

# SCAPS manual

Marc Burgelman, Koen Decock,  
Alex Niemegeers, Johan Verschraegen, Stefaan Degrave

Version: 17 february 2016





# Table of contents

Table of contents .....	.i
Chapter 1 : About SCAPS .....	1
Chapter 2 : Getting started.....	1
2.1    The basics.....	1
2.2    Run SCAPS:.....	2
2.3    Define the problem: .....	2
2.4    Define the working point .....	2
2.5    Select the measurement(s) to simulate.....	2
2.6    Start the calculation(s): .....	2
2.7    Display the simulated curves, .. ....	3
2.8    ...e.g. the <i>I-V</i> curves .....	3
2.9    Editing the problem.....	4
2.10    Speeding up: Batch calculations .....	4
2.11    Speeding up: Recorder .....	4
Chapter 3 : Solar cell definition .....	5
3.1    Editing a solar cell structure .....	5
3.2    Reference conventions for voltage and current.....	5
3.3    Contacts .....	8
3.4    Layer thickness .....	10
3.5    (Graded) semiconductor layers .....	10
3.5.1 Adding, duplicating, splitting, removing layers from the cell structure.....	10
3.5.2 Temperature dependence of parameters .....	11
3.5.3 Grading: general approach .....	12
3.5.4 Composition grading .....	13
3.5.5 Parameter grading.....	14
3.5.6 The actual position dependent grading results .....	18
3.5.7 A materials approach.....	19
3.5.8 A frequently asked question (FAQ) about grading .....	20
3.6    Defects and recombination .....	22
3.6.1 Adding defects.....	22
3.6.2 Multivalent defects .....	23
3.6.3 Energetic distribution of defect levels .....	25
3.6.4 Impurity photovoltaic effect (IPV) .....	26
3.6.5 Short overview of defect values .....	26
3.6.6 Radiative and Auger recombination .....	26
3.7    Metastable defect transitions.....	27
3.7.1 Principles .....	27
3.7.2 Introduction of a metastable transition .....	28
3.7.3 Help, the buttons to introduce/edit metastable properties are not available!.....	29
3.7.4 Numerical settings .....	29
3.8    Interfaces.....	30
3.9    Tunnelling .....	30

3.9.2 Numerical tunnel settings .....	32
3.10 The blue button .....	33
3.11 Saving and loading problem definitions.....	33
Chapter 4 : Working point definition.....	35
4.1 General .....	35
4.2 Illumination conditions .....	35
4.2.1 Internal SCAPS calculation .....	35
4.2.2 Generation from file .....	36
4.3 The initial working point.....	37
4.4 Shunt conductance and series resistance.....	38
Chapter 5 : Single shot calculations.....	39
5.1 Calculation roadmap .....	39
5.1.1 Meshing .....	39
5.1.2 The pathway to a solution.....	41
5.1.3 Small signal analysis.....	42
5.2 Setting up a single shot simulation.....	42
5.3 Numerical parameters .....	43
5.4 Numerical limitations .....	44
Chapter 6 : Result analysis .....	45
6.1 Navigating to the analysis .....	45
6.2 Zooming and scaling .....	45
6.3 Curve info and legend .....	46
6.4 Measurement specific options .....	46
6.4.1 The energy band panel.....	46
6.4.2 The generation-recombination panel .....	47
6.4.3 The IV-panel.....	47
6.4.4 The ac-panel .....	48
6.4.5 The CV-panel .....	48
6.4.6 The Cf-panel .....	49
6.4.7 The QE-panel.....	50
6.5 Managing measurement data.....	51
6.5.1 The Manage measurements panel.....	51
6.5.2 Structure of a measurement file .....	52
6.6 Saving results .....	54
Chapter 7 : Batch calculations .....	59
7.1 The batch set-up panel.....	59
7.1.2 Custom defined values.....	60
7.2 Varying entire definition files .....	61
7.3 Varying parameters of the initial state work point .....	61
7.4 During calculation.....	61
Chapter 8 : Recorder calculations.....	63
8.1 Setting a recorder .....	63
8.2 Recorder calculations .....	64
8.3 Analysing the recorder results.....	64
Chapter 9 : Curve fitting .....	65
9.1 General principles .....	65
9.2 Setting up the curve fitter .....	65
9.3 Analysing the curve fitting results .....	68
9.4 Curve fitting résumé .....	70
Chapter 10 : Scripting .....	71
10.1 Running SCAPS externally .....	71
10.2 Running a script .....	71
10.2.1 Automate mouse clicking .....	72
10.2.2 Run external programs within SCAPS.....	72
10.2.3 Running dynamically linked libraries.....	72
10.3 The script editor .....	72

10.4 The SCAPS script language .....	73
10.4.1 General .....	73
10.4.2 Load commands .....	74
10.4.3 Save commands .....	75
10.4.4 Action commands .....	77
10.4.5 Clear commands .....	79
10.4.6 Set commands .....	80
10.4.7 Get commands .....	86
10.4.8 Math commands .....	92
10.4.9 Loop commands .....	97
10.4.10 Show command .....	98
10.4.11 Plot commands .....	98
10.4.12 Calculate commands .....	99
10.4.13 The run commands .....	100
References .....	103



# Chapter 1: About SCAPS

SCAPS is a one dimensional solar cell simulation program developed at the department of Electronics and Information Systems (ELIS) of the University of Gent, Belgium. Several researchers have contributed to its development: Alex Niemegeers, Marc Burgelman, Koen Decock, Johan Verschraegen, Stefaan Degraeve. A description of the program, and the algorithms it uses, is found in the literature [1-6].

The program is freely available to the PV research community (universities and research institutes). It runs on PC under Windows 95, 98, NT, 2000, XP, Vista, Windows 7, and occupies about 50 MB of disk space.

The program can be freely downloaded (but...: don't sell, don't distribute further, refer when you publish results obtained with SCAPS). Please report to Marc Burgelman ([Marc.Burgelman@elis.ugent.be](mailto:Marc.Burgelman@elis.ugent.be)) when you have downloaded a SCAPS version (your name and your institution name and address, and the promotor's name for doctorate's students).

Up to now, there was no consistent manual for the program but there was (and is) a collection of add-on manuals describing the novelties in every new version. This manual is based on those previous documents. Also, there are two short and recommendable documents: a text *Getting Started 2011.pdf*, and a presentation *SCAPS introduction.pdf*. These documents are doing exactly what they promise.

SCAPS is originally developed for cell structures of the CuInSe<sub>2</sub> and the CdTe family. Several extensions however have improved its capabilities so that it is also applicable to crystalline solar cells (Si and GaAs family) and amorphous cells (a-Si and micromorphous Si). An overview of its main features is given below:

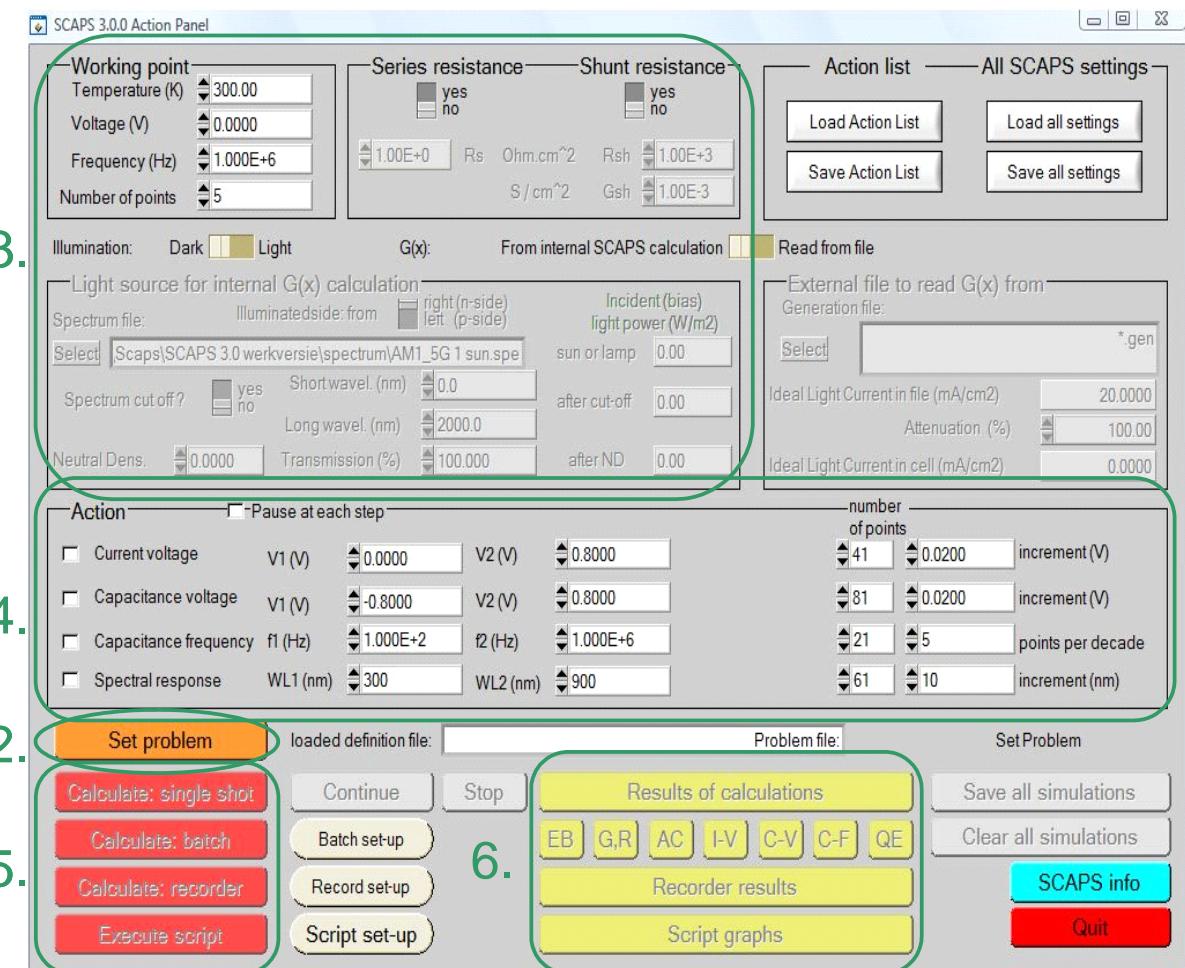
- up to 7 semiconductor layers
- almost all parameters can be graded (i.e. dependent on the ocal composition or on the depth in the cell):  $E_g$ ,  $\chi$ ,  $\varepsilon$ ,  $N_C$ ,  $N_V$ ,  $v_{thn}$ ,  $v_{thp}$ ,  $\mu_n$ ,  $\mu_p$ ,  $N_A$ ,  $N_D$ , all traps (defects)  $N_t$
- recombination mechanisms: band-to-band (direct), Auger, SRH-type
- defect levels: in bulk or at interface; their charge state and recombination is accounted for
- defect levels, charge type: no charge (idealisation), monovalent (single donor, acceptor), divalent (double donor, double acceptor, amphoteric), multivalent (user defined)
- defect levels, energetic distributions: single level, uniform, Gauss, tail, or combinations
- defect levels, optical property: direct excitation with light possible (impurity photovoltaic effect, IPV)
- defect levels, metastable transitions between defects
- contacts: work function or flat-band; optical property (reflection of transmission filter) filter
- tunneling: intra-band tunneling (within a conduction band or within a valence band); tunneling to and from interface states
- generation: either from internal calculation or from user supplied  $g(x)$  file
- illumination: a variety of standard and other spectra included (AM0, AM1.5D, AM1.5G, AM1.5Gedition2, monochromatic, white,...)

- illumination: from either the *p*-side or the *n*-side; spectrum cut-off and attenuation
- working point for calculations: voltage, frequency, temperature
- the programme calculates energy bands, concentrations and currents at a given working point, *J-V* characteristics, ac characteristics (*C* and *G* as function of *V* and/or *f*), spectral response (also with bias light or voltage)
- batch calculations possible; presentation of results and settings as a function of batchparameters
- loading and saving of all settings; startup of SCAPS in a personalised configuration; a script language including a free user function
- very intuitive user interface
- a script language facility to run SCAPS from a ‘script file’; all internal variables can be accessed and plotted via the script.
- a built-in curve fitting facility
- a panel for the interpretation of admittance measurements

# Chapter 2: Getting started

## 2.1 The basics

SCAPS is a Windows-oriented program, developed with LabWindows/CVI of National Instruments. We use here the LW/CVI terminology of a ‘Panel’ (names used in other softwares are: a window, a page, a pop-up...). SCAPS opens with the ‘Action Panel’.



**Figure 2.1** The SCAPS start-up panel: the Action panel or main panel. The meaning of the blocks numbered 1 to 6 is explained in the text.

There are dedicated panels for the basic actions:

1. Run SCAPS .
2. Define the problem, thus the geometry, the materials, all properties of your solar cell
3. Indicate the circumstances in which you want to do the simulation, i.e. specify the working point

4. Indicate what you will calculate, i.e. which measurement you will simulate.
5. Start the calculation(s)
6. Display the simulated curves, ... (see section 6)

This is further explained below.

## 2.2 Run SCAPS:

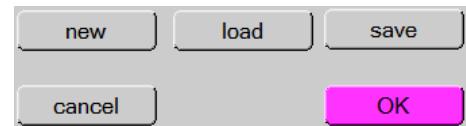


Click the above pictogram on the Desktop, or double-click the file `scaps3200.exe` in the file manager (or any other SCAPS version). SCAPS opens with the Action Panel.

## 2.3 Define the problem:

**Set problem**

Click the button `set problem` in the action panel, and chose `load` in the lower right corner of the panel that opens. Select and open e.g. the file `NUMOS CIGS baseline.def`: that is the example problem file of the practicum session at the NUMOS workshop, Gent, 30 march 2007. This file is supposed to be in the folder `/scaps/def`, where `/scaps/` stands for the directory where you installed SCAPS, and where the SCAPS .exe file resides. If necessary, browse to find this file. In a later stage, you can alter all properties of the cell by clicking `set problem` in the action panel.



## 2.4 Define the working point

The working point specifies the parameters which are not varied in a measurement simulation, and which are relevant to that measurement. Thus:

- the temperature  $T$ : relevant for all measurements. Note: in SCAPS, only  $N_C(T)$ ,  $N_V(T)$ , the thermal velocities, the thermal voltage  $kT$  and all their derivatives are the only variables which have an explicit temperature dependence; you must input for each  $T$  the corresponding materials parameters yourself.
- the voltage  $V$ : is discarded in  $I$ - $V$  and  $C$ - $V$  simulation. It is the dc-bias voltage in  $C$ - $f$  simulation and in  $QE(\lambda)$  simulation. SCAPS always starts at 0 V, and proceeds at the working point voltage in a number of steps that you also should specify.
- the frequency  $f$ : is discarded in  $I$ - $V$ ,  $QE(\lambda)$  and  $C$ - $f$  simulation. It is the frequency at which the  $C$ - $V$  measurement is simulated.
- the illumination: is used for all measurements. For the  $QE(\lambda)$  measurement, it determines the bias light conditions. The basis settings are: dark or light, choice of the illuminated side, choice of the spectrum. A one sun (= 1000 W/m<sup>2</sup>) illumination with the ‘air mass 1.5, global’ spectrum is the default, but you have a large choice of monochromatic light and spectra for your specialized simulations. If you have an optical simulator at your disposal you can immediately load a generation profile as well in stead of using a spectrum.

Working point	
Temperature (K)	300.00
Voltage (V)	0.0000
Frequency (Hz)	1.000E+6
Number of points	5

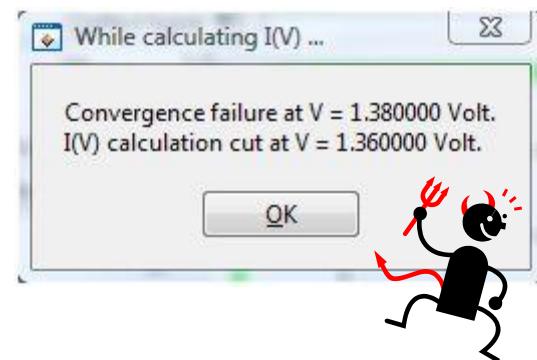
## 2.5 Select the measurement(s) to simulate

In the action-part of the Action Panel, you can select one or more of the following measurements to simulate:  $I$ - $V$ ,  $C$ - $V$ ,  $C$ - $f$  and  $QE(\lambda)$ . Adjust if necessary the start and end values of the argument, and the number of steps. Initially, do one simulation at a time, and use rather coarse steps: your computer and/or the SCAPS program might be less fast than you hope, or your problem could be really tough... A hint: in a  $C$ - $V$  simulation, the  $I$ - $V$  curve is calculated as well, no need then to specify it separately.

## 2.6 Start the calculation(s):

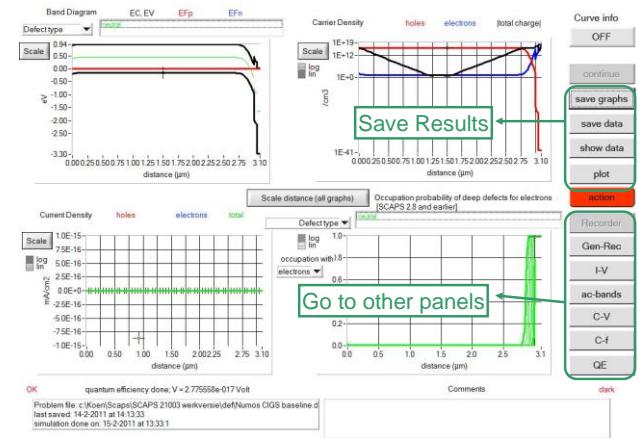
**Calculate: single shot**

Click the button calculate: single shot in the action panel. The Energy Bands Panel opens, and the calculations start. At the bottom of the Panel, you see a status line, e.g. "iv from 0.000 to 0.800 Volt: V = 0.550 Volt", showing you how the simulation proceeds. Meanwhile, SCAPS stands you a free movie how the conduction and valence bands, the Fermi levels and the whole caboodle are evolving. When you see the hated divergence message, you're entitled to get into a bad mood, but don't exaggerate. Anyway, you did not loose the *I-V* points already calculated.



## 2.7 Display the simulated curves, ...

After the calculation(s), SCAPS switches to the Energy band panel (or the AC-band panel). You can now look at your ease to the band diagrams, carrier densities, current densities,... at the last bias point calculated (stop your calculations earlier, or use the pause button on the Action Panel if you want to look at an intermediate state at ease). You can output the results (buttons print, save graphs, show (then the numbers are shown on screen; cut & paste to e.g. Excel is possible), or save (then the numbers are saved to a file). You can switch to one of the specialized output Panels (if you have already simulated at least one corresponding measurement). We only show the example of the IV Panel.



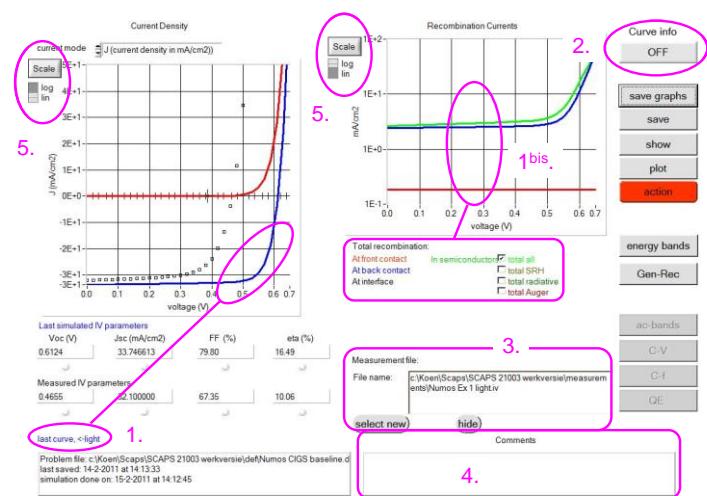
## 2.8 ...e.g. the *I-V* curves

The meaning of the plot, show or save buttons is as for the Energy Bands Panel. Again, you can switch to the other output panels (energy bands, ac, *C-V*, *C-f* and *QE*, if already calculated), and to the Action Panel to do a new calculation, or to stop (important: you can only leave SCAPS from the Action Panel!). Several small remarks:

The color of the last calculated curve is indicated (tip: when the graph gets too crowded, go to the Action Panel and click clear all simulations to clear all graphs). The recombination curves are only shown for the last simulation. The color of the legend corresponds to the color of the curve (indicated as 1bis).

If Curve Info is switched ON and you click the cursor on a curve in a graph, a pop-up panel will appear which gives information about the graph, curve and the point which you clicked.

Here you can display a measurement file (only one measurement at a time!). Select e.g. the file Numos Ex 1 light.iv



or Numos Ex 1 dark.iv which you should find under /SCAPS210/measurements.

Hint: when you are doing many simulations, be friendly and helpful to yourself, and write some comments in the comment box before printing the Panel: you'll be glad to have done so when the time of writing (an article, your doctorate) comes...

You can change the range and scaling of the axes with the Scale button. If you press the CTRL-button and select a rectangular area in a graph, the graph will zoom-in to the selected area. Pressing the CTRL-button and clicking the right mouse button results in zooming out.

## 2.9 Editing the problem

Go to the Action Panel, click set problem. You are now in the Solar Cell Definition Panel. Click on a layer name, and you enter the Layer Properties Panel where you can change all parameters of that layer. Use your intuition and/or read the rest of this manual.

## 2.10 Speeding up: Batch calculations

**Calculate: batch**

**Batch set-up**

When you want to explore the influence of one or a few parameters to the solar cell characteristics, you can take profit of the batch option. When you click Batch set-up, a panel opens where you can choose which parameter to vary, over which range, and in which mode (Lin, Log or custom). You can also define more than one parameter, and vary all of them (in a nested way or ‘simultaneous’), but be modest to start. A batch calculation is launched when calculate: batch is clicked.



## 2.11 Speeding up: Recorder

**Calculate: recorder**

**Record set-up**

In a regular single shot or batch calculation, the detailed panels are only available for the last measurement point. To be able to see them as a function of the batch parameters you can launch a record calculation. You should first select the properties which you want to keep track of by clicking Record set-up. Browse through the property-lists, and don't forget to press one of the insert buttons to add a property to the recorder list. By clicking calculate: recorder, a recorder calculation is launched. Cell parameters are varied according to the Batch set-up, and all simulations are performed which are needed to determine the asked properties. This means the selected measurements on the action panel are ignored!

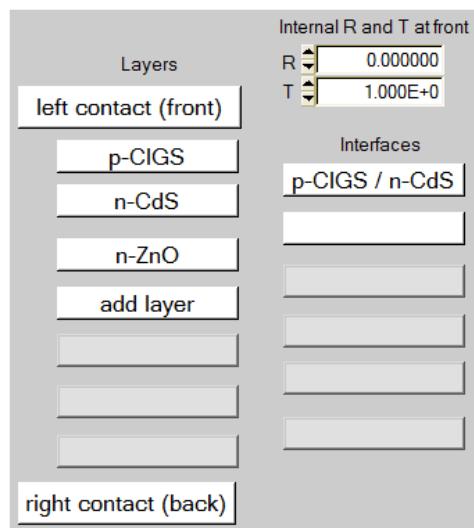
# Chapter 3: Solar cell definition

The recommended way to introduce your solar cell structure into SCAPS is to use the graphical user interface. This way you can interactively set all parameters while SCAPS watches over you, so that you don't define impossible or unrealistic situations. This chapter explains which situations can be modelled and how to introduce them in SCAPS.

## 3.1 Editing a solar cell structure

When clicking the 'Set Problem'-button on the action panel, the 'Solar cell definition'-panel is displayed. This panel allows to create/edit solar cell structures and to save those to or load from definition files. These definition files are standard ASCII-files with extension '\*.def' which can be read with e.g. notepad. Even though the format of these files seems self-explaining it is however strongly disadvised to alter them manually.

Layer-, contact-, and interface properties can be edited by clicking on the appropriate box as shown in Figure 3.1. In a similar way, layers can be added by clicking 'add layer'



**Figure 3.1** Defining a solar cell structure

## 3.2 Reference conventions for voltage and current

The user can input own reference conventions for the applied voltage  $V$  and the current  $J$  in the external contacts. When setting a new problem, or editing an existing problem that does not contain any reference data (e.g. an older .def file), the new options in the solar cell definition panel (Figure 3.2 right) are invisible, and the default reference conventions are set. Upon checking the option in the More Numerical Settings Panel (Figure 3.2 left), these options are visible and can be operated

right away. When a newer problem is loaded that contains reference information, the checkbox ‘allow change of...’ is set automatically, and the three options of Figure 3.2 right are enabled. (As of 2-1-2014, this More Numerical Settings Panel is not yet available to the user; the option “allow change of ... references” is always enabled).

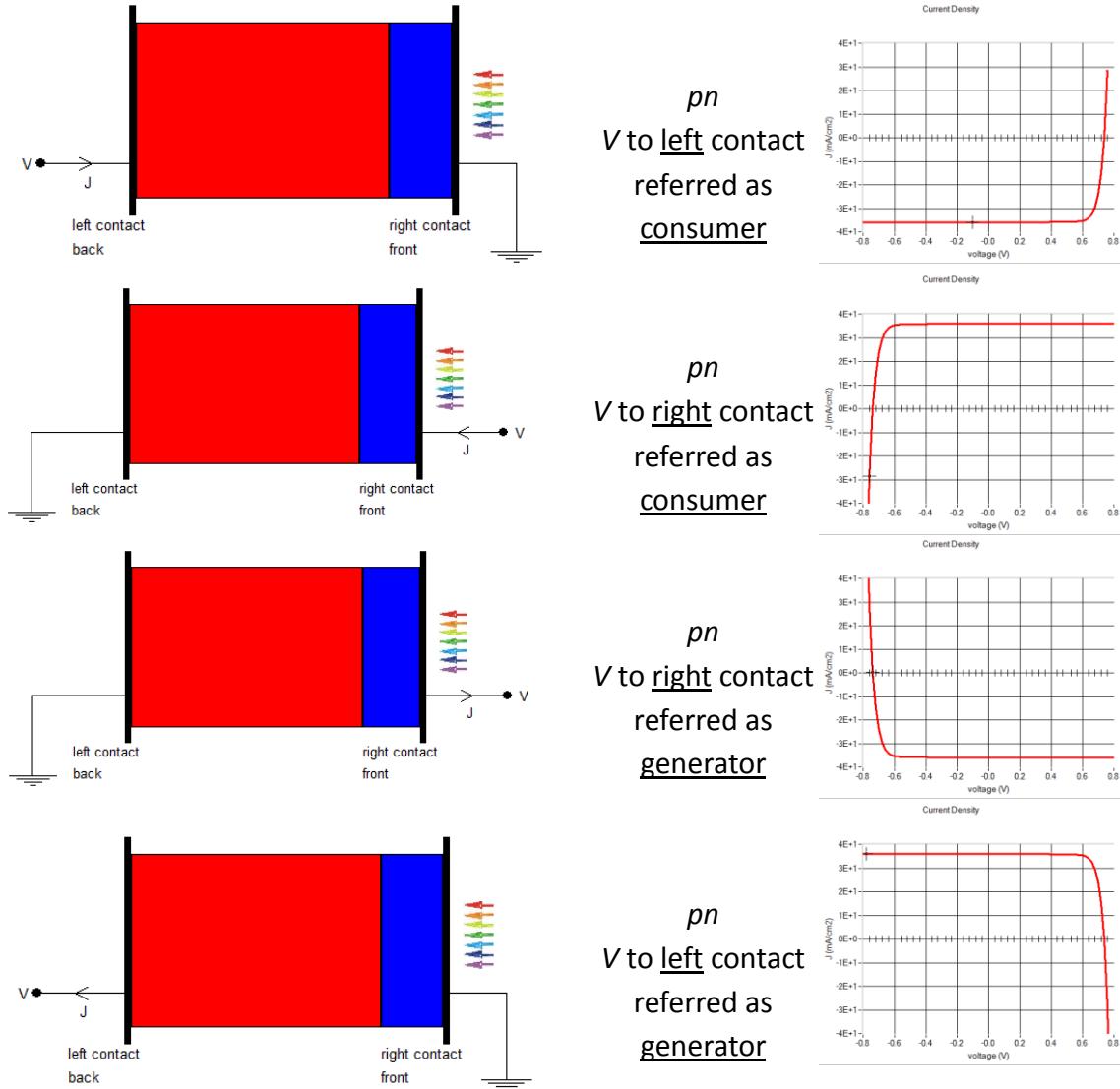


**Figure 3.2** Setting user reference conventions for voltage and current. Left: checkbox in the More Numerical Settings Panel. Right: new facilities in the Solar Cell Definition Panel.

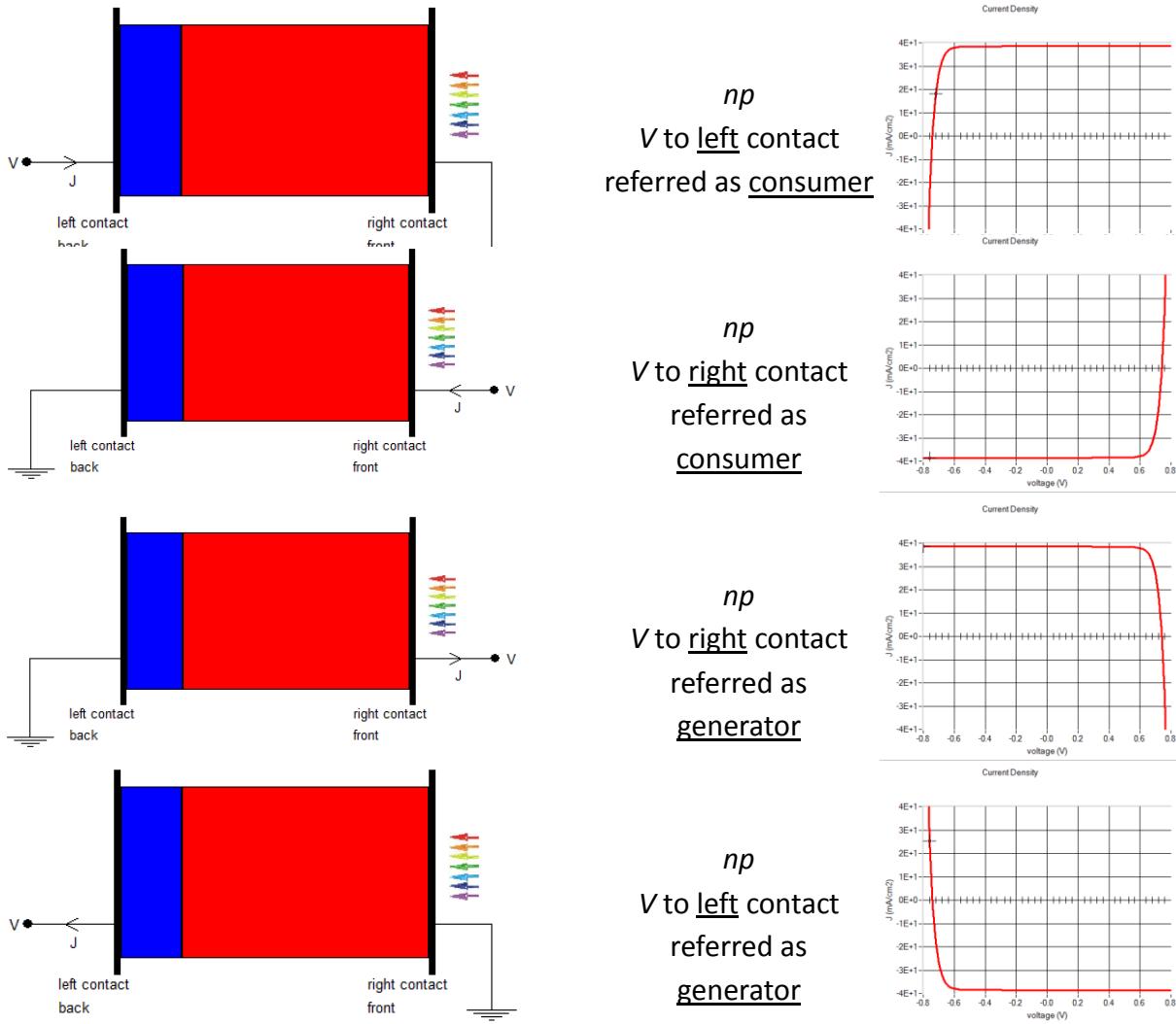
The three new facilities are:

1. ‘apply voltage  $V$  to’: when ‘left’ is set, then the right contact is the reference contact, and the voltage  $V$  is applied to the left contact; this is the default, and the only possible option in SCAPS<4.0. When ‘right’ is set, the left contact is the reference contact, and the voltage  $V$  is applied to the right contact; in an  $JV$  curve, this correspond to a reversal of voltage axis compared to the traditional  $JV$  curves in SCAPS.
2. ‘current reference as a’: when ‘consumer’ is set, then the current reference arrow is set such that  $P = J \times V$  is the power consumed by the cell, and thus  $-J \times V$  the power generated by the cell. When ‘generator’ is set, then the current reference arrow is set such that  $P = J \times V$  is the power generated by the cell, and thus  $-J \times V$  the power consumed by the cell. Setting of the current reference arrow thus depends both on the selected voltage reference and on the consumer/generator selection.
3. ‘Invert the structure’: the solar cell structure is mirrored along the  $x$  axis: the leftmost layer becomes the rightmost layer, and so on. This inversion of structure also swaps the interfaces, and all grading information in the layers and the defects. Clicking two times the inversion button brings the original cell back. This inversion only concerns the structure: the illumination side, the voltage and current reference settings all remain unchanged.

With these 3 settings, one can define 8 different problems, resulting in 4 different aspects of the  $JV$  curves. We illustrate this in Figure 3.3 and Figure 3.4, to make the user more familiar with the concepts of voltage, current and power references.



**Figure 3.3** Possible references of  $V$  and  $J$  for  $pn$  structures. The calculations are for the problem file simple pn.def, and the illumination is always from the right.

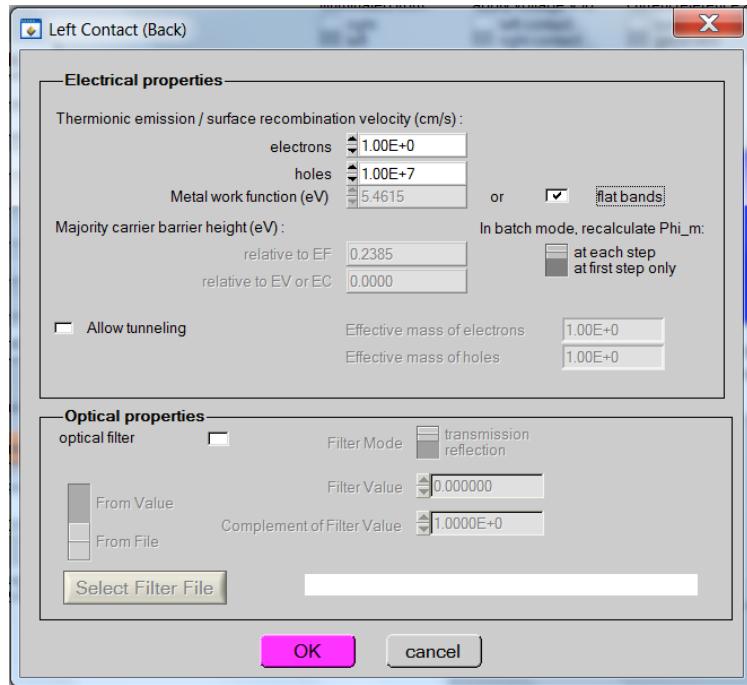


**Figure 3.4** Possible references of  $V$  and  $J$  for  $np$  structures. The calculations are for the problem file simple pn.def, and the structure is inverted with the button ‘Invert the structure’; the illumination is always from the right.

Internally in SCAPS, only the default reference is used (voltage applied at the left contact, current reference arrow from left to right, resulting in a reference as a consumer). In all output (graphs, show/save tables), the result is shown consistent with the user’s choice of reference. Note that the electric field in the SCAPS output is not subject to the user-set  $V$  and  $J$  references: it is always referred to the positive  $x$ -axis, thus from left to right.

### 3.3 Contacts

The contact properties can be set by either clicking the front or back contact button on the cell definition panel, which opens the ‘contact properties panel’, Figure 3.5.



**Figure 3.5** Contact properties panel.

The metal work function  $\Phi_m$  (for majority carriers) can be input by the user. However, the user can also choose the option “*flat bands*”. In this case, SCAPS calculates for every temperature the metal work function  $\Phi_m$  in such a way that flatband conditions prevail. In SCAPS versions before 1-1-2014, a simplified algorithm described below was used. When the layer adjacent to the contact is *n*-type Eq.(1) is used, when it is *p*-type Eq.(2) is used, when it can be considered to be intrinsic Eq.(3) is used.

$$\Phi_m = \chi + k_B T \ln \left( \frac{N_C}{N_D - N_A} \right) \quad (1)$$

$$\Phi_m = \chi + E_g - k_B T \ln \left( \frac{N_C}{N_A - N_D} \right) \quad (2)$$

$$\Phi_m = \chi + k_B T \ln \left( \frac{N_C}{n_i} \right) \quad (3)$$

As can be seen, only the shallow doping density is taken into account in order to calculate the flat band metal work function. When there is a considerable amount of charge in defects present near the contact however, it is thus possible that the flat band option will not lead to flat bands  $\circlearrowright$ .

In recent SCAPS versions (after 1-1-2014), also charge in deep defects is considered; an algorithm involving the solution of a non-linear algebraic equation (expressing that total charge = zero) then replaces the simple equations (1) to (3).

When the layer next to the contact is either *n*- or *p*-type (NOT INTRINSIC) a recalculation of the barrier height with respect to the Fermi level and conductance/valence band are calculated and displayed in the contact properties panel. These values however only serve as an indication to the user, they are not used in the simulation.

