

University of Ljubljana
Faculty of Electrical Engineering
Laboratory of Photovoltaics and Optoelectronics

CROWM

Combined ray optics / wave optics model

User Manual

Version 3.3.1

Ljubljana, January 2019

CROWM

User Manual

Benjamin Lipovšek, Janez Krč, Marko Topič

Copyright © 2019

University of Ljubljana

Faculty of Electrical Engineering

Laboratory of Photovoltaics and Optoelectronics

All rights reserved.

Ljubljana, January 2019



Laboratory of Photovoltaics and Optoelectronics
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana
Slovenia
<http://lpvo.fe.uni-lj.si>

Head of LPVO:

Prof. Dr. Marko Topič
E-mail: marko.topic@fe.uni-lj.si
Tel.: +386 1 4768 470
Fax: +386 1 4264 630

Author and contact person of CROWM:

Dr. Benjamin Lipovšek
E-mail: benjamin.lipovsek@fe.uni-lj.si
Tel.: +386 1 4768 845

Official CROWM webpage:

<http://lpvo.fe.uni-lj.si/en/software/crowm/>

Contents

1.	INTRODUCTION.....	1
1.1	Short overview	1
1.2	Description of the general simulated structure.....	2
1.3	Description of the optical model.....	3
2.	INSTALLATION OF CROWM	4
2.1	Installation of CROWM program files.....	4
2.2	Installation of the MATLAB Compiler Runtime (MCR)	4
2.3	Files and folder structure of CROWM.....	5
3.	RUNNING CROWM FROM COMMAND LINE ...	6
3.1	Execution of a single simulation	6
3.2	Running a batch of simulations.....	7
3.3	Running CROWM from other programs	8
4.	RUNNING CROWM USING THE GRAPHICAL USER INTERFACE (GUI)	9
4.1	Main simulation panel.....	9
4.2	Texture viewer	16
4.3	Ray propagation viewer	17
4.4	Results viewer	18

5.	DESCRIPTION OF THE INPUT FILES	20
5.1	Simulation input files (".txt", located in "In" folder).....	20
5.2	Complex refractive index files (".nk", located in "nk" folder).....	30
5.3	Spectrum files (".spc", located in "Spectrum" folder)	31
5.4	Texture files (".txt", located in "Texture" folder)	31
6.	DESCRIPTION OF THE OUTPUT FILES	33
6.1	Log file ("_Log.txt").....	33
6.2	Reflectance, transmittance, absorptance file ("_RTA.txt")	33
6.3	Power density RTA files ("_PRTA.txt")	34
6.4	Short-circuit current density file ("_Jsc.txt").....	34
6.5	Copy of the simulation input file ("_IN.txt").....	35
6.6	Ray propagation figure file ("_Rays.fig").....	35

1. INTRODUCTION

1.1 Short overview

The optical simulator CROWM (Combined Ray Optics / Wave Optics Model) has been developed for optical simulation of multi-layered optoelectronic structures comprised of multiple thick (incoherent) and thin (coherent) layers. The thick layers can be either flat or textured; in the latter case, arbitrary 1D or 2D textures with lateral and vertical features in the range of micrometres or millimetres can be applied on either side of the layers (see section 1.2 for a detailed description of the general device structure that can be simulated by CROWM).

The main input parameters of a typical CROWM simulation are the illumination conditions, the exact structure of the simulated device, and the optical properties of the materials used in the device. The main output parameters are the total reflectance and the total transmittance of the entire device, and the absorptances within each of the individual layers (see chapters 5 and 6 for a detailed description of the input and output files).

CROWM is equipped with a graphical user interface (GUI) which can be used for running the simulations, for viewing and editing the input files, for drawing the 3D schematics of the textures, and for the analysis and comparison of simulation results (see chapter 4 for a detailed description of the GUI features). Optionally, CROWM can also be run directly from the command line or even from within other programs such as MATLAB (see chapter 3).

The purpose of this manual is to inform the user about the various features and settings of CROWM and at the same time explain how to best utilise the software in order to perform accurate and reliable simulations. The user is expected to be familiar with the general topics in optics, optoelectronics and photovoltaics.

1.2 Description of the general simulated structure

The schematic of the general optoelectronic device structure that can be simulated by means of the optical simulator CROWM is presented in Figure 1.1. The device consists of:

- One or more **thick incoherent layers** with the thickness in the range of $10\ \mu\text{m} - 10\ \text{mm}$.
- Optional one-dimensional or two-dimensional **surface textures** applied to the top surface and/or the bottom surface of the thick incoherent layers. The textures at the top and bottom surface do not need to be the same. The lateral and vertical features of the textures are in the range of micrometres or larger. The textures are assumed to be **periodic in both directions** (see section 5.4) and should be continuous and single-valued everywhere (i.e. a single z value per (x,y) coordinate).
- An arbitrary number of **thin films** located in between the thick incoherent layers. The thicknesses of thin films should be in the range of $1\ \text{nm} - 1\ \mu\text{m}$ and are assumed to be much smaller than the lateral and vertical features of the textures. As a result, the thin films are assumed to be locally flat (parallel to each other and the surface of the thick layers).
- The **incident medium** (typically air) and the **medium in transmission**, which are assumed to be infinitely thick and thus **incoherent**.

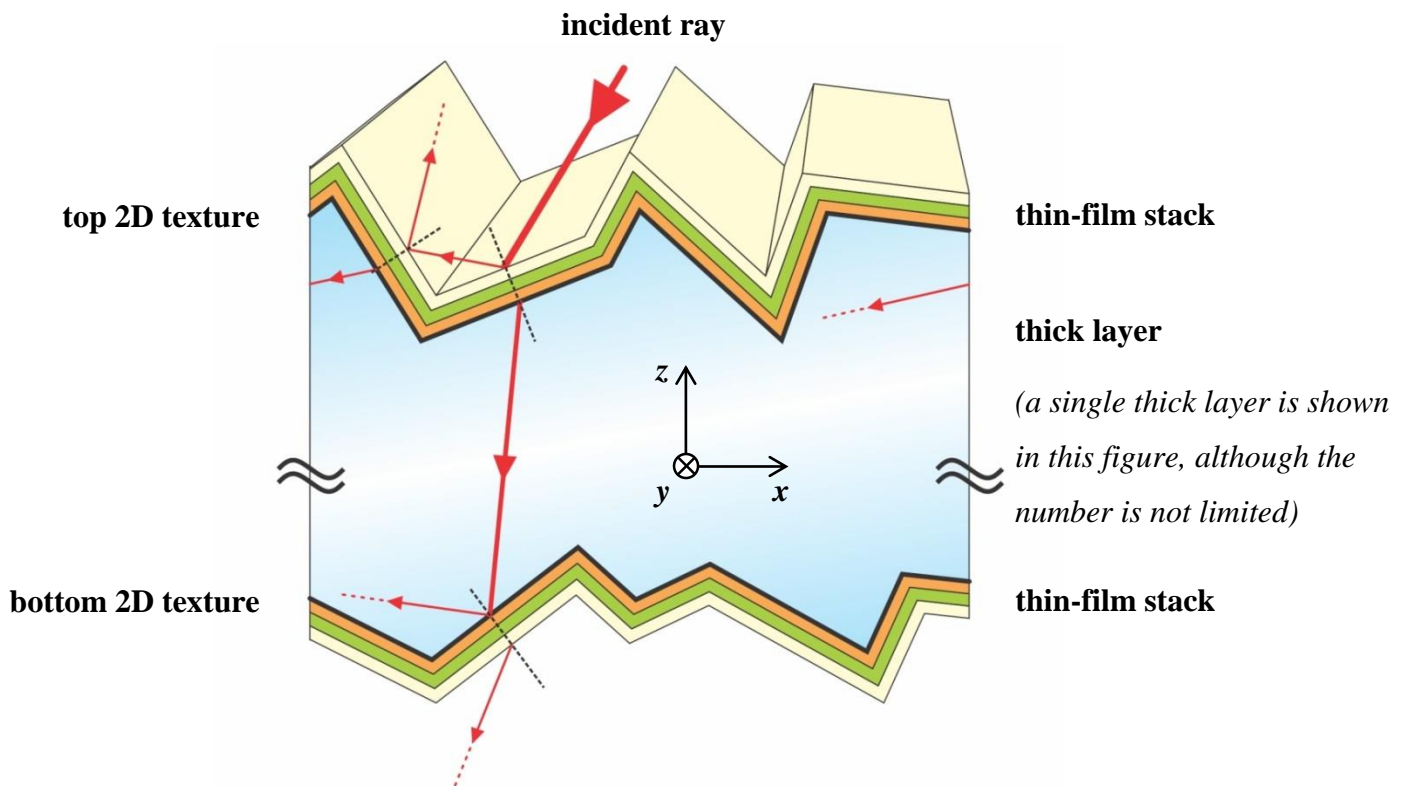


Figure 1.1: Schematic of the general device structure that can be simulated in CROWM.

1.3 Description of the optical model

Since the thicknesses of the thick layers are much larger than the effective wavelength of light, **incoherent light propagation** can be assumed within these layers. In the thin-film stacks, on the other hand, layer thicknesses are in the range of the effective wavelength of light, which results in fully **coherent light propagation**. In order to simulate this combined incoherent/coherent light propagation, CROWM combines two numerical approaches for the optical analysis of the thick and thin components of the simulated structure:

- Light propagation through the incident medium, the thick surface-textured layers, and the medium in transmission is analysed by means of **incoherent three-dimensional ray tracing** based on geometric optics. The incident illumination, which can be applied under an arbitrary incident angle (combination of the zenith and the azimuth angle), is divided into a number of rays which are then traced through the structure. The intensities and the directions of the rays are determined by the texture morphology (refraction and reflection effects at the interfaces), the optical absorption in the material (in the case of absorptive layers), and the calculated reflectances and transmittances through the thin-film components (calculated with a different model; see below). Different boundary conditions can be applied to the lateral borders of the ray tracing domain. In the case of periodic conditions, for example, a ray reaching the left border will reappear at the right border, and vice-versa, as schematically shown in Figure 1.1.
- The thin-film stacks in the simulated structure are treated by means of **fully-coherent transfer matrix formalism**, which is a widely established one-dimensional method for simulation of light propagation through a stack of thin layers, taking into account coherence effects such as destructive and constructive interference. The thin-film stacks are assumed to be parallel to each other (i.e. the textures of the thick layers are transferred to all the interfaces within the thin-film stacks, as schematically shown in Figure 1.1).

Since the total thicknesses of the thin-film stacks are assumed to be much smaller than the thicknesses and the texture features of the thick layers, the reflected and the transmitted rays through the thin-film stacks will always emerge from the same point (relative to the local surface) where the incident ray impinged upon the thin-film stack (there are no lateral shifts, as also shown in Figure 1.1). In other words, infinitely thin thin-film stacks are assumed when calculating the location of the reflected and transmitted rays in the incoherent components of the simulated structure. Further details about the model can be found in [1].

¹ B. Lipovšek et al., *Informacije MIDEM* **41**(4), 264-271 (2011).

2. INSTALLATION OF CROWM

2.1 Installation of CROWM program files

The optical simulator CROWM runs on Microsoft Windows operating systems. It is distributed as a single installation file "**CROWM_Installer.exe**" located on the installation medium. This installation package contains all the necessary program files and folders required to perform CROWM simulations. The simulator can be installed to an arbitrary folder on the hard drive. After installation, the folder structure within the main (root) program folder must not be changed in any way, although the main program folder itself (with all the sub-folders and files) can be freely copied elsewhere on the system. To uninstall CROWM from your operating system, simply delete the main program folder with all the contents.

2.2 Installation of the MATLAB Compiler Runtime (MCR)

The optical simulator CROWM was written and compiled in the MATLAB programming language. In order to run CROWM on operating systems which either don't have MATLAB installed or have a different version than the one required by CROWM, the MATLAB Compiler Runtime (MCR) must be installed in order to provide all the necessary shared functions and libraries required by the simulator.²

The MCR installation file "**MCR_Installer.exe**" is located in the "**MCR**" folder on the CROWM installation medium. Before starting to use CROWM, MCR needs to be installed to an arbitrary folder on the system. After a successful installation, reboot of the system is also advised.

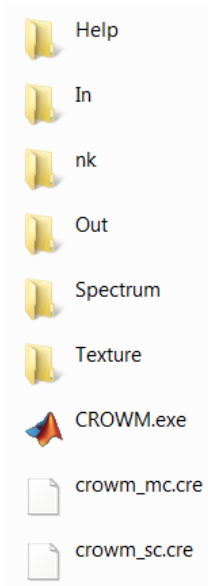
The MCR installation file as well as the detailed instructions for installing the MCR libraries are located also at the official MATLAB support page:

<https://www.mathworks.com/products/compiler/matlab-runtime.html>

² MCR release 2018a (9.4) 64-bit is required to run CROWM v3.3.1.

2.3 Files and folder structure of CROWM

After a successful installation of CROWM, the following sub-folders and files can be located in the main program folder:



- **Help** – this folder contains the User Manual (this file)
- **In** – this folder contains the simulation input files (see section 5.1)
- **nk** – this folder contains the complex refractive index files for each layer of the simulated device (see section 5.2)
- **Out** – here is where the simulation results are stored (see chapter 6)
- **Spectrum** – this folder contains the spectrum files used in the simulations (see section 5.3)
- **Texture** – this folder contains the files with the description of the textures applied to the front and/or back surface of the thick incoherent layer (see section 5.4)
- **CROWM.exe** – CROWM GUI execution file (see chapter 4).
- **crowm_mc.cre** – simulation execution file for running CROWM simulations on multiple processor cores ("mc" = multi-core)
- **crowm_sc.cre** – simulation execution file for running CROWM simulations on a single processor core ("sc" = single-core)

3. RUNNING CROWM FROM COMMAND LINE

3.1 Execution of a single simulation

When all the required input data files are present within the corresponding folders (see chapter 5 for a detailed description of the input files), a CROWM simulation can be executed:

- directly from the **command line** (e.g. Command Prompt),
- by means of a **batch file**, or
- by using the **graphical user interface** (GUI), which is the preferred option.

When running CROWM from the command line, the execution command consists of the simulation execution file (either "**crowm_mc.cre**" for multi-core or "**crowm_sc.cre**" for single-core processing) followed by one or more arguments (separated by spaces) in the following order:

- a) **name of the input file** – e.g. "Input.txt" (located in the "In" folder). If this is the final argument of the execution command, both the name of the output files and the textures used need to be specified within the input file (see section 5.1.4), or else they are set to the default values (same name as the input file name, flat interfaces).

This is an example command:

```
D:\CROWM\crowm_mc.cre Input.txt
```

- b) **name of the output files** – e.g. "Output", which is the name that all the output files will begin with. This argument overrides the name of the output files specified within the input file (see section 5.1.4). If this is the final argument of the execution command, the textures used need to be specified within the input file (see section 5.1.4), or else they are set to the default values (flat interfaces).

This is an example command:

```
D:\CROWM\crowm_mc.cre Input.txt Output
```

- c) **name of the texture files** (in order top-down through the structure) – e.g. "Texture.txt" (located in the "Texture" folder). This arguments override the textures specified within the input file (see section 5.1.4). If there are a total of N thick incoherent layers present in the structure, $N+1$ textures need to be specified (if not, flat interfaces are assumed for the missing textures). Note that in the case of flat interfaces, the texture name in the execution command can be omitted and replaced with a single character "0" (zero). No texture input files are required for specifying a flat interface (flat texture).

This is an example command (four textures are given; two of them being flat):

```
D:\CROWM\crowm_mc.cre Input.txt Output Texture_1.txt 0 Texture_3.txt 0
```

Once the simulation has been executed, the status of the simulation is displayed in the command window, and the simulation results are stored in the "Out" folder. The names of all the output files created during a simulation consist of the name specified in the command (e.g. "Output") followed by different endings (e.g. "Output_Jsc.txt" in the case of Jsc files; see chapter 6 for a detailed description of the output files).

3.2 Running a batch of simulations

Batch files are a convenient way for running a series of CROWM simulations one after the other. Batch files are simple text files with ".bat" extension which contain one or more commands that are sent directly to the command line after the execution of the file. Thus, extensive series of simulations can be prepared in advance (i.e. a list of commands in the batch file) which can then be executed at leisure (e.g. an overnight batch). Any text editor can be used to create, open, and edit batch files. The contents of an example batch file are shown in Figure 3.1.

```
echo off
echo *****
echo *                CROWM v3 - SIMULATION BATCH FILE                *
echo *****
echo An example batch file for a series of three CROWM simulations
echo of the same HIT solar cell with different textures applied.
echo on

crowm_mc.cre HIT_cell.txt HIT_cell_flat_flat 0                      0
crowm_mc.cre HIT_cell.txt HIT_cell_text_flat Pyramids.txt 0
crowm_mc.cre HIT_cell.txt HIT_cell_text_text Pyramids.txt Pyramids.txt
```

Figure 3.1: Contents of an example batch file.

3.3 Running CROWM from other programs

The optical simulator CROWM can be run easily from within other applications such as MATLAB, which makes it fairly simple to implement custom simulation procedures for automatic program execution and optimisation of the simulated structures. A typical process would consist of the following steps:

- Open the input file
- Change one or more parameters
- Save the input file
- Run simulation
- Process results
- Repeat procedure from above...

In the case of MATLAB, a CROWM simulation can be executed e.g. by means of the "**dos**" command, which sends the string in the argument directly to the command line, as shown by the following example:

```
dos('crowm_mc.cre Input.txt Output Texture_1.txt Texture_2.txt')
```


4. RUNNING CROWM USING THE GRAPHICAL USER INTERFACE (GUI)

4.1 Main simulation panel

The main simulation panel, which is used to setup and execute CROWM simulations, is displayed when running the "CROWM.exe" file in the CROWM installation folder. The panel is divided into several sections, each of which contains a number of controls that can be operated as described in the following. The screenshot of the main simulation panel is shown in Figure 4.1.

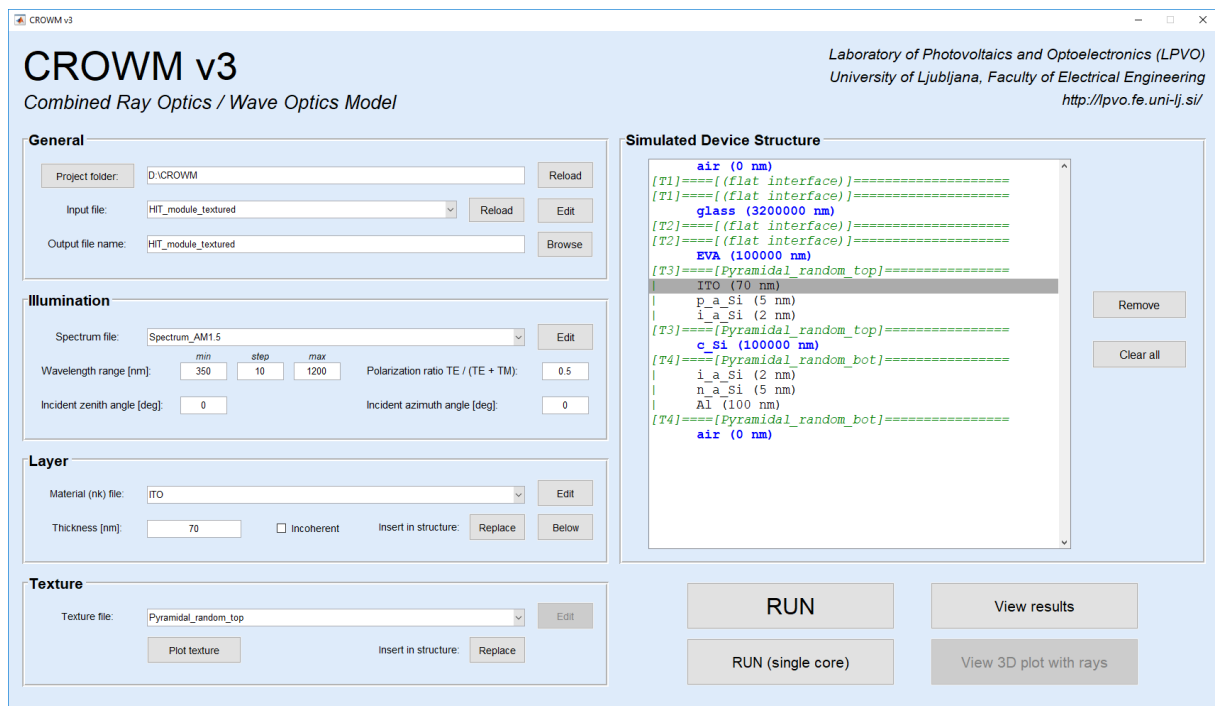


Figure 4.1: Main simulation panel. Displayed are the settings for the simulation of an example encapsulated heterojunction silicon solar cell (module).

4.1.1 General

4.1.1.1 Project folder (*button & text box*)

Select the current project folder. The project folder must contain all the folders required for running a CROWM simulation, as described in section 2.3. The user can either browse for the folder by clicking on the button, or enter the folder location directly into the text box. By default, the project folder is set to the CROWM installation folder.

4.1.1.2 Reload (*button*)

Reload all the input, spectrum, nk, and texture files located in the "In", "Spectrum", "nk", and "Texture" folders of the project folder and list them into the corresponding menus (see below). This is useful if any changes are made to the input folders (e.g. creating a new input file), which would otherwise require a restart of the GUI.

4.1.1.3 Input file (*menu*)

Select the simulation input file from the menu. The menu lists all the input files that are located in the "In" folder of the project folder.

4.1.1.4 Reload (*button*)

Reload all the parameters from the selected input file. This can be used if the user wants to discard any changes and revert to the initial state of the input file. In certain cases, such as opening the input file in an external editor, the button turns yellow. This reminds the user that some parameters may have been changed in the input file externally and therefore the current state of the GUI may not be up-to-date.

4.1.1.5 Edit / Save (*button*)

Edit the selected input file by opening the input file in the default external editor (typically Notepad). This can be used for setting up all the possible parameters and settings in the input file (see section 5.1 for a detailed explanation of the input file). However, if any changes have already been made to the parameters within the GUI, the button turns to "Save" instead. In this case, it can be used to save the present state of the input file.

4.1.1.6 Output file name (*text box*)

Enter the base name of the output files which will be created during the simulation. Up to seven output files are normally created during each simulation, all of which have the same base name and different endings (see chapter 6 for a detailed description of the output files).

4.1.1.7 Browse (*button*)

Select the base name of the output files by browsing through the previously created output files. This is useful if trying to access the results of a previous simulation.

4.1.2 Illumination

4.1.2.1 Spectrum file (*menu*)

Select the spectrum file (see section 5.3) which will be used to calculate the short-circuit current densities (J_{sc}) from the reflectance, transmittance, and absorptance (RTA) results of the simulation (see section 6.4 for more details about J_{sc} results). The spectrum file must be located in the "Spectrum" folder of the current project folder.

4.1.2.2 Edit (*button*)

Edit the selected spectrum file by opening the spectrum file in the default external editor (typically Notepad).

4.1.2.3 Wavelength range (*text box*)

Enter the wavelength range of the simulation: the minimum wavelength, the maximum wavelength, and the wavelength step (all in nanometres). Alternatively, if any of the above three parameters is set to 0 (zero), the wavelength range is taken from the specified spectrum file; the simulation is performed at all wavelengths contained in the spectrum file.

4.1.2.4 Polarization ratio (*text box*)

Enter the power ratio between the TE-polarized and the TM-polarized incident illumination. In the case of purely TE-polarized illumination, this setting should be set to 1. In the case of purely TM-polarized illumination, this setting should be set to 0. Global polarization in CROWM is defined with respect to the horizontal xy plane – the electric field of the incident TE-polarization is always parallel to the xy plane (the orientation of the coordinate system as assumed in CROWM is also shown in Figure 1.1 and Figure 5.3).

4.1.2.5 Incident zenith angle (*text box*)

Enter the zenith angle (in degrees) of the incident illumination. Zenith angle values in the range of $\pm 90^\circ$ are allowed. In the case of perpendicular top-down illumination (relative to the horizontal xy plane), the zenith angle should be set to 0° . See also Figure 5.3 for definition of incident angles with respect to the global coordinate system.

4.1.2.6 Incident azimuth angle (*text box*)

Enter the azimuth angle (in degrees) of the incident illumination. Azimuth angle values in the range of $\pm 360^\circ$ are allowed. The origin of the azimuth angle (0°) is along the x axis. See also Figure 5.3 for definition of incident angles with respect to the global coordinate system.

4.1.3 Layer

4.1.3.1 Material (nk) file (*menu*)

Select the refractive index file (see section 5.2) which will be used for the layer. Refractive index files are located in the "nk" folder of the project folder.

4.1.3.2 Edit (*button*)

Edit the selected refractive index file by opening the refractive index file in the default external editor (typically Notepad).

4.1.3.3 Thickness (*text box*)

Enter the thickness of the layer (in nanometres). Note that infinite thicknesses are assumed for the incident (top) medium and the medium in transmission (bottom), although the printed values for these two media are always 0.

4.1.3.4 Incoherent (*check box*)

Check the box if the layer should be treated as incoherent (analysis based on ray tracing).

4.1.3.5 Replace (*button*)

Replace the selected (highlighted) layer in the device structure with the current layer (combination of the refractive index file, the layer thickness, and the incoherence setting).

4.1.3.6 Below (*button*)

Insert the current layer (combination of the refractive index file, the layer thickness, and the incoherence setting) below the selected (highlighted) layer in the device structure.

4.1.4 Texture

4.1.4.1 Texture file (*menu*)

Select the texture file (see section 5.4) from the menu which will be used in the simulation to describe the texture between two incoherent layers. The menu lists all the texture files that are located in the "Texture" folder of the project folder. A perfectly flat interface can be selected at the top of the menu.

4.1.4.2 Edit (*button*)

Edit the selected texture file by opening the texture file in the default external editor (typically Notepad).

4.1.4.3 Plot texture (*button*)

Plot the selected texture in a rendered 3D figure. The texture viewer is opened after clicking the button (see section 4.2).

4.1.4.4 Replace (*button*)

Replace the selected (highlighted) texture in the device structure with the current texture.

4.1.5 Simulated Device Structure

This panel lists the complete structure of the simulated device. Each item in the list can be selected and thus manipulated by other controls described above. The list is colour coded so that:

- **the thick blue text is used for the incoherent layers,**
- the black text is used for the coherent layers,
- *the italic green text is used for the textures between the incoherent layers.*

4.1.5.1 Remove (*button*)

Remove the selected layer or texture from the structure.

4.1.5.2 Clear all (*button*)

Clear the entire structure and revert to the default structure consisting of a single flat incoherent glass layer in air.

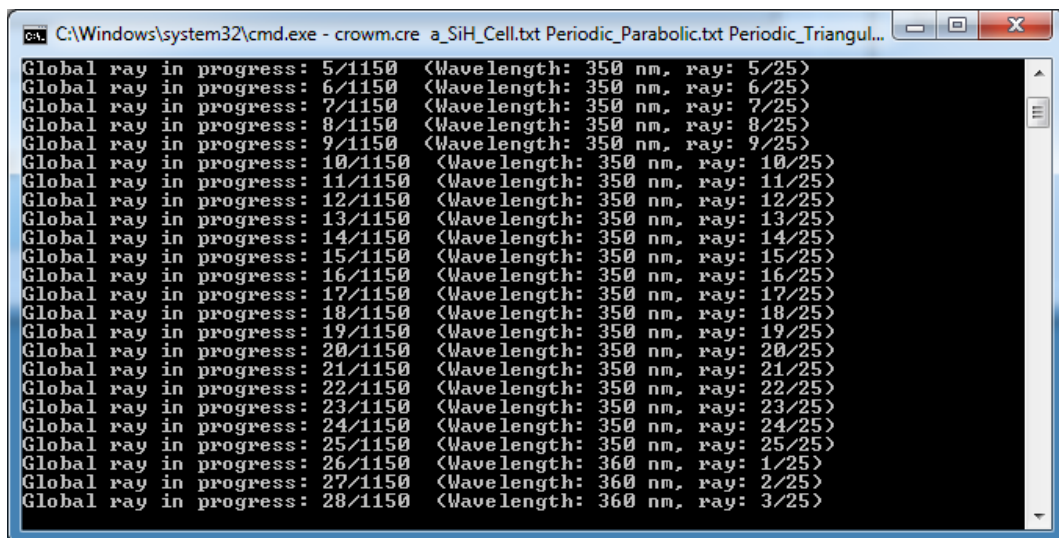
4.1.6 RUN (button)

Execute a single CROWM simulation, using multiple processor cores. After executing the simulation, a Command Window is opened displaying the initialisation procedures, including the distribution of the load between the processor cores, followed by the current status of the simulation. The current status of the simulation is given by the index of the current ray in relation to the total number of rays that need to be simulated. After the simulation is completed, the window is closed.

(Please note that the current status of the simulation in the case of multi-core simulation is only useful for confirming that the simulation is running. It does not, however, reveal any information about the work done so far, since the order of the incident rays that are simulated is chosen randomly each time.)

4.1.7 Run (single core) (button)

Execute a single CROWM simulation, using a single processor core. After executing the simulation, a Command Window is opened displaying the initialisation procedures, followed by the current status of the simulation. The current status of the simulation is given by the index of the current ray in relation to the total number of rays that need to be simulated, as shown in Figure 4.2. After the simulation is completed, the window is closed.



```
C:\Windows\system32\cmd.exe - crows.cre a_SiH_Cell.txt Periodic_Parabolic.txt Periodic_Triangul...
Global ray in progress: 5/1150 <Wave length: 350 nm, ray: 5/25>
Global ray in progress: 6/1150 <Wave length: 350 nm, ray: 6/25>
Global ray in progress: 7/1150 <Wave length: 350 nm, ray: 7/25>
Global ray in progress: 8/1150 <Wave length: 350 nm, ray: 8/25>
Global ray in progress: 9/1150 <Wave length: 350 nm, ray: 9/25>
Global ray in progress: 10/1150 <Wave length: 350 nm, ray: 10/25>
Global ray in progress: 11/1150 <Wave length: 350 nm, ray: 11/25>
Global ray in progress: 12/1150 <Wave length: 350 nm, ray: 12/25>
Global ray in progress: 13/1150 <Wave length: 350 nm, ray: 13/25>
Global ray in progress: 14/1150 <Wave length: 350 nm, ray: 14/25>
Global ray in progress: 15/1150 <Wave length: 350 nm, ray: 15/25>
Global ray in progress: 16/1150 <Wave length: 350 nm, ray: 16/25>
Global ray in progress: 17/1150 <Wave length: 350 nm, ray: 17/25>
Global ray in progress: 18/1150 <Wave length: 350 nm, ray: 18/25>
Global ray in progress: 19/1150 <Wave length: 350 nm, ray: 19/25>
Global ray in progress: 20/1150 <Wave length: 350 nm, ray: 20/25>
Global ray in progress: 21/1150 <Wave length: 350 nm, ray: 21/25>
Global ray in progress: 22/1150 <Wave length: 350 nm, ray: 22/25>
Global ray in progress: 23/1150 <Wave length: 350 nm, ray: 23/25>
Global ray in progress: 24/1150 <Wave length: 350 nm, ray: 24/25>
Global ray in progress: 25/1150 <Wave length: 350 nm, ray: 25/25>
Global ray in progress: 26/1150 <Wave length: 360 nm, ray: 1/25>
Global ray in progress: 27/1150 <Wave length: 360 nm, ray: 2/25>
Global ray in progress: 28/1150 <Wave length: 360 nm, ray: 3/25>
```

Figure 4.2: Command Window during a simulation running on a single core.

4.1.8 View results (*button*)

Open the results viewer (see section 4.4). By default, the results of the simulation that is specified by the output name given in the output file name text box (see section 4.1.1.6) are loaded into the simulation results list of the results viewer (see section 4.4).

4.1.9 View 3D plot with rays (*button*)

Plot the ray propagation schematic in a rendered 3D figure (see section 6.6). The ray propagation viewer is opened after clicking the button (see section 4.3). The schematic corresponds to the results of the simulation specified by the output name given in the output file name text box (see section 4.1.1.6). To access and display other available ray propagation figure files located in the “Out” folder, click the “Browse” button next to the output file name text box and select the chosen simulation results (see section 4.1.1.7).

The button is greyed out if no ray propagation figure file with the specified output name is found in the "Out" folder (e.g. if generation of the ray propagation figure file has been disabled by the “Period of plotted rays” setting; see section 5.1.1), or if the GUI has not yet been refreshed with the most recent data. For this purpose, please click the “Reload” button (see section 4.1.1.2) after completing the simulation; this way, the GUI will be updated with the most recent data.

4.2 Texture viewer

Texture viewer is used for graphical representation of the selected texture. If the texture to be viewed contains a large number of data points (i.e. a large matrix in the texture file), the texture can be interpolated to a maximum of 200 x 200 points before being plotted in the texture viewer. An example parabolic texture displayed in the texture viewer is shown in Figure 4.3.

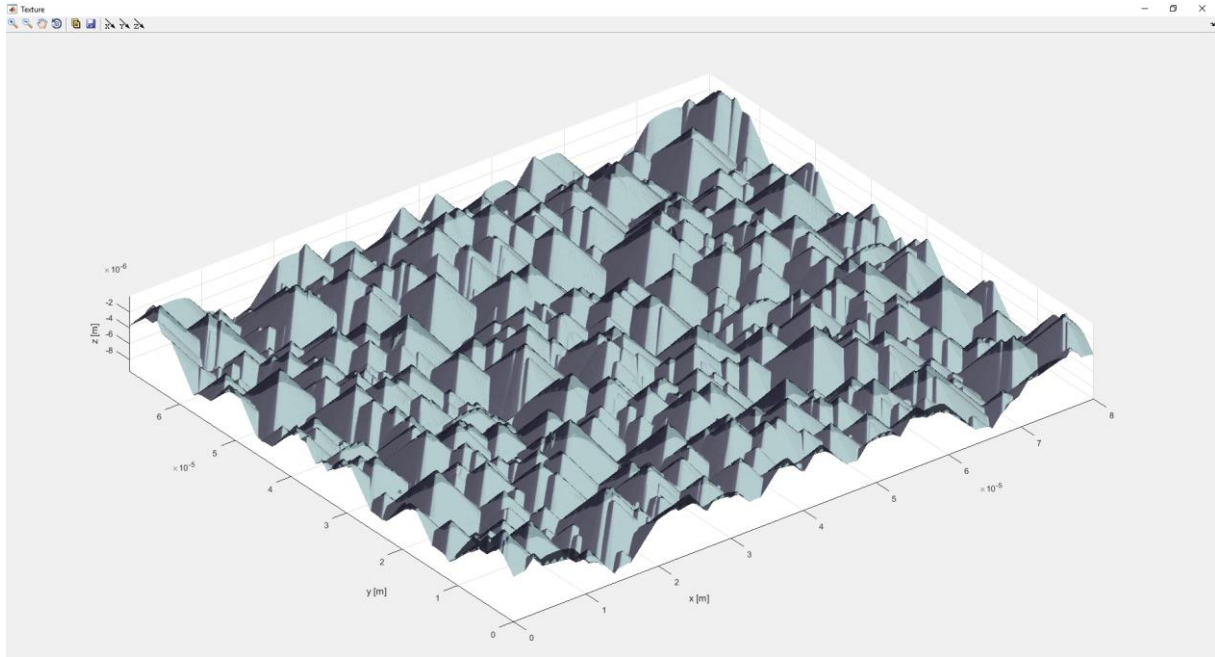


Figure 4.3: An example random pyramidal texture displayed in the texture viewer.

The following controls can be accessed from the toolbar in the top-left corner of the texture viewer (in order from left to right):

- **Zoom in**
- **Zoom out**
- **Pan**
- **Rotate**
- **Open texture file** (automatically opens the texture file in the default text editor)
- **Save figure** (save the figure as a bitmap file)
- **Set the point of view along the x axis** (view the texture in the yz plane)
- **Set the point of view along the y axis** (view the texture in the xz plane)
- **Set the point of view along the z axis** (view the texture in the xy plane)

4.3 Ray propagation viewer

Ray propagation viewer (shown in figure Figure 4.4) is used for graphical representation of ray propagation through the device structure. The viewer opens the ray propagation figure file which was created during the simulation (see section 6.6) and displays it in the ray propagation viewer window. Creation of the ray propagation figure file is controlled by the "Period of plotted rays" setting in the input file (see section 5.1.1).

Please note that full propagation of rays through thick incoherent layers, as shown in figure Figure 4.4 on the left, is displayed only if the "Trace entire thicknesses of layers" option is enabled in the input file. Otherwise, only propagation in the vicinity of the textures is displayed, as shown in figure Figure 4.4 on the right. While the simulation results are identical in both cases, the simulation takes much longer if tracing through the entire thickness of incoherent layers is enabled, and so this option should be used solely for the purpose of plotting the ray propagation schematics. For further details, please consult section 5.1.2.

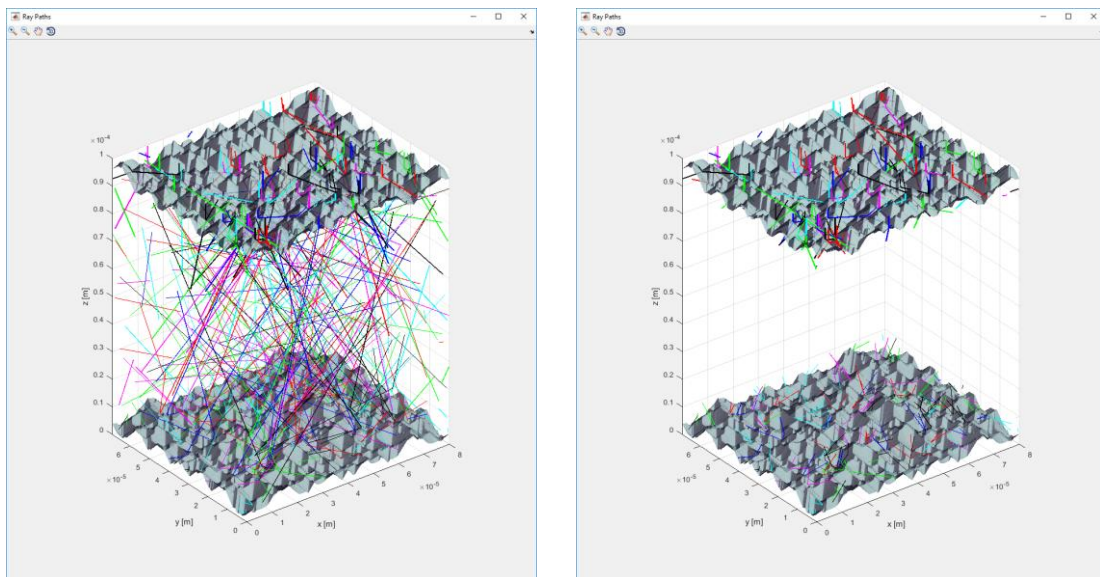


Figure 4.4: Ray propagation through a textured incoherent layer.

The following controls can be accessed from the toolbar in the top-left corner of the ray propagation viewer (in order from left to right):

- **Zoom in**
- **Zoom out**
- **Pan**
- **Rotate**

4.4 Results viewer

Results viewer (shown in Figure 4.5) is used for viewing the results of CROWM simulations.

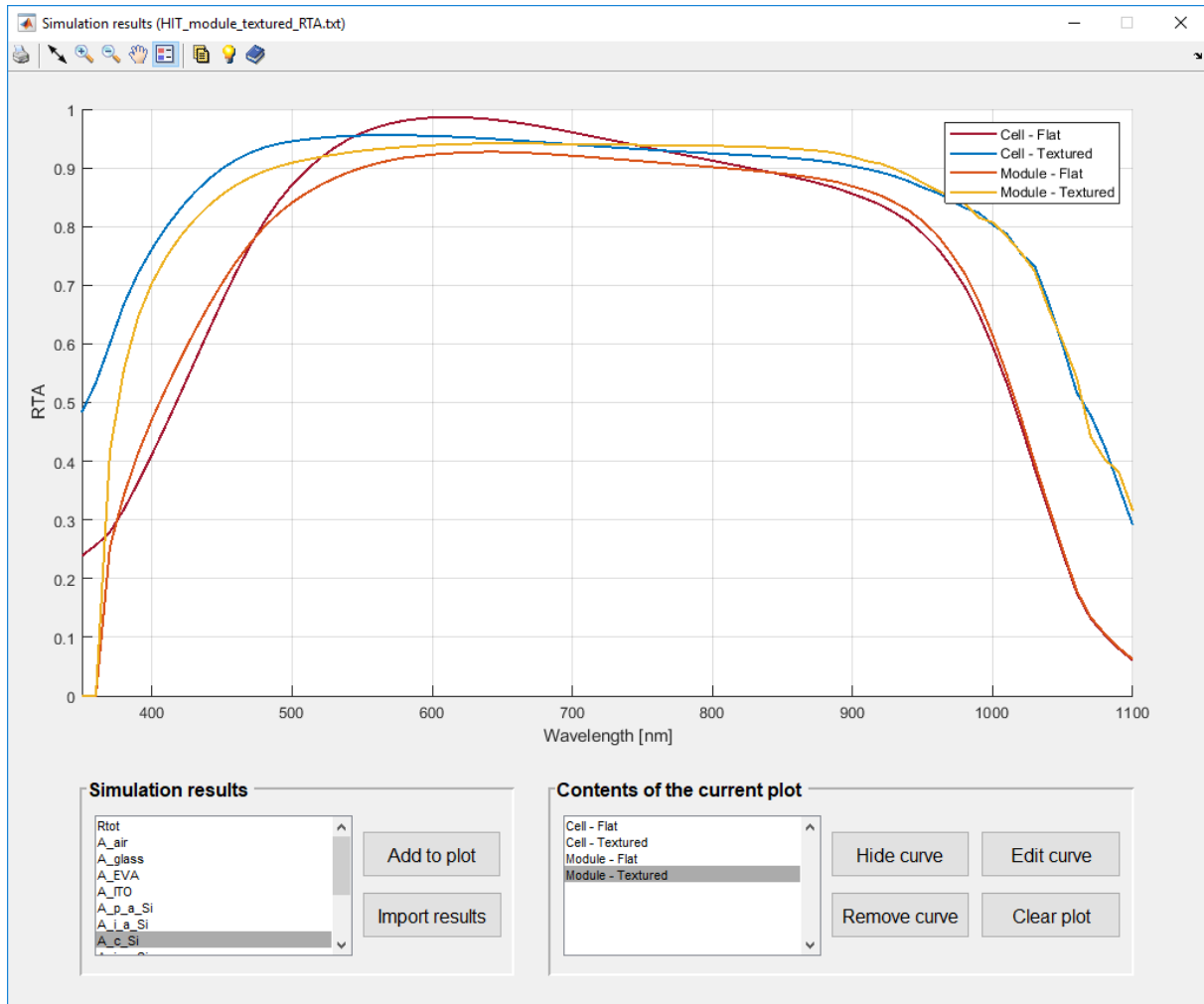


Figure 4.5: Results viewer.

The basic controls can be accessed from the toolbar in the top-left corner:

- **Print** (prints the contents of the results viewer window)
- **Set plot boundaries** (used for editing the range of both plot axes)
- **Zoom in**
- **Zoom out**
- **Pan**
- **Legend** (toggles display of the legend)
- **Open RTA results** (automatically opens the "RTA" file in the default text editor)
- **Open Jsc results** (automatically opens the "Jsc" file in the default text editor)
- **Open log file** (automatically opens the "log" file in the default text editor)

The bottom part of the results viewer is divided into two sections. The first section is titled "*Simulation results*" and lists the reflectance, transmittance, and absorptance (RTA) results of the current simulation, whereas the second section titled "*Contents of the current plot*" lists the selected RTA results which are displayed in the current plot. The following controls can be accessed:

4.4.1 Add to plot (*button*)

Add the selected (highlighted) RTA simulation result to the current plot.

4.4.2 Import results (*button*)

Open RTA results from a previous simulation (see section 6.2).

4.4.3 Hide curve (*button*)

Hide the selected (highlighted) curve in the current plot.

4.4.4 Edit curve (*button*)

Edit the properties of the selected (highlighted) curve in the current plot. Curve colour and text displayed in the plot legend can be changed as desired.

4.4.5 Remove curve (*button*)

Remove the selected (highlighted) curve from the current plot.

4.4.6 Clear plot (*button*)

Clear all curves from the current plot.

5. DESCRIPTION OF THE INPUT FILES

The input files required for a CROWM simulation are located in four input folders: "In", "nk", "Spectrum", and "Texture". In the following, each of the input files is explained in detail. Please note that additional files of the same type can be created freely as long as the general structure of the files (as explained in the following sections) remains unchanged.

5.1 Simulation input files (".txt", located in "In" folder)

The input files are located in the "In" folder of the project folder. An input file is a plain text file (".txt" extension) which lists all the important simulation parameters, such as the structure definition, the wavelength range etc. The name of the input file can be chosen freely as long as there are no spaces and no non-standard characters present in the text. For example, "Sim_0deg_10percent.txt" is a valid input file name, whereas "Sim 0deg 10%.txt" is not. The first few lines of an example input file are shown in Figure 5.1.

```
#*****#
#                               CROWM v3.3 SIMULATION - INPUT FILE                               #
#*****#
# Optional header (arbitrary number of lines): notes, descriptions, etc.

# GENERAL SIMULATION PARAMETERS:
#-----
Illumination spectrum file:           Spectrum_AM1.5.spc
Illumination side (front/rear):       0
Illumination type (direct/diffuse):   0
Incident zenith angle [deg]:          0
Incident azimuth angle [deg]:         0
Incident TE/(TE+TM) ratio:            0.5
Scale spectrum by the zenith angle:   0
Wavelength start [nm]:                350
...                                   ...
```

Figure 5.1: The first few lines of an example input file.

IMPORTANT: The structure of the input files must not be changed in any way! All the spaces, titles, and empty lines must remain at their original position.

In the following, the function of each of the settings in the input files (each of the lines in each of the sections) will be explained in detail.

5.1.1 General simulation parameters

- Illumination spectrum file

The name of the spectrum file (see section 5.3) which will be used to calculate the short-circuit current densities (J_{sc}) from the reflectance, transmittance, and absorptance (RTA) results of the simulation (see section 6.4 for more details about J_{sc} results). The spectrum file must be located in the "Spectrum" folder of the current project folder.

- Illumination side

If set to **0**, the incident illumination is located at the front (top) side of the simulated device. If set to **1**, the incident illumination is located at the back (bottom) side of the simulated device. This can be used to easily simulate the device from both sides. Note, however, that this setting does not alter any of the incident angles (see more details below in the description of the incident zenith angle).

- Illumination type

If illumination type is set to **0**, direct incident illumination is applied, in which all the power of the incident rays is assumed to be propagating along the single trajectory, as defined by the incident zenith and the incident azimuth angles. The normalized angular intensity distribution (AID) of such illumination, shown as a function of the zenith angle and integrated over all azimuth angles, is presented by the red curve in Figure 5.2 (left).

If illumination type is set to **1**, diffuse incident illumination is applied, in which all the power is assumed to be distributed according to the Lambertian angular intensity distribution, with the main direction of propagation still defined by the incident zenith and the incident azimuth angles. The normalized AID for this case, shown as a function of the zenith angle and integrated over all azimuth angles, is presented by the red curve in Figure 5.2 (right). Additionally, the normalized AID in a single plane (as if measured using e.g. an angular resolved scattering system) is also presented in the figure by the blue curve.

Under diffuse illumination conditions, the trajectory of each generated "incident ray" is chosen randomly with the probability function defined by the Lambertian angular intensity distribution. Consequently, a much larger total number of incident rays is required when running simulations under such conditions in order to obtain statistically meaningful results.

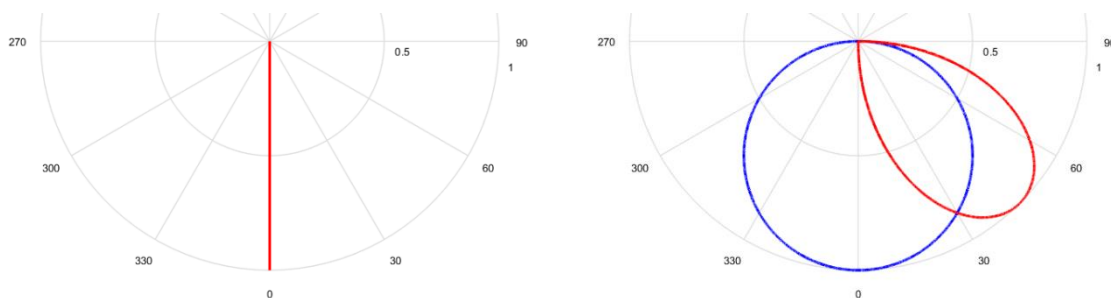


Figure 5.2: Normalized AID of direct and diffuse (Lambertian) incident illumination.

- Incident zenith angle [deg]

The zenith angle (in degrees) of the incident illumination. Zenith angle values in the range of ± 180 are allowed. In the case of perpendicular top-down illumination (relative to the horizontal xy plane), the zenith angle should be set to **0**. For perpendicular bottom-up illumination, the zenith angle should be set to **180**. See Figure 5.3 for definition of incident angles with respect to the global coordinate system.

- Incident azimuth angle [deg]

The azimuth angle (in degrees) of the incident illumination. Azimuth angle values in the range of ± 360 are allowed. The origin of the azimuth angle (0°) is along the x axis. See Figure 5.3 for definition of incident angles with respect to the global coordinate system.

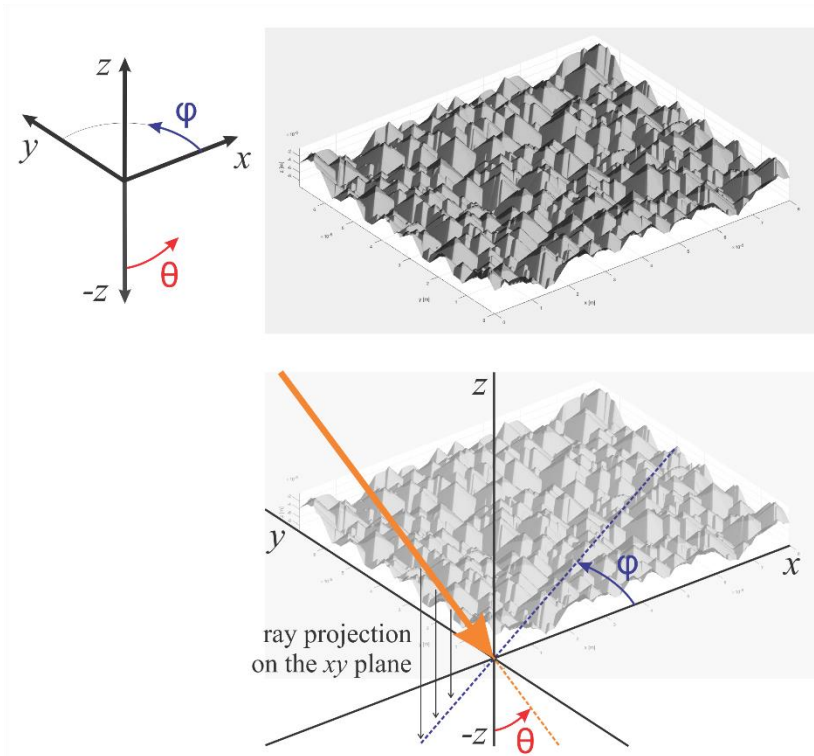


Figure 5.3: Definition of incident angles with respect to the global coordinate system.

- Incident TE/(TE+TM) ratio

The power ratio between the TE-polarized and TM-polarized incident illumination. In the case of purely TE-polarized illumination, this setting should be set to **1**. In the case of purely TM-polarized illumination, this setting should be set to **0**. Global polarization in CROWM is defined with respect to the horizontal xy plane – the electric field of the incident TE-polarization is always parallel to the xy plane (the orientation of the coordinate system as assumed in CROWM is also shown in Figure 1.1 and Figure 5.3).

- Scale spectrum by the zenith angle

If set to **0**, the power density of the incident illumination, given in the spectrum file (see section 5.3) will remain the same even if the device is illuminated under some non-zero zenith angle. If set to **1**, the power density of the incident illumination will be scaled by the cosine of the zenith angle, which is the case in realistic applications. Note that this setting only affects the calculation of J_{sc} values (see section 6.4 for more details about J_{sc} results).

- Wavelength start [nm]

The minimum wavelength of the simulation wavelength range (in nanometres).

- Wavelength step [nm]

The wavelength step of the simulation wavelength range (in nanometres).

- Wavelength stop [nm]

The maximum wavelength of the simulation wavelength range (in nanometres).

Note that if any of the above three parameters is set to **0**, the wavelength range is taken from the specified spectrum file; the simulation is performed at all wavelengths contained in the spectrum file.

- Period of plotted rays

If set to **0**, generation of ray propagation figure file (see section 6.6) is disabled. Any other number enables generation of ray propagation figure file (for the final wavelength in the simulation range), while the number itself determines how many rays will be stored. If the period is set to 1, each ray will be stored; if it is set to 2, every second ray along each of the x and y axes will be stored, etc. Please note that this setting does not alter the number of incident rays which are simulated, but merely controls the size and density of the ray propagation figure file which can be created during the simulation. If graphical representation of ray propagation is not required (see section 4.1.9), this option should be disabled in order to save disk space.

It should also be noted that in the case when forced domain periods are applied (see the “Forced domain period” settings in section 5.1.5), the actual texture profiles are not plotted in the generated ray propagation figure file (since that would typically result in an unreasonable amount of data to be displayed). Instead, all interfaces are plotted as flat, although the rays themselves are still plotted realistically as if the textures were there (and as was also the case in the simulation).

- Number of reserved computer cores

When running CROWM simulations using parallelization, all computer cores available can be used. This can slow down the computer significantly. Using this option, the user can specify how many of the computer cores should be kept free. For example, if set to **0**, all cores will be used. If set to **1**, one core will be kept free. If set to **2**, two cores will be kept free etc.

5.1.2 Ray tracing parameters

- Ray tracing precision (P min/P in)

When a ray with the initial (incident) power P_0 is propagating through the device, its power P is diminished by numerous refractions, reflections, and absorption encountered on its path. If the power compared to the incident power drops below a certain limit specified by the precision level (e.g. $P/P_0 < 0.0001$ or $1e-4$), the ray is ignored and the simulator proceeds to the next ray in the analysis. The power of the incident ray P_0 is defined as the total power given at the current wavelength in the spectrum file (see section 5.3) divided by the total number of incident rays: $P_0 = P_{spc}(\lambda)/N_{inc.rays}$.

An increased level of precision (lower power limit) leads to more accurate results, however, it also leads to dramatically longer simulation times! A careful balance thus must be achieved by selecting just the right precision level which will still result in accurate results but won't bog down the simulation to unreasonable durations. The setting between **0.001** (or $1e-3$) and **0.0001** (or $1e-4$) will generally suffice in most cases.

- Ray tracing resolution (step) [nm]

During a CROWM simulation, finite ray tracing steps are employed to incrementally trace the propagation of rays along their trajectories. A finer (lower) ray tracing resolution generally leads to more accurate results, however, it also leads to longer simulation times! Again, the most important factor in choosing the right ray tracing resolution is the nature of the textures – smaller and more dynamic surface features generally require shorter ray tracing steps (finer ray tracing resolution). For micrometer-size textures, the setting of **10 nm** can typically be used.

- Import resolution from texture files

If set to **0**, ray tracing resolution as defined in the input file will be used (see above). If set to **1**, ray tracing resolution will be imported separately for each texture from the texture file (see section 5.4). This is useful since textures with smaller features generally require a finer ray tracing resolution compared to textures with larger features. By enabling this option, ray propagation near a specific texture can use ray tracing resolution best suited for this particular texture.

- Number of incident rays (X axis)

In a typical CROWM simulation, the power of the incident illumination is evenly distributed among the incident rays which are evenly spaced along the x and/or y axis. This setting controls the number of rays to be generated along the x axis.

- Number of incident rays (Y axis)

This setting controls the number of rays to be generated along the y axis.

Note that the numbers of rays generated along each of the axes do not need to be equal. Indeed, when performing CROWM simulations of device structures which only include 1D textures (i.e. the texture morphology changes only in one dimension), the user is advised to generate rays only along the dimension where the texture morphology changes, while keeping the number of rays along the other dimension at 1 (this in effect turns 3D simulations to 2D).

The total number of rays that need to be simulated is one of the critical parameters which significantly affect the speed of the simulation! Choosing the right number of rays is crucial for obtaining the most information in the least amount of time. For this decision, the textures of the thick incoherent layer present the most important factor. While indeed a single ray is enough to simulate a perfectly flat structure, the number of rays generally needs to be increased when the texture becomes more dynamic.

- Ray matrix shift factor (X axis)

In a typical CROWM simulation, the incident rays are generated so that the origin of the first ray is located in the (0,0) point of the xy plane. By using this option, the rays can be shifted along the x axis by the specified amount (**0** – **1**). This value presents the relative shift compared to the full distance between two adjacent rays along the x axis.

- Ray matrix shift factor (Y axis)

By using this option, the rays can be shifted along the y axis by the specified amount (**0** – **1**). This value presents the relative shift compared to the full distance between two adjacent rays along the y axis.

- Trace entire thicknesses of layers

By default, in order to speed up simulations, incremental ray tracing between the lowest point of the top texture (minimum z value) and the highest point of the bottom texture (maximum z value) of an incoherent layer is not performed since the layer is assumed to be perfectly uniform between these two points. Enabling this option by setting it to **1** overrides the default simulation method and forces ray tracing to be performed throughout the thick incoherent layers. Note that this setting results in significantly longer simulation times and should only be used when generation of full ray propagation figure file is required (see section 6.6). If set to **0**, this option is disabled.

- Max number of total R/T events

By means of this option, the user can limit the number of times that a single ray experiences reflection or transmission events at the interfaces. If the specified maximum number of reflection/transmission events is reached, the ray is assumed to be absorbed in the current medium and the simulator proceeds to the next ray in the analysis. If set to **0**, this option is disabled.

- Max number of horizontal passes

By means of this option, the user can limit the number of times that a single ray passes horizontally inside an incoherent layer from one lateral border to the other, before reaching the top or the bottom texture. If the specified maximum number of horizontal passes is reached, the ray is assumed to be absorbed in the thick incoherent layer and the simulator proceeds to the next ray in the analysis. This setting can also be used as a safety option to prevent a ray becoming infinitely trapped when propagating horizontally within a non-absorptive incoherent layer (i.e. its power is never reduced below the limit given by the precision level; see above). If set to **0**, this option is disabled.

- Max number of internal reflections

By means of this option, the user can limit the number of times that a single ray reflects from the textures. If the specified maximum number of reflections is reached, the ray is assumed to be absorbed in the vicinity of the texture and the simulator proceeds to the next ray in the analysis. Note that in the case of a perfectly flat structure, the number of reflections is equal to the number of vertical passes. In the case of textured structures, however, the numbers are no longer the same (the ray may experience multiple reflections from a texture before being reflected to the other side of the thick incoherent layer). If set to **0**, this option is disabled.

5.1.3 Randomization parameters

- Random number generator seed

When using random events in CROWM, random number generation (RNG) functions need to be called. If this option is set to **0**, the initial value (seed) of the RNG functions is determined automatically at the beginning of each simulation based on the system clock. If set to any other number, this number now defines the seed. Running different simulations with the same seed will result in the same random numbers being generated and, therefore, the same results being obtained.

- Random position of incident rays

If set to **1**, this option enables random positioning of incident rays, which is an alternative to evenly spaced (grid) positioning along the x and y axis. The total number of incident rays generated per each wavelength is still controlled by the product of the two values specified in the "Number of incident rays" settings, however the point of origin of each ray for each wavelength is now determined completely randomly.

As an additional way to control incident ray generation, this option can also be set to **2**. In this special case, the origins of the incident rays are determined randomly for the first wavelength in the simulation. Afterwards, the same origins are used also for every other wavelength in the simulation range. This setting helps to reduce the noise present in the

spectral results due to different positioning of rays each wavelength. It is important to note, however, that the accuracy of the results is not improved this way; the option merely hides the uncertainty related to the limited number of the incident rays.

If set to **0**, this option is disabled, and the incident rays are generated in a fixed evenly spaced grid, as described previously.

- Random treatment of R/T events

Normally, as a ray hits an interface between two layers, both the reflected ray and the transmitted ray are generated. This can result in a large number of new rays being generated constantly, which leads to longer simulation times. If this option is set to **1**, only a single ray is generated during each such event – either the reflected ray or the transmitted ray. The probability of generation is defined by the power of each of the two rays. This option helps to speed up simulations significantly, however a sufficient number of incident rays are required to obtain statistically meaningful results. If set to **0**, this option is disabled.

- Random treatment of A events

Normally, when a ray crosses a thick incoherent layer, some of its initial power is absorbed by the layer, while the remaining is incident on the layer surface, at which point it is distributed into the reflected and/or the transmitted ray. If this option is set to **1**, the initial power of the ray is either completely absorbed in the layer, or completely transmitted across the layer. The probability of this event is defined by the ratio of the absorbed to the remaining power after crossing the layer. This option helps to speed up simulations, however a sufficient number of incident rays are required to obtain statistically meaningful results. If set to **0**, this option is disabled.

5.1.4 Texture and output file names

- Base name of the output files

The base name of the output files which will be created during the simulation. Up to seven output files are created during each simulation, all of which have the same base name and different endings (see chapter 6 for a detailed description of the output files).

- List of texture files (top-down)

Following this line is the list of texture files to be used, in order, for each interface between the layers. If there are a total of N thick incoherent layers present in the structure, $N+1$ textures must be specified. If some interface is assumed to be flat, a single **0** (zero) can be specified instead of the texture file.

5.1.5 Structure definition

- Lateral domain periodicity

This setting controls the boundary conditions that are applied to the lateral borders of the simulation domain. The simulation domain is defined either by the full period of the texture, or by the “Forced domain period” settings (see further below for details).

If set to **0**, a finite-size (bounded) device is assumed. In this case, the rays are extinguished when they reach the lateral boundaries of the simulation domain (their power is assumed to be absorbed within the incoherent layer).

If set to **1**, periodic boundary conditions are applied. In this case, as a ray reaches one of the lateral borders of the simulation domain, it reappears at the opposite border.

If set to **2**, symmetric periodicity is assumed. In this case, the textures specified in the simulation should represent only one half of the entire period (in both lateral dimensions). Before running the simulation, the texture is then “double-mirrored” internally (by the simulator) in order to obtain the complete periodic structure, and the period of this double-mirrored structure is now assumed to be the simulation domain. Note, however, that only the initial one half (or better one quarter) of the simulation domain is illuminated, which enables roughly four times shorter simulation times at no expense at accuracy.

Symmetric periodicity conditions (the option set to **2**) should typically be used also in the case of non-periodic textures, such as the random pyramidal texture of an etched crystalline silicon wafer. In this case, a part of the texture (e.g. an AFM scan) is taken as the initial texture, and then this texture is double-mirrored internally in order to achieve a quasi-periodicity required by the simulation of a laterally unlimited device. The process of double-mirroring is represented in Figure 5.4.

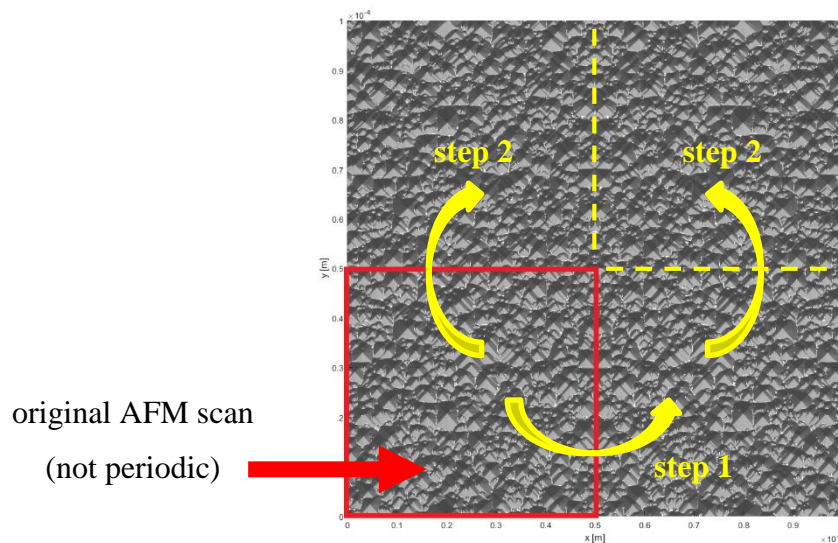


Figure 5.4: Example of “double-mirroring” of a random pyramidal texture.

- Forced domain period (X axis) [um]

If set to the default value of **0**, the period of the simulated domain along the x axis is assumed to be the period of the textures used (either as specified in the texture file or, in the case of symmetric periodicity, twice as much). By using this option, the user can specify a different period along the x axis. The specified period must be larger or at least equal to the period of the textures.

- Forced domain period (Y axis) [um]

If set to the default value of **0**, the period of the simulated domain along the y axis is assumed to be the period of the textures used (either as specified in the texture file or, in the case of symmetric periodicity, twice as much). By using this option, the user can specify a different period along the y axis. The specified period must be larger or at least equal to the period of the textures.

If a larger period is specified by the above two settings, the textures are multiplied as many times as needed to match the new period (in both lateral dimensions). There are two specific cases in which this is typically used. The first is the case in which we want to simulate a much larger device than a single period of the texture (typically in combination with bounded boundary conditions), while the second is the case in which we want to include textures with different periods in the same device (e.g. we can set the “Forced domain period” to the largest period among the textures, and the smaller ones will then be multiplied accordingly).

Note that if any of the above two options is set to **0**, both options are disabled.

- Fit textures to the periods

If a larger period is specified by the above two settings, the textures are multiplied as many times as needed to match the new period (in both lateral dimensions). However, if no multiplication by an integer is possible (i.e. the specified period is not an exact multiplication of the texture period) and this option is set to **0**, the simulation is terminated. Alternatively, if this option is set to **1**, both lateral dimensions (periods) of the textures are scaled in order to ensure a fit to the specified forced domain periods. Note, however, that this scaling may change the texture features (e.g. the aspect ratio) and consequently also the optical behaviour of the texture. Therefore, this option should be used with caution.

- Ray-traced incoherent layers

The index numbers, in order and separated by commas, of the incoherent layers in the structure. The incident medium (index 1) and the medium in transmission should not be specified here; they are automatically assumed to be incoherent.

- Isotropic TF layer growth ratio

If set to **1**, the specified thickness of the thin coherent layers in the input file is considered to be the thickness along the normal of the surface (i.e. isotropic growth of thin layers). If set to **0**, the specified thickness of the thin coherent layers in the input file is

considered to be the thickness along the vertical z axis (i.e. conformal growth of thin layers). In the case of flat interfaces, there is no distinction between the two options. In the case of textures, the first option results in a larger volume of material compared to the second option. The option can also be set to any number between **0** and **1**, in which case a ratio between the two cases is assumed.

- List of nk files of the layers

Following this line is the list of nk files to be used, in order, for each layer of the simulated structure, from front (top) to back (bottom) of the simulated device, including the incident medium and the medium in transmission.

- Layer thicknesses [nm]

Following this line is the list of thickness values (in nanometres), in order, for each layer of the simulated structure, from front (top) to back (bottom) of the simulated device. The thicknesses of the incident medium (first layer) and the medium in transmission (last layer) should be set to 0.

5.2 Complex refractive index files (".nk", located in "nk" folder)

The complex refractive index files are located in the "nk" folder of the project folder. They are used in the simulation to define the material of each layer in the analysed structure. A refractive index file is a plain text file with ".nk" extension. It consists of a simple textual header (ignored by the simulator) followed by three columns of data, as shown in Figure 5.5. The first column lists the wavelengths (in nanometres) at which the refractive index data is given. The second column lists the real refractive index of the material (n), and the third column lists the imaginary refractive index (or the extinction coefficient, k) for each wavelength. A fourth column listing the absorption coefficient (α) may also be present in the file, however it is ignored by the simulator.

Glass			
wavelength nm	n	k	
300.000	1.530	1.671e-5	
305.000	1.530	1.358e-5	
310.000	1.530	1.104e-5	
...	

Figure 5.5: Data structure of an example refractive index file.

5.3 Spectrum files (".spc", located in "Spectrum" folder)

Spectrum files are located in the "Spectrum" folder of the project folder. They are used in the simulation to calculate the potential short-circuit current densities (J_{sc}) from the simulated absorptances in each of the layers (see section 6.4). A spectrum file is a plain text file with ".spc" extension. It consists of a simple textual header (ignored by the simulator) followed by two columns of data, as shown in Figure 5.6. The first column lists the wavelengths (in nanometres) at which the spectrum data is given. The second column lists the power density (in mW/cm^2) of each spectral component.

```
AM1.5 Spectrum IEC60904

wavelength      P
nm              mW/cm2

300.000         3.1430e-3
310.000         0.0653
320.000         0.2043
...             ...
```

Figure 5.6: Data structure of an example spectrum file.

5.4 Texture files (".txt", located in "Texture" folder)

Texture files are located in the "Texture" folder of the project folder. They are used in the simulation to define the texture morphology applied to the top and/or bottom surface of the thick incoherent layers. A texture file is a plain text file with ".txt" extension. It consists of a number of textual headers (ignored by the simulator) and two blocks of numerical data, as shown in Figure 5.7.

The first block (line) of numerical data contains the maximum x and y values (in micrometres) of the texture, and the ray tracing resolution (step) to be used in the vicinity of this texture (if the appropriate option is enabled in the input file; see section 5.1.2). In the example in Figure 5.7, the texture spans from 0 to 5 μm along the x axis, and from 0 to 10 μm along the y axis. The ray tracing resolution is 0.01 μm .

The second block of numerical data contains the matrix of z values (in micrometres), which represents the morphology of the texture as seen from above. The origin of the xy plane is assumed to be in the bottom left corner of the matrix. The x axis increases from left to right, and the y axis increases from bottom to top (see also Figure 5.3). The matrix of z values must have at least 4 elements in each dimension; however the number of elements in both dimensions does not need to be the same. The matrix of z values is interpolated to fit across the entire domain, as defined by the maximum x and y values. During this (linear) interpolation, the data points of the matrix are evenly distributed along each dimension.

If periodic boundary conditions are used in CROWM simulations (see section 5.1.5), all textures used need to be periodic in both dimensions! Therefore, each value at the right-most column of the z matrix should be the same as the value at the left-most column (in the same row), and each value at the top-most row of the z matrix should be the same as the value at the bottom-most row (in the same column). This can also be observed in the example texture file shown in Figure 5.7.

```
# *****
# *                               CROWM v3.3 SIMULATION - TEXTURE FILE                               *
# *****
#
# Example triangular texture.
#
# LATERAL BORDERS and RAY-TRACING STEP (X_max Y_max d_RT - all dimensions in [um]):
# -----
5 10 0.01
#
# TEXTURE MORPHOLOGY (at least 4x4 matrix with Z values - all dimensions in [um]):
# -----
0 1 2 3 2 1 0
0 1 2 3 2 1 0
0 1 2 3 2 1 0
0 1 2 3 2 1 0
```

Figure 5.7: Data structure of an example texture file.

6. DESCRIPTION OF THE OUTPUT FILES

The output files created during a CROWM simulation are stored in the "Out" folder. They all have the same basic name defined by the user (e.g. "Out_name"; see section 4.1.1.6), but different endings.

Typically, seven textual output files are always generated:

- Out_name_Log.txt
- Out_name_RTA.txt
- Out_name_PRTA.txt
- Out_name_PRTA_TE.txt
- Out_name_PRTA_TM.txt
- Out_name_Jsc.txt
- Out_name_IN.txt

In addition, the following file can also be generated if the “Period of plotted rays” setting is set to any number larger than zero (see section 5.1.1):

- Out_name_Rays.bmp

6.1 Log file ("_Log.txt")

The log file summarises the most important information about the CROWM simulation. It lists any potential errors that occurred during the execution (such as missing input data etc.), stores the names of the input and texture files for future reference, and saves the begin/end times of the simulation.

6.2 Reflectance, transmittance, absorptance file ("_RTA.txt")

Reflectance, transmittance, and absorptance file ("RTA file" in short) lists the simulated wavelength-dependent total reflectance from the device, total transmittance through the device, and the absorptance within each layer of the device. The reflectance, transmittance, and absorptance are defined as the relative fractions of the incident light power (or photon flux) that gets reflected, transmitted, or absorbed on its path through the structure, respectively:

$$\begin{aligned} R &= P_{reflected}/P_{spc} & T &= P_{transmitted}/P_{spc} & A &= P_{absorbed}/P_{spc} \\ R &= \phi_{reflected}/\phi_{spc} & T &= \phi_{transmitted}/\phi_{spc} & A &= \phi_{absorbed}/\phi_{spc} \end{aligned}$$

The RTA file is constructed from the power density RTA files (see section 6.3) and contains the values for the total (combined) light, taking the contributions from both polarizations into account (see section 4.1.2.4 for the definition of polarization). The data structure of an example RTA file is presented in Figure 6.1.

Wavelength [nm]	Rtot	A_air	A_glass	...	Ttot
350.0000	0.0186	0.0003	0.0021	...	0.0000
360.0000	0.0190	0.0003	0.0016	...	0.0000
370.0000	0.0196	0.0004	0.0013	...	0.0000
...

Figure 6.1: Data structure of an example RTA file.

6.3 Power density RTA files ("_PRTA.txt")

These files are similar to the RTA file presented in section 6.2. The only difference is that they list the actual power density that gets reflected, transmitted, or absorbed on light's path through the structure. Three PRTA files exist, one for the TE-polarized light, one for the TM-polarized light, and one for the total (combined) polarization (see section 4.1.2.4 for the definition of polarization). The sum of all the values given for a particular wavelength should match the power density given for this wavelength in the spectrum file (see section 5.3).

6.4 Short-circuit current density file ("_Jsc.txt")

The short-circuit current density file lists the potential J_{sc} values calculated from the simulated reflectance, transmittance, and absorptance spectra in each of the layers by applying the illumination spectrum specified by the input file (see section 5.3) according to the following equation:

$$J_{sc} = \frac{q}{hc} \int P_{spc}(\lambda) \cdot A_{layer}(\lambda) \cdot \lambda \cdot d\lambda$$

Please note that the J_{sc} values should be treated merely as indicators of short-circuit current gains/losses throughout the optoelectronic device. J_{sc} calculated for a TCO layer, for example, thus represents the optical losses due to the parasitic absorption in this layer, whereas J_{sc} calculated for the photoactive absorber layer, on the other hand, represents the actual expected generated photocurrent (assuming that all the absorbed photons contribute to charge-carrier generation; i.e. $EQE = A$).

6.5 Copy of the simulation input file ("_IN.txt")

This file is a complete copy of the simulation input file (see section 5.1). It is used for future reference.

6.6 Ray propagation figure file ("_Rays.fig")

The ray propagation figure file is a MATLAB figure which contains the structure of the device (including all textures) superimposed by the trajectories of the rays as they are propagating through the device. Only the rays calculated for the final wavelength of the simulation are included. The thickness of each ray segment corresponds to the current power of the ray.

Note that the ray propagation figure only shows the thick incoherent layers and the textures applied to their surfaces, while the thin coherent layers are not shown as they are assumed to be infinitely thin for the purpose of ray tracing (see section 1.3 for further details).

Further on, if multiplication of the textures is enabled by the “Forced domain period” settings (see section 5.1.5), the actual texture profiles are not plotted in the ray propagation figure file, since that would typically result in an unreasonable amount of data to be displayed. Instead, all interfaces are in this case plotted as flat, although the rays themselves are still plotted realistically as if the textures were there (and as was also the case in the simulation).

The generation of ray propagation figure file can be controlled by the "Period of plotted rays" setting in the input file (see section 5.1.1).