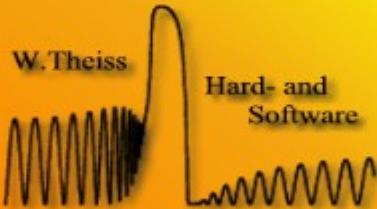
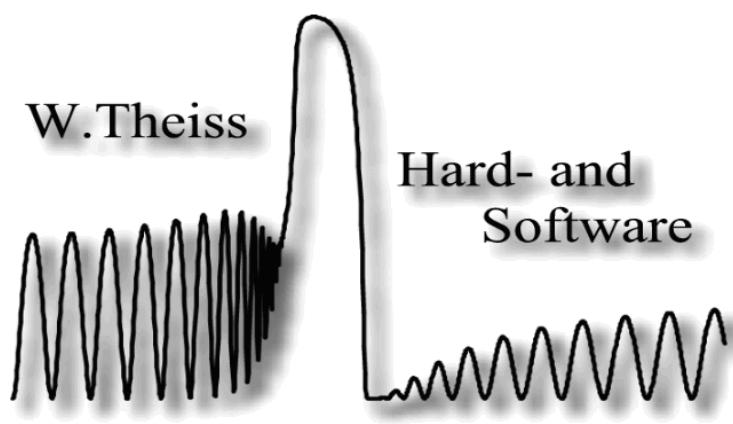


SCOUT



Technical manual

W.Theiss Hard- and Software for Optical Spectroscopy
Dr.-Bernhard-Klein-Str. 110, D-52078 Aachen
Phone: (49) 241 5661390 Fax: (49) 241 9529100
E-mail: theiss@mtheiss.com Web: www.mtheiss.com



SCOUT

Optical Spectrum Simulation

by Wolfgang Theiss

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 18.02.2011, 19:39 in Aachen, Germany

Table of Contents

Foreword	0
Part I Overview	2
1 About this document	2
2 About SCOUT	3
3 About the author	8
Part II Quick start: Getting around in SCOUT	10
1 Key commands in the main view	10
2 Working in the treeview level	11
3 Building a simple optical model	14
4 Generating the user interface	23
Part III Mastering the main window	33
1 Main window overview	33
2 Routine work	33
3 Manipulation of the displayed graphs	36
4 Working with the batch control window	38
Part IV The optical model	43
1 Overview	43
2 Defining Materials	43
Introduction	43
Properties	44
Overview	44
Refractive index	44
Absorption coefficient	45
Energy loss function	46
Others	46
Material types	46
Imported dielectric function: Fixed data sets	46
Importing data from the w orkbook	47
Importing data from text files	48
Dielectric function models	52
Introduction	52
Susceptibilities	53
Constant	53
Constant refractive index	54
Harmonic oscillator	54
Drude model	54
Extended Drude model	55
Brendel oscillator	57
Kim oscillator	58
Gervais oscillator	58
Formula	60
Imported data	61
KKR susceptibility	61

KKR susceptibility II.....	62
KKR dielectric function models.....	62
Introduction	62
Susceptibilities	63
Free formula.....	63
OJL2 model.....	64
Campi-Coriasso model.....	66
Tauc-Lorentz model.....	67
Master models	68
Introduction	68
Master parameters.....	69
Slave parameters.....	69
Fit parameters	72
Effective dielectric functions.....	72
Introduction	72
Maxwell Garnett.....	73
Bruggeman.....	74
Looyenga	74
Bergman representation	74
Database connection	76
3 Layer stacks	77
Introduction	77
Layer types	80
Halfspace	80
Simple layer	80
Thin film	82
Thick layer	82
Variable thickness.....	82
Thickness averaging.....	85
Rough interface.....	85
Superlattice	89
Concentration gradient.....	90
Anisotropic layer.....	96
Included layer stack	97
Assigning materials to layers	101
Working with pre-defined layer stacks	102
Thickness fluctuations	103
Wizards	105
Overview	105
Pane wizard.....	105
Foil wizard.....	112
Glass coating wizard.....	117
4 Spectra	117
Overview	117
Reflectance, Transmittance, ATR	118
Introduction.....	118
Spectrum normalization.....	121
Polarization of the incident radiation.....	122
Manipulation of the computed spectrum.....	122
Thickness determination by Fourier transform.....	122
Data smoothing.....	123
The spectrum type 'Layer mix'	124
Ellipsometry	126
Introduction.....	126
Graphics.....	126
Stack averaging	128
Layer absorption	129
Charge carrier generation	129

Photoluminescence	131
Introduction	131
Algorithm	133
Overview	133
Absorption of the incident radiation	133
Internal efficiency	135
Emission	135
User interface	137
Qualitative examples	139
Overview	139
Thickness variations	140
Additional layers	141
Electroluminescence	143
EL overview	143
EL example	145
Part V Parameter fitting	152
1 Parameter fit overview	152
2 Fit parameters, sliders and 'fit on a grid'	152
3 Master parameters	157
4 Fit parameter sets	157
5 Automatic fitting	158
6 Rating	159
7 Batch fit	159
Overview	159
Preparation of the tables	160
Go!	164
Inspecting the results	165
Part VI Optical functions	167
1 Materials	168
2 Layer stacks	168
3 Spectra	169
Part VII Special computations	171
1 ColorBox	172
2 Optical function fluctuation	174
3 Function fluctuation	176
Part VIII Data acquisition	180
1 Overview	180
2 File import	181
3 MiniScan	182
4 Minolta CM-2600d	183
5 OceanOptics spectrometer	184
6 Avantes spectrometer	188
7 tec5 spectrometer	188
Part IX Views	191

1	Introduction	191
2	List of views	193
3	Defining views	193
4	View objects	194
	Display items	194
	Rectangle	194
	Text view	195
	Bitmap view	196
	Metafile view	197
	Parameter view	199
	Field view	200
	Workbook display (columns)	201
	Workbook display (rows)	205
	Deviation view	206
	Rating view	206
	Sample view	206
	Layer stack view	207
	Action items	209
	Parameter slider	209
	Vertical bar	211
	Horizontal bar	211
	Next view button	212
	Previous view button	212
	Next configuration button	212
	Previous configuration button	213
	Acquire button	213
5	Printing views, clipboard copy	213
6	Automatic view generation	213
7	View examples	214
	View example: Simple spectrum fit	214
	View example: Fit and parameters	214
	View example: Display many spectra	215
8	Presentations	216
Part X	Configuring SCOUT for routine work	219
Part XI	Spatial and time distributions, optical monitoring	223
1	Overview	223
2	Electric field	223
3	Absorption	227
4	Optical monitoring	227
	R vs. total thickness during deposition	227
	R vs. time during deposition	230
	R vs. time during etching	231
Part XII	Working with the workbook	234
1	The SCOUT workbook	234
2	Generating graphs with the workbook	235
3	Display graphs of workbook data in a view	240

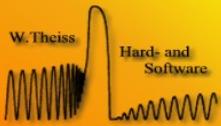
4 Workbook file formats	241
5 Actions involving the workbook	241
Parameter variation	241
Parameter fluctuation	243
Part XIII The database	247
1 Introduction	247
2 Materials	247
3 Layer stacks	248
Part XIV Technical notes	250
1 General features of SCOUT objects	250
2 List properties	250
3 Spectral range selection	252
4 User defined functions	253
5 Import formats	254
Overview	254
SCOUT standard format	254
xy-format	255
JCAMP format	256
SpectraCalc format (GRAMS)	256
Perkin-Elmer format	256
Perkin-Elmer ASCII format	256
Zeiss format	256
Opus format	256
HunterLab UltraScan	256
Hitachi ASCII format	257
WVASE32 ASCII format	257
Import from Excel	258
6 Export formats	259
Overview	259
7 Options	259
Part XV Tools	264
1 Overview	264
2 Data Factory	264
3 Collect	264
4 DirectDF	264
5 Digit	264
6 DF_DB	264
Part XVI Ole automation	267
1 Introduction	267
Introductory remarks	267
Installation	267
2 Automation object properties	268
a_filename (string)	268
absorption_coefficient	268
average_time (real number)	268
batch_configuration_file (string)	268

batch_fit_deviat	(sample:string) (float)	269
batch_fitparameter_name	(index: integer) (string)	269
batch_fitparameter_value	(sample: string, index: integer) (float)	269
batch_fp_set_stop_reason	(sample : string, fp_set_index: integer) (string)	269
batch_input_path	(index : integer) (string)	269
batch_name_of_spectrum	(i : integer) (string)	269
batch_number_of_spectra	(integer)	270
batch_prefit_result	(sample : string) (string)	270
batch_sample_parameter	(sample : string) (float)	270
batch_spectrum_filename	(sample : string, index : integer) (string)	270
batch_status	(integer)	270
color_coordinate	(spectrum_index,simulated,color_type,illuminant,observation_angle) (float)	71
configuration_file	(string)	271
dielectric_function_name	(i:integer) (string)	272
df_real_part	(n : integer; wavenumber : real number) (real number)	272
df_imag_part	(n : integer; wavenumber : real number) (real number)	272
experimental_spectrum_comment	(n:integer): string	272
experimental_spectrum_value	(n:integer;wavenumber:real number):real number	272
fft_thickness	273
fft_strength	273
file_format	(integer)	273
file_object_generation	(real number)	273
fit_deviation	(real number)	273
fit_parameter_mode	(i:integer) (integer)	274
fit_parameter_name	(i:integer) (string)	274
fit_parameter_set	(string)	274
fit_parameter_value	(i:integer) (real number)	274
fit_parameter_value_min	(i:integer) (real number)	275
fit_parameter_value_max	(i:integer) (real number)	275
fitting	(integer)	275
foundfiles	(integer)	275
found_filename	(i:integer) (string)	275
fp_set_name	(index : integer) (string)	275
get_sim_val1	(n : integer; wavenumber : real number) (real number)	275
get_sim_val2	(n : integer; wavenumber : real number) (real number)	276
grid_test_max	(single)	276
grid_test_min	(single)	276
grid_test_no_points	(integer)	276
grid_test_para	(integer)	276
incidence_angle	(i:integer) (real number)	276
kappa	(n : integer; wavenumber : real number) (real number)	276
layer_material	(n,m: integer) (string)	277
layer_thickness	(n,m : integer) (float)	277
layer_type	(n,m : integer) (string)	277
load_experiment	(n: Integer; const filename: String; format: Integer);	277
main_window_height	277
main_window_left	277
main_window_top	277
main_window_width	277
material_is_in_material_list	(material_name : string) (integer)	278
material_is_in_database	(material_name : string) (integer)	278
measurement	(i:integer) (string)	278
multidimensional_gridfit	(integer)	278
number_of_dielectric_functions	(integer)	278
number_of_fit_parameters	(integer)	278
number_of_fp_sets	(integer)	279
number_of_layer_stacks	279
number_of_layers_in_stack	(n:integer)	279

number_of_spectra (integer)	279
object_generation (real number)	279
optical_function(a_string : string) (real number)	279
optical_function_by_index(i : integer) (real number)	279
parameter (i:integer) (real value)	279
rating (string)	280
rating_threshold (level_index:integer) (float)	280
rating_description (level_index:integer) (string)	280
reference_spectrum(n:integer;formal:integer):string	280
refractive_index(n : integer; wavenumber : real number) (real number)	280
sample_description: string	280
simulated_spectrum_value(n:integer;wavenumber:real number):real number	281
simulation_exportfile(n:integer;formal:integer):string	281
spectrum_graphics_xmax(n : integer) (real number)	281
spectrum_graphics_xmin(n : integer) (real number)	281
spectrum_graphics_xticks(n : integer) (real number)	281
spectrum_graphics_xtitle(n : integer) (real number)	281
spectrum_graphics_xunit(n : integer) (real number)	282
spectrum_graphics_ymax(n : integer) (real number)	282
spectrum_graphics_ymin(n : integer) (real number)	282
spectrum_graphics_yticks(n : integer) (real number)	282
spectrum_graphics_ytitle(n : integer) (real number)	282
spectrum_name(i:integer) (string)	282
spectrum_range_max(n : integer) (real number)	282
spectrum_range_min(n : integer) (real number)	282
spectrum_range_points(n : integer) (integer)	282
spectrum_range_unit(n : integer) (string)	283
tolerance (real number)	283
weight_formula (spectrum_index, subindex : integer) (string)	283
3 Automation object methods	283
acquire	283
acquire_dark	283
acquire_reference	283
add_layer_definition_on_top(stack_index : integer;definition : string)	283
add_layer_on_top (stack_index : integer; layer_type,material:string;thickness:real number; thickness_unit : string)	283
add_to_report(text : string)	285
automatic_spectrum_scaling(i : index)	285
batch_add_sample (name: string)	285
batch_clear_samples	285
batch_restore_fit_results (sample : string)	285
batch_go	285
clear_fit_parameters	286
clear_layer_stack (index : integer)	286
clear_material_list	286
clear_report	286
create_fit_parameter (list, object, subobject, subsubobject, parameter : integer) (copy)	286
df_load_from_file(n: Integer; const filename:String);	288
df_load_from_file_2(n: Integer; const filename, unit_string:String);	289
df_load_from_x_n_k_table(n:integer; const filename : string);	289
df_load_from_x_n_k_table_2 (n:integer; const filename : string);	289
export_simulated_spectrum(n:integer; filename:string; format : integer)	290
df_to_clipboard (n : integer)	290
export_measured_spectrum(n:integer; filename:string; format : integer)	290
export_simulated_spectrum(n:integer; filename:string; format : integer)_2	290
grid_test	291
hide	291
hide_statusbar	291
import_measured_data_via_variants(index : string; spectral_positions, spectral_data : variant; unit_string : string)	291

load_df(n : integer, filename : string)	291
load_material_from_database (name : string)	291
load_experiment (n:integer, filename: string, format:integer)	292
prepare_shutdown n	292
print_fit(i : integer)	292
rename_material (old_name, new_name : string)	292
report_to_clipboard	292
report_to_file(filename:string)	292
save_configuration(filename : string)	292
save_df(n : integer, filename : string)	293
scout_normal	293
scout_on_top	293
searchfiles(dir,pattern : string)	293
send_message(the_text : string)	293
set_bottom_halfspace_material (i : integer; name : string)	293
set_global_range(min, max : real number, no_points : integer, spectral_unit : string)	293
set_spectrum_simulation_range(n : integer; min, mix : real number; no_points : integer; spectral_unit : string)	
set_spectrum_x_axis(i : integer; min, max, tick_spacing : float)	294
set_spectrum_y_axis(i : integer; min, max, tick_spacing : float)	294
set_top_halfspace_material (i : integer; name : string)	294
show	294
show_statusbar	295
show_maximized	295
show_minimized	295
show_normal	295
smooth_data	295
smooth_data_like_batch_control	295
spectrum_to_clipboard (n:integer)	295
start	295
stop	295
update_data	295
update_plot	296
view_to_clipboard_wmf	296
view_to_clipboard_bmp	296
4 Controlling SCOUT from Excel	296
Introduction	296
Table structure	296
SCOUT menu items in Excel	298
Excel macros	300
Overview	300
Application example	300
Part XVII References	302
1 Articles related to optical spectroscopy	302
Index	304

SCOUT

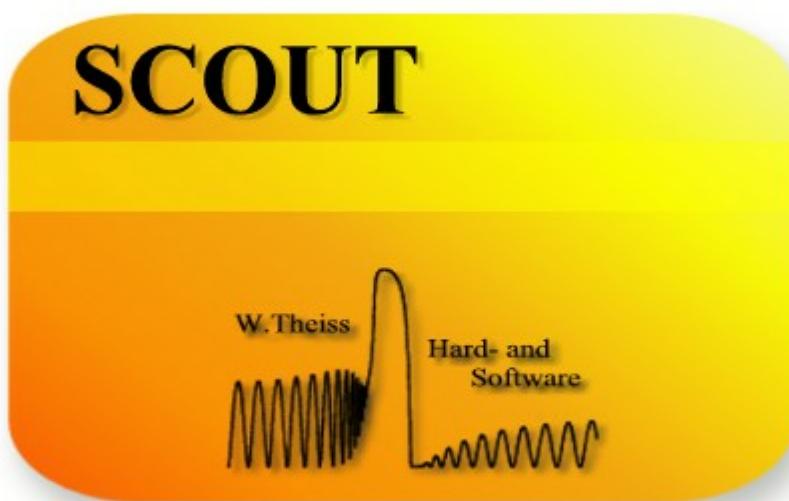


Part

I

1 Overview

1.1 About this document



Spectrum interpretation by simulation

written by [W.Theiss](#)

W. Theiss – Hard- and Software for Optical Spectroscopy
Dr.-Bernhard-Klein-Str. 110, D-52078 Aachen, Germany
Phone: + 49 241 5661390 Fax: + 49 241 9529100
e-mail: theiss@mtheiss.com web: www.mtheiss.com

July 2008

The goal of optical spectroscopy is - in most cases - the determination of microscopic quantities such as resonance frequencies of oscillating atoms, impurity concentrations or thin film thicknesses from macroscopic experiments. This is possible if the microscopic phenomena are coupled to the electric fields of the probing radiation. The key property of a material representing this connection is

the dielectric function or its square root, the complex refractive index - both quantities are more generally called 'optical constants'.

In the case of layered systems a very powerful technique to interpret optical spectra is given by a simulation of the experiment (based on a physical model) and adjustment of the model parameters to fit the measured data. Fortunately, rather simple models of optical constants lead to realistic simulations of optical spectra. This enables fast parameter fits from which the wanted information can be obtained.

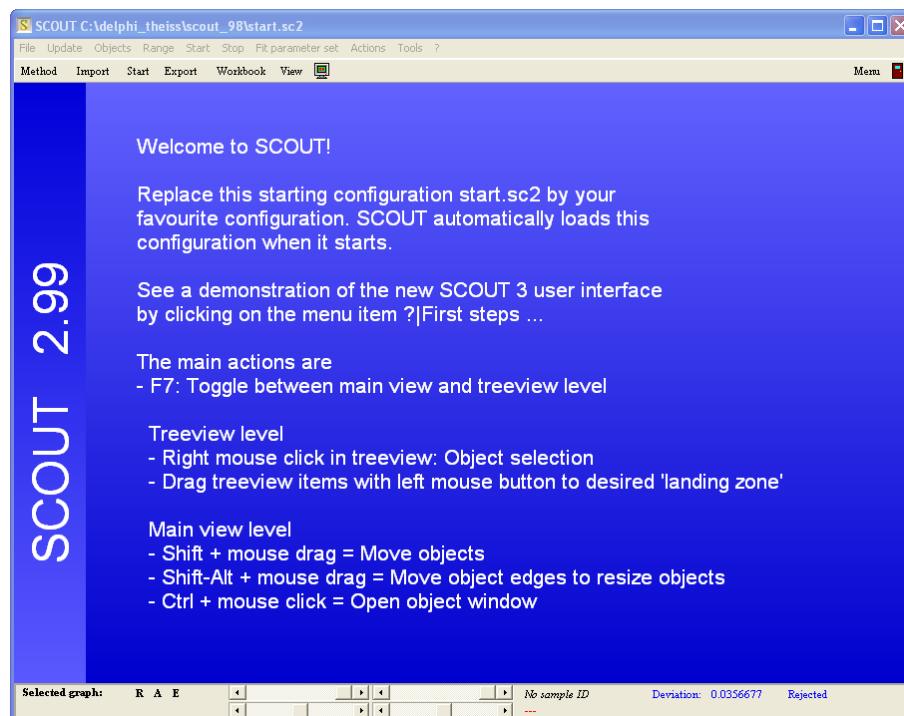
The interpretation of optical spectra is the central task of SCOUT - this manual gives technical information about the program features. Discussions of successful SCOUT applications are found on our homepage www.mtheiss.com and in the tutorials accompanying this document.

The [next section gives](#) an overview on this manual - from there you can jump to the individual parts of this documentation.

This text was written using the program Help&Manual (from EC Software, <http://www.helpandmanual.com>). With this software we produce the printed manual as well as the online help and HTML code for the internet version of the document - with exactly the same text input! This is a very productive feature and makes the development of the documentation quite easy. However, for this reason the printed manual may sometimes contain some 'strange' text fragments which seem to have no relation to the rest of the text. These might be hypertext jumps in the online help system which - of course - lose their function in the printed version of the manual.

1.2 About SCOUT

SCOUT is a Windows NT/2000/XP/Vista software for the analysis of optical spectra by computer simulation. At program start the main window opens which may look like this:

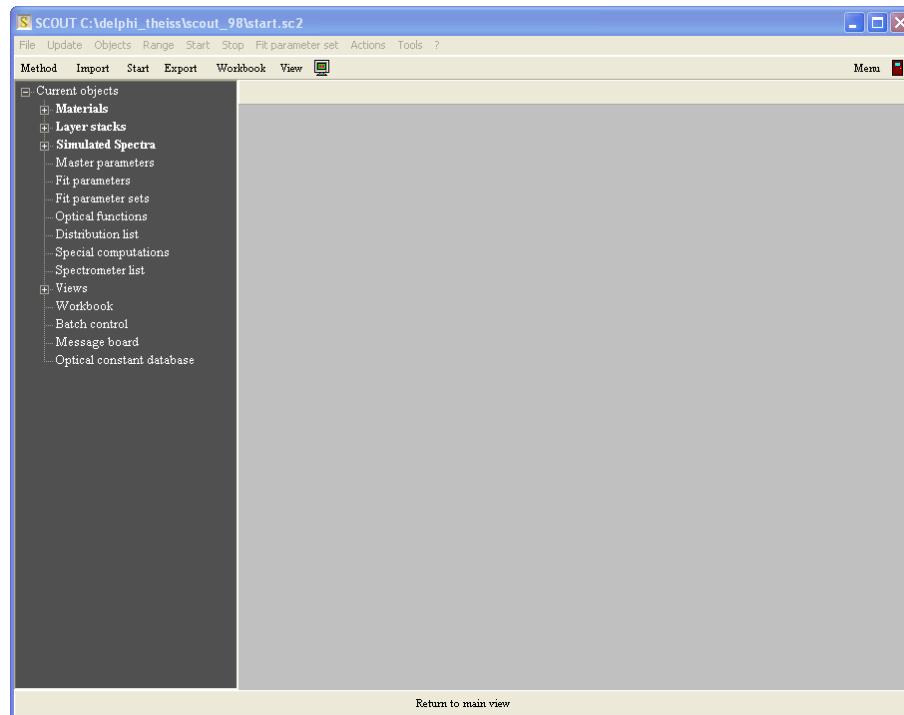


The main window has 2 levels. When the program starts, you see the so-called 'main view level' which consists of the

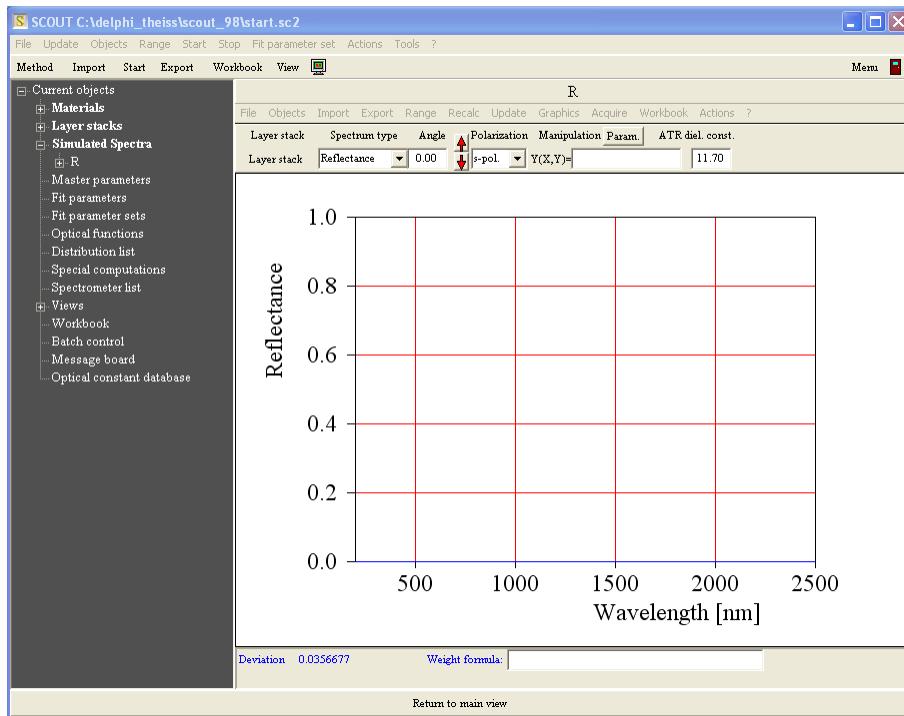
- menu, from which all parts of the program can be reached
- the toolbar (below the menu), featuring selected commands for easy access

- the drawing area (the blue part in the example above) showing the main view (text, spectra, parameter values or illustrative graphs, see [user-defined views](#))
- and the status bar at the bottom (including information and some control elements for the displayed graphs)

Pressing **F7** the main window switches to the 'treeview level' which looks like this:



On the left side you see a treeview of all objects in the current SCOUT configuration. With a right click on an object in the tree you can open it - most objects will display their controls or graphs in the right gray section. Whereas some objects will display only a few items or even nothing, others will show a graph, buttons and an object menu. Here is the example of an opened spectrum window:



Pressing **F7** again switches SCOUT back to the main view level. Alternatively, you can press the large '**Return to main view**' button at the bottom of the window.

See the chapter '[Quick start: Getting around in SCOUT](#)' for a deeper user interface description.

Working with pre-defined configurations, using the main window only

If you belong to the lucky people that can work with pre-defined SCOUT configurations, and you don't have to develop optical models of your own, it might be sufficient for you to only read the section about [using the main window](#). However, if you are eager to know how the program works, you can read the other parts of this manual as well. This will certainly help to improve your SCOUT skills.

Required steps for developing optical models

Advanced spectrum simulation is not easy. Setting up realistic optical models, you have to define a large number of parameters which reflect the complexity of the problem. Concerning its structure, SCOUT tries to be as close to the physical problem as possible and groups the many parameters of spectrum simulation according to the scheme described below.

1. Define the optical constants of all relevant materials

All optical constant definitions are done in the so-called material list which you can access from the main window by pressing F7. Materials are managed in the first branch of the object tree called **Materials**. Here you can define objects representing [fixed optical constants](#) (imported from files) or [optical constant models](#) that give you flexibility if you want to adjust optical constants to fit experimental data.

Optical constant models are composed of susceptibility terms which are managed in susceptibility lists. At present you can use the following types:

- [Drude model](#) for free carriers
- [Extended Drude model](#) for free carriers with frequency dependent damping
- [harmonic oscillator](#)
- [extended oscillator model suggested by Brendel](#)
- [extended oscillator model suggested by Kim](#)
- [oscillator model suggested by Gervais](#)

[OJL interband transition model](#)
[Campi-Coriasso interband transition model](#)
[Tauc-Lorentz interband transition model](#)
[user-defined expressions for optical constants](#)

Both fixed optical constants and models can be imported easily from the built in [database of optical constants](#).

For the description of heterogeneous materials (two-phase composites) you may want to mix optical constants. This can be done in SCOUT using [various effective medium concepts for inhomogeneous materials](#). Besides the classical [Maxwell Garnett](#) and [Bruggeman](#) approaches we offer the [Looyenga](#) formula and the general [Bergman representation](#) which is certainly the most advanced concept.

2. Define the structure of the layer stacks

The tree branch **Layer stacks** shows the list of layer stacks. In most cases you will investigate one layer stack only, but SCOUT lets you work with several stacks if you want to (or have to).

Each layer stack may consist of an unlimited number of layers which are again managed in lists, called [layer stack](#). For [simple layers](#) you can select [coherent or incoherent superposition of partial waves](#). Several types reflect a certain choice: A [Thin film](#) is always handled with coherent superposition whereas a layer of type [Thick layer](#) is treated with incoherent superposition. For cases in between full coherent or incoherent superposition we have implemented an efficient averaging algorithm for lateral [layer thickness inhomogeneities](#). You can define optical superlattices, corrections for scattering losses at [rough interfaces](#) and [gradually changing optical properties](#). [Anisotropic layers](#) (with heavy restrictions at the moment) may also occur in the stack.

3. Define which type of spectra are to be simulated and compared to experiments

After the definition of all optical constants and layer stacks you have to tell SCOUT which spectra it should simulate and compare to experimental data. These spectra are collected in the spectra list which is the **Simulated spectra** branch in the treeview. The following types of spectra can be used at present:

- [Reflectance](#)
- [Transmittance](#)
- [Absorbance](#)
- [ATR](#)
- [Layer Mix](#) (computes the average spectrum of a patterned sample)
- [Ellipsometry](#)
- [Ellipsometry stack average](#)
- [PL spectrum](#)
- [EL spectrum](#)
- [Layer absorption](#) (computes the absorbed fraction of incident light intensity in a certain layer in a stack)
- [Charge carrier generation](#) (computes the number of photon-generated charge carriers in a layer)

4. Select the fit parameters

The treeview branch **Fit parameters** holds the [list of fit parameters](#). These are the

parameters of the model that are to be adjusted in order to reach optimal agreement between simulation and measurements. Besides fitting to experimental data, you can also select a model parameter as fit parameter if you want to compute its value by a user-defined function (which may contain master parameters, optical function values or the time).

5. Fit of the model (manual, visual, automatic)

SCOUT lets you [fit model parameters](#) manually (type in new values) or visually, i.e. by mouse-driven sliders. With the 'fit on a grid' feature you can automatically scan a certain parameter range for the best starting value for the final completely automatic optimization. Using sequences of [fit parameter sets](#) (available in the treeview branch **Fit parameter sets**) you can develop efficient optimization strategies which enable SCOUT to adjust many fit parameters completely automatic. In cases for which completely automatic parameter fitting is possible you can process a whole series of input spectra in a [batch operation](#). SCOUT will import the spectra, do the fit, and store the obtained results in appropriate tables.

Enhancing your SCOUT work

Customizing SCOUT for routine spectroscopy using Views and the toolbar

Once you have developed a convincing optical model that reproduces your measurements in a high quality you may want to make that solution accessible to other people which are probably less experienced. In this case, you can hide all the complexity of the model and show in the main window only the information relevant to the final result. Using [Views](#) and the flexible [toolbar](#) you can achieve this goal and create nice and simple user interfaces.

Inspecting the influence of model parameters on optical spectra

Some automated actions help you to find out how model parameters affect optical spectra. You can vary one selected parameter on a set of equidistant values (in a user-defined range) using the [Parameter variation](#) action. SCOUT will compute all defined spectra for the set of values, and you can generate quite useful plots visualizing the parameter influence. In order to check how production tolerances influence the optical properties of your multilayer stack, you can compute the spectral variations for the case that some model parameters fluctuate randomly in user-defined intervals ([Parameter fluctuation](#) action).

Working with databases

You can work with SCOUT very efficiently making use of [databases](#). A SCOUT database contains materials with pre-defined optical constants and pre-defined layer stacks. If you save your favourite substrates and materials to your personal database these objects can be used to quickly build SCOUT configurations from scratch.

You can also benefit from the large collection of optical constants which is stored in the SCOUT database.

Programming SCOUT by OLE automation

You can control the actions of SCOUT from external programs by [OLE automation](#). Create automated reports and batch operations with the Windows Scripting Host (WSH), MS Word, MS Excel, LabView or any other OLE automation controller. Using OLE automation you can even hide SCOUT completely, feed it with spectra and have it doing the analysis in the

background while your data acquisition hardware collects already the next set of spectra .

To get a quick start into the details of spectrum simulation you should follow the introductory SCOUT tutorials (which are distributed with SCOUT and also available on our homepage www.mtheiss.com in the [support section](#)).

You can also have a look at the following documents which are distributed as PDF documents on the SCOUT CD (folder docs). You can also download these documents from our homepage using the link http://www.mtheiss.com/?Support:Downloads:Things_to_read:

- **Analyzing optical spectra by computer simulation - from basics to batch mode** (a PDF-document about spectrum simulation, summarizing the background and typical applications of this technique)
- **Developing optical production control methods for SCOUT** (a PDF-document about developing SCOUT methods. Good to read if you are looking for a thin film production control solution - even if you are not developing the method yourself but let us do the work)

1.3 About the author

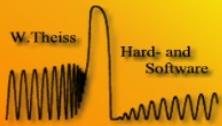
Responsible for all good and bad program features:

Wolfgang Theiss
Consultant for Optical Material and Process Control
Dr.-Bernhard-Klein-Str. 110

D-52078 Aachen
Germany

Phone: + 49 241 5661390
Fax: + 49 241 9529100
e-mail: theiss@mtheiss.com
web: www.mtheiss.com

SCOUT



Part

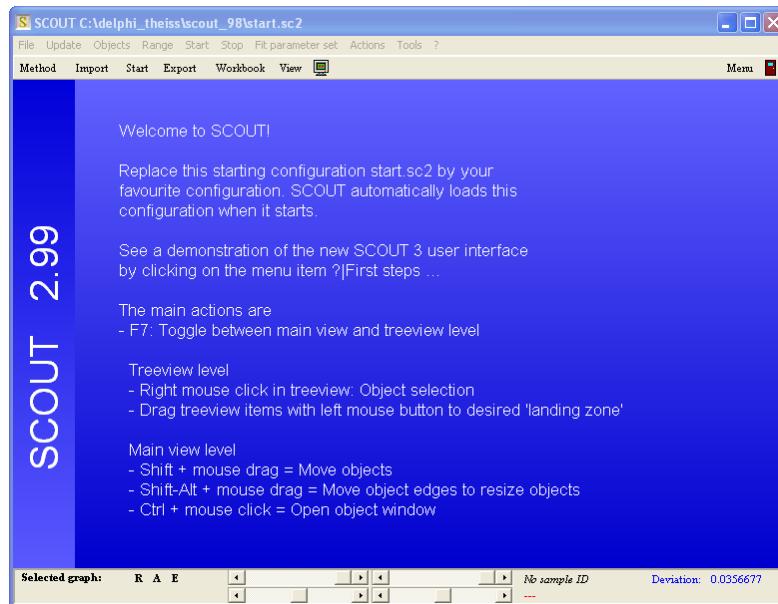
II

A large, solid gray circle is positioned to the right of the word "Part". Inside the circle, there are two thick, white, vertical bars of equal height, spaced evenly apart, resembling a double-lined colon or a play button symbol.

2 Quick start: Getting around in SCOUT

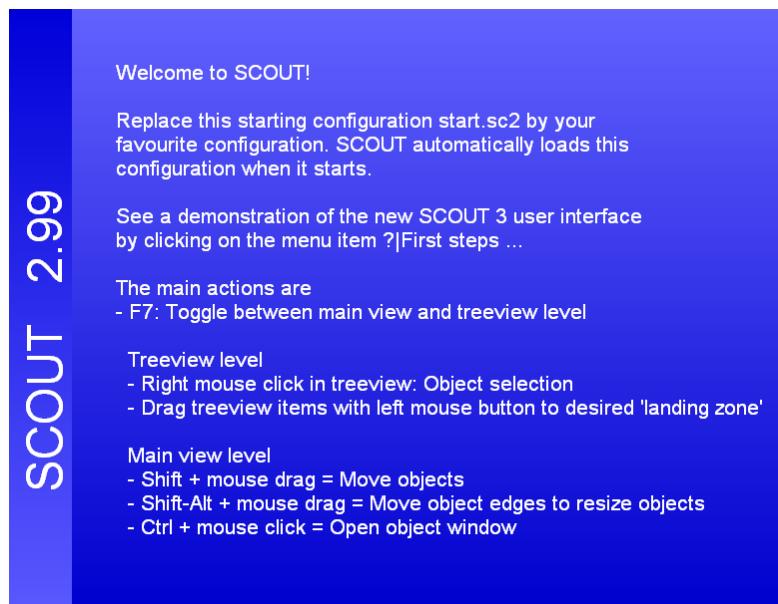
2.1 Key commands in the main view

Having started SCOUT, you see the main view:



The following key commands can be used to configure the main window:

- Pressing **m** you can switch on and off the menu
- The key **b** switches on and off the toolbar
- With **s** the statusbar at the bottom is shown or hidden
- Both the toolbar and the window frame and caption can be hidden or shown pressing **p** (presentation mode). If you switch off the menu and the statusbar as well and the SCOUT window is maximized, the whole screen is filled with the main view (presentations are discussed [below](#)):



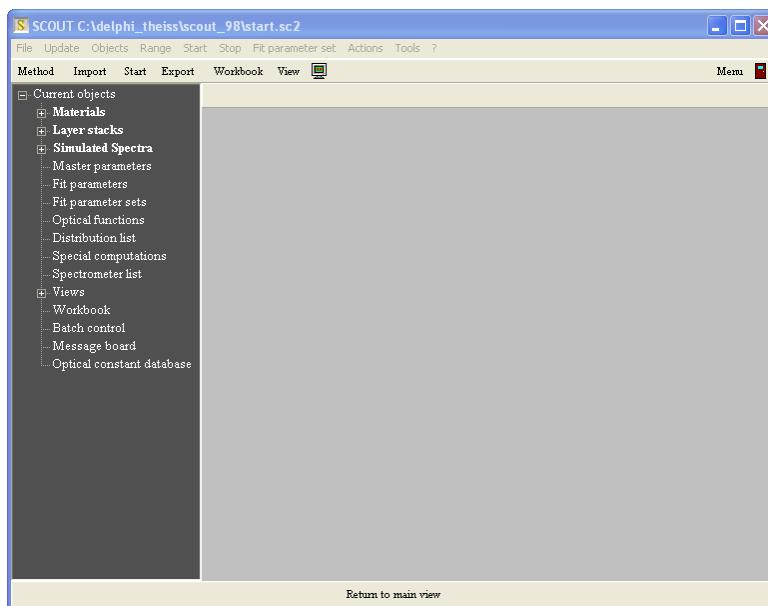
- If you have a SCOUT configuration which is configured to perform a parameter fit, you can start

and stop the fit pressing the **f** key.

- Pressing **e** opens the list of view elements of the view which is currently displayed.
- A time evolution of a model (this will be explained later) can be started and stopped with **t**.
- In order to setup a new configuration or to modify an existing one, press **F7** to switch from the main view level to the treeview level. The treeview level is discussed in the [following section](#).

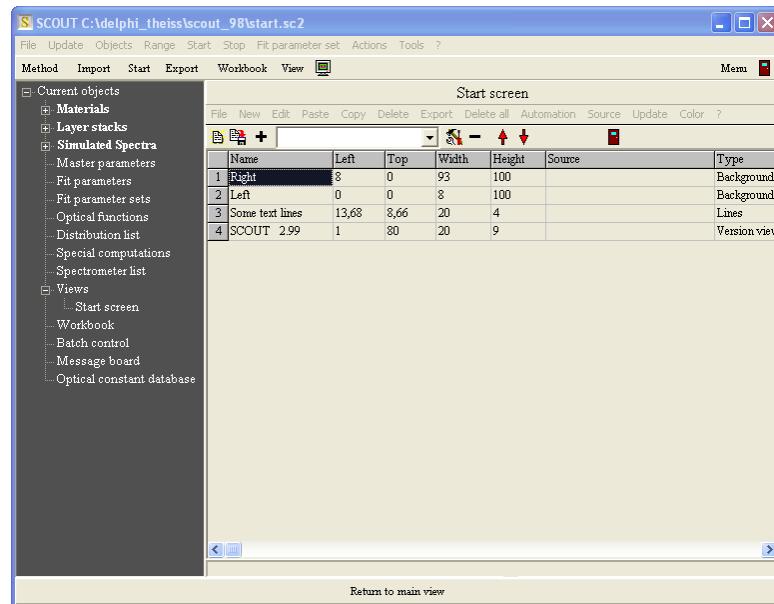
2.2 Working in the treeview level

Having pressed the F7 key in the main view you enter the treeview level which is used to make changes to the SCOUT configuration:



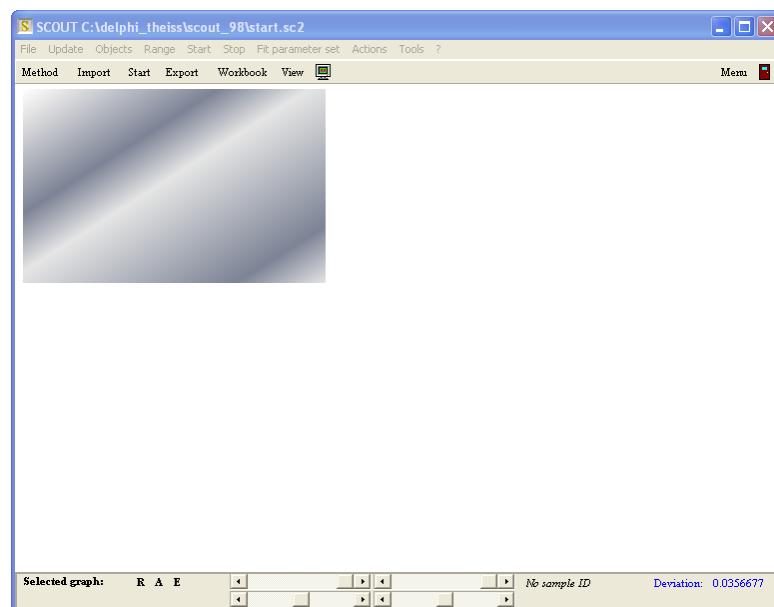
We will setup a simple SCOUT model now in order to show how to work in the treeview level. The goal is to compute the reflectance spectrum of a silver layer on a glass substrate, load experimental data and fit the silver thickness. The relevant items are to be placed in the main view in order to give a good overview.

Let's start to modify the main view: The current view items will be deleted and a new background picture will be loaded. In order to do so, open the item **Views** in the treeview (by clicking at the plus sign to the left) and right-click the item 'Start screen'. On the right side the list of view items appears:

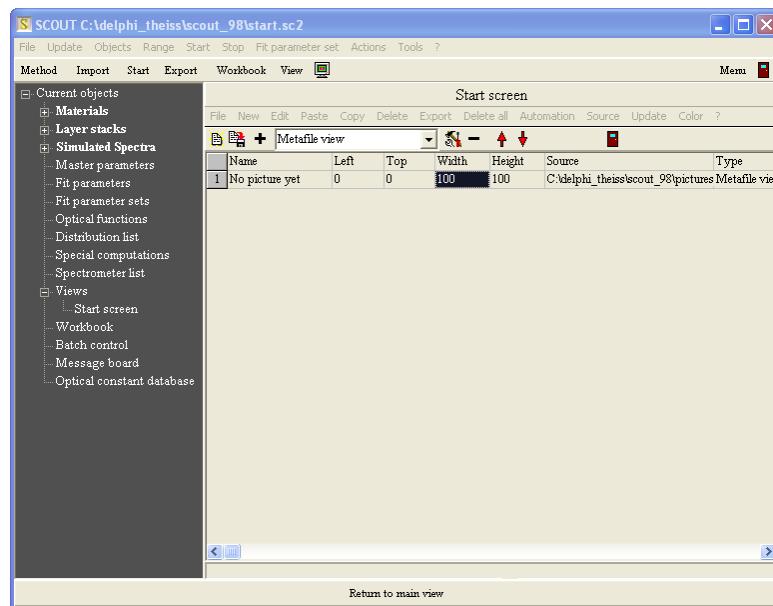


Note that the view item list has a menu and controls of its own - in fact, it's a frameless window of its own, displayed inside the area of the main window. Use the menu item **Delete all** to delete all items. Then select in the dropdown box to the right of the + sign the new object type 'Metafile view'. Then press the + sign or use the menu command **New**. Click on the new object in the list which is called 'No picture yet' and use the menu item **Edit** to change the object. Alternatively you can right-click the list item in order to edit it.

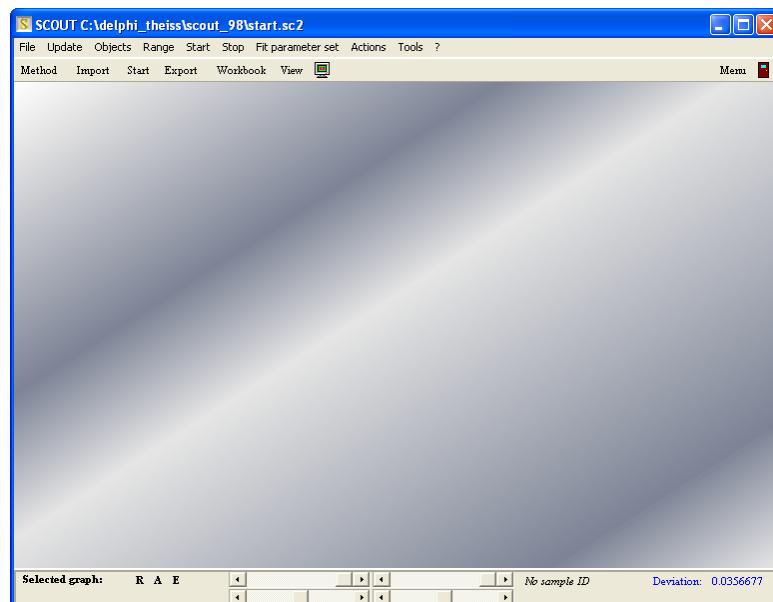
Editing a metafile object means to load a metafile graphics from a datafile. Select the file 'silver.emf' in the subfolder **pictures** in the SCOUT program directory. Then press F7 which switches back to the main view level:



In order to get a background picture, the graph should fill the whole main view. The size and position of a view element (all view elements are rectangles) is given by the parameters **Left**, **Top**, **Width** and **Height** in the list of view elements. These numbers are percentages with respect to the full size of the view (current values are 1,1,40,40). Press F7 again and type in the numbers 0, 0, 100, 100 as shown below (having typed in a number, press **Return** to close the edit mode of this parameter):



Now press F7 again:



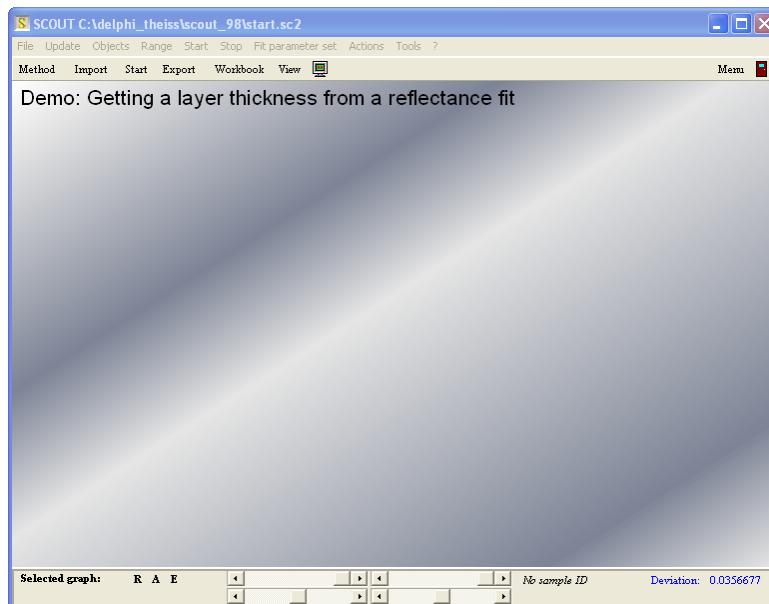
Now we will add some text. Press F7 and generate a new view item of type 'Text view': Select this type in the dropdown box and then use **New** to generate an object of this type. The new object is called 'Dummy name'. Select it and press **Edit**. You can now enter the text to be displayed. Type in 'Demo: Getting a layer thickness from a reflectance fit':



The next dialog lets you define a tilt angle. Keep the value of 0° for the moment - we do not want to rotate the text. Finally a color selection dialog is shown. Set the color to black which means we will have black text.

Make sure that the text view object in the list of view items is positioned underneath the metafile view object. The view elements in the list are painted from top to bottom. Hence background objects must be on top of the list so that they are painted first. You can change the position of an object in a list by using the red **Up** and **Down** arrows.

Pressing F7 should give something like this now:



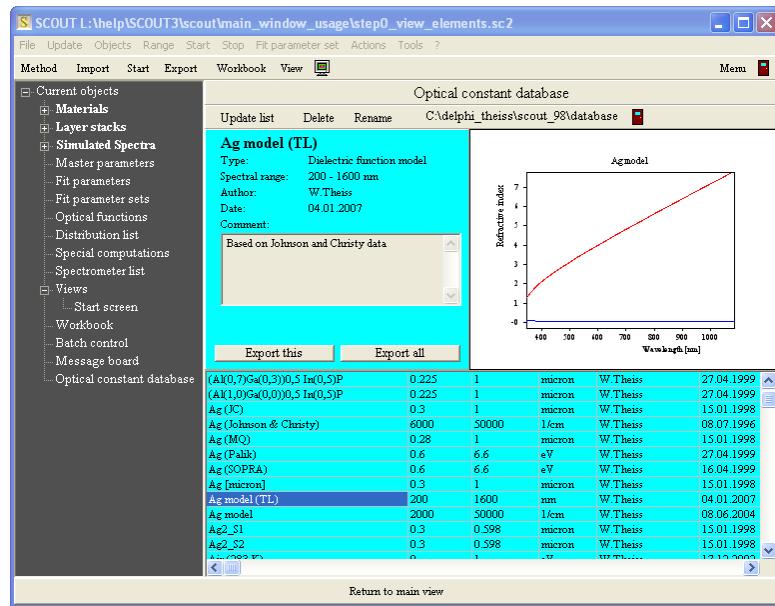
In the next steps, we will define the optical model and show the reflectance spectrum and the layer stack in the main view.

2.3 Building a simple optical model

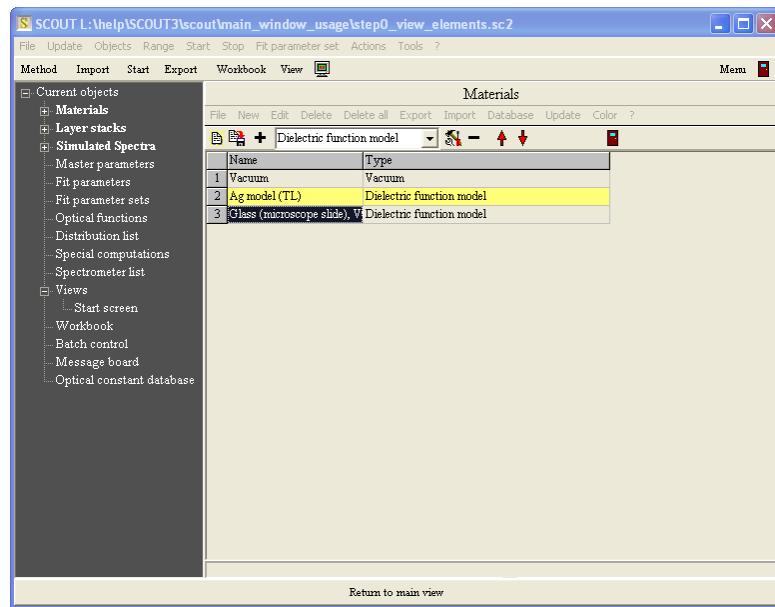
We will now setup the optical model the following way:

- Take the optical constants of glass and silver from the database
- Compose the layer stack
- Setup the reflectance spectrum
- Select the fit parameters

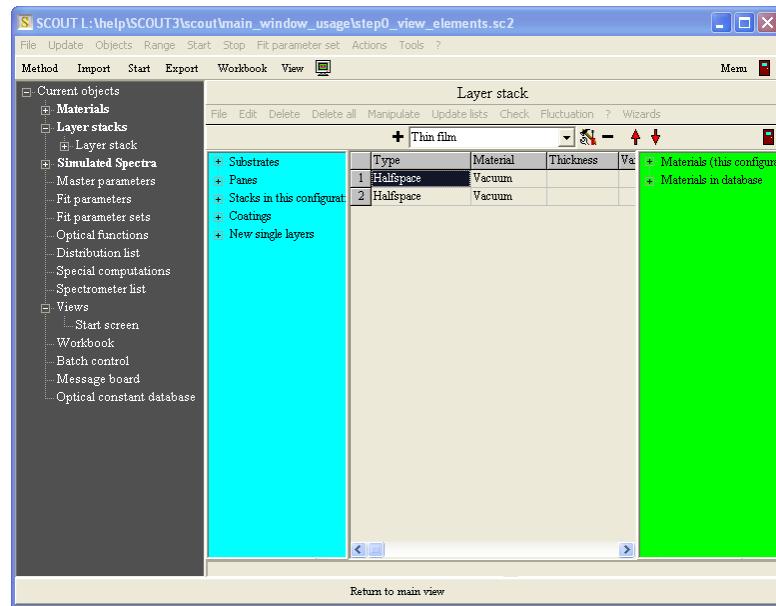
All required actions are done in the treeview level. Right-click the treeview branch 'Optical constant database' and select in the list of database materials the one called 'Ag model (TL)':



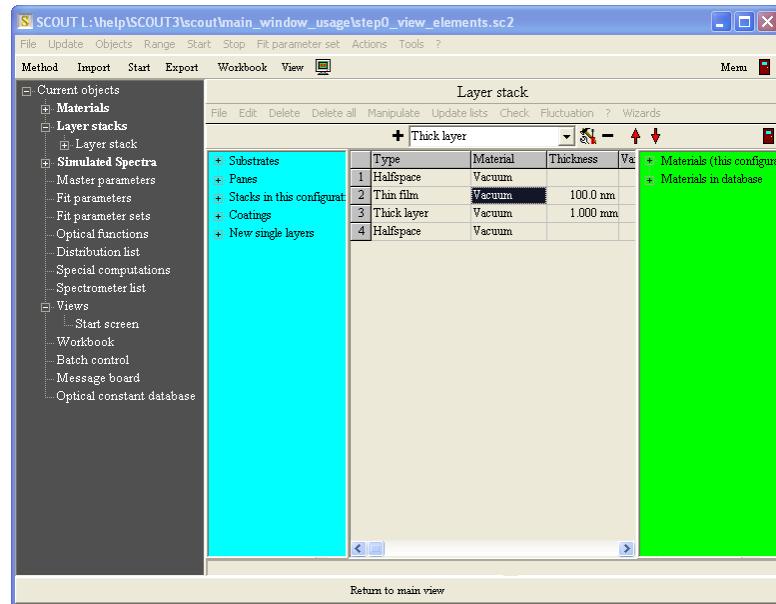
With the left mouse button, drag&drop it to the treeview branch called **Materials**. Repeat this step for the material called 'Glass (microscope slide)', Vis'. Now right-click the treeview branch **Materials** in order to open the current list of materials. It should look like this now:



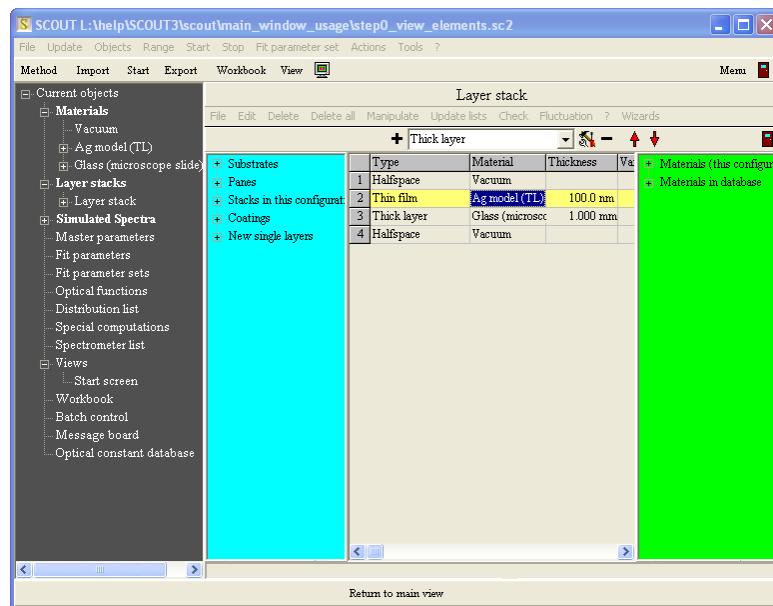
With these materials in our configuration, we are now ready to define the layer stack. Open the treeview branch **Layer stacks**. This list may contain several layer stacks. In this simple example, we need a single stack only. Right-click the treeview item 'Layer stack' which should look like this at the moment:



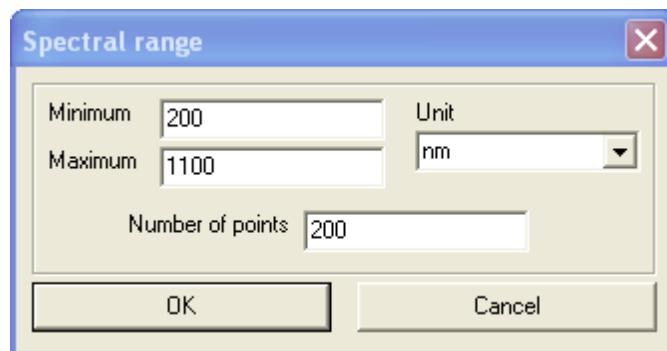
Select the bottom halfspace in the layer stack and use the **+** button in order to create a layer of type 'Thin film'. This will be used to represent the silver layer. Afterwards, select the new layer type 'Thick layer' and generate an object of this type which we will use for the glass substrate. Using the red arrow buttons, bring the objects into the following order:



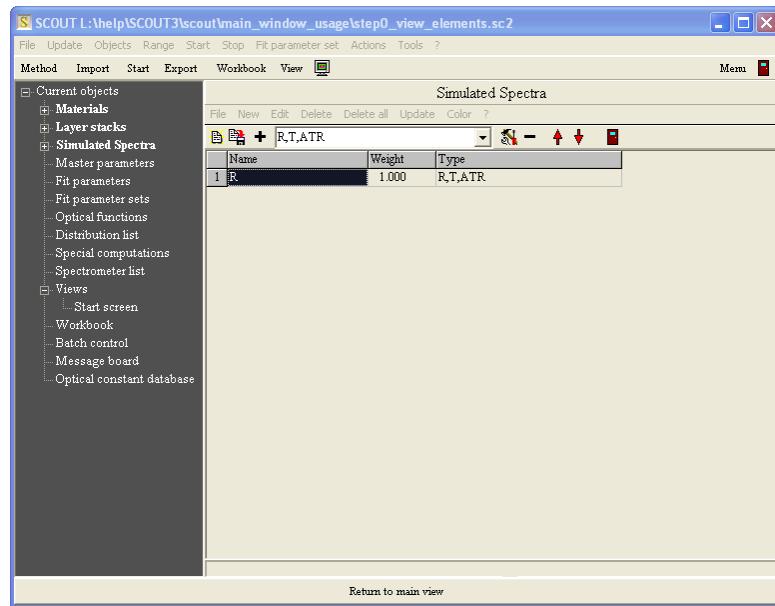
We will now assign materials to the individual layers. Open the **Materials** branch in the treeview. Drag the treeview item 'Ag model (TL)' to layer #2 in the stack and drop it there. Repeat this drag&drop for the glass material. The layer stack should look like this after these actions:



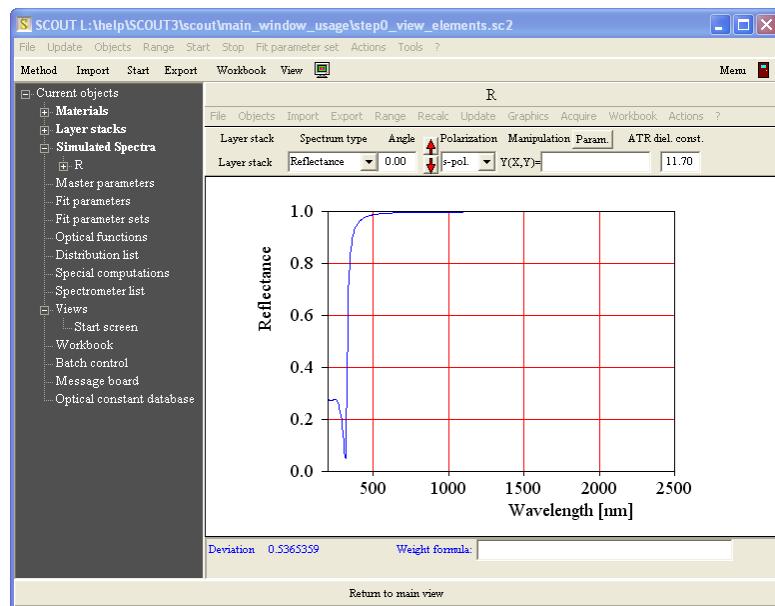
Having defined all required materials and the layer stack, we can now compute the reflectance spectrum of the layer stack. The experimental data that we are going to load have been measured in the spectral range 200 ... 1100 nm. Hence it makes sense to compute the optical model in this range. Since the loaded materials are models for optical constants (indicated by their type 'Dielectric function model') it is useful to compute the material models as well as the spectrum in the same wavelength range 200 ... 1100 nm. We can achieve this by using the global **Range** command in the main menu of SCOUT. A spectral range dialog appears and you should do the following settings:



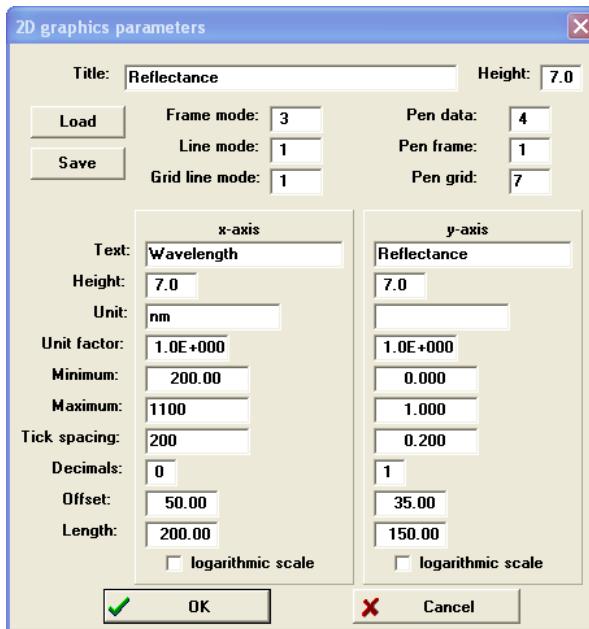
The reflectance spectrum must be defined in the list of simulated spectra. Open the corresponding treeview branch which contains an object already (if not in your situation, please select the new object type 'R, T, ATR' in the dropdown box and generate an object of this type):



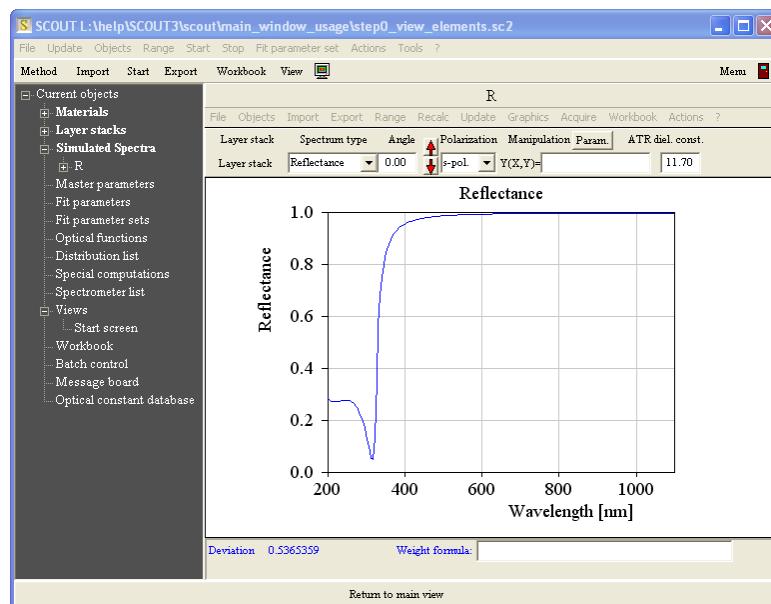
In order to have a closer look at the spectrum, open the tree branch 'Simulated spectra' by clicking at the + symbol and right-click the object 'R':



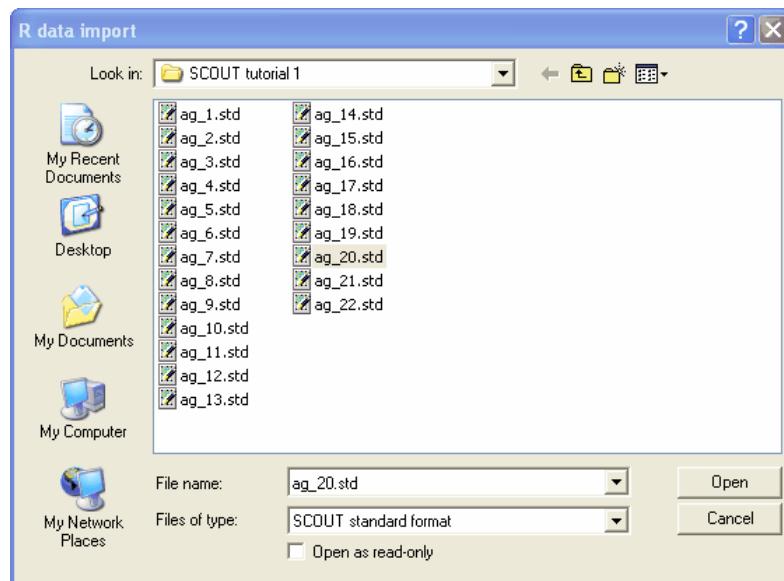
This shows already the wanted reflectance spectrum (the angle of 0° is a good approximation in the present case). We only have to adjust the axis settings for the graph which we can do using the local menu item 'Graphics|Edit plotparameters'. Please use the following settings in the graphics parameters dialog:



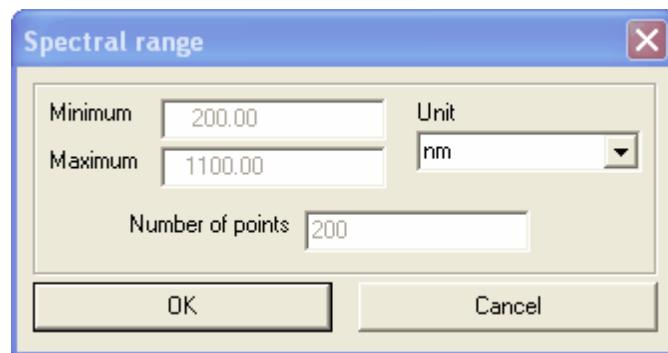
After pressing OK the graph is updated and looks like this now:



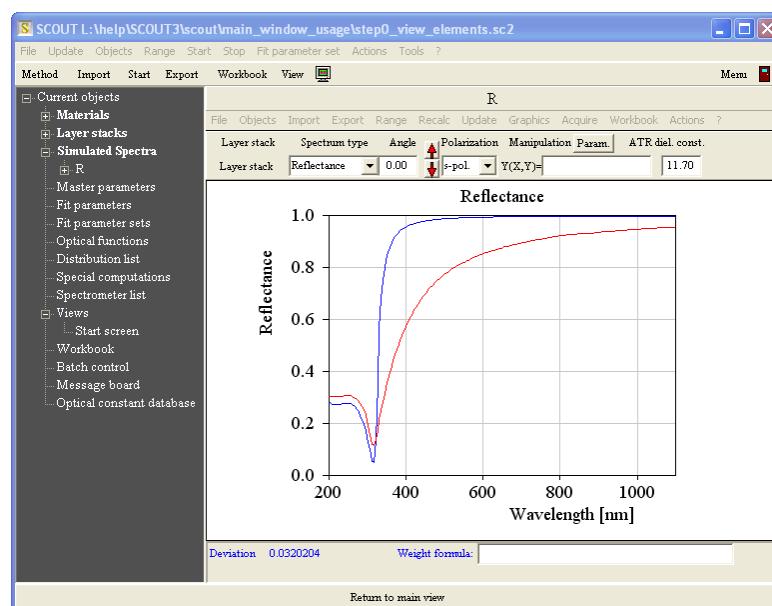
The optical model is ready now. The simulated reflectance can be compared to measured data. Use the local menu command **Import from file** to load experimental spectra from data files. The spectra we are using here are distributed with SCOUT tutorial 1. Open the file ag_20.std from the SCOUT tutorial 1 subfolder using the file type 'SCOUT standard format' as indicated in the following screenshot:



Make sure that you select the spectral unit **nm** in the following dialog which shows the spectral range covered by the data found in the file:

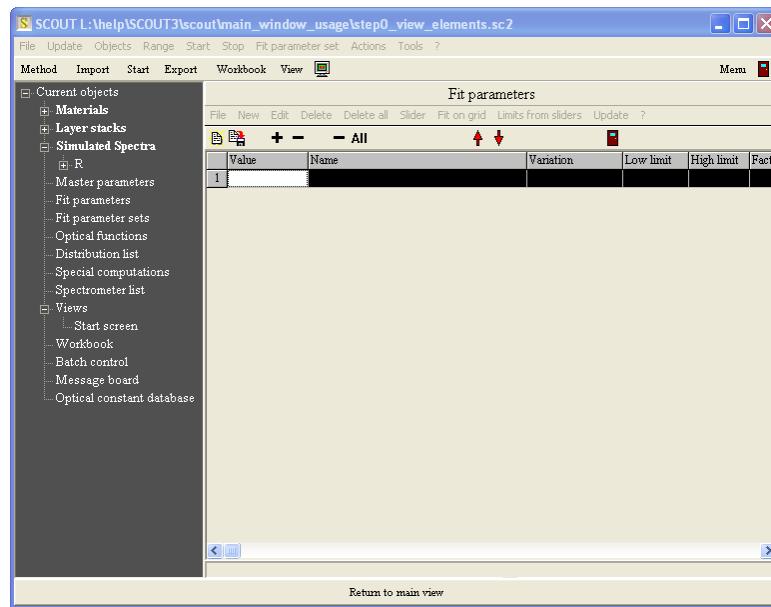


After pressing OK the spectrum window looks like this now:

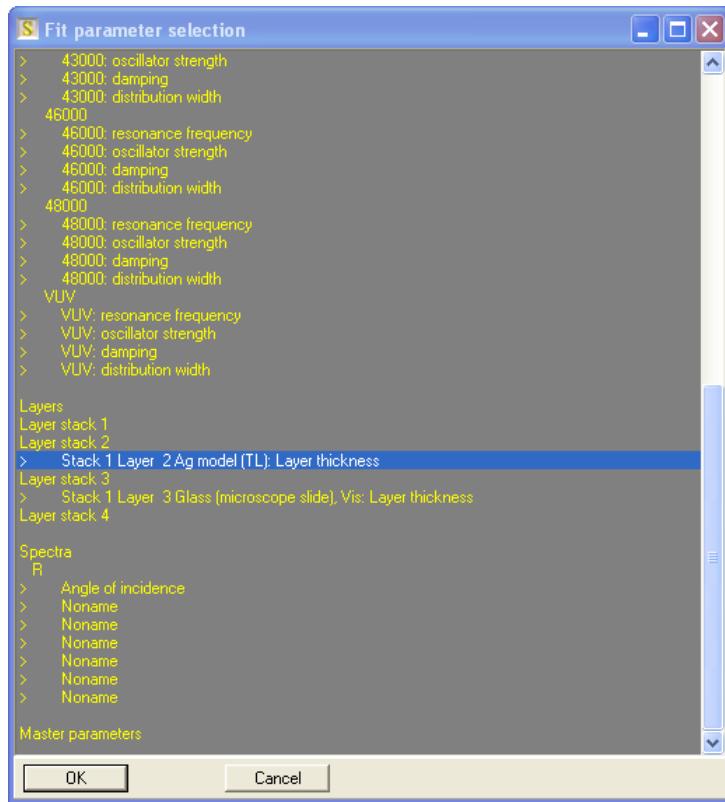


The red curve shows the loaded experimental data whereas the blue one represents the current model.

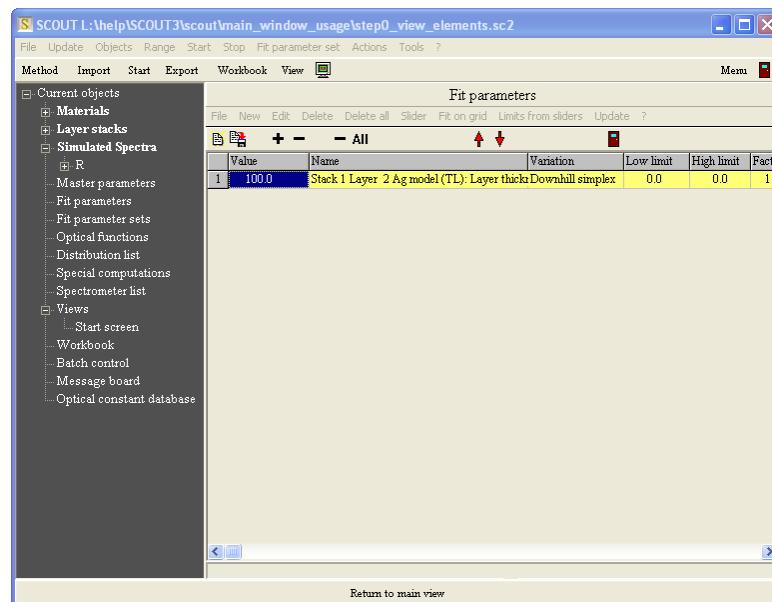
Obviously there is a significant disagreement between model and measurement which is a good reason to modify the model. The thickness of the silver layer is the parameter of the model which needs to be adapted in order to improve the agreement. We have to declare the silver thickness to be a fit parameter in order to do an automatic parameter adjustment (fit). Right-click the treeview branch **Fit parameters**:



Note that the list of fit parameters has no dropdown box in order to specify object types. There is only one type of objects in this list. Press the **+** button and you will get a dialog which displays all current model parameters. Select the silver thickness as shown below (while holding the **ctrl-key** down, you can select and de-select multiple items in this dialog by mouse-clicks):

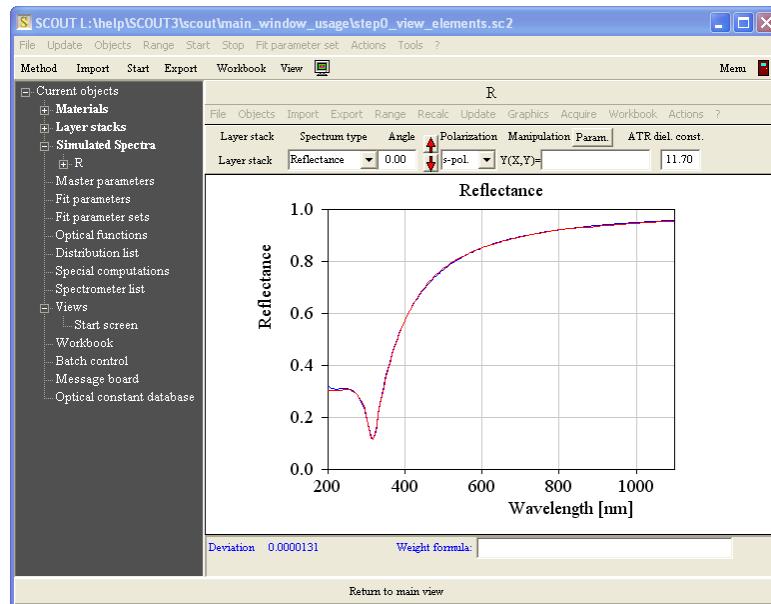


With this selection, leave the dialog pressing OK. The list of fit parameters is now the following:



The current problem is quite simple, and we can hope that the automatic fit routine of SCOUT can find the best value for the thickness without further help. Press **Start** in the toolbar to launch the fit routine (this button becomes a red **Stop** button while the fit is running). After a very short time the fit finishes and the red **Stop** button becomes a black **Start** button again.

Right-click the spectrum object R in the treeview and see the following fit result:

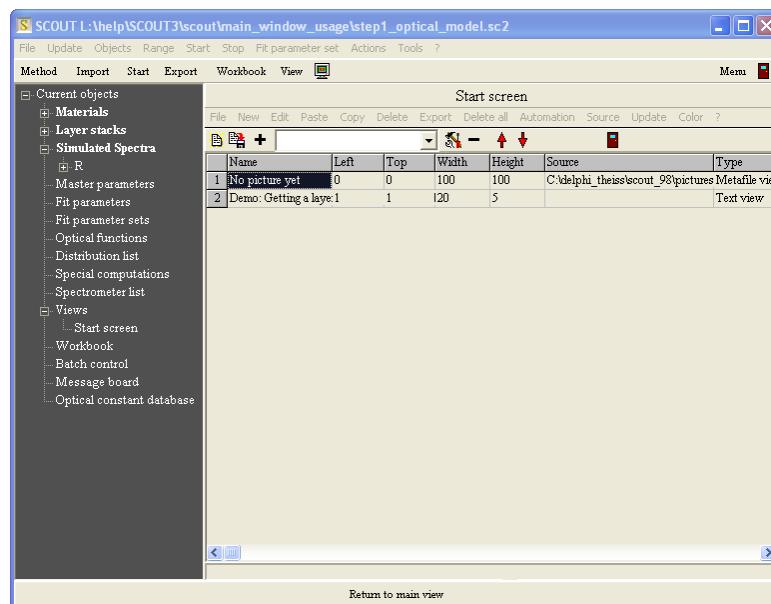


We have now an optical model matching the experimental results. If we want to do parameter fits for several experimental input spectra it might be useful to build a simple user-interface which offers the required actions in the main view. This will be done in the next section.

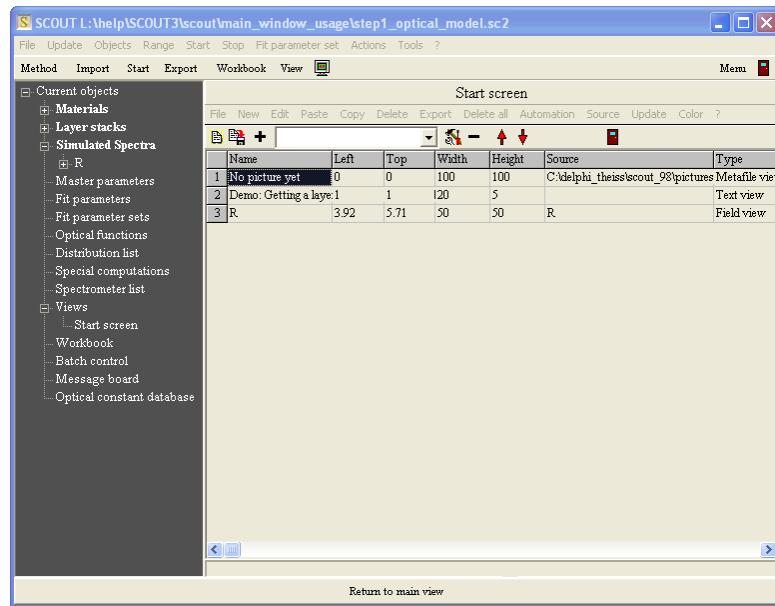
2.4 Generating the user interface

We will now configure a simple user-interface in the main view of SCOUT which will show the reflectance spectrum, the layer stack, the value of the silver thickness and some information on the fit quality. There will also be a slider for manual thickness variation. For all these things we will generate view elements which are able to draw the wanted items in the main view.

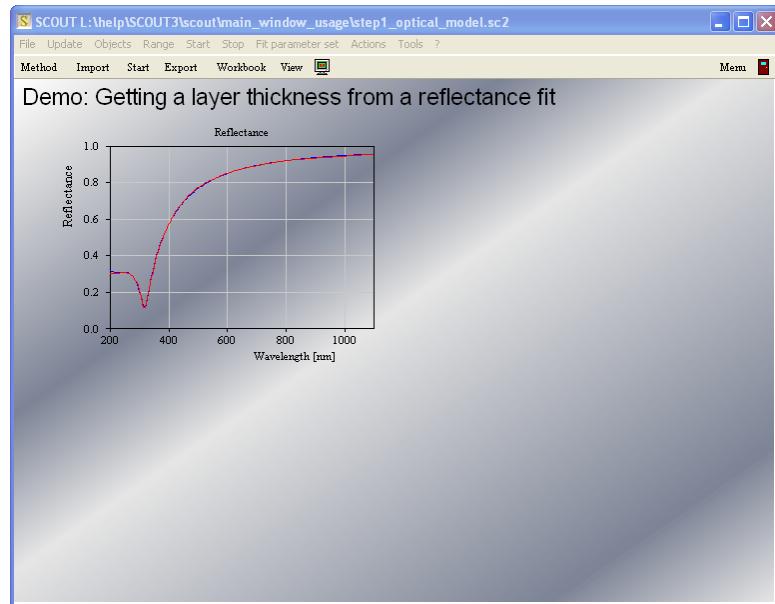
Let's start with the display of the reflectance spectrum. Open the treeview branch **Simulated spectra**. In addition, open the treeview branch **Views** and there the subbranch **Start screen**. SCOUT should look like this now:



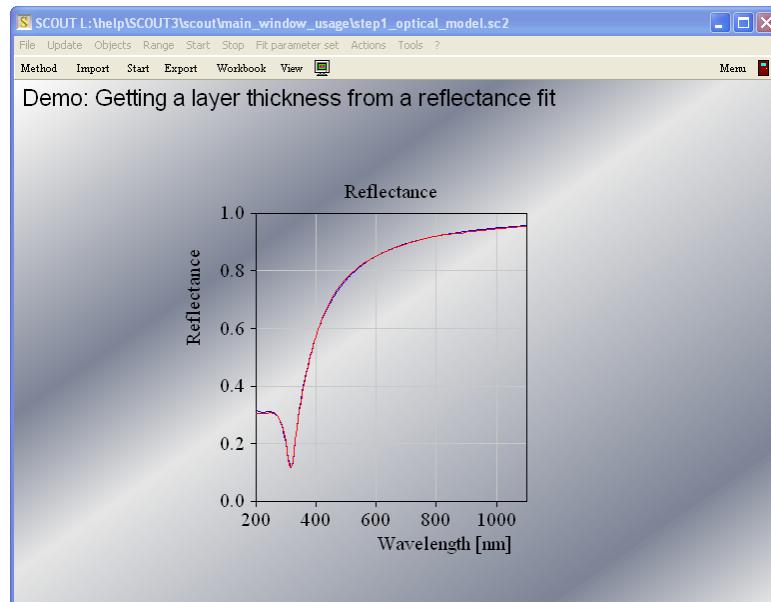
A view element that displays the reflectance spectrum can be generated by a drag&drop operation from the treeview to the list of view elements on the right. Start to drag the treeview item 'R' and drop it in the list to the right:



Press F7 to see the new view element in the main view:

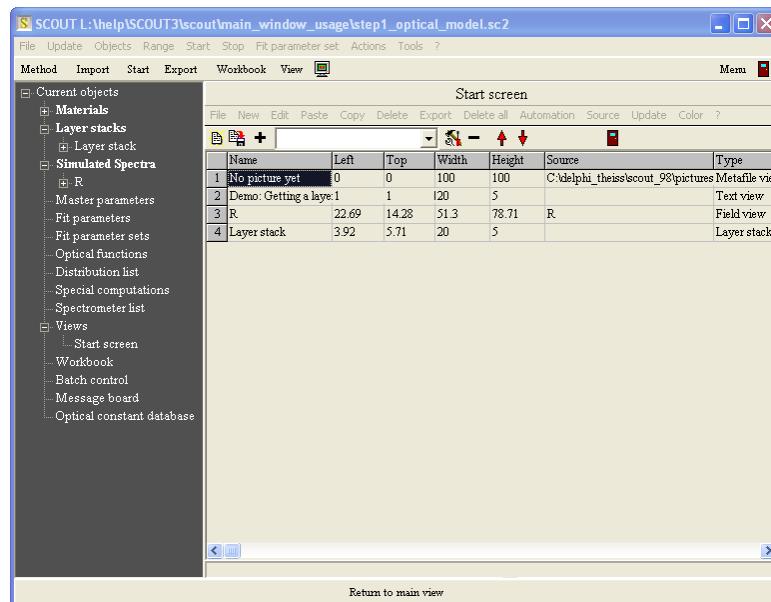


View elements can be moved within the main view with the **mouse** while the **Shift** key is pressed down. Try to move the reflectance spectrum to the center of the window. Holding both the **Shift** and the **Alt** key simultaneously down, you can move the left, top, bottom or right side of the rectangle individually by mouse drags. This is a very convenient re-size mechanism. Try to position the spectrum like in the following example:

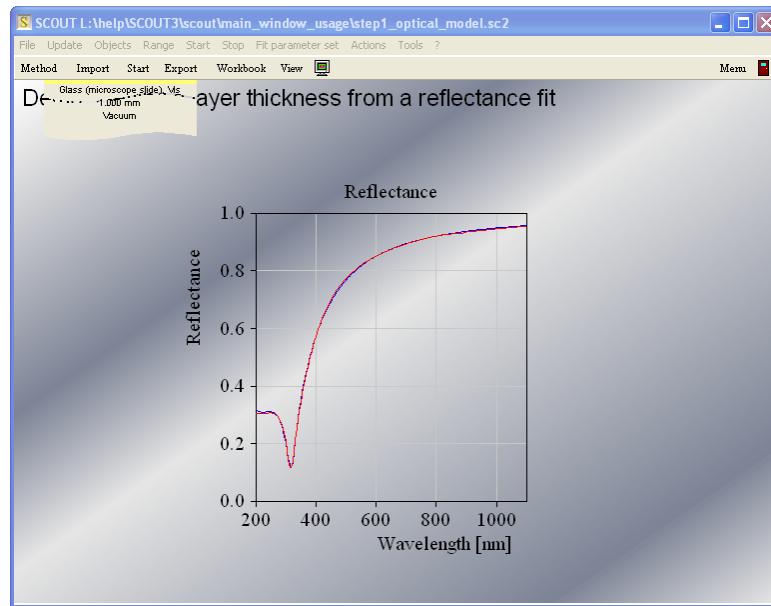


The manipulation of view elements is explained more detailed in the [introduction to views](#).

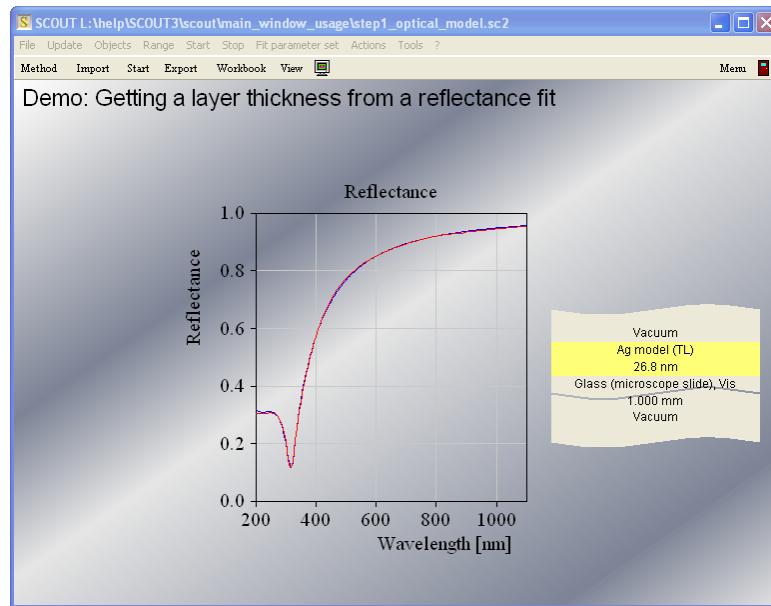
A view element to display the layer stack can be generated the same as shown for the spectrum. Simply drag the layer stack item from the treeview to the view element list:



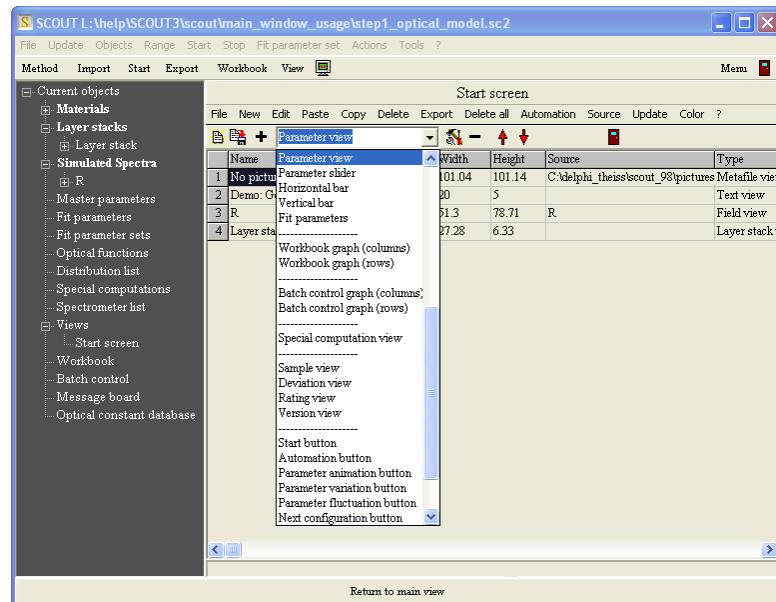
The result is shown below:



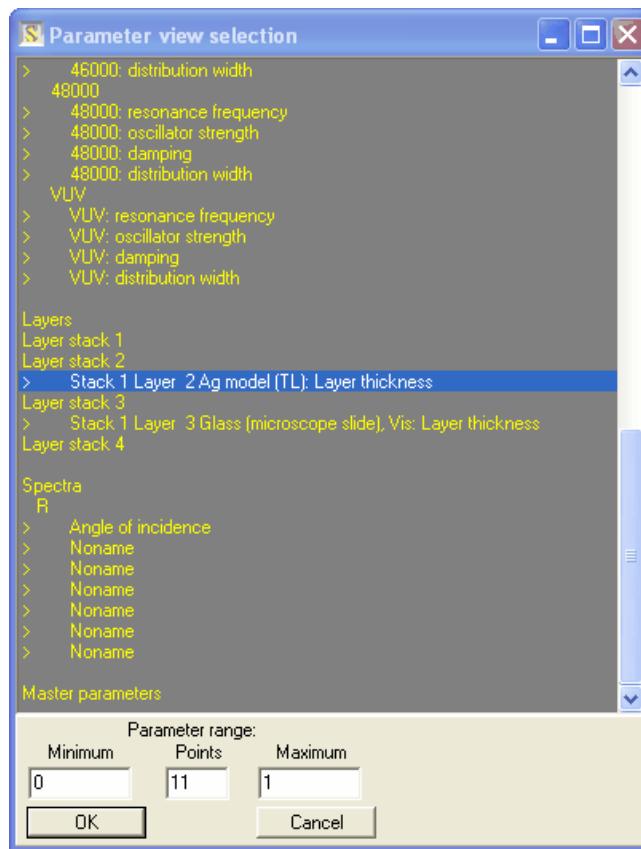
Layer stack view elements can be moved with the mouse using the bottom halfspace as 'mouse sensitive area' (click [here for details](#)). Drag the layer stack item to the right as shown here:



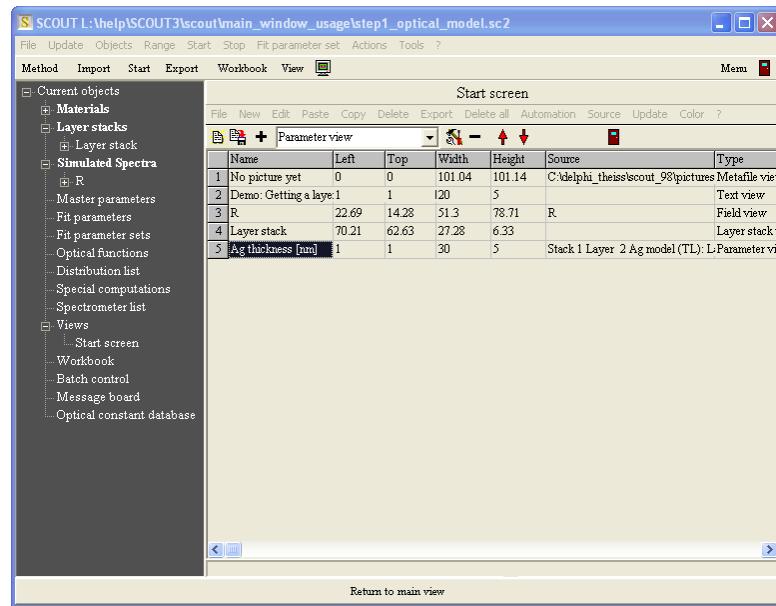
Now a view element is created which shows the current value of a model parameter. In the list of view elements, select the new object type 'Parameter view' in the dropdown box:



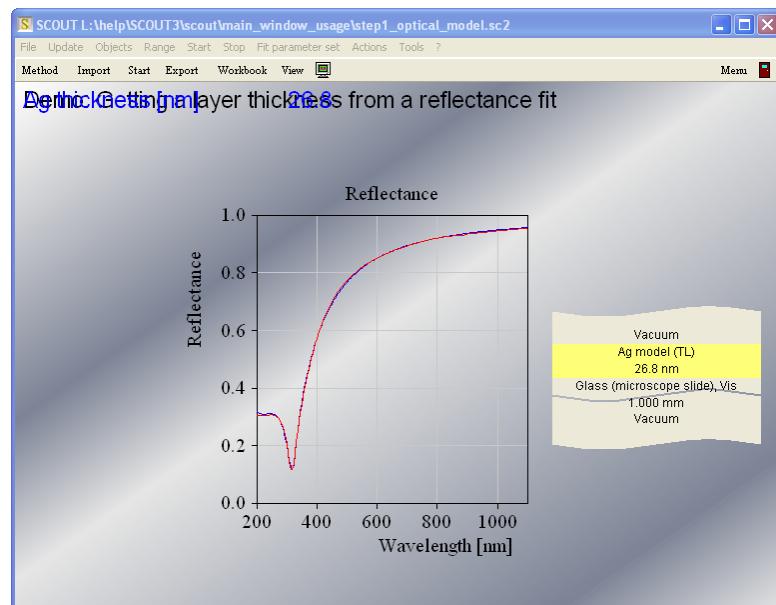
Then press **+** in order to create a new object of this type. You will get the following dialog for parameter selection. Mark the silver thickness as indicated:



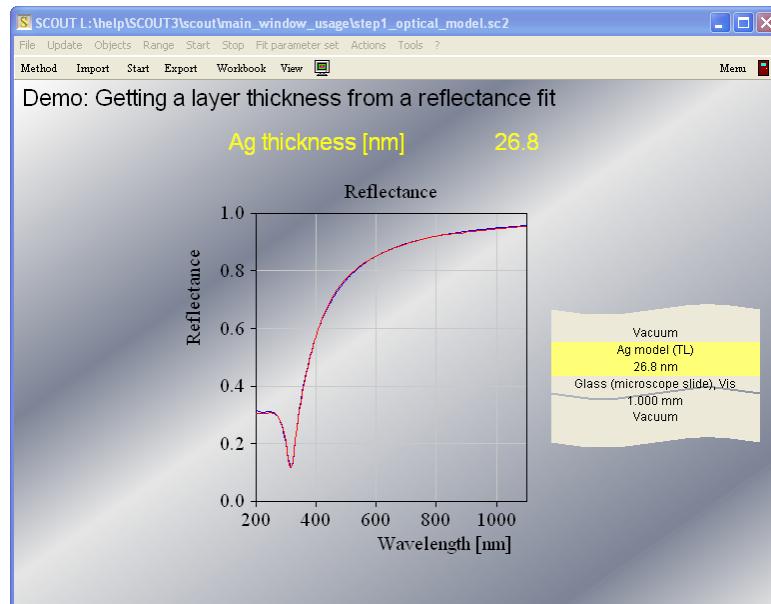
Then press **OK**. The new view element is created with the name 'No assignment yet'. Click on the name and overwrite it with 'Ag thickness [nm]' - then press **Return**:



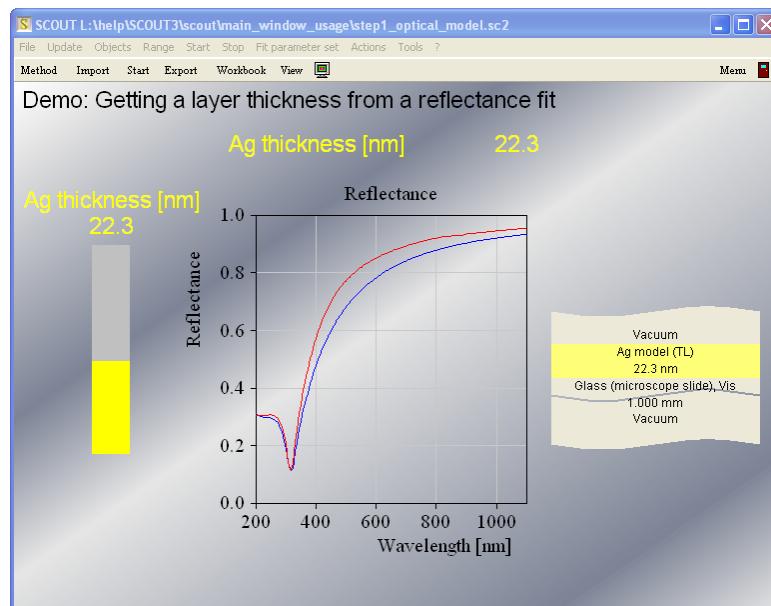
Pressing F7 shows that 2 view elements are in conflict with each other:



Move the blue parameter view element from the upper left corner down as shown above. Note that text is moved with the mouse using the lower left corner. Pressing the Ctrl-key and clicking on a view element raises the **Edit** action of the element. In case of the parameter view element a color selection dialog is shown. Use this to assign yellow to the element. In the end the main view should look like this:

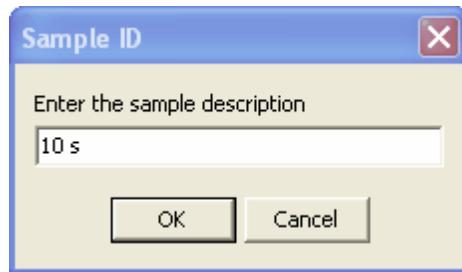


Now we will create a slider for manual parameter variation. In the list of view elements, select as new object type 'Vertical bar' and create an object of this type. The parameter selection dialog appears again and you should select again the silver thickness parameter. In the following color selection dialog, select yellow. Finally set the slider range. The minimum is 0 and the maximum 50. Change the 'No assignment yet' text to 'Ag thickness [nm]'. Finally move the new bar to the left side of the main view and try to re-size it as shown here:

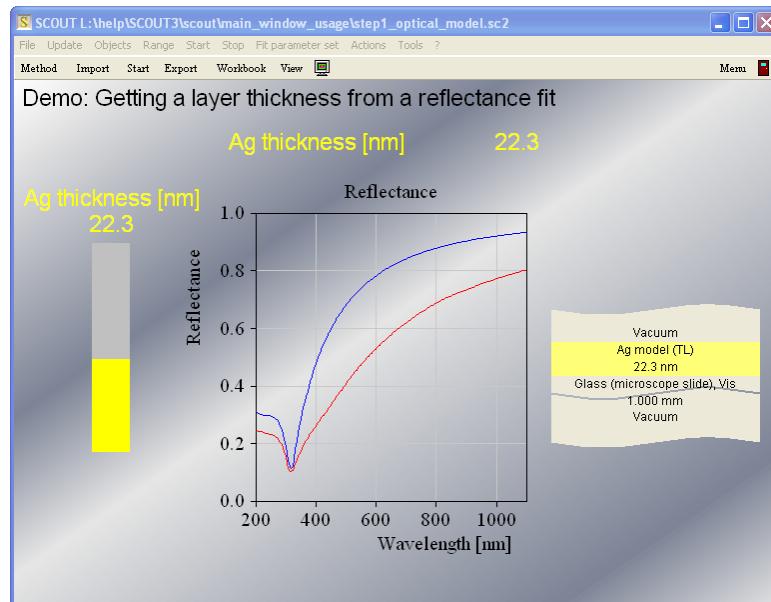


You can now use the mouse to move the parameter value by the slider position.

Use the Import button in the toolbar to load a new experimental spectrum. First you are asked for a sample name. Enter '10 s' (10 seconds sputtering time)



and select the spectrum ag_10.std:

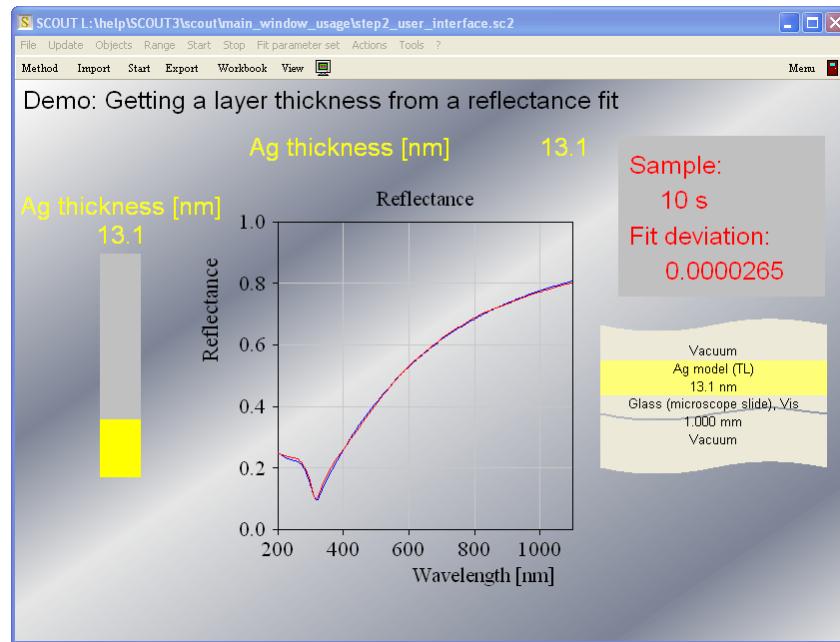


Try to fit the thickness manually with slider motions. Then press **Start** for an automatic fit.

Now, as an exercise, we do several steps at a time. Generate the following view elements:

- An object of type 'Rectangle' (this should produce a gray background for the following items)
- A 'text view' object called 'Sample:'
- An object of type 'Sample view' (this will display the name of the sample)
- A text view object called 'Fit deviation:'
- An object of type 'Deviation view' (this shows the mean square deviation of model and measurement)

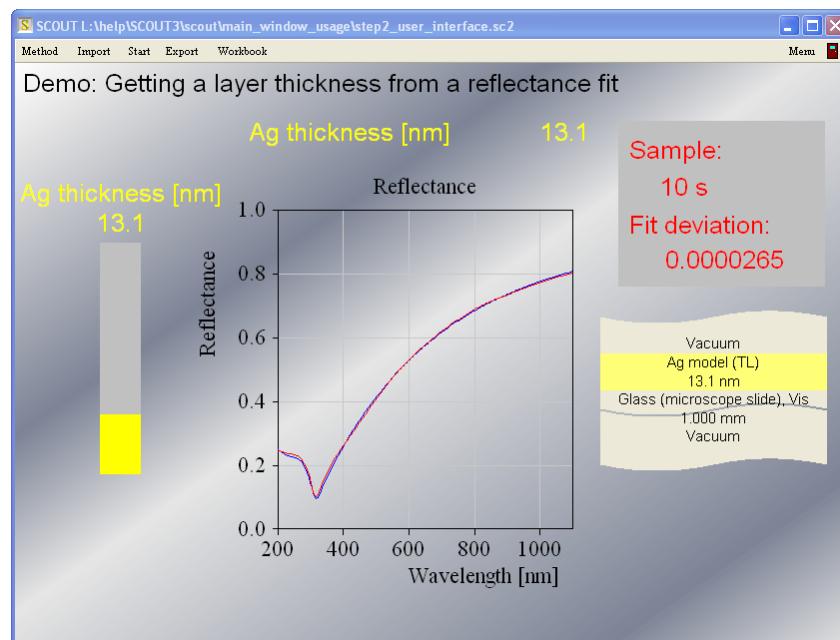
Try to get a main view appearance similar to the following:



In order to make the work with this method as simple as possible we will hide the menu when the configuration is loaded. This is done by de-activation of the menu option **File|Options|Appearance|Menu visible at start**.

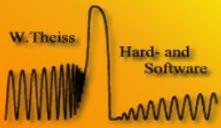
We can also hide the toolbar items 'View' and the 'Autoscale button'. This is achieved by de-selection of the menu items **File|Options|Toolbar|View button** and **File|Options|Toolbar|Autoscale button**.

Once this is done, you should save the new configuration using the **File|SaveAs** command. When you load it again with **File|Open** the following main view should appear:



In the following chapter we will see how such a SCOUT configuration can be used by operators in order to analyze measured spectra.

SCOUT



Part

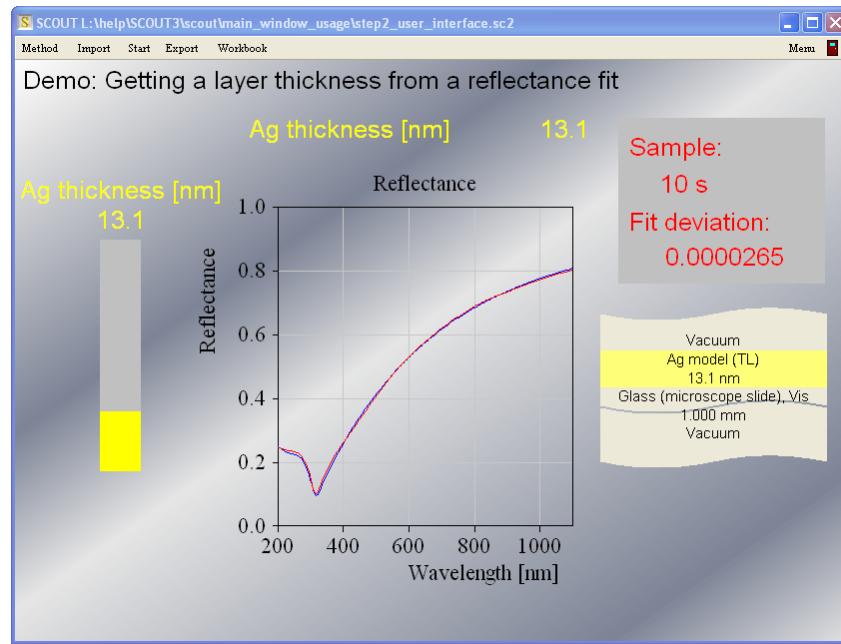
III

3 Mastering the main window

3.1 Main window overview

This section explains how you work with the main window. It is assumed that a working optical model exists and the main window has been configured in a user-friendly way. Since there is a large variation of possible designs we cannot cover all variations of the main window appearance. The following example shows a typical case covering the most important features.

The example used here is a simple thickness determination. The task is to get the thickness of a sputtered silver layer on microscope slide glass from a measured reflectance spectrum in the spectral range 200 ... 1100 nm. This configuration has been made in the previous 'Quick start' section. The main window looks like this:



The task of this configuration is to provide a user-interface for importing spectra, do the analysis, and export the results. In the ideal case of a well-defined and tested configuration the user has to handle only a small number of commands in [routine work](#).

Sometimes the graphical settings are not appropriate and the user may want to change it. How this is done is explained in a [separate section](#).

A little more advanced is the combination of the main window and the batch control window. This is also explained [separately](#).

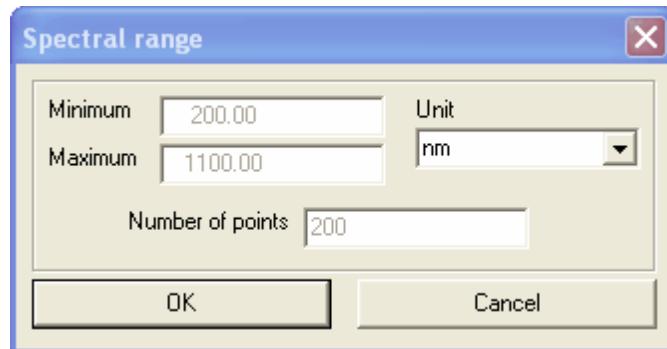
3.2 Routine work

Here is the sequence of commands for a simple routine case with spectrum import, analysis and export of the results.

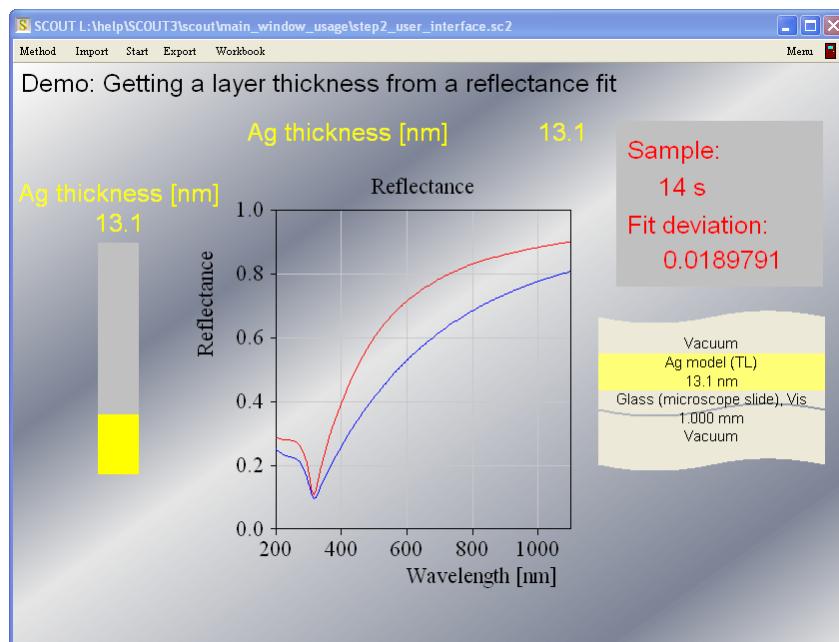
Spectrum import

Use the **Import** command to load new spectra. First the user is asked to enter a short text characterizing the sample. This text will replace the text '10 s' which is shown in the upper right rectangle underneath 'Sample':

Type in '14 s' and then select the file ag_14.std (SCOUT standard format) that comes with tutorial 1. Make sure the spectral unit in the following range dialog is set to nm:



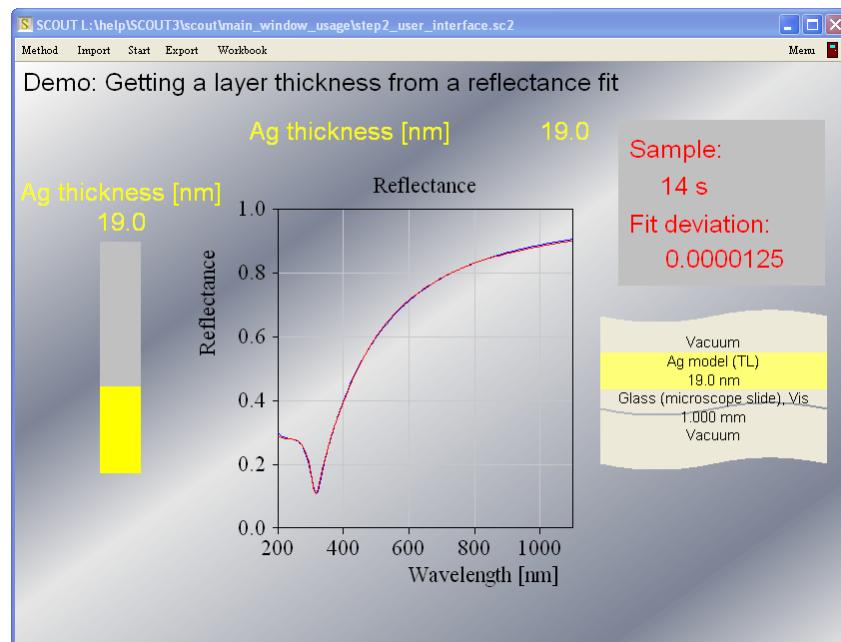
The SCOUT window should now have changed to this:



Doing the analysis

You can change the silver thickness value in the optical model by moving the height of the yellow bar on the left with the mouse. In this simple example it is very easy to find a good solution just be manually trying different values.

You can also press the **Start** button to tell SCOUT to do the analysis. The progress of the fit is indicated by the number given underneath the label 'Fit deviation'. This number describes the average difference between model spectrum and measurement - it should be as small as possible. While the fit is running, the **Start** button becomes a **Stop** button. Use this button to interrupt the fit, freezing the current best parameter values. When the fit stops the **Stop** button turns into a **Start** button again. The fit result for sample '14 s' is this:



Write the results to the Workbook

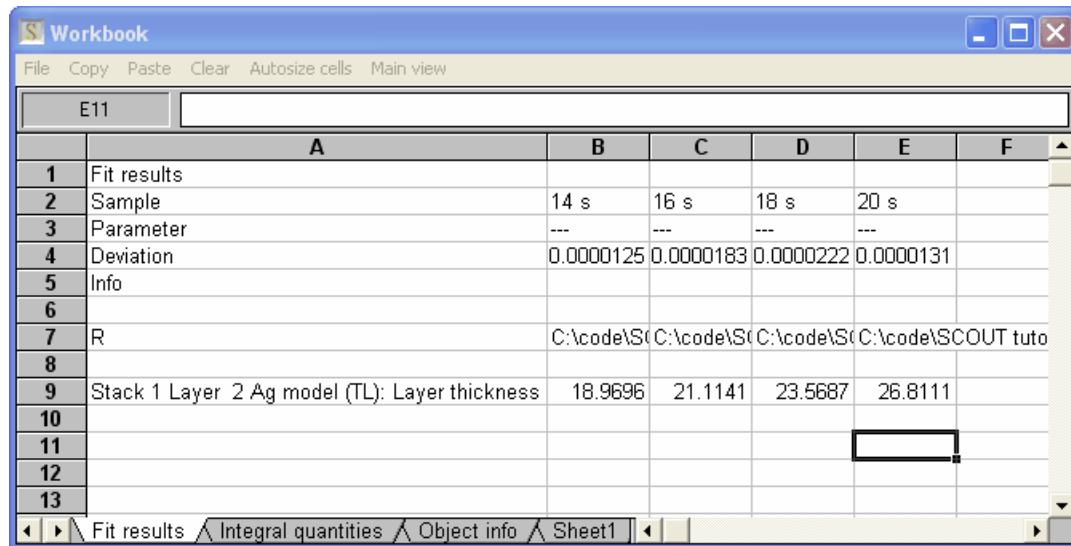
Apply the **Export** command to write the obtained parameter values for this sample to the workbook. The values of all defined [optical functions](#) are written to the workbook as well. If this is the first time you write to the workbook, SCOUT will erase the workbook page for the results and create a new one. It will ask you for the permission to do so:



Having answered 'Yes', you can now go back to the import step and analyze as many spectra as you like, exporting for each one the results to the workbook.

Saving the results

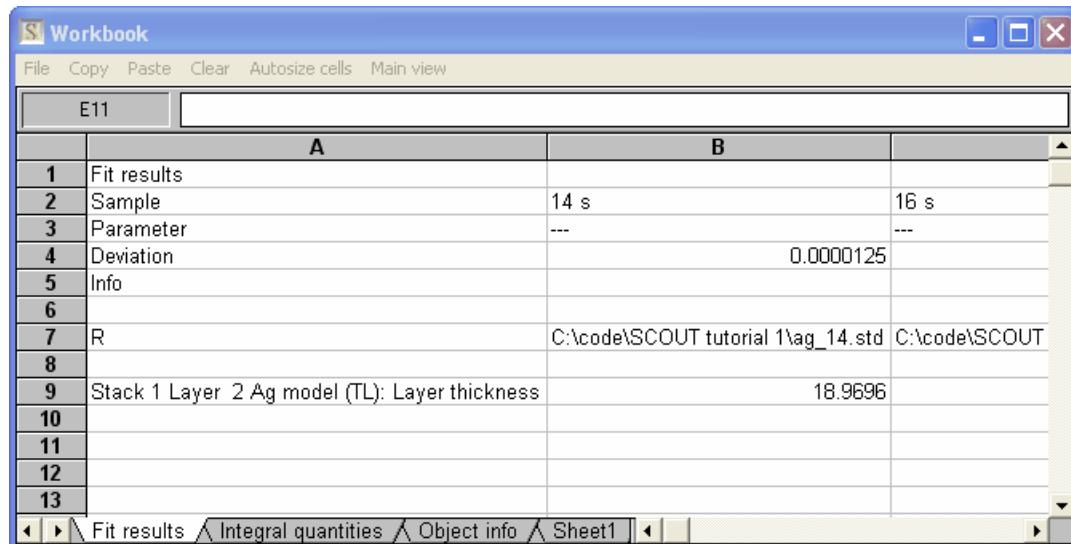
Once you have processed all samples, you have to save the results that are listed in the workbook. In order to do so, open the workbook with the **Workbook** command:



The screenshot shows the SCOUT Workbook window with a table containing the following data:

	A	B	C	D	E	F
1	Fit results					
2	Sample	14 s	16 s	18 s	20 s	
3	Parameter	---	---	---	---	
4	Deviation	0.0000125	0.0000183	0.0000222	0.0000131	
5	Info					
6						
7	R	C:\code\SCOUT tutorial 1\ag_14.std				
9	Stack 1 Layer 2 Ag model (TL): Layer thickness	18.9696	21.1141	23.5687	26.8111	
10						
11						
12						
13						

If you like you can use the **Autosize cells** command to adjust the size of the cells automatically:

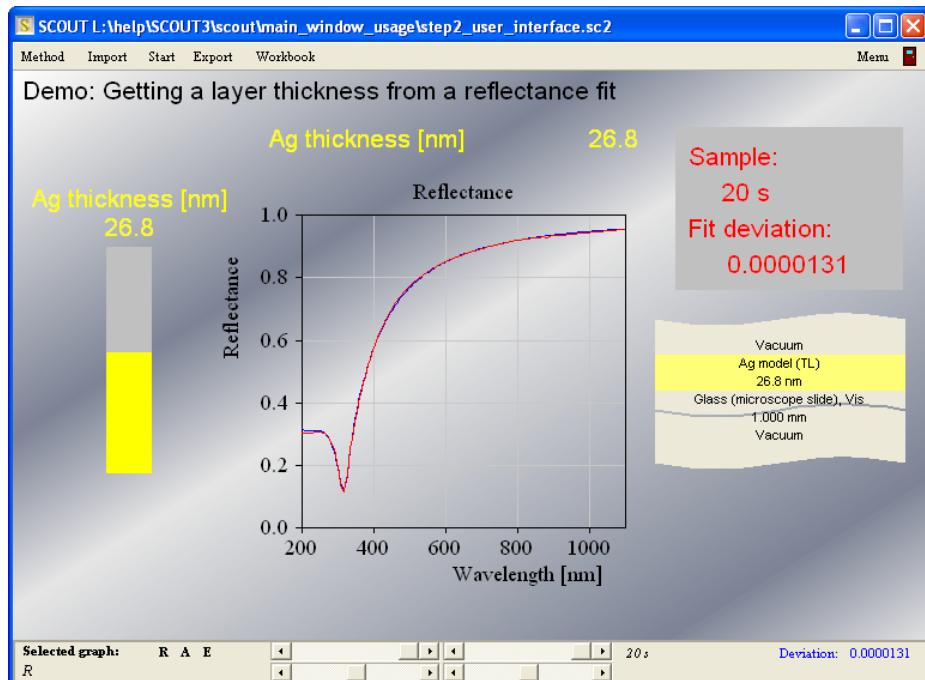


The screenshot shows the SCOUT Workbook window with the same table as above, but with the columns B and C automatically resized to fit their contents. The data is identical to the previous screenshot.

The workbook has file commands to save its content in various formats, including text files and Excel 5 format. You can also select a range of cells or the whole sheet and use the **Copy** command to copy the cell content into the clipboard (for a subsequent paste action into Excel, for example).

3.3 Manipulation of the displayed graphs

Sometimes you may want to change the pre-defined graphics settings in a view in order to inspect details of a spectrum. The statusbar provides (on the left side) some controls that you can use to do so. If the statusbar is not visible, press the key **s** on your keyboard to let it appear:



Graph selection

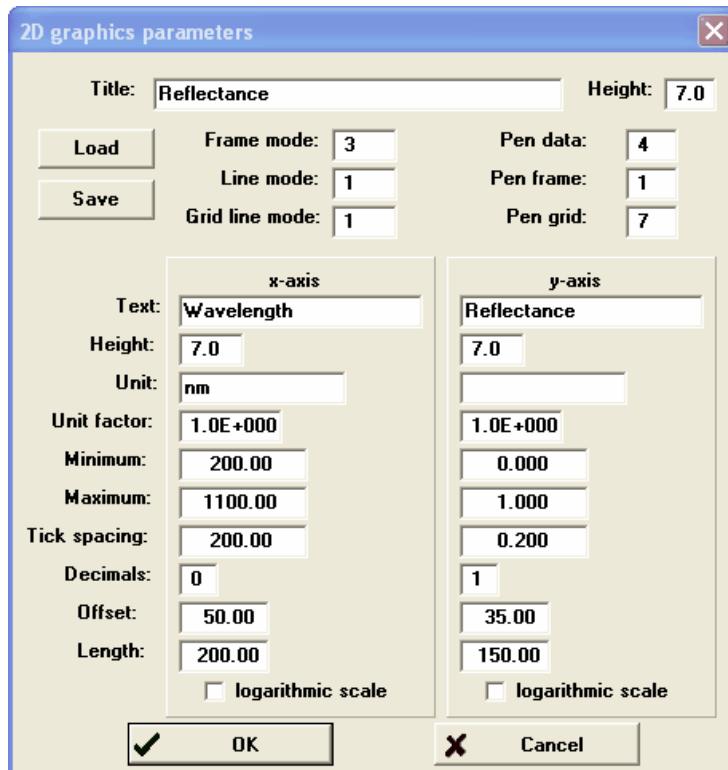
Before you use one of the controls you must make sure that the graph you want to change is selected. The name of the selected graph is indicated in the bottom left corner of the status bar. You can change the **selection by a simple mouse click on the graph** you want to modify.

Automatic scaling

Press the speed button labeled '**A**' for automatic scaling. SCOUT will determine the required range of the x- and y-axis to display both the measurement and the simulation. If the measurement contains large fluctuations due to noise the automatic scaling may fail to give a good overview about the spectra. In this case, you have to do the settings manually or using the sliders described below.

Edit graphics parameters manually

If you want to set the displayed region manually, or change other parameters like the frame mode (to display grid lines, e.g.), apply the '**E**' button which opens the graphics dialog of the graph:



See the section 'Graphics course' delivered with SCOUT for more details of graphics settings.

Changes by slider motions

The status bar has four sliders which can be used to change the displayed range of a graph conveniently. Here is how they work:

As soon as you move one of the sliders, **automatic scaling** is performed. The computed ranges for the x- and the y-axis are stored internally. With the two top sliders you can now set the size of the displayed range for the x- and y-axis, respectively. The left slider position means 0%, the right position 100% of the maximum range.

The two bottom sliders are used to set the position of the center of the graph (left slider: x-axis, right slider: y-axis): The left slider position sets the center of the graph to the minimum of the range found by auto-scaling, the right position corresponds to the maximum of the range.

You are encouraged to play for a while with the four sliders to get used to their function. After a short time of practise it should be clear how to move around.

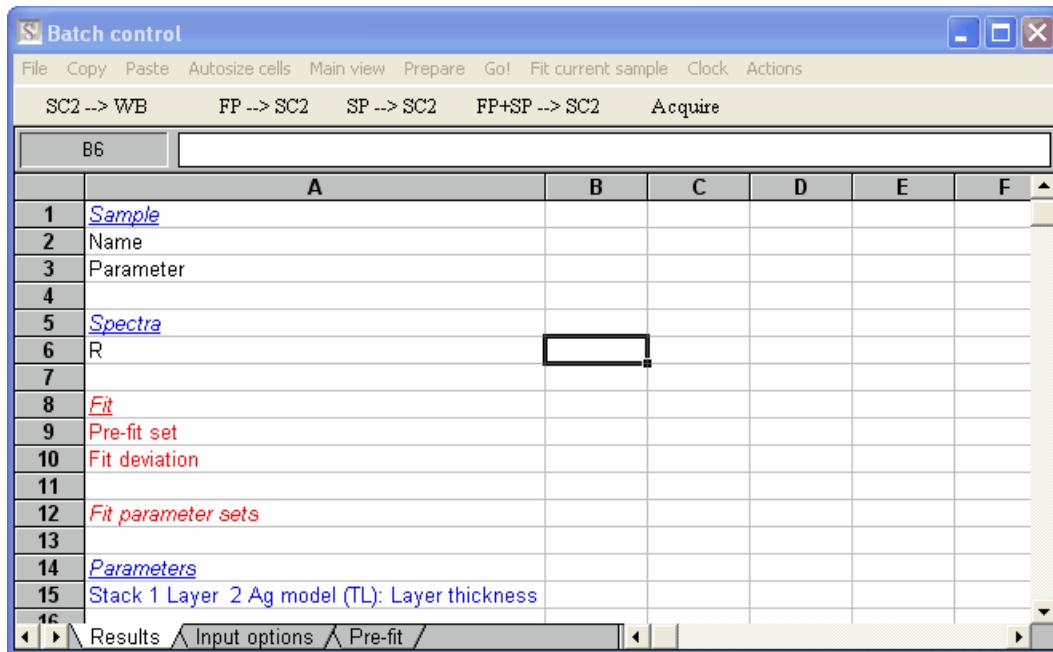
Restore graphics settings

Once you have zoomed into the graph or auto-scaled the graph and you want to go back to the original display, press the '**R**' button. This restores the graphics parameters that were set in the last edit operation (button '**E**'). If you did not edit the parameters manually, the initial settings of the loaded SCOUT configuration are restored.

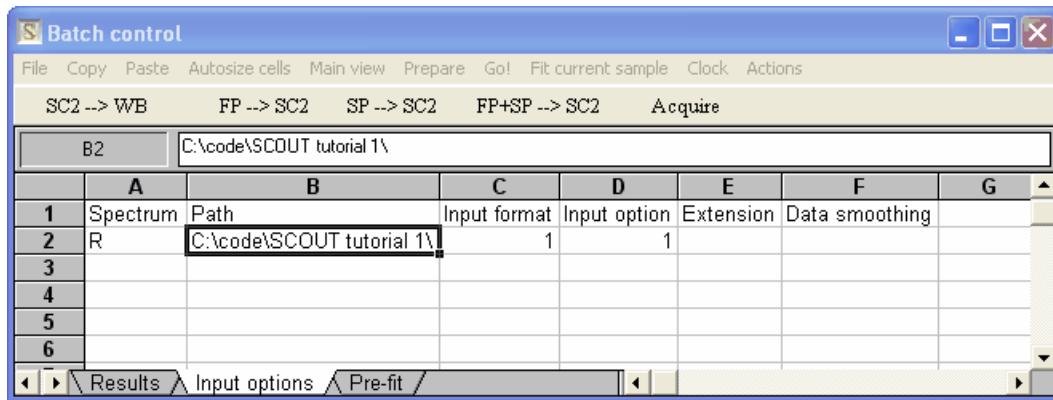
3.4 Working with the batch control window

If a SCOUT analysis method works fine for almost all input spectra, you may want to process the spectra to be analyzed completely automatic. You can do so using the batch control window as explained below. The proper configuration of the batch control window should have been done by an experienced SCOUT user (the batch features of SCOUT are explained in a [separate section](#)). This section describes how to use such a solution.

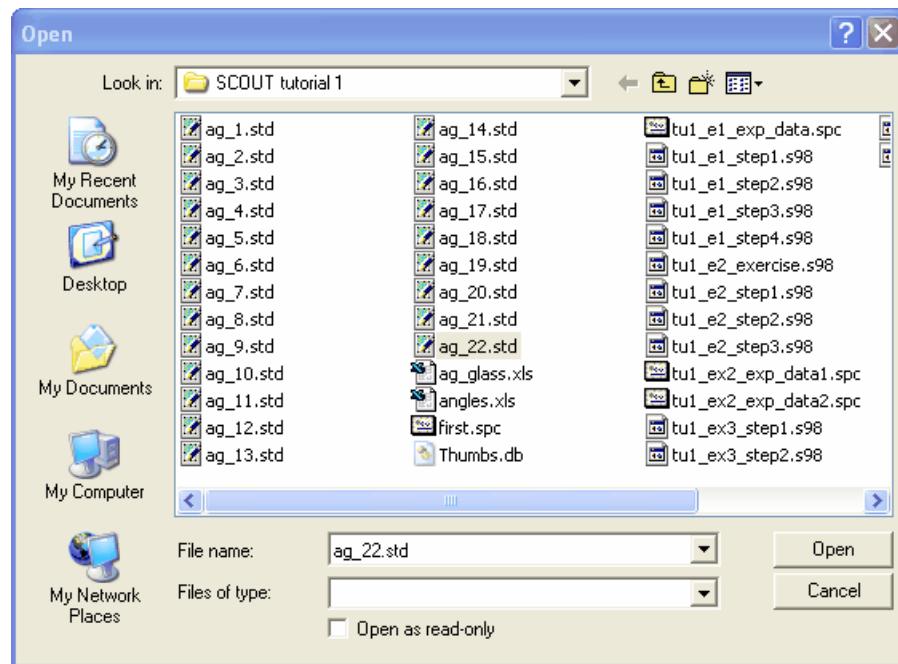
In order to work in batch mode, you have to have access to the **main menu** of SCOUT. If the main menu is not visible you can make it appear pressing **m** on your keyboard. The menu command **Actions|Batch control** opens the batch control window. The batch window is a kind of workbook which will hold the names of the input spectra and - after having completed the batch fit - the results. It may look like this:



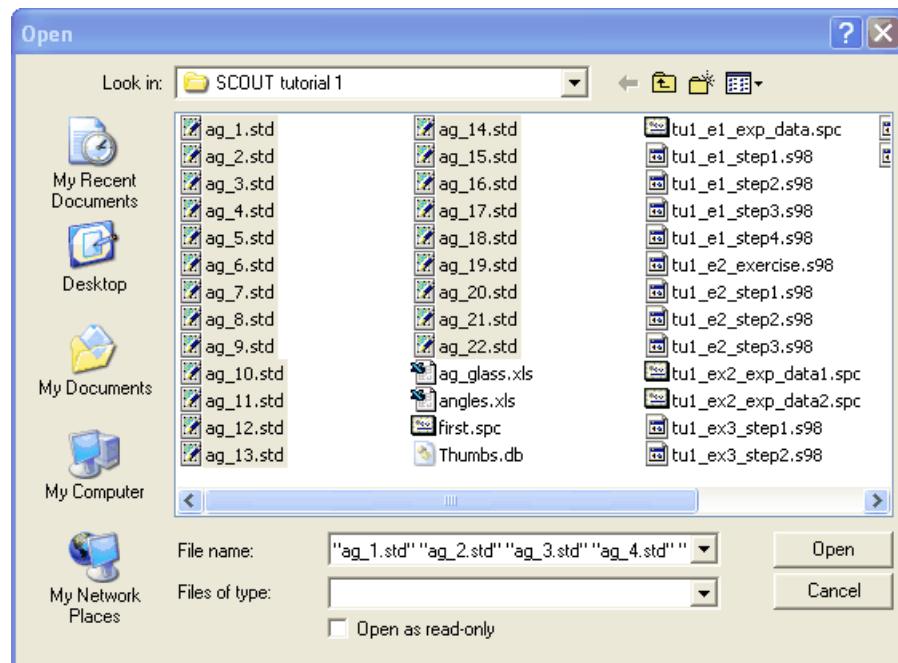
First you should inspect the 'Input options':



Check, if your input files are really available in the specified path, and if the input format matches (see the section about [preparing batch files](#) for more information). Then go back to the 'Results' page and place the cursor in the row of the spectrum for which you have measured spectra (row 6 in our case). Use the batch control menu command **Add multiple files** and select the last spectrum you want to load in the following dialog (note that the file ag_22.std is selected). Do not press the **Open** button yet:



Now hold the **Shift** key down and click on the first file to be loaded, in our case `ag_1.std`. This will select all spectra in between:



Now press Open and verify that all filenames have been entered as input spectra into the Results page of the batch control window:

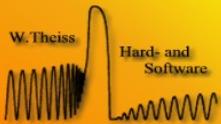
Batch control									
File Copy Paste Autosize cells Main view Prepare Go! Fit current sample Clock Actions									
SC2 --> WB FP --> SC2 SP --> SC2 FP+SP --> SC2 Acquire									
W6	ag_22.std								
A	B	C	D	E	F	G	H	I	
1 <i>Sample</i>									
2 Name	1	2	3	4	5	6	7	8	
3 Parameter									
4									
5 <i>Spectra</i>									
6 R	ag_1.std	ag_2.std	ag_3.std	ag_4.std	ag_5.std	ag_6.std	ag_7.std	ag_8.std	ag_9
7									
8 <i>Fit</i>									
9 Pre-fit set									
10 Fit deviation									
11									
12 <i>Fit parameter sets</i>									
13									
14 <i>Parameters</i>									
15 Stack 1 Layer 2 Ag model (TL): Layer thickness									
16									
17 <i>Functions</i>									
Results	Input options	Pre-fit							

Finally press **Go!** and wait until the batch fit finishes to process all input spectra. The results are written to the batch control window:

Batch control									
File Copy Paste Autosize cells Main view Prepare Go! Fit current sample Clock Actions									
SC2 --> WB FP --> SC2 SP --> SC2 FP+SP --> SC2 Acquire									
X2									
A	B	C	D	E	F	G	H	I	
1 <i>Sample</i>									
2 Name	1	2	3	4	5	6	7	8	
3 Parameter									
4									
5 <i>Spectra</i>									
6 R	ag_1.std	ag_2.std	ag_3.std	ag_4.std	ag_5.std	ag_6.std	ag_7.std	ag_8.std	ag_9
7									
8 <i>Fit</i>									
9 Pre-fit set	Set 1	Set 1	Set 1	Set 1	Set 1	Set 1	Set 1	Set 1	Set 1
10 Fit deviation	0.0003169	0.0010484	0.0006478	0.000282	2.69E-005	3.02E-005	3.28E-005	3.22E-005	2.571
11									
12 <i>Fit parameter sets</i>									
13									
14 <i>Parameters</i>									
15 Stack 1 Layer 2 Ag model (TL): Layer thickness	0.7520679	1.6971824	2.3869147	3.8397772	6.3048539	7.6462736	8.9435101	10.225013	11.8
16									
17 <i>Functions</i>									
Results	Input options	Pre-fit							

Save the batch control window with **SaveAs** (you could generate Excel files, for example) or paste the data to the clipboard for subsequent processing. More information about inspecting the results is given in a [section below](#).

SCOUT



Part

IV

4 The optical model

4.1 Overview

The optical model in SCOUT is composed of three types of objects:

- [Materials](#) (selected from the database or user-defined)
- [Layer stacks](#) (selected from the database or user-defined)
- [Spectra](#) (user-defined)

For each object type there is a list that manages the individual objects (create, delete, modify). Please read the section about [lists](#) if you are not used to work with SCOUT lists.

4.2 Defining Materials

4.2.1 Introduction

In order to use a material in SCOUT you have to define its optical constants (see [below](#)). In addition, you can define several [other quantities](#) such as thermal conductivity, density or deposition rates which may be required by some SCOUT or CODE computations. These numbers are stored together with the material's optical constants. Each material window has a menu item called **Properties** which has submenu commands to edit the available material properties.

Optical constants

In most cases it is the electric field of the probing light wave that interacts with the sample. Hence excitations can be observed in optical experiments that are going along with a polarization. The polarization P induced by an externally applied electric field E in a homogeneous material is given by the electric susceptibility χ :

$$P = \epsilon_0 \chi E$$

The dielectric function ϵ which connects the dielectric displacement and the electric field vector is closely related to the susceptibility:

$$D = \epsilon_0 \epsilon E, \quad \epsilon = 1 + \chi$$

The frequency dependence of the susceptibility is very characteristic for a material since it incorporates vibrations of the electronic system and the atomic cores as well as contributions from free charge carriers.

The complex refractive index $n + i k$ is the square root (with positive imaginary part) of the dielectric function. Both the dielectric function and the refractive index are called optical constants. Internally SCOUT uses the dielectric function for all computations. In a material's window you can display both quantities, as described [below](#).

The list of materials in SCOUT can exchange objects with a [database](#) of materials.

The SCOUT software offers several ways to define the frequency dependence of real and imaginary part of the dielectric function. You can import fixed data sets, i.e. tables of real and imaginary part for several spectral points, or you can setup models that give some mathematical expressions for the optical constants.

Optical constants can be displayed in [various ways](#): As dielectric function or complex refractive index $n + i k$. In addition you can show the absorption coefficient α or the energy loss function.

The following sections give more details on the individual types of materials available in SCOUT:

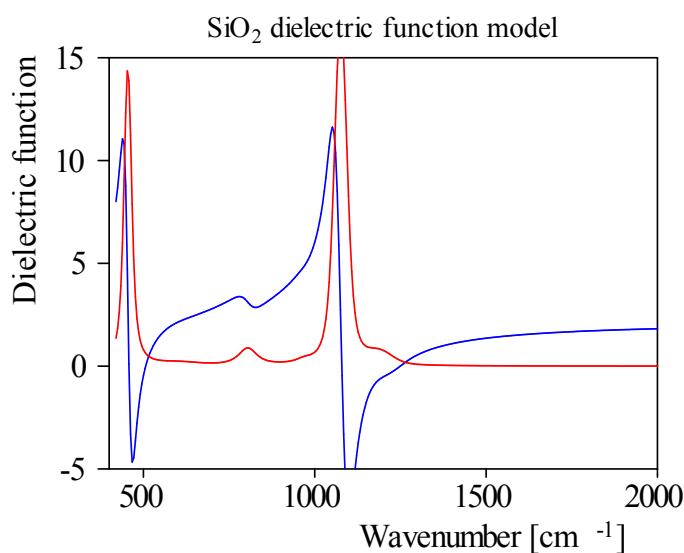
- [Imported dielectric function](#) (fixed data sets)
- [Dielectric function model](#)
- [KKR dielectric function models](#)
- [Master models](#)
- [Effective dielectric functions](#)

4.2.2 Properties

4.2.2.1 Overview

Each material window can display different representations of the optical constants.

In the following the computed quantities are demonstrated using a dielectric function for quartz glass in the mid infrared. It looks like this:



Use the **Properties** submenu to select the property which is to be displayed in the material's window. At present, the following properties (besides the dielectric function) can be computed and displayed:

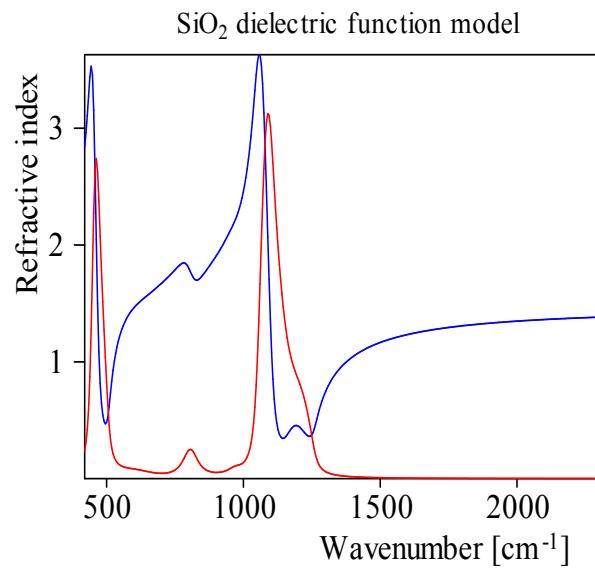
- [Refractive index](#)
- [Absorption coefficient](#)
- [Energy loss function](#)

If you change the selection of the displayed property the graphics parameters are automatically changed and auto-scaled.

4.2.2.2 Refractive index

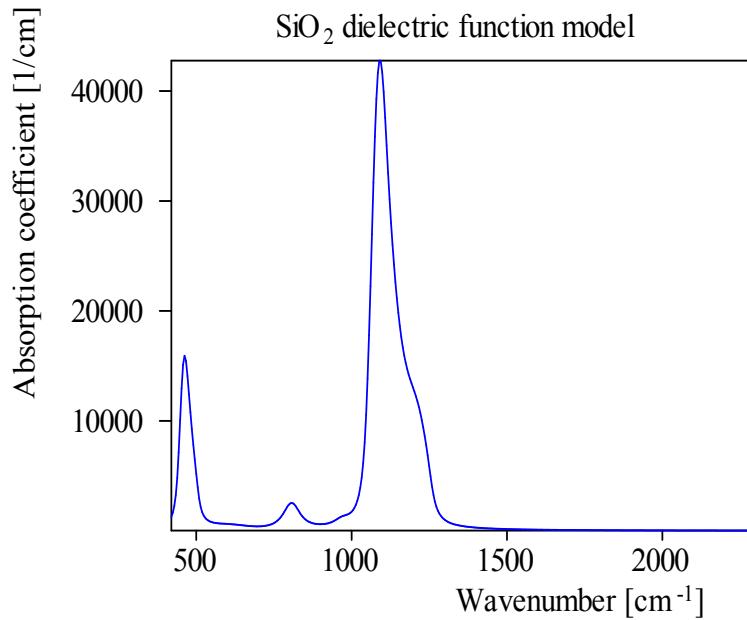
If the display property is set to 'Refractive index' the refractive index $n + i k$ will be computed from the dielectric function and displayed. The real part n is drawn in blue, the imaginary part k in red if the default graphics settings have not been changed.

For the glass example you get here the following graph:



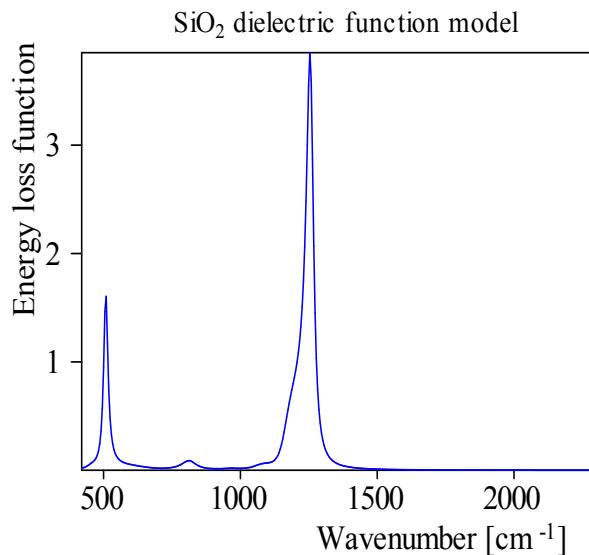
4.2.2.3 Absorption coefficient

Here the absorption coefficient is computed from the corresponding dielectric function data. It is given in 1/cm. For the glass example one gets the following values:



4.2.2.4 Energy loss function

Here the energy loss function is shown. The glass example has these values:



4.2.2.5 Others

Besides optical constants, a material in SCOUT can have the following properties:

- Thermal conductivity
- Density
- Dynamic viscosity
- Specific heat
- Etching rate
- Deposition rate

The menu item Properties offers submenu commands to set the numerical values of these quantities. Please note the indicated unit if you type in new values.

4.2.3 Material types

4.2.3.1 Imported dielectric function: Fixed data sets

Fixed data sets are tables of the real and imaginary part of the dielectric function given in a certain spectral range. Most of the data are obtained from literature somehow (by typing in or scanning published table values or digitizing published curves). Some may have been obtained from more or less direct evaluation of measured spectra (like KKR analysis of reflectivity data or spectroscopic ellipsometry).

Importing external optical constant data into SCOUT

If you want to work with fixed data sets for optical constants you have to create an object of type 'Imported dielectric function' in the list of materials. Objects of this type have an **Import** command that is used to import optical constant data from text files. In addition you can import data from the [workbook](#).

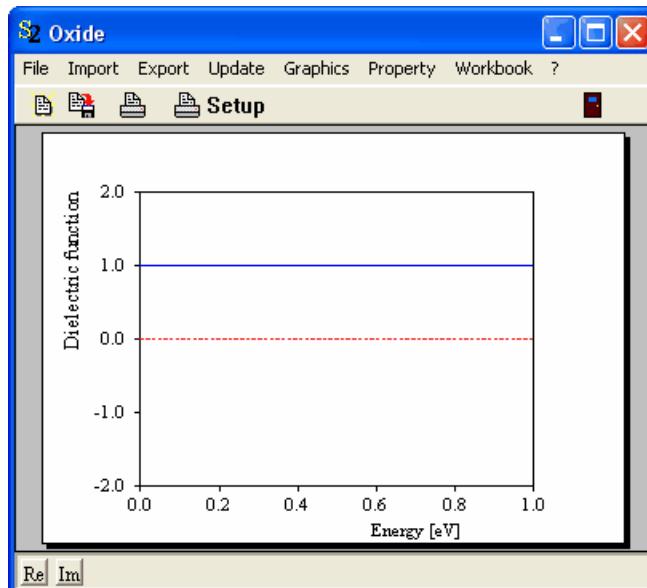
The import of n and k data from textfiles is explained in a [step-by-step example](#). [Getting data from the workbook](#) is also explained below.

Comments on spectral ranges of different objects

If you import fixed data sets from the database and you use them for a computation, the spectral range of the spectrum to be computed may differ from that of the fixed data set. Like in all SCOUT objects linear interpolation is applied to obtain dielectric function values in between two specified spectral positions of the dielectric function object. In case of extrapolation the dielectric function values of the closest specified spectral point are used – this means that you should **avoid extrapolation** unless the dielectric function is almost constant.

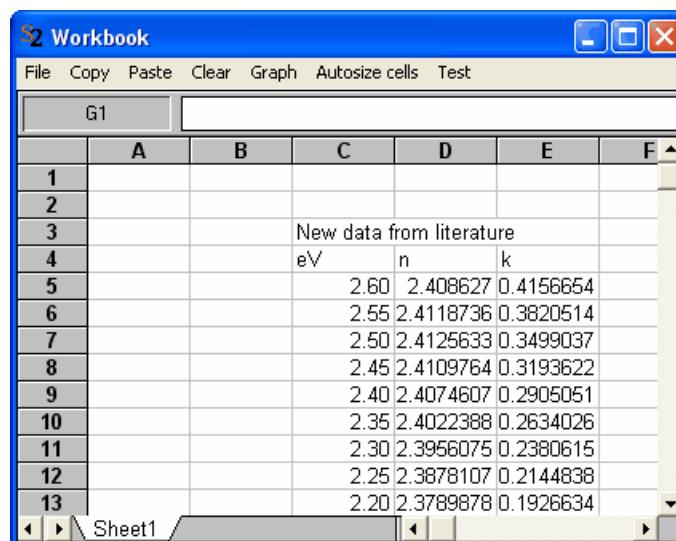
4.2.3.1.1 Importing data from the workbook

If you have original optical constant data in a spreadsheet format (e.g. in Microsoft Excel) and you want to use them in SCOUT you can import the data via the built-in workbook. In the list of materials, create a new object of type 'Imported dielectric function' and open its window (**Edit**):



Use the **Workbook|Open workbook** command to open the workbook.

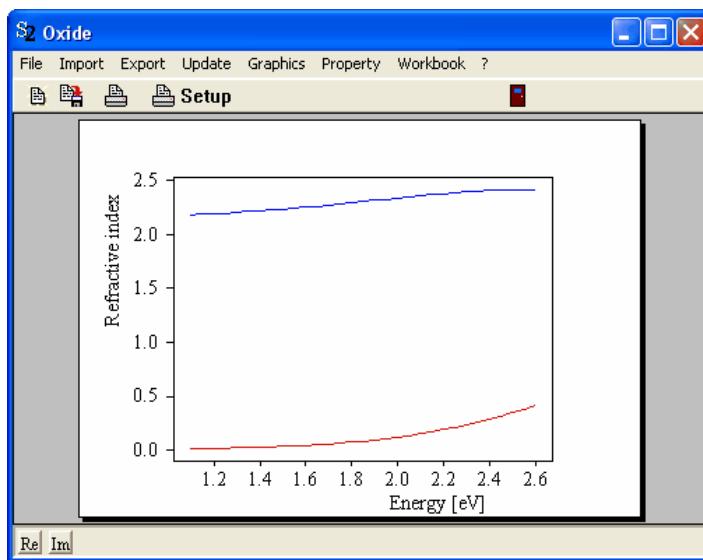
Now copy and paste the data to be imported into one of the sheets of the workbook. It should look like this:



Place the cursor in the upper left corner of data block that you want to read, in this case the cell C5 that contains the first energy value of 2.60 eV:

S2 Workbook	
File Copy Paste Clear Graph Autosize cells Test	
C5	2.6
1	
2	
3	New data from literature
4	eV n k
5	2.60 2.408627 0.4156654
6	2.55 2.4118736 0.3820514
7	2.50 2.4125633 0.3499037
8	2.45 2.4109764 0.3193622
9	2.40 2.4074607 0.2905051
10	2.35 2.4022388 0.2634026
11	2.30 2.3956075 0.2380615
12	2.25 2.3878107 0.2144838
13	2.20 2.3789878 0.1926634

Now go back to the new 'Imported dielectric function' object. If the workbook data contain n and k values, use the command **Workbook|Import x,n,k** and press 'a' to autoscale the graph:



If the spreadsheet data block contains dielectric function data, use the command **Workbook|Import x, real part, imaginary part**.

Up to now you cannot change the spectral unit if you import data from the workbook. This will be improved soon. Use the possibility to import data from text files if your spectral unit does not match the unit of the 'Imported dielectric function' object.

4.2.3.1.1.1 Importing data from text files

In order to import optical constant data from text files you must create input files of a certain structure. You can import dielectric functions or refractive index data. In both cases, the files must consist of three columns. The first column contains the spectral position (wavelength, wavenumber, frequency, ...), the second and third columns the real and the imaginary part, respectively. The columns must be separated by spaces or tab stops. Floating point numbers must be stored using

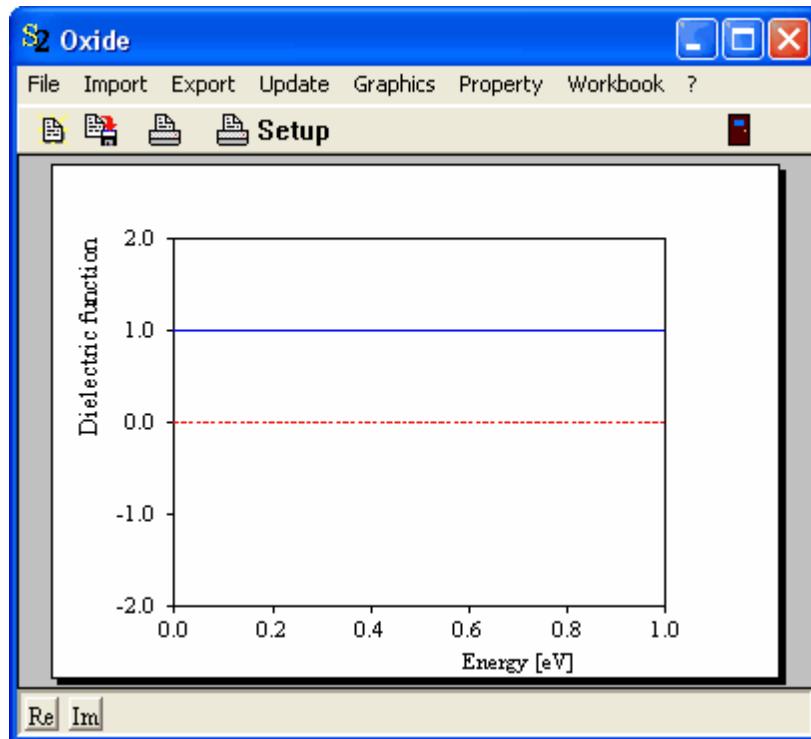
decimal points, independent of the regional settings in your Windows system.

No header line and no line after the last data point is allowed.

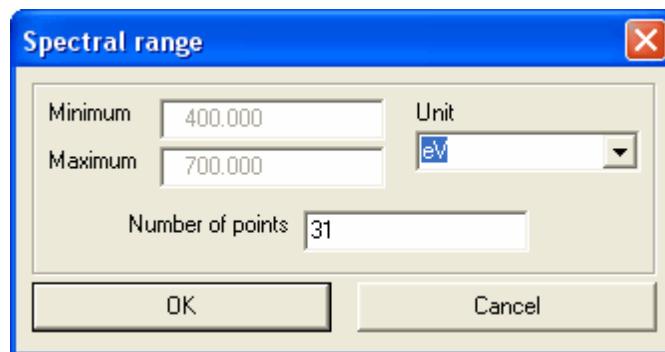
Here is an example for n and k data:

400	2.408627033	0.4156653583
410	2.411873579	0.3820514381
420	2.412563324	0.3499037027
430	2.41097641	0.3193621635
440	2.40746069	0.2905050516
450	2.402238846	0.2634026408
460	2.395607471	0.2380614579
470	2.387810707	0.2144837677
480	2.378987789	0.1926633567
490	2.369399548	0.1725632399
500	2.359213591	0.1541357189
510	2.348565817	0.1373167932
520	2.337641954	0.1220459864
530	2.326568365	0.10827066
540	2.315360069	0.09589903057
550	2.304250717	0.08483409137
560	2.293296814	0.07504497468
570	2.282603025	0.06639868021
580	2.272218227	0.0588131547
590	2.262247324	0.05217254534
600	2.252709866	0.04636290297
610	2.24363637	0.04127727076
620	2.234975576	0.036819987
630	2.226799965	0.03293710575
640	2.219054222	0.02952171676
650	2.211731434	0.02650727145
660	2.204816103	0.02387103066
670	2.198265076	0.02153578214
680	2.192091942	0.01947694644
690	2.18625164	0.01764915884
700	2.180743456	0.01603478752

Once you have created a new object of type 'Imported dielectric function' in your material list, open its window which should look like this initially:

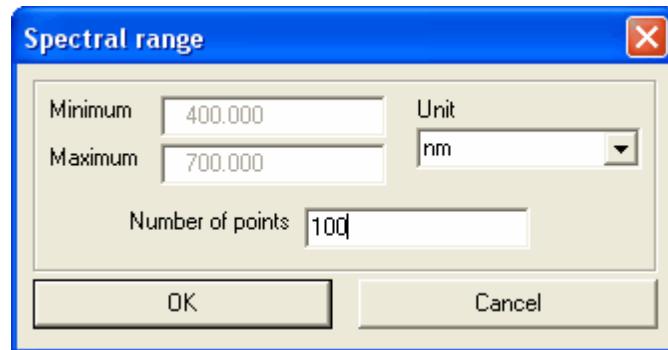


In order to import optical constant data from a text file, use the **Import** command and select the file you want to read from. At the bottom of the file dialog you can select the file type. Choose 'Data table (x, real, imag)' if you want to import dielectric function data, and select 'n' and 'k' from data table '(x,n,k)' if the file contains complex refractive index data.
SCOUT reads the file line by line and determines the spectral range covered by the data. The following dialog gives you a chance to **set the proper spectral unit**:

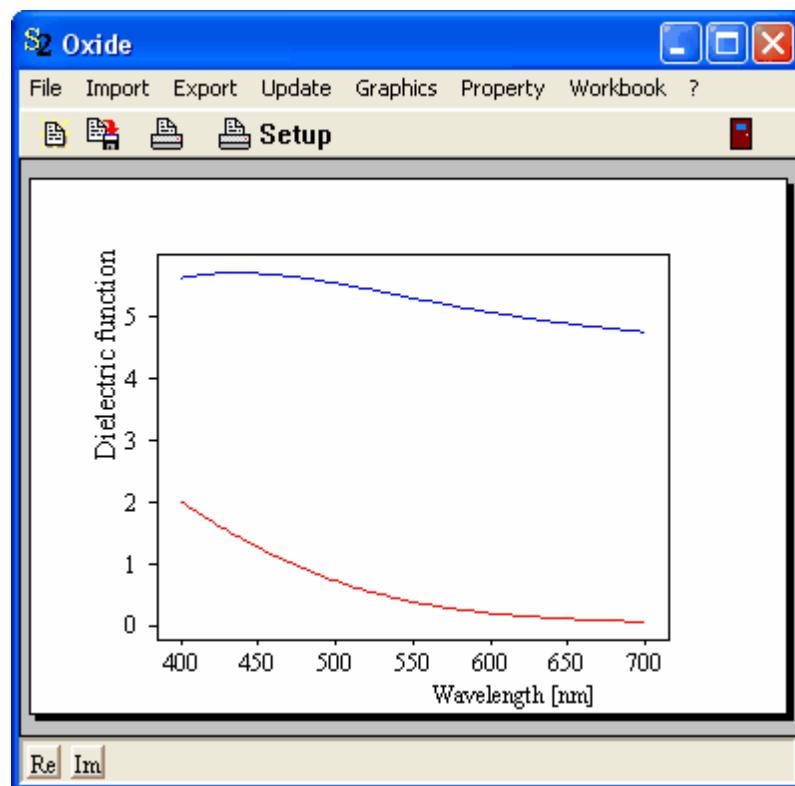


In this case the spectral range is given in nanometers, so the 'eV' should be changed to 'nm'. In addition, you can change the **number of spectral points** in the dialog. SCOUT gives the number of points found in the file. SCOUT always works with equidistant spectral positions. If the data in the file are non-equidistant, SCOUT will create an equidistant grid of spectral positions covering the found spectral range. If you don't change the number of points, it will use as many points as found in the file. The optical constant data on this equidistant grid of spectral positions are obtained by linear interpolation of the original non-equidistant data points. If there are large differences in the spectral distances of the original points it might be useful to increase the number of points of the equidistant grid.

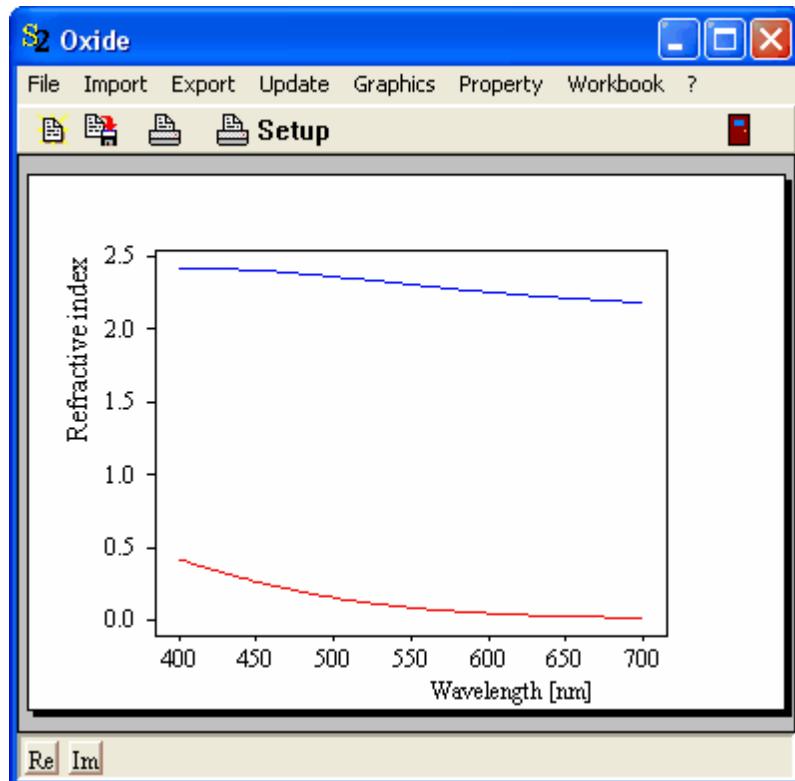
Having changed the spectral unit to 'nm' and the number of points to '100' the dialog looks like this:



Press **OK** and then 'a' to autoscale the graphics. Note that the new 'Imported dielectric function' object might not display the complex refractive index but the dielectric function:



If you want to see the complex refractive index, use the **Property** command and select 'Refractive index':



It is a good idea to permanently store the new object in your database.

4.2.3.2 Dielectric function models

4.2.3.2.1 Introduction

The dielectric function ϵ which connects the dielectric displacement and the electric field vector is closely related to the susceptibility χ :

$$\vec{D} = \epsilon_0 \epsilon \vec{E}, \quad \epsilon = 1 + \chi$$

The frequency dependence of the susceptibility is very characteristic for a material since it incorporates vibrations of the electronic system and the atomic cores as well as contributions from free charge carriers.

The following susceptibility types are supported by SCOUT:

- [Constant](#)
- [Constant refractive index](#)
- [Harmonic oscillator](#)
- [Drude model for free carriers](#)
- [Extended Drude model](#)
- [Brendel oscillator](#)
- [Kim oscillator](#)
- [Gervais oscillator](#)
- [User-defined formula](#)
- [Imported data](#)
- [KKR susceptibility](#)
- [KKR susceptibility II](#)

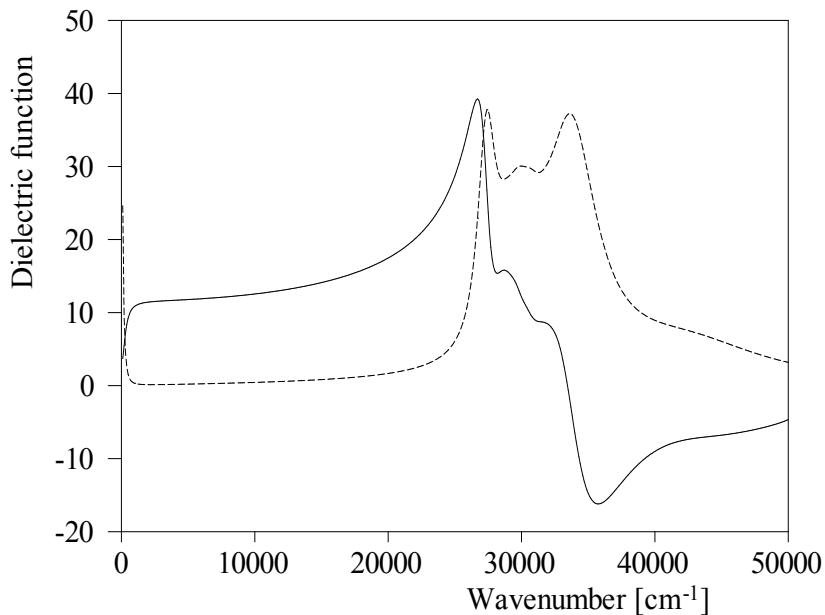
Appropriate models for interband transitions are discussed in the section on [KKR dielectric function](#)

[models.](#)

4.2.3.2.2 Susceptibilities

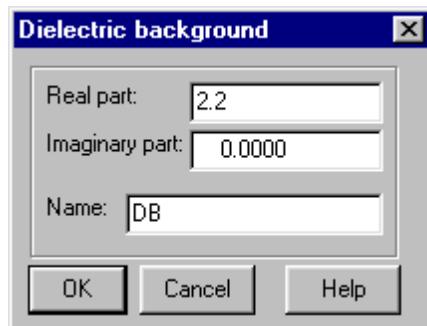
4.2.3.2.2.1 Constant

The figure below shows a dielectric function typical for a semiconductor in the infrared, visible and ultraviolet spectral region. The very strong structures in the UV can be assigned to electronic interband transitions (critical points in the joint density of states). In the infrared region one is far away from the interband transitions which can be noticed only by an almost constant and real contribution to the dielectric function - the so-called dielectric background. Therefore the optical properties in the infrared can be described quite well by just the sum of a constant and real dielectric function (sometimes called ϵ_∞ where ∞ stands for 'very high frequency' from the infrared viewpoint) and the contributions describing infrared excitations. Free carriers are responsible for the structure in the dielectric function below 1000 cm⁻¹.



Typical dielectric function of a group IV semiconductor (silicon) characterized by an almost constant dielectric function in the mid IR in between strong interband transitions in the UV and free carrier contributions in the far IR.

The dialog to edit a constant susceptibility in SCOUT is the following:



Although the imaginary part of realistic dielectric functions is close to zero in regions of a constant real part you can define a constant imaginary part different from zero for test

purposes.

4.2.3.2.2.2 Constant refractive index

For those who prefer to think of refractive indices when dealing with optical constants here is a constant refractive index susceptibility. It has just a constant real value the square of which is the real part of the susceptibility. The imaginary part is zero.

Note the remark about the combination of a constant refractive index susceptibility with interband transition models given below in the section about [KKR susceptibilities](#).

4.2.3.2.2.3 Harmonic oscillator

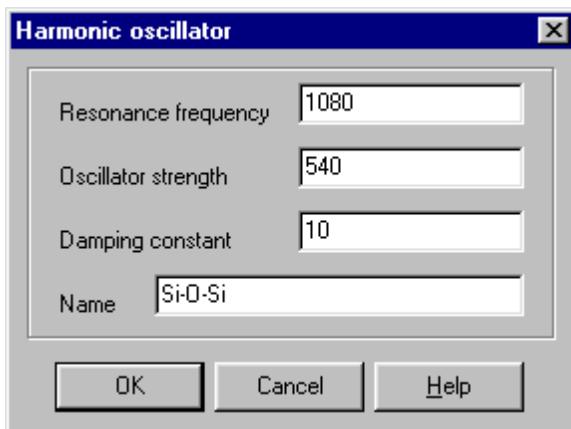
Microscopic vibrations involving the motion of the atomic nuclei (which are much heavier than the electrons) usually have their resonance frequencies in the infrared region. These characteristic frequencies depend on the oscillating masses and the strength of the bonding between them and hence can be used for material identification (which is the reason for the dominant role IR spectroscopy plays in analytical chemistry). In the case of silicon important impurities like carbon and oxygen are detected by their characteristic vibrational modes. Susceptibilities describing microscopic vibrations can be modeled by harmonic oscillator terms:

$$\chi_{\text{Harmonic oscillator}} = \frac{\Omega_p^2}{\Omega_{TO}^2 - \tilde{\nu}^2 - i\tilde{\nu}\Omega_\tau}$$

Ω_p gives the

oscillator strength, Ω_τ the damping and Ω_{TO} the resonance position. The index TO is used here since with the same oscillator model transverse optical phonons (lattice vibrations, TO-modes) are described.

The corresponding SCOUT dialog is this:



4.2.3.2.2.4 Drude model

In the case of doped semiconductors the charge carriers set free by the donors or acceptors can be accelerated by very little energies and hence do respond to applied electric fields with frequencies in the infrared region. A simple expression for the susceptibility of free carriers (which is also applicable to metals, of course) is given by the so-called Drude model where the carrier concentration and a damping constant enters:

$$\chi_{Drude}(\tilde{\nu}) = -\frac{\Omega_p^2}{\tilde{\nu}^2 + i\tilde{\nu}\Omega_\tau} \quad \text{with} \quad \Omega_p^2 = \frac{ne^2}{\epsilon_0 m}$$

Ω_p : Plasma frequency

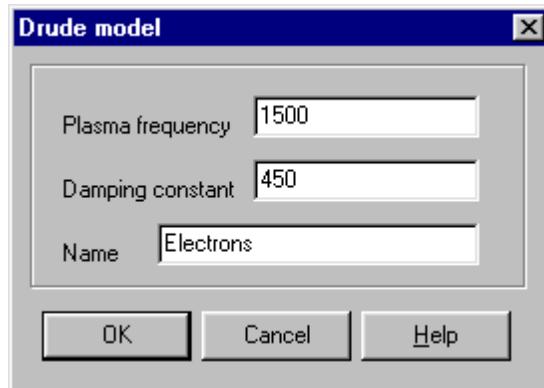
Ω_τ : Damping constant

n : Charge carrier density

m : Effective mass

e : Elementary charge

The parameters plasma frequency and damping constant can be set in the corresponding SCOUT dialog:



4.2.3.2.2.5 Extended Drude model

The classical [Drude](#) model works with a damping constant which does not depend on frequency. This is a good approximation in most cases. However, there are situations where the damping of the free carriers exhibits a characteristic dependence on frequency, for example in the case of scattering at charged impurities. A rather simple, but successful choice of the frequency dependence of the damping term is the following:

$$\chi_{Drude}(\tilde{\nu}) = -\frac{\Omega_p^2}{\tilde{\nu}^2 + i\tilde{\nu}\Omega_\tau} \quad \text{with} \quad \Omega_p^2 = \frac{n e^2}{\epsilon_0 m}$$

$$\Omega_\tau(\tilde{\nu}) = \Omega_{\tau,Low} - \frac{\Omega_{\tau,Low} - \Omega_{\tau,High}}{\pi} \left[\arctan\left(\frac{\tilde{\nu} - \Omega_{\tau,Crossover}}{\Omega_{\tau,Width}}\right) + \frac{\pi}{2} \right]$$

Ω_p : Plasma frequency

$\Omega_{\tau,Low}$: Damping constant at low frequencies

$\Omega_{\tau,High}$: Damping constant at high frequencies

$\Omega_{\tau,Crossover}$: Crossover frequency, center of the transition region

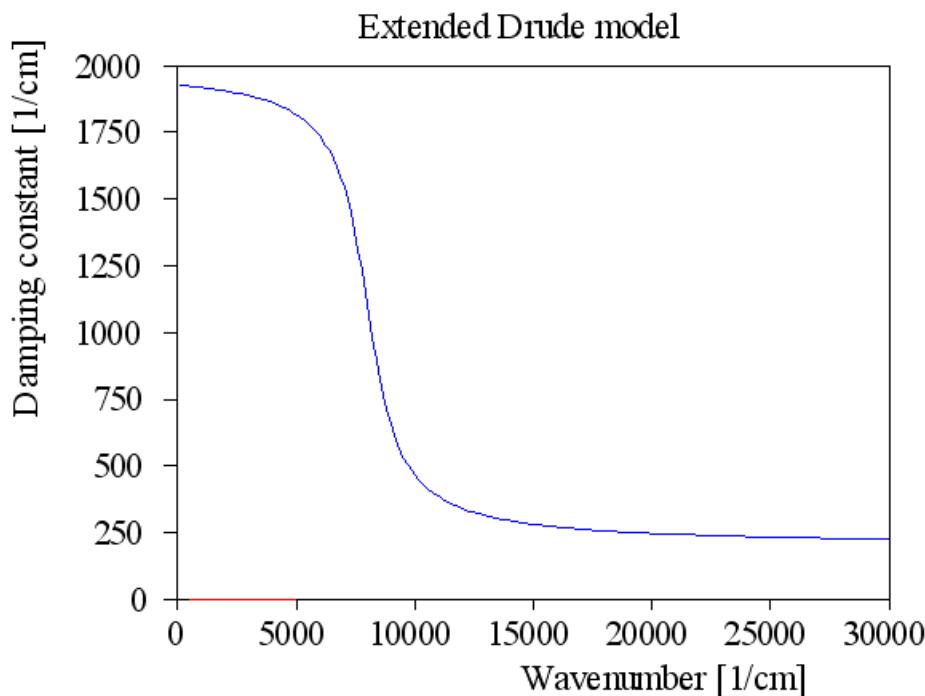
$\Omega_{\tau,Width}$: Width of the transition region

n : Charge carrier density

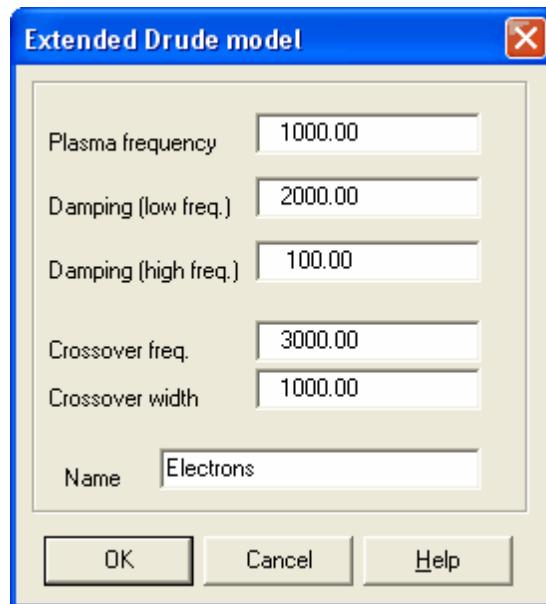
m : Effective mass

e : Elementary charge

The function for the damping constant is chosen to change smoothly from a constant at low frequencies to another constant level at high frequency. The transition region is defined by the crossover frequency and the width parameter. A typical case of the frequency dependence of the damping constant is shown below:



The parameters of the extended Drude model are set in the following dialog:



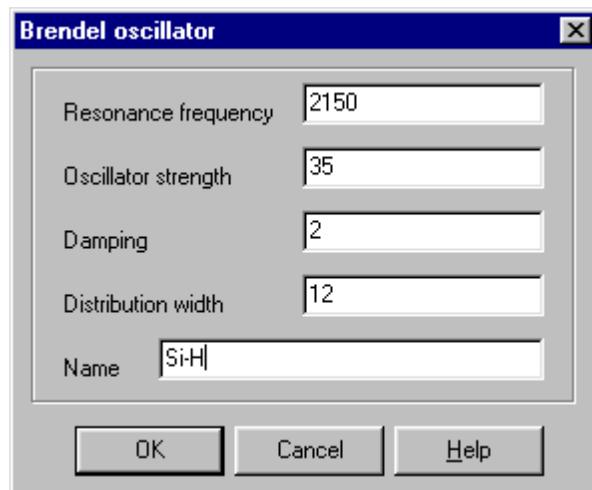
4.2.3.2.2.6 Brendel oscillator

The simple [harmonic oscillator model](#) for vibrational modes can be refined by an improved version (suggested by [Brendel](#)) that accounts for local variations in disordered systems by a Gauss-distribution of resonance frequencies:

$$\chi_j(\tilde{\nu}) = \frac{1}{\sqrt{2\pi}\sigma_j} \int_{-\infty}^{\infty} \exp\left(-\frac{(x - \Omega_{0,j})^2}{2\sigma_j^2}\right) \frac{\Omega_{P,j}^2}{x^2 - \tilde{\nu}^2 - i\tilde{\nu}\Omega_{\tau,j}} dx$$

Depending on the values of the width s of the Gauss-distribution and the damping constant Ω_τ any line shape between a Gaussian and a Lorentzian can be achieved. Due to this flexibility the 'Brendel' model has turned out to be very useful in many cases.

The four quantities of Brendel oscillators can be set in the following SCOUT dialog:



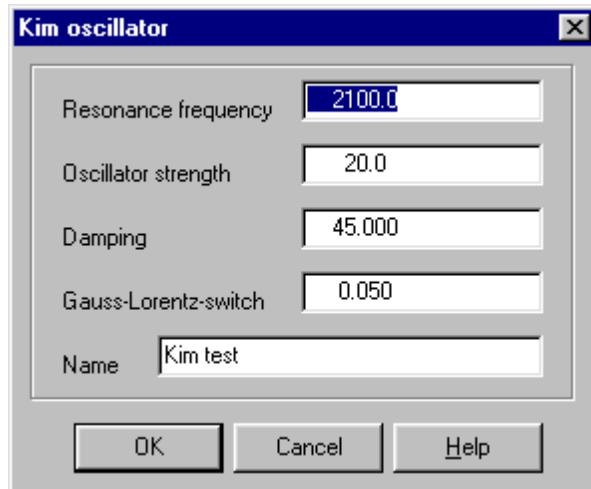
4.2.3.2.7 Kim oscillator

An extension of the simple [harmonic oscillator model](#) for vibrational modes suggested by Kim et al. [[Kim 1992](#)] allows a continuous shift of the line shape between a Gaussian and a Lorentzian profile. This is achieved by the following frequency dependence of the damping constant:

$$\chi_{\text{Kim oscillator}} = \frac{\Omega_p^2}{\Omega_{TO}^2 - \tilde{\nu}^2 - i\tilde{\nu}\tau(\tilde{\nu})} \quad \text{with} \quad \tau(\tilde{\nu}) = \Omega_\tau \exp\left(-\frac{1}{1+\sigma^2}\left(\frac{\tilde{\nu} - \Omega_{TO}}{\Omega_\tau}\right)^2\right)$$

The constant σ is called Gauss-Lorentz-switch. Like almost all fit parameters it may vary between 0 and infinity. For $\sigma = 0$ a Gaussian line shape is achieved. Large values of σ (larger than 5) lead to a Lorentzian.

The Kim model is similar to the Brendel oscillator which also gives lineshapes between a Gaussian and a Lorentzian. However, the computational effort is much less. Thus, if you have to work with many oscillators, the use of Kim oscillators instead of Brendel oscillators can save a lot of time. The four quantities of Kim oscillators can be set in the following dialog:



4.2.3.2.8 Gervais oscillator

An extension of the harmonic oscillator model described in [[Gervais 1974](#)] has been included in the SCOUT software since it turned out to be useful for the description of infrared active phonons. The original formula is

$$\varepsilon(\omega) = \varepsilon_\infty \prod_j \frac{\Omega_{jLO}^2 - \omega^2 - i\gamma_{jLO}\omega}{\Omega_{jTO}^2 - \omega^2 - i\gamma_{jTO}\omega}$$

and it represents the dielectric function as a product of individual oscillator terms. For each term there are four parameters, the resonance frequencies Ω_{jTO} of the TO modes (poles of the dielectric function), those of the LO modes (poles of the inverse dielectric function) Ω_{jLO} and a damping constant for each of the modes ($\gamma_{jTO}, \gamma_{jLO}$).

Due to multiplicative form of the expression it is not recommended to mix this model with others. The parameters of all ten possible oscillators and the dielectric background are set in a dialog box like the following:

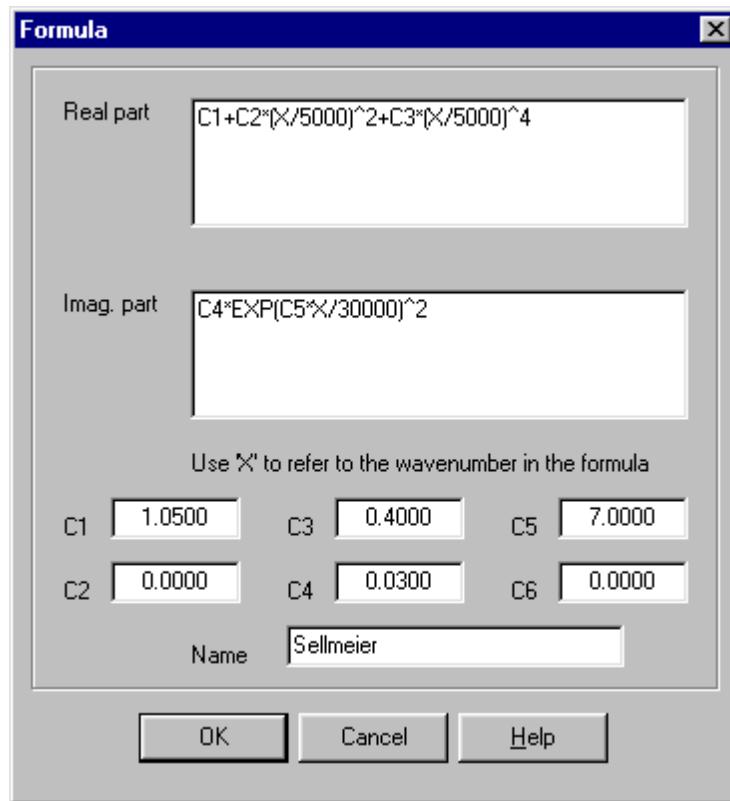
Gervais oscillator parameters					Dielectric background:	<input type="text" value="3.0"/>
	TO	damping TO	LO	damping LO	Name	
1	947.2	77.9415	1654.3	20.5500	Main phonon	
2	1400.0	50.0000	1500.0	100.0000		
3	0.0	0.0000	0.0	0.0000		
4	0.0	0.0000	0.0	0.0000		
5	0.0	0.0000	0.0	0.0000		
6	0.0	0.0000	0.0	0.0000		
7	0.0	0.0000	0.0	0.0000		
8	0.0	0.0000	0.0	0.0000		
9	0.0	0.0000	0.0	0.0000		
10	0.0	0.0000	0.0	0.0000		

Note that **only those parameters appear as fit parameters and are used for the computation which belong to an oscillator that was given a name**. In the example above the four parameters of the 'Main phonon' (and the dielectric background) would appear as possible fit parameters. The others are completely ignored by the program.

Warning: Not all combinations of the four parameters are reasonable. Large differences of the damping constants for the TO and LO modes may lead to an unphysical negative imaginary part of the dielectric function. This may also happen if the resonance frequency of the LO mode is chosen to be smaller than that of the TO mode.

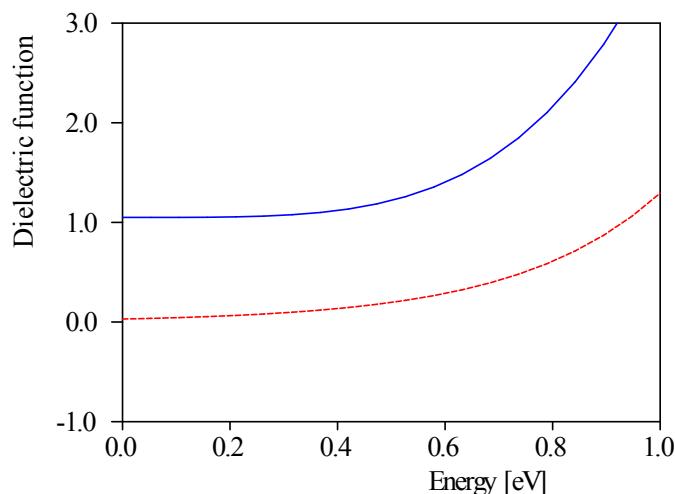
4.2.3.2.9 Formula

This susceptibility type gives most freedom to the user who can define arbitrary functions for real and imaginary part. The dialog looks like this:



In the formula you can refer to six constants labeled C1 ... C6 which appear in the list of possible fit parameters.

Frequency dependent expressions can be composed using the symbol X which represents the wavenumber. The dielectric function computed from the formulas of the sample dialog shown above is the following:



4.2.3.2.2.10 Imported data

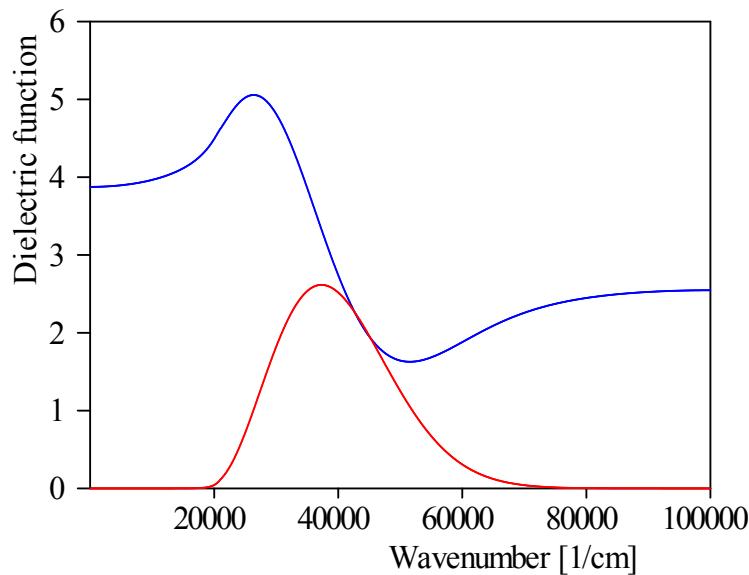
This susceptibility type is used to superimpose fixed, imported data for dielectric functions and flexible, parameterized susceptibilities. The 'Imported data' susceptibility contributes the fixed data to the composed dielectric function. The **Edit** command opens a window that shows the fixed data graphically. Here you can import dielectric function data sets.

There are no parameters and consequently no fit parameters available from this susceptibility type.

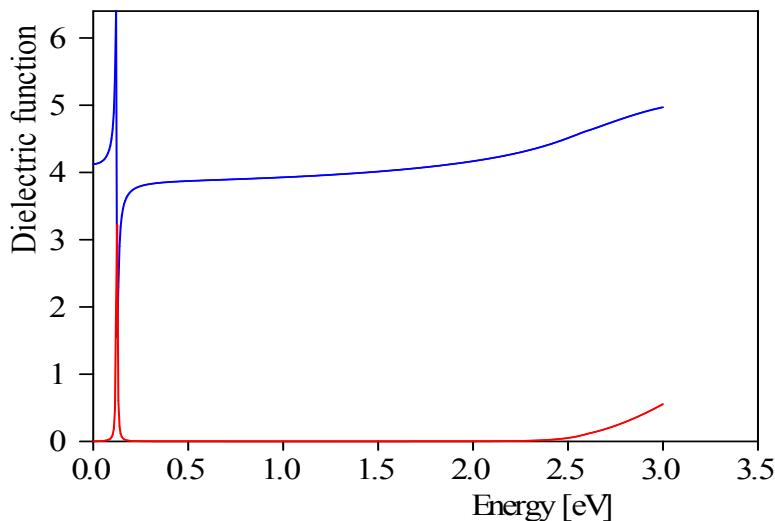
4.2.3.2.2.11 KKR susceptibility

Susceptibility objects of this type contain a complete KKR dielectric function model which is explained [below](#). The reason to have KKR dielectric function models also to appear as susceptibilities is the following: Often KKR dielectric function models must be computed on a very large spectral range which prohibits a high spectral resolution if one has to keep the number of data points low. If you use, for example, 2048 data points between 0 and 200 000 1/cm the spectral resolution is approximately 100 1/cm. If you have to describe infrared vibrational modes with the same model you run into severe difficulties with this resolution.

In these cases you can use KKR susceptibilities to describe e.g. interband transitions in the UV as in this example (1024 data points covering the range between 0 and 100 000 1/cm):



You can then use this susceptibility in a dielectric function model describing the visible and infrared properties with higher spectral resolution (where linear interpolation is used to get the required intermediate values for the real and imaginary part):



The parameters (and possible fit parameters) of KKR susceptibilities are exactly the same as those of the underlying KKR dielectric function model.

4.2.3.2.2.12 KKR susceptibility II

This susceptibility type is derived from the KKR susceptibility type which is described [above](#). The only difference is that after the computation of the real part (via the Kramers-Kronig transformation) the value of the real part for zero frequency is subtracted in the whole spectral range. This drags down the real part to zero for low wavenumbers.

Note that this behaviour of a susceptibility is quite unphysical. Usually any polarizability mechanism with some dynamics can instantly follow the slowly varying electric fields for low wavenumbers which is described by a positive real static dielectric constant. For high wavenumbers with time scales far below the response times of the mechanism no polarization is excited and the susceptibility should approach zero. In KKR susceptibility II objects the situation is - due to the subtraction of the static real part - reverse and hence not very recommendable from a physical point of view. However, there might be situations where the introduction of this susceptibility type could be useful.

Susceptibilities of this type must be combined with other contributions that add to the real part of the dielectric functions. Good choices are the types '[Constant](#)' and '[Constant refractive index](#)'.

4.2.3.3 KKR dielectric function models

4.2.3.3.1 Introduction

Important note: It is strongly recommended not to use KKR dielectric function models in the list of materials directly. Please work with objects of type 'Dielectric function model' and use in their susceptibility list a KKR susceptibility. This kind of susceptibility makes use of a KKR dielectric function model internally.

In some cases one has only a formula for the imaginary part of the dielectric function. This is especially true for interband transitions where quantum mechanical expressions can be derived from the band structure of a material.

Fortunately, with a little luck the missing real part can be constructed from the imaginary part using the Kramers-Kronig-Relation (KKR) that connects real and imaginary part of susceptibilities.

Of course, there are some restrictions and requirements. The imaginary part of the dielectric function

must be known in a wide spectral range. To be specific, from 0 to infinity. Using a finite maximum frequency instead of infinity is acceptable only if the imaginary part approaches smoothly zero as its high frequency limit.

In addition, the KKR relation can only be used with certain numbers of data points, namely powers of 2 (256, 512, 1024, 2048, 4096, 8192, 16384). This is due to the fact that the Hilbert transformation that brings you from the imaginary to the real part of the susceptibility is realized in SCOUT using two successive Fourier transforms. These are programmed as so-called Fast Fourier Transforms (FFT) which are limited to the restricted number of data points.

Using KKR dielectric function models you can mix susceptibility types that require a KKR construction of the real part with those that have explicit expressions for real and imaginary part. The algorithm used for the computation of the final dielectric function is the following:

- First the imaginary parts of all KKR susceptibilities are summed up
- Then the KKR step is done to construct the corresponding real part
- Then all the 'normal' susceptibility terms are added that do not require a KKR transformation

Although the computational effort is large the time to perform the KKR on modern computers is still in a range that you should definitely consider using KKR dielectric function models for 'fitting' work. This is especially true for UV spectroscopy.

Up to now the following KKR susceptibilities have been implemented:

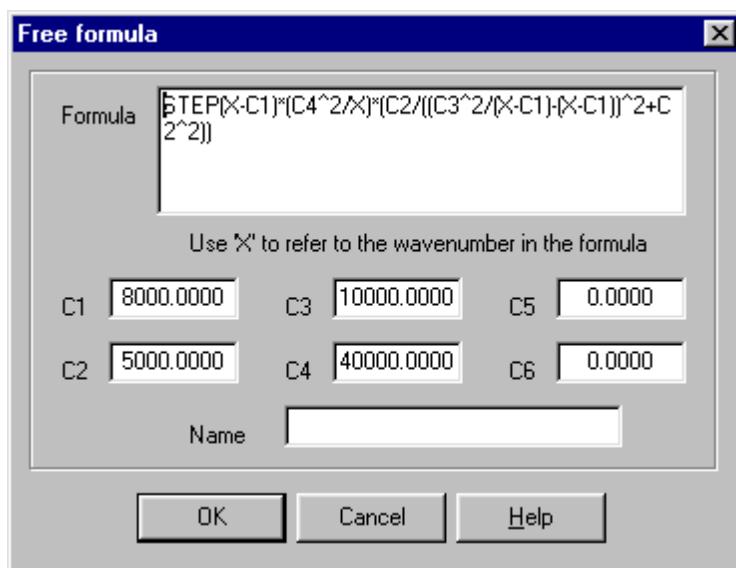
- [Free formula](#)
- [OJL model](#)
- [Campi_Coriasso](#)
- [Tauc-Lorentz](#)

4.2.3.3.2 Susceptibilities

4.2.3.3.2.1 Free formula

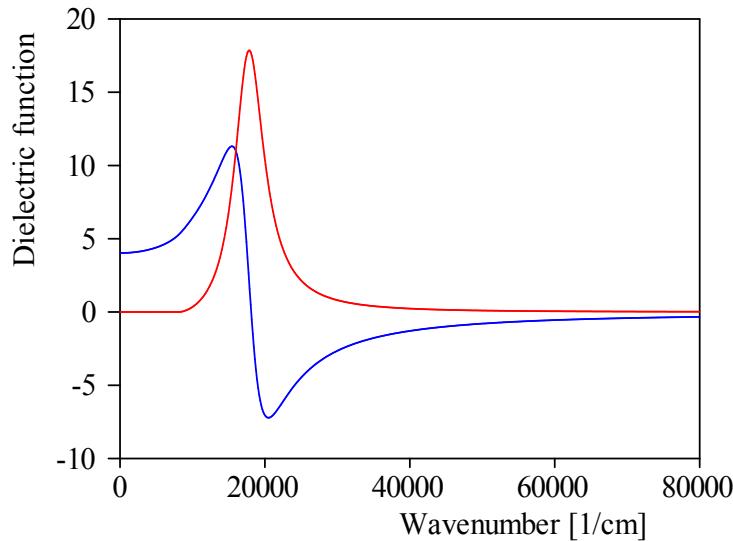
This is the most flexible KKR susceptibility since you can define an arbitrary function for the imaginary part, specified by a user-defined [formula](#).

The SCOUT dialog is this:



The formula string represents the formula, of course, in which you can use up to 6 constants (C1, ..., C6) which all appear as possible fit parameters. You will need to refer to the wavenumber in the formula which can be done by the symbol X.

The formula in the dialog above is taken from literature [[Campi 1988](#)] and leads to the following dielectric function:

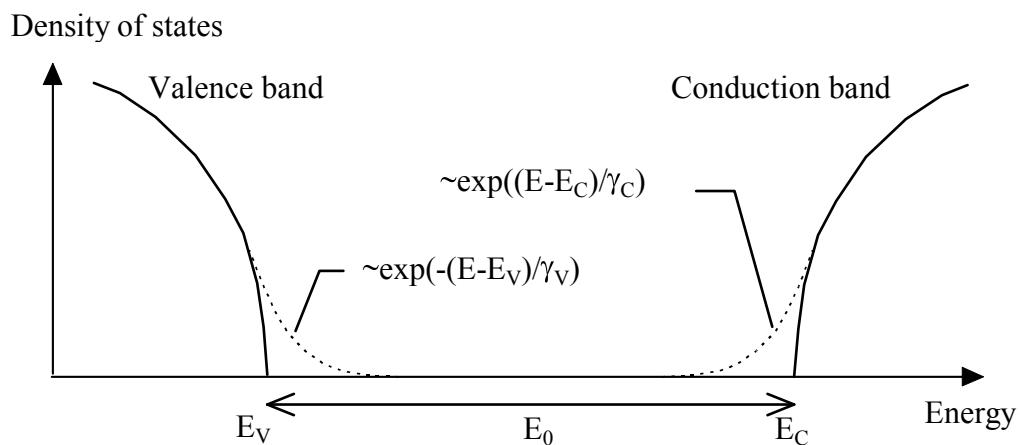


Note that this model ([Campi-Coriasso](#)) is also available as a built-in susceptibility.

4.2.3.3.2.2 OJL2 model

This model replaces the OJL model of previous program versions which had a small bug in the computation of the optical constants. If you import old configurations containing OJL susceptibilities the old model is still used, however.

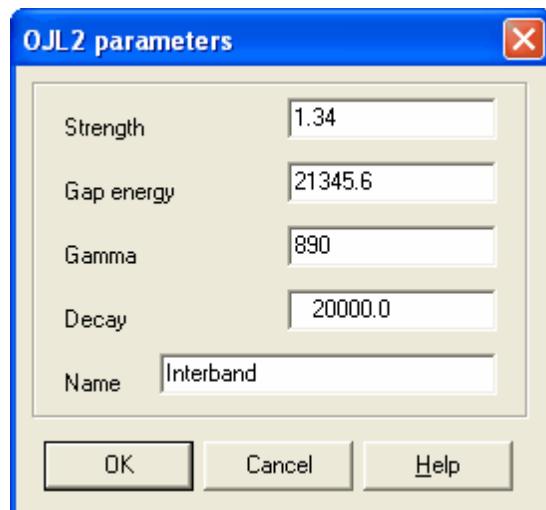
The so-called OJL model for interband transitions is due to [[O'Leary 1997A](#)] where expressions for the joint density of states are given for the optical transition from the valence band to the conduction band. Parabolic bands are assumed with tail states exponentially decaying into the band gap as sketched in the following graph:



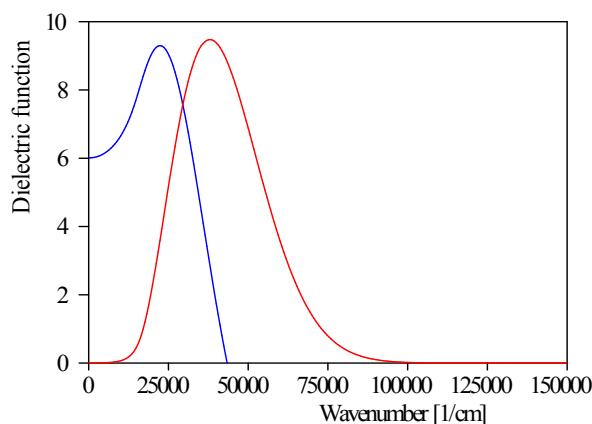
The fit parameters of the OJL interband transition model are the gap energy E_0 , the tail state exponent of the valence band γ and the overall strength of the transition.

The imaginary part of the dielectric function is proportional to the combined density of states obtained from the above assumption on the density of states of the valence and conduction band. Without modification, the original expression for the imaginary part of the dielectric function would increase to infinity with increasing energy. Due to this unrealistic behaviour the construction of the real part of the dielectric function by two Fourier transforms is not possible. Therefore, a so-called decay parameter has been added to the original expressions to drag down the imaginary part to zero for high frequencies. The OJL term is simply multiplied by the exponential $\exp(-\text{sqr}(\text{wavenumber}-E_0)/\text{decay}))$.

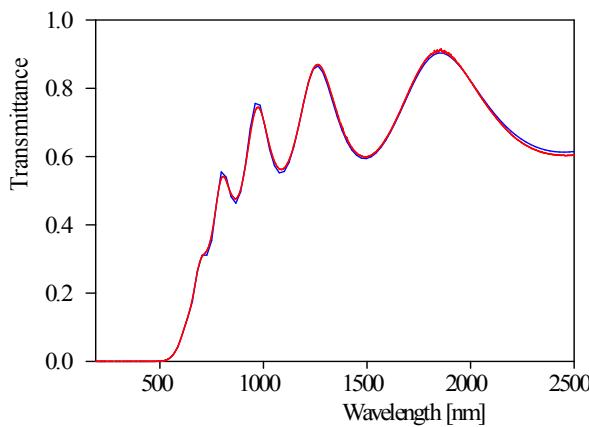
The SCOUT dialog to set these parameters is the following:



The gap energy, gamma and the decay constant are to be specified in wavenumbers (1/cm). An OJL dielectric function example is shown below:



In this case the dielectric function had been adjusted to a transmission spectrum of a thin amorphous film on a microscope glass slide:

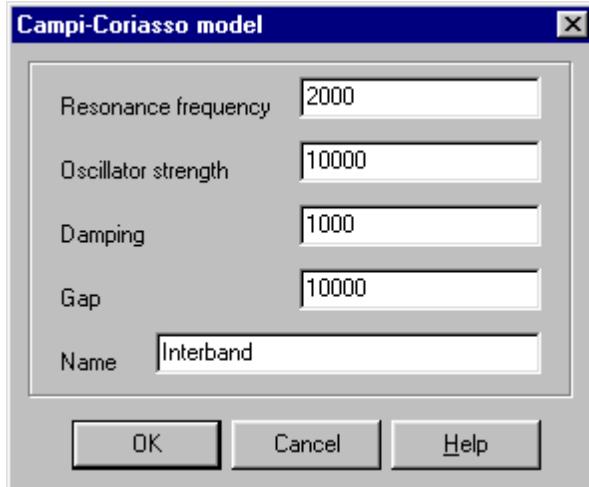


4.2.3.3.2.3 Campi-Coriasso model

The interband transition model suggested by Campi and Coriasso [[Campi 1988A](#)] leads to the following expression for the imaginary part of the susceptibility:

$$\chi_j(\omega) = \frac{1}{\omega} \frac{\omega_p^2 \omega_\tau (\omega - \omega_{Gap})^2}{[(\omega - \omega_{Gap})^2 - \omega_0^2]^2 + (\omega - \omega_{Gap})^2 \omega_\tau^2} \Theta(\omega - \omega_{Gap})$$

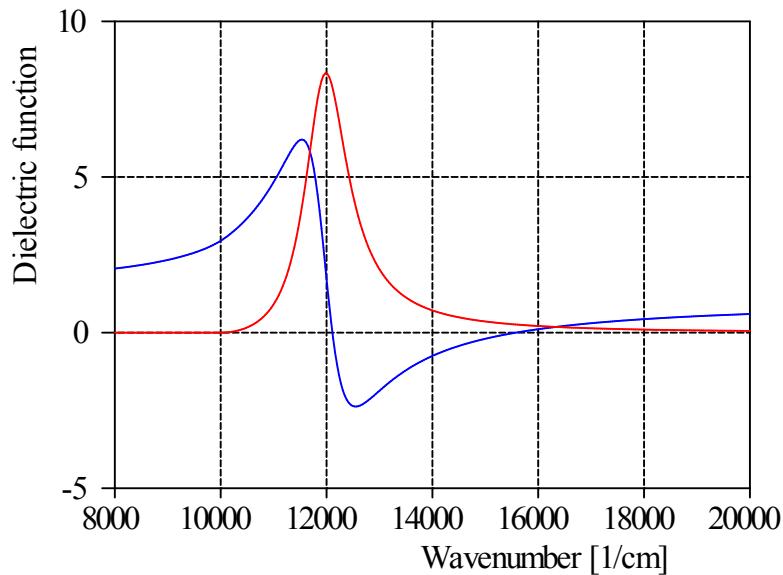
The 4 parameters of the model must be entered using wavenumber units in the following dialog:



Here the following assignments apply:

ω_p	Oscillator strength
ω_τ	Damping constant
ω_0	Resonance frequency
ω_{Gap}	Gap energy

The following example shows the dielectric function obtained with this model (the parameters are those displayed in the dialog above) where a constant dielectric background of 1 has been added:



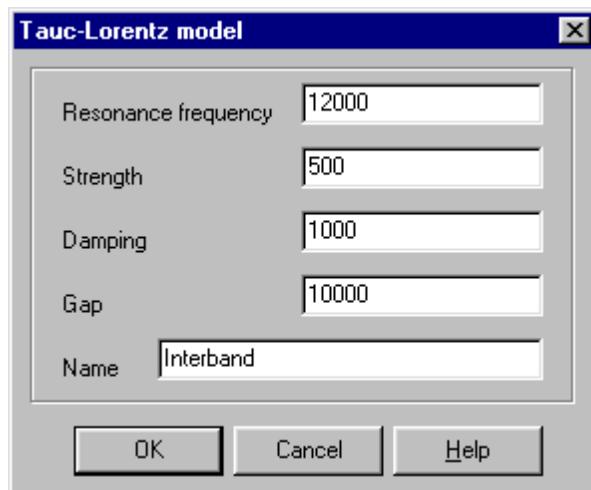
Note that the peak of the imaginary part occurs at the sum of the gap and the resonance frequency. Below the gap the imaginary part is exactly zero, i.e. there is no absorption. For large resonance frequencies the Campi-Coriasso model approaches the harmonic oscillator model.

4.2.3.3.2.4 Tauc-Lorentz model

Another suggestion for an interband transition model was given by [[Jellison 1998A](#)]. This model is called the Tauc-Lorentz model and gives the following expression for the imaginary part of the susceptibility:

$$\chi_j(\omega) = \frac{1}{\omega} \frac{S^2 \omega_0 \omega_\tau (\omega - \omega_{Gap})^2}{(\omega^2 - \omega_0^2)^2 + \omega^2 \omega_\tau^2} \Theta(\omega - \omega_{Gap})$$

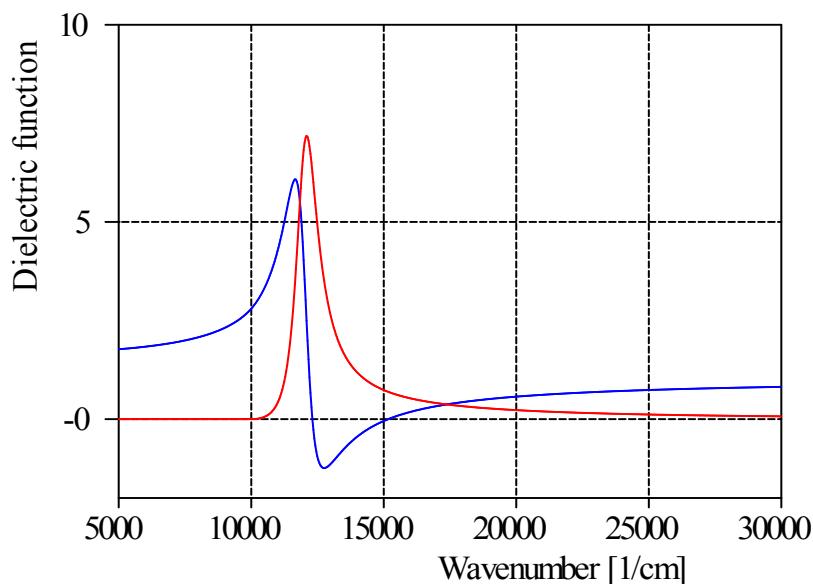
The 4 parameters of the model must be entered using wavenumber units (except for S which is dimensionless) in the following dialog:



The following assignments apply:

S	Strength (in the Jellison paper a constant A was used which is the square of S)
ω_τ	Damping constant
ω_0	Resonance frequency
ω_{Gap}	Gap energy

The following example shows the dielectric function obtained with this model (the parameters are those displayed in the dialog above) where a constant dielectric background of 1 has been added:



Note that the peak of the imaginary part occurs at the resonance frequency. Below the gap the imaginary part is exactly zero, i.e. there is no absorption.

4.2.3.4 Master models

4.2.3.4.1 Introduction

Master model objects are dielectric function models with the additional feature that you can compute the values of selected susceptibility parameters from user-defined parameters by user-defined formulas. This provides a very high modelling flexibility.

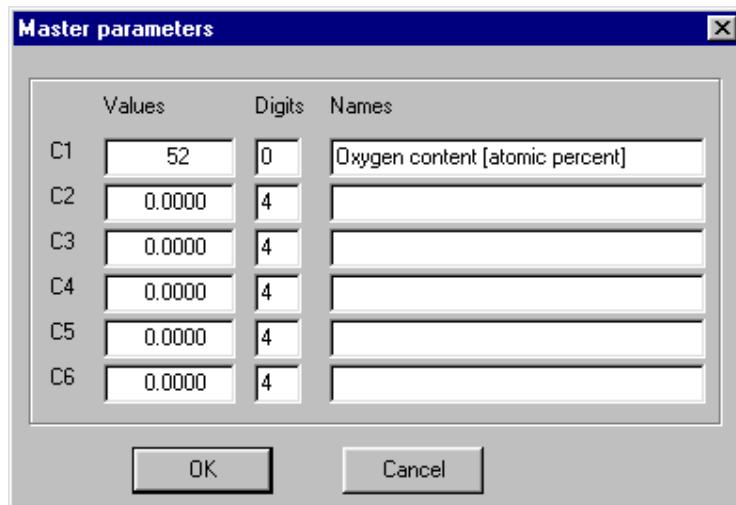
There are two types of situations where master models are useful. In the first one the parameters that you want to obtain from the optical analysis are not those that appear in the susceptibility types that the program provides. A typical example is the determination of carrier concentrations by infrared spectroscopy: This quantity is proportional to the square of the Drude parameter plasma frequency which is the fit parameter to be used here. In master model objects you can define a fit parameter called 'Carrier concentration' and compute from its value the plasma frequency of the Drude model which is then used in the calculation of the optical spectra. The user-defined fit parameters are called master parameters whereas the model parameters computed from the master parameters are called slave parameters.

In the second case of problems you want to control many model parameters by only a few master parameters. A good example is the description of amorphous oxides for various oxygen contents. If

you use the [OJL](#) model for the interband transition and you investigated already how the OJL parameters like gap energy, mass and gamma depend on the oxygen concentration in the oxide, you may want to describe the oxide by only one parameter, namely the oxygen content. Applying a master model object you can define the fit parameter 'Oxygen content' and specify for all model parameters that vary with the oxygen content how they depend on the master parameter by suitable expressions. This way a complex optical constant model can be completely controlled by only one parameter.

4.2.3.4.2 Master parameters

In the master model object window you can access the master parameters by Parameters|Masters which opens the following dialog:



You can define up to 6 master parameters each of which has a name, a value and the number of digits that is used to display the master parameter's value. In the formulas of the [slave parameters](#) you can refer to the master parameters by C1, C2, ... and C6, respectively. The master parameters appear as fit parameters of the master model objects.

4.2.3.4.3 Slave parameters

Master model objects have a list of slave parameters the values of which are computed from the [master parameters](#) by user-defined formulas or user-defined lookup tables. Slave parameters are parameters of the susceptibility contributions that make up the dielectric function model. The list of slave parameters may look like this:

Value	Name	Formula
21675.9980	Tauc I: resonance frequency	$249620*C1^6-778128*C1^5+827957*C1^4-336811*C1^3+32586*C1^2+15932*C1+10932$
702.3641	Tauc I: strength	$-30466*C1^6+490905*C1^5-94762*C1^4+38727*C1^3-4427.9*C1^2+254.2*C1+517.88$
5907.6074	Tauc I: damping	$-96861*C1^6+226811*C1^5-196992*C1^4+78137*C1^3-13545*C1^2+244.61*C1+6132.6$
19296.8594	Tauc I: gap	$-30317*C1^6+70638*C1^5-52968*C1^4+16313*C1^3-3422.1*C1^2+10969*C1+10856$
25952.9434	Tauc II: resonance frequency	$-221370*C1^6+590075*C1^5-599315*C1^4+292475*C1^3-57903*C1^2-406.09*C1+23743$

This list is almost the same as the fit parameter list of SCOUT. With **New** (or pressing the corresponding speed button '+') you get a list of all susceptibility parameters from which you can select those that are to be slave parameters (remember that you can select many items at a time keeping the Ctrl-key pressed while clicking items). In the example above (the AlGaAs model in the database delivered with SCOUT) several slave parameters are chosen which are computed from the first master parameter (C1, this is the Al content ranging from 0 to 1) according to the specified formulas.

Instead of using a formula, you can also define the relation between the master parameter and a slave parameter by defining a lookup table (at the moment, this is only possible for the first master parameter C1). This feature is especially useful if you have analyzed a sample series with varying composition (= variation of the master parameter) in the batch control window: In this case you have already a table of slave values for certain master parameter values. Making use of the batch control results, create a table like the following in the workbook (don't mix up the workbook and the batch control window - these are different objects in SCOUT):

S2 Workbook							
	File	Copy	Paste	Clear	Graph	Autosize cells	Test
B41	0						
38							
39	AlxGa(1-x)As example						
40							
41	Al content	0	0.2	0.4	0.6	0.8	1
42	Tauc II: resonance frequency	23743.0000	22901.2148	22827.6563	23713.6348	25952.9434	25952.9434
43							
44							
45							
Slave parameters Integral quantities Object info Sheet1							

In order to use this relation for the parameter 'Tauc II: resonance frequency' select the cell B41 in the workbook as shown and select the parameter 'Tauc II: resonance frequency' in the list of slave parameters. Then use the command **Lookup table|Read from workbook rows** in order to import the C1 master parameter values (denoted as 'Al content' in the table) from the selected row (SCOUT will read to the right until the first empty cell is found) and the corresponding slave parameter values from the row below (denoted as 'Tauc II: resonance frequency'). The lookup table is used when there is no formula that defines the relation of master and slave parameter:

S2 Slave parameters		
File New Edit Delete Update Color Lookup table ?		
Value	Name	Formula
1	Tauc I: resonance frequency	$249620*C1^6-778128*C1^5+827957*C1^4-336811*C1^3+32586*C1^2+15932*C1+10932$
2	Tauc I: strength	$-30466*C1^6+90905*C1^5-94762*C1^4+38727*C1^3-4427.9*C1^2+254.2*C1+517.88$
3	Tauc I: damping	$-96861*C1^6+226811*C1^5-196992*C1^4+78137*C1^3-13545*C1^2+244.61*C1+6132.6$
4	Tauc I: gap	$-30317*C1^6+70638*C1^5-52968*C1^4+16313*C1^3-3422.1*C1^2+10969*C1+10856$
5	Tauc II: resonance frequency	$46231*C1^6+124490*C1^5-121520*C1^4+52079*C1^3-10169*C1^2+1036.7*C1+779.44$
6	Tauc II: strength	$-701312.8*C1^6-2106822.7*C1^5+2389532.5*C1^4-1266546.6*C1^3+324992.2*C1^2-55735.9*C1$
7	Tauc II: damping	

In order to check or modify the lookup table of a slave parameter, select it in the list of slave parameters and use the command **Lookup table|Write to workbook columns** or **Lookup table|Write to workbook rows**. The data will be written starting at the current cursor position in the workbook.

In addition to a lookup table import for every slave parameter separately, you can import lookup tables for all slave parameters in one step if you prepare a corresponding data block in the workbook. Here is an example:

S2 Workbook							
	A	B	C	D	E	F	G
1							
2	AI content	0	0.2	0.4	0.6	0.8	1
3	Tauc I: resonance frequency	10932.0000	13819.0576	15212.7676	17913.2500	21675.9980	21675.9980
4	Tauc I: strength	517.8800	576.9406	769.7953	807.5839	702.3641	702.3641
5	Tauc I: damping	6132.6001	6016.0112	5946.8188	5888.2715	5907.6074	5907.6074
6	Tauc I: gap	10856.0000	12979.3350	14983.2695	16942.7402	19296.8594	19296.8594
7	Tauc II: resonance frequency	23743.0000	22901.2148	22827.6563	23713.6348	25952.9434	25952.9434
8	Tauc II: strength	779.4400	839.0980	874.6394	764.0809	664.2000	664.2000
9	Tauc II: damping	17077.6992	11991.7881	8193.8545	5636.6152	4245.9209	4245.9209
10	Tauc II: gap	14057.2002	16655.1973	18793.7109	20315.3281	21680.4316	21680.4316
11	Tauc III: resonance frequency	23366.0000	24600.6653	25697.4551	26864.1367	27998.8438	27998.8438
12	Tauc III: strength	460.4500	499.9493	630.3735	551.7571	172.8235	172.8235
13	Tauc III: damping	1565.0000	2177.0498	2567.7915	2382.4536	2007.1764	2007.1764
14	Tauc III: gap	17541.8301	18307.9160	19879.0684	20317.1563	20144.4219	20144.4219
15	Tauc IV: resonance frequency	29713.9004	28785.0039	29871.1133	31562.4746	31418.2109	31418.2109
16	Tauc IV: strength	785.4000	1726.9529	1552.5227	1613.0676	1651.3461	1651.3461
17	Tauc IV: damping	9211.7998	8478.5938	8718.3975	11199.8516	9030.5703	9030.5703
18	Tauc IV: gap	24293.0000	27068.8691	26425.9453	26435.3711	26473.9629	26473.9629
19	Tauc V: resonance frequency	24438.0000	26141.7520	27310.3398	28484.1543	29414.4668	29414.4668
20	Tauc V: strength	1444.4000	1131.5055	932.9470	905.1089	848.5321	848.5321
21	Tauc V: damping	3568.7000	3961.2588	3670.6794	3912.3513	3803.7571	3803.7571
22	Tauc V: gap	21664.0000	21925.5859	22494.2168	22718.1895	22711.9492	22711.9492
23	Tauc VI: resonance frequency	36766.0000	36636.0078	36685.3320	37465.4258	37441.3945	37441.3945
24	Tauc VI: strength	640.6500	603.1868	582.4344	544.5126	579.3859	579.3859
25	Tauc VI: damping	5169.2998	4426.0342	4203.0020	3268.0068	3254.8057	3254.8057
26	Tauc VI: gap	20753.9004	23932.0605	24130.9277	23551.2969	24257.6250	24257.6250
27	Tauc VII: resonance frequency	39040.1016	38555.6367	38600.6914	39292.5820	39525.0859	39525.0859
28	Tauc VII: strength	883.3100	508.7683	588.9135	363.2008	379.7808	379.7808
29	Tauc VII: damping	2682.3000	3369.5930	3661.1416	3577.9028	3303.3755	3303.3755
30	Tauc VII: gap	31620.9004	23886.3635	25145.0840	23396.4980	23814.8047	23814.8047
31	Tauc VIII: resonance frequency	49702.0000	54939.0742	53634.7695	50921.8047	51529.2773	51529.2773
32	Tauc VIII: strength	567.3600	476.8036	485.4659	507.7878	483.2701	483.2701
33	Tauc VIII: damping	9762.9004	3783.2356	3865.2429	2755.9873	2592.1309	2592.1309
34	Tauc VIII: gap	34820.8008	35940.7891	37048.4023	40236.3359	39954.3984	39954.3984
35	db: real part	0.6650	0.4869	0.9420	0.8204	1.0983	1.0983
36							
37							

Slave parameters Integral quantities Object info Sheet

Note that the sequence of parameters in the workbook and the slave parameter list must be exactly the same. In order to import the lookup tables, place the workbook cursor in the cell B2 (as shown above) and use the command **Lookup table|Read all from workbook rows** in the slave parameter list. Finally delete all formulas in the slave parameter list:

The screenshot shows a software dialog titled "S2 Slave parameters". The menu bar includes File, New, Edit, Delete, Update, Color, Lookup table, and ?. Below the menu is a toolbar with icons for file operations and a search bar. A main table lists 15 slave parameters:

	Value	Name	Formula
3	5898.1904	Tauc I: damping	
4	16196.3779	Tauc I: gap	
5	23376.1660	Tauc II: resonance frequency	
6	806.1926	Tauc II: strength	
7	6610.6680	Tauc II: damping	
8	19735.7441	Tauc II: gap	
9	26419.7480	Tauc III: resonance frequency	
10	581.7021	Tauc III: strength	
11	2453.0488	Tauc III: damping	
12	20150.2891	Tauc III: gap	
13	30918.2344	Tauc IV: resonance frequency	
14	1590.0061	Tauc IV: strength	
15	10054.6650	Tauc IV: damping	

Now all slave parameters are computed using their lookup tables.

If the set of master parameter values for the lookup tables of all slave parameters are the same, you can write the current lookup tables to one block in the workbook. The command is **Lookup table| Write all to workbook rows**. You can then modify the relations in the workbook and re-import the modified lookup tables as shown above.

4.2.3.4.4 Fit parameters

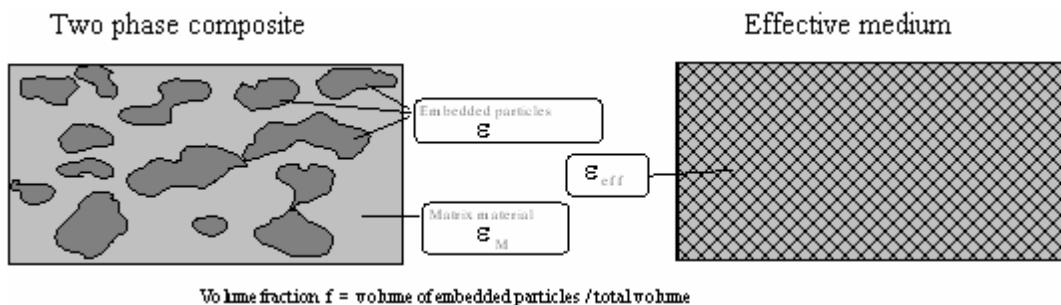
The fit parameters of master models are the 6 master parameters and, in addition, all those susceptibility parameters which do not appear as slave parameters. Hence you can work with master models the same way as with dielectric function models fitting 'normal' susceptibility parameters to measured data. At any time you can introduce master and slave parameters and refine or simplify your choice of parameters.

4.2.3.5 Effective dielectric functions

4.2.3.5.1 Introduction

The optical properties of inhomogeneous materials can be described by so-called effective dielectric functions if the wavelength of the probing radiation is much larger than the typical sizes of the inhomogeneities of the system [Theiss 1994A, Theiss 1995B]. In this case retardation effects can be neglected and the response of the mixed material to an incoming electromagnetic wave can be calculated in a quasi-static approximation, i.e. one has to answer the question which volume averaged polarization will exist in the sample in response to an applied static electric field. Macroscopically the inhomogeneities cannot be seen (there is no light scattering in the long wavelength limit) and the system can be treated then as being quasi-homogeneous. An effective dielectric function ϵ_{eff} can be introduced which is a non-trivial average of the dielectric functions of the individual components.

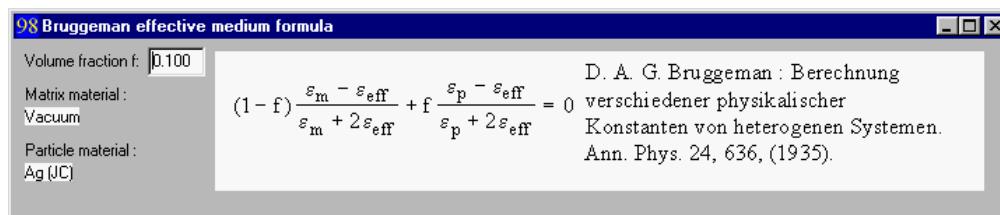
For the case of a two-phase composite the situation is shown below:



Obviously the microgeometry plays an important role for the effective dielectric function: if, for example, the embedded particles are metallic and the host material is an insulator the effective medium can show metallic or insulating behaviour, depending on whether there is a percolating network of the embedded particles or not. But also other properties of the topology may influence the effective dielectric function significantly.

Several very simple 'mixing formulas' have been established, the most prominent being the ones of **Maxwell Garnett** [Maxwell Garnett 1903A] and **Bruggeman** [Bruggeman 1935A], both being implemented in SCOUT. The model due to **Looyenga** [Looyenga] has been added also since it leads for some typical microstructures to good results. All these simple effective medium concepts use just one parameter to characterize the microgeometry, namely the volume fraction f of the embedded particles. Due to this simplification in many cases quite wrong results are the consequence. The most general concept due to **Bergman** [Bergman 1978A]/[Theiss 1994A] holds in any case (as long as the electrostatic approximation is valid) and is therefore included in SCOUT with best recommendations from the author.

All effective dielectric function objects need to point to two dielectric functions, namely those of the host material and the embedded particles. The assignments are usually done using drag&drop operations like in the assignment of dielectric functions to layers in layer stacks. To do so, you have to open both the dielectric function list and the parameter window of the effective dielectric function. In case of a 'Bruggeman object' this looks like the following:



Start the drag operation by pointing on the wanted material in the dielectric function list, press the left mouse button and keep it down while you move the mouse cursor to the effective medium parameter window. If the material is going to be the matrix material place the cursor in the white rectangle below the text 'Matrix material' and release the mouse button. In the case shown above vacuum was selected as matrix material. In an analogous way, drag and drop the particle material to the white area under the text 'Particle material'.

4.2.3.5.2 Maxwell Garnett

The Maxwell Garnett formula [Maxwell Garnett 1903] is the following:

$$\frac{\epsilon_{\text{eff}} - \epsilon_M}{\epsilon_{\text{eff}} + 2\epsilon_M} = f \frac{\epsilon - \epsilon_M}{\epsilon + 2\epsilon_M}$$

It is applicable to systems of low volume fractions of the embedded particles which must also be far

away from each other.

4.2.3.5.3 Bruggeman

The Bruggeman theory (also known as 'EMA' for effective medium approximation) gives the relation

$$(1-f) \frac{\varepsilon_M - \varepsilon_{\text{eff}}}{\varepsilon_M + 2\varepsilon_{\text{eff}}} + f \frac{\varepsilon - \varepsilon_{\text{eff}}}{\varepsilon + 2\varepsilon_{\text{eff}}} = 0$$

and is probably the most often used effective medium concept. For very low volume fractions it is equivalent to the Maxwell Garnett theory, but for increasing volume fraction its results are quite different. Below $f=1/3$ there is no percolation, above that threshold the embedded particles are assumed to be partially connected.

4.2.3.5.4 Looyenga

The Looyenga formula

$$\varepsilon_{\text{eff}}^{\vee_3} = f \varepsilon^{\vee_3} + (1-f) \varepsilon_M^{\vee_3}$$

which is quite unknown is included in SCOUT since it has some features being useful for certain cases. For any volume fraction the expression implies a percolation! This means one can try to use it for very loosely packed systems which have nevertheless a connected network of the particles, e. g. metal blacks.

4.2.3.5.5 Bergman representation

The so-called **Bergman representation of effective dielectric functions** is the most general form of effective medium approaches:

$$\varepsilon_{\text{eff}} = \varepsilon_M (1 - f \int_0^1 \frac{g(n, f)}{t-n} dn) \quad \text{with } t = \frac{\varepsilon_M}{\varepsilon_M - \varepsilon}$$

The function $g(n, f)$ is called spectral density and holds all topological details of the microgeometry. It is a real, non-negative function, normalized in the interval $[0, 1]$ for n . In the case of isotropy there is also a condition for the first moment of the spectral density:

$$\int_0^1 g(n, f) dn = 1 \quad \text{and} \quad \int_0^1 n g(n, f) dn = \frac{1-f}{3}$$

Any microgeometry is represented by a certain spectral density. However, in general it cannot be computed exactly for real systems since the topology is too complicated (and in most cases not even known).

To gain some insight into the relation of spectral density and microtopology some research projects in the past had the goal to fit theoretical optical spectra (of systems with known optical constants of the matrix and particle materials) to experimental ones by adjusting $g(n, f)$. This way a lot of experience has been achieved being useful in the treatment of new two-phase composites.

In many cases an important property of the microgeometry is the 'degree of connectivity' of the embedded particles which usually is called percolation strength. Especially metal-insulator composites can switch from metallic to dielectric behaviour when the percolation strength is

reduced. It can be shown that this feature shows up in the function $g(n,f)$ as a δ -function at $n=0$. Therefore, a separation of this diverging term from the continuous part of the spectral density is useful:

$$g(n,f) = g_0(f) \delta(n) + g_{\text{cont}}(n,f)$$

The factor $g_0(f)$ is called percolation strength.

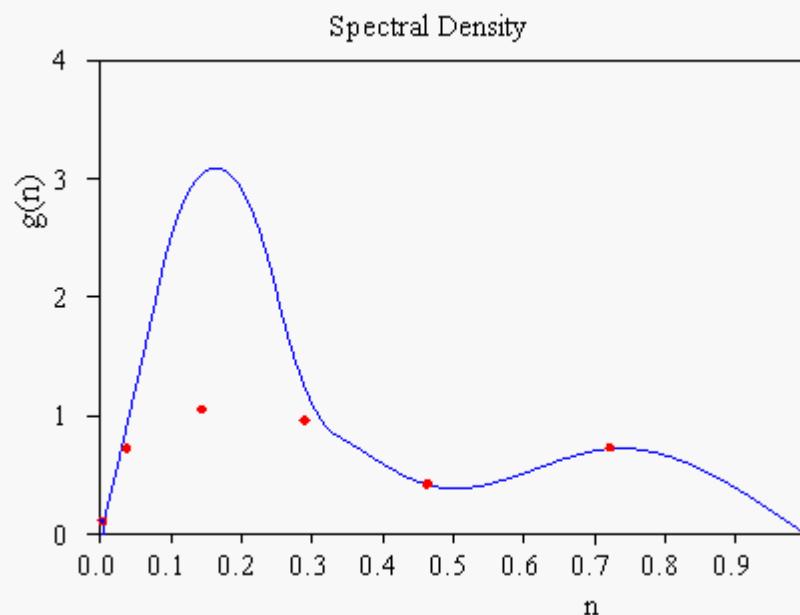
Implementation

The two parameters volume fraction f and percolation strength $g_0(f)$ are specified in the **Parameters** dialog of Bergman representation objects in SCOUT. Then, to adjust spectral densities to fit experimental data, one has to parameterize the shape of the spectral density in a suitable way. We have chosen to do that in SCOUT by a set of definition points. Each point is defined by a value of n and a corresponding $g(n)$ value. You can use an arbitrary number of definition points. The position of the points can be set 'graphically', i.e. by the mouse.

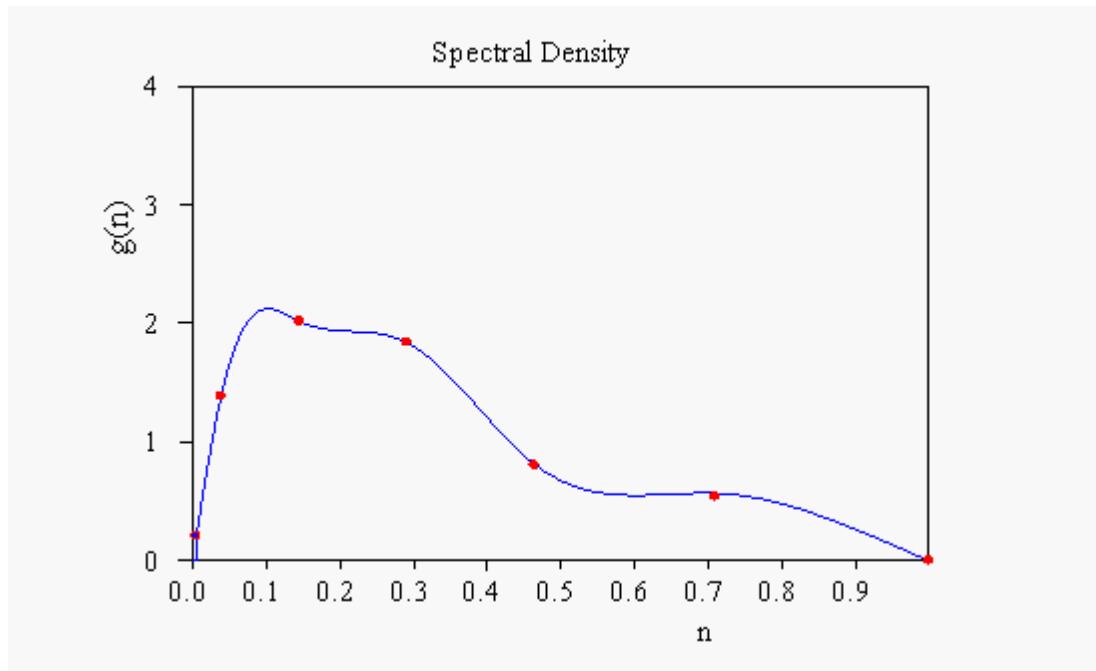
The shape of the spectral density is then obtained by cubic spline interpolation through the definition points. Usually, any user-defined shape (drawn visually with the mouse) will violate the restrictions mentioned above (normalization, coupling of the first moment of the spectral density to the volume fraction). Therefore, SCOUT adds a smooth peak to the user-defined curve to ensure the proper value of the 1st moment. This is followed by an automatic rescaling which normalizes the spectral density.

The **Points** submenu contains some commands to control the definition points. **Add points** brings up a mode where a mouse click creates a new definition point at the mouse cursor position. After the **Move points** command you can catch an already existing point with the cursor and drag it to a new position. If the **Continuous update** option is selected the spectral density shape is updated with each mouse move. If the option is switched off, the spectral density is updated when the mouse button is released at the final position of the definition point. Clicking at points after **Delete points** deletes the points. **Delete all points** deletes all points and starts with three new definition points at default positions.

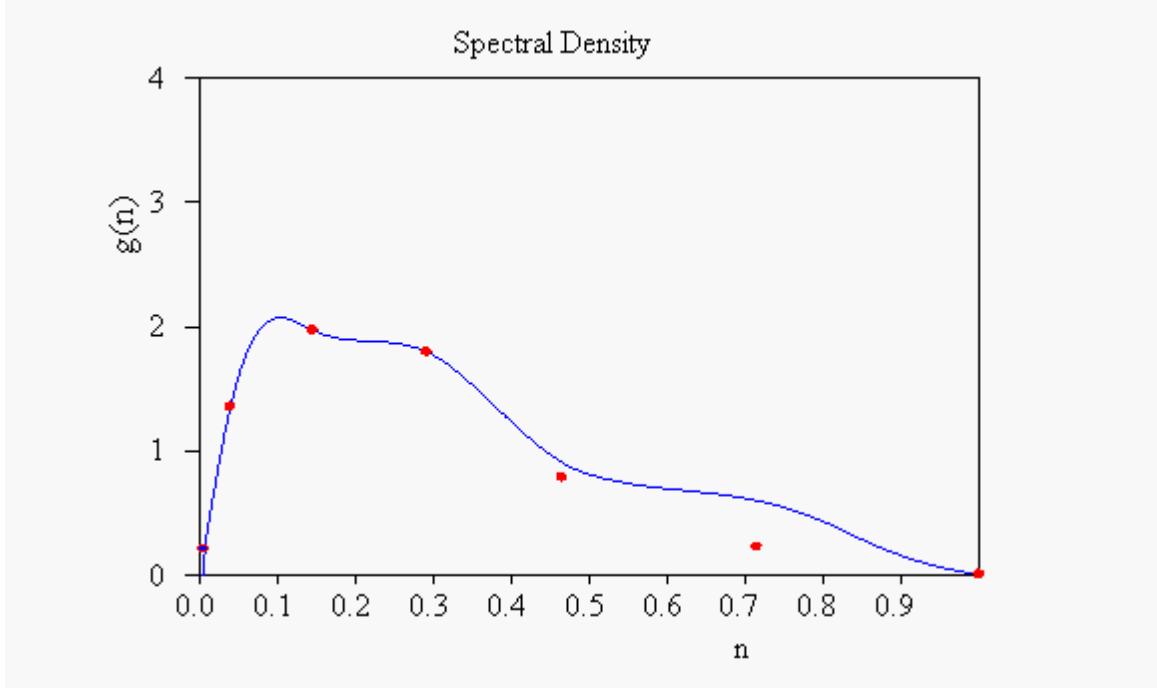
A typical situation is this:



Here the user-defined curve had too much weight on the right side, hence SCOUT added the peak to the left. If you move down the point at $n=0.72$ you reduce the weight on the right side and the peak on the left side is not needed any more:

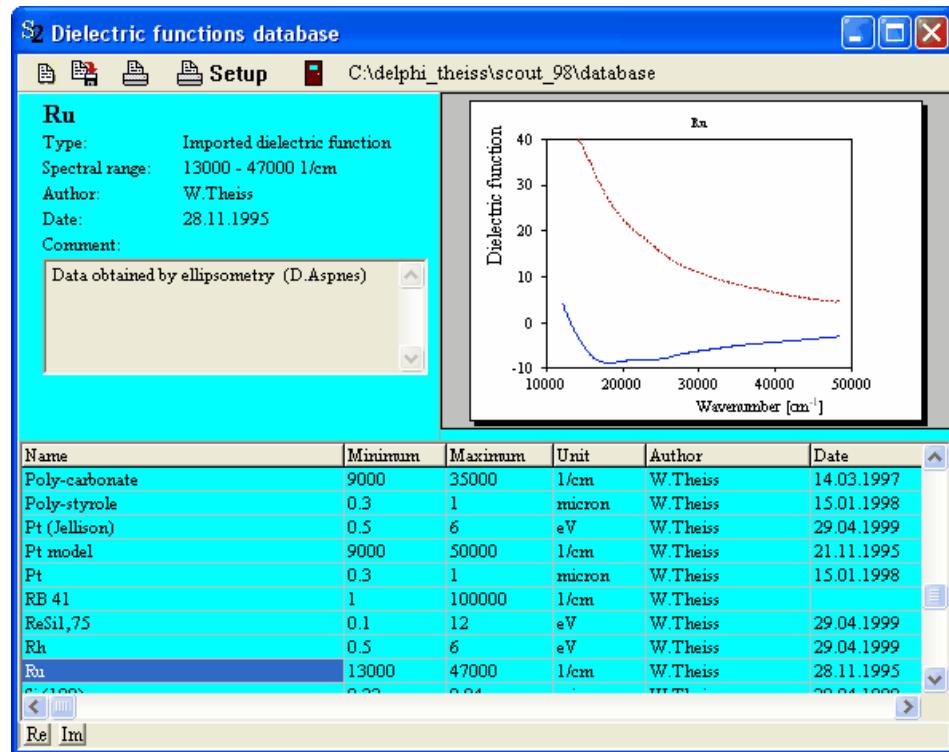


Note the re-scaling of the height of the definition points. If you reduce the height of that point further on, SCOUT has to add something on the right side to achieve the correct value of the 1st moment:



4.2.4 Database connection

The material list of SCOUT can exchange objects with the [database](#). The **Database** menu command opens a window showing the records in the database. At program start SCOUT loads the default database that was created during setup (located in the chosen program directory). In the **Options** submenu of SCOUT you can connect to a database of your choice. The path to the active database is indicated in the upper right corner of the database window.



The exchange of material objects is done quite easy with drag and drop operations.

From the database to SCOUT

To export a material from the database to the SCOUT material list, select the corresponding record in the database table and start a drag&drop operation (i.e. press the left mouse button). Then move (while the left mouse button is pressed) the cursor to the material list and release the mouse button, i.e. drop the material object there.

From SCOUT to the database

Start the drag operation at the desired row of the SCOUT material list and move the cursor with pressed left mouse button to the record table of the database window. Release the mouse button (i.e. drop the material). If the database already contains a record with the name of your material you will get an error message. In this case, rename your material and try to move it to the database again.

If the material has been added to the database you are asked for your name (you will be the author of this database record) and a comment that describes the material. If you have specified a user in the [Options](#) submenu SCOUT will use this name and fill it into the appropriate database record automatically.

4.3 Layer stacks

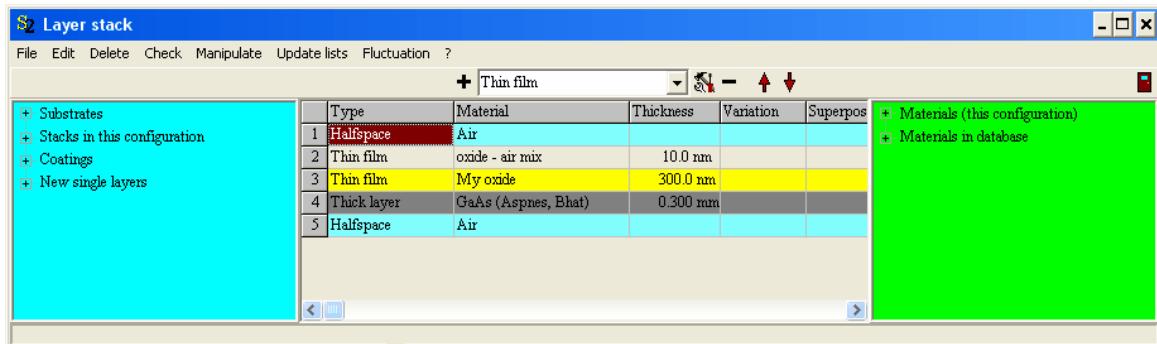
4.3.1 Introduction

The list of layer stacks is opened by a right-click of the corresponding treeview item (**Layer stacks**). In most cases you will investigate one layer stack only, but SCOUT lets you work with an unlimited number of stacks if you want to. In the list of layer stacks you can create, edit and delete layer stacks. To change a layer stack, select in the list of layer stacks and open its window by the **Edit** command or a right click with the mouse.

You can open the first layer stack in the list of layer stacks from the main window of SCOUT using the command **Actions|Open first layer stack**. This is the shortest way to the layer stack definition. If a layer stack is displayed in a [View](#) you can open the layer stack definition window by holding the

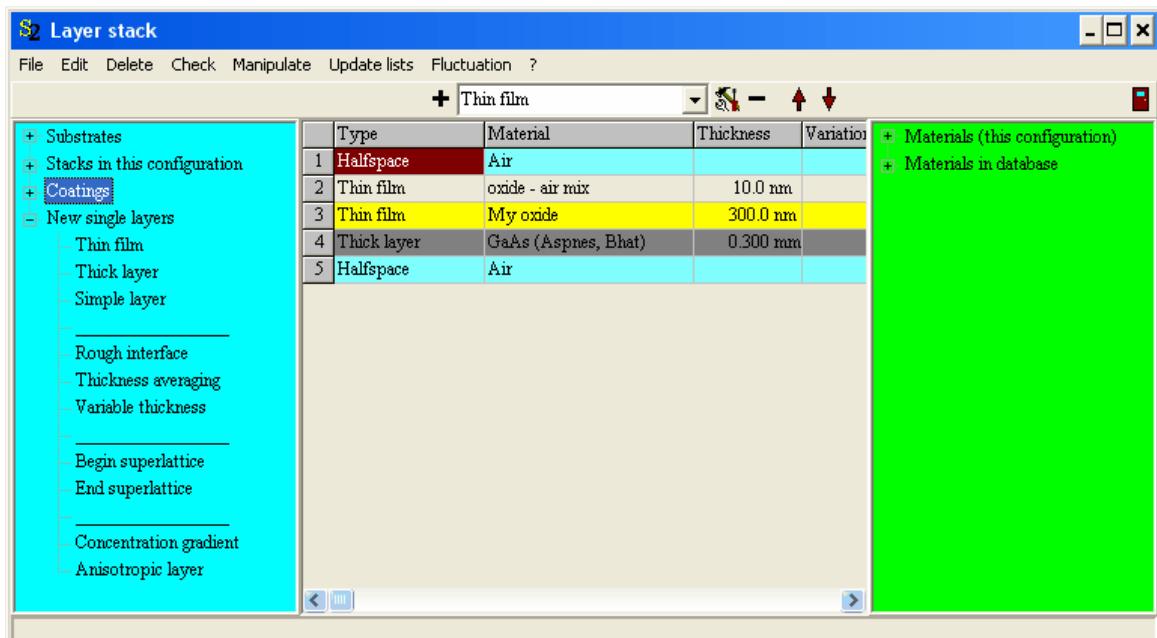
Ctrl key down while you click on the [hotspot of the view element](#).

Layer stacks in SCOUT are defined in a [list](#) beginning with the top halfspace of the stack and ending with the bottom halfspace. In between an almost arbitrary sequence of thick layers or thin films can be defined. The light beam is incident on the layer stack from the top halfspace. Hence in most cases the top halfspace should be vacuum (or air) as is the case in almost all experiments (except [ATR](#), see below). An example of a simple layer stack definition is this:



The current layer stack definition is shown in the center of the window. On the left side you have access to [pre-defined layer stacks](#), on the right side the available materials to be used in the individual layers are given. The assignment of materials is discussed [below](#).

In order to add new layers you can select the type of the new layer in the dropdown box given in the top of the window (where 'Thin film' is selected). Pressing the '+' button will insert a new layer of the indicated type above the currently selected layer. Use the '-' button to delete the selected layer. An alternative way to add new layers to the stack is the following: Open in the treeview on the left side the node 'New single layers' by pressing the '+'-sign to the left of it. You will get a list of all available new layer types:



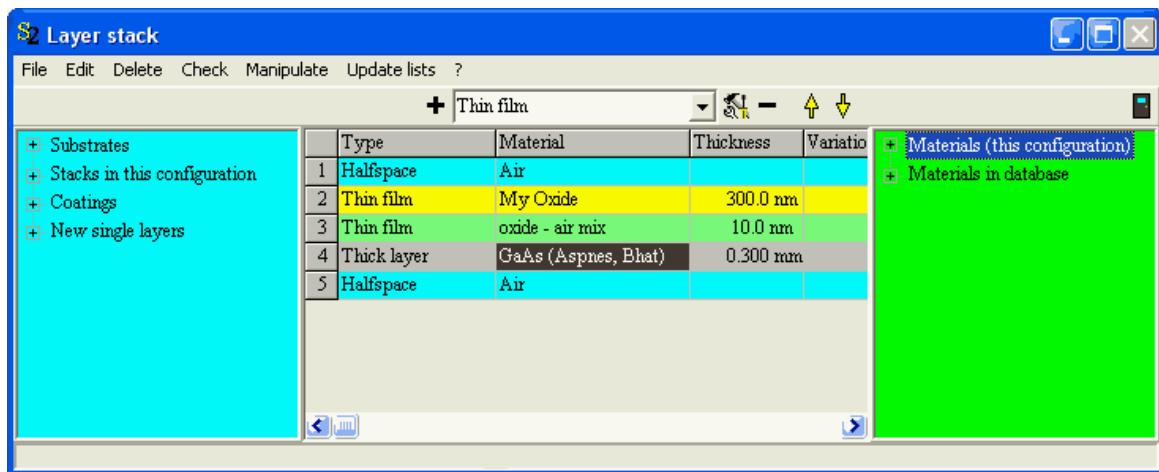
Just drag&drop the layer type to the wanted position in the stack and a new layer of this type is created.

The individual layer types that may occur in a layer stack are discussed below:

- [Halfspace](#)
- [Simple layer](#)
- [Thin film](#)
- [Thick layer](#)
- [Variable thickness](#)
- [Rough interface](#)
- [Begin superlattice](#)
- [End superlattice](#)
- [Concentration gradient](#)
- [Anisotropic layer](#)
- [Included layer stack](#)

Useful commands

Use the command **Manipulate|Flip part of layer stack** to reverse the order of layers. You are asked to type in the top and the bottom layer of the section to be reversed. Type in the number of the layers that is assigned on the left side, e.g. 2 for the top layer and 3 for the bottom layer if you want to switch 'oxide-air mix' and 'My oxide' in the example given above. The situation afterwards is this:



With **File|Report** you can write the current layer stack definition to the workbook:

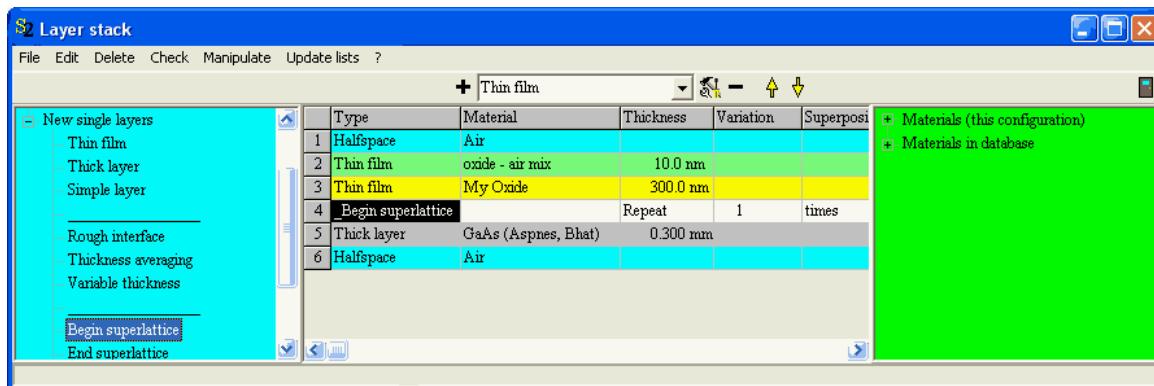
	A	B	C	D	E	F
1	Layer stack				06.11.2004 16:02:12	
2						
3		Type	Material	Thickness	Variation	Superposition
4	1	Halfspace	Air			
5	2	Thin film	oxide - air mix	10.0 nm		
6	3	Thin film	My Oxide	300.0 nm		
7	4	Thick layer	GaAs (Aspnes, Bhat)	0.300 mm		
8	5	Halfspace	Air			
9						

In the workbook you can **Copy** text fractions and paste them into your word processor.

Check

Use this command to check the validity of the current layer stack definition. In a situation like the

following



you will get the error message



which reminds you that something is wrong with the 'Begin superlattice' layer (The 'End superlattice' is missing).

4.3.2 Layer types

4.3.2.1 Halfspace

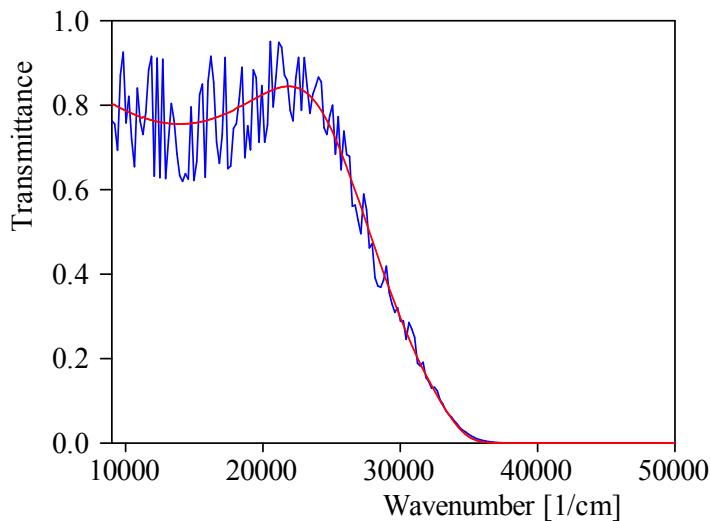
As mentioned above the stack of layers is enclosed in between two semi-infinite 'halfspaces'. Every correctly defined layer stack must have exactly two halfspace layers, namely the top and the bottom halfspace. The only thing to be done with halfspace layers is to [assign a material](#) to it.

4.3.2.2 Simple layer

Simple layers are used in most cases to represent a thick or thin layer of a material in the stack. Simple layers have a thickness (always specified in **microns**) and you must [assign a material](#) to it. In addition, you can choose between two modes of handling the contributions from multiply reflected partial waves, as discussed below.

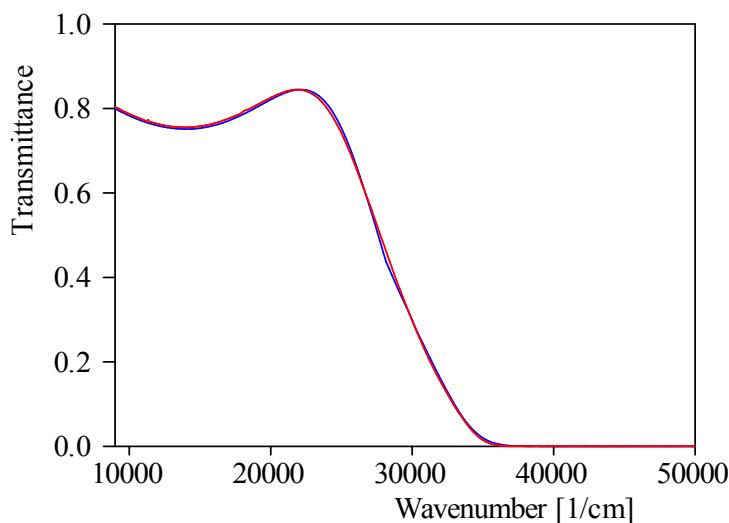
Some considerations about wave propagation through layer stacks are discussed in [\[Harbecke 1986A\]](#), in particular the superposition of partial waves. In the case of thin layers, the partial waves reflected at the top interface and those that travel through the layer and are reflected back at the bottom interface must be superimposed to the total reflected wave by summing up their amplitudes, taking into account the phase. This may lead to constructive or destructive interference, depending on the wavelength of light. This kind of superposition is called 'coherent'. The interference fringes observed in reflection or transmission spectra get very narrow if the thickness of the layer becomes larger and larger.

Eventually the narrow structures are not resolved experimentally due to the limited resolution of spectrometers, or due to thickness inhomogeneities over the investigated sample spot which may lead to a cancellation of the interference structures. In these cases the measured spectra are quite smooth as the red one shown below, whereas the simulated blue spectrum shows interference fringes. As can be seen the sharp structures are not resolved completely in the simulation, either.



Measured transmission spectrum (red) of an ITO layer (90 nm) on top of a glass substrate (990 micron thickness). The blue curve shows a computed spectrum with coherent superposition of partial waves for the glass layer.

Instead of doing the tedious task to compute a lot of spectra for various thicknesses and take the average to remove the interference fringes in the simulation one can follow a shortcut and drop the phase terms in the superposition of the partial waves, as suggested in [\[Harbecke 1986A\]](#). This is called the incoherent superposition which leads to multiple reflection, but no multiple interference effects. As shown below the agreement between computed and measured spectra can be excellent if the layer thickness is really much larger than the wavelength of light:



Same as above, but now with incoherent superposition of partial waves for the glass layer.

To switch between coherent and incoherent superposition mode for a layer, select the cell labeled 'coherent' or 'incoherent' in the layer stack list and press the function key F4 or F5. This will change from coherent to incoherent treatment or the other way round.

Warning: if you choose incoherent superposition for thin layers your computed spectrum will be wrong!

4.3.2.3 Thin film

Layers of this type are very similar to [simple layers](#). Their thickness is always specified in **nanometers**, and the treatment of partial waves is fixed to **coherent**.

4.3.2.4 Thick layer

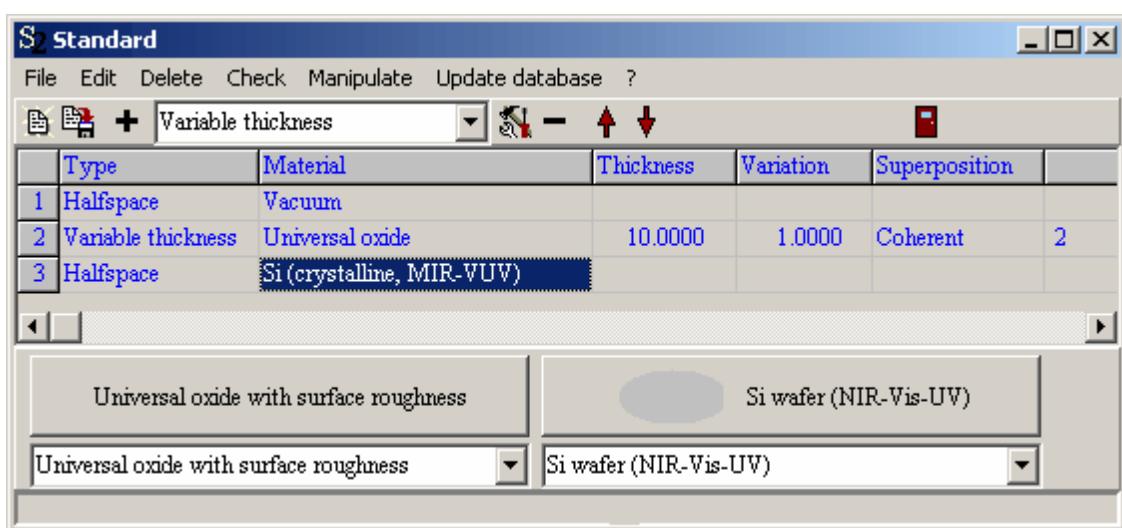
Layers of this type are very similar to [simple layers](#). Their thickness is always specified in **millimeters**, and the treatment of partial waves is fixed to **incoherent**.

4.3.2.5 Variable thickness

This layer type was introduced for cases where the thickness of one layer in the stack is not homogeneous over the investigated sample spot, but varies due to some lateral gradient.

The algorithm used to compute the spectra is described in [[Theiss 1997A](#)]. It involves the Fourier transform of the layer thickness distribution function.

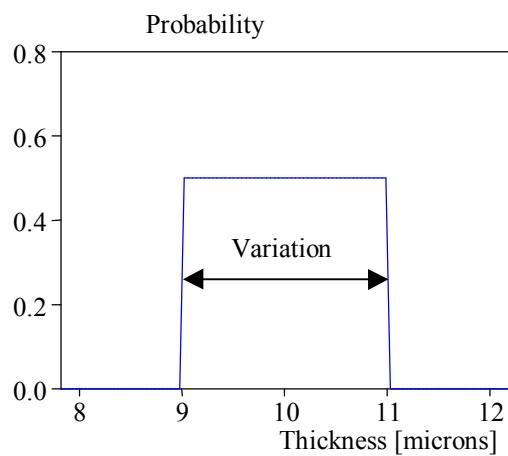
In the layer stack list objects of this type appear as follows:



Three parameters are to be set to define a layer of variable thickness. The average thickness, the thickness 'variation' and a third one which is an integer specifying the type of thickness distribution function. Whereas the first two parameters (thickness and variation specified in **microns**) appear as adjustable fit parameters, the latter is just a user-defined constant. In the dialog shown above type 2 has been chosen.

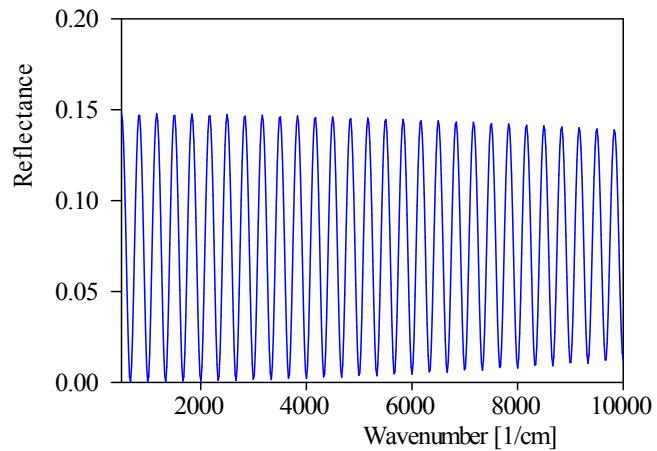
The following distribution functions are defined at present:

Distribution function type 1:

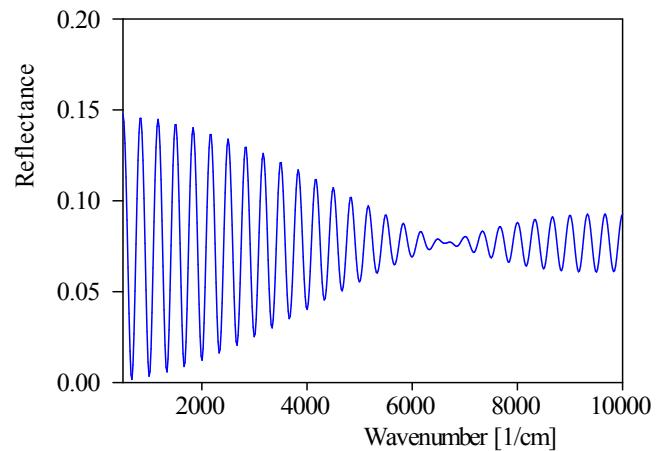


For a single 10 μm thick layer of a material with refractive index 1.5 one gets the following reflectance spectra:

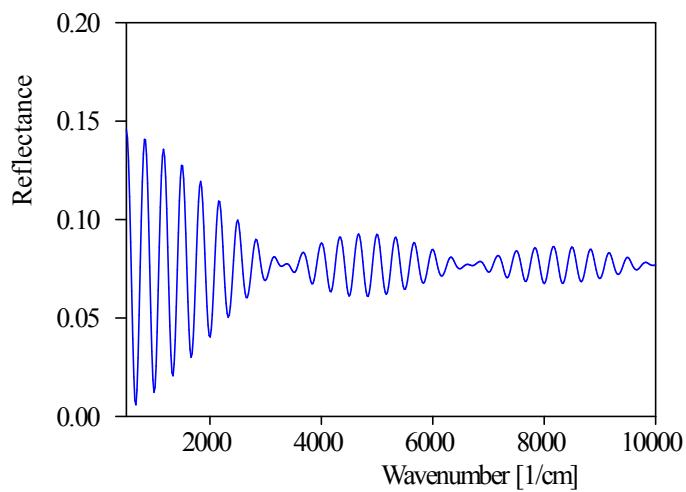
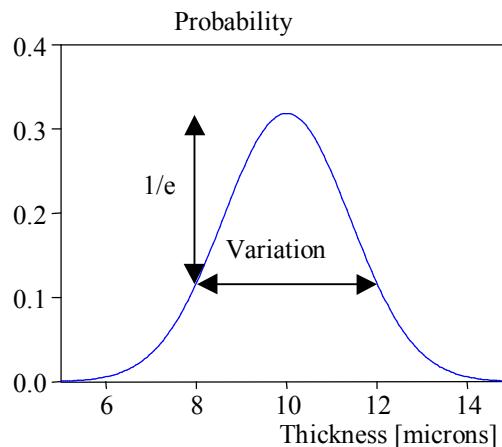
Variation 0.1 μm :



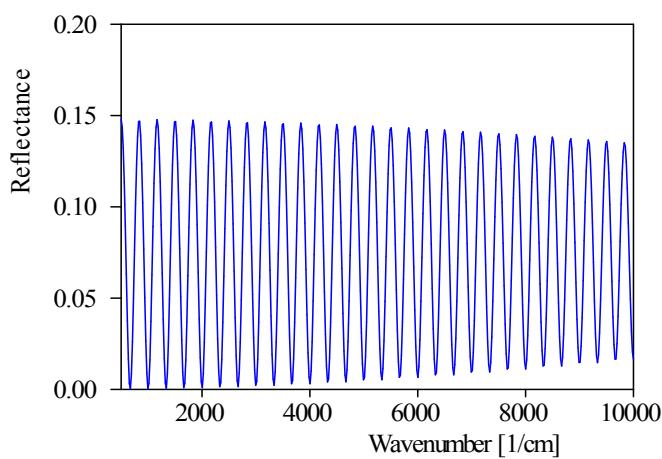
Variation 0.5 μm :



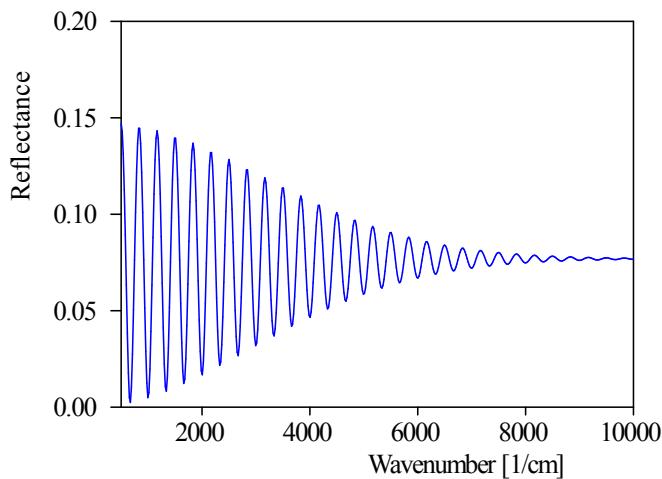
Variation 1.0 μm :

**Distribution function type 2:**

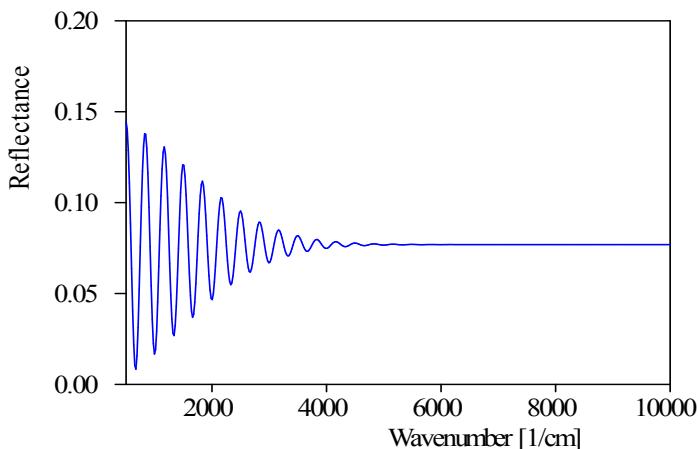
Variation 0.1 μm :



Variation 0.5 μm :



Variation 1.0 μm :



What can be observed here is the transition from the coherent superposition of the partially reflected waves (from the front and end interface of the layer) at low wavenumbers to the incoherent superposition (no interference fringes) at high wavenumbers.

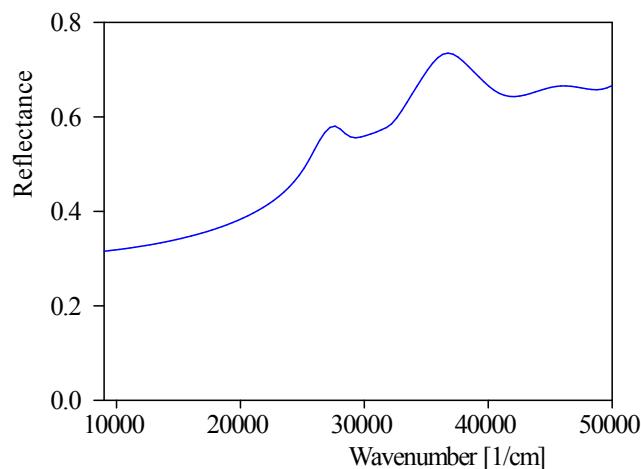
Warning: Layers of this type should not be combined with layers of type thick layer or simple layers with incoherent superposition!

4.3.2.6 Thickness averaging

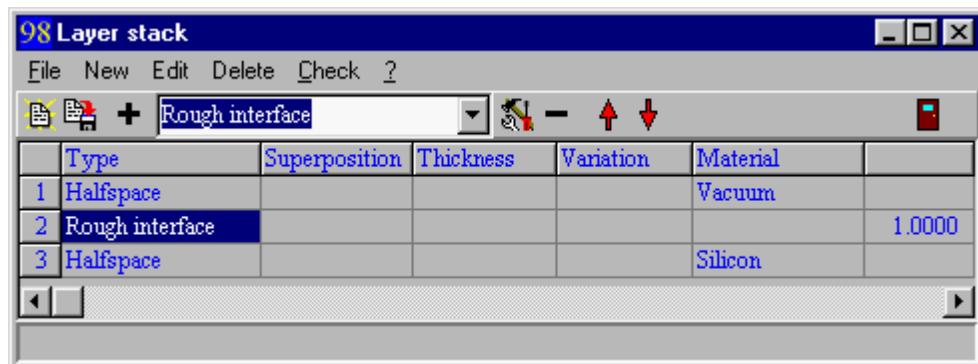
This type is almost identical to the layer type [Variable thickness](#). The only difference: Thicknesses are specified in nm instead of microns.

4.3.2.7 Rough interface

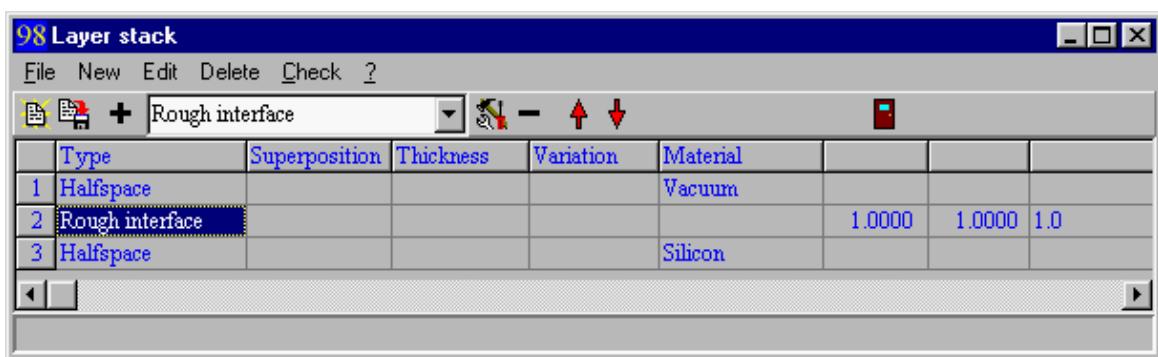
Since many people asked for a convenient description of scattering losses due to interface roughness we have developed a special layer type which may help in many cases. To demonstrate the application of this feature we consider a simple interface between vacuum and silicon in the Vis/UV region. The reflectance (10° angle of incidence) is the following:



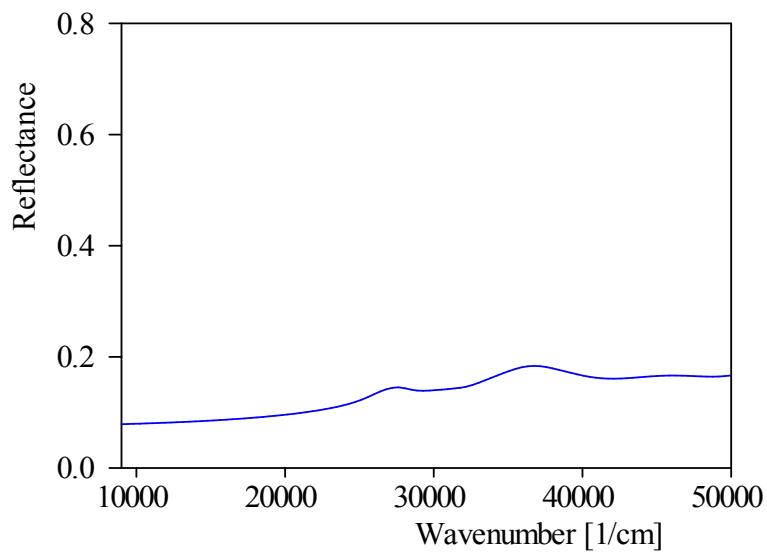
To introduce roughness losses a layer of the new type 'Rough interface' is inserted between the vacuum and the silicon halfspace:



To get access to all the layer parameters you have to expand the window a little:

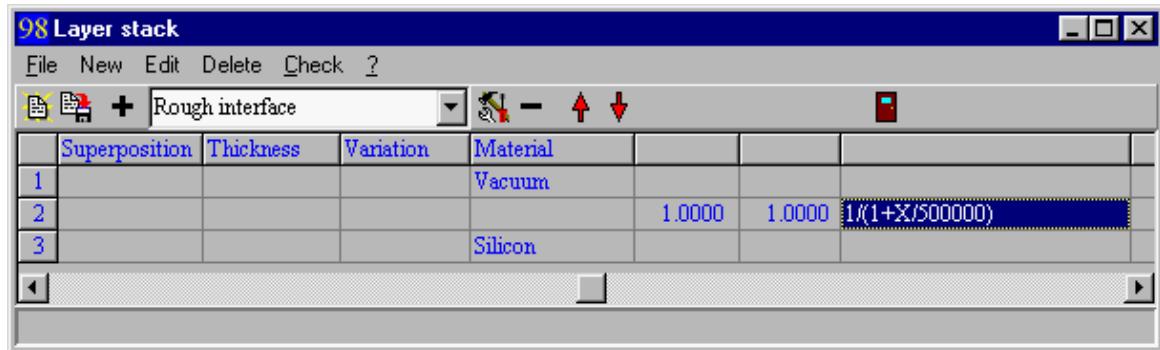


The most important parameter is given in the 8th column ('1.0' is the default). Here you can define a function which determines a frequency-dependent correction factor which multiplies both the amplitude reflection and transmission coefficient of the interface. Click on the '1.0' and type a 0.5 instead. Now you have (after a re-calculation) the following spectrum:

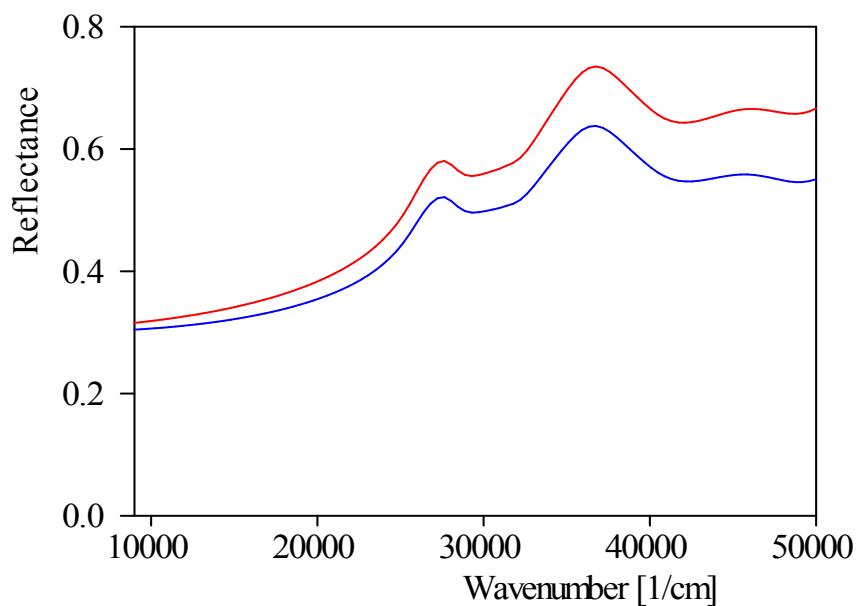


The amplitude reflection coefficients are multiplied by 0.5 which means that the intensity is now one fourth only.

Of course, you may want to define a frequency dependent correction since scattering losses typically are more pronounced for small wavelength than for large ones. You can do so. The wavenumber can be introduced into the formula as the variable 'X' as shown below:



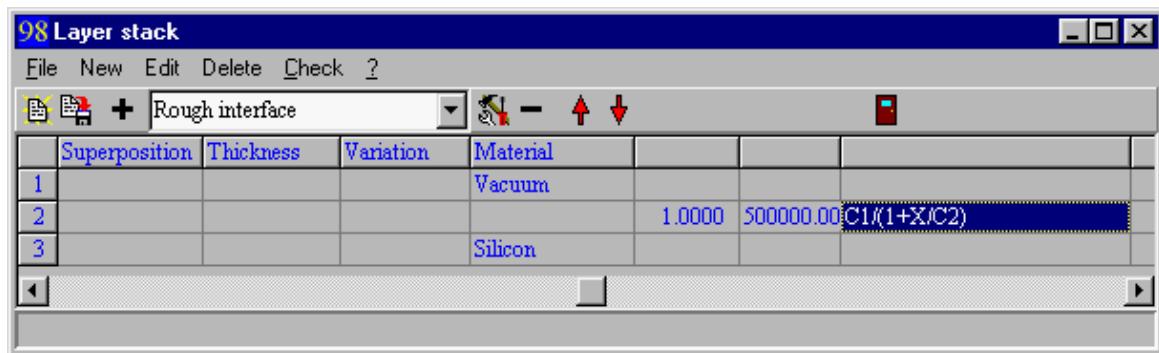
Now for low frequencies the correction factor is close to one and for high frequencies it gets larger:



Here the blue curve is the new spectrum including scattering losses whereas the red one shows the original spectrum without interface roughness for comparison.

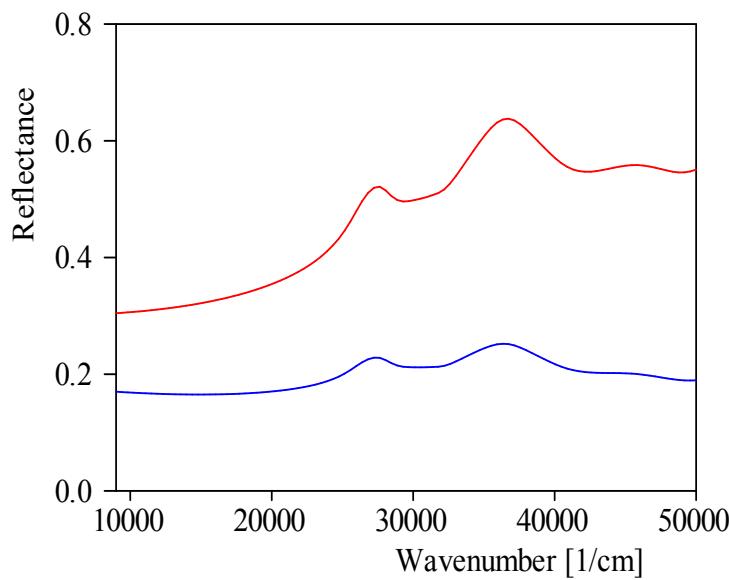
As you can see you can be quite creative and implement all kinds of scattering loss corrections specifying the corresponding 'loss function'.

In order to increase your flexibility even more the 6th and 7th column of an interface roughness layer specify two possible fit parameters that you can use in the formula. As shown in the following example you can refer to the two parameters as C1 and C2, respectively:

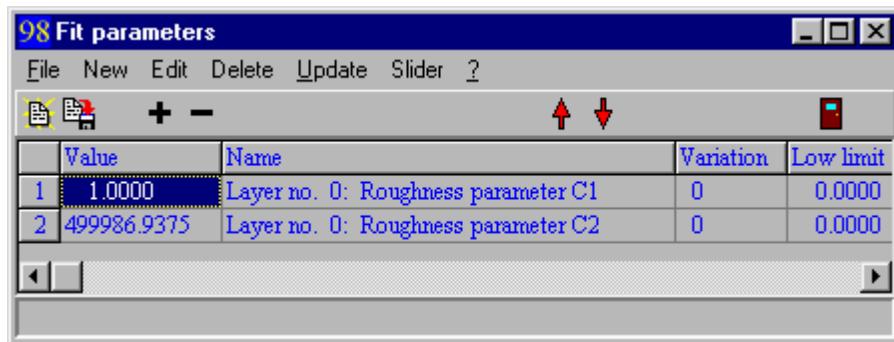


Of course, the given formula gives the same correction factor as before, but now you could fit the C1 and C2 values to experimental data.

Here is an example: setting C1=0.8 and C2=100000 and loading the previous spectrum (that was created using the explicit formula $1/(1+X/500000)$) as a 'measured spectrum' the following situation is created:



Declare the two roughness parameters as fit parameters and start the downhill method. After a short while the original values are recovered by the fit:



We hope that this explanation of the interface roughness layer type is clear enough and that two adjustable parameters in a scattering loss model turn out to be sufficient.

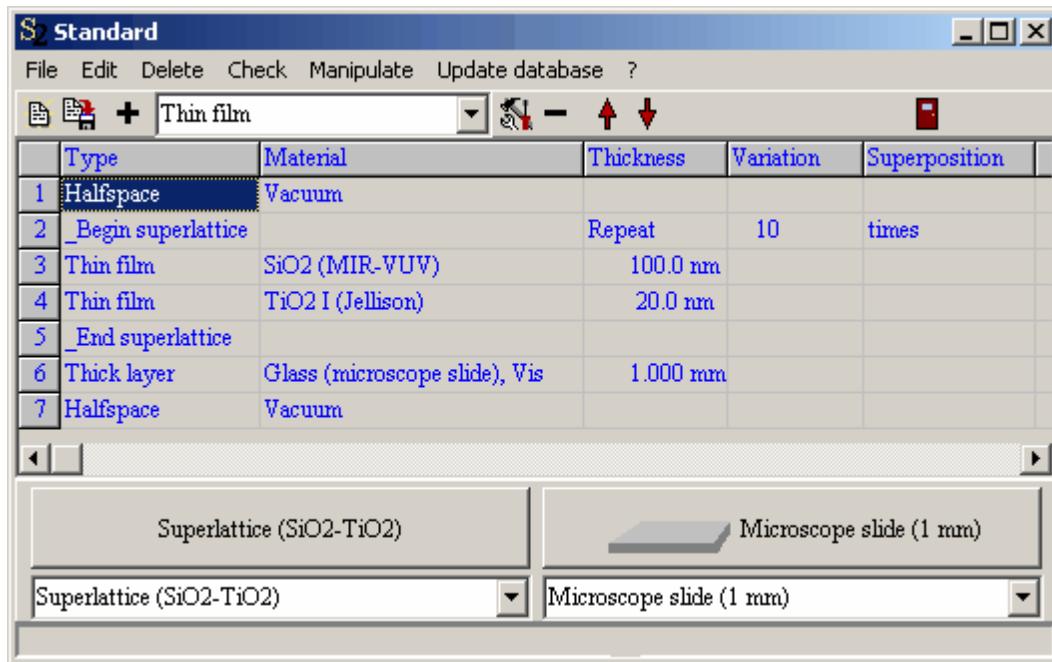
Current limitations

- **interface roughness layers cannot be used in connection with anisotropic layers.**

A 'real world' application example of a roughness correction is discussed in [[Valk 1998A](#)].

4.3.2.8 Superlattice

You can repeat parts of layer stacks by enclosing the layers between 'Begin superlattice' and 'End superlattice' lines. These lines are no real layers but indicate that the enclosed layers are repeated as often as stated in the 'Begin superlattice' line. Here is an example:



4.3.2.9 Concentration gradient

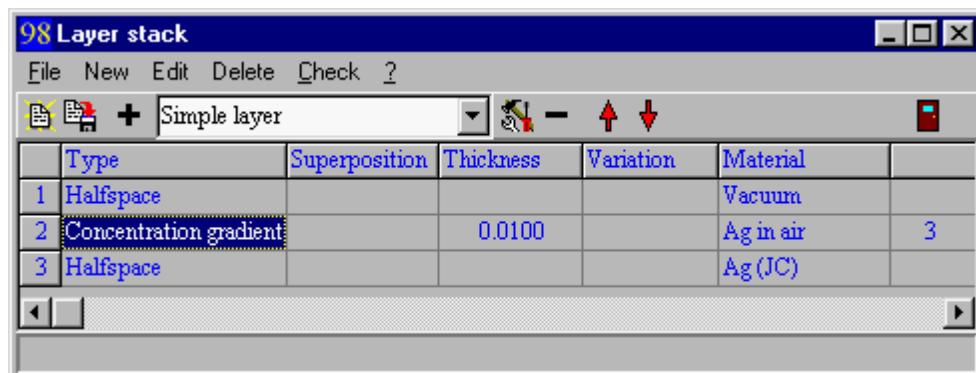
This type of layer has been introduced for the description of gradually changing composition in a part of the layer stack. A 'concentration gradient layer' must use an effective dielectric function for which the first fit parameter is the volume fraction. It is recommended that you use one of those effective medium theories that have as their only parameter the volume fraction, such as the Maxwell Garnett, Bruggeman or Looyenga formulas.

Like simple layers concentration gradient layers have a thickness and a dielectric function (once again: this must be an effective dielectric function). However, the layer will be divided into a set of sublayers (whose number can be specified by the user). All sublayers have the same thickness (the total layer thickness divided by the number of sublayers). Each sublayer can have its own value of the volume fraction which is defined by a user-defined formula.

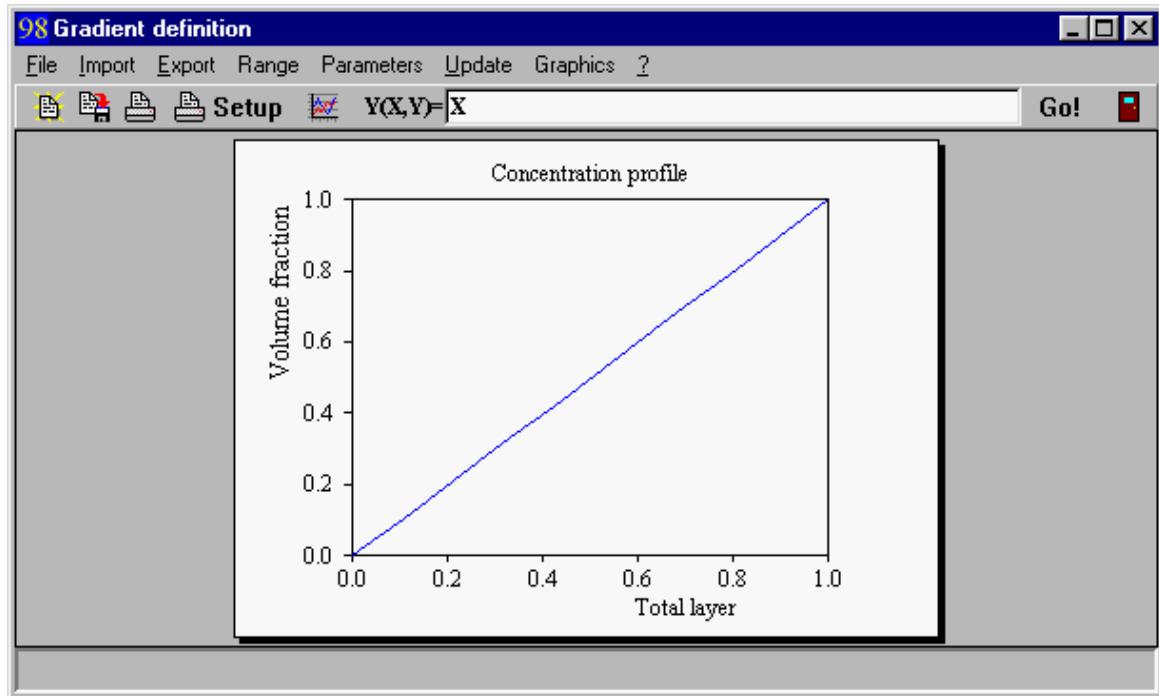
Simple example

The application of concentration gradient layers is demonstrated using the following example. We consider a reflectance spectrum in the visible spectra range. The layer stack is a concentration gradient layer on top of a silver halfspace. The light enters from vacuum. The concentration gradient uses a Bruggeman effective medium with Ag particles embedded in vacuum. The dielectric function of silver is taken from the dielectric function database (Ag(JC)).

The layer stack is this:

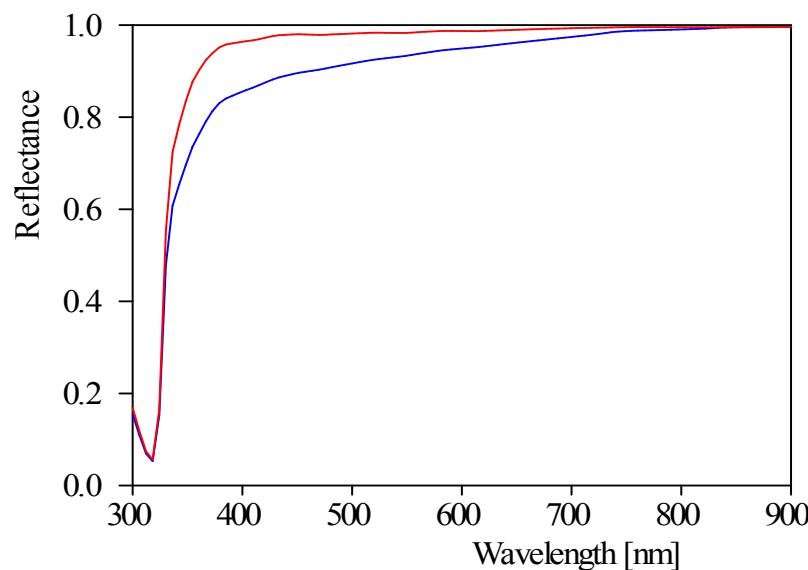


Note the number three in the 7th column of the concentration gradient layer: This is the number of sublayers to be used for dividing the total thickness which is 10 nm in this case. With a right mouse click or the menu command **Edit** you can open a window which shows the volume fraction profile:



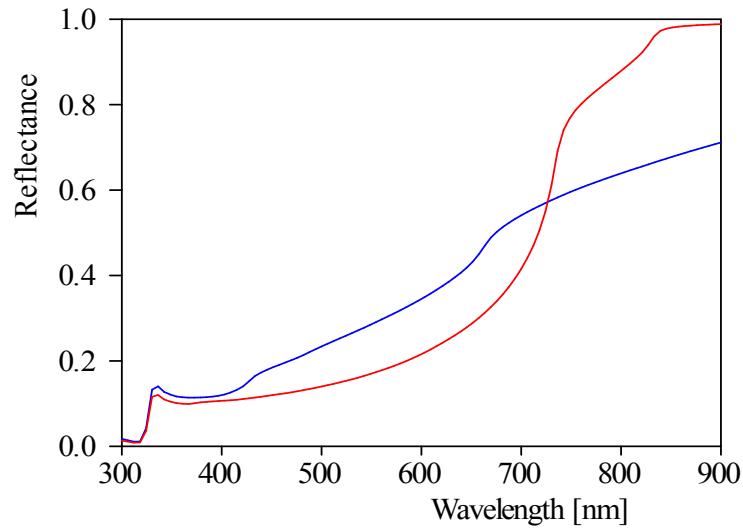
In the example a linear profile is defined with the simple formula 'X'. At the top of the layer (the left side of the graph) the silver volume fraction is 0 and at the bottom solid silver (volume fraction 1.0) is reached.

The reflectance spectrum (normal incidence of light, blue curve) is not very different from that of a sharp vacuum-air interface (red curve):

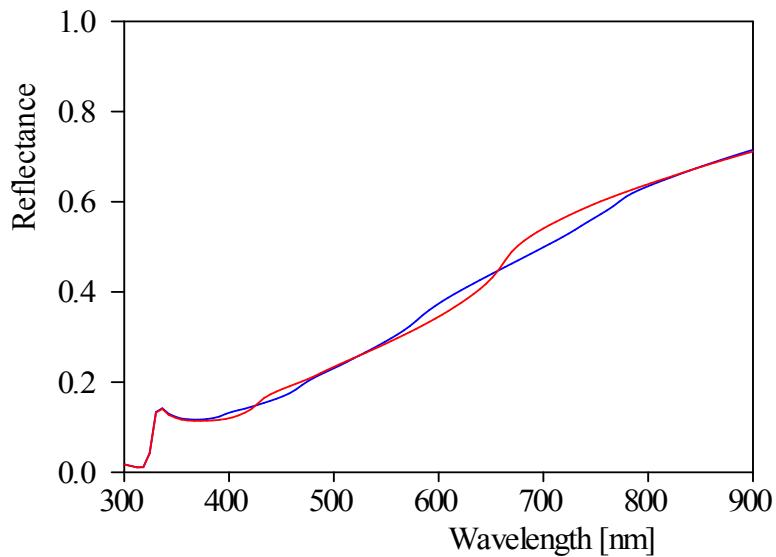


Note that you should sample the total thickness into sublayers whose optical thickness is much less than the wavelength of light. Otherwise you will see features in the spectra caused by the steps in the profile. If we increase the total thickness to 100 nm 3 sublayers are not sufficient. The following picture compares the computed spectrum with 3 sublayers (red curve) to that with 10

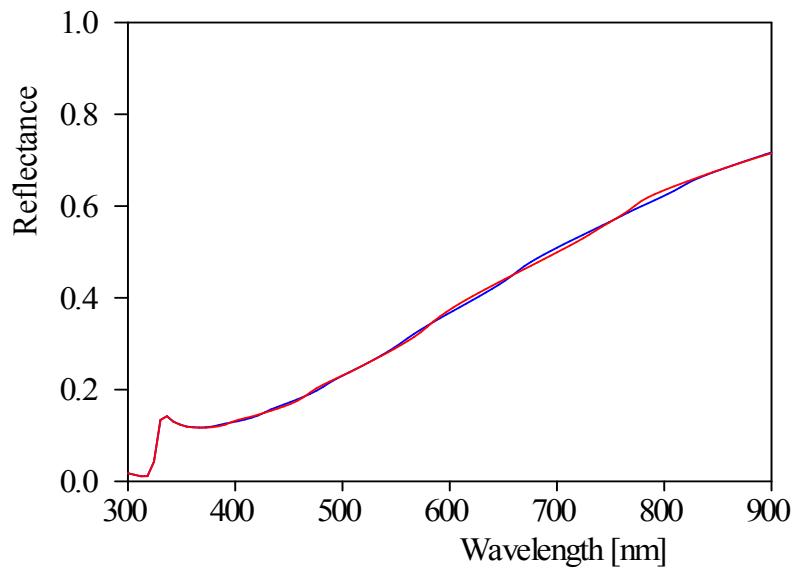
(blue):



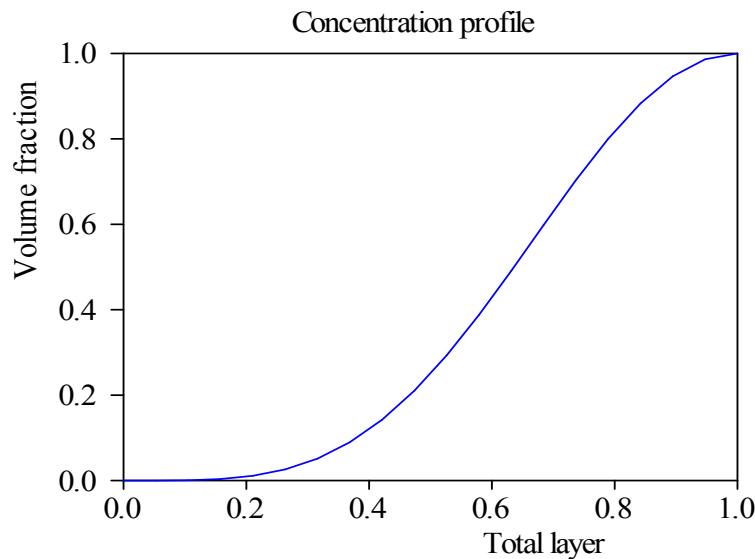
The difference between 10 (in red) and 20 sublayers (blue) is not very large:



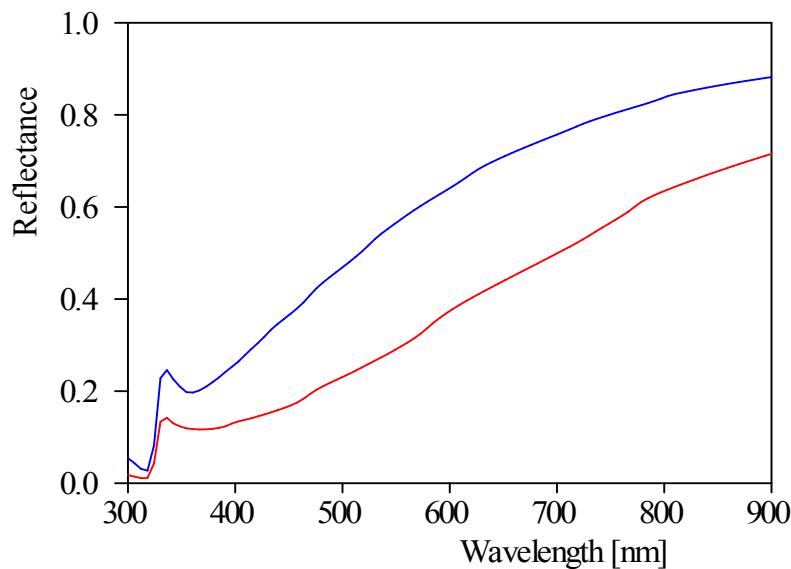
Hence 20 sublayers seem to be sufficient for this case. Going from 20 (red) to 30 (blue) doesn't change the spectrum any more:



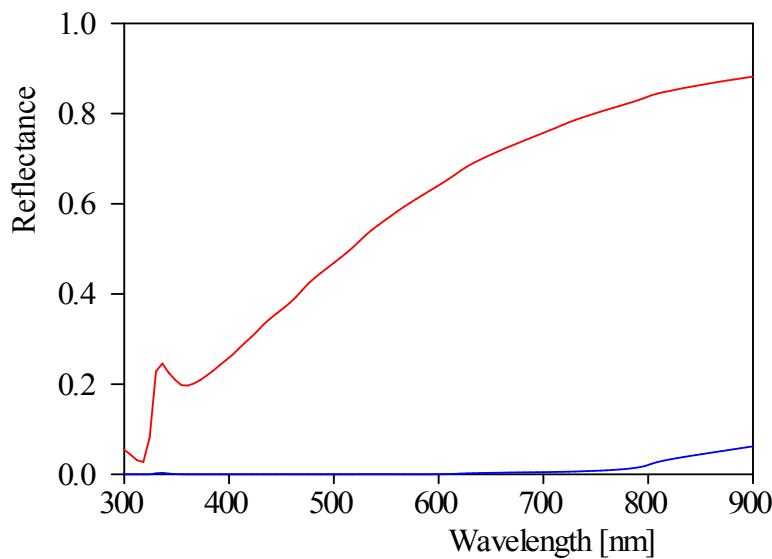
Of course, you can define more advanced profiles with the flexible choices of user-defined [formulas](#). A smoother profile than the linear one is defined by the formula $\text{SQR}(0.5 - 0.5 * \text{COS}(\text{PI} * X))$, for example:



This leads to the following spectrum (red: linear profile, blue: smooth profile):



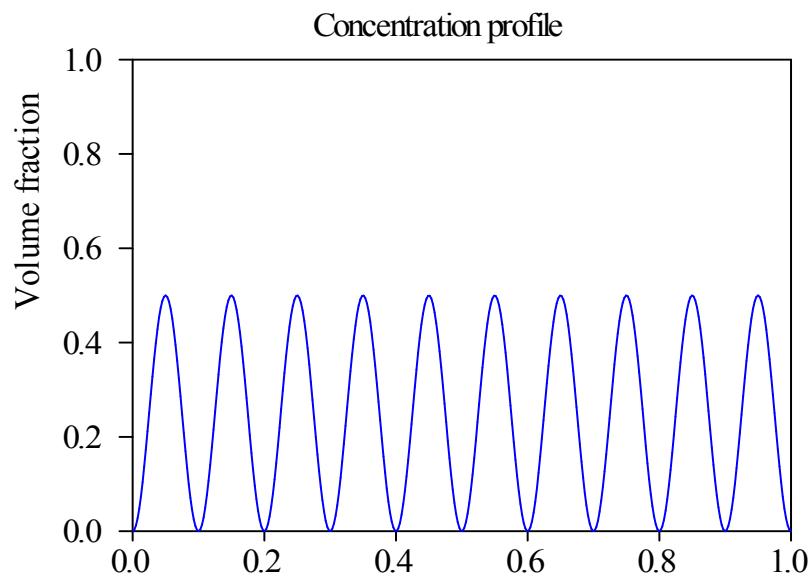
This example is closed with an interesting spectrum of the smooth profile for a thickness of 500 nm (blue curve) which leads to a very low reflectivity (for comparison the 100 nm spectrum is repeated here in red):



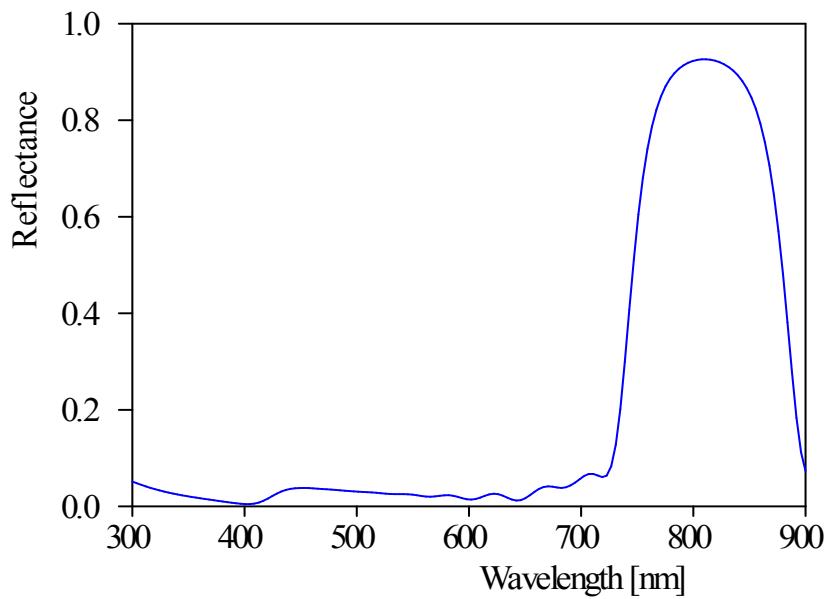
Noble metal/air-composites with low volume fractions are called 'blacks' for obvious reasons!

Advanced example

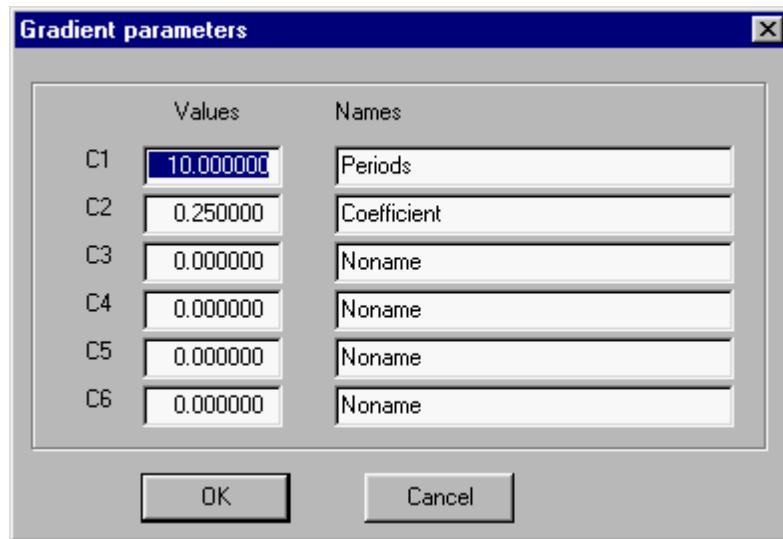
Now we show some more features of concentration gradient layers using an example of silicon oxide layers of various oxygen concentration on a 1 mm thick glass plate. The example makes use of two database entries, namely those of SiO and SiO₂. A concentration gradient layer on the glass substrate is used with a mixture of SiO particles embedded in an SiO₂ matrix. The SiO volume fraction is modeled by the formula $C_2 - C_2 \cdot \cos(2\pi C_1 \cdot X)$ which produces profiles like the following (for $C_1=10$ and $C_2=0.25$):



The periodical repetition leads to constructive superposition of partially reflected waves producing pronounced interference maxima:



You can use the six constants C₁, ..., C₆ in the formula. Together with the total layer thickness these constants appear as fit parameters. You can set these constants and their names in the dialog that you open with the **Parameters** command in the **Gradient definition** window:



Hence you can select each of the constants as fit parameters and use sliders to learn how profile changes influence your optical spectra. If you have a fast PC and a large screen SCOUT can be a very powerful tool for intuitive thin film design!

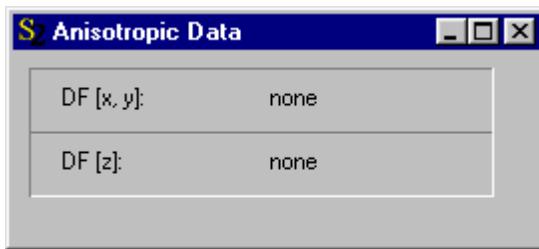
4.3.2.10 Anisotropic layer

Optically anisotropic layers may occur in the layer stack. They are handled using the wave propagation method described in [\[Parikh 1992A\]](#). However, compared to simple layers, there are some severe **limitations** due to the more complex algorithm to be used in this case.

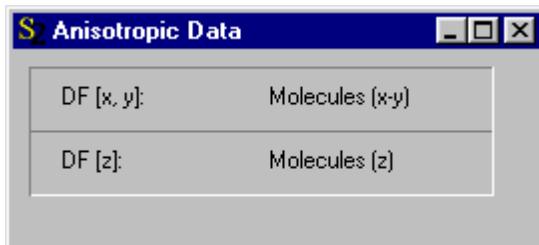
First of all, up to now SCOUT supports a special case of anisotropy only, namely uniaxial layers where the dielectric function in the direction of the layer stack normal (called the z-direction in this section) may be different from the dielectric function in the plane of the layer stacks, called the x-y-plane. You can use as many of such layers in the stack as you like.

Another limitation in combination with anisotropic layers is the following: You can have only one thick layer in the stack for which you want to switch off the narrow interference patterns (e.g. a thick, but transparent substrate). As discussed above for [simple layers](#), this can be done setting the incoherent option for the superposition of partial waves. If you use incoherent superposition for one of the layers in the stack, and there are anisotropic layers defined, SCOUT will do the following: Lacking an elegant and fast way of averaging out the narrow interference features SCOUT does it the tedious and stupid way, namely by computing several spectra (at present: 20) with slightly varying thickness of the thick layer. The average of these spectra is then taken as the result of the computation and displayed on the screen. Obviously, the computational times will be large since many spectra have to be computed, and it may occur that the averaging is not complete and there remain some 'interference artefacts' in the spectrum. To identify these artefacts it is recommended to repeat the computation for a slightly different thickness of the thick layer and compare the results.

Uniaxial anisotropic layers require the definition of two different dielectric functions for the z-direction and the x-y-plane, respectively. For this reason, there is no entry in the 'material' column of the layer stack definition. Instead, the text 'See parameters' reminds you that you have to do the assignment differently here. Use the **Edit** command (or a click with the right mouse button) to open the following dialog which is used to do the dielectric function assignment:



To replace the 'none' assignments you have to drag the wanted materials from the list of dielectric functions to the 'none' text and drop them there. Afterwards the dialog may look like this:

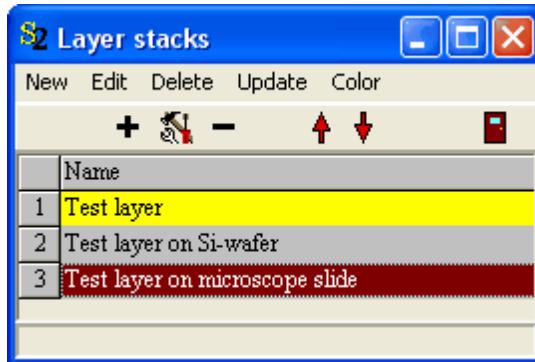


This example is taken from the second SCOUT tutorial where you can learn how to determine molecular orientation angles from infrared spectra using anisotropic layers.

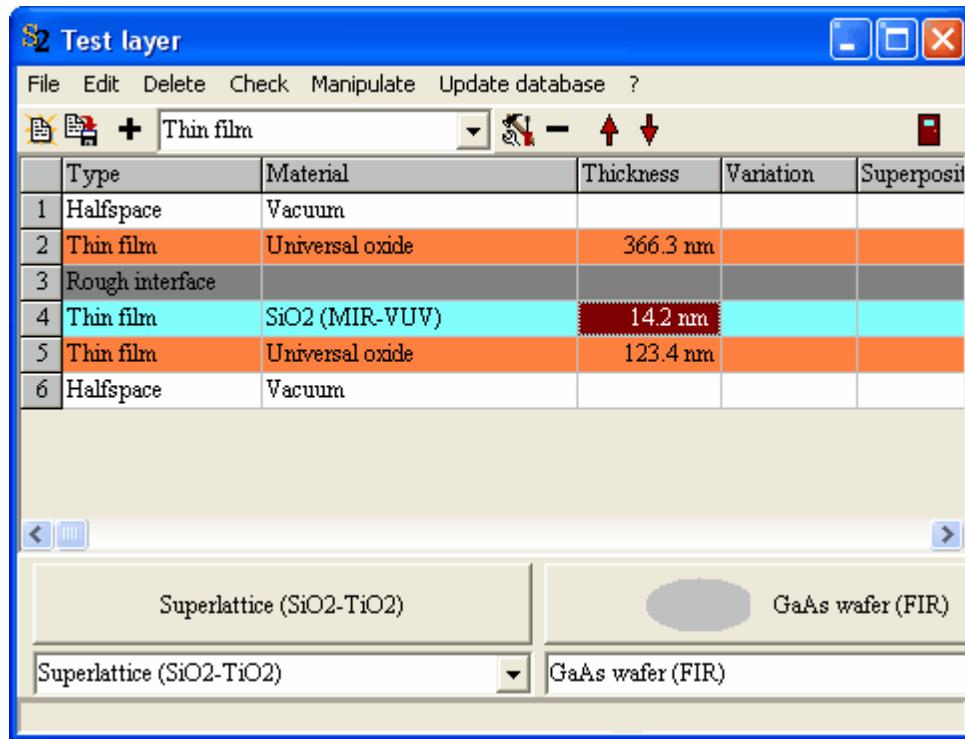
4.3.2.11 Included layer stack

You can include in a layer stack definition another layer stack. This is useful if you investigate a thin film coating deposited on different substrates, for example. In this case, you should define the coating in a layer stack definition of its own and include it as a block into the layer stack definitions for the individual substrates.

Here is an example: Suppose you want to analyze spectra of a thin film system deposited on a microscope slide and a silicon wafer. You have to create three layer stacks in the layer stack list:

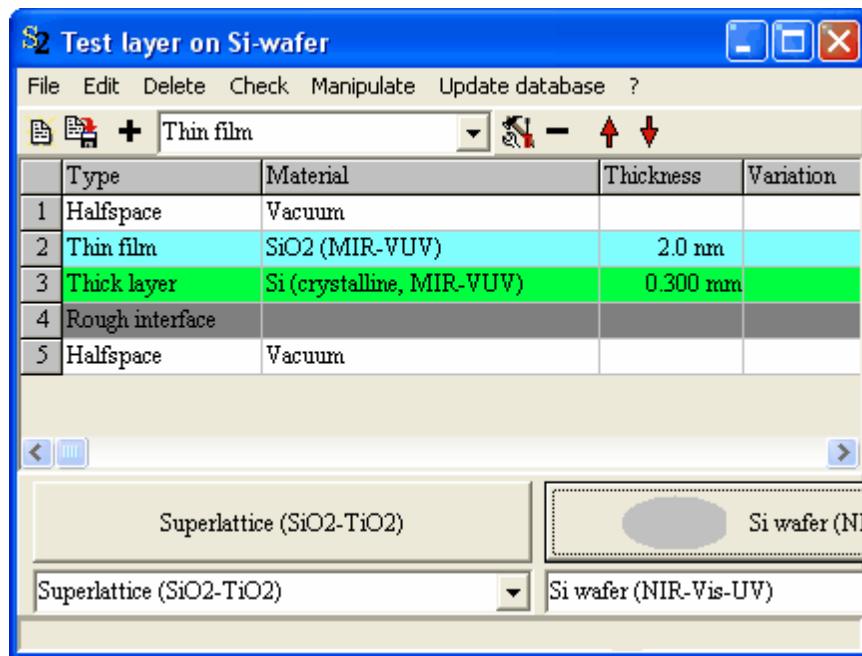


Open the object 'Test layer' and define the thin film system between two vacuum halfspaces. Do not use any substrate here:



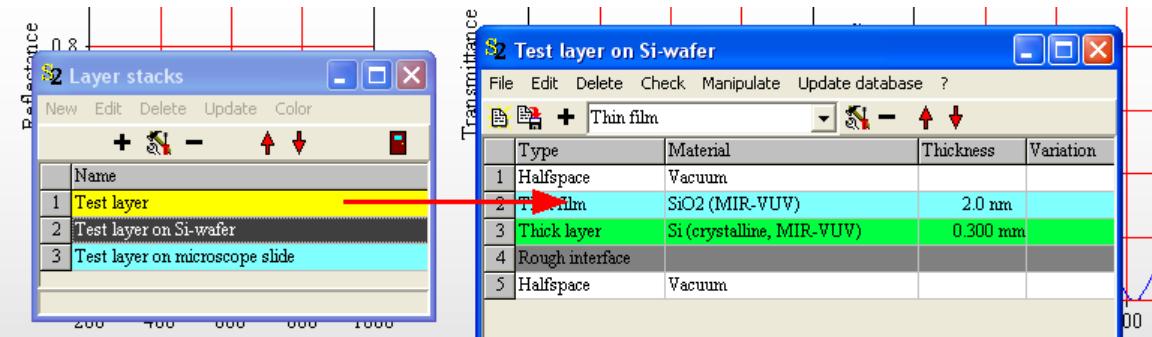
You can use any kind of layers with the exception that **the last layer above the bottom halfspace must not be a rough interface**. As mentioned above, this layer stack will be included into the others (see below) without the top and bottom halfspaces - hence it does not matter which materials you assign to the halfspaces in this list.

Now open the layer stack 'Test layer on Si-wafer' and define the Si-wafer substrate (it's a single side polished wafer with a rough backside in this example) as follows:



The test layer is now inserted as an included layer stack by drag&drop. Keep the layer stack 'Test layer on Si-wafer' open and open the list of layer stacks as well. Take the object 'Test layer' in the layer stack list with the mouse and drag it above layer 2 in the Si-wafer layer stack (i.e. 2 nm SiO2).

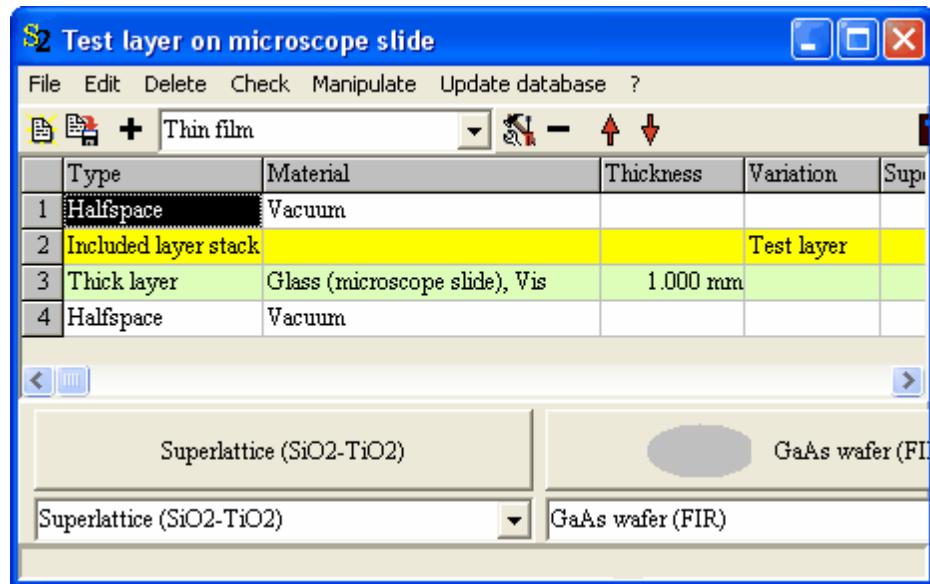
layer) as shown below:



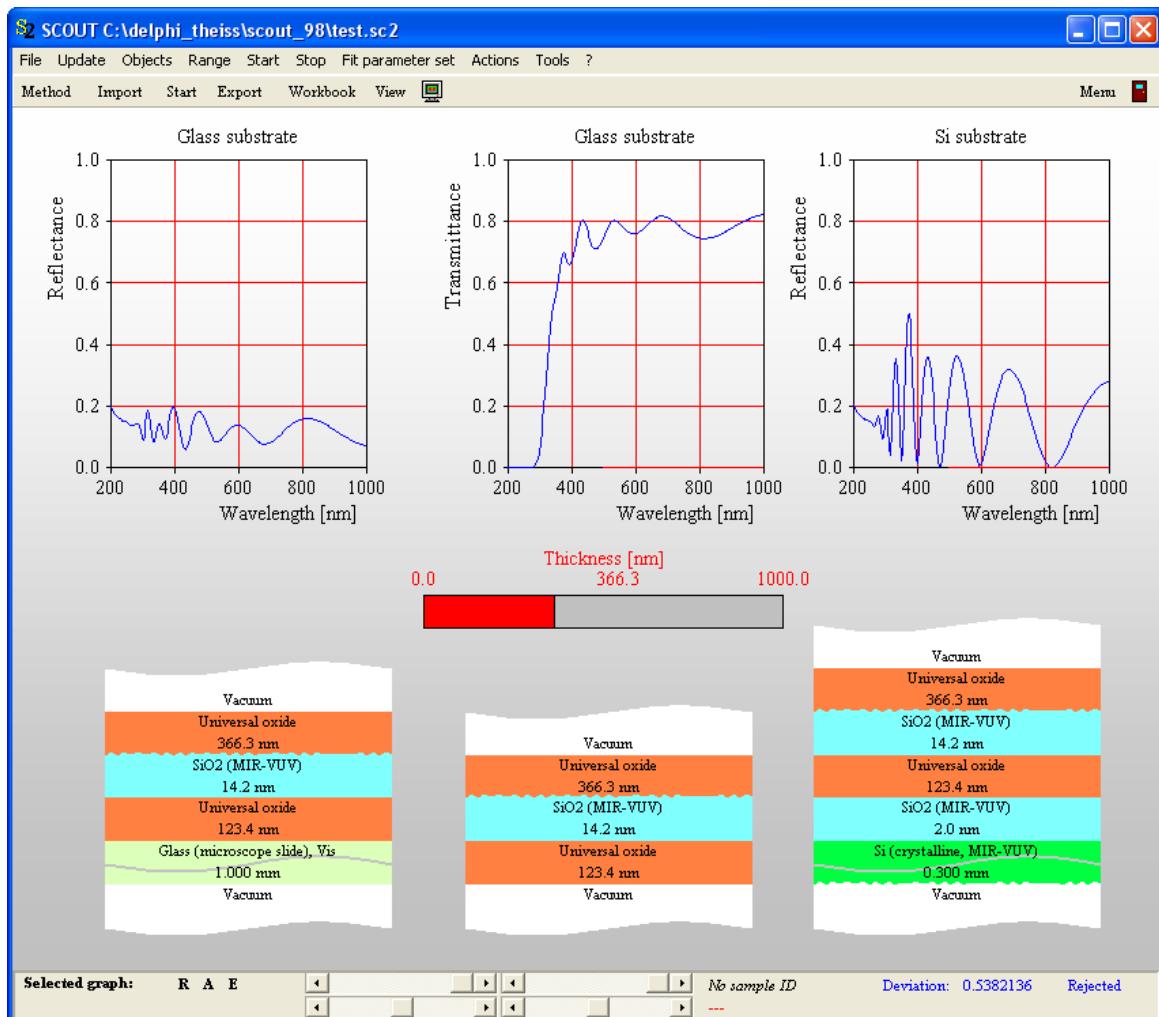
Drop it there and watch the update of the layer stack definition:



Do the same for the layer stack with the microscope slide substrate:



You can now use the layer stacks with the two substrates to compute optical spectra. The following example shows the computation of R and T for the microscope slide substrate and R for the Si-wafer substrate. If you change the thickness of one of the layers in the 'Test layer' stack it is automatically changed in all layer stacks which contain the 'Test layer' stack as included layer stack:



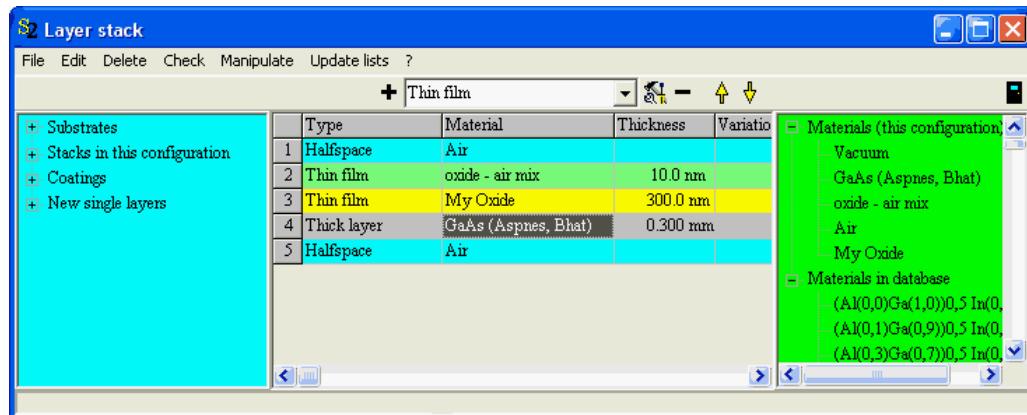
Note that layer stacks can be conveniently displayed in views. This is explained in the section '[Layer stack view](#)' (see below).

4.3.3 Assigning materials to layers

Each layer in the layer stack represents a slab of a certain material. There are several ways to 'fill' a layer with a material, either by drag&drop operations or using function keys on the keyboard.

Drag&Drop within the layer stack definition window:

On the right side of the layer stack definition window you see in a tree view the available materials:



Clicking the + or - signs in the treeview you can open or close the individual nodes. The first node called **Materials (this configuration)** gives a list of all materials in the list of materials of your current SCOUT configuration. The second one called **Materials in database** shows all materials available in the database to which you are connected at present. In order to assign a material to a layer in the layer stack, just drag it from the lists to the right and drop it onto the layer.

In case you drop a material from the database SCOUT checks if a material of this name is already in its list of materials. If yes, this material is used and nothing is loaded from the database. If the wanted material is not yet in the list of materials, it is fetched from the database and entered into the list of materials. Note that the treeview of the available materials is not updated automatically, so the new material will not show up in the node 'Materials (this configuration)' unless you apply the menu command **Update lists**.

Drag&Drop from the material list:

For this method you have to open both the layer stack list and the material list. In the material list, point with the mouse cursor on the wanted material, press the left mouse button without releasing it and move the cursor to the appropriate layer in the layer stack list. Now release the mouse button and check if the correct material is entered in the layer. Each layer in the stack can be filled by such a drag-and-drop operation.

Keyboard commands:

If you have only a few entries in the list of materials this method is quicker and should be your favourite one. In the layer stack list, place the cursor in the cell that specifies the material of the layer of interest. Now press the F4 function key and observe that the material changes. By repeatedly pressing F4 you iterate through all entries of the material list, and you can stop when the appropriate material appears. You can move backwards in the list of materials by pressing F5.

In the case of [anisotropic layers](#), only the drag&drop assignment from the material list can be used.

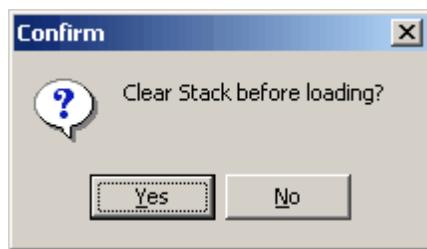
4.3.4 Working with pre-defined layer stacks

Saving and loading layer stack definitions

Layer stack definitions can be saved to files and re-imported as components to build complex layer stacks. **These operations are only possible if the assigned materials of all layers are stored in the database.** If you save a layer stack definition, SCOUT will check if all used materials are found in the database. If a layer stack definition is imported SCOUT will load all required materials from the database if there are not yet in the material list.

You can save the layer stack definition to a file using the command **File|Save layer stack definition**. SCOUT will create a file with the extension *.ls which contains the layer structure and the names of the assigned materials. **Note that only the layers between the halfspaces are saved, but not the assignments of the halfspaces.**

You can re-import a layer stack definition with the command **File|Import layer stack definition**. First you are asked if you want to delete the current layer stack before you import the new one:



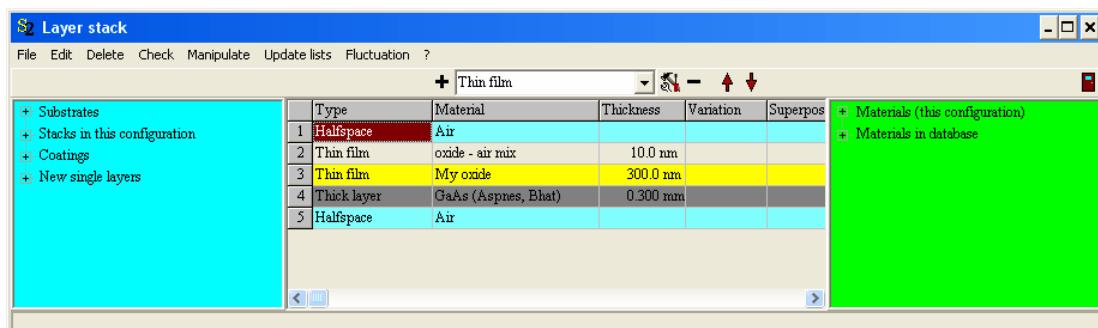
If the current layer stack is not cleared the new layers are added before the currently selected layer. Then you have to specify if the imported layer stack should be imported in the original order or inversed:



Finally a dialog searching for a layer stack definition file (extension *.ls) is used to specify the layer stack definition to be imported.

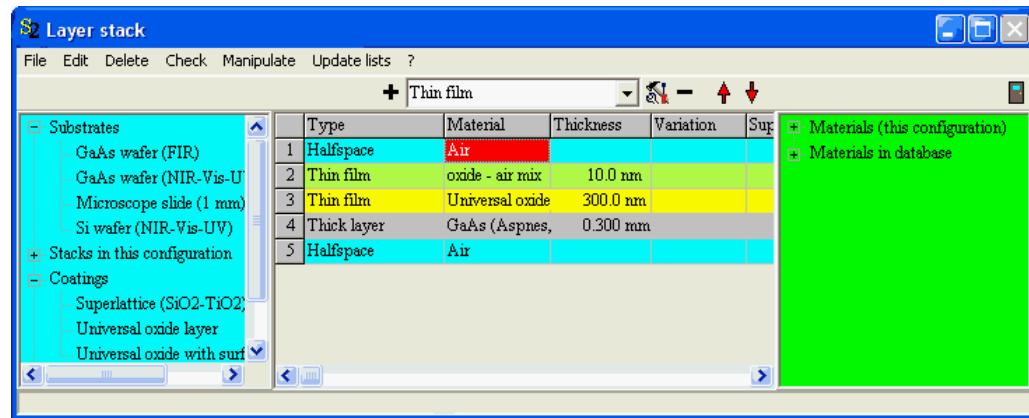
Database access

The treeview on the left side of the layer stack window makes the [pre-defined layer stacks of the database](#) accessible:

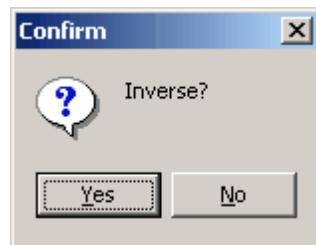


The nodes **Substrates** and **Coatings** show pre-defined layer stacks in the subfolders 'Substrates'

and 'Coatings' of the database (SCOUT looks for layer definitions in the subfolders 'Panes' and 'Gases' as well which are mainly used by users of the CODE software). Clicking the + or - signs you can open or close the nodes of the treeview:



Drag a pre-defined layer stack from the treeview and drop it into the layer stack definition. You will be asked, if the partial layer stack should be inverted before it is inserted in the stack definition:



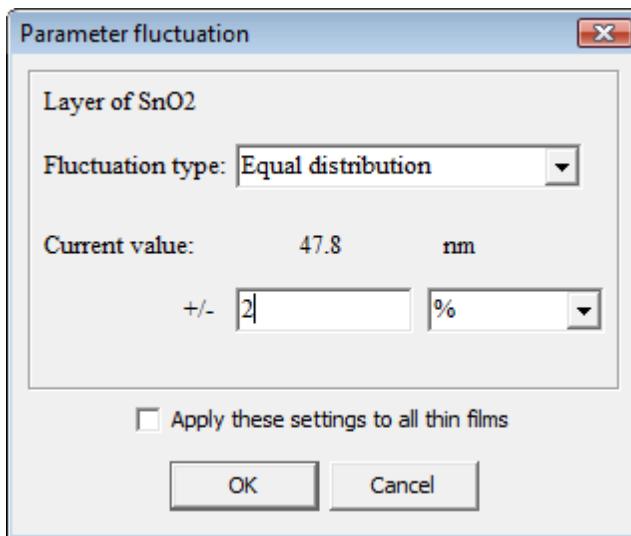
If you drop it on a layer, the coating will be inserted above this layer.

Insertion of other stacks of your SCOUT configuration

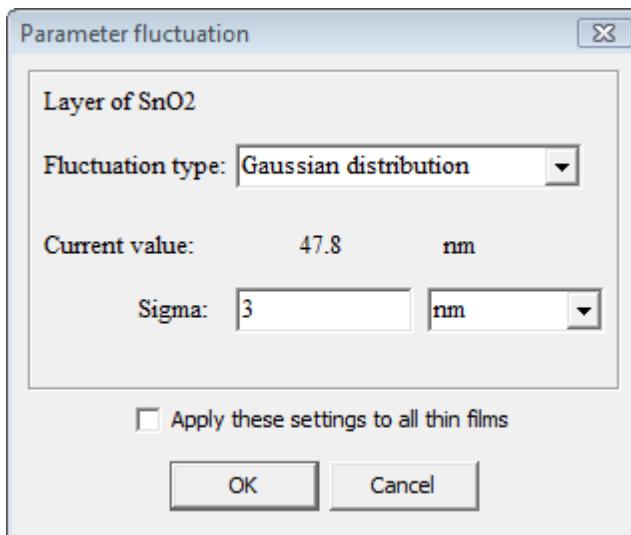
If you work with several layer stacks in your configuration, the treeview node 'Stacks in this configuration' will show all the stacks (Use the menu command 'Update lists' to update the treeview). If you drag one of the layer stacks in your layer definition it will be entered as an 'Included layer stack' (see [above](#)).

4.3.5 Thickness fluctuations

Designing new layer stacks, you might be interested in the change of properties, e.g. color coordinates, due to thickness uncertainties in the production of the stack. Special objects like the [ColorBox](#) object compute property variations as the result of thickness fluctuations. