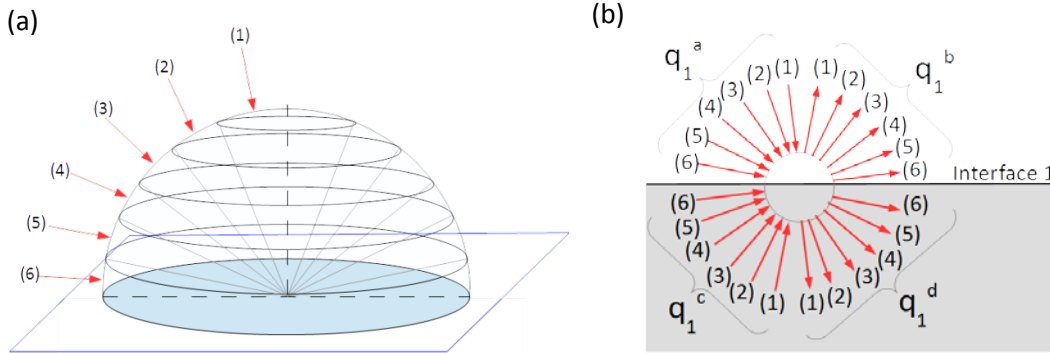


Fig. 4: a) Division of every hemispherical direction into angular intervals. b) Net-radiation sub-fluxes at interface 1. (For clarity the figure shows six intervals. GENPRO4 typically uses 30 intervals for higher precision.)

The vector  $\mathbf{q}_1^b$  contains all fluxes leaving the top of the first interface (see Fig. 4b) and therefore contains the information regarding the angular intensity distribution of light reflected by the multilayer structure. The total reflectance  $R$  is simply sum of the intensity of all the elements of this vector  $\sum \mathbf{q}_1^b = q_1^b(1) + q_1^b(2) + \dots + q_1^b(V)$ . Similar to eq. (2-4),  $R$ ,  $T$  and  $A_i$  are given by

$$R = \sum \mathbf{q}_1^b \quad (7)$$

$$T = \sum \mathbf{q}_i^d \quad (8)$$

$$A_i = \sum \mathbf{q}_i^d + \sum \mathbf{q}_{i+i}^b - \sum \mathbf{q}_i^c - \sum \mathbf{q}_{i+1}^a \quad (9)$$

### Example 3: Pyramid textured SHJ cell

The goal of this example is to simulate the SHJ cell of example 2, but with a pyramid texture on both sides. The input for this simulation is shown in Fig. 5a. As in example 2, there are three 'layers' (air/c-Si/air) and the 'coatings' are part of the interfaces. The pyramid texture consists of approximately 5  $\mu\text{m}$  wide pyramids with a steepness of 54.7°. These large features should be modelled with the ray-optics model therefore the interface model was changed from 'flat' to 'ray'. The surface morphology shown in Fig. 5b is available in the form of a height matrix 'H', which is loaded from a file. The corresponding lateral size is 20  $\mu\text{m}$  and has to be specified as well. Note that at the second interface a minus-sign is added to height matrix 'H' to assure that on the bottom interface the pyramids are pointing downward.

The simulation result shown in Fig. 5c reveals that the pyramid texture has a strong anti-reflective effect and even more light is absorbed in c-Si (orange area). As a result, the implied photocurrent of this SHJ cell increased from 33.5  $\text{mA}/\text{cm}^2$  to 38.5  $\text{mA}/\text{cm}^2$ . The current densities of the other layers give an indication of the parasitic absorption losses. Zhang et al. compared the simulated reflectance and absorptance in c-Si with measured reflectance and EQE and showed excellent agreement (not shown here) [5].

In this example '[Lay,Int] =' was added to the line that calls GENPRO4. This collects the output as numeric values in the structures 'Lay' and 'Int'. This data can then be plotted or exported to a file manually, by clicking the variables in the Matlab workspace. This can also be done automatically, as the last lines of this example show. More information about accessing and exporting data in matrices or structures can be found in Matlab's help.

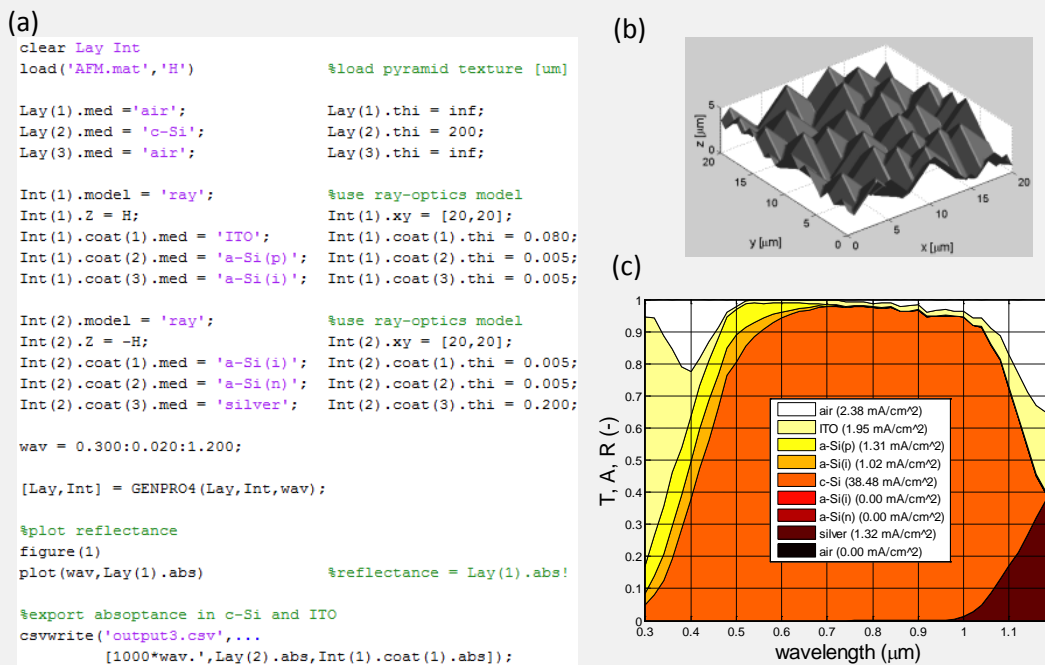


Fig. 5: a) Input file 'example3.m' for simulating a SHJ cell with pyramid texture. b) The pyramid texture used in the simulation. c) Simulation output: an area plot of  $R$ ,  $T$  and  $A_i$  versus wavelength.

## 4. Parameter variation

When instead of 'flat' the 'ray' or 'wave' model is used the simulation time increases from seconds to minutes. Most of the additional computation time is spent on

calculating the scatter matrices  $r_i$  and  $t_i$  for every wavelength. When repeating the simulation with a different layer thickness, the scatter matrices stay the same. In that case the previously calculated scatter matrices can be re-used without recalculation to save computation time (see example 4). Note that when changing the properties of an *interface*, e.g. surface morphology or coating thickness, the scatter matrices of that interface change and should not be re-used.

#### Example 4: c-Si wafer thickness

The goal of this example is to investigate the effect of varying the c-Si wafer thickness on the implied photocurrent of the SHJ cell of example 3. This example should be run after running example 3, so that the input structures 'Lay' and 'Int' are available in the workspace. As shown in Fig. 6a, a for-loop is used to repeat the simulation while varying the wafer thickness 'Lay(2).thi' from 50 to 500  $\mu\text{m}$ , in steps of 5  $\mu\text{m}$ . In the end the implied photocurrent is plotted versus wafer thickness.

In this example the scatter matrices are re-used to speed up the simulation. After running example 3, the input structure 'Int' contains the scatter matrices. To make GENPRO4 re-use the scatter matrices the model is set to 'prev', which is done before the for-loop. As result, a single simulation takes only about one second. Fig. 9b shows the implied photocurrent of the c-Si wafer as a function of its thickness. This is the result of 96 simulations (each of 91 wavelengths) and finished within two minutes. As expected, a thicker wafer absorbs more light and gives rise to a higher implied photocurrent.

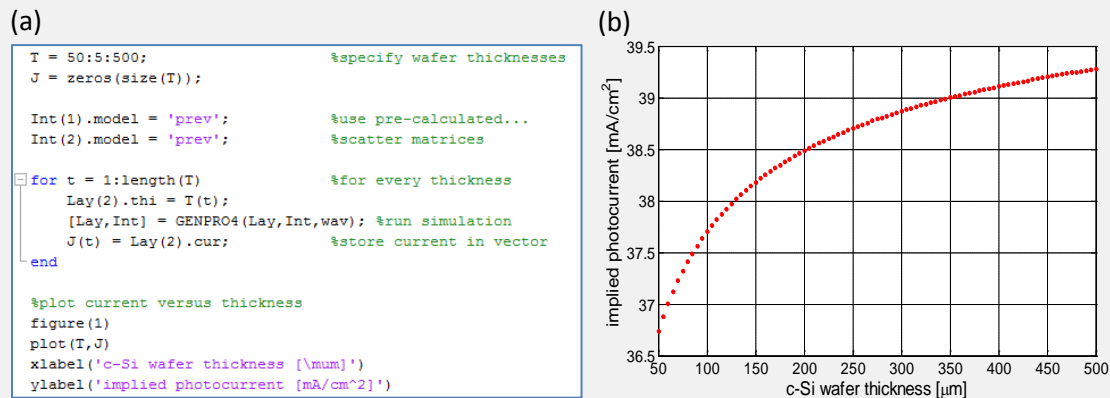


Fig. 6: a) Input file 'example4.m' for varying c-Si wafer thickness of the SHJ cell of example 3. b) Simulation output: implied photocurrent versus c-Si wafer thickness calculated in two minutes.

## 5. Concluding remarks

The previous examples demonstrate the power and flexibility of GENPRO4 for simulation of wafer based c-Si solar cells. Besides c-Si solar cells, thin-film solar cells or combinations of both have been simulated successfully. The possibility of quickly calculating the implied photocurrent as a function of absorber layer thickness is especially useful for matching the current in multi-junction solar cells [6].



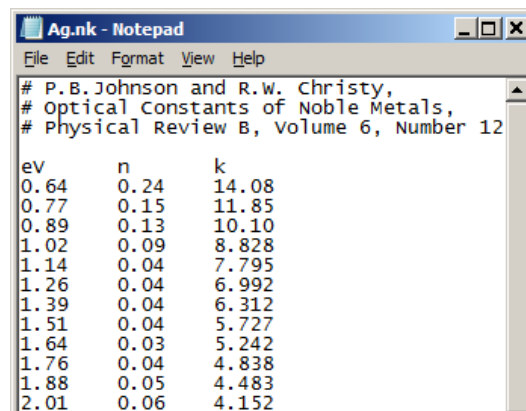
## References

- [1] R. Siegel, Net radiation method for transmission through partially transparent plates, *Solar Energy* **15** (1973) p. 273-276.
- [2] R. Santbergen, R.J.C. van Zolingen, The absorption factor of crystalline silicon PV cell: A numerical and experimental study, *Solar Energy Materials and Solar Cells* **92** (2008) p. 432-444.
- [3] R. Santbergen, A.H.M. Smets, M. Zeman, Optical model for multilayer structures with coherent, partly coherent and incoherent layers, *Optics Express* **21** (2013) p. A262-267.
- [4] K. Jäger, M. Fischer, R.A.C.M.M. van Swaaij, M. Zeman, A scattering model for nano-textured interfaces and its application in opto-electrical simulations of thin-film silicon solar cells, *Journal of Applied Physics* **111** (2012) p. 083108.
- [5] D. Zhang, I.A. Digdaya, R. Santbergen, R.A.C.C.M. van Swaaij, P. Bronsveld, M. Zeman, J.A.M. van Roosmalen, A.W. Weeber, Design and fabrication of SiO<sub>x</sub>/ITO double-layer anti-reflective coating for heterojunction silicon solar cells, *Solar Energy Materials and Solar Cells* **117** (2013) p. 132-138.
- [6] H. Tan, A. Furlan, W.Li, K. Arapov, R. Santbergen, M.M. Wienk, M. Zeman, A.H.M. Smets, R.A.J. Janssen, Highly efficient hybrid polymer amorphous silicon multijunction solar cells with effective optical management, *Advanced materials* (2016) p. 1-8.

## Appendix A: Creating nk-files

The nk-folder provided with GENPRO4 holds many nk-files, each containing a table of  $n$  and  $k$  versus wavelength for one material. An nk-file is a plain text file with '.nk' extension and can be opened in a text editor such as Notepad. Only materials for which the nk-file is present can be used in the simulations. If another material is required, a corresponding nk-file has to be created and placed in the nk-folder. As shown in Fig. A1 below, the nk-file consists of four parts:

1. First column: here the wavelength is specified. The column heading can be 'nm', 'um' or 'm' depending whether the wavelength is specified in nm,  $\mu\text{m}$  or m. Alternatively it is possible to specify the photon energy in eV, in that case the column heading should be 'eV'.
2. Second column: here the refractive index  $n$  is specified, which is dimensionless. The column heading should be 'n'.
3. Third column: here the extinction coefficient  $k$  should be specified, which is dimensionless. The column heading should be 'k'. Alternatively it is possible to specify the absorption coefficient  $\alpha$ . In that case the column heading should be 'alpha' and it is assumed that the units are  $\text{cm}^{-1}$ .
4. Heading (optional): here background information such as the source of the data can be provided. This information is not used by GENPRO4.



```
Ag.nk - Notepad
File Edit Format View Help
# P.B.Johnson and R.W. Christy,
# Optical Constants of Noble Metals,
# Physical Review B, Volume 6, Number 12
eV      n      k
0.64    0.24   14.08
0.77    0.15   11.85
0.89    0.13   10.10
1.02    0.09   8.828
1.14    0.04   7.795
1.26    0.04   6.992
1.39    0.04   6.312
1.51    0.04   5.727
1.64    0.03   5.242
1.76    0.04   4.838
1.88    0.05   4.483
2.01    0.06   4.152
```

Fig. A1: Example of nk-file showing heading and data in three columns.

'readnk.m' is one of the files provided with GENPRO4. It contains the function that reads the data from the nk-files. It uses the Matlab function 'importdata', which may not work properly in case the file contains redundant spaces or tabs. In case an error message appears when using a new nk-file for the first time, the file most likely contains redundant spaces or tabs that should be removed, especially near the top of the table.

In case a material is considered with a constant (not wavelength dependent)  $n$  and  $k$  no nk-file is needed. In that case it is possible to specify the complex refractive index  $N = n + ik$  directly in the input file. For example instead of writing 'ITO' (which refers to the 'ITO.nk' file) one could use '1.85' or '1.85 + 0.03i' without requiring a corresponding nk-file. Note that the number should be given in quotation marks.