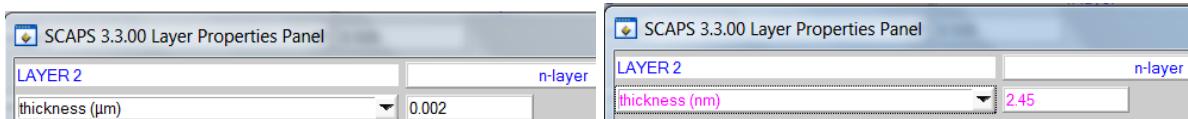
At the contacts a (wavelength dependent) reflection/transmission can be set, see §4.2.1. These can be set either as a constant value (wavelength independent) or as a filter file. These filter-files are standard ASCII-files with the extension ‘\*.ftr’. Several files are provided with the SCAPS installation, however, the user can easily make more files. If a line in this file can be interpreted as starting with at least two numeric values, the

first value is interpreted as the wavelength (in nm) and the second as the reflection (in %). All other lines are ignored and treated as comment.

### 3.4 Layer thickness

Until SCAPS 3.3.00, January 2014, the input values of layer thickness was always in units of  $\mu\text{m}$  (micrometer), the thickness range was from  $1\text{ nm} = 0.001\text{ }\mu\text{m}$  to  $10\text{ cm} = 10^5\text{ }\mu\text{m}$ , and thickness was always displayed with three decimal digits shown. This was annoying when one had input  $d = 0.0025\text{ }\mu\text{m}$  (thus 2.5 nm): SCAPS was calculating with 2.5 nm, but the thickness display showed 0.002  $\mu\text{m}$  (thus 2 nm); and input of a thickness lower than 1 nm was automatically set to 0.001  $\mu\text{m} = 1\text{ nm}$ .

Since SCAPS 3.3.00, August 2014, the allowed input range of thickness is extended to  $0.01\text{ nm} = 10^{-5}\text{ }\mu\text{m}$  at the thin side to  $1\text{ m} = 10^6\text{ }\mu\text{m}$  at the thick side. The number of decimal digits is still limited to 3, but the user can select several units for the thickness display: Å (1 Ångström = 0.1 nm), nm,  $\mu\text{m}$ , mm and cm. When the unit selected is not the traditional micrometer, the value and unit are displayed in magenta colour. When the display unit was set e.g. to  $\mu\text{m}$ , and one had input 0.00245, the display would show the value rounded off to 3 digits: 0.002  $\mu\text{m}$ ; but when one selects nm as display unit, 2.45 nm will be shown (see Figure 3.6). When a new problem is loaded from file, thickness will be displayed in nm if  $d < 10\text{ nm}$ ; in  $\mu\text{m}$  if  $10\text{ nm} \leq d \leq 1000\text{ }\mu\text{m}$ ; and in mm if  $d > 1\text{ mm}$ .



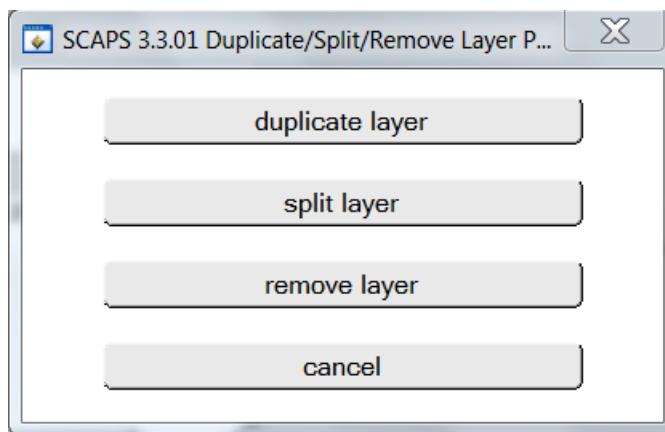
**Figure 3.6** Display of layer thickness, when  $d = 0.00245\text{ }\mu\text{m}$  was input: once with  $\mu\text{m}$  as display unit (in black, left), and once with nm as display unit (coloured, right).

### 3.5 (Graded) semiconductor layers

All parameters of a semiconductor layer can be edited by clicking ‘add layer’ or on the appropriate layer button in the ‘Solar cell definition panel’. All properties can be graded, as will be discussed below. However, some remarks not related to grading should be made first.

#### 3.5.1 Adding, duplicating, splitting, removing layers from the cell structure

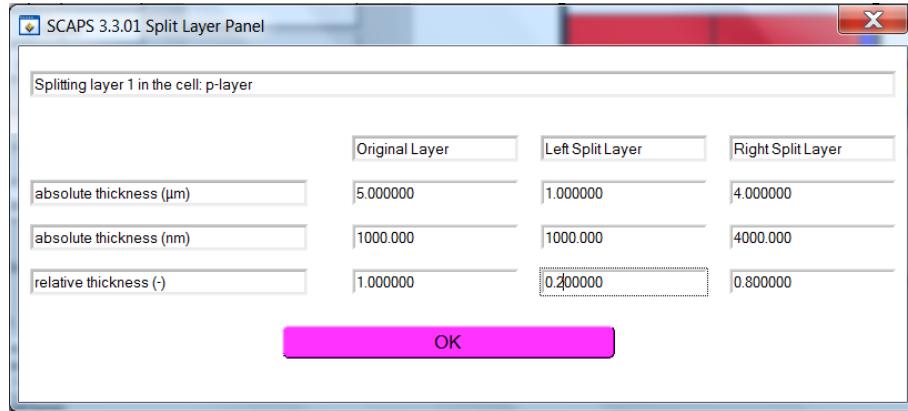
By right-clicking a layer button in the ‘Solar cell definition panel’, a panel opens where you can remove this layer, or duplicate it, or split it, see Figure 3.7.



**Figure 3.7** Panel to remove a layer, or to duplicate or split it.

With ‘duplicate’ an identical layer is inserted after (= to the right of) the layer you right-clicked; in particular, the inserted layer has the same thickness as the original layer. The split option is there from SCAPS 3.3.01, 4-3-2015 on. With split, the thickness and the grading properties (see below) of the original layer are

changed. The original layer is called the ‘left split layer’, and the inserted layer is called the ‘right split layer’. The splitting action conserves the thickness of the original layer. Upon clicking ‘split’ in the panel of Figure 3.7, the Split Layer Panel of Figure 3.8 opens. There, you can set the thickness of the left split layer or of the right split layer, as absolute thickness in  $\mu\text{m}$  or  $\text{nm}$ , or as relative thickness (a fraction of the original thickness). The name of the layer and its duplicate are changed, e.g. in the example of Figure 3.8 with original layer name ‘p-layer’, the left split layer will be named ‘p-layer left’ and the right split layer ‘p-layer right’.



**Figure 3.8** The Split Layer Panel allows to split a layer whilst conserving the total thickness and the grading properties of the original layer.

Also, the grading properties of the split layers are adapted so that the overall grading of all properties is conserved (advancing to the terminology explained in 3.5.3). This works perfect for most grading types. For ‘exponential grading’, a set of transcendental equations has to be solved, we hope that this will work fine for all values you would try, but there is no guarantee. For ‘Beta-function grading’, it is mathematically not possible to find Beta-function solutions for the grading in both split layers. SCAPS will set the Beta-function  $a$  and  $b$  of the split layers in a very rough approximation, that will be unsatisfactory in many cases, especially when  $a > 1$  and  $b > 1$ . It is up to the user to check the results, and judge if they are acceptable or not.

For ‘grading from file’, two new grading files are created for the split layers, and their names are the original grading filename preceded by ‘leftsplit\_’ or ‘rightsplit\_’. Exception: (SCAPS 3.3.03, february 2016) When the grading file has the *range: shared over adjacent layers* keyword, the grading file is used without modification in the two split layers; no two files with different names are created.

When there is no space to duplicate or split a layer, i.e. when the maximum number of layers is already in use (actually 7), the duplicate and split buttons are disabled (and have an appropriate label). Also, you cannot remove a unique layer from a cell structure; when you try, the remove button will be disabled and have an appropriate label.

### 3.5.2 Temperature dependence of parameters

The density of states in the conduction/valence band are temperature dependent according to (4).  $T_0$  is the default temperature (at which the parameter value should be defined in SCAPS and equals 300 K). Likewise the thermal velocity is temperature dependent (5). All other parameters are assumed to be temperature independent. The diffusion coefficient  $D=\mu kT/q$  which is used in the calculations is temperature dependent (contrary to what was mentioned in the very first SCAPS manual (version 2.0))

$$N_C(T) = N_C(T_0) \left( \frac{T}{T_0} \right)^{1.5} \quad (4)$$

$$N_V(T) = N_V(T_0) \left( \frac{T}{T_0} \right)^{1.5}$$

$$v_{th}(T) = v_{th}(T_0) \left( \frac{T}{T_0} \right)^{0.5} \quad (5)$$

Should you want to give a temperature dependence to other parameters, e.g.  $E_g(T)$ ,  $\mu_n(T)$ ,  $\mu_p(T)$ , ..., it is entirely your task. The way to do so is:

In the Batch Set-up, define the temperature  $T$  as a batch parameter. Define for example  $E_g$  as a next, ‘simultaneous’ batch parameter, and set its value from an  $E_g(T)$  list (should the temperatures  $T$  in your list not form a regular series, you should set the  $T$  values also from a list). You should take care to save both lists and maintain them together.

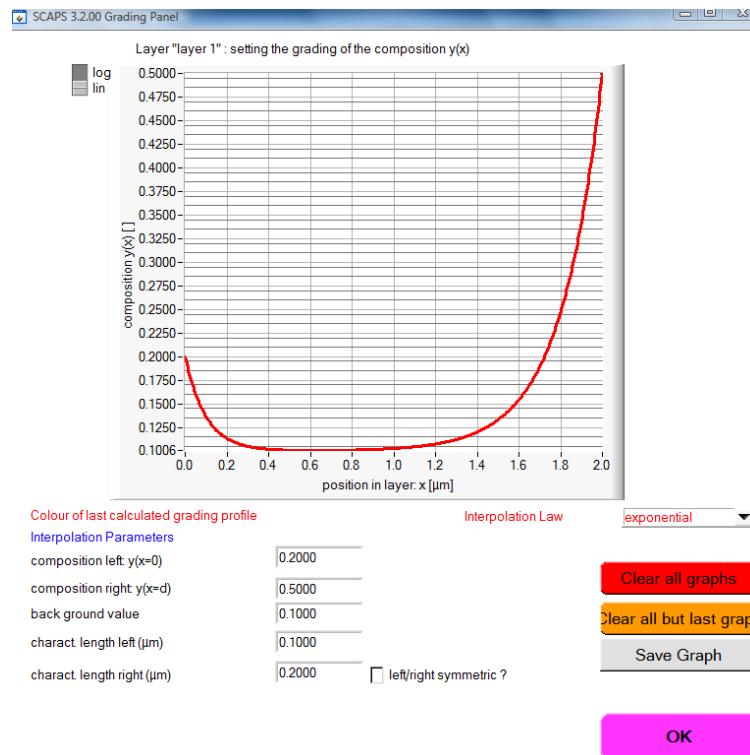
### 3.5.3 Grading: general approach

All layer-parameters can be graded. The principles of the algorithms used to simulate graded solar cell structures have been presented in [3]. To give a suitable and materials oriented description of the grading of the various materials parameters, SCAPS derives all parameters consistently from the composition grading of a layer. Each layer is assumed to have composition  $A_{1-y}B_y$ . The user defines the properties of the pure compounds A (e.g. A = CuInSe<sub>2</sub>) and B (e.g. B = CuInS<sub>2</sub>), and the composition grading  $y(x)$  over the thickness of the layer: thus defining the composition values  $y$  at the left and right side of the layer, and by specifying some grading law in between. All materials properties  $P$  are then derived from the local composition parameter  $y(x)$ , that is,  $P[y(x)]$  is evaluated. Several grading laws are implemented in SCAPS and offered by the user interface: uniform, linear, logarithmic, parabolic (two laws), power law, exponential, effective medium, from file and a Beta function. These grading laws can be used to set the composition grading  $y(x)$  over a layer, as well as to set the composition dependence  $P(y)$  of a property. These grading laws and their parameters can be set on the Grading panel, see Figure 3.9. The grading laws used are summarized in Table 3.1.

**Table 3.1** Available basic grading laws. SCAPS takes care of possible numerical problem, that could occur for very small or very large arguments; hence these laws are not always strictly followed but are more complicated under specific circumstances.

Name	$P(y) = \dots$	remarks
Uniform	$P_A = P_B$	
Linear	$\frac{P_A(y_B - y) + P_B(y - y_A)}{y_B - y_A}$	
Parabolic	$\frac{P_A(y_B - y) + P_B(y - y_A)}{y_B - y_A} - b \frac{(y_B - y)(y - y_A)}{(y_B - y_A)^2}$	$b$ : bowing factor
Parabolic2	$y < y_0 : P_0 + (P_A - P_0) \left( \frac{y - y_0}{y_A - y_0} \right)^2$ $y > y_0 : P_0 + (P_B - P_0) \left( \frac{y - y_0}{y_B - y_0} \right)^2$	There are two parabolas, one to each side of the point $[y_0, P_0]$ which can be given in the user interface. For extrapolation a fourth order equation is used
Logarithmic	$P_A^{\frac{y_B - y}{y_B - y_A}} \times P_B^{\frac{y - y_A}{y_B - y_A}}$	
Exponential	$P_0 + (P_A - P_0) \frac{\sinh\left(\frac{y_B - y}{L_A}\right)}{\sinh\left(\frac{y_B - y_A}{L_A}\right)} + (P_B - P_0) \frac{\sinh\left(\frac{y - y_A}{L_B}\right)}{\sinh\left(\frac{y_B - y_A}{L_B}\right)}$	$P_0$ : background value $L_{A,B}$ : characteristic lengths

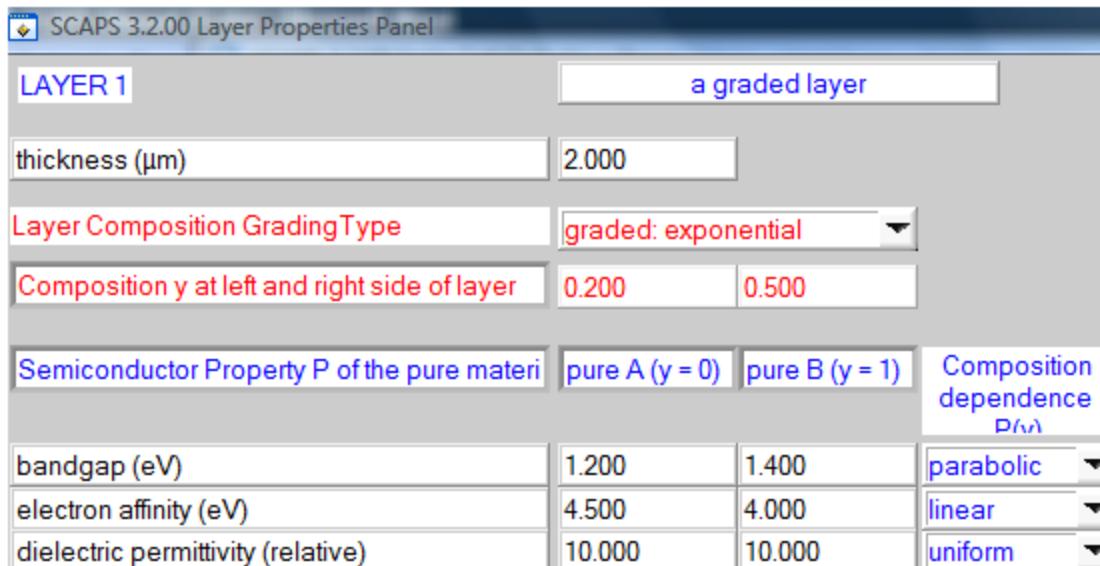
Beta function	$P_A + (P_A - P_B) \beta_{a,b} \left( \frac{y - y_A}{y_B - y_A} \right)$	$\beta_{a,b}(x)$ is the incomplete beta-function ( $I_x(a,b)$ ) in [7] on p.226)
Power law	$\left( P_A^{1/m} \times \frac{y_B - y}{y_B - y_A} + P_B^{1/m} \times \frac{y - y_A}{y_B - y_A} \right)^m$	$m$ : power
Effective medium	$(y_B - y) \frac{P_A - P}{P_A + 2P} + (y - y_A) \frac{P_B - P}{P_B + 2P} = 0$ , thus: $P = \frac{b + \sqrt{b^2 + 8P_A P_B}}{4}$ with $b = \frac{(y - y_A)(2P_B - P_A) + (y_B - y)(2P_A - P_B)}{y_B - y_A}$	Not available for composition. It is the Bruggeman equation.



**Figure 3.9** The Grading panel, in this example an exponential composition grading is set.

### 3.5.4 Composition grading

Composition grading  $y(x)$  is the basic grading of the layer and has extra possible grading laws: the definition of uniform is somewhat more complicated than for parameter grading and a the grading can be loaded from a file. The composition grading can be set by clicking the ‘Layer composition grading type’, see Figure 3.10, which displays the ‘Grading Panel’



**Figure 3.10** Setting the composition grading  $y(x)$ . Select a grading law for the composition (in this example exponential). The values  $y_{\text{left}}=0.2$  and  $y_{\text{right}}=0.5$  are just indications here. You can only set them on the grading panel which appears when selecting a grading law.

### 3.5.4.2 Uniform composition grading

There are three possible definitions of ‘uniform’:

- ‘uniform pure A ( $y = 0$ )’. The composition in this layer is  $y = 0$  for all positions  $x$ . You see only the column of the materials properties of the pure material A ( $y = 0$ ), with no button available to set a grading of these parameters. All parameters  $p$  get the value  $p(y = 0)$ . Position grading of the doping and defect density is still possible, e.g.  $N_A(x), \dots$ .
- ‘uniform pure B ( $y = 1$ )’. The composition in this layer is  $y = 1$  for all positions  $x$ . You see only the column of the materials properties of the pure material B ( $y = 1$ ), with no button available to set a grading of these parameters. All parameters  $p$  get the value  $p(y = 1)$ . Position grading of the doping and defect density is still possible, e.g.  $N_A(x), \dots$ .
- ‘uniform  $y, 0 < y < 1$ ’. The composition in this layer is  $y = \text{constant}$  for all positions  $x$ , and you can set this constant composition in the grading panel. You see both columns of the materials properties of the pure material A ( $y = 0$ ) and B ( $y = 1$ ), and you can set a grading of each of these parameters, to give them the uniform value  $p(y)$ .

Even though it is possible to set a (position dependent) grading of doping and defect densities when the composition grading of the layer is either uniform A or uniform B, it is strongly advised to use the uniform-y-option.

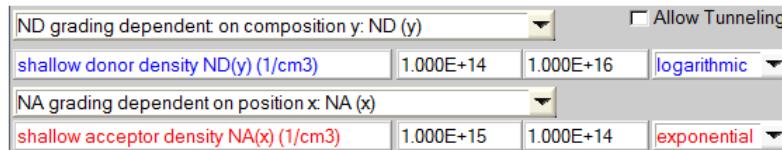
### 3.5.4.3 Composition grading from file

The composition grading profile can be loaded from a file. The rules and conventions used are the same as for specifying parameter grading ‘from file’, and are described further in section 3.5.5.2.

## 3.5.5 Parameter grading

### 3.5.5.1 Position dependent parameter grading

Materials physics impose that we should implement also position dependent grading (and not only composition dependent grading) as an option for the shallow doping densities  $N_D$  and  $N_A$ , and for the defect densities  $N_t$ .



**Figure 3.11** Here, the shallow donor density is given a grading as a function of composition  $y$ :  $N_D(y)$ . The two values displayed are the values of the pure A material ( $10^{14} \text{ cm}^{-3}$ ) and the pure B material ( $10^{16} \text{ cm}^{-3}$ ). The shallow acceptor density  $N_A(x)$  is given a grading as a function of position  $x$ :  $N_A(x)$ . The two values displayed are the values at the left side ( $10^{15} \text{ cm}^{-3}$ ) and at the right side ( $10^{14} \text{ cm}^{-3}$ ). Notice the use of the colour code (red for position, blue for composition grading).

### 3.5.5.2 Grading ‘from file’

All graded profiles (composition grading  $y(x)$  and parameter grading  $P(y)$  or  $P(x)$ ) can be specified ‘from file’. The (ASCII-text) files containing these profiles should be saved in the folder ‘grading’, and the default extension is .grd. The file which contains the grading profile should consist of two columns of numerical data, the first number representing the position in the layer in  $\mu\text{m}$ , the second representing the composition (usually a number between 0 and 1). All lines which cannot be interpreted as two numerical data are treated as comment. Make sure that the data are in the first column are in ascending order!

When the grading file is used to specify grading of a parameter as a function of composition  $y$ , thus  $P(y)$ , the first column is interpreted as the composition. Normally, only the values between  $y = 0$  and  $y = 1$  are used (see remark in the next section). If necessary, SCAPS will extrapolate the file data to obtain the property for the pure A material ( $y = 0$ ) and for the pure B material ( $y = 1$ ).

When the grading file is used to specify grading of a parameter as a function of position  $x$ , thus  $P(x)$ , the first column is by default interpreted as the position  $x$  in  $\mu\text{m}$ . Note that this position-dependent grading is only possible for the composition  $y(x)$ , for the doping densities  $N_A(x)$  and  $N_D(x)$  and for the defect densities  $N_t(x)$ . If the layer thickness exceeds the range of the first column, extrapolation is used. However, you can also force SCAPS to interpret the first column as the relative position  $x/d$ , that is the position relative to the layer thickness. In that case, only the data between 0 and 1 are used. To specify whether the first column is the absolute or the relative position, scaled to the thickness, insert a line in the grading file, starting with:

position: absolute (the default), or:

position: relative

To get the property of the  $x$  or  $y$  values required by the programme, interpolation between the file input values is used. This interpolation can be linear, or ‘logarithmic’ (that is, the property is first ‘plotted’ in a logarithmic plot, and is then linearly interpolated). To specify which interpolation mode to use, insert a line in the grading file, starting with:

interpolation: linear (this is the default for  $y, E_g, \chi, \varepsilon, \mu, m_{\text{eff}}$ ), or:

interpolation: logarithmic (this is the default for  $N_C, N_V, N_D, N_A, N_t, v_{\text{th}}, C_{\text{rad}}, C_{\text{Auger}}$ )

Addendum SCAPS 3.3.03, february 2016. There is an option to apply one and the same grading file over several layers. You can specify this by inserting a line in the grading file, starting with:

range: this layer only (the default), or:

range: shared over adjacent layers

The grading file is checked for the occurrence of ‘range: this’ or ‘range: shared’ only. When the shared option is set, the grading file applies over all layers of a group of adjacent layers:

- it can only be applied to properties that are graded as a function of position  $x$ , not as a function of composition  $y$ . These are:  $y(x)$ ,  $N_A(x)$  and  $N_D(x)$ . As of now, not  $N_t(x)$ .
- the layers in the group should be adjacent to each other
- ‘grading from file’ must be set in all layers of the group

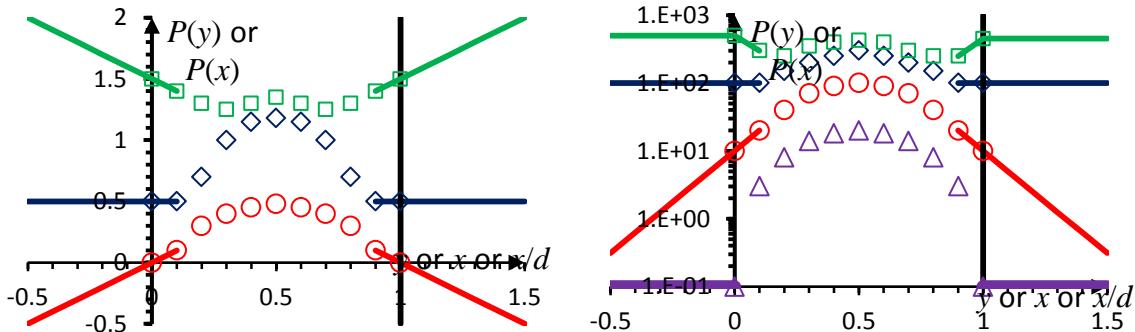
- the same grading file (exactly the same name, thus the same file) must be set in all layers of the group
- this grading from file must be set for the same property: all  $y(x)$ , or all  $N_A(x)$ , ...

The rules for extrapolation, should it be necessary, are described in the next paragraph. They apply to all interpolation of file input data in SCAPS, in particular also to absorption  $\alpha(\lambda)$  files.

### 3.5.5.3 Extrapolation conventions in SCAPS

To prevent that extrapolated data shoot out to infinity, SCAPS applies a few precautions when logarithmic extrapolation is asked. Therefore, the first two points and the last two points in a file table are examined. The basic rule is that an extrapolated property is not allowed to increase when going away from either side of the specified interval. Additionally, when an end point is zero, all values before or after this end point are set to zero. These rules have been adapted and refined somewhat in the evolution of the SCAPS versions. Therefore, slightly different results can be obtained with different versions; a main cause is a different extrapolation of  $\alpha(\lambda)$  files for long wavelengths. Figure 3.12 below illustrates these extrapolation conventions. Tip: you can make sure that values outside the specified range are extrapolated to zero by adding manually a zero end point to your data file.

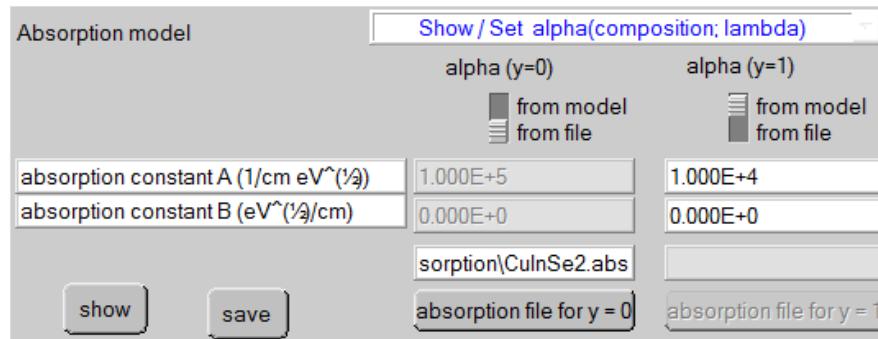
Notice that a property can get a negative value by extrapolation when linear interpolation is used. We allow this because we want to support users who run simulations with a negative composition  $y < 0$ . For example, a user could define  $\text{Ga}_{0.75}\text{Al}_{0.25}\text{As}$  as the ‘pure A’ material, and  $\text{Ga}_{0.25}\text{Al}_{0.75}\text{As}$  as the ‘pure B’ material. Then  $y = -0.5$  corresponds to GaAs and  $y = 1.5$  corresponds to AlAs. There is little chance that a user would like to do so with the well documented Ga-Al-As system, but e.g. with the still largely unknown CZTS system, we cannot foresee for which compositions  $\alpha(\lambda)$  data would be available to a user, and for which compositions she or he would like to simulate...



**Figure 3.12** Conventions for extrapolation of file input data in SCAPS. The definition interval is between the two vertical thick lines. The data specified in the file are shown with open symbols; the SCAPS extrapolation with solid lines of the same colour. Left: ‘Linear interpolation’. Extrapolation is also linear, and notice that the extrapolated values can be negative. Right: ‘Logarithmic interpolation’. Extrapolation is also logarithmic, but with the restriction that  $P$  cannot rise. For the lowest curve,  $P = 0$  at the edges  $y = 0$  and  $y = 1$  of the specified abscissa range, and the extrapolation yields  $P = 0$  outside the specified abscissa range; these values are shown here as  $P = 0.1$ , just to make them appear in the figure, but internally they are set rigorously to  $P = 0$ .

### 3.5.5.4 The optical absorption constant $\alpha$

The optical absorption constant can be set from either from a model or from a file, see Figure 3.13. When it is set from a model,  $\alpha(\lambda)$  is given by (6).



**Figure 3.13** Setting the absorption constant from file (left) or from model (right). The grading can be shown or set by clicking the ‘Show/set alpha’-button.

$$\alpha(\lambda) = \left( A + \frac{B}{h\nu} \right) \sqrt{h\nu - E_g} \quad (6)$$

A wavelength independent  $\alpha$  can be set by using one of the ‘Gray xEx.abs’-absorption files which are available. Next to a limited library of absorption files which are present in the SCAPS distribution, you can also make your personal absorption files. These are ASCII-files with the extension ‘\*.abs’. If a line in this file can be interpreted as starting with at least two numeric values, the first value is interpreted as the wavelength (in nm) and the second as  $\alpha$  (in 1/m). All other lines are ignored and treated as comment.

The grading of the optical absorption (no matter whether it has been defined from a model or from a file) needs a dedicated interpolation algorithm [3] to determine a grading dependent  $\alpha(\lambda, y(x))$ .

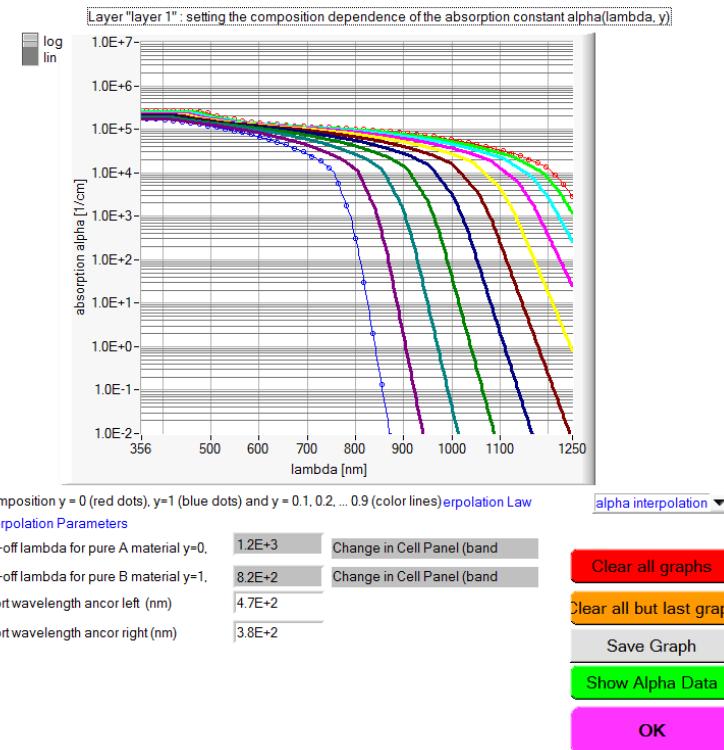
To interpolate the optical absorption constant  $\alpha(\lambda, y)$  for some composition between the pure material A with composition  $y = 0$  and absorption  $\alpha_A(\lambda)$ , and the pure material B with composition  $y = 1$  and absorption  $\alpha_B(\lambda)$ , SCAPS uses the following algorithm. First, determine the cut-off wavelengths  $\lambda_{gA}$  and  $\lambda_{gB}$  of the pure materials, and a characteristic wavelength  $\lambda_{0A}$  and  $\lambda_{0B}$  in the near UV wavelength range. Usually, the  $\alpha(\lambda)$  curves have a maximum (peak) in the near UV, if not one can take an arbitrary value for  $\lambda_{0A}$  and  $\lambda_{0B}$ . Then determine the cut-off wavelength  $\lambda_g$  of the compound with composition  $y$ :  $\lambda_g = 1240\text{nm.eV}/E_g(y)$ , and the UV peak wavelength  $\lambda_0$  from linear interpolation between  $\lambda_{0A}$  and  $\lambda_{0B}$ . A first estimation for  $\alpha(\lambda)$  is then obtained by evaluation  $\alpha_A$  at a wavelength  $\lambda_A$  given by

$$\lambda_A = \frac{\lambda_{gA}(\lambda - \lambda_0) + \lambda_{0A}(\lambda_g - \lambda)}{\lambda_g - \lambda_0} \quad (7)$$

A second estimation is found by evaluation  $\alpha_B$  at a wavelength  $\lambda_B$  found in a way similar to Eq. (7). Then take a weighted logarithmic average between the two estimations:

$$\log \alpha = (1 - y) \log [\alpha_A(\lambda_A)] + y \log [\alpha_B(\lambda_B)] \quad (8)$$

The merits of this interpolation algorithm are discussed in [3].

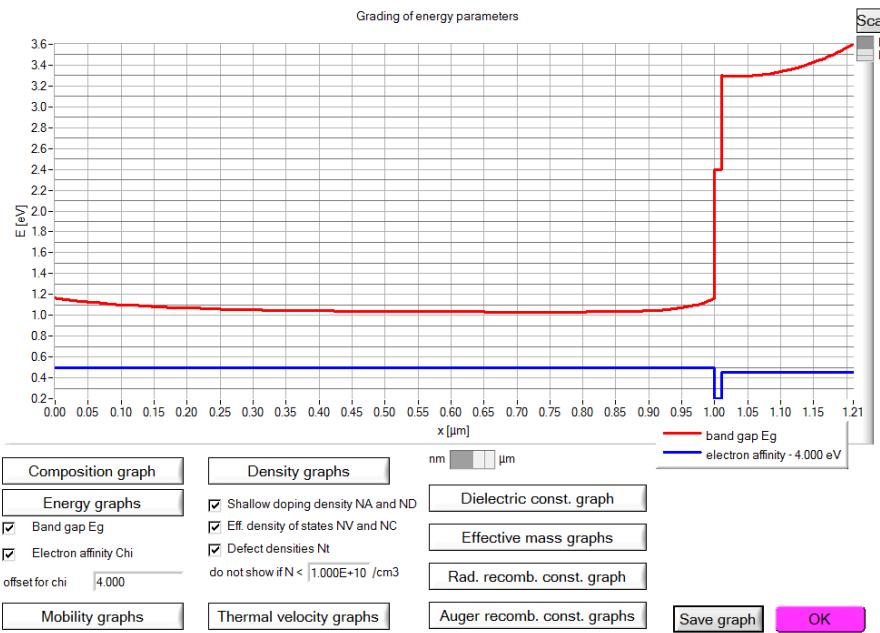


**Figure 3.14** The optical absorption constant  $\alpha(\lambda, y)$  of a compound material with composition  $y$ , thus  $A_{1-y}B_y$ , calculated by SCAPS with a dedicated interpolation algorithm. Here, the pure A material is  $CuInSe_2$ , and the pure B material  $CuInS_2$ . The data can be obtained in tabular form by clicking Show Alpha Data (the green button)

### 3.5.6 The actual position dependent grading results

The graded properties get their position dependence directly ( $N_D(x)$ ,  $N_A(x)$ ,  $N_i(x)$  when these options are selected), or indirectly via the  $x$ -dependence of the composition  $y$ :  $P[y(x)]$ , all other cases. The  $x$ -dependent values that are finally used in the calculations can be obtained from the Cell Definition Panel, but only after at least a calculation is done (a working point calculation is sufficient, unclick  $I-V$ ,  $C-V$ ,  $C-f$ , and  $QE$ ). Hereto click one of the green save, show or graph view buttons on the Cell definition panel. Graph View was added in SCAPS3.3.1, March 2015, finally ☺.

All grading data:



**Figure 3.15** The Graph View Panel allows a quick overview of all parameters that can be graded. Parameters that one would possibly like to view in one graph are grouped together.

When using the saving button (not the show-button) you also get the values of all  $\alpha(\lambda,x)$ -values, where the  $\lambda$ -values are the wavelengths present in the spectrum you used to perform the simulation and the  $x$ -values the meshpoints. Combining this facility with a uniform  $y$ -graded layer allows you to construct personalized absorption files as well ☺.

If you want to have a graphical view of the graded data, the best way to go is to use the recorder function which is introduced in Chapter 8. Then you have acces to more variables and saving and showing options as well.

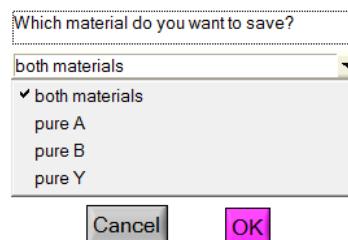
### 3.5.7 A materials approach

All parameters of a layer can be defined separately, but they can also be loaded from and saved to a '.material' file, using the appropriate buttons on the layerpanel.

A material consists of all layer parameters excepting the layer name, the layer thickness, the doping density and the defect properties.

#### 3.5.7.1 Saving materials

Materials can be saved. This results in an ascii-file with extension '\*.material'. There are four different options to save a material: both materials, pure A, pure B and pure Y. The options pure Y and both materials are not available when the compositiongrading is set to pure A or pure B. In that case only one option (pure A or pure B) is available.



**Figure 3.16** Available options for saving materials

Saving a material using the pure A/B option only saves the parameter values of the A/B-material of the layer. All parametergrading is set to uniform.

Saving a material using the both materials option saves both parameters of the A/B-material of the layer together with the appropriate parametergrading parameters.

When saving a material using the pure Y option you should give a fraction  $y$  which defines the material to be saved as  $A_{1-y}B_y$ . The appropriate parameter values for this material are then calculated according to the values for the A/B-material of the layer and saved as a uniform material (similar to pure A/B). The absorption for this material is also calculated as an interpolation of the absorption for A and B and is saved in an absorption file. This file is automatically added to your absorption folder and gets the same name as your material-file extended with '.abs'.

### 3.5.7.2 Loading materials

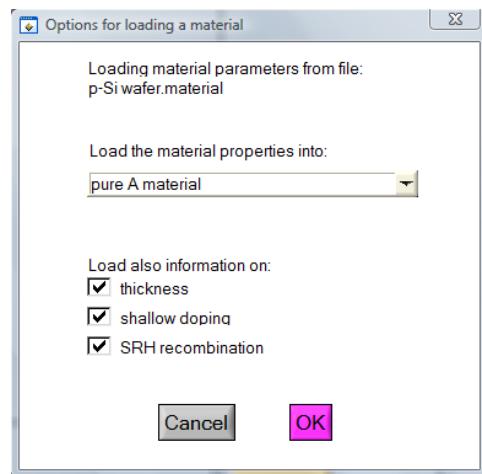
Materials which have been saved, can of course be loaded as well. Three options are available: pure A, pure B and both materials. When the composition grading of the layer was saved to pure A/B however, only the pure A/B option remains available for loading.

Loading a material using the both materials option sets all material parameters of the A- and B-side of the layer, including the parametergrading. When loading a material which has been saved as a pure A/B/Y material, the parameters of the A- and B-side of the layer will be identical and all parametergrading will be set to uniform.

Loading a material using the pure A/B option only sets the parameters of the A-/B-side material of the layer. If the parametergrading was uniform it will be set to linear.

If the materials file contains information on thickness, on the shallow doping densities or on defects, you can load this information or not: select or unselect the appropriate check box in the Load Material Panel (Figure 3.17). As a materials file can contain file names (optical absorption files, grading files), you should make sure that these files are indeed present in the appropriate SCAPS directory.

Material files can also be varied in a batch calculation (see Chapter 7).



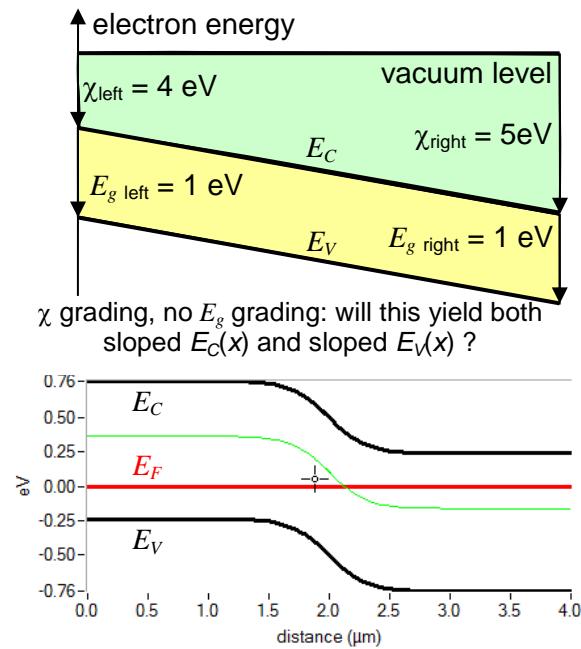
**Figure 3.17** Available options for loading materials

### 3.5.8 A frequently asked question (FAQ) about grading

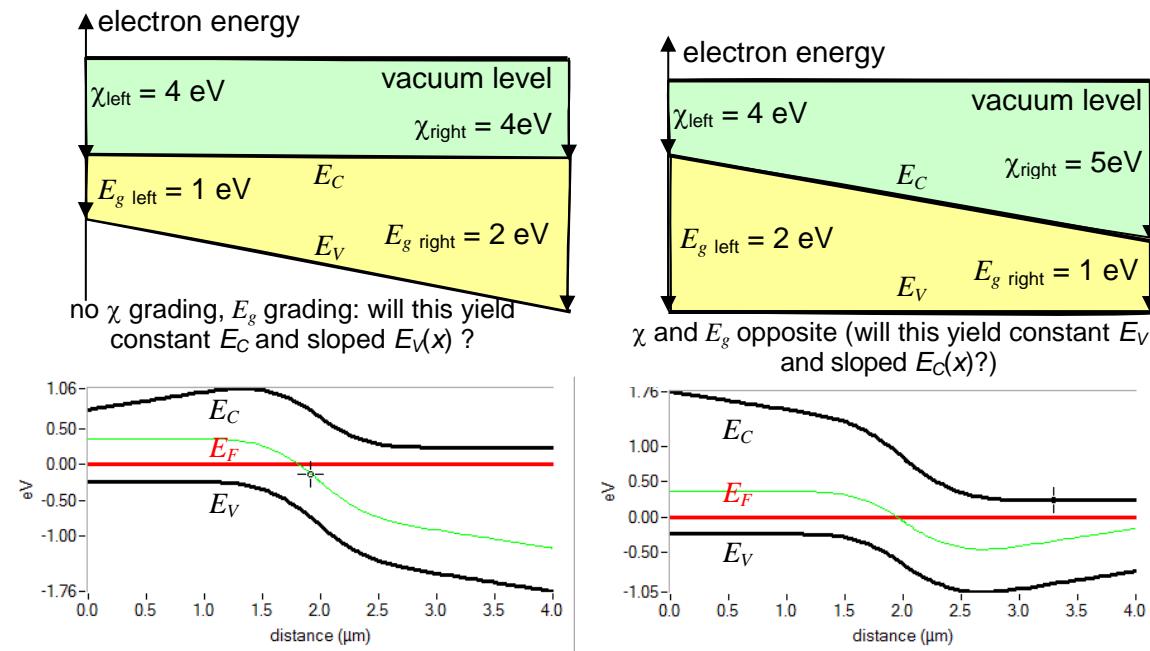
Sometimes we think of a desired or supposed band diagram, with a sloped (non horizontal) conduction band and/or valence band. The model input parameters are the grading of the electron affinity  $\chi(x)$  and of the band gap  $E_g(x)$ . There is sometimes misunderstanding about the relation between the  $\chi(x)$  and  $E_g(x)$  input, and the band bending  $E_C(x)$  and  $E_V(x)$  they cause: the result depends on the doping density and on the voltage and illumination conditions. The rules of thumb are given below and illustrated in Figure 3.18 and Figure 3.19 below:

- In equilibrium, in a neutral, *p*-type region (e.g. the absorber bulk in CIGS),  $E_F = E_{Fp}$  = horizontal (equilibrium),  $E_V$  is a fixed amount  $kT \times \ln(N_V/N_A)$  under it (depending on the doping density  $N_A$ ; supposing that both  $N_A$  and  $N_V$  are uniform and not graded): thus the valence band will be horizontal, regardless the band gap and/or electron affinity grading. The conduction band  $E_C$  is then placed at a distance  $E_g$  above  $E_V$ , thus:  $E_C(x) = E_V + E_g(x)$ , and will thus be sloped when  $E_g(x)$  is graded. You will not see any effect on electron affinity grading.
- in equilibrium, in an *n*-type region,  $E_F = E_{Fn}$  = horizontal. The conduction band  $E_C$  is placed at a fixed distance  $kT \times \ln(N_C/N_D)$  above it (this is a constant when both  $N_D$  and  $N_C$  are uniform and not graded). Thus the conduction band will be horizontal, regardless the electron affinity grading. The valence band will then be placed a distance  $E_g$  under  $E_C$ , thus:  $E_V(x) = E_C - E_g(x)$  and will be sloped when  $E_g(x)$  is graded.

In all other circumstances (applied  $V$ , illumination, in a depletion layer, grading of the doping densities  $N_D(x)$ ,  $N_A(x)$  or of the densities of states  $N_C(x)$ ,  $N_V(x)$ ) there are no simple rules of thumb, it gets too complicated. Trust e.g. SCAPS for it. But always check SCAPS (that is, check what you have input) by always looking to the equilibrium band diagram and applying the rules of thumb above where possible (that is, in the flat-band regions).



**Figure 3.18** Developing ‘intuition’ about grading. Grading schemes of electron affinity  $\chi(x)$  and of band gap  $E_g(x)$ :  $\chi(x)$  is graded, but  $E_g$  is not. What is suggested by the naïve sketch (top) and what really happens in a *pn* junction (SCAPS calculation, bottom).



**Figure 3.19** [more...] Developing ‘intuition’ about grading. Grading schemes of electron affinity  $\chi(x)$  and of band gap  $E_g(x)$ : [left] no  $\chi$  grading but an  $E_g(x)$  grading and [right] both  $\chi(x)$  and  $E_g(x)$  are graded, but their sum  $\chi(x) + E_g(x) = 6$  eV = constant. What is suggested by the naïve sketch (top) and what really happens in a *pn* junction (SCAPS calculation, bottom).

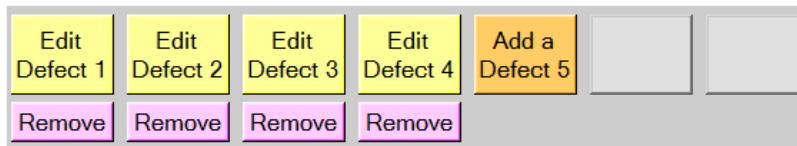
### 3.6 Defects and recombination

In a diode, current is converted from hole current at the *p*-contact to electron current at the *n*-contact. This means that somewhere in the diode recombination MUST take place, even in the most ideal device. So the user MUST specify recombination somewhere, at least at one place (in a layer, at a contact or at an interface). If (s)he does not do so, a convergence failure will result in non-equilibrium conditions (non zero voltage, and/or illumination).

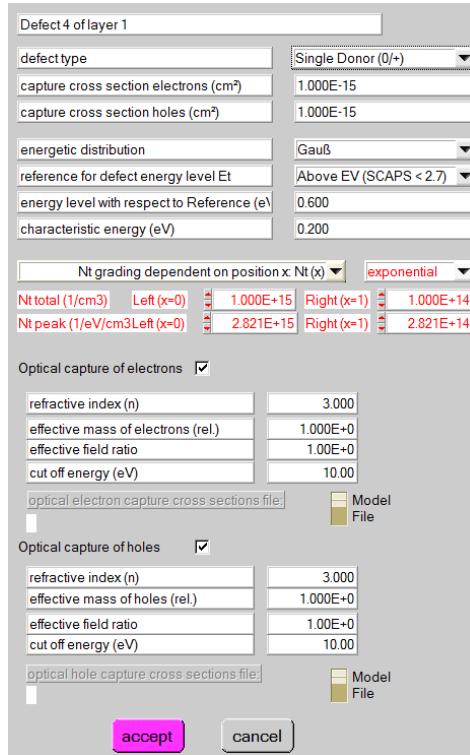
In the bulk of a semiconductor layer, three different kinds of recombination processes can be introduced: through defects, radiative and Auger.

#### 3.6.1 Adding defects

Up to seven defects can be introduced in a semiconductor layer. The parameters governing each defect can be edited by clicking the appropriate Add/Edit-button, Figure 3.20, which opens the ‘defect properties panel’, Figure 3.21. Also, you can right-click on one of the ‘defect summary text boxes’; a panel then opens where you can remove, duplicate or add a defect, much the same as it was with removing, duplicating and adding a layer (Section 3.5.1 and Figure 3.7, page 10); of course, ‘splitting a defect’ is not available as it has no meaning.



**Figure 3.20** Adding, editing and removing defects.



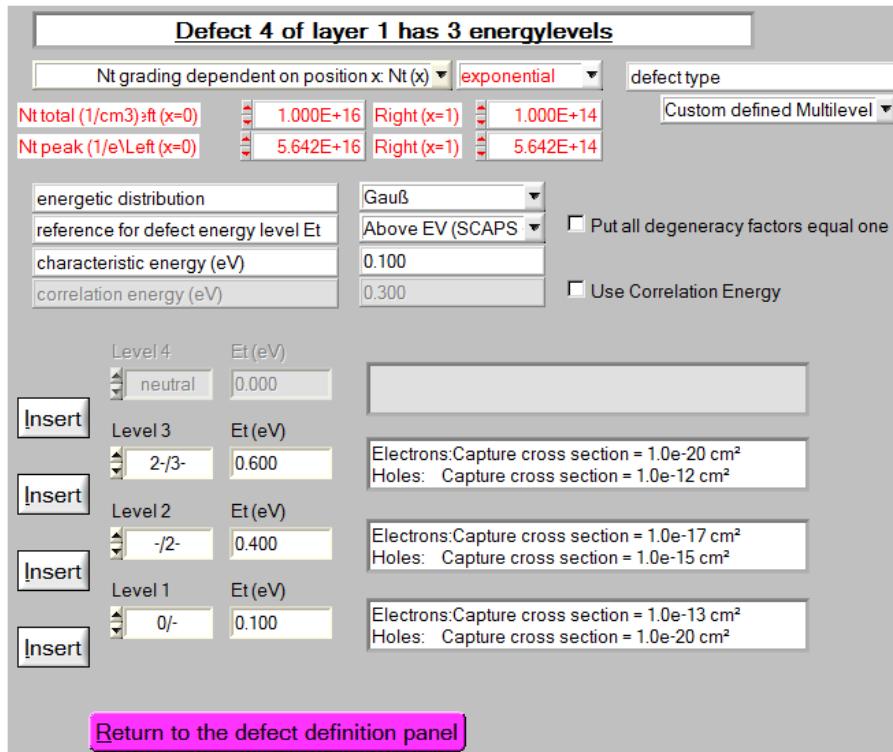
**Figure 3.21** The defect properties panel

### 3.6.2 Multivalent defects

The most common defects in semiconductors are either donor or acceptor defects having two possible charge states. However, there exist defects with more than two different charge states (multivalent defects). In SCAPS you can add up to four defect levels (= five charge states) for each defect. The charge present on a defect can be varied from -3 to +3 elementary charges. Because the most common multivalent defects are double donors (charge states 0, +1 and +2), double acceptors (charge states 0, -1 and -2) and amphoteric defects (charge states 0, +1 and -1), special facilities for them are provided in SCAPS. The charge states connected to a defect can be set by the 'defect type button'

If the defect is not multivalent (donor-acceptor-neutral), all defect parameters can be set on the defect density panel, otherwise clicking the defect type button will result in opening the 'Multiple level defects properties panel', Figure 3.22.

The algorithms used by SCAPS to calculate the recombination through defects are explained in [2].



**Figure 3.22** ‘Multiple level defects properties panel’

### 3.6.2.2 A neutral defect

The “**neutral**” defect is an idealization of a defect which contributes to the Shockley-Read-Hall recombination but does not contribute to the space charge. In the case of a “neutral” defect, only the product of  $\sigma$  and  $N_t$  affects the dc and ac solutions, through the carrier lifetimes;  $\tau_n$  e.g. is given by  $1/(\sigma_n N_t v_{th})$ : in this case, the defect centers contribute to recombination but not to the space charge. This type of defect can be chosen if one wants to specify electron and hole lifetimes without specifying a defect density which affects the space charge.

In other words: a neutral defect does not exist in reality, it is an idealization to help you to create a model step by step.

### 3.6.2.3 More than two charge states

When choosing the defect type as double donor, double acceptor, amphoteric or custom defined multilevel the user is redirected to the multiple level defects properties panel (see Figure 3.22). Some of the properties which were already available on the defect properties panel are available here as well, together with the properties which are specific for a multivalent defect. Starting from this panel the user can edit the defect and its levels.

Only when the defect type *custom defined multilevel* has been selected the user can add up to four different levels, otherwise the number of levels is restricted to two and the charge types are predefined. The ‘insert’ buttons are used to add a specific level. In order to remove a level the user has to click on a level with the right button of the mouse.

The charge type and the energy of a level can immediately be accessed on the *multiple level defects properties panel*. The charge types MUST be introduced in an ascending way (level 1 the most positive charge), or all levels must be neutral. In most cases the most positive level will be situated the closest to the conduction band, a warning is given with the option for the user to define the defect differently, but the user is not obliged to comply with the warning.

The other defect properties of a level can be edited by double-clicking on the text-box belonging to each level. This opens a panel similar to the defect density panel, however the properties which are not level specific but general to the entire defect are dimmed and can't be edited. The only properties which can be edited are the capture cross sections (for electrons and holes), the energy level and the IPV-properties (§3.6.4).

Several features of the energetic distribution can also be accessed on the multiple level defects properties panel. The *energetic distribution*, the *reference for defect energy levels* and the *characteristic energy* are the same for all levels of one defect. More info about these properties can be found in §3.6.3. If the checkbox *Use Correlation Energy* is checked, the energy difference between level 2 and level 1 is displayed in the window next to the checkbox. When the user edits this correlation energy the energy level of level 2 will be changed accordingly with respect to level 1.

The degeneracy factors (see [2]) take as a default value Eq. (9), with  $H$  the total number of charge states.

$$g_s = C_H^s = \frac{H!}{s!(H-s)!} \quad (9)$$

However, their values can be set to one as well by clicking the appropriate checkbox.

When the user has finished editing the properties of the levels the *multiple level defects properties panel* can be quitted by the *return to the defect definition panel*-button. Then the *defect properties panel* is again displayed, but the level specific properties are no longer visible. It is a kind of survey of the defect. Now the changed to the defect can be saved using the *add/accept*-button or ignored using the *cancel*-button.

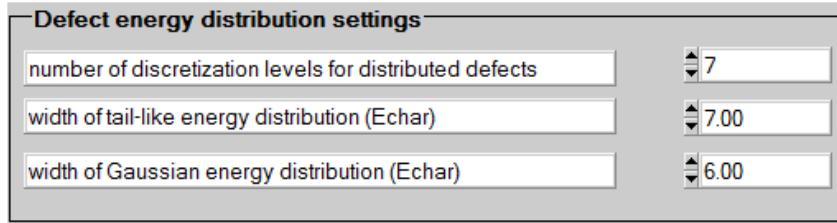
### 3.6.3 Energetic distribution of defect levels

Next to a discrete defect energy level, also defect whose levels display a distribution in the band gap can be modeled in SCAPS. There are five different options, listed in Table 3.2.

**Table 3.2** Overview of the energetic distributions of defects available.  $E_t$  is the energy level,  $E_c$  is the characteristic energy (not the conduction band edge energy here!). The defect density is given in  $\text{cm}^{-3}/\text{eV}$  (except for *single*, where it is given in  $\text{cm}^{-3}$ ).

	Range	$N_t(E) = \dots$	remarks
Single	$[E_t; E_t]$	$N_t \times \delta_{E_t}$	
Uniform	$\left[ E_t - \frac{E_c}{2}; E_t + \frac{E_c}{2} \right]$	$N_t$	
Gauss	$\left[ E_t - \frac{w_G}{2} E_c; E_t + \frac{w_G}{2} E_c \right]$	$N_t \times \exp\left[-\left(\frac{E - E_t}{E_c}\right)^2\right]$	$w_G$ can be defined on the numerical panel default value: $w_G = 6.0$ see Figure 3.23
CB tail	$[E_t - w_t E_c; E_t]$	$N_t \times \exp\left(\frac{E - E_t}{E_c}\right)$	$w_t$ can be defined on the numerical panel default value: $w_t = 7.0$ see Figure 3.23
VB tail	$[E_t; E_t + w_t E_c]$	$N_t \times \exp\left(-\frac{E - E_t}{E_c}\right)$	$w_t$ can be defined on the numerical panel default value: $w_t = 7.0$ see Figure 3.23

Whenever the defect energy distribution is not single, this distribution is discretized as being a specified number of single defect levels. This number can be set on the numerical panel, its default value is 7.



**Figure 3.23** Defect settings on the numerical panel with their default values. Top: the number of discretization levels, bottom:  $w_t$  and  $w_G$ .

The energy level has to be defined with respect to a reference energy level. This can be set on the defect density panel or on the multiple level defects properties panel. The available choices are: above  $E_V$  (above the valence band level); below  $E_C$  (below the conduction band level) and above  $E_i$  (above the intrinsic level

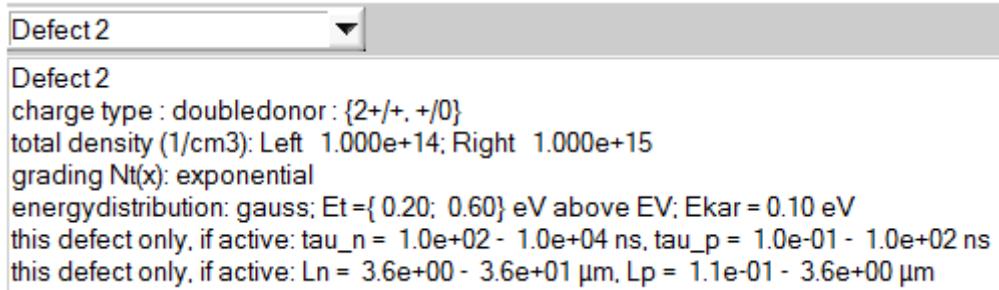
$$E_i = E_V + \frac{E_g}{2} + \frac{k_B T}{2} \ln\left(\frac{N_V}{N_C}\right).$$

### 3.6.4 Impurity photovoltaic effect (IPV)

SCAPS is able to simulate the IPV-effect. Its parameters can be set on the defect density panel. The algorithms and an example is given in [8, 9]. More examples are found in articles referring to this article, e.g. [10]. An example ‘\*.def’-file is provided with the SCAPS installation with ample comments.

### 3.6.5 Short overview of defect values

When all defect-parameters are set, an overview is given in the layer definition panel, Figure 3.24.



**Figure 3.24** Overview of the defect parameters

Next to the charge type, the defect density (and its spatial and energetic distribution) also an overview of the carrier lifetime and diffusion length is given. These are calculated (Eqs. (10) and (11)) only at the left and right side of the layer for every energy level. Only its maximum and minimum value are listed.

$$\tau = \frac{1}{\sigma v_{th} N_t} \quad (10)$$

$$L = \sqrt{D\tau} \quad (11)$$

**IT SHOULD BE STRONGLY EMPHASIZED THAT THESE VALUES ARE ONLY CALCULATED TO GIVE THE USER A ROUGH IDEA ABOUT THE RECOMBINATION OF THIS DEFECT, NONE OF THESE VALUES ARE USED IN THE ACTUAL SIMULATIONS! MORE ADVANCED ALGORITHMS ARE USED THERE, see e.g. [2].**

### 3.6.6 Radiative and Auger recombination

It is possible to introduce radiative and Auger (band-to-band) recombination in SCAPS, according to Eqs. (12) and (13).

$$U_{\text{radiative}} = K(np - n_i^2) \quad (12)$$

$$U_{\text{Auger}} = (c_n^A n + c_p^A p)(np - n_i^2) \quad (13)$$

The parameters  $K$ ,  $c_n^A$  and  $c_p^A$  can be set on the layer panel. SCAPS does not give a hint for a plausible  $K$  value  $\circledcirc$ ; for Si you can take  $K_{\text{Si}} = 1.8 \times 10^{-15} \text{ cm}^{-3}\text{s}^{-1}$ , for GaAs  $K_{\text{GaAs}} = 7.2 \times 10^{-10} \text{ cm}^{-3}\text{s}^{-1}$ . In CIGS the recombination is most probably dominated by defects, you can take  $K = 0$ , unless you want to simulate electroluminescence. In Si appropriate values for the Auger constants are:  $c_n^A \approx c_p^A \approx 3 \times 10^{-31} \text{ cm}^{-6}\text{s}^{-1}$ .

Band to band recombination			
Radiative recombination coefficient ( $\text{cm}^3/\text{s}$ )	1.000E-5	1.000E-5	uniform
Auger electron capture coefficient ( $\text{cm}^6/\text{s}$ )	1.000E-25	1.000E-25	uniform
Auger hole capture coefficient ( $\text{cm}^6/\text{s}$ )	1.000E-25	1.000E-25	uniform

**Figure 3.25** Band to band recombination parameters. From top to bottom:  $K$ ,  $c_n^A$  and  $c_p^A$ .

The contributions of the different recombination processes to the current can be assessed on the IV-panel (see §6.4.3) and in the recorder facility (see Chapter 8). When you are sure that all direct band-to-band recombination is radiative, and that all emitted photons can get out of the cell to your measurement set-up, you can interpret  $J_{\text{radiative}}$  as the electroluminescence signal. However it is not labeled as such in SCAPS.

## 3.7 Metastable defect transitions

In the chalcopyrite material system often metastable behaviour of the samples is observed. This is possibly due to the presence of defects which undergo transitions which are accompanied by large lattice relaxations and thermal activation over energy barriers. SCAPS is able to simulate this behaviour, the algorithms which are implemented and an example can be found in [11]. An introduction to metastable defects inSCAPS was given in [11'].

### 3.7.1 Principles

Metastable effects take place on a long timescale (which can vary a lot, depending on the actual conditions). SCAPS does not solve transient problems. Hence, the following philosophy has been followed.

A metastable defect is considered to exist of two different configurations, a donor and an acceptor configuration, which are separated by energy barriers associated with lattice relaxations. Each separate configuration behaves as a conventional defect, which can have a multivalent character. The transition between the different configurations requires a simultaneous capture or emission of two free carriers together with the thermal activation of an energy barrier. Hence, at lower temperatures, this transition will not occur. Similarly the transition will not be observed in an admittance measurement when the change of the quasi-Fermi levels occurs at higher frequencies.

In a typical experiment, the studied sample is brought to a well defined metastable state, i.e. a distribution over the donor and acceptor configuration, by applying certain initial voltage/illumination condition at elevated temperatures for a sufficiently long time. Afterwards the sample is cooled down in order to inhibit further changes in the metastable state of the sample and measurements are performed.

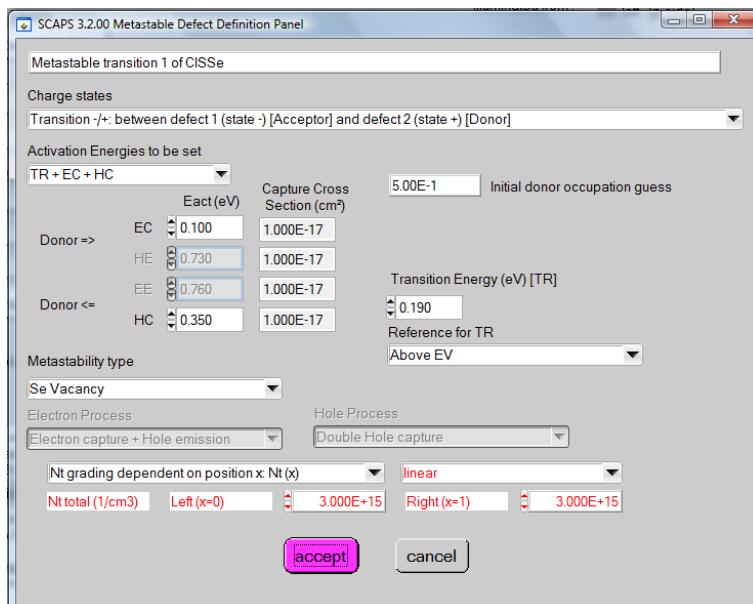
In order to simulate this, first the distribution over the acceptor and donor configuration of the defect is calculated under initial voltage/illumination conditions, see §4.3. This distribution, together with the total defect density is used to calculate the defect density belonging to the different configurations of the metastable defect. These defect densities are kept constant and are used to perform further simulations under measurement voltage/illumination conditions.

### 3.7.2 Introduction of a metastable transition

A metastable defect is assumed to have two different configurations (acceptor and donor). Each of these configurations has to be introduced as if it were a regular (multivalent) defect. Metastable properties can only be set for defects with single energy levels, not for defects that occupy a band of energies. If at least two defects are present in a layer which have compatible charge states, they can be set to be the two configurations of a metastable defect by checking one of the ‘metastable transition’-boxes on the layer panel, Figure 3.26. By clicking the ‘edit’-button, the ‘Metastable Defect Definition Panel’ (Figure 3.27) opens, where all parameters governing the metastable transition between the two configurations can be set/edited.



**Figure 3.26** Allowing a metastable transition.



**Figure 3.27** The Metastable Defect Definition Panel

Its main components are discussed below:

**Charge states:** This is a list which gives all possible combinations of charge states available which can be transformed into one another through a double capture/emission process. If none are available, you will not be able to access the Metastable Defect Definition Panel. If more than one option is available, you can select the one you want.

**Activation Energies:** The transition between the two configurations of the metastable defect can proceed through four processes: electron capture (EC), hole emission (HE), electron emission (EE) and hole capture (HC). Each of these processes has its own activation energy which can be set. The Fermi level position where the two configurations are equally stable under thermal equilibrium conditions is called the transition energy (TR) and can also be set. These four activation energies together with the transition energy are not independent as they have to obey to the principle of detailed balance. Only three of them can be chosen independently, the two remaining are then calculated by the program. You can decide which energies you want to provide in the ‘Activation Energies to be set’-list. When a band gap grading is present, the calculated energies can vary throughout the layer. The values which are displayed in the user interface are then those which are valid for the right side of the layer. The transition energy can be set with respect to the valence band, conduction band or intrinsic energy level in a similar way as a regular defect level.

**Metastability type and electron/hole processes:** The double capture and emission processes can be obtained in different ways. The electron capture process e.g. can consist out of two simultaneous single

electron capture processes (*Double electron capture*) or a single electron capture together with a simultaneous hole emission process (*Electron capture + Hole emission*). This has an influence on the details of the detailed balance calculation and needs to be set. These properties can also automatically be set to the standard value for the common  $V_{Se}$ -complex and  $In_{Cu}$ -complex in the CIGS material by selecting respectively ‘*Se Vacancy*’ or ‘*DX centre*’ in the *Metastability type-list*.

**Defect density:** The defect density of the metastable defect can be set in a similar way as for a regular defect. Pay attention! When a defect is set to be one of the configurations of a metastable defect density, the defect density which you can set on the *defect properties panel* will be ignored. Instead at every meshpoint the relevant defect density of this configuration will be calculated from the total defect density of the metastable defect.

**Initial donor occupation guess:** In order to start the iteration process an initial guess has to be made for the fraction of the metastable states that are in the donor configuration. This fraction is the same for all meshpoints.

### 3.7.3 Help, the buttons to introduce/edit metastable properties are not available!

One of the aims of SCAPS is to keep the user interface very intuitive and easy to learn. Since the first release, a multitude of extra facilities have been added to the program. These extensions increase the number of situations which can be simulated, but also complicate the learning process of a user making his/her first simulations.

The simulation of metastable defects is considered to be ‘advanced’ and should not be one of the worries of a rookie in the bright world of numerical simulations with SCAPS. As a result, most of the buttons governing metastabilities are only accessible/visible when the user really expresses her/his intention to use them:

- The ‘Metastable Defect Definition Panel’ can only be accessed if at least two defects are present in the layer AND at least one of the charge states of one defect can be transformed into one of the charge states of a second defect through a double capture process.
- The ‘Initial State Workingpoint Panel’ can only be accessed if at least one metastable defect is present in the definition file which is currently loaded.

### 3.7.4 Numerical settings

The calculation of the occupation of the different metastable configurations is done in an iterative way. The maximum number of iterations and the minimum remaining error can be set on the numerical panel, see Figure 3.28.

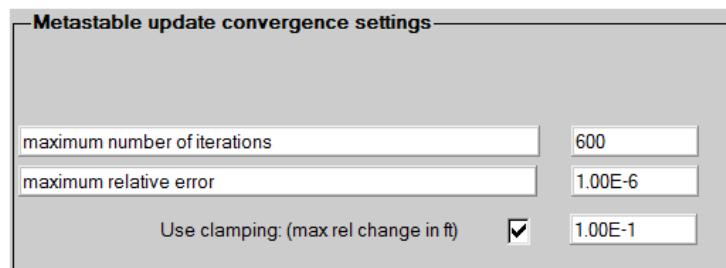


Figure 3.28 Numerical settings for metastable simulations

When the calculation is numerically unstable it can be stabilized by using a clamping factor. This process is described in [11]. This factor should be chosen between 0 and 1, 1 corresponds to no clamping. The smaller this factor the more stable, but also the slower the calculation gets. Values below 0.001 are usually very slow.

### 3.8 Interfaces

Between any two semiconductor layers an interface can be defined. The main algorithms used in SCAPS are discussed in [12].

The model which is implemented for interface transport in SCAPS is thermionic emission. The thermal velocity of the interface transport equals the smallest thermal velocity of the two neighboring layers. The use of this model implies that there will always be a (small) step in the quasi-Fermi level energy values at an interface, even when there are no band offsets.

Just as in a bulk layer recombination centers can be present at an interface. The definition of interface defects is very similar to bulk defects. However, there are only three possible defects and their charge type cannot be multivalent. Recombination in interface states is modeled by the Pauwels-Vanhoutte theory [13], which is an extension of the Shockley-Read-Hall theory.

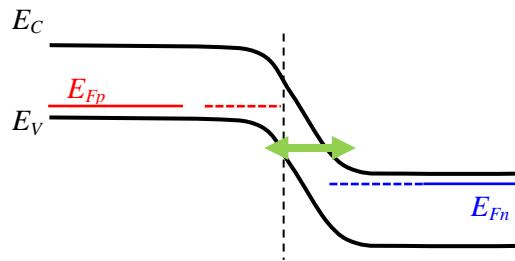
Two tunneling processes are implemented which involve interfaces: intraband tunneling and tunneling to interface defects, see §3.9.

### 3.9 Tunnelling

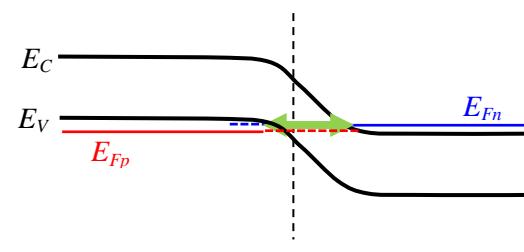
SCAPS treats (some) tunnelling mechanisms in a semiclassical way: band to band tunnelling, intraband tunnelling, tunnelling to interface defects and tunnelling to contacts. The main algorithms are discussed in [4], a more elaborate version is available in [14] (in dutch, on request). PAY ATTENTION: Tunnelling is only taken into account in the solution of the dc-problem, not in the ac-problem. Hence only indirect tunnel influences on the admittance (through the setting of the dc-state of the sample) are simulated!

Band-to-band tunnelling is the tunnelling between conduction and valence band states, see Figure 3.29. Intraband tunnelling is the tunnelling between states both in the same (conduction/valence) band, see Figure 3.30. The valence and conduction band consists in (almost) every structure out of bulk layers and interfaces. Band-to-band tunnelling and intraband tunnelling can be allowed/forbidden for every layer/interface separately by checking the appropriate checkbox on the layer/interface panel.

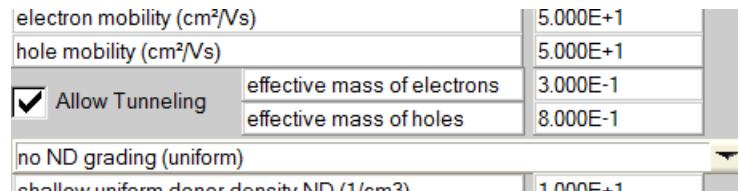
Next to band-to-band and intraband tunnelling, also tunnelling between the bands and an interface defect, or the bands and a metal contact is possible, see Figure 3.31. These processes can be set for each interface defect separately on the interface defect properties panel and for each contact on the contact definition panel. Whether or not these tunnelling processes are allowed does not depend on the layer/interface settings. E.g. if tunnelling to the left contact is allowed on the contact properties panel, tunnelling between this contact and all layer/interfaces will be allowed even if the ‘allow tunnelling’-checkbox is unchecked in one or more layers/interfaces.



Band-to-band tunneling at reverse bias

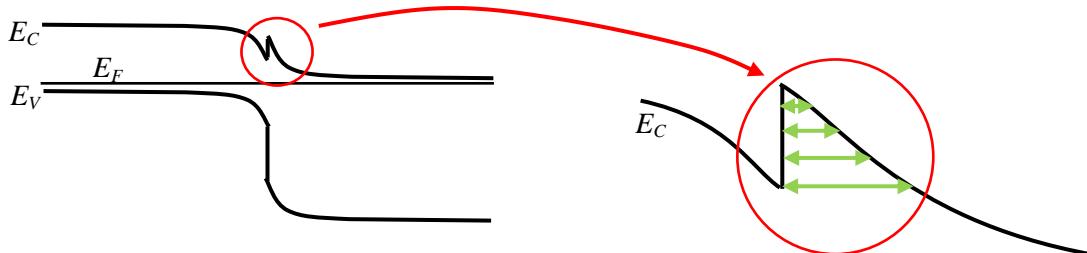


Band-to-band tunneling at forward bias

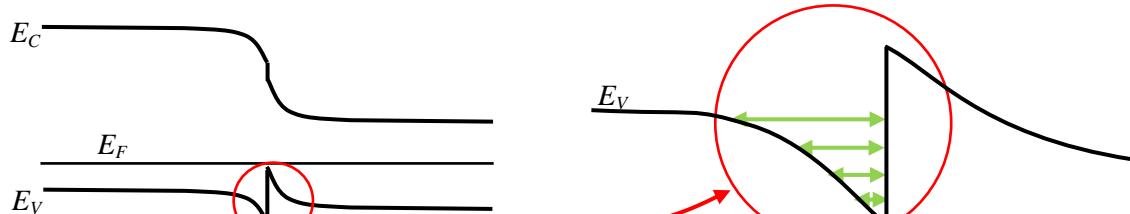


Relevant part of the Layer Properties

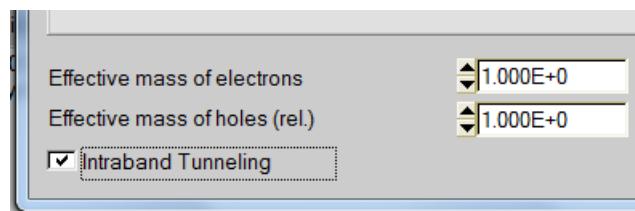
**Figure 3.29** Band-to-band tunnel mechanisms at reverse and at forward bias. Note that band-to-band tunneling at forward bias (Esaki tunnel diode) is only possible with very heavily doped, degenerated semiconductors, where the Fermi level is inside the conduction and/or valence band; these heavy doping phenomena (Fermi statistics, band gap narrowing...) are not implemented in SCAPS. At the bottom, the check box and  $m_{\text{eff}}^*$  values to be enabled in the Layers Properties Panel are shown.



intra-band tunneling in the conduction band

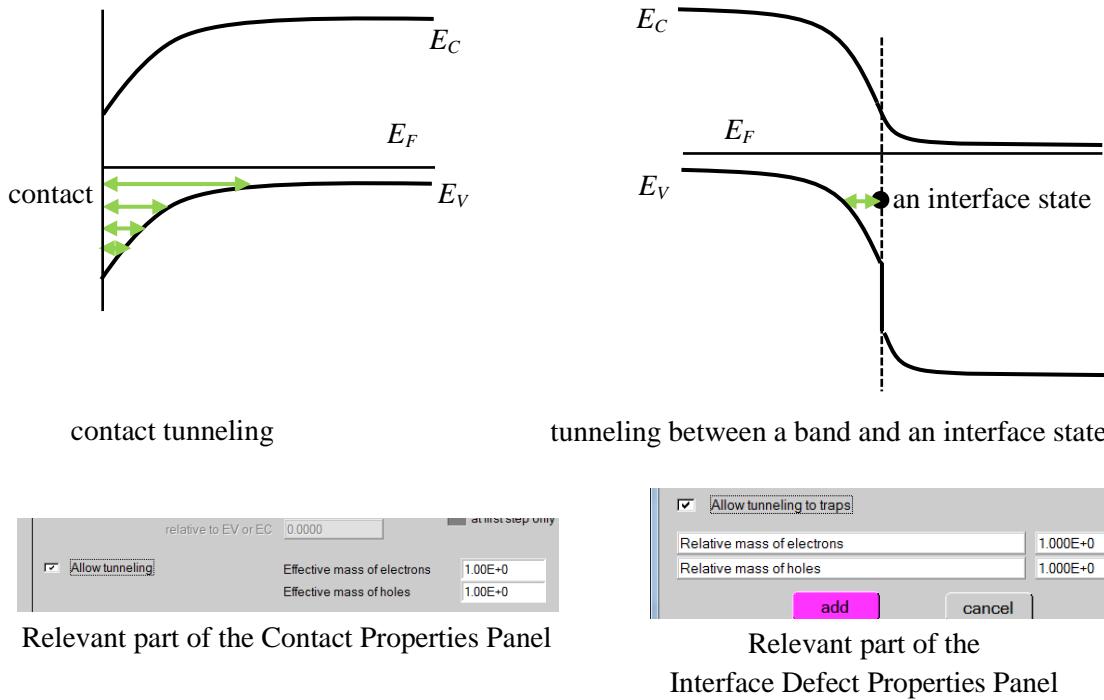


intra-band tunneling in the valence band



Relevant part of the Interface Panel

**Figure 3.30** Intra-band tunneling in the conduction and in the valence band. At the bottom, the check box and  $m_{\text{eff}}^*$  values to be enabled in the Interface Panel are shown.



**Figure 3.31** Contact tunneling and tunneling to interface states: implemented in SCAPS. At the bottom, the check box and  $m_{\text{eff}}^*$  values to be enabled in the Contact Properties Panel and the Interface Defect Properties Panel are shown.

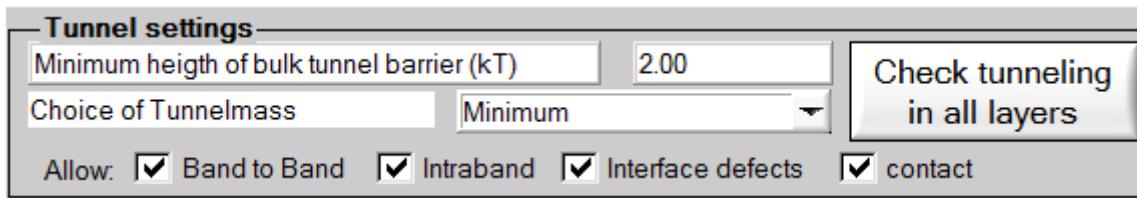
### 3.9.2 Numerical tunnel settings

The tunnel probability is determined by the effective mass of the two states which are involved in the process. These masses can be different, and can be set differently in SCAPS. You can select on the numerical panel, which of the two masses to use. The default choice is to use the minimum effective mass.

SCAPS uses the WKB-approximation in order to calculate tunnel probabilities. This approximation is only strictly valid for large enough barriers. When no precautions are taken, the tunnel rate for very thin and shallow barriers is hugely overestimated. In order to avoid this you can set a minimum height of the tunnel barrier (default =  $2k_B T$ ), for such small barriers the current is determined by drift and diffusion rather than tunnelling anyway.

Up to now four different tunnel processes are present in SCAPS. Each of them can easily be forbidden/allowed by (un)checking the appropriate box on the numerical settings panel.

As it is often inconvenient to check all *allow tunneling* boxes on the different layer and interface panels, this can automatically be performed by clicking the *check tunneling in all layers* button on the numerical panel; it is still advised to check afterwards which tunnel mechanisms are enabled/disabled in the various panels.



**Figure 3.32** Tunnelling settings on the numerical panel

### 3.10The blue button

In SCAPS 2.8.4 and earlier, it happened that a panel requiring input was hidden under other panels, and difficult to find; it could also happen that such panel disappeared, which caused a hang-up of the programme. These problems have been fixed now. There are some limitations to the number and kind of panels that can be opened or operated simultaneously. Above all, the F1 key acts as an emergency button that brings the cell definition panel on top; on that panel, you find a new blue button Other panels need input first! that displays a nicely structured list of all input panels currently open, and that allows you to navigate between them (or to quit all input at once). Pressing the F2 key redirects you to the current layer or interface panel.

### 3.11Saving and loading problem definitions

Almost every setting in SCAPS can be saved to a file and loaded again. These files are standard ASCII-files which can be read and edited with e.g. *Notepad*. However, editing is at own risk and highly disadvised. An overview of the main file-types with their default extension and directory are given in Table 3.3.

**Table 3.3** Main file-types with their default extension and directory

SCAPS filetypes	default extension	default directory
problem definition	.def	scaps\def
material properties	.materials	scaps\materials
action list	.act	scaps\def
initial workingpoint	.wp2	SCAPS\def
batch settings	.sbf	scaps\bdf
recorder settings	.srf	scaps\bdf
scaling	.scl	scaps\scaling
SCAPS scripts	.script	scaps\script
all SCAPS settings in one	.scaps	scaps\def

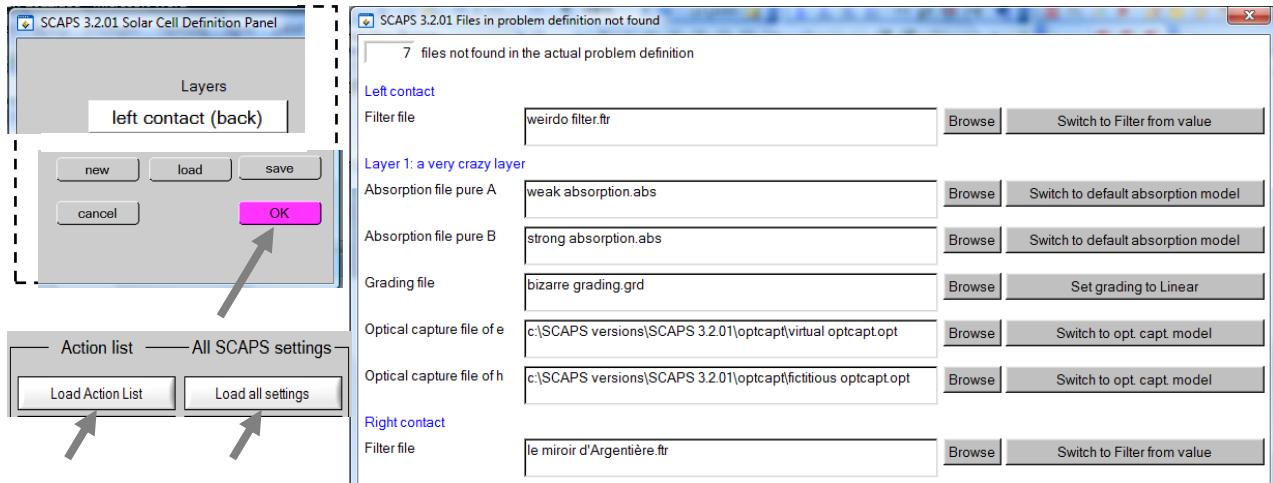
Other files can not be edited within SCAPS but are needed as input files. These files are text files (ASCII files) that you can construct and edit with a simple text processor, e.g. *Notepad*, see Table 3.4. Each file has a header that can be used to explain what the data represent and where they are coming from. The files distributed with SCAPS can serve as an example to construct ones own input files.

**Table 3.4** File-types of SCAPS input data files

SCAPS filetypes	default extension	default directory
spectrum $\Phi(\lambda)$	.spe	scaps\spectrum
generation $G(x)$	.gen	scaps\generation
absorption $\alpha(\lambda)$	.abs	scaps\absorption
filter $T(\lambda)$ or $R(\lambda)$	.ftr	scaps\filter
optical capture cross section $\sigma_n(\lambda), \sigma_p(\lambda)$	.opt	scaps\optcapt
grading $y(x)$	.grd	scaps\grading

When definition files are exchanged between users (or between your lab computer and your home computer), it can happen that a .def file contains an input file name (.abs, .ftr, .opt, .grd, ...) that is not

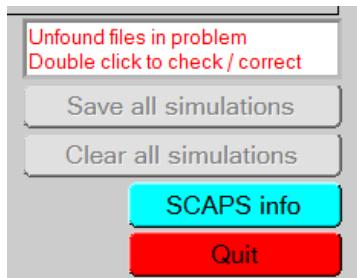
present on the computer. When leaving the solar cell definition panel with the OK button, SCAPS checks if all files are found, and if not brings you to a panel where you can either browse for an existing filename, or set the property to the default internal model instead of ‘from file input’.



**Figure 3.33** SCAPS checks if all necessary files can be found: after loading an action list or loading all SCAPS settings, and after leaving the Solar Cell Definition Panel with ‘OK’ (see arrows in screen shots at the left side). Then, a panel opens that displays a list of all files not found in the problem definition (see screen shot at the right side). You can either browse for an existing file, or set the property ‘to model’.

Please note that such check is not carried out when the problem has been input or changed by the SCAPS the batch set-up: you still have the possibility to make the program to crash by inputting non-existing files. Also, when SCAPS is in non-interactive mode (see 7.4 and Figure 7.5), the Check Files Panel is not opened, but everywhere a file is not found, the corresponding property is set ‘to model’ (this is equivalent with clicking all the ‘Set to model’ buttons, at the right side of Figure 3.33); a message of what was unfound and has been set to model is written to the SCAPSErrorLogFile.log file.

You can leave the Check files Panel without fixing all unfound files (that is, either browsing for an existing file, or setting the property to model). The calculate button is then disabled, and you see a red warning at the bottom, right of the action panel (Figure 3.34).



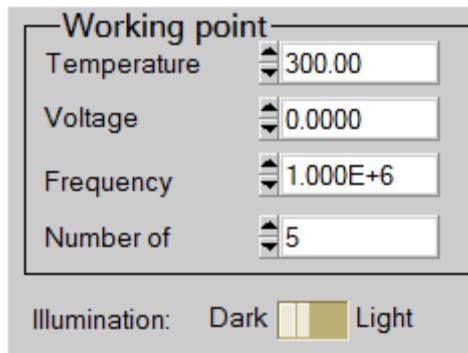
**Figure 3.34** When not all unfound files are fixed, a red warning is displayed at the bottom/right of the action panel.

# Chapter 4: Working point definition

The workingpoint specifies the parameters which are not varied in a measurement simulation, and which are relevant to that measurement.

## 4.1 General

These parameters are (immediately) available on the action panel (Figure 4.1).



**Figure 4.1** Setting the working point conditions, available on the action panel.

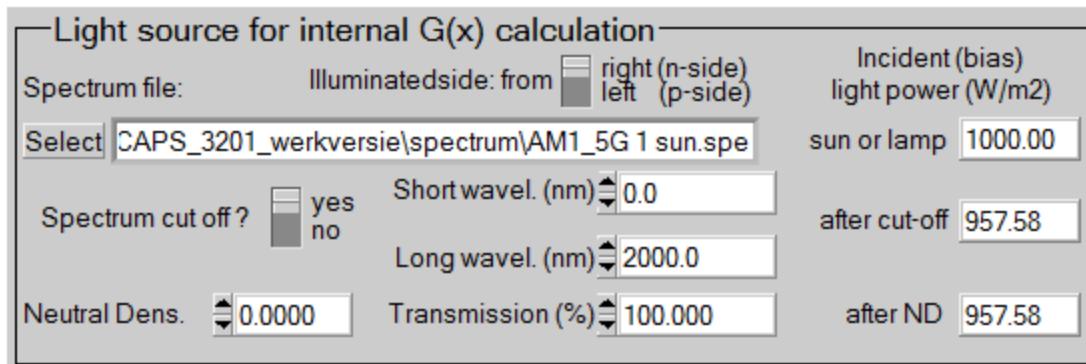
- The temperature  $T$ : relevant for all measurements. Note: in SCAPS, only  $N_C(T)$ ,  $N_V(T)$ , the thermal velocities, the thermal voltage  $k_B T$  and all their derivatives are the only variables which have an explicit temperature dependence; you must input for each  $T$  the corresponding materials parameters yourself.
- The voltage  $V$ : is discarded in  $I$ - $V$  and  $C$ - $V$  simulation. It is the dc-bias voltage in  $C$ - $f$  simulation and in  $QE(\lambda)$  simulation. SCAPS always starts at 0 V, and proceeds to the working point voltage in a number of steps that you also should specify, see §5.1.2.
- The frequency  $f$ : is discarded in  $I$ - $V$ ,  $QE(\lambda)$  and  $C$ - $f$  simulation. It is the frequency at which the  $C$ - $V$  measurement is simulated.
- The illumination: is used for all measurements and is discussed below in §4.2.

## 4.2 Illumination conditions

When performing simulations under illumination, you can further specify the illumination conditions. The basis settings are: dark or light, choice of the illuminated side, choice of the spectrum. If you have an optical simulator at your disposal you can immediately load a generation profile as well in stead of using a spectrum.

### 4.2.1 Internal SCAPS calculation

SCAPS is able to perform simple optical simulations in order to calculate the generation in your structure. Hereto the illumination conditions have to be specified (Figure 4.2).

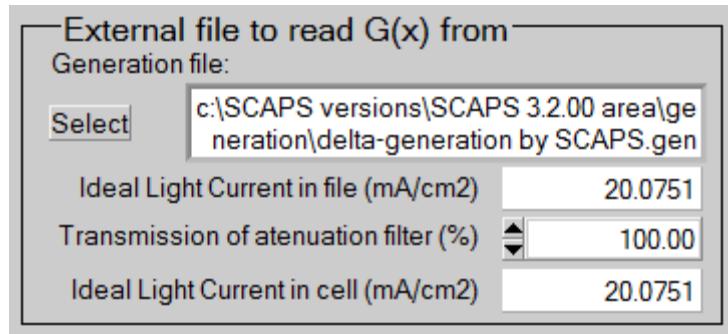


**Figure 4.2** Specifying the illumination conditions.

- The illumination side can be changed (left/right)
- A light spectrum file can be selected. Several spectrum files are available in the SCAPS distribution. The default is a one sun ( $= 1000 \text{ W/m}^2 = 100 \text{ mW/cm}^2$ ) illumination with the ‘air mass 1.5, global’ spectrum. A lot of monochromatic spectra are available as well. They come in two versions. They either contain a total illumination power of  $1000 \text{ W/m}^2$  (1 sun) in the wavelength band (e.g. ‘600 nm.spe’), or they contain a number of photons that would yield a light current  $J_L = 20 \text{ mA/cm}^2$  if  $QE$  were unity (e.g. ‘600nm fixed JL.spe’). All spectrum files which come with the SCAPS distribution are documented with comments. All spectrum files are ASCII-files and can hence be user defined. The first line contains the number of wavelengths that appear in the spectrum file. This line is immediately followed by two columns. The first column specifies the wavelength  $\lambda$  (nm). Suppose the spectral density (in Watt/m<sup>2</sup>.nm) is given by  $P$ : the incident power in the wavelength interval  $[\lambda - d\lambda/2, \lambda + d\lambda/2]$  is then given by  $Pd\lambda$  (in Watt/m<sup>2</sup>). The second column in the spectrum file contains this incident power  $Pd\lambda$  (in Watt/m<sup>2</sup>) in a wavelength interval  $d\lambda$  around  $\lambda$ . As such, the total incident power is given by the sum of the values in the second column. The wavelength interval  $d\lambda$  considered around a wavelength  $\lambda_i$  is:  $d\lambda = (\lambda_{i+1} - \lambda_{i-1})/2$ . For the first wavelength  $\lambda_1$  it is  $d\lambda = \lambda_2 - \lambda_1$ , and for the last wavelength  $\lambda_N$  it is  $d\lambda = \lambda_N - \lambda_{N-1}$ . The maximum number of wavelengths appearing in the spectrum file equals 2500. The number of incident photons, per m<sup>2</sup> and per s, of wavelength between  $\lambda - d\lambda/2$  and  $\lambda + d\lambda/2$  is then set equal to  $5.035 \times 10^{15} \times \lambda \times Pd\lambda$ ,  $\lambda$  in nm,  $Pd\lambda$  in Watt/m<sup>2</sup>. The numerical factor equals  $10^{-9}/hc$  where Planck’s constant  $h$  and the light velocity  $c$  are expressed in SI units. For convenience, the spectrum file may be preceded by comment lines, which begin with the character “>” or “/”. These lines are not read by SCAPS. They can be used to identify the source or the purpose of a given spectrum file.
- The incident power of the incoming light can be weakened or amplified by applying a neutral density filter. The value  $ND$  of this filter can immediately be set, or the transmission of the filter  $T = 100\% \times 10^{-ND}$ . You can set  $ND < 0$  or  $T > 100\%$ , to do simulations under concentrated light. The incident power can also be attenuated by an internal/reflection at the front contact which should be specified on the cell definition panel.
- No reflections at the interfaces are implemented. Only at the contacts you can specify a (wavelength dependent) reflection/transmission (§3.2).
- A short/long wavelength cut-off filter can be applied.

#### 4.2.2 Generation from file

There is also a possibility to input the generation in the structure immediately. This can be useful if you have a better optical simulator than SCAPS, or if you want to mimic e.g. EBIC measurements.



**Figure 4.3** Settings when using a generation file.

The advised way to create a  $g(x)$  file is as follows: set up a problem in SCAPS; perform a calculation with the internally calculated  $g(x)$ . Save the results in the EB-panel (energy bands panel), or, alternatively, press show in the action panel and copy/paste the desired data in a user file (e.g. Origin or Excel). Then you have the  $x$ -coordinates of the nodes used by SCAPS. Make a generation file with the  $g(x)$  data from your own model, calculation or measurement... For an example of a valid format, see the examples coming with SCAPS (file extension.gen). Click Generation of eh pairs determined from file in the action panel, and select your file. When the  $g(x)$  data in your file do not use the  $x$  values used by SCAPS, the program will have to interpolate between your  $x$  values, and this might cause inaccuracy. To judge the interpolation, you can visualize the  $g(x)$  data in the *Generation-Recombination profiles panel*, see §6.4.2. The  $g(x)$  profile you will see then is, however, only updated after a calculation has been performed with the calculated or user specified  $g(x)$ .

Additionally you can still use an attenuation filter, which would mimic the effect of a neutral density filter. Again, setting an attenuation  $> 100 \%$  corresponds to a measurement under concentrated light. The ideal light current in the file (Figure 4.3) is the total number of  $eh$  pairs in the generation file, thus integral of all  $g$  values over the cell thickness, and divided through the elementary charge  $q$ . The ideal light current in the cell is the ideal light current in the file, multiplied with the transmission of the attenuation filter.

A generation file is a table of  $x, g$  values ( $x$  in  $\mu\text{m}$ ,  $g$  in  $\text{m}^{-3}\text{s}^{-1}$ ). It contains information about the number of  $eh$  pairs generated, but not about the energy of the photons that created the  $eh$  pairs. As a consequence, no solar cell efficiency can be calculated. However, a ‘collection efficiency’, that is the ratio of the short circuit current to the ideal light current in the file, can be calculated, and is displayed in the IV panel. Also, a relevant spectral response  $QE$  cannot be calculated, since the generation file does not contain any information on wavelength.

### 4.3 The initial working point

In order to perform simulations for metastable defects (§3.7) two (different) working points are needed. The working point which is present since the birth of SCAPS still corresponds to the working point which is applied during the ‘measurement’. A second working point ‘the initial working point’ corresponds to the conditions which are applied to the structure in order to bring it in the desired metastable state. For example if you want to simulate a red-on-bias experiment, this working point will consist of a negative voltage bias and red illumination.

Pay attention! The temperature at this initial working point is usually high as the metastable transitions are thermally activated. Many metastabilities can not be induced at low temperature because the transition is too slow and you never get to an equilibrium situation in a reasonable time span. SCAPS always calculates equilibrium conditions, and waits in fact until an infinite amount of time has passed. You should take this into account when setting a low initial working point temperature .

The ‘Initial State Workpoint Panel’, where you can set the initial working point can be accessed from the action panel only if at least one metastable defect is present in the problem definition.

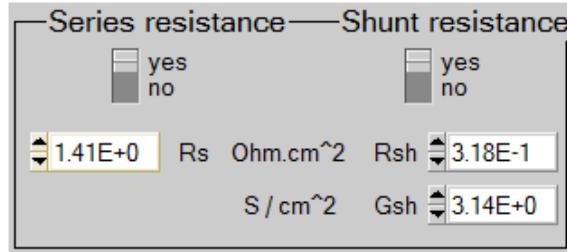
**Metastability: initial workpoint**

**Figure 4.4** Accessing the initial workpoint panel: press this button on the action panel.

You can save and load all settings of the initial state workpoint into and from a file. These files have extension .wp2 (from ‘second work point’), and reside in the \def folder.

#### 4.4 Shunt conductance and series resistance

It is possible to introduce an external shunt conductance and series resistance to the structure on the action panel, see Figure 4.5.



**Figure 4.5** Introducing external shunt conductance and series resistance

Both resistances can be switched on/off and for a shunt conductance you can either define its resistance or conductance.

These external resistances only affect *I-V*, *C-V* and *C-f* simulations (no *QE* simulations). The voltage which you apply to the sample (either the working point voltage or the voltage in the *iv* or *C-V* simulation) is interpreted as the internal voltage over the sample, without external resistance. This voltage gets then recalculated after simulation in order to get the external applied voltage in *I-V* and *C-V* simulations. In small signal simulations, the value of *G* and *C* are also recalculated after simulations.

The values of the external resistances are saved in the definition file of the solar cell structure.

# Chapter 5: Single shot calculations

The main functionality of SCAPS is to solve the one-dimensional semiconductor equations. In the bulk of the layers these equations are given by (14) together with the constitutive equations (15).

$$\begin{cases} \frac{\partial}{\partial x} \left( \varepsilon_0 \varepsilon \frac{\partial \Psi}{\partial x} \right) = -q \left( p - n + N_D^+ - N_A^- + \frac{\rho_{def}}{q} \right) \\ -\frac{\partial J_n}{\partial x} - U_n + G = \frac{\partial n}{\partial t} \\ -\frac{\partial J_p}{\partial x} - U_p + G = \frac{\partial p}{\partial t} \end{cases} \quad (14)$$

$$\begin{cases} J_n = -\frac{\mu_n n}{q} \frac{\partial E_{Fn}}{\partial x} \\ J_p = +\frac{\mu_p p}{q} \frac{\partial E_{Fp}}{\partial x} \end{cases} \quad (15)$$

Together with appropriate boundary conditions at the interfaces and contacts, this results in a system of coupled differential equations in  $(\Psi, n, p)$  or  $(\Psi, E_{Fn}, E_{Fp})$ . SCAPS numerically calculates a steady state and a small signal solution of this system. Hereto, the structure is first discretized (creating a mesh). A steady state workingpoint situation (see §Chapter 4) is calculated and when required a small signal analysis is performed.

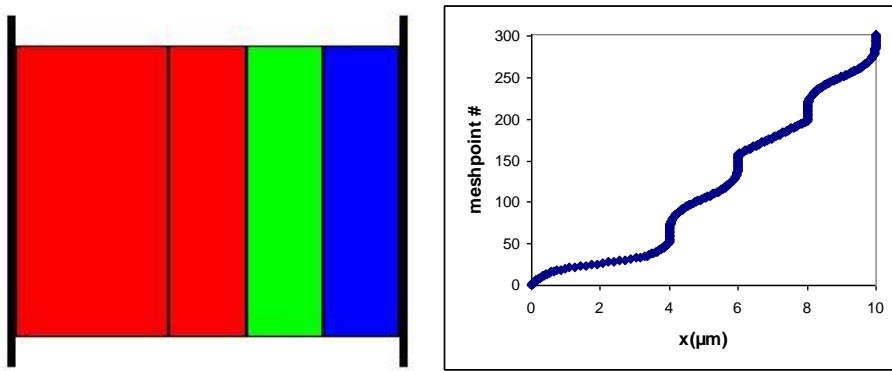
## 5.1 Calculation roadmap

### 5.1.1 Meshing

The first step in every calculation is to discretize the structure. The meshing algorithm of SCAPS provides:

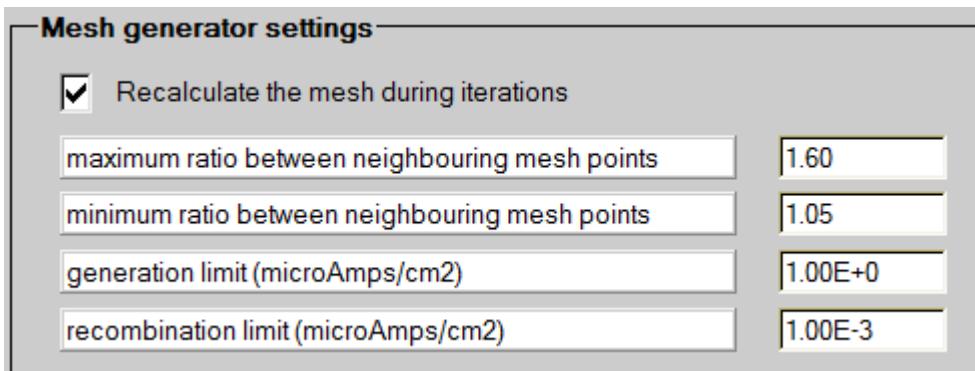
- Coarse meshing in the middle of a layer
- Finer meshing near the interfaces and contacts
- Two discretization points (with identical spatial coordinate) for each interface
- The mesh can be optimized during calculation.

The result of the discretization algorithm is illustrated in Figure 5.1.



**Figure 5.1** Result (*right*) of the discretization of the structure on the *left*. (Red: *p*-type, blue: *n*-type, green: intrinsic)

The basic algorithm is designed to provide a lot of points in regions where properties experience large variations (close to interfaces/contacts) and fewer points where the properties are expected to remain fairly constant (in the bulk). However, when performing simulations with strong gradings, with multivalent defects or with the IPV-effect this meshing procedure might be unsufficient. Hence, it is possible to optimize the mesh at every iteration step. This option can be set on the numerical panel as shown in Figure 5.2.

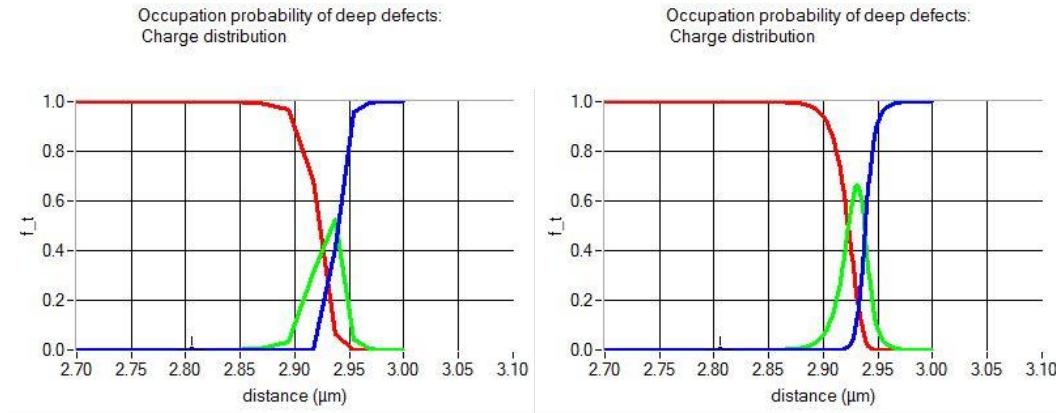


**Figure 5.2** Recalculate mesh settings on the numerical panel

When *recalculate mesh* is switched on at each iteration the ratio between the following properties at two adjacent meshpoints are assessed:  $\exp(q\Phi/kT)$ ,  $\exp(E_{Fn}/kT)$ ,  $\exp(E_{Fp}/kT)$  (all dimensionless), the recombination rate  $R$  and the generation rate  $G$  (both  $R$  and  $G$  in  $/m^3$ ). When that ratio is larger than the maximum ratio  $f_{max}$  set on the numerical panel, extra meshpoints will be inserted to enable a better simulation. If the ratio is smaller than the minimum ratio  $f_{min}$  set on the numerical panel, the meshpoint is removed. Precautions are taken not to remove too many meshpoints in a row. The ratio of the recombination and generation rate are only taken into account if  $q|R| < R_{limit}/L$  or  $qG < G_{limit}/L$  respectively. Here  $R_{limit}$  and  $G_{limit}$  are parameters set on the numerical panel (atypically for SCAPS, these parameters are displayed and saved in units of  $\mu\text{A}/\text{cm}^2$ ), and  $L$  is the total cell thickness. At roomtemperature, the default value  $f_{max} = 1.60$  corresponds to a difference of about 12 mV or 12 meV of  $\Phi$ ,  $E_{Fn}$  or  $E_{Fp}$  between two adjacent mesh points, whereas the default value  $f_{min} = 1.05$  corresponds to a difference of 1.2 mV or meV.

The *recalculate mesh* settings are all saved in the definition file.

The effect of the recalculation of the mesh is illustrated in Figure 5.3, where the defect occupation of an amphoteric defect is calculated.

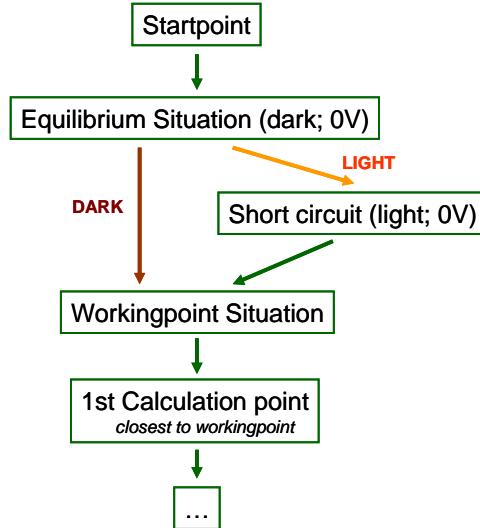


**Figure 5.3** Occupation of an amphoteric defect calculated without (*left*) and with (*right*) mesh recalculation

### 5.1.2 The pathway to a solution

When clicking the ‘Calculate: single shot’-button, the measurements you selected are calculated for the given structure. Hereto the system of (14) is solved numerically, using a Gummel iteration scheme with Newton-Raphson substeps [12, 15]. The parameters of this scheme are available on the numerical panel as convergence settings. As a rule of thumb, the best way to improve convergence is first to increase the number of iteration steps and then perhaps increase the termination criterion thresholds.

In order to get to a solution, an initial guess has to be made. Care should however be taken, because when this initial guess differs too much from the solution of the system, the iteration procedure will fail to converge (after a reasonable number of iterations steps). SCAPS does not ask an initial guess from the user, but calculates several situations in order to get to a solution under the working point circumstances and at the first measurement point. The strategy which is followed is illustrated in Figure 5.4.



**Figure 5.4** Getting to the working point and first calculation point...

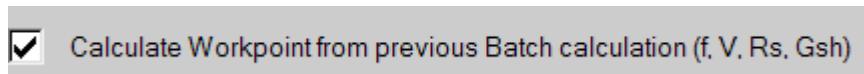
Every calculation starts at the startpoint. This is a very simple situation, assuming the quasi-Fermi levels to be zero throughout the structure and no potential drop to be present over the structure. This is used as an initial guess to get to the equilibrium situation (no illumination, no voltage applied). When the working point conditions are under darkness, this equilibrium condition is used as an initial guess to calculate the solution under working point conditions. When illumination is switched on however, the short circuit situation is calculated in an intermediate step to serve as the next initial guess.

When the difference between two situations is too large, the iteration procedure will diverge. Therefore, SCAPS implicitly introduces intermediate situations between the equilibrium/working point; equilibrium/short circuit and the short circuit/working point situation. Hereby gradually increasing the applied bias voltage or

the illumination strength. The number of intermediate steps to be taken can be controlled on the action panel, see Figure 4.1. When divergence occurs, increasing this number can sometimes lead to convergence, however, this slows down your calculation.

When the working point situation is calculated SCAPS immediately calculates the first calculation point of the measurements to be simulated (without any substeps). For *C-f* and *QE* simulations, this first point is identical to the working point conditions. For *I-V* and *C-V* simulations a voltage range has been given, independently from the working point voltage. SCAPS will then jump to the edge of this voltage range which is the closest to the working point voltage. IT IS **STRONGLY ADVISED TO CHOOSE THE WORKING POINT VOLTAGE IDENTICAL TO THE STARTING POINT OF THE VOLTAGE RANGE YOU WANT TO SIMULATE**.

When performing batch (or recorder) simulations, every calculation is started at the startpoint again. This is often unwanted, e.g. when gradually increasing the working point voltage it is more logical to use the previous batch calculation as an input for the next calculation. Varying, voltage, frequency, series/shunt resistance in a batch calculation, this latter approach can be taken by checking the appropriate checkbox on the numerical panel.



### 5.1.3 Small signal analysis

Capacitance and conductance simulations are calculated by performing a small signal analysis at the working point conditions (of course with the voltage (*C-V*) and frequency (*C-f*) get changed). This analysis leads to an expression for the small signal current, which is a complex number. The cell structure is then analysed as if it were a parallel connection of a (frequency dependent) capacitance and a (frequency dependent) conductance.

$$Y(\omega) = \frac{\tilde{J}}{\tilde{u}_{ext}} = j\omega C(\omega) + G(\omega) \quad (16)$$

## 5.2 Setting up a single shot simulation

SCAPS is able to simulate four different measurements: *I-V*; *C-V*; *C-f* and *QE*. The settings of these measurements are immediately available on the action panel, see Figure 5.5. Of course, many more user defined measurements can be simulated by exploiting the recorder facility ☺.

Action	Pause at each step				number of points		
<input type="checkbox"/> Current voltage	V1 (V)	0.0000	V2 (V)	0.8000	<input type="checkbox"/> Stop after Voc	41	0.0200 increment (V)
<input type="checkbox"/> Capacitance voltage	V1 (V)	-0.8000	V2 (V)	0.8000		81	0.0200 increment (V)
<input type="checkbox"/> Capacitance frequency	f1 (Hz)	1.000E+2	f2 (Hz)	1.000E+6		21	5 points per decade
<input type="checkbox"/> Spectral response	WL1 (nm)	300	WL2 (nm)	900		61	10 increment (nm)

Figure 5.5 Measurement simulation settings

- Each of the measurement simulations will only be performed when its appropriate checkbox is checked.
- A voltage/frequency/wavelength range can be set up.
- The number of points (limited to 201) can be set up, or a voltage/wavelength increment or the number of points per frequency decade can be given.
- When a *C-V* simulation is performed you get the *I-V* simulation for free (no need to specify separately).

When performing *I-V* simulations under illuminated conditions, one is often only interested in the voltage range up to the open circuit voltage. However, before starting the simulation  $V_{oc}$  is still unknown. Hence, one can ask SCAPS to stop the *I-V* simulation under illumination as soon as the current becomes positive by checking the ‘stop after Voc’-option.

### 5.3 Numerical parameters

All parameters contained in the Numerical Panel are listed in **Fout! Verwijzingsbron niet gevonden..**

**Table 5.1** The numerical parameters in SCAPS, with their units as displayed in the Numerical Panel, and their default values.

numerical parameter name	default value	unit
Convergence settings		
maximum number of iterations	1000	-
clamping of changes between iterations		
of electrostatic potential $\Phi$	0.5	$kT/q$
of electron Fermi level $E_{Fn}$	0.5	$kT$
of hole Fermilevel $E_{Fp}$	0.5	$kT$
stop criterion: stop if all updates < stop criterion		
of electrostatic potential $\Phi$	0.001	$kT/q$
of electron Fermi level $E_{Fn}$	0.001	$kT$
of hole Fermilevel $E_{Fp}$	0.001	$kT$
Mesh generator settings, see section 5.1.1		
recalculate mesh between iterations	yes	-
maximum number of mesh adaptions at each voltage	10	-
maximum ratio between neighbouring mesh points ( $f_{\max}$ )	1.60	-
maximum ratio between neighbouring mesh points ( $f_{\min}$ )	1.05	-
generation limit ( $G_{\text{limit}}$ ), expressed as current density	1.0	$\mu\text{A}/\text{cm}^2$
recombination limit ( $R_{\text{limit}}$ ), expressed as current density	0.001	$\mu\text{A}/\text{cm}^2$
Calculation of quantum efficiency $QE$ , see section 6.4.7		
calculation mode	constant $N_{\text{phot}}$	-
monochromatic photon particle flux $N_{\text{phot}}$ in $QE$ calculation	$3 \times 10^{18}$	$\text{cm}^{-2}\text{s}^{-1}$
monochromatic photon power flux $P_{\text{phot}}$ in $QE$ calculation	0.1	$\text{mW}/\text{cm}^2$
Defect settings, see section 3.6.3		
number of discretization levels for distributed defects	7	-
width of tail-like energy distribution in multiples of $E_{\text{char}}$	7.0	$E_{\text{char}}$
width of Gaussian energy distribution in multiples of $E_{\text{char}}$	6.0	$E_{\text{char}}$
Tunnel settings, see section 3.9.2		
minimum height of bulk tunnel barrier	2.0	$kT$
choice of tunnel mass	min of adjacent layers	-
allow band-to-band tunneling	no	-
allow intra-band tunneling	no	-
allow tunneling to interface defects	no	-
allow tunneling to contacts	no	-
Update of metastable occupation, see 3.7.4		-
use clamping in iteration of occupation $f_{\text{metastable}}$	no	-
maximum number of iterations	250	-
maximum relative error	0.001	-
Batch and Recorder calculations, see sections 7.4 and 8.2		
to calculate work point, start from previous batch result	yes	-
maximum $V$ for the recording of solar cell characteristics	2.0	Volt
minimum $\Delta V$ for the recording of solar cell characteristics	0.02	Volt

Since SCAPS 3.3.02, version august 2015, the numerical parameters of **Fout! Verwijzingsbron niet gevonden..** can be set back to their default values by clicking a button in the Numerical Panel. When saving a

problem as an all SCAPS file (extension .scaps), or as a .def file from within a script, all numerical parameters are saved. When saving as a .def file in interactive mode, the numerical parameters are only saved when their value differs from the default.

## 5.4 Numerical limitations

Even though many people believe computers to be perfect when performing calculations, they are not!! Every computer has its machine accuracy, which is defined as the smallest (in magnitude) number which, when added to 1.0, produces a floating-point result different from 1.0. This accuracy usually is a small number ( $< 10^{-15}$ ), so that in a pragmatic view one can consider it “accurate enough and close to perfection”. Even though this view is commonly adopted, it is branded as “naïve” and plain “fiction” in [7]. For the simulation of semiconductors, this accuracy is indeed often insufficient, and care should be taken to avoid numerically unstable operations. SCAPS does take care, but sometimes things can still go wrong. Below you can find a list of phenomena which should set off an alarm in your head indicating a numerical problem, and tips and tricks to avoid them and to avoid convergence failures:

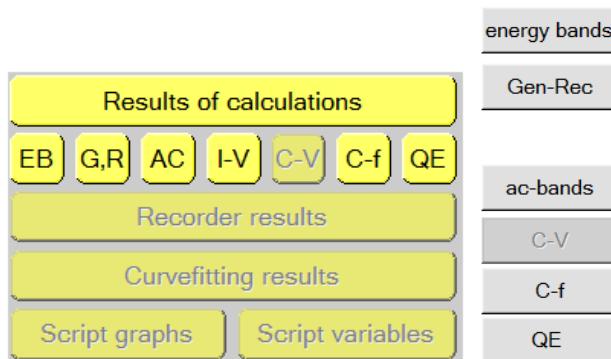
- A negative capacitance or a capacitance increasing with increasing frequency is often not physical. Though exceptions exist, most of the time this behaviour is a result of a numerical error when calculating the capacitance from the small signal current. This occurs when the imaginary part of the small signal current is (in magnitude) much smaller/larger than the real part. This might be the case when large voltage biases or strong illumination conditions are applied.
- The current through the sample (visible in the lower left corner of the EB-panel, in green) should be constant through the sample. SCAPS is optimized to calculate *pn*-junctions (*p* to the left, *n* to the right) and is much less stable for *np*-junctions. IT IS HENCE STRONGLY RECOMMENDED TO KEEP THE *P*-SIDE OF YOUR STRUCTURE TO THE LEFT AND THE *N*-SIDE TO THE RIGHT. *np*-junctions tend to lead to non-uniform currents through the cell, which is unphysical.
- Keep the variation between different calculation steps limited to avoid divergence. The reason for this should be obvious when you have read §5.1.2.
- Keep it realistic! SCAPS is developed and tested to simulate realistic situations, hence things can go wrong when simulating unphysical situations.
- Don't overdo! Do you really want to know the current at  $T = 2$  K and  $V = 3$  kV? Charge carrier densities scale exponentially with  $1/k_B T$ . Hence, the stability of the algorithms exponentially decreases with decreasing temperature, which initially leads to a higher number of iterations needed to get to convergence and at even colder temperatures to an inescapable divergence.

# Chapter 6: Result analysis

SCAPS performs a lot of calculations any time you click the ‘calculate’-button. Your job as a user is to analyse those results. Fortunately there is a lot of help. The analysis-panels can easily be accessed from the action panel or any other analysis panel, see Figure 6.1.

## 6.1 Navigating to the analysis

The analysis-panels can easily be accessed from the action panel or any other analysis panel, see Figure 6.1.



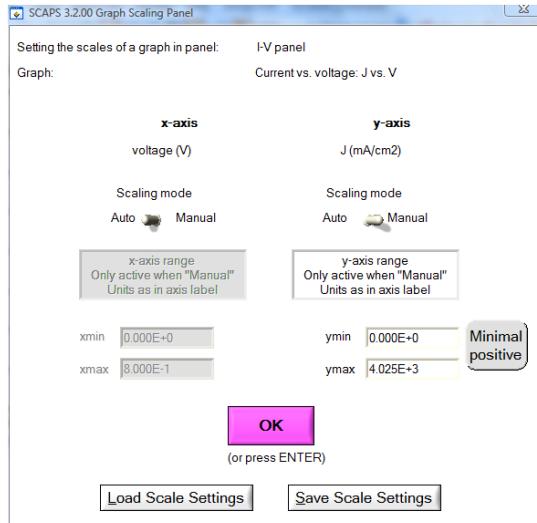
**Figure 6.1** Navigating to the results from the action panel (*left*) or any other panel (*right*)

Several options are available on every panel: saving data, showing data, saving graphs and plotting the panel (sending to a printer). There are options available for scaling and zooming of graphs and to show more info about the plotted curves. Other options are panel specific.

At the bottom of every panel there are two comment windows. The left window is autogenerated and gives the definition file used with its last saving and the when the simulation was performed. The right window can be used to write personal comments.

## 6.2 Zooming and scaling

Most of the graphs are scalable. For most axes a logarithmic or linear scale can be chosen and/or the absolute value of the property can be plotted. The axis-range can be set by clicking a ‘scale’-button, which opens the following pop-up menu.



**Figure 6.2** The Scaling panel

- The scaling of both axis can be set automatically (default) or manually
- A minimum/maximum value can be given
- The settings on this panel can be saved-loaded (generating an ASCII-txt-file)
- Clicking the minimal positive button sets the y-axis scaling to manual and sets ymin to the smallest positive datapoint which needs to be plotted. This might be very useful when there are negative datapoints in a semilogarithmic diagram.

(Almost) all diagrams have moreover zooming facilities. In order to zoom in you should hold the CTRL-button pressed whilst drawing a rectangle with the mouse on a graph. For zooming out you should hold the CTRL-button pressed whilst right-clicking with the mouse. Resetting the initial zooming can often be done as well by changing the log/lin property of the axis. This only works if the axis scaling is set to auto.

### 6.3 Curve info and legend

Performing several simulations, the graphs can get quite crowded (unless you click ‘clear all simulations’ before every simulation). Hence, facilities are provided to get more info about the simulated curves. When the curve info option (top right corner of any panel) is switched on, any click on a graph will render a pop-up menu with information about the graph, curve and point you clicked on. The point/curve closest to the position where you clicked will be selected.

It is also possible to interactively hide and show curves on graphs. This option is available by right-clicking on the **plot/legend**-button (left-clicking leads to plotting the entire panel). A pop-up window is opened which gives a list of all available curves (with a small legend). (Un)checking items in the list leads to hiding/showing the curve on the panel.

### 6.4 Measurement specific options

#### 6.4.1 The energy band panel

This panel shows the band diagram, the carrier densities, the current densities and the occupation of defects of the last calculated work point. When no measurement is selected or a *QE* or *C-f* measurement is selected it displays thus the working point conditions as set on the action panel (or in the batch). If a *C-V* or *I-V* simulation is performed it displays the conditions of the last calculated voltage point. These conditions are mentioned on the bottom left part of the panel.

The panel is displayed during calculations, as such it provides a small movie of how the bands evolve during simulation. When your computer is too fast, you might want to slow down this movie. This is possible by checking the ‘pause at each step’-option on the action panel, see Figure 5.5. This way the program will pause at every calculation step and you can move forward by clicking the continue button (on the EB-panel or the action panel) or abort by clicking the stop-button (on the action panel).

The scaling of the  $x$ -axis on all graphs of the EB-panel is the same. Hence, unless using the zooming option, all graphs display the same part of the structure you simulated.

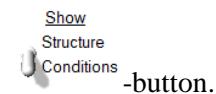
The defect level energy of the defects present and the transition energy of metastable defects are indicated on the band diagram graph. When the defect has an energy distribution (e.g. Gauß or CB tail), the energy  $E_t$  as set in the problem definition is displayed. This energy can of course be graded. The colour legend used for displaying the defect levels can be chosen either grouped according to the defect type (e.g. single acceptor, amphoteric,...) or grouped according to the charge state of the defect level (e.g. 0 $\text{-}$ , 2 $\text{+}$ ,...).

On the occupation probability graph the defect level occupation is displayed. One can choose whether to display the occupation with electrons (default) or holes. For a metastable defect transition the ‘occupation with electrons’ represents the fraction of defects in the acceptor configuration and the ‘occupation with holes’ the fraction of defects in the donor configuration. Just as for the band diagram graph a colour legend can be selected according to defect type or defect charge states. Additionally, a colour code according to the charge of the states can be chosen. In this case the fraction of defects in a specific charge state are displayed. When this option is selected, neutral defects and metastable transitions are not displayed however.

When the defect has an energy distribution (uniform, Gauß, tail) the occupation of several (typically 7) sublevels in this distribution is shown, and the occupation of outermost levels of the distribution, considered in the calculation, is drawn in a thicker line. You will observe that this occupation diagram can get quite complicated and not straightforward to interpret. The use of the Curve Info feature (Section 6.3) is highly recommended in interpreting the occupation graphs: click on a graph, and Curve Info shows you what this graph represents.

#### 6.4.2 The generation-recombination panel

A more detailed view of the defect occupation and of the recombination and generation can be found on the generation-recombination panel. On this panel several recombination-generation processes can be show by checking the appropriate checkboxes. An overview of the available defect(level)s is given in a tree structure. (Un)checking them will make the corresponding curves (un)visible. A graphical view of the simulated structure or a more detailed overview of the simulation conditions can be schown by switching the



#### 6.4.3 The IV-panel

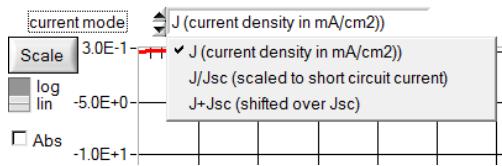
The results of the current-voltage simulations are shown on the IV-panel. The left graph displays all  $I$ - $V$  simulations. The right graph gives detailed information about the recombination currents in the last simulation. This allows to see the main recombination mechanism in the structure for varying voltages.

If the simulation is performed under illumination, the solar cell parameters are calculated and shown, see Figure 6.3. If SCAPS needed to perform an extrapolation to determine these parameters, e.g. because the simulation range was too narrow, the warning LED below the parameter turns red. As allready mentioned, when the generation was ‘from file’, no efficiency can be calculated, but the collection efficiency is given instead.



**Figure 6.3** Visualization of the solar cell parameters

Moreover, the *I-V* graph can be scaled or subtracted automatically with the short circuit current by changing the current mode on top of the graph (Figure 6.4). This enables you e.g to perform analyses as discussed in [16] or to study an illuminated *I-V*-curve with a logarithmic scaling (this is also possible by clicking the ‘absolute’ checkboxes next to the *V* and *I* axes).



**Figure 6.4** Options to display the *I-V* curves: full custom scaling; fast buttons for linear/logarithmic and absolute value/algebraic value; options to scale (with  $J_{sc}$ ) or shift (over  $J_{sc}$ ) all *J-V* curves.

#### 6.4.4 The ac-panel

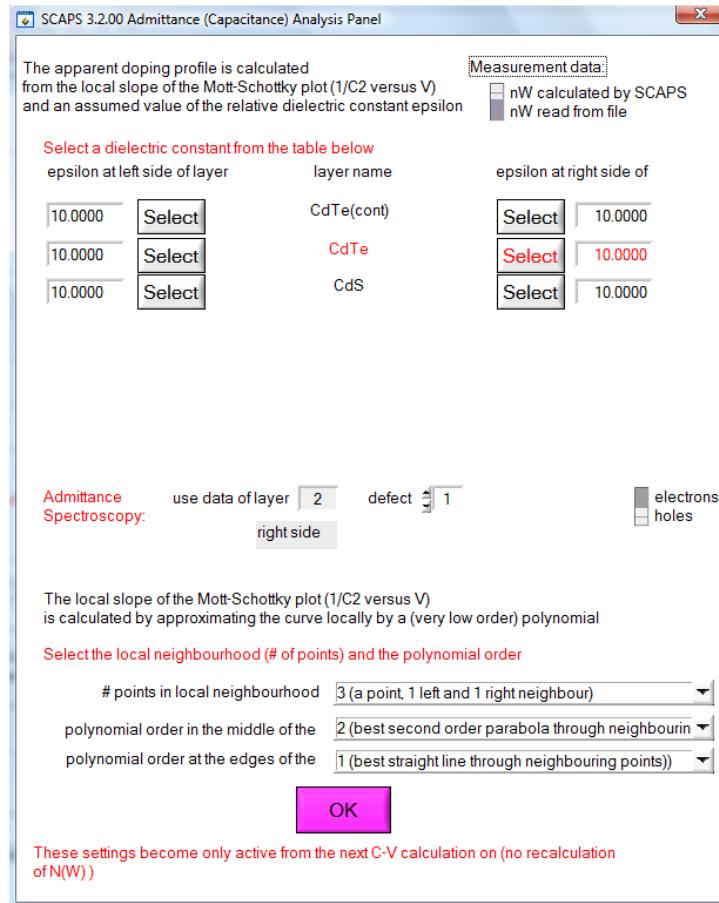
This panel gives an overview of the small signal current (left side of the panel) and of the small signal variations in potential and quasi-Fermi levels (right side of the panel). These properties are complex numbers. The top graphs give the amplitude, the bottom graphs the phase.

#### 6.4.5 The CV-panel

This panel displays the capacitance and conductance as a function of applied voltage. Moreover, a Mott-Schottky diagram and the apparent doping density are calculated. In order to calculate this apparent doping density profile, a numerical differentiation of the data is needed together with a choice of the relative permittivity. These options can be set by clicking the ‘Analysis method’-button, which opens the ‘Admittance (capacitance) analysis panel’ which is discussed in §6.4.5.1.

##### 6.4.5.1 Admittance (capacitance) analysis panel

This panel allows to set the dielectric permittivity and the parameters of the numerical differentiation used in the calculation of the apparent doping density profile and the AS-analysis (§6.4.6). It also allows an automatic calculation of the attempt-to-escape frequency on the AS-panel. The graph of the apparent doping density profile is not immediately adapted to a change of these parameters. The new parameters are only valid for new simulations. The parameters can however be immediately updated in the AS-analysis as there is a ‘recalculate’-button on the AS-panel.



**Figure 6.5** The Admittance (capacitance) analysis panel

Only relative dielectric permittivity values on the left/right side of the layers can be chosen by clicking any of the select buttons on the panel.

The layer which is chosen here determines automatically the choice of the layer-parameters (density of states and thermal velocity) which is used to calculate an attempt-to-escape frequency  $v_0$ . This frequency is calculated as (17) when the switch is set to electrons or as (18) when it is set to holes. The capture cross section is determined by choosing a defect(level).

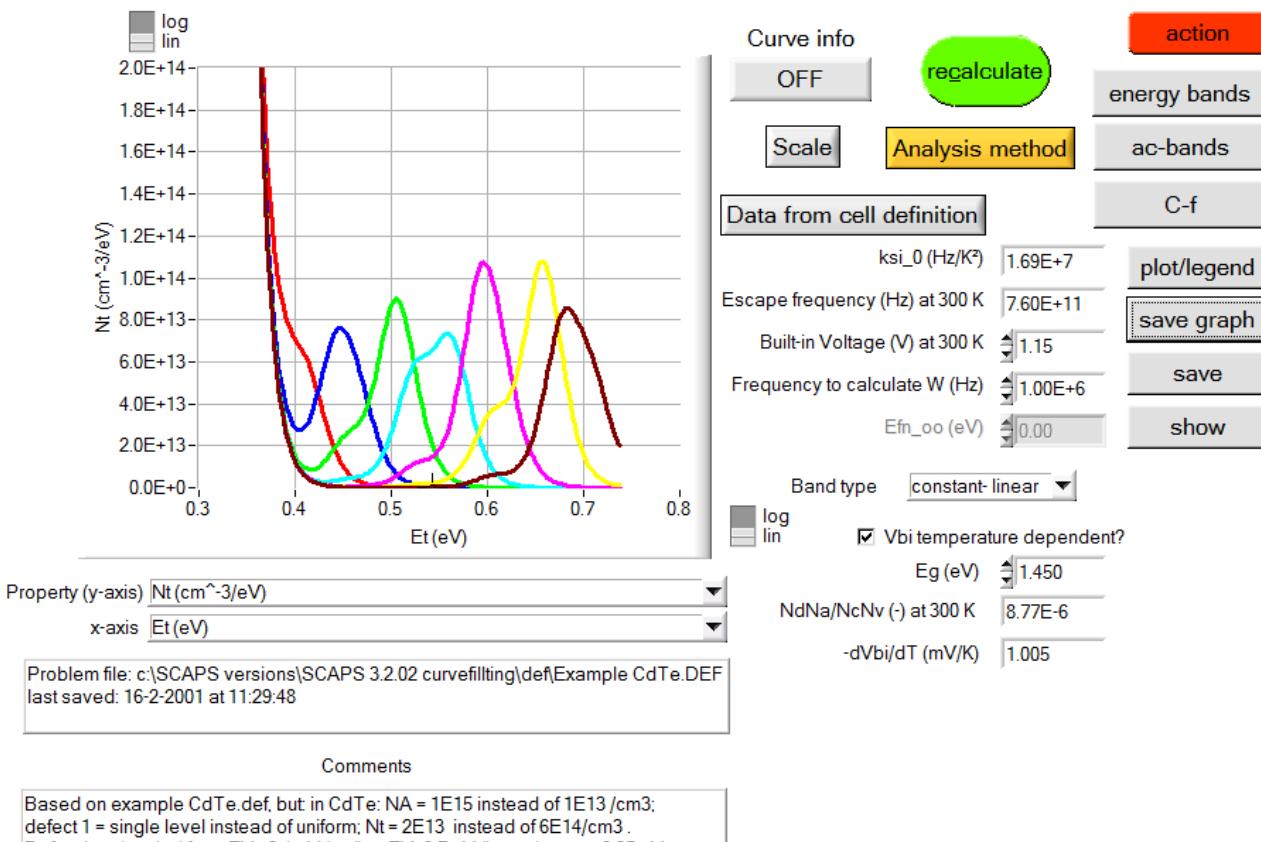
$$v_0 = \frac{\xi_0 T^2}{2} = \sigma_n v_{th,n} N_C \quad (17)$$

$$v_0 = \frac{\xi_0 T^2}{2} = \sigma_p v_{th,p} N_V \quad (18)$$

The numerical derivative is calculated by fitting either a best straight line or a best parabola through a point and its neighbours. The number of neighbours which have to be taken into account and the order of the polynomial (straight/parabola) in the middle and at the edges of the simulation/measurement range can be chosen.

#### 6.4.6 The Cf-panel

This panel displays the capacitance and conductance as a function of simulation frequency. Moreover, a Nyquist plot and a plot of  $G/\omega$  is given. A facility has been added to analyse the results according to the admittance spectroscopy (AS) method outlined in [17] and extended in [18]. By clicking the ‘Admittance spectroscopy’- button the AS-panel is displayed, see Figure 6.6.



**Figure 6.6** The AS panel

The terminology and parameters used in this panel are those defined in [17, 18]. You can select several properties as abscissa and ordinate



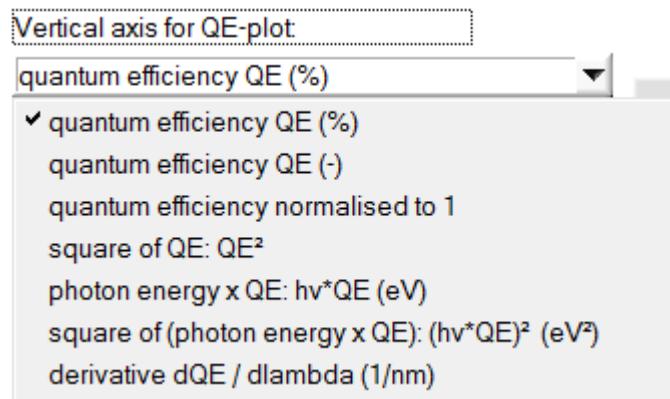
**Figure 6.7** AS panel: abscissa (left) and ordinate (right) choices

The depletion width used in the formulas is immediately determined from the capacitance value at a specified frequency. The dielectric permittivity used to calculate this depletion width and the parameters governing the numerical differentiation can be specified by clicking the ‘Analysis method’-button, which opens the ‘Admittance (capacitance) analysis panel’, see §6.4.5.1. When clicking the ‘Data from cell definition’- button the attempt to escape frequency is calculated from the data set on the ‘Admittance (capacitance) analysis panel’ as well. Its value can however afterwards be changed again. The built-in voltage which is needed for the calculation of  $N_t$  can be chosen temperature dependent according to (19).

$$V_{bi} = E_g + k_B T \ln \left( \frac{N_A N_D}{N_C N_V} \right) \quad (19)$$

#### 6.4.7 The QE-panel

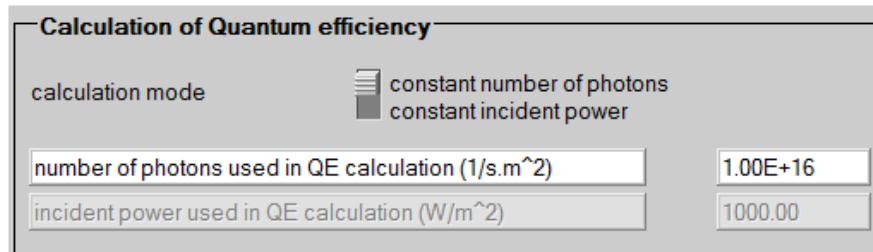
This panel allows to analyze the *QE* simulations. On the horizontal axis one can display either the wavelength or the photon energy of the monochromated light. On the vertical axis one has a wider choice:



**Figure 6.8** Ordinate choices on the QE-panel

The four last options are sometimes used to derive a band gap value from *QE* measurements or simulations.

The *QE* is calculated by comparing the current at the workingpoint conditions and the current when adding an additional amount of monochromatic photons. This number of photons can be set on the numerical panel, Figure 6.9. When both currents are large compared to their difference, it is possible that your computer makes a numerical error in the subtraction leading to a non-physical *QE* which is negative or larger than 100 %.



**Figure 6.9** Setting the number of photons for the *QE* simulation in the Numerical Panel.

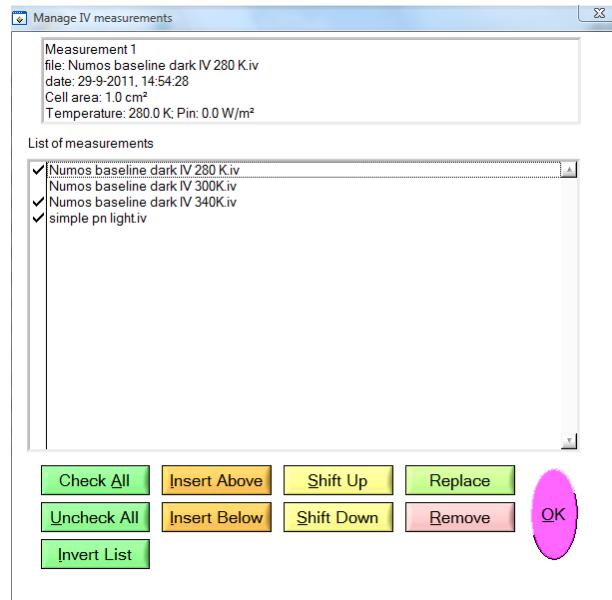
In SCAPS 3.3.02, version august 2015, the defaults values displayed in standard units of the SCAPS panels (thus: cm, mA, mW-based), and were changed to  $P_{\text{phot}} = 0.1 \text{ mW/cm}^2$ , corresponding to 0.001 sun, and to  $3 \times 10^{18} / \text{cm}^2 \text{s}$ , that is for ( $\lambda = 620 \text{ nm}$ ,  $h\nu = 2 \text{ eV}$ ) photons, also roughly corresponding to 0.001 sun. These default values could be representative for actual *QE* measurements with a monochromator set-up.

## 6.5 Managing measurement data

SCAPS provides facilities to compare simulation results with measurement data. Measurement data can be loaded by clicking the **Measurements**-button, which redirects to the ‘Manage measurements panel’, Figure 6.10.

### 6.5.1 The Manage measurements panel

This panel shows a list of the measurement files which are currently loaded. Additional measurements can be added, measurements can be removed or replaced and the order of the list can be changed. By (un)checking items in the list you can control which measurements have to be displayed in the graphs. When selecting an item in the list, a summary of its properties are displayed on top of the panel. This allows to check whether you have loaded the measurement you intended and whether the working point conditions of the file have been read correctly.



**Figure 6.10** Manage measurements panel

### 6.5.2 Structure of a measurement file

Any ASCII-text file can be read as a measurement file. In particular, any of the simulation results which have been saved by SCAPS can be read as a measurement file. So, if you feel unsure whether your file will be read correctly, you can try to start from a SCAPS-results file.

The file extension indicates which kind of measurement it contains. The allowed extensions are ‘.iv’, ‘.cv’, ‘.cf’ and ‘.qe’.

A measurement file is read until the first line which can be interpreted as measurement data is found. This is a line starting with at least two numeric values. All lines above are considered to be part of the header (which might be mere comments or contain information about the working point conditions) All lines below the first data line are considered to contain measurement data. As a result, any line which can not be interpreted as data which occurs below the first data line will be discarded and can thus not be interpreted as working point conditions. Hence, when more than one measurement is contained in a file only the working point conditions of the first measurement will be read, even though the data points of all measurements will be loaded!

#### 6.5.2.1 The workingpoint conditions

SCAPS will try to interpret the information in the header of the file as the working point conditions of the file. When any of these words is found at the beginning of a line, the numerical value which follows is saved. None of these words is case-sensitive. An overview of the code-words used is listed below.

**Table 6.1** Workingpoint code words

code-word(s)	unit	Default value	remarks
area; cell area; cellarea;	cm <sup>2</sup>	1.0	See below
temperature; temp	K	300	
bias	V	0.0	Only for C-f and QE
frequency; freq; frekw	Hz	10 <sup>6</sup>	Only for C-V
Incident power; Pin; P in; P_in	mW/cm <sup>2</sup>	100	Only for I-V and QE
Incident light; Light power dark	#	light	Whenever this word is found, the incident power is set to zero Only for I-V and QE

Units	See §6.5.2.2
-------	--------------

- The area is used to scale the current, capacitance and conductance values to an area of 1 cm<sup>2</sup>.
- Alternative units for the area are supported. Adding any of the words [“cm2”, “cm<sup>2</sup>”, “cm<sup>2</sup>”]; [“mm2”, “mm<sup>2</sup>”, “mm<sup>2</sup>”] or [“m2”, “m<sup>2</sup>”, “m<sup>2</sup>”] to the area statement will change the unit of the area read to cm<sup>2</sup>; mm<sup>2</sup> or m<sup>2</sup>. If none of these words are found the unit of the area is cm<sup>2</sup>.
- In a similar way as for the area, different units for the incident power are supported: mW/cm2; mW/cm<sup>2</sup>; W/m2; W/m<sup>2</sup>; mW/m2; mW/m<sup>2</sup>; W/cm2; W/cm<sup>2</sup>; W/cm<sup>2</sup>.
- The temperature and bias voltage are only used to calculate admittance spectra
- The incident power is used to calculate the efficiency.
- When the incident power is zero (or when the code-word dark has been found) the I-V parameters (thus:  $\eta$ ,  $J_{sc}$ ,  $V_{oc}$  and  $FF$ ) will not be calculated.

### 6.5.2.2 The measurement data

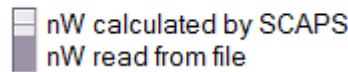
When a first data line is found, all subsequent lines will be interpreted as data lines. Up to five numeric values will be read. The meaning of these numbers varies according to the measurement type as listed below.

**Table 6.2** Data in the columns of a measurement file

measurement	#1	#2	#3	#4	#5
I-V	V (V)	I (mA)*			
C-V	V (V)	C (nF)*	G (S)*	W ( $\mu$ m)	$N_{apparent}$ (cm <sup>-3</sup> )
C-f	f (Hz)	C (nF)*	G (S)*		
QE	$\lambda$ (nm)	QE (%)			

- The properties marked with a \* will be scaled with the area, when no area has been read, an area of 1 cm<sup>2</sup> is assumed.
- The depletion width  $W$  and the apparent doping density  $N_{apparent}$  can be read from the measurement-file, but they can also be calculated from  $C$  and  $V$  by SCAPS. This choice is governed by a switch button on the *capacitance analysis panel*.

#### Measurement data:



The default units for the measurement data are given in the table. It is however possible to change them. In order to do this, a line starting with the code-word “units” has to be added in the header. Next to this code-word a list of properties with their unit can be specified.

Examples: units: I: A V: mV

Units: C: pF V: V G: mS

A list of property-names which can be changed for every measurement type are given in the Table below:

I-V	V:	I:	J:		
C-V	V:	C:	G:	W:	N:
C-f	f:	C:	G:		
QE	lambda:	QE:			

Pay attention! The semicolon ‘;’ is obligatory.

## 6.6 Saving results

When clicking the save or show button, the following properties are saved in an ASCII file which can be read using e.g. notepad or MS Excel. More options to keep track of and save properties are available in the recorder facility, see Chapter 8.

**Table 6.3** Overview of the properties which can be saved on the EB panel

property	remark
x(μm)	
Ec(eV)	Conduction band energy
Fn(eV)	Quasi-Fermi level for electrons
Fp(eV)	Quasi-Fermi level for holes
Ev(eV)	Valence band energy
n(/cm <sup>3</sup> )	Free electron density
p(/cm <sup>3</sup> )	Free hole density
rho(defect) (/cm <sup>3</sup> )	Charge density in defects
net doping (/cm <sup>3</sup> )	$N_D - N_A$
rho(/cm <sup>3</sup> )	$p - n + N_D - N_A + \text{charge in defects}$
E(V/cm)	Electric field
jn(mA/cm <sup>2</sup> )	Electron current density
jp(mA/cm <sup>2</sup> )	Hole current density
jn_tunnel(mA/cm <sup>2</sup> )	Electron tunnel current density
jp_tunnel(mA/cm <sup>2</sup> )	Hole tunnel current density
jtot(mA/cm <sup>2</sup> )	Total current density
generation(#/cm <sup>3</sup> .s)	Generation rate
recombination(#/cm <sup>3</sup> .s)	Recombination rate
cumulative generation (mA/cm <sup>2</sup> )	Cumulative generation rate (summation from the side where the light is incident, to the opposite side)
cumulative recombination (mA/cm <sup>2</sup> )	Cumulative recombination rate (once with summation from left to right, and once summed from right to left). Also interface and contact recombination taken into account
jn1(mA/cm <sup>2</sup> )	Interface recombination current for electrons (left side of the interface)
jn2(mA/cm <sup>2</sup> )	Interface recombination current for electrons (right side of the interface)
jp1(mA/cm <sup>2</sup> )	Interface recombination current for holes (left side of the interface)
jp2(mA/cm <sup>2</sup> )	Interface recombination current for holes (right side of the interface)
jn1[with tunnel](mA/cm <sup>2</sup> )	Same as jn1 but now with tunneling
jn2[with tunnel](mA/cm <sup>2</sup> )	Same as jn2 but now with tunnelling
jp1[with tunnel](mA/cm <sup>2</sup> )	Same as jp1 but now with tunnelling
jp2[with tunnel](mA/cm <sup>2</sup> )	Same as jp2 but now with tunneling
js(mA/cm <sup>2</sup> )	Total surface recombination current density
pi(C/cm <sup>2</sup> )	Surface charge density

### A note on “cumulative” generation and recombination

The cumulative generation is always calculated starting from the side where the light is incident. Thus:

$$\begin{aligned}
 G_{\text{cumulative}}(x) &= \int_0^x G(x') dx' && \text{illumination from left, thus:} \\
 G_{\text{cumulative}}(0) &= 0 \\
 G_{\text{cumulative}}(d) &= G_{\text{total}}
 \end{aligned} \tag{20}$$

or

$$\begin{aligned}
 G_{\text{cumulative}}(x) &= \int_x^d G(x') dx' && \text{illumination from right, thus:} \\
 G_{\text{cumulative}}(0) &= G_{\text{total}} \\
 G_{\text{cumulative}}(d) &= 0
 \end{aligned} \tag{21}$$

The cumulative recombination is offered in two directions of summation (integration): from left to right, and from right to left:

$$R_{\text{cumulative,L}\rightarrow\text{R}}(x) = \int_{0^-}^x R(x') dx' \quad R_{\text{cumulative,R}\rightarrow\text{L}}(x) = \int_x^{d^+} R(x') dx' \tag{22}$$

The notation  $0^-$  means that the left contact is included, and  $d^+$  that the right contact is included. At an interface,  $R_{\text{cumulative}}$  takes a discontinuity (jump) equal to the interface recombination. With these definitions,

$$\begin{aligned}
 R_{\text{cumulative,L}\rightarrow\text{R}}(0) &= R_{\text{left contact}} & R_{\text{cumulative,L}\rightarrow\text{R}}(d) &= R_{\text{total}} \\
 R_{\text{cumulative,R}\rightarrow\text{L}}(0) &= R_{\text{total}} & R_{\text{cumulative,R}\rightarrow\text{L}}(d) &= R_{\text{right contact}}
 \end{aligned} \tag{23}$$

**Table 6.4** Overview of the properties which can be saved on the Generation recombination panel when saving the generation data. All properties are split up per layer, per defect in this layer and per charge state in this defect. (first index = layer, second index = mechanism, third index = defectlevel (0 is the most positively charged and hence the closest to the valence band))

property	remark
x ( $\mu\text{m}$ )	
Total recombination (#/ $\text{cm}^3\text{s}$ )	Total recombination rate
SRH recombination (#/ $\text{cm}^3\text{s}$ )	Schokley-Read-Hall recombination rate (at defects)
Geh (#/ $\text{cm}^3\text{s}$ )	Direct band-to-band excitation of electron-hole pairs by light (this equals zero in dark)
Radiative recombination (#/ $\text{cm}^3\text{s}$ )	Radiative recombination rate
Auger recombination (#/ $\text{cm}^3\text{s}$ )	Auger recombination rate
IPV Gn	Gross IPV electron effect: direct optical excitation from defect to conduction band
net Gn	Net electron generation: IPV Gn + thermal generation - thermal recombination
Gnth	Thermal electron generation (thermal emission of electron from defect to conduction band)
Rnth	Thermal electron recombination (capture by defect of conduction band electron)
IPV Gp	Gross IPV hole effect: direct optical excitation from defect to valence band
net Gp	Net hole generation: IPV Gp + thermal generation - thermal

	recombination
Gpth	Thermal hole generation (thermal emission of hole from defect to valence band)
Rpth	Thermal hole recombination (capture by defect of valence band hole)

**Table 6.5** Overview of the properties which can be saved on the IV-panel.

property	remark
v(V)	voltage
jtot(mA/cm <sup>2</sup> )	Total current density
jbulk(mA/cm <sup>2</sup> )	Bulk recombination current density
jifr(mA/cm <sup>2</sup> )	Interface recombination current density
js_n(mA/cm <sup>2</sup> )	Back contact recombination current density
js_p(mA/cm <sup>2</sup> )	Front contact recombination current density
j_SRH(mA/cm <sup>2</sup> )	Defect recombination current density
j_Radiative(mA/cm <sup>2</sup> )	Radiative recombination current density
j_Auger(mA/cm <sup>2</sup> )	Auger recombination current density
Voc (V)	Open circuit voltage (only under illumination)
Jsc (mA/cm <sup>2</sup> )	Short circuit current density (only under illumination)
FF (%)	Fill factor (only under illumination)
eta (%)	Efficiency (only under illumination)
V_MPP (V)	Voltage at the maximum power point (only under illumination)
J_MPP (mA/cm <sup>2</sup> )	Current density at the maximum power point (only under illumination)

**Table 6.6** Overview of the properties which can be saved on the AC-panel.

property	remark
x (μm)	
jn.re(mA/cm <sup>2</sup> )	Real part of the small signal electron current density
jn.im(mA/cm <sup>2</sup> )	Imaginary part of the small signal electron current density
jp.re(mA/cm <sup>2</sup> )	Real part of the small signal hole current density
jp.im(mA/cm <sup>2</sup> )	Imaginary part of the small signal hole current density
jdispl.re(mA/cm <sup>2</sup> )	Real part of the small signal displacement current density
jdispl.im(mA/cm <sup>2</sup> )	Imaginary part of the small signal displacement current density
j.re (mA/cm <sup>2</sup> )	Real part of the small signal current density
j.im (mA/cm <sup>2</sup> )	Imaginary part of the small signal current density
psi.re (#kT/q)	Real part of the small signal potential
psi.im (#kT/q)	Imaginary part of the small signal potential
Fn.re (#kT)	Real part of the small signal electron quasi-Fermi level
Fn.im (#kT)	Imaginary part of the small signal electron quasi-Fermi level
Fp.re (#kT)	Real part of the small signal hole quasi-Fermi level
Fp.im (#kT)	Imaginary part of the small signal hole quasi-Fermi level

**Table 6.7** Overview of the properties which can be saved on the CV-panel.

property	remark
v(V)	voltage
C(nF/cm <sup>2</sup> )	Capacitance
G(S/cm <sup>2</sup> )	Conductance

$W(\mu\text{m})$	Depletion width
$N_{app}(\text{cm}^3)$	Apparent doping density
$j_{tot}(\text{mA/cm}^2)$	Total current density
$j_{bulk}(\text{mA/cm}^2)$	Bulk recombination current density
$j_{ifr}(\text{mA/cm}^2)$	Interface recombination current density
$j_{opp\_n}(\text{mA/cm}^2)$	Back contact recombination current density
$j_{opp\_p}(\text{mA/cm}^2)$	Front contact recombination current density

**Table 6.8** Overview of the properties which can be saved on the Cf-panel.

property	remark
$f(\text{Hz})$	Frequency
$C(\text{nF/cm}^2)$	Capacitance
$G(\text{S/cm}^2)$	Conductance
$Z.\text{re}(\text{ohm.cm}^2)$	Real part of the impedance
$Z.\text{im}(\text{ohm.cm}^2)$	Imaginary part of the impedance
$Z.\text{magn}(\text{ohm.cm}^2)$	Magnitude of the impedance
$Z.\text{phase}(\text{°})$	Phase angle of the impedance

**Table 6.9** Overview of the properties which can be saved on the AS-panel.

property	remark
$f(\text{Hz})$	Frequency
$C(\text{nF/cm}^2)$	Capacitance
$E_t(\text{eV})$	Activation energy
$-w^*dC/dw(\text{nF/cm}^2)$	Scaled derivative (see [18])
$N_t(\text{cm}^{-3}/\text{eV})$	Calculated defect density

**Table 6.10** Overview of the properties which can be saved on the QE-panel.

property	remark
$\lambda(\text{nm})$	Wavelength
$\text{QE}(\%)$	Quantum efficiency
photon energy (eV)	Photon energy



# Chapter 7: Batch calculations

When you want to explore the influence of one or a few parameters to the solar cell characteristics, you can take profit of the batch option. When you click ‘Batch set-up’, a panel opens where you can choose which parameter to vary, over which range, and in which mode (Lin, Log or custom). You can also define more than one parameter, and vary all of them (in a nested way or ‘simultaneous’). Now, up to nine batch parameters can be defined, but be modest to start. A batch calculation is launched when ‘calculate: batch’ is clicked. After a batch simulation all parameters on the panels are reset as they were before the calculation.

## 7.1 The batch set-up panel

The batch set-up panel allows you to vary up to nine different parameters, when you want to vary more you can also vary entire definition files (§7.2) or exploit the script facilities (Chapter 10). An example of the batch set-up panel is shown in

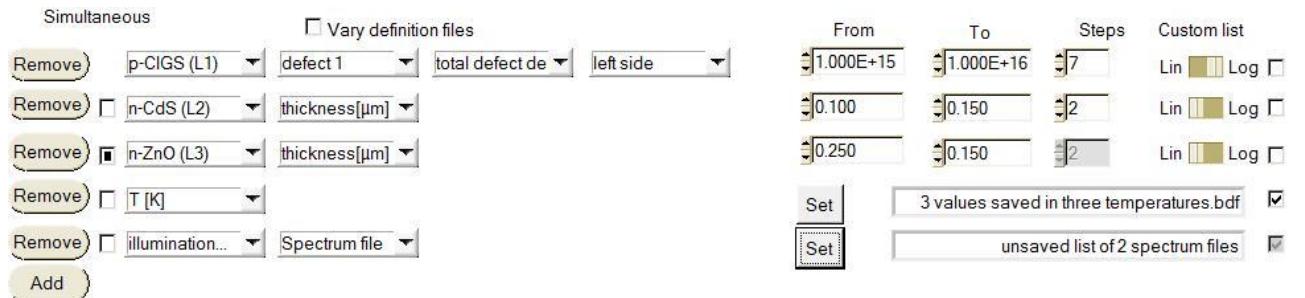


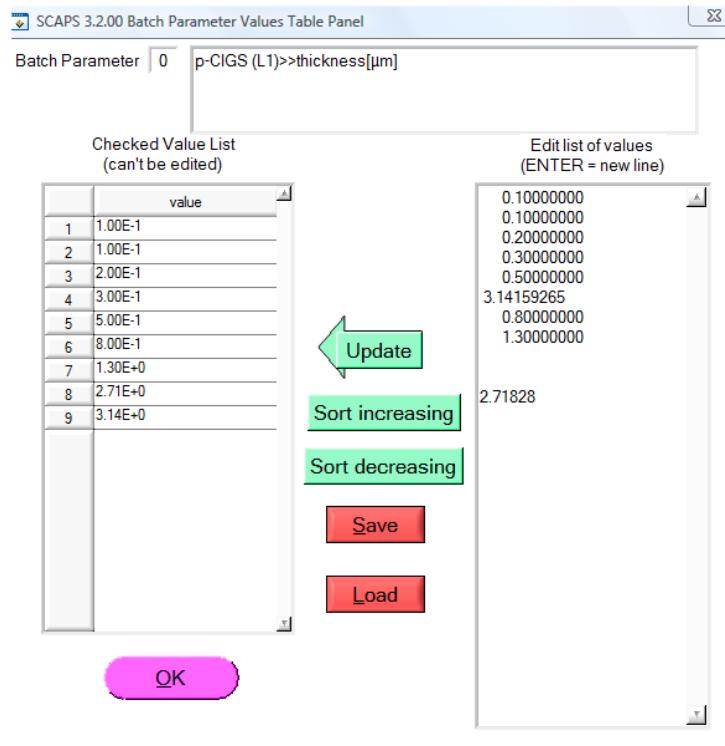
Figure 7.1 The batch set-up panel, illustrating most of the options

- All (most of the) parameters present in the currently defined structure can be varied. Also working point conditions can be varied.
- When checking ‘simultaneous’, this parameter will be varied together with the parameter above (and thus in the same number of steps). If ‘simultaneous’ is not checked, the parameters are varied in a nested way.
- Parameters can be varied in a linear, a logarithmic or a custom defined way.
- Parameters which are files can be varied by entering a list of file names: generation files, spectrum files, filter files, grading files, optical capture cross-section files, initial state workpoint files.
- Some parameters can only take two values (on/off), e.g. illumination.
- When changing any of the illumination parameters (ND filter, spectrum file...) as a batch parameter, the illumination is automatically switched on.
- When changing one of the effective mass parameters, the accompanying tunnelling mechanism is automatically switched on.

- It is possible to save/load the parameters on the batch panel in a ‘\*.sbf’-file which is a standard ASCII-file.
- It is possible to print the batch panel in order to remember your settings.

### 7.1.2 Custom defined values

When checking the custom list option, a Set-button appears which when clicking it opens a panel which allows you to give a enter a list of parameter values.



**Figure 7.2** Setting custom batch values

- Parameters can be typed or copy/pasted in the right list interactively. Afterwards you should press ‘Update’ to allow SCAPS to interpret your typing work. Data which can not be interpreted as a number will be ignored.
- The values in the left list are the values SCAPS will use. These can be sorted.
- Parameter value lists can be saved and loaded. The resulting files are standard ASCII files with extension ‘\*.bdf’. This allows you to make custom lists with any other program (MS Excel, Origin, Matlab...) and load them in SCAPS. The layout of this file is rather tolerant. You just add the parameter values one below the other. All lines which can not be interpreted as only one number are ignored as being comment. We strongly recommend the user to add some comments to the file, so that she/he remembers what data are present in the file, even after not using it for a while.

Some parameters are files rather than numbers, e.g. a spectrum file. Then you have no choice but varying the parameter in a custom way. Clicking the Set-button will open a panel where you can set a list of files. Files can be added/removed/replaced. The order of the file-list can be changed. The list of files can be saved/loaded as well. This is again a standard ASCII files with extension ‘\*.bdf’. In the bdf-file you then have to list the filenames of these files preceded by “file:”. All lines in the bdf-file which are not preceded by this will be treated as comment. The listed files should be placed in the appropriate SCAPS folder: For example, the spectra and the generationfiles which you list should be placed in the “spectrum” and “generation”-file of your SCAPS-folder respectively.

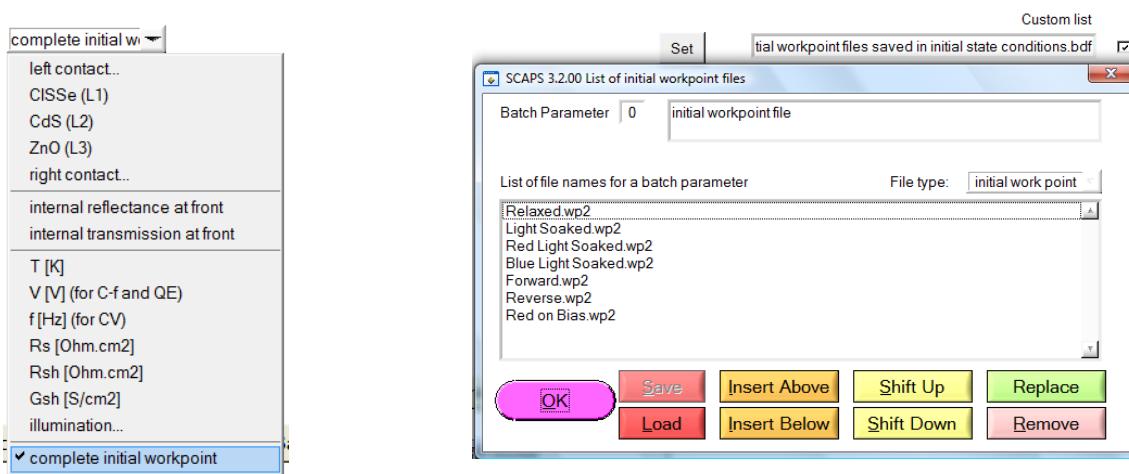
## 7.2 Varying entire definition files

SCAPS offers the opportunity to vary entire definition-files in a batch calculation. This option is activated by checking the ‘vary definition files’-checkbox. A list of files can then be set by clicking the ‘edit/load list of definition files’-button. This list of files is treated in a similar way as any other parameter which has a file-nature rather than a numeric-nature, see §7.1.2.

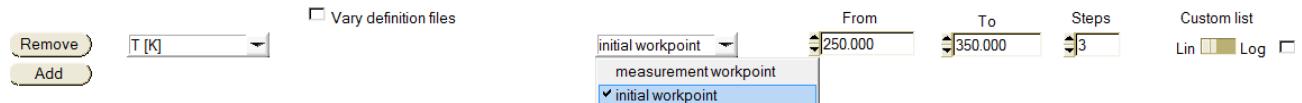
When definition files are varied in a batch, only those parameters which are present in ALL of these definition files can be varied further as a batch parameter.

## 7.3 Varying parameters of the initial state work point

There are two ways to vary the initial workpoint variables in the batch set-up: all together, by specifying a list of .wp2 files (Figure 7.3), or each parameter separately (Figure 7.4).



**Figure 7.3** Selecting all parameters of the initial state workpoint as a batch parameters; a list of .wp2 files has to be specified, where each .wp2 file was made by saving the initial state workpoint in a file.



**Figure 7.4** Selecting a single parameter of the initial workingpoint as a batch parameter.

## 7.4 During calculation...

It might happen that you launched a (big) batch calculation, and all of a sudden you realize you made an error in the set-up. Instead of waiting until all calculations are performed or instead of aborting SCAPS, you can interrupt the batch calculation by keeping the SHIFT-button pressed. The calculation will then stop (but only after it has finished the calculation of the current parameter value)

You might want to launch a big batch calculation overnight, but when you return next morning, you see that the calculation stopped at step 3 due to a convergence error ☺ which gives a pop-up window that should be acknowledged. Inventive users have used pieces of wood to keep the ENTER-button pressed (which solved the problem ☺), but there exists a better way out. On the numerical panel you can ask SCAPS to write error messages to a log-file rather than to inform on the screen.



**Figure 7.5** Error handling on the numerical panel

The batch facility is a powerful tool, but it also enables you to let things go wrong. Be aware that you are more able to set unrealistic parameter values. For example you can set a layer width equal to zero (which will abort SCAPS  $\otimes$ ). When varying files, take care that all files exist in the correct folders.

Every new calculation in a batch starts at the starting point, as explained in §5.1.2, this can however be changed to speed up the calculations. Do take care however to add the voltage/frequency/resistance as the last (bottom-most) parameter in the batch then.

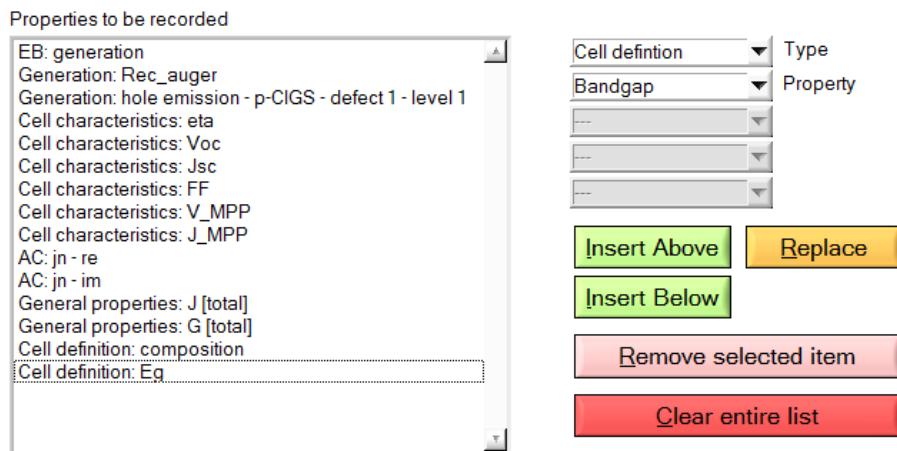
# Chapter 8: Recorder calculations

In a regular single shot or batch calculation, the detailed panels are only available for the last measurement point. To be able to see them as a function of the batch parameters you can launch a record calculation. You should first select the properties which you want to keep track of by clicking ‘Record set-up’. By clicking ‘calculate: recorder’, a recorder calculation is launched. Cell parameters are varied according to the Batch set-up, and all simulations (and only those) are performed which are needed to determine the asked properties. This means the selected measurements on the action panel are ignored!

If this option is used with a little bit of imagination, you can device all kind of measurement simulations with SCAPS.

## 8.1 Setting a recorder

The list of properties to be recorded is made in the Record Set-up, see Figure 8.1. With the help of the five choice-menus (type-property-layer-defect-level) the user can chose a property and add it to the list on the left side of the panel using an Insert or Replace button. There are eight types of properties to be recorded: *I-V* characteristics, General properties, Cell definition, Interfaces, Energy band panel, Generation panel, Occupations and AC panel.



**Figure 8.1** Setting up a recorder.

Please note carefully that it is also possible to record the cell definition properties! This allows you to see how SCAPS interpreted the parameters you set (in the cell definition or in the batch). This might be very helpful when dealing with graded variables!!!

You are able to record properties which are not available in the simulations, e.g. the occupation of defect 3 when there is only one defect. Of course SCAPS can not record what is not available. However, this will not lead to an interruption of the calculations.

Recorder settings can be loaded and saved in a similar way as batch settings, only the extension of the files are now ‘\*.srf’.

## 8.2 Recorder calculations

Clicking the ‘Calculate: recorder’-button leads to a batch recording, which is only available when a batch is set. In this case the action list as it is set on the action panel will be ignored. SCAPS will determine which calculations need to be done in order to record the asked properties. For example when a property of the AC-panel is to be recorded SCAPS will perform a *C-f* simulation at the frequency determined by the working point. When a cell characteristic is to be recorded an *I-V* simulation will be performed starting from 0 V stopping at the open circuit voltage.

SCAPS doesn’t a priori know the open circuit voltage. Hence it will perform an *I-V* simulation from 0 V up to a predefined maximum voltage with the ‘stop at Voc’ option on. This predefined maximum voltage can be set on the numerical panel and has a default value of 2.00V. The accuracy of the determination of the cell characteristics is determined by the increment voltage used in the *I-V* simulation. SCAPS uses the value which is present on the action panel unless it is larger than a predefined minimum value. This value can also be set on the numerical panel and has a default value of 0.05V.

Maximum voltage for the recording of Solar cell characteristics (V)	<input type="text" value="2.00"/>
Minimum voltage increment for the recording of Solar cell characteristics (V)	<input type="text" value="0.05"/>

**Figure 8.2** Recorder settings on the numerical panel

## 8.3 Analysing the recorder results

The results of a recording can be accessed through the ‘Recorder Results’ button on the action panel (or on the EB- or AC-panel). Recorded properties of the type *IV characteristics*, *General properties* or *Interfaces* are immediately plot as a function of one of the batch parameters. Record properties of the other types are plot as a function of the position in the cell (the mesh). The results can also be saved to a file, see Figure 8.3. For each type of property type a different file is saved. The user can chose which files are to be made using the checkboxes seen in Figure 8.3. The show option has the same meaning as everywhere in SCAPS. But as not all property types can be showed in one window, only the last one (= the lowest checkbox checked on Figure 8.3) will be showed.

<input checked="" type="checkbox"/> IV characteristics
<input type="checkbox"/> General properties
<input checked="" type="checkbox"/> Cell definition
<input type="checkbox"/> EB panel
<input type="checkbox"/> Interfaces
<input checked="" type="checkbox"/> Generation
<input type="checkbox"/> Occupations
<input type="checkbox"/> AC panel

<input type="button" value="save"/>
<input type="button" value="show"/>

**Figure 8.3** Saving and showing recorder results

# Chapter 9: Curve fitting

The purpose of curve fitting is to vary one or more parameters in the cell definition to obtain a fit between one or more measured curves and the simulation. The numerical algorithms are taken from [7]. If there is only one parameter to fit: the golden section search ([7], p. 401). When there are 2 to 9 parameters to fit: the downhill simplex method, or Nelder-Mead method ([7], p. 408).

## 9.1 General principles

The parameters to fit are the parameters set in the batch set-up, together with their range and linear or logarithmic character. Not all batch parameters are usable as curve fitting parameters: parameters that are file names (e.g. definition files, spectrum files, absorption files), and parameters of an on/off nature (e.g. ‘illumination on/off’, ‘tunnelling on/off’) are not usable. The curve fitter can only be set up when at least one parameter allowable for curve fitting has been defined in the batch set-up.

The measurements to be fit should be selected in the curve fitting set-up. Currently, a maximum of 9 measurements is possible. A fixed parameter can be attached to each measurement. This can be useful when you want e.g. to fit 4 dark  $I$ - $V$  measurements at 4 different temperatures, or 5 illuminated  $I$ - $V$  curves at 5 illumination intensities. A limited selection of parameters is offered by the user interface as a fixed parameter of a measurements: only those which can be controlled in an experiment (as we think: we do e.g. not believe that one can make different samples that differ by their electron capture constant  $c_n$  of some defect...).

## 9.2 Setting up the curve fitter

The curve fitter can be set by clicking the ‘Curve fit set-up’-button on the action panel. This button is however only available when a batch is set. Once the curve fitter is set, it can be launched by clicking the ‘calculate: curve fitter’-button.

As already mentioned, the curve fitting set-up takes the allowable parameters from the batch set-up, complete with their range and linear/logarithmic nature. In the curve fitting set-up, the starting value of this parameter should be set, see Figure 9.1: by default it is the central value in the range (taking into account the linear or logarithmic character). You can also start from the actual value of a parameter in the present cell definition.

Parameters to be fitted to the measurements		Start Value = Actual Value ?			
(These can only be changed in the Batch Set-up Panel !)					
p-CIGS (L1)	thickness[μm]	Minimum	Maximum	Lin / Log	Start Value
1.000E+0	5.000E+0	linear	3.140E+0	<input checked="" type="checkbox"/>	All
n-CdS (L2)	shallow donor density	1.000E+14	1.000E+18	log.	1.000E+16
				<input type="checkbox"/>	None

**Figure 9.1** Curve fitting parameters taken over from the batch set-up; setting the start value to the central value in the range, or to the actual value in the cell definition.

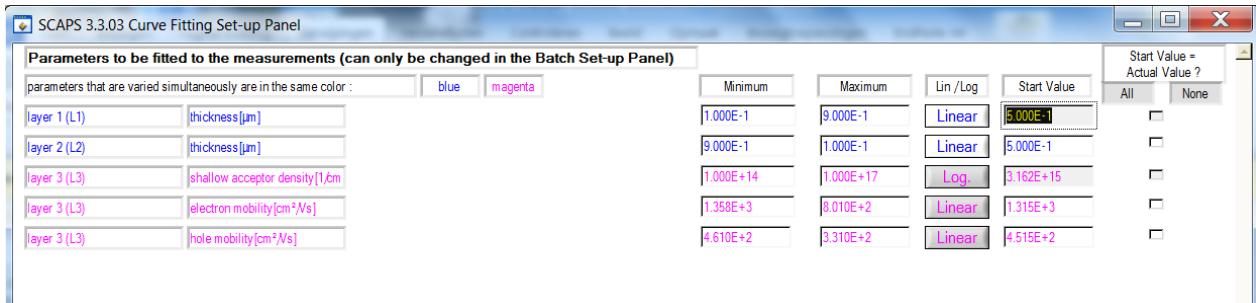
Addition SCAPS 3.3.03, february 2016. The ‘simultaneous’ option in the batch settings is now also used by the curvefitter. The first parameter of a group of ‘simultaneous’ parameters is treated as a master curvefit

parameter, and is varied independently by the curvefitter. The other simultaneous parameters are treated as slaves of that master; each time the master is given a next value in the curvefitting process, the slave values follow automatically. Thus, a group of one master + several slaves is treated as only one curvefit variable.

When a parameter was defined as a table list in the batch set-up, you did not have the option to set its linear or logarithmic character (the batch calculation does not need it, it just takes the values from the list). But the curvefitter needs to know the linear or logarithmic character. Therefore, you have the possibility to set the lin or log property of a parameter in the curvefit set-up, when this parameter was defined as a table list in the batch set-up; even when this curvefit parameter was a slave parameter, you can set its lin/log property then.

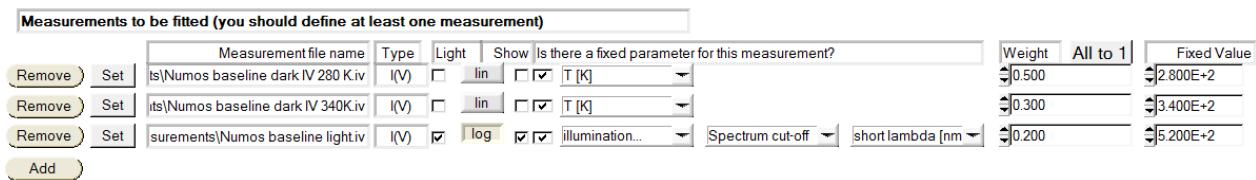
The slave parameter values, that must always follow the master values, are found from interpolation. As usual in SCAPS, the lin/log property of a variable affects the interpolation.

To assist you with the curvefit set-up, groups of curvefit variables (thus one master and one or more slaves) are set in an own color, see Figure 9.2. The first curvefit variable is a group of two thicknesses  $d_1$  (the master) and  $d_2$  (the slave); the second variable is a group of 3 variables:  $N_A$  (the master) and  $\mu_{n3}$  and  $\mu_{p3}$  (the slaves). With these settings,  $d_1 + d_2 = d = 1 \mu\text{m}$  for all values of master  $d_1$ . This is thus a way to vary the thickness of two layers, while the total thickness remains constant. In the second group, doping-depend mobilities  $\mu_n(N_A)$  and  $\mu_p(N_A)$  are set. The values of  $\mu_n(N_A)$  and  $\mu_p(N_A)$  are for each value of master  $N_A$  interpolated between the table values set in the batch set-up; note that we have set here the lin/log property of  $N_A$  to ‘logarithmic’.



**Figure 9.2** Example of a batch and curvefit parameter set-up. There are two independent curvefit variables: one is a group with two thicknesses; and the other is a group with one doping density  $N_A$  and two mobilities  $\mu_n$  and  $\mu_p$ .

Then the measurements should be set, see Figure 9.3. If ‘show’ is clicked, the measurement is shown in the simulation panel ( $I$ - $V$ ,  $C$ - $V$ ,  $C$ - $f$  or  $QE$ ).



**Figure 9.3** Measurement set-up in the curve fitter: the measurement type and illumination, the linear/logarithmic character, and a possible fixed parameter; the (relative) weight of a measurement is available from SCAPS≥3.3.02 on.

The correspondence between a simulation and a measurement is expressed by the  $\chi^2$  sum (‘Chi square’), that we define here as:

$$\chi^2 = \frac{\sum_i (y_{\text{meas}} - y_{\text{calc}})^2}{\sum_i (y_{\text{meas}})^2} \quad (\text{linear}) \quad \text{or} \quad \chi^2 = \frac{\sum_i (\log|y_{\text{meas}}| - \log|y_{\text{calc}}|)^2}{\sum_i (\log|y_{\text{meas}}|)^2} \quad (\text{logarithmic}) \quad (24)$$

where  $y_{\text{calc}}$  is the simulated property and  $y_{\text{meas}}$  the measured property (thus:  $I$ ,  $C$  or  $QE$ ), and only the points  $i$  are considered from the overlap in the simulation and measurement range of  $x$  (thus:  $V$ ,  $f$  or  $\lambda$ ). In these expressions, the simulations are linearly or logarithmically interpolated at the  $x_{\text{meas}}$  values. Our normalisation of  $\chi^2$  ensures that the  $\chi^2$  of all measurements are dimensionless and are scaled to 1: a fit is good if  $\chi^2 \ll 1$ . It also allows that the  $\chi^2$  values of the individual measurements/simulations can be compared and summed. Please note that the definition Eq. (24) of  $\chi^2$  deviates somewhat from the more standard definition as e.g. given in the *Recipes*:

$$\chi^2_{\text{Recipes}} = \sum_i \frac{(y_{\text{meas}} - y_{\text{calc}})_i^2}{\sigma_i^2} \quad (\text{the Recipes, p. 660}) \quad (25)$$

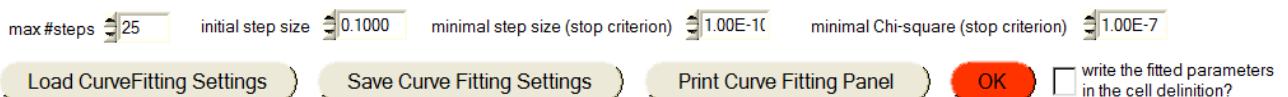
where  $\sigma_i$  is the standard deviation of measurement point  $i$ . This definition Eq. (25) is also dimensionless, but it does not show the ‘scaled to unity’ property as discussed above, and it can cause numerical inaccuracies or problems if the measurement is thought to be nearly perfect or even perfect ( $\sigma \rightarrow 0$  or  $\sigma = 0$ ). Also, attributing an individual standard deviation to each measurement point, even if it were known, is laborious for the user, and (more important ? ☺) very laborious for the programmer. For this reason we prefer our own definition (24) of  $\chi^2$  above the mathematically more standard definition (25).

We use  $\chi^2_{\text{tot}}$  as the figure of merit of the whole curve fitting action:

$$\chi^2_{\text{tot}} = \frac{\sum_i w_i \chi^2_i}{\sum_i w_i} \rightarrow \frac{1}{n} \sum_i \chi^2_i \text{ if all } w_i = 1 \text{ (or no weights set, SCAPS} \leq 3.3.01\text{)} \quad (26)$$

where  $\chi^2_i$  are the  $\chi^2$  values of the individual measurements,  $n$  is the number of measurements, and  $w_i$  the relative weight that the user attributed to each measurement in the Curvefit Set-up Panel (Figure 9.3). In SCAPS versions  $\leq 3.3.01$ , no weight could be set, and Eq. (26) then reduces to its most right-hand member. A user can interpret the weight  $w_i$  of measurement  $i$  as a parameter for the standard deviation, if all measurement point in a measurement had the same standard deviation (then  $w \approx (\sum y_{\text{meas}}^2)/\sigma^2$ ).

Some numerical parameters should also be set in the curve fitting set-up, see Figure 9.4.

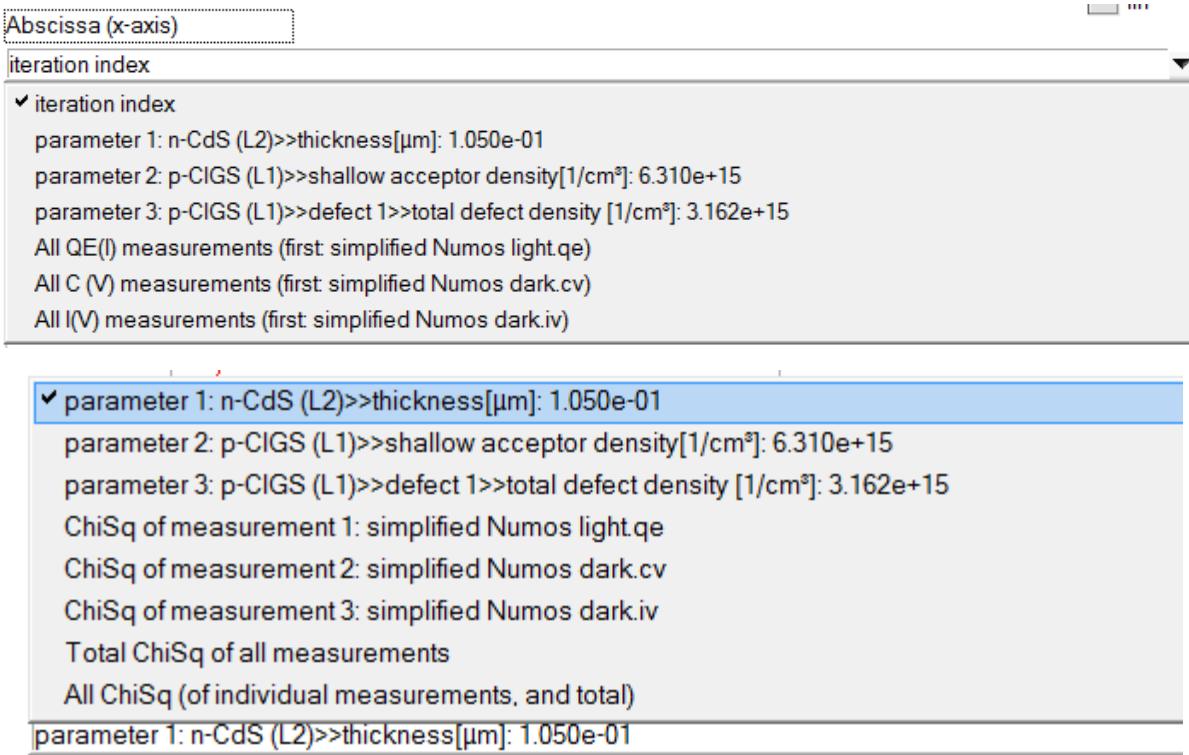


**Figure 9.4** Numerical parameters in the curve fitting set-up

The step size of a parameter variation is relative to the range of this parameter (taking into account the linear or logarithmic nature of this parameter). Do not expect a perfect curve fitting after 25 steps only: 100 steps or so is more realistic, ... but takes more time. The curve fitter walks around though the parameter space, and stops when any of the stop criteria is met (maximum number of iterations exceeded, or step size small enough, or total  $\chi^2_{\text{tot}}$  low enough). The parameter set then obtained is only written in the cell definition when the button at the bottom right is clicked. When you visualise the results of the curve fitting (§9.3), you will be able to judge the quality of the curve fitting, and you will get another chance to write the ‘fitted’ parameters in the cell definition. The curve fitting settings can be saved and loaded, and are also included in the ‘save/load all SCAPS settings’ actions.

### 9.3 Analysing the curve fitting results

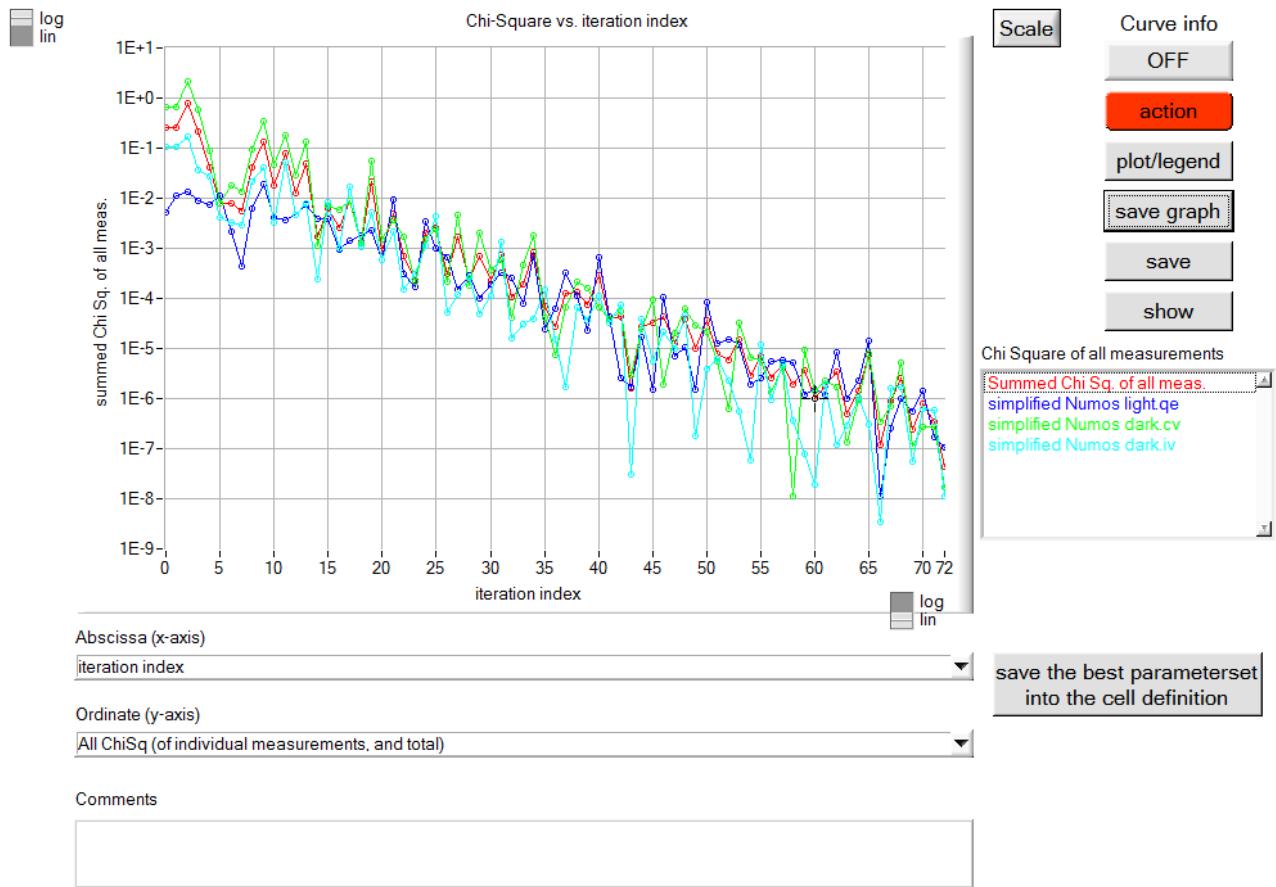
The curve fitting results can be accessed by clicking the ‘Curvefitting results’-button on the action panel, which opens the curve fitting results panel, that allows you to select a variety of graphs to visualise and evaluate the entire curve fitting, see Figure 9.5.



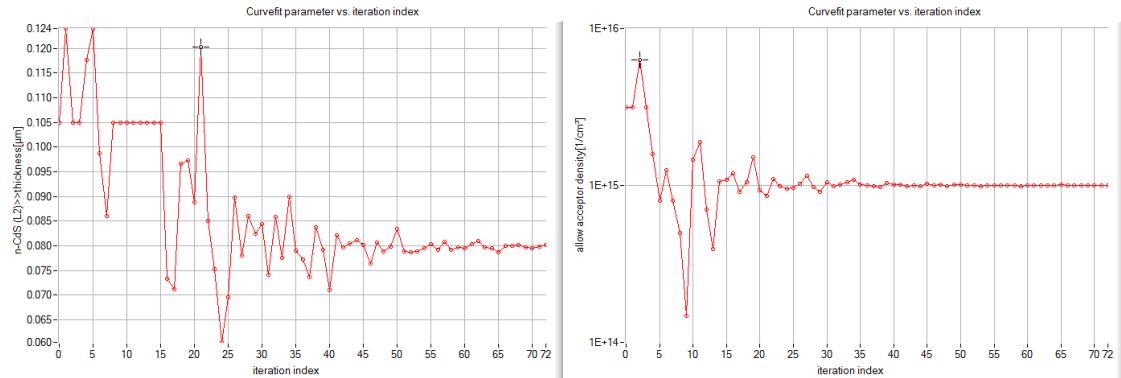
**Figure 9.5** Choices of abscissa (x-axis, top) and ordinate (y-axis, bottom) to visualise the curve fitting results.

The examples shown in Figure 9.6 to Figure 9.9 illustrate the Numos exercise 1 example. The file Numos CIGS baseline.def first has been simplified to simplified Numos CIGS baseline.def, by taking a single energy defect level wherever a band of defects was present: this is speeding up the calculations substantially. All settings, including the batch settings and the curve fitting settings, are saved in ‘simplified Numos curvefitting.scaps’. This calculation took 24 minutes on my computer... The iteration was stopped after 73 steps because the required accuracy ( $\chi^2_{tot} < 10^{-7}$ ) was met. Please note:

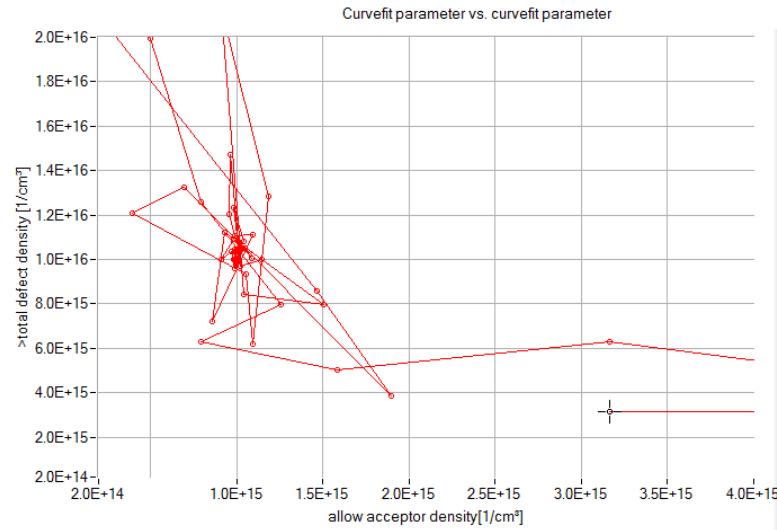
- do not expect a decent curve fit after e.g. 10 iteration steps only! In the example here, 50 – 60 steps seem to be necessary (Figure 9.6, Figure 9.7).
- do not expect such a very nice fit in a real situation! Here, the measurements were ‘constructed’, that is, calculated with SCAPS. Thus, an exact fit is possible. With real measurements, it can very well be that  $\chi^2$  never gets below  $10^{-2}$  or  $10^{-3}$ , meaning that your measurements can not be fitted very well with the present cell definition.
- it is also clear that you should not exaggerate with the number of parameters and the number of measurements. Though 9 is (now) the allowed maximum for both, I would recommend to use much more modest numbers, e.g.  $\leq 3$ , certainly to start with.
- *The recipes* state that it is a good idea to start a curve fitting over again starting with the parameters obtained from a previous curve fitting calculation. You can do so by saving the fitted parameter values in the cell definition, and running the curve fitting calculation again starting from the ‘actual values’, possibly with a smaller initial step, and/or with a narrower range for the parameters.



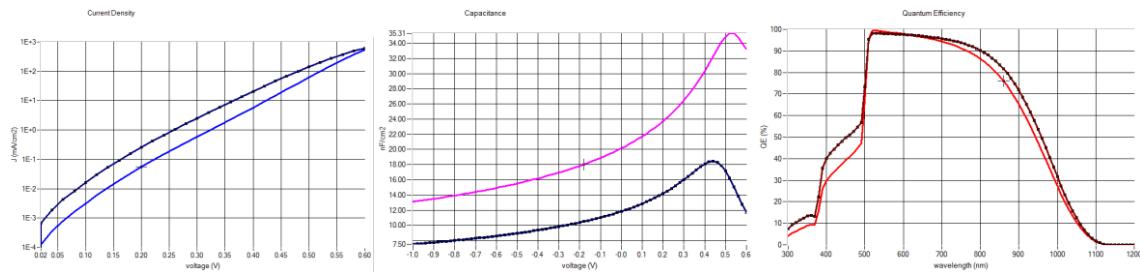
**Figure 9.6** Some examples of curve fitting results: Numos exercise 1, with a simplified problem file. The evolution of all  $\chi^2$  values with iteration index. When you are satisfied with the results, you can insert the fitted parameter values in the cell definition by clicking the grey button.



**Figure 9.7** Some examples of curve fitting results: Numos exercise 1, with a simplified problem file. The evolution of the parameters  $d_{CdS}$  and  $N_A, CIGS$  with the iteration index



**Figure 9.8** Some examples of curve fitting results: Numos exercise 1, with a simplified problem file. Walking in the parameter space to find optimum values for  $N_A,_{CIGS}$  and  $N_t,_{CIGS}$ .



**Figure 9.9** Some examples of curve fitting results: Numos exercise 1, with a simplified problem file. The measurements (dots), the initial calculation (simplified Numos CIGS baseline.def) and the calculation after 73 curve fitting iterations:  $I$ - $V$  (left),  $C$ - $V$  (middle) and  $QE$  (right).

## 9.4 Curve fitting résumé

1. Set up a rough SCAPS model for your cell
2. Verify that scaps is reading your measurements correctly (units, normalisation per area,...): display your measurements and check if all is OK.
3. Refine the model, by hand. As there are probably  $> 100$  parameters to input in your model, you can't leave this to the computer, you should do it yourself. Consider the scaps curve fitting facility only as a fine-tuning.
4. Define the parameters you want to fit in the batch set-up, together with their range and linear/logarithmic nature. Do a quick batch calculation (only very few steps per parameter) to check if your measurement indeed could be fitted by the parameters you have selected
5. Set-up the curve fitter: the parameters were already set in the batch set-up; now only specify the starting value (mid-range or actual value). You have already checked that the measurements are available and readable by scaps. Now select the measurements to curve-fit, set their linear/logarithmic nature, and set one fixed parameter for a measurement if necessary.
6. First use a modest number of iteration steps, e.g.  $< 10$ , and run the curve fitting calculations, to see if everything is going as expected.
7. If it looks promising, set a higher number of steps, run the curve fitter, be patient and wait, and hope that the computer will have solved your problem. Do not be too disappointed when the result is less than you hoped for...

# Chapter 10: Scripting

SCAPS is and has been designed to be a user-interactive program. The most important computer should be based on neurons instead of on transistors. It is important that the user understands what is physically happening rather than performing more simulations than (s)he can analyse. Nevertheless, SCAPS also provides the possibility to write a script and run it.

There are several levels of sophistication for the SCAPS script. One can use it to create a personalized version of SCAPS, to automate actions within the user interface, to use SCAPS in symbiosis with another program, to run SCAPS without mouse-clicks...

An overview of the SCAPS script language is given in §10.4, its use is explained in the preceding sections.

## 10.1 Running SCAPS externally

Normally SCAPS is started by clicking an icon on the desktop. Internally in Windows, a command line attached to this icon is executed. This command line just contains the full path of the SCAPS .exe file (thus scaps2902.exe or scaps3002.exe or so). Now you can add extra arguments to this command line: a list of filenames to be loaded/executed before SCAPS starts: these can be one or more of the definition, action,..., script files listed above; also a spectrum file and a generation file can be given. The order of execution is: first .def, then .act, .scaps, .spe, .gen, .sbf, .srf, and finally .script. By doing so, you can ensure that SCAPS starts in the condition that you prefer, not in the condition fixed by the SCAPS developers: good news for those who had to set e.g. the wavelength range appropriate for CIGS each time again and again.

Here are several ways to start SCAPS from a command line, and to edit this line:

- Make, e.g. with Notepad, a batch file with extension .bat (this stems from the very old MS-DOS times, but is still supported in Windows). Write the command line in one line, e.g.

```
scaps2903.exe all CIGS.scaps AM0.spe
```

A .bat file is run by double clicking it; you can also make a shortcut to it on the desktop. To edit the .bat file, right-click the name and select ‘edit’, or directly open from Notepad.

- Right-click the normal SCAPS-icon on your desktop, select ‘properties’, and edit the third line (‘target’). This is well suited when you do this once and for all. When you would alter the command line more frequently, the previous method might feel more comfortable.
- Some programs (e.g. Matlab) provide options to run external programs. Of course you can benefit from these options to run SCAPS or a SCAPS script within such a program environment. Example ‘\*.m\*’-files (for Matlab users) which allow communication with SCAPS are available for users on request.

## 10.2 Running a script

A script-file is a normal text file, that contains commands that are equivalent to a mouse-click. During execution of the script, SCAPS will not respond to user interventions (mouse or keyboard). After execution of the script, SCAPS returns to its normal interactive mode, or is switched off (to be set with a script command).

When no script file was found in the command line, SCAPS remains the same interactive program as it has always been. A few SCAPS script are distributed with SCAPS to serve as an example.

There are several levels of sophistication to use scripts.

### 10.2.1 Automate mouse clicking

The basic use is that you write down all actions you would do in the interactive mode, and then can leave the lab. One advantage of doing so would be that you could split up your to-do list in several smaller batch jobs instead of one gigantic batch job (by giving several `load batchsettings` commands), and that you save the results in between (by giving `save results` or `save graphs` commands). This is safer when there is a risk that your computer (or SCAPS ☺) would hang up underway; also, all results are waiting for you when you come back to the lab.

This automation of interactive mouse clicking can be initiated from a batch file, an external program, but also from within SCAPS. Hereto set up a script using the ‘script set-up’-button and run it using the ‘execute script’-button.



**Figure 10.1** Running a script from within the SCAPS user interface.

### 10.2.2 Run external programs within SCAPS

A more sophisticated use is that you start and run an own (or another) program somewhere during the execution of the script, by giving a `runsystem` command. For example, you could load a problem, do just the equilibrium calculation, and save the results of the generation panel to a file; this also contains the mesh. You could then open an own program, read this mesh-file and use it to do an own calculation of the optical generation, and save this in a file with the format of a SCAPS generation file. When your function returns, the script is continued. You could then load this generation file, and do all calculations you want. This procedure is more convenient than the full-manual method that several SCAPS users have intensely used.

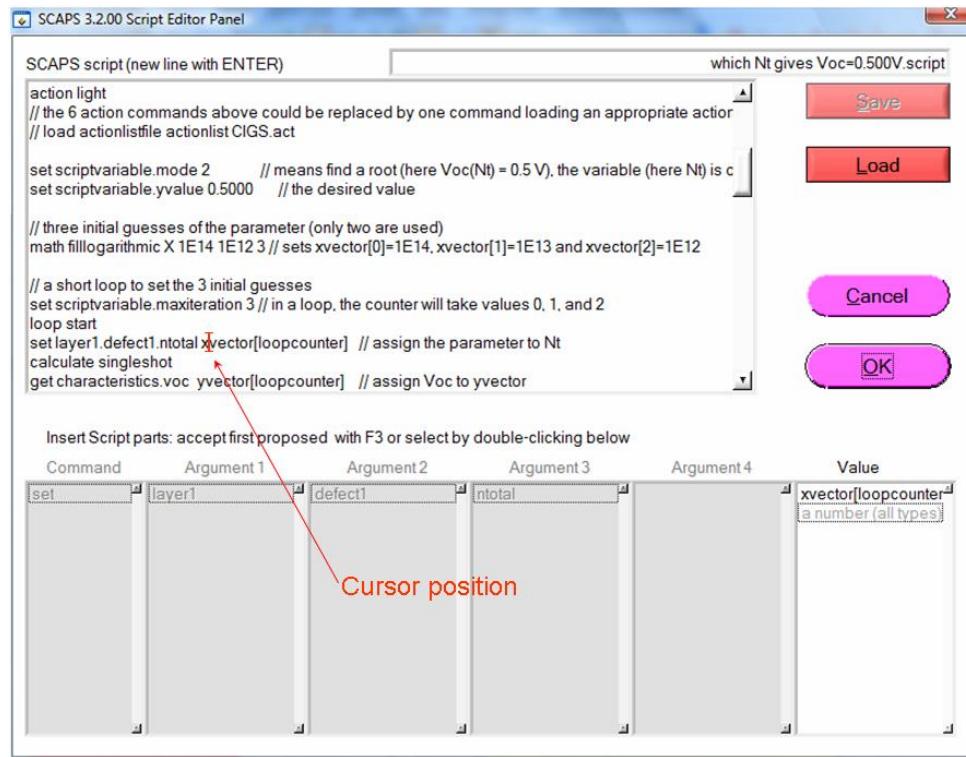
In this way, the communication between scaps and your own program is only via the file system: both scaps and your program read and write files, but do not communicate directly with each other. Also, reading and especially writing files in a scaps format might be cumbersome for a programmer.

### 10.2.3 Running dynamically linked libraries

For advanced users, a more direct communication method between SCAPS and an own program has been implemented. The user program should have been compiled as a *dynamically linked library*, and should be placed in the SCAPS mother directory as a .dll and a .lib file. Also, these filenames, the name of the dll-program(up to now there is only one), and the definition (the argument list) of this function is fixed. This functionality seems to address well skilled programmers only. However, one such dll, with a flexible functionality, is distributed with SCAPS, and using it does not require high level programmer skills: see the description of the `rundll` command below, and the examples distributed with SCAPS. In contrast, we are confident that the basic use of SCAPS scripts can facilitate the simulation work of a broad class of SCAPS users.

## 10.3 The script editor

A script-file can be written in any text editor, e.g. notepad. A typing error is however quickly made. In order to avoid this, it is strongly recommended to use the script editor, which is launched when clicking the ‘script set-up’-button on the action panel, see Figure 10.1, which opens the panel shown in Figure 10.2.



**Figure 10.2** Screen shot of the SCAPS script editor.

- When placing the cursor in an existing line of a script (as shown in Figure 10.2), the components of this command line (thus: the command, the arguments and the value) are shown in the six blocks at the bottom of the panel. When typing a new line in a script, the parts of the command line available so far are proposed in these blocks, and can be selected and placed in the editor box of the script, either by double-clicking on the desired argument or by pressing F3.
- Existing script files can be saved and loaded.

A tip:

When developing a script, frequently insert a `show scriptvariable` statement. The execution of a script stops at such statement, and you can inspect the script variables to see if all is going as you intended. When everything is checked and OK, you can remove or outcomment these `show` statements.

## 10.4 The SCAPS script language

### 10.4.1 General

The SCAPS-directory, this is where the `scaps.exe` file resides, is noted as ‘`scaps\`’.

A comment line in a script is a line that cannot be interpreted as a command line. E.g. any line starting with a punctuation character is treated as comment. You can also add comment at the end of a command line. The Script Editor will recognize such in-line comment when it starts with a double punctuation, except ‘`]]`’ (thus e.g. ‘`//`’ or ‘`!!`’ or ‘`>>`’ are OK ...).

All command lines in a script consist of up to three parts:

command argument value

where `command` and `argument` are reserved words, and `value` is free with some restrictions, depending on the command line. The three components of the command line are separated by whitespace (spaces, tabs,...), but should be on one line. They are not case-sensitive (upper case or lower case letters do not matter). The possible commands are given in Table 10.1.

**Table 10.1** Available script commands

load	action	set	math	calculate
save	clear	get	loop	run
			show	rundll
			plot	runsystem

Whilst processing a script, SCAPS internally maintains a few variables, as specified in Table 10.2. The user can use these variables in `set` and `get` commands, and some are used internally in a `loop`. Also, these variables are passed to an external dll function, that can be made by the user. In this list (and in this entire manual) `{m}` should be replaced by `x`, `y`, `z`, `u`, `v`, `w`, in order to get expressions like `xvector`, `wvector`, `uvalue`, `ny`, `nv`, `zname`, `uindex`...

**Table 10.2** SCAPS script variables

name	C-type	default value	max value
<code>{m}value</code>	double	0	
<code>{m}vector</code>	array of double	0	
<code>n{m}</code>	int	0	
<code>{m}name</code>	character string	empty	max size 256 bytes
<code>{m}index</code>	int	0	
<code>loopcounter</code>	int	0	
<code>maxiteration</code>	int	25	
<code>looperror</code>	double	1E30	
<code>maxerror</code>	double	1E-3	
<code>status</code>	int	0	
<code>mode</code>	int	0	
<code>filename</code>	character string	empty	max size 256 bytes

#### 10.4.2 Load commands

Syntax:

```
load argument value
```

Where `load` is the reserved command word, `argument` can take 8 reserved values, and `value` is a filename, without path. The filename can contain spaces. The files are supposed to reside in their default directories. There is (exceptionally) some freedom allowed in the name of the argument: just writing `definition`, `action`, `batch`, `record`, `allscaps`, `spectrum` or `generation` will also do.

**Table 10.3** SCAPS load commands

command	argument	value	default-directory
load	<code>definitionfile</code>	a filename	<code>scaps\def</code>
load	<code>actionlistfile</code>	a filename	<code>scaps\def</code>

---

load	initialworkpointfile	a filename	scaps\def
load	batchsettingsfile	a filename	scaps\bdf
load	recordersettingsfile	a filename	scaps\bdf
load	allscapssettingsfile	a filename	scaps\def
load	spectrumfile	a filename	scaps\spectrum
load	generationfile	a filename	scaps\generation

**load definitionfile.filnamelist listfile[index]**

Here, **listfile** is a file that contains a list of filenames, in this example, a list of deffiles. The **index** can be a number, or one of the SCAPS script variables **loopcounter**, **maxiteration** or **mode**. This format works for the **load** statements **definitionfile** to **generationfile**.

---

load	singleshotbatch	scaps\bdf
------	-----------------	-----------

---

The last argument (**load singleshotbatch**) is slightly deviating from the others as it does not take a value. The purpose of this command is to work together with the command **get recorder**. When **load singleshotbatch** is called the batch settings file *singleshotbatch.sbf* is loaded. This file sets a batch calculation with one calculation at the working point temperature. So it enables you to perform a recording of a singleshot calculation. This option is very useful as a lot of properties can only be accessed in the script through performing a record calculation and taking the value via **get recorder**. In this way you can access e.g. the electrical field distribution in the structure and do calculations with it. The temperature in this batch is set to the working point value when the command **load singleshotbatch** is called. Hence when you vary the temperature afterwards you should repeat the command again.

#### 10.4.3 Save commands

Syntax:

**save argument value**

Where **save** is the reserved command word, **argument** has a compound syntax; the first part can take 3 reserved values (**settings**, **results** or **graphs**). The **value** is a filename, without path. The filename can contain spaces. The files are supposed to reside in their default directories.

**Commands in blue are new in SCAPS version 3.3.02, june 2015**

**Table 10.4** SCAPS save commands

command	argument	value	default-directory
save	scriptvariables	a filename	scaps\results
save	settings.definitionfile	a filename	scaps\def
save	settings.actionlistfile	a filename	scaps\def
save	settings.batchsettingsfile	a filename	scaps\bdf
save	settings.recordersettingsfile	a filename	scaps\bdf
save	settings.allscapssettingsfile	a filename	scaps\def
save	results.eb	a filename	scaps\results
save	results.genrec	a filename	scaps\results

save	results.SCAPSgenerationfile	a filename	scaps\generation contains only an $G(x)$ table, in the format of a SCAPS generation file (SI units for $G$ )
------	-----------------------------	------------	--

```
save settings.definitionfile.filenamelist listfile[index]
```

Here, listfile is a file that contains a list of filenames, in this example, a list of deffiles. The index can be a number, or one of the SCAPS script variables `loopcounter`, `maxiteration` or `mode`. This format works for the save statements `settings.definitionfile` to `results.SCAPSgenerationfile`.

save	results.ac	a filename	scaps\results
save	results.iv	a filename	scaps\results
save	results.cv	a filename	scaps\results
save	results.cf	a filename	scaps\results
save	results.qe	a filename	scaps\results
save	results.recorder	a filename	scaps\results
save	graph.eb.wholepanel	always.png !	scaps\results
save	graph.eb.energybands	always.png !	scaps\results
save	graph.eb.carrierdensities	always.png !	scaps\results
save	graph.eb.currents	always.png !	scaps\results
save	graph.eb.ftraps	always.png !	scaps\results
save	graph.ac.wholepanel	always.png !	scaps\results
save	graph.ac.currents.amplitude	always.png !	scaps\results
save	graph.ac.currents.phase	always.png !	scaps\results
save	graph.ac.potentials.amplitude	always.png !	scaps\results
save	graph.ac.potentials.phase	always.png !	scaps\results
save	graph.genrec.wholepanel	always.png !	scaps\results
save	graph.genrec.genrec	always.png !	scaps\results
save	graph.genrec.ftraps	always.png !	scaps\results
save	graph.iv.wholepanel	always.png !	scaps\results
save	graph.iv.iv	always.png !	scaps\results
save	graph.iv.recombination	always.png !	scaps\results
save	graph.cv.wholepanel	always.png !	scaps\results
save	graph.cv.cv	always.png !	scaps\results
save	graph.cv.gv	always.png !	scaps\results
save	graph.cv.Mott-Schottky	always.png !	scaps\results
save	graph.cv.dopingprofile	always.png !	scaps\results
save	graph.cf.wholepanel	always.png !	scaps\results

---

save	graph.cf.cf	always .png !	scaps\results
save	graph.cf.gf	always .png !	scaps\results
save	graph.cf.Nyquist	always .png !	scaps\results
save	graph.cf.G(f) /f-f	always .png !	scaps\results
save	graph.qe.wholepanel	always .png !	scaps\results
save	graph.qe.qe	always .png !	scaps\results
save	graph.recording.wholepanel	always .png !	scaps\results
save	graph.recording.resultsgraph	always .png !	scaps\results
save	graph.grading.wholepanel	always .png !	scaps\results
save	graph.grading.gradinggraph	always .png !	scaps\results

---

#### 10.4.4 Action commands

Syntax:

action argument value

Where `action` is the reserved command word, `argument` can take the values in Table 10.5, and `value` is a numerical value or a script variable or a filename, without path. The filename can contain spaces. The files are supposed to reside in their default directories. Some values can take two values only (0 or 1). There is a (very) limited degree of freedom in the exact arguments. E.g. instead of `iv.checkaction`, you can also write `iv.doiv` or `iv.iv`. Instead of `batch.checkaction`, you can also write `batch.dobatch` (as in the user interface of SCAPS < 2.10); and alike with `recording.dorecord`. When the value of these commands is omitted, the value 1 is assumed (giving a clear meaning to the form `doiv, docv,..., dobatches...`).

**Table 10.5** SCAPS action commands

command	argument	value	remark
action	workingpoint.temperature		Kelvin
action	workingpoint.kT		Volt or eV
action	workingpoint.voltage		Volt
action	workingpoint.frequency		Hz
action	workingpoint.numberofpoints	≥ 2	
action	dark	none	overrides light
action	light	none	overrides dark
action	illumination.fromleft	none	overrides fromright
action	illumination.fromright	none	overrides fromleft
action	generationfrominternalmodel	none	overrides generationfromfile
action	spectrumfile	filename	scaps\spectrum
action	spectrumcutoff.on	none	overrides spectrumcutoff.off
action	spectrumcutoff.off	none	overrides spectrumcutoff.on
action	spectrumcutoff.shortlambda		Nm

action	spectrumcutoff.longlambda		Nm
action	intensity.ND		
action	intensity.T		%
action	generationfromfile	none	overrides generationfrominternalmodel
action	generationfile	filename	scaps\generation
action	generationfromfile.attenuation		%
action	initialwp.temperature		Kelvin
action	initialwp.kT		Volt or eV
action	initialwp.voltage		Volt
action	initialwp.numberofpoints	≥ 2	
action	initialwp.dark	none	overrides light
action	initialwp.light	none	overrides dark
action	initialwp.generationfrominternalmodel	none	overrides generationfromfile
action	initialwp.spectrumfile	filename	scaps\spectrum
action	initialwp.spectrumcutoff.on	none	overrides spectrumcutoff.off
action	initialwp.spectrumcutoff.off	none	overrides spectrumcutoff.on
action	initialwp.spectrumcutoff.shortlambda		nm
action	initialwp.spectrumcutoff.longlambda		nm
action	initialwp.intensity.ND		
action	initialwp.intensity.T		%
action	initialwp.generationfromfile	none	overrides generationfrominternalmodel
action	initialwp.generationfile	filename	scaps\generation
action	initialwp.generationfromfile.attenuation		%
action	iv.startV		Volt
action	iv.stopV		Volt
action	iv.points	≥ 2	
action	iv.increment		Volt
action	iv.doiv	none	equivalent to action iv.checkaction 1
action	iv.checkaction	0 or 1	1 is the default
action	iv.stopafterVoc	none	
action	iv.continueafterVoc	none	
action	cv.startV		Volt
action	cv.stopV		Volt
action	cv.points	≥ 2	
action	cv.increment		Volt
action	cv.docv	none	equivalent to action cv.checkaction 1

action	<code>cv.checkaction</code>	0 or 1	1 is the default
action	<code>cf.startf</code>		Hz
action	<code>cf.stopf</code>		Hz
action	<code>cf.total.points</code>	$\geq 2$	
action	<code>cf.points.per.decade</code>	$\geq 1$	
action	<code>cf.totalpoints</code>	$\geq 2$	
action	<code>cf.pointsperdecade</code>	$\geq 1$	
action	<code>cf.checkaction</code>	0 or 1	1 is the default
action	<code>cf.docf</code>	none	equivalent to action <code>cf.checkaction</code> 1
action	<code>qe.startlambda</code>		nm
action	<code>qe.stoplambda</code>		nm
action	<code>qe.points</code>	$\geq 2$	
action	<code>qe.increment</code>		nm
action	<code>qe.checkaction</code>	0 or 1	1 is the default
action	<code>qe.doe</code>	none	equivalent to action <code>qe.checkaction</code> 1

#### 10.4.5 Clear commands

Syntax:

`clear argument`

With `clear scriptvariables`, all script variables (or all but 2 or 3 elements) are set to their defaults. `clear simulations` is equivalent to pressing the ‘clear all simulations’ button in the SCAPS action panel.

**Table 10.6** SCAPS clear commands

command	argument	value	remarks
<code>clear</code>	<code>scriptvariables.all</code>	no value	see text above
<code>clear</code>	<code>scriptvariables.allbutfirst3</code>	no value	leaves $\{m\}vector[i]$ with $i = 0, 1, 2$ . $n\{m\}$ are set to 3. The other script variables are not affected.
<code>clear</code>	<code>scriptvariables.allbutfirst2</code>	no value	idem, but with $i = 0, 1$
<code>clear</code>	<code>scriptvariables.allbutlast3</code>	no value	idem, but shifts elements $i = n\{m\}-1, n\{m\}-2, n\{m\}-3$ to $i = 0, 1, 2$ and leaves them
<code>clear</code>	<code>scriptvariables.allbutlast2</code>	no value	idem, but shifts elements $i = nx-1, nx-2$ (or with $ny$ ) to $i = 0, 1$ and leaves them
<code>clear</code>	<code>simulations</code>	no value	see text above
<code>clear</code>	<code>plot</code>	no value	clears all script graphs; identical with <code>plot clear</code>
<code>clear</code>	<code>scriptgraphs</code>	no value	identical with <code>plot clear</code> or <code>clear plot</code>
<code>clear</code>	<code>actions</code>	no value	unchecks all 4 actions (IV, CV, Cf and

clear all	no value	QE) and restores the workpoint settings to a default (300 K, 0 V, 1 MHz, 5 pts) clears all simulations, all scriptvariables and all plots: equivalent to clear simulations plus clear scriptvariables.all <u>but not</u> clear actions
-----------	----------	--

#### 10.4.6 Set commands

Syntax:

```
set argument value
```

where `set` is the reserved command word, `argument` can take the reserved values from Table 10.8. The `set` command can also be used to set the script variables. The third part of the `set` command line is `value`: this is a numerical value, a script variable or a filename, without path. The filename can contain spaces. The files are supposed to reside in their default directories.

Some values can take two values only (0 or 1). When the value is a numerical value, you can specify a number, e.g. 1.25E16, or one of the internal script variables `mode`, `loopcounter`, `maxiteration`, `{m}index`, `{m}value`, `{m}vector` and `n{m}`. Here `{m}` can be one of the letters x, y, ..., w., and `n{w}` is the number of elements in the corresponding `{w}vector`.

The values of the internal variables `{m}value`, `{m}vector`, ... can be set directly with a `set`-command; also, they are used and possibly changed in `SCAPSUserFunction.dll` (see §10.4.13). The value of `n{m}` can be set directly with the `set` command; it is also updated in some commands: `get`, `math` and `clear`, see later. The allowed indices in SCAPS script vectors are listed in the Table 10.7.

When you set a new value of `n{m}`, the length of the corresponding vector is updated. If the new value is smaller than the previous one, data gets lost, if it is larger, the vector is extended with uninitialised (random) numbers. Before setting a script variable, you might want to re-initialise them with one of the `clear` commands, see later.

These conventions for the use of scriptvectors in the `set` and `get` (see further) commands are summarised in the Table below.

**Table 10.7** SCAPS set command-values

script vector format	index	meaning; remarks
<code>{m}vector</code>	no index	Only as an argument of <code>set scriptvariable...</code> or as the value of <code>get characteristics...</code> The value of <code>n{m}</code> is incremented, all existing elements of <code>{m}vector</code> are shifted one up, and the value of the <code>set scriptvariable...</code> command, or the parameter to <code>get</code> , is placed at <code>{m}vector[0]</code>
<code>{m}vector[-1]</code>	-1	Only as an argument of <code>set scriptvariable...</code> or as the value of <code>get characteristics...</code> The value of <code>n{m}</code> is incremented, and the value of the <code>set scriptvariable...</code> command, or the parameter to <code>get</code> , is placed as the new last element of <code>{m}vector</code>
<code>{m}vector[i]</code>	a number	i is an integer number and should be $0 \leq i \leq n{m} - 1$
<code>{m}vector[last]</code>		For your comfort: internally, last is replaced with the appropriate <code>n{m}-1</code>
<code>{m}vector[loopcounter]</code>	a scriptvariable	

{m}vector[mode]	a scriptvariable	
{m}vector[maxiteration]	a scriptvariable	
{m}vector[{n}index]	a scriptvariable	m and n can differ: you can specify e.g. zvector[yindex]
{m}vector[{n}value]	a scriptvariable	m and n can differ: you can specify e.g. uvector[wvalue]. The value of {n}value is first rounded to the nearest integer.
{m}vector[{n}vector[...]]	a scriptvariable	Here the index ... of the inner {m}vector takes one of the forms allowed in this Table. You can nest many vectors, but that should not be a reason to exaggerate

**Table 10.8** SCAPS set command-arguments

command	argument	value	remark
set the script variables			
set	scriptvariable.maxiteration	integer	
set	scriptvariable.status	integer	
set	scriptvariable.mode	integer	
set	scriptvariable.loopererror		
set	scriptvariable.maxerror		
set	scriptvariable.xvalue		
set	scriptvariable.xvalue		
set	scriptvariable.{m}vector[i]		$0 \leq i \leq nx - 1$ , or i = -1, or no index
set	scriptvariable.n{m}	integer	
set	scriptvariable.{m}name	character string	length < 256
set	scriptvariable.filename	character string	length < 256
set	scriptvariable.filename.SCAPSpath	character string	length < 256

The filename is completed to (or changed to) the full default SCAPS path. E.g. the command `scriptvariable.filename.SCAPSpath mycell.def` will set `filename` to (e.g.) `c:\MB\SCAPS try-outs\def\mycell.def`. If no value is given, the actual `filename` is completed to the SCAPS default path. This command is useful to pass a filename to another programme, that might need to know the full path (e.g. the SCALSDll function).

general set commands			
set	quitscript.interactiveSCAPS	no value	the default
set	quitscript.quitSCAPS	no value	
set	errorhandling.toscreen	no value	
set	errorhandling.appendtofile	no value	the default
set	errorhandling.overwritefile	no value	
set	errorhandling.outputlist.truncate	no value	the default
set	errorhandling.outputlist.fillzeros	no value	
set	errorhandling.outputlist.fillwhite	no value	
set	external.Rs		$\Omega\text{cm}^2$

---

set	external.Rsh	$\Omega\text{cm}^2$
set	external.Gsh	$\text{Scm}^{-2}$
set	internal.reflection	fraction, not %
set	internal.transmission	fraction, not %
<hr/>		
illumination set commands		
<hr/>	set illumination.fromleft	no value
	set illumination.fromright	no value
	set qe.photonflux	$\#\text{cm}^{-2}\text{s}^{-1}$
	set qe.photonpower	$\text{Wcm}^{-2}$
<hr/>		
numerical set commands (see Numerical Panel, and CV-analysis panel) <a href="#">(new in SCAPS 3.3.02 version 2-10-2015)</a>		
<hr/>	set numerical.CV-analysis.layer	first layer = 1
	set numerical.CV-analysis.side	-1=LEFT; +1=RIGHT
	set numerical.CV-analysis.points	# points in local neighbourhood
	set numerical.CV-analysis.order_middle	polynomial order within local neighbourhood
	set numerical.CV-analysis.order_edge	polynomial order at edges of local neighbourhood
	set numerical.recalculate_mesh	0 = NO; 1 = YES;
	set numerical.recalculate_mesh.points	maximum # of mesh points introduced/deleted per iteration cycle
	set numerical.recalculate_mesh.min_ratio	see Numerical Panel
	set numerical.recalculate_mesh.max_ratio	see Numerical Panel
	set numerical.recalculate_mesh.generation_limit	see Numerical Panel
<hr/>	set numerical.recalculate_mesh.	see Numerical Panel
	recombination_limit	
<hr/>		
set	contact.Sn	$\text{cm.s}^{-1}$
set	contact.Sp	$\text{cm.s}^{-1}$
set	contact.opticalfilter.on	no value
set	contact.opticalfilter.off	no value
set	contact.opticalfilter.transmission	no value
set	contact.opticalfilter.reflection	no value
set	contact.opticalfilter.value	fraction, not %
set	contact.opticalfilter.file	a filename
set	contact.opticalfilter	0 or 1
set	contact.workfunction	V or eV
set	contact.flatband.off	no value

---

set	contact.flatband.once	no value
set	contact.flatband.always	no value
<hr/>		
layer set commands: replace layer with layer1, layer2, ... layer7		
set	layer.remove	S
set	layer.duplicate	
set	layer.split.leftthickness	µm
set	layer.split.rightthickness	µm
set	layer.split.leftfraction	-
set	layer.split.rightfraction	-
set	layer.add_default	a layer with default properties is added to the right of the cell structure; the layernumber in the set layer1... command is ignored
set	layer.name	character string
set	layer.thickness	µm
set	layer.Eg	eV
set	layer.chi	V or eV
set	layer.epsilon	-
set	layer.NC	cm <sup>-3</sup>
set	layer.NV	cm <sup>-3</sup>
set	layer.vthn	cm.s <sup>-1</sup>
set	layer.vthp	cm.s <sup>-1</sup>
set	layer.mun	cm <sup>2</sup> V <sup>-1</sup> s <sup>-1</sup>
set	layer.mup	cm <sup>2</sup> V <sup>-1</sup> s <sup>-1</sup>
set	layer.NA	cm <sup>-3</sup>
set	layer.ND	cm <sup>-3</sup>
set	layer.radiative	cm <sup>3</sup> s <sup>-1</sup>
set	layer.Augern	cm <sup>6</sup> s <sup>-1</sup>
set	layer.Augerp	cm <sup>6</sup> s <sup>-1</sup>
set	layer.absorption.file	a filename
set	layer.absorption.A	eV <sup>1/2</sup> cm <sup>-1</sup>
set	layer.absorption.B	eV <sup>+1/2</sup> cm <sup>-1</sup>
<hr/>		
defect set commands: replace layer with layer1, ..., and defect with defect1, defect2 or defect3		
set	layer.defect.remove	
set	layer.defect.duplicate	inserts a defect after

---

		the defect with the given number, and with identical properties as this defect
set	layer.defect.add_default	adds a defect with default properties after the last defect in this layer; the defect number in the command is ignored
set	layer.defect.singlelevel	no value
set	layer.defect.uniform	no value
set	layer.defect.gauss	no value
set	layer.defect.CBtail	no value
set	layer.defect.VBtail	no value
set	layer.defect.neutral	no value
set	layer.defect.singledonor	no value
set	layer.defect.doubledonor	no value
set	layer.defect.singleacceptor	no value
set	layer.defect.doubleacceptor	no value
set	layer.defect.amphoteric	no value
set	layer.defect.aboveEV	no value
set	layer.defect.belowEC	no value
set	layer.defect.aboveEi	no value
set	layer.defect.Et	eV
set	layer.defect.Echar	eV
set	layer.defect.capture_cross_section.electrons	cm <sup>2</sup>
set	layer.defect.capture_cross_section.holes	cm <sup>2</sup>
set	layer.defect.Ntotal	cm <sup>-3</sup>
set	layer.defect.Npeak	cm <sup>-3</sup> eV <sup>-1</sup>

---

Metastable defect set commands: replace layer with layer1, ...,

set	layer.metadefect.E_HE	eV
set	layer.metadefect.E_EE	eV
set	layer.metadefect.E_EC	eV
set	layer.metadefect.E_HC	eV
set	layer.metadefect.E_TR	eV
set	layer.metadefect.aboveEV	no value
set	layer.metadefect.belowEC	no value
set	layer.metadefect.aboveEi	no value
set	layer.metadefect.allowmeta	0 or 1

set	layer.metadefect.Ntotal	$\text{cm}^{-3}$
interface set commands: replace interface with interface1, interface2, ... interface6		
set	interface.name	character string
set	interface.IBtunneling.off	no value
set	interface.IBtunneling.on	no value
set	interface.IBtunneling.me	--
set	interface.IBtunneling.mh	--

---

interface defect set commands: replace interface with interface1,... and IFdefect with  
IFdefect1, IFdefect2, IFdefect3

---

set	interface.IFdefect.remove	
set	interface.IFdefect.duplicate	as for bulk defects
set	interface.IFdefect.add_default	as for bulk defects
set	interface.IFdefect.singlelevel	no value
set	interface.IFdefect.uniform	no value
set	interface.IFdefect.gauss	no value
set	interface.IFdefect.CBtail	no value
set	interface.IFdefect.VBtail	no value
set	interface.IFdefect.neutral	no value
set	interface.IFdefect.singledonor	no value
set	interface.IFdefect.singleacceptor	no value
set	interface.IFdefect.abovehighestEV	no value
set	interface.IFdefect.aboveEVleft	no value
set	interface.IFdefect.belowlowestEC	no value
set	interface.IFdefect.aboveEileft	no value
set	interface.IFdefect.aboveEiright	no value
set	interface.IFdefect.aboveEiGap	no value
set	interface.IFdefect.Et	eV
set	interface.IFdefect.Echar	eV
set	interface.IFdefect.Ntotal	$\text{cm}^{-2}$
set	interface.IFdefect.Npeak	$\text{cm}^{-2}\text{eV}^{-1}$
set	interface.IFdefect.capture_cross_section.electrons	$\text{cm}^2$
set	interface.IFdefect.capture_cross_section.holes	$\text{cm}^2$
set	interface.IFdefect.tunneling.on	no value
set	interface.IFdefect.tunneling.off	no value
set	interface.IFdefect.tunneling.me	--

---

```
set      interface.IFdefect.tunneling.mh
```

---

### 10.4.7 Get commands

Syntax:

```
get argument variable
```

Here, variable is one of the internal script variables.

When you ask for a scalar property, you can use  $\{m\}value$  or  $\{m\}vector[index]$ : the actual value of the variable will then be overwritten with the result of the get action. Here index is one of the allowed formats for indices in the SCAPS script. Other scalar script variables that can be used are `looperror` and `maxerror`.

When you ask for a vectorial properties, like a full *I-V* or *QE* curve, these are placed in two vectors: e.g. *I* in  $\{m\}vector$ , and *V* in  $\{n\}vector$ . If no vectors are specified, `xvector` and `yvector` are assumed: thus `get cv` is identical to `get cv xy`. Also note that only the result of the last simulation is acquired: the last single shot simulation, or the last simulation in a batch run.

The get command updates  $\{m\}name$  as well.

The purpose of the get command is that the script file, or the program launching the script file (e.g. from within SCAPS, from MatLab, another C-programme, Windows script or MS-DOS command language...) would have access to variables such as  $V_{oc}$ ,  $J_{sc}$ ,  $\eta$ , ... or even arrays as  $J(V)$ , ... in a more convenient way then having to retrieve them from a SCAPS output file.

Also, these internal variables can be passed to and updated by the SCAPSUserFunction, that is under the control of the SCAPS user, see §10.4.13.

**Table 10.9** SCAPS get commands

command	argument	value and remarks
<hr/>		
get calculated solar cell characteristics		
get	characteristics.eta	a scalar script value:
get	characteristics.voc	$\{m\}value$ or $\{m\}vector[i]$ where the index <i>i</i> should be in the range $0 \leq i \leq n\{m\}-1$ . Using <i>i</i> = -1 means that the value is appended at the end of $\{m\}vector$ , and that $n\{m\}$ are incremented with one. Using $\{m\}vector[0] \{m\}vector$ (thus without index) means that the size $n\{m\}$ is incremented with one, all elements of the vectors are shifted one position up, and the value returned by <code>characteristics...</code> is placed at $\{m\}vector[0]$ .
get	characteristics.jsc	
get	characteristics.ff	
get	characteristics.vmpp	
get	characteristics.jmpp	
<hr/>		
get general calculated characteristics		
get	iv	Two letters should be passed for the value, corresponding with two vectors. The abscissa is saved in the first, the ordinate in the second. (e.g. <code>get cf zw</code> , saves the frequency in the vector <code>zvector</code> and the capacitance in
get	cv	
get	gv	
get	cf	
get	gf	

get	qe	the vector <b>wvector</b> ). If the value is omitted, <b>xy</b> is assumed. The sizes <b>n{m}</b> are set automatically, and also the names <b>{m}name</b> are set.
get	gx	
get	measurement.iv	In the same way, you can now also get a measurement. It places the <u>last calculated</u> <i>I-V, C-V,...</i> measurement in the vectors specified (if none specified, <b>xy</b> is assumed)
get	measurement.cv	
get	measurement.gv	
get	measurement.cf	
get	measurement.gf	
get	measurement.qe	
get	recombination.tot	... and likewise the recombination information on the IV panel (at least one <i>I-V</i> or <i>C-V</i> calculation should have been done). The first vector will contain the voltage <b>V</b> .
get	recombination.SRH	
get	recombination.rad	
get	recombination.aug	
get	recv.tot	These commands are equivalent to the <b>get recombination</b> commands.
get	recv.SRH	
get	recv.rad	
get	recv.aug	
get	energybands.EC	Also the output of the energy bands panel (obtained by clicking <b>show</b> or <b>save</b> in the EB-panel), can be accessed in the script. The output goes to the two vectors specified (or to <b>xvector</b> and <b>yvector</b> if <b>none</b> were specified). The first vector will contain the mesh <b>x</b> in $\mu\text{m}$ . The second vector the specified property, e.g. the electric field <b>E</b> (note that <b>get energy.E</b> and <b>get energy.field</b> are equivalent).
get	energybands.EV	
get	energybands.EFn	
get	energybands.EFp	
get	energybands.n	
get	energybands.p	
get	energybands.chargeindef	
get	energybands.netdoping	
get	energybands.chargedensity	
get	energybands.E	
get	energybands.field	Remember that at least one calculation ( <i>I-V, C-V, C-f</i> or <i>QE</i> ) should have been done before this
get	energybands.Jn	
get	energybands.Jp	
get	energybands.Jntunnel	
get	energybands.Jptunnel	
get	energybands.Jtot	
get	energybands.generation	
get	energybands.recombination	

get	energybands.cumulativeGeneration	information is accessible.
get	energybands.cumulativeRecombination	
get	energybands.Jtotal_scalar scalar	All energyband output comes in the form of two vectors, e.g. $E_{Fp}(x)$ or $J_{tot}(x)$ . This command puts the value of $J_{tot}$ in a script scalar, e.g. into xvalue, or yvector[loopcounter],...
-----	-----	-----
get	recorder	You can get also the result of a recorder-calculation. You get the result of one recorded property, that is on position mode in the recording list as you have specified it. Here mode is the script variable. It is somewhat laborious to set up the recorder list from a script (you should set up a recorder file, and load this in the script), set the mode variable to the property you want, and then get it. But at least you can access any variable in SCAPS in thus way.

---

get mesh characteristics; layer should be substituted by layer1, layer2, ... or layer7

get	layer.leftindex	the index of the leftmost point in the specified layer
get	layer.leftx	the position $x$ (in $\mu\text{m}$ ) of the leftmost point in the specified layer
get	layer.rightindex	the index of the rightmost point in the specified layer
get	layer.rightx	the position $x$ (in $\mu\text{m}$ ) of the rightmost point in the specified layer
get	numberoflayers	the number of layers in the actual cell definition
get	celllength	the total cell length $x$ (in $\mu\text{m}$ ) of the actual cell definition; both celllength and celllength will work
	celllength	(

---

From SCAPS 3.0.02 on (may 2011), the scalar cell parameters that are available in `set` are made available in `get`. When your cell has graded properties, the parameters that you can `set` or `get` relate to the ‘pureA’ material (when grading is a function of composition) or to the left side of a layer (when grading is a function of position) (see the SCAPS2.8 add-on manual on grading). The units and remarks are as for the corresponding `set` commands.

**Table 10.10** SCAPS get commands (cell definition related)

contact get commands: replace contact with either leftcontact or rightcontact	
get contact.Sn	get contact.opticalfilter.file
get contact.Sp	get contact.opticalfilter
get contact.opticalfilter.on	get contact.workfunction
get contact.opticalfilter.off	get contact.flatband.off
get contact.opticalfilter.transmission	get contact.flatband.once
get contact.opticalfilter.reflection	get contact.flatband.always
get contact.opticalfilter.value	
layer get commands: replace layer with layer1, layer2, ... layer7	
get layer.thickness	get layer.NA
get layer.Eg	get layer.ND
get layer.chi	get layer.radiative
get layer.epsilon	get layer.Augern
get layer.NC	get layer.Augerp
get layer.NV	get layer.absorption.file
get layer.vthn	get layer.absorption.A
get layer.vthp	get layer.absorption.B
get layer.mun	get layer.composition
get layer.mup	
defect get commands: replace layer with layer1, ..., and defect with defect1, defect2 or defect3	
get layer.defect.energydistribution	returns an integer that encodes for single, uniform, Gauß, ...
get layer.defect.chargetype	returns an integer that encodes for neutral, single donor, ....
get layer.defect.referencelevel	returns an integer that encodes for above $E_V$ , below $E_C$ , above $E_i$

```
get    layer.defect.Et  
  
get    layer.defect.Echar  
  
get    layer.defect.capture_cross_section.electrons  
  
get    layer.defect.capture_cross_section.holes  
  
get    layer.defect.Ntotal  
  
get    layer.defect.Npeak
```

Metastable defect get commands: replace layer with layer1, ...,

get	layer.metadefect.referencelevel	returns an integer that encodes for above $E_V$ , below $E_C$ , above $E_i$
get	layer.metadefect.E_HE	
get	layer.metadefect.E_EE	
get	layer.metadefect.E_EC	
get	layer.metadefect.E_HC	
get	layer.metadefect.E_TR	
get	layer.metadefect.Ntotal	
get	layer.metadefect.allowmeta	

About the Script and **grading**: The SCAPS script does not fully support grading, as of now (1-9-2013, version 3.2.01). The properties that you set or get are either the property of the ‘pure A material’ (composition  $y = 0$ ) if the property is not graded, or graded as a function of composition  $y$ , or the property at the left side of the layer, if the property is graded as a function of position  $x$ . You cannot access or set any other grading property (thus e.g. the value of the pure B material, bowing parameters, characteristic exponential decaylengths...). However, from SCAPS 3.2.01 (2-9-2013) on you can set a grading file for all the properties above that can be graded. The grading profile for this property is then automatically set to ‘graded from file’. To do so, use e.g. `set layer1.Eg.file mypersonal_Eg_gradingfile.grd` or `set layer3.composition.file myfavourite_composition_gradingfile.grd`. The extension `.grd` is necessary, and the programme assumes that the file resides in the .../grading subdirectory, unless you specify a full file path. (And make sure that the file exists and can be found: do a single shot calculation before you launch complicated scripts ☺).

interface get commands: replace interface with interface1, interface2, ... interface6

```
get  interface.IBtunneling.off           get  interface.IBtunneling.me  
get  interface.IBtunneling.on           get  interface.IBtunneling.mh
```

---

interface defect get commands: replace interface with interface1,... and IFdefect with IFdefect1, IFdefect2, IFdefect3

get	<code>interface.IFdefect.energydistribution</code>	returns an integer that encodes for single, uniform, ...
get	<code>interface.IFdefect.chargetype</code>	returns an integer that encodes for neutral, single donor, ...

---

```

get interface.IFdefect.referencelevel      returns an integer that encodes for above  $E_V$  left,
                                              above highest  $E_V$ , below lowest  $E_C$ , ...

get interface.IFdefect.Et

get interface.IFdefect.Echar

get interface.IFdefect.Ntotal

get interface.IFdefect.Npeak

get interface.IFdefect.capture_cross_section.electrons

get interface.IFdefect.capture_cross_section_holes

get interface.IFdefect.tunneling.on

get interface.IFdefect.tunneling.off

get interface.IFdefect.tunneling.me

get interface.IFdefect.tunneling.mh

```

---

Some action commands and some other set commands now also have their corresponding get command

---

get action.workingpoint.frequency	get action.generationfromfile.attenuation
get action.workingpoint.temperature	get action.initialwp.temperature
get action.workingpoint.kT	get action.initialwp.kT
get action.workingpoint.voltage	get action.initialwp.voltage
get action.spectrumcutoff.shortlambda	get action.initialwp.spectrumcutoff.shortlambda
get action.spectrumcutoff.longlambda	get action.initialwp.spectrumcutoff.longlambda
get action.intensity.ND	get action.initialwp.intensity.ND
get action.intensity.T	get action.initialwp.intensity.T
get action.Pin.from_lamp	
get action.Pin.after_cutoff	
get action.Pin.after_cutoff_and_ND	
get action.Pin.in_cell (same as above)	
get action.Jideal.in_genfile	
get action.Jideal.after_attenuation	

---

get action.Jideal.in_cell (same as above)	
get external.Rs	get action.initialwp.generationfromfile.attenuation
get external.Rsh	get internal.reflection
get external.Gsh	get internal.transmission

---

The command `get recorder` gets the data from the record results, and hence allows to access almost any property in script mode. The recorded property is selected by the value of the script variable `mode` (the first property in the record setting list is accessed when `mode = 0`, the next when `mode = 1...`). If the recorded property is a scalar value as a function of the batch calculation (e.g. the open circuit voltage) the abscissa consists of the numbers of the batch calculations. If the recorded property is a vector (e.g. the conduction band profile) the abscissa value is the mesh. In this case only the recorded vector of the last batch calculation is copied to the script variable. In this view, performing a batch with only one calculation using `load singleshotbatch` is very recommended, see §10.4.2.

#### 10.4.8 Math commands

Syntax:

```
math argument value
```

The `math` commands allows to perform mathematical operations on the script vectors. The argument is followed by a list of one to four letters form the set {`x,y,z,u,v,w`}. Uppercase or lower case do not matter; however, for clearness in the description below, we will use upper case letters when vectors are meant, and lower case letters otherwise.

If a variable is a vector, e.g. `Y`, it is interpreted as `Yvector`. If a variable is a scalar, e.g. `z`, it is interpreted as `zvalue`. If a variable is an index, e.g. `w`, it is interpreted as `windex`.

Some operations are on vectors. Then operations are performed element by element and can be performed ‘in place’ (e.g. `A ← A+B`) where the original content of `A` is lost.

**Table 10.11** SCAPS math commands

---

comma	argument	value	remark
nd			
A,B and C represent a vector variable			
a, b, c represent a scalar variable			
i represents an index variable			
math add		ABC	$A = B + C$ . Vector operation
math multiply		ABC	$A = B * C$ . Vector operation
math subtract		ABC	$A = B - C$ . Vector operation
math divide		ABC	$A = B / C$ . Vector operation
math exp		AB	$A = \exp(B)$ . Vector operation
math log		AB	$A = \ln(B)$ . Vector operation
math power		ABC	$A = B ^ c$ . Vector operation, A and B are vectors, but c is a scalar
math abs		AB	$A_i =  B_i $ . Vector operation

---

math	change_sign	AB $A_i = -B_i$ . Vector operation
math	reciproque	AB $A_i = 1/B_i$ . Vector operation
math	vectoradd	ABC $A = B + C$ . Vector operation. Identical to add
math	vectormultiply	ABC $A = B * C$ . Vector operation Identical to multiply
math	vectorsubtract	ABC $A = B - C$ . Vector operation Identical to subtract
math	vectordivide	ABC $A = B / C$ . Vector operation Identical to divide
math	vectorexp	AB $A = \exp(B)$ . Vector operation Identical to exp
math	vectorlog	AB $A = \ln(B)$ . Vector operation Identical to log
math	vectorpower	ABC $A = B ^ c$ . Vector operation, A and B are vectors, but c is a scalar. Identical to power
math	vectorabs	AB $A =  B $ . Vector operation, identical to abs
math	vectorchange_sign	AB $A = -B$ . Vector operation. Identical to change_sign
math	vectorreciproque	AB $A = 1/B$ . Vectoroperation,identical] to reciproque
math	scalaradd	abc $a = b + c$ . Scalar operation
math	scalarmultiply	abc $a = b * c$ . Scalar operation
math	scalarsubtract	abc $a = b - c$ . Scalar operation
math	scalardivide	abc $a = b / c$ . Scalar operation
math	scalarexp	ab $a = \exp(b)$ . Scalar operation
math	scalarlog	ab $a = \ln(b)$ . Scalar operation
math	scalarpower	abc $a = b ^ c$ . Scalar operation.
math	scalarchange_sign	ab $a = -b$ . Scalar operation
math	scalarabs	ab $a =  b $ . Scalar operation
math	scalarreciproque	ab $a = 1/b$ . Scalar operation
math	integrate	ABC $A(B) = \int_0^B C(B') dB'$ . Vector operation
math	differentiate	ABC $A(B) = \frac{dC(B)}{dB}$ . Vector operation
math	interpolate	aAbB When A is considered as a function of B, thus $A_i = A(B_i)$ , it returns by interpolation $a = A(b)$
SCAPS 3.3.03, february 2016. Four more interpolation functions: <code>interpolate.</code> <code>linlin.</code> <code>linlog.</code> <code>.loglin</code> and. <code>loglog</code>		
math	interpolate.linlin	aAbB Identical with <code>math interpolate</code>

math	interpolate.linlog	aAbB First “plot” data in a $A$ vs. $\log(B)$ plot. Then apply math interpolate
math	interpolate.loglin	aAbB First “plot” data as $\log(A)$ vs. $B$ Then apply math interpolate
math	interpolate.loglog	aAbB First “plot” data as $\log(A)$ vs. $\log(B)$ Then apply math interpolate
math	closestindex	iaA Returns the index $i$ for which the element $A_i$ is closest to $a$
math	extract	ABcd $c$ and $d$ are indices. The elements $c$ up to and including $d$ of vector $B$ are copied into vector $A$ , that now gets dimension $d-c+1$ ; the previous contents and dimension of $A$ is lost. The operation can be ‘in place’: $A=B$ is allowed. This function is useful to pick out the information of one layer from the full $x$ -dependence, when the indices $c$ and $d$ are obtained with get layer.leftindex and get layer.rightindex .
math	increment	i The index $i$ is incremented with one. When $i$ is one letter from {x, y, ..., w}, the index is interpreted as xindex, yindex,..., or windex. But you can also use loopcounter, maxiteration, status, mode, or one of the words xindex... written in full.
math	decrement	i The index $i$ is incremented with one. $i$ is a SCAPS script index, see the above statement for the valid format.
math	min	aA $a = \min(A_i)$
math	max	aA $a = \max(A_i)$
math	sum	aA $a = \sum_i A_i$
math	average	aA $a = \frac{1}{n_A} \sum_{i=0}^{n_A-1} A_i$
math	median	aA $a = A(i = n_A/2)$
math	sumofsquares	aA $a = \sum_i A_i^2$
math	averageofsquares	aA $a = \frac{1}{n_A} \sum_{i=0}^{n_A-1} A_i^2$
math	product	aA $a = \prod_i A_i$

math    geometricaverage

$$aA \quad a = \left( \prod_i A_i \right)^{1/n_A}$$

math    chi\_square

E        aBCD where B contains  $X_{\text{measured}}$  and C contains  $Y_{\text{measured}}$ ; and D contains  $X_{\text{calculated}}$  and D contains  $Y_{\text{calculated}}$ . Then chi\_square is calculated as:

$$\chi^2 = \frac{\sum_i (y_{\text{meas}} - y_{\text{calc}})^2}{\sum_i (y_{\text{meas}})^2} \quad \text{The sum}$$

is taken at the measurement points  $x_{\text{meas},i}$  that fall within the range of the calculated  $x_{\text{calc}}$  points.  $y_{\text{calc}}$  is linearly interpolated between two calculated points  $x_{\text{calc},j}$  and  $x_{\text{calc},j+1}$  around the measured point  $x_{\text{meas},i}$ . The  $\chi^2$  sum is normalised: dimensionless, and should ever become small compared to 1.

math    chi\_square\_log

The same as chi\_square but first the logarithm of (the absolute value of) all  $y_{\text{meas}}$  and  $y_{\text{calc}}$  is taken.

math    push

ABC A = [B , C] A is a concatenation of B and C. B is placed on top of C

math    constant

ABC A = c; Watch out: c is a scalar, A gets the same length as B. B is only used to set the length of A. AAC is allowed as well.

math    linear

AB A = [1, 2 ,3...]; A gets the same length as B. B is only used to set the length of A. AA is allowed as well.

math    rangeLin

A The first point A[0] and the last point A[nA-1] of the vector A are conserved. The points in between are scaled in a linear way between those end points.

math    rangeLog

A The first point A[0] and the last point A[nA-1] of the vector A are conserved. The points in between are scaled in a logarithmic way between those end points.

math    characteristics.eta

aBC Vector A contains the voltage data, vector B the current density data of a calculated IV curve. Scalar a will then contain  $\eta$ ,  $V_{oc}$ ,  $J_{sc}$ ,  $FF$ ,  $V_{mpp}$ ,  $J_{mpp}$  ( $V$  and  $J$  at the maximum power point). If you acquired the IV curve with get\_iv, then you can better use get\_characteristics.eta... (the scalar that should receive  $\eta$  has a more general format then here, where it can only be xvalue, yvalue,...). This function is

math    characteristics.Voc

math    characteristics.Jsc

math    characteristics.FF

math    characteristics.Vmpp

math    characteristics.Jmpp

math characteristics.all

useful if the *IV* curve was processed in the script (from series or parallel connection, all kinds of correction, ...)

ABC Works the same as the previous commands, but the results are stored in vector A:  $A[0] = \eta$ ,  $A[1] = V_{oc}$ , ...,  $A[5] = J_{mpp}$ .

The five math commands below are special: they require a composed value. The first part of the `fill...` commands is a vector: a letter from {x, y, z, u, v, w} (noted here as A), that stands for the corresponding script variable xvector, yvector,.... The next parts of the value must be separated by whitespace (space or tab) from the first part and from each other. They can be a number, or a SCAPS script variable. The first part of the `force_in_range` command is a scalar: : a letter from {x, y, z, u, v, w} (noted here as a), that stands for the corresponding script variable xvalue, yvalue, ... The format of the `series` and `parallel` commands are explained below.

math fillConstant

A constant n n is the number of points

math fillLinear

A startvalue stopvalue n n is the number of points

math fillLinear.FixedIncrement

A startvalue stopvalue  $\Delta x$   $\Delta x$  is the increment

math fillLogarithmic

A startvalue stopvalue n n is the number of points

fillLogarithmic.FixedNumberOfPoints

math fillLogarithmicPerDecade

A startvalue stopvalue n n is the number of points per decade

fillLogarithmic.PerDecade

math fillLogarithmic.FixedMultiplier

A startvalue stopvalue  $q$   $q$  is the multiplier

math force\_in\_range

a minvalue maxvalue the value of the scalar a is forced in the range [minvalue, maxvalue]

math series

ABCDEFn The script vectors CD and EF contain the *IV* curves of two cells (voltage in C and E, current density in D and F); both contain n datapoints. The *IV* data of the series connection is placed in the script vectors AB (voltage in A, current density in B).

math parallel

ABCDEFnxy The script vectors CD and EF contain the *IV* curves of two cells (voltage in C and E, current density in D and F); both contain n datapoints. The area of cell 1 is multiplied with the scalar weight factor x, the area of the

second cel with the weight factor y (only  $x+y=1$  really makes sense). The IV data of the parallel connection is placed in the script vectors AB (voltage in A, current density in B).

### Examples:

```
math filllinear x 1 3 5 results in xvector = [1.0, 1.5, 2.0, 2.5, 3.0]
math filllinear.fixednumbeofpoints y 2 8 4 results in yvector = [2, 4, 6, 8]
math filllinear.fixedincrement z 1 5 2.0 results in zvector = [1, 3, 5]
math filllinear.fixedincrement z 1 6 2.0 results in zvector = [1, 3, 5, 6]. Note that the last increment is not the 'fixed' one (=2), but that the stop value (=6) is respected instead.
math fillogarithmic u 10 1000 5 results in uvector ≈ [10, 31.6, 100, 316, 1000]
math fillogarithmic.perdecade v 10 2000 2 results in vvector ≈ [10, 31.6, 100, 316, 1000, 2000]
math fillogarithmic.fixedmultiplier w 1 20 2.0 results in wvector = [1, 2, 4, 8, 16, 20]. Note that the last multiplier is not the 'fixed' one (=2), but that the stop value (=20) is respected instead.
```

### 10.4.9 Loop commands

Syntax:

```
loop argument variable
```

On encountering a `loop start` command line, the internal script variables are set to: `loopcounter = 0` and `looperror = 1.0E30` (or the value of `looperror` that was set before).

The next script commands are executed until `loop stop` is met. Then, if `loopcounter < maxiteration-1` and `looperror > maxerror`, the internal script variable `loopcounter` is incremented, and the script is retaken from the preceding `loop start` command. Thus, when the error condition is never met, `loopcounter` will successively be set to 0 ... `maxiteration-1`, thus `maxiteration` values. The internal variables `maxiteration` and `maxerror` can be set with `set loop.maxiteration` and `set loop.maxerror` at any time, see §10.4.6.

There is no `set` command to set the internal script variable `loopcounter`. The variable `loopcounter` is internally set to zero on starting a loop, and then incremented with one each times the loop is run. The variable `looperror` can be set directly or be returned by the dll programme SCAPSUserFunction.dll (§10.4.13), that should be set-up by the user (one example of such dll is distributed with the SCAPS installation). Two of the `loop` commands are equivalent with a `set` command:

E.g. `loop maxiteration 20` is equivalent to `set scriptvariable.maxiteration 20`  
 E.g. `loop maxerror 1E-6` is equivalent to `set scriptvariable.maxerror 1E-6`

**Table 10.12** SCAPS loop commands

command	argument	value	default-directory
---------	----------	-------	-------------------

---

loop	start	no value
loop	stop	no value
loop	maxiteration	min=5; max=100; default = 25.
loop	maxerror	min=1E-8; max=1E25; default=1E-5

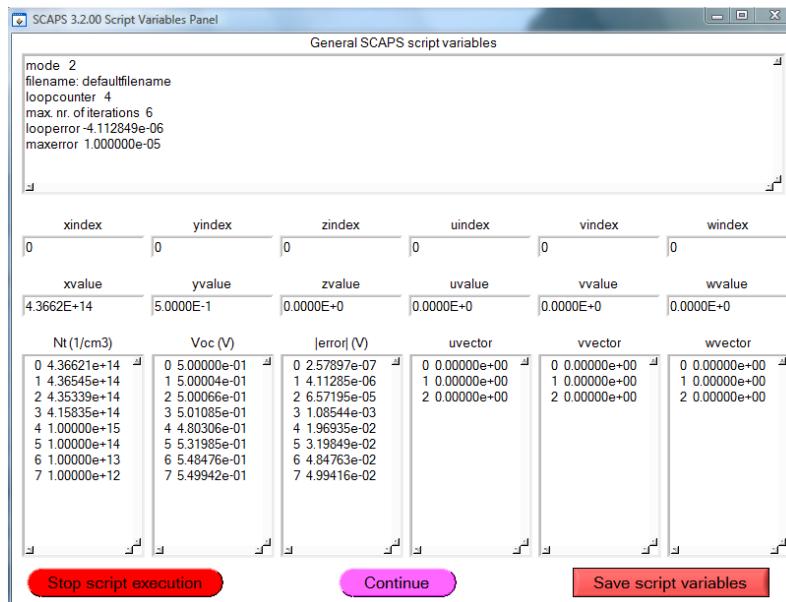
---

#### 10.4.10 Show command

There is only one show command. The command line to do this is:

```
show scriptvariables
```

These are shown on the screen (see §Figure 10.3), if errorhandling.toscreen is set, or to the standard error file, if errorhandling.appendtofile or errorhandling.overwritefile are set. This command is very useful in debugging your script files. Also, you can stop the execution of a script when the script variables are shown on the screen. When scripts are nested, you exit all scripts upon clicking ‘stop script execution’. You can comment out the show commands once the script is OK. The show scriptvariables panel is also available from the action panel (the SCAPS main panel) after execution of a script. The show command does not show all values of (very) long vectors, if the output is directed to the standard error file. In order to access these, you should use save scriptvariables.



**Figure 10.3** The script variables panel. When the panel is displayed after a show scriptvariables command, the script is interrupted and can now be aborted or continued.

#### 10.4.11 Plot commands

Syntax:

```
plot argument value
```

The plot command works in a similar way as the math command §10.4.8.

**Table 10.13** SCAPS plot commands

---

command	argument	value	remark
A,B and C represent a vector and should be chosen out of the set {X,Y,Z,U,V,W}			

---

---

plot draw	AB	Plot A on the abscissa and B on the ordinate
plot	AB	Identical to plot draw
plot drawversusindex	A	Plot the index $i$ on the abscissa and $A_i$ on the ordinate
plot clear		Clear the plots drawn by the script. Identical to clear plot

---

Whenever a `plot draw` command was met, a graph is added to the ‘script plot panel’, see Figure 10.4. This option allows you to make personalized graph of any data in SCAPS . This panel is shown immediately after the script calculation is finished and can also be accessed from the action panel.

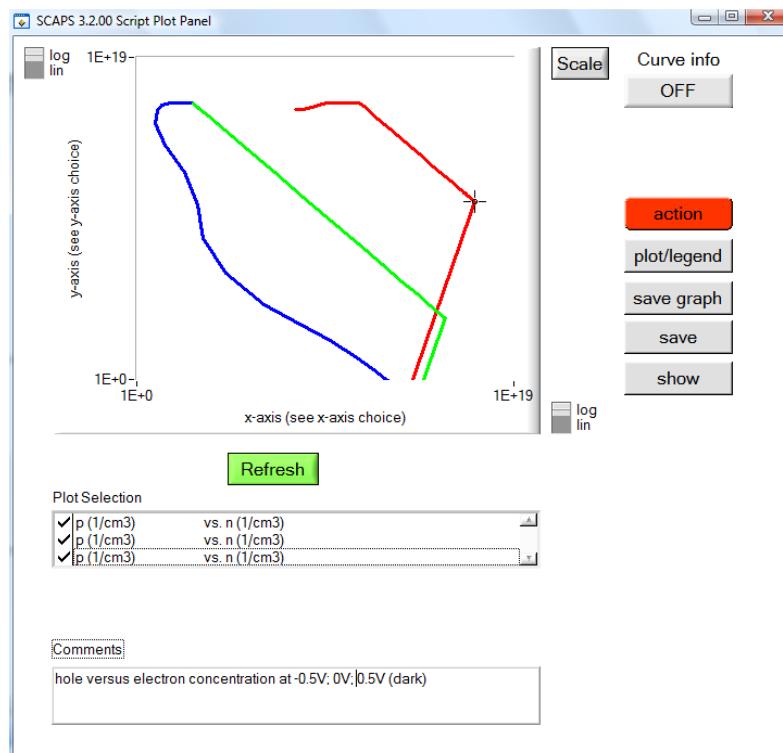


Figure 10.4 The script plot panel

#### 10.4.12 Calculate commands

Syntax:

`calculate argument`

This is equivalent with pressing one of the “Calculate”-buttons in the interactive user interface. If no argument is present the command gets interpreted as `calculate singleshot`

---

command	argument	remark
calculate	singleshot	this argument can be omitted
calculate	batch	
calculate	recorder	

---

### 10.4.13 The run commands

#### 10.4.13.1 Running SCAPSUserFunction.dll

This function is run by

rundll scapsuserfunction or

run dll scapsuserfunction or

run dll

(As of now, only one user dll is recognized is scaps, named SCAPSUserFunction.dll. The format of this command allows possible later addition of more dll's).

This dll is the method that SCAPS is using to implement two-way communication with the user. When you do not (want to) know how to write an own program and make a dll (dynamic link library) of it, you are restricted to use only the SCAPSUserFunction.dll as delivered with SCAPS, or not to use loops in a SCAPS script. The following information is for SCAPS users with programming skills. By writing their own SCAPSUserFunction.dll, they now can realize the following (in the formulation of an external SCAPS user):

*"I would need the possibility to do a simulation, evaluate the result with an external program and let it adjust the problem definition for the next simulation, do a simulation, and so on..."*

... well, this external program should be named SCAPSUserFunction, and be present as a dll file in the scaps/lib directory. When implemented in C or C++, this function must comply with the function definition:

```
int DLLIMPORT SCAPSUserFunction (int mode, double *xvalue, double
*xvalue, double *svector[6], int sn[6], double *looperror, char
*filename);
```

The keyword DLLIMPORT might be dependent on the development environment; here it is for LW/CVI of National Instruments.

The meaning of the other items is:

- SCAPSUserFunction: the name of the dll. The user must provide a SCAPSUserFunction.dll and SCAPSUserFunction.lib with this name, in the scaps/lib directory.
- int SCAPSUserFunction: the function should return an integer value, indicating the success of the function evaluation. SCAPS interprets 0 as ‘success’ and a negative value as a failure. This value is stored in the script-variable **status**, and shown in the error output (to screen or in the SCAPS error logfile).
- int mode: an integer that can be used to implement several strategies in one dll function. In the example delivered with SCAPS, mode = 1 or 2 means ‘find a root’ (e.g. find some  $N_A$  such that  $V_{oc} = 0.50$  V), and mode 3 or 4 means ‘find an extremum’ (e.g. find some  $N$ , such that  $\eta$  is maximal).
- double \*xvalue, double \*yvalue: (pointers to) two scalar values, passed to the function by reference, such that a new value of them can be returned by the function. Note with SCAPS 3.0.02: though there are now 6 scalar values xvalue, ..., wvalue, only xvalue and yvalue are passed to the SCAPS dll. Also, the new integer variables xindex, ..., windex, are not passed to the dll: thus, this dll remains compatible with earlier SCAPS versions.
- double \*svector[6]: array of (pointers to) one dimensional arrays, with dimensions specified in sn[]. These vectors correspond to the vectors xvector (=0), yvector (=1), zvector (=2), uvector (=3), vvector (=4), wvector (=5). These arrays can get new values in the function that is returned to SCAPS.
- int sn[6]: the dimension of the above vectors. These are passed by value, not by reference: their value cannot be updated and returned by the function.

- double \*looperror: a pointer to a scalar variable, that can be updated and returned by the function. In the SCAPS script processor, it is treated as the internal `looperror` variable. Returning its value by `SCAPSUserFunction.dll` is the only way to change `looperror` in a loop. Since the script processor only checks if  $|looperror| < maxerror$ , so you can also return a negative value here.
- char \*filename: a pointer to a string variable of max. 256 characters. The SCAPS script processor will treat it as a filename, that can be used to set e.g. a spectrum file, a generation file, a filter file,... with the `set` command.

To set up your own dll, you can use other variable names; however, the type, size and order of the variables must be exactly as specified here. Those not using C or C++ should use variable types of the same size (in bits) than the C types int, double, char, pointer.

**The users who want to develop their own dll, or to modify the existing dll (that is easier to start with `(S)`), should ask us for the source code: `SCAPSdll.c` and `SCAPSdll.h`.**

#### 10.4.13.2 Executing system commands in a script

The command line to do this is:

```
runsystem systemcommand or  
run system sytemcommand
```

where `systemcommand` is something that is recognized by MS-Windows as a valid command. These can be .exe files, .bat files or WINDOWS commands. Here you can run any of your own programmes (extension .exe; the arguments on the command line can be included), or any of your batch files (extension .bat).

Examples are:

```
runsystem myownopticalprogramme.exe inputfile1 inputfile2 outputfile  
runsystem myownwindowsbatchprogramme.bat  
runsystem print ivresults.iv
```

(in the last command, it is likely that Windows will need to know the full path and not only the filename...).

#### 10.4.13.3 Executing a script from within a script

The command line to do this is:

```
run script scriptfile
```

where `scriptfile` is a file containing a script. You can nest script files (that is, run a script file from within another script file) as you like, but that should not be a reason for exaggeration. All the script variables are transferred from one script to the other, with the exception of some loop-variables `loopcounter`, `looperror`, `maxerror`, `maxiterations`, which are local to each script file.



## References

- [1] M. Burgelman, P. Nollet, S. Degraeve, *Modelling polycrystalline semiconductor solar cells*, Thin Solid Films, 361 (2000) 527-532.
- [2] K. Decock, S. Khelifi, M. Burgelman, *Modelling multivalent defects in thin film solar cells*, Thin Solid Films, 519 (2011) 7481-7484.
- [3] M. Burgelman, J. Marlein, *Analysis of graded band gap solar cells with SCAPS*, Proceedings of the 23<sup>rd</sup> European Photovoltaic Solar Energy Conference, Valencia, 2008, pp. 2151-2155.
- [4] J. Verschraegen, M. Burgelman, *Numerical modeling of intra-band tunneling for heterojunction solar cells in SCAPS*, Thin Solid Films, 515 (2007) 6276-6279.
- [5] S. Degraeve, M. Burgelman, P. Nollet, *Modelling of polycrystalline thin film solar cells : new features in SCAPS version 2.3*, Proceedings of the 3rd World Conference on Photovoltaic Energy Conversion, Osaka, 2003, pp. 487-490.
- [6] A. Niemegeers, M. Burgelman, *Numerical modelling of ac-characteristics of CdTe and CIS solar cells*, Proceedings of the 25th IEEE Photovoltaic Specialists Conference, Washington DC, 1996, pp. 901-904.
- [7] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in C*, 2<sup>nd</sup> ed., Cambridge University Press, New York, 1992.
- [8] J. Verschraegen, S. Khelifi, M. Burgelman, A. Belgachi, 21st European Photovoltaic Solar Energy Conference, Dresden, Germany, Sept. 2006, 2006.
- [9] S. Khelifi, M. Burgelman, J. Verschraegen, A. Belghachi, *Impurity photovoltaic effect in GaAs solar cell with two deep impurity levels*, Solar Energy Materials and Solar Cells, 92 (2008) 1559-1565.
- [10] S. Khelifi, J. Verschraegen, M. Burgelman, A. Belgachi, *Numerical simulation of the impurity photovoltaic effect in silicon solar cells*, Renewable Energy, 33 (2008) 293-298.
- [11] K. Decock, P. Zabierowski, M. Burgelman, *Modeling metastabilities in chalcopyrite-based thin film solar cells*, Journal of Applied Physics, 111 (2012) 043703.
- [11'] M. Burgelman, K. Decock, S. Khelifi and A. Abass, *Advanced electrical simulation of thin film solar cells*, Thin Solid Films, 535 (2013) 296-301.
- [12] A. Niemegeers, S. Gillis, M. Burgelman, *A user program for realistic simulation of polycrystalline heterojunction solar cells: SCAPS-1D*, Proceedings of the 2<sup>nd</sup> World Conference on Photovoltaic Energy Conversion, Wien, 1998, pp. 672-675.
- [13] H.J. Pauwels, G. Vanhoutte, *Influence of interface states and energy barriers on efficiency of heterojunction solar-cells*, J. Phys. D-Appl. Phys., 11 (1978) 649-667.
- [14] J. Verschraegen, *Karakterisering en modellering met SCAPS van de CISCuT dunne-filmzonnecel*, dissertation Universiteit Gent. Faculteit Ingenieurswetenschappen, 2006.
- [15] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer Verlag, Wien-New York, 1984.
- [16] J. Marlein, M. Burgelman, Proceedings of NUMOS (Int. Workshop on Numerical Modelling of Thin Film Solar Cells, Gent (B), 28-30 March 2007). p. 227-233 2007, 2007.
- [17] T. Walter, R. Herberholz, C. Müller, H.W. Schock, *Determination of defect distributions from admittance measurements and application to Cu(In,Ga)Se<sub>2</sub> based heterojunctions*, Journal of Applied Physics, 80 (1996) 4411-4420.
- [18] K. Decock, S. Khelifi, S. Buecheler, F. Pianezzi, A.N. Tiwari, M. Burgelman, *Defect distributions in thin film solar cells deduced from admittance measurements under different bias voltages*, Journal of Applied Physics, 110 (2011) 063722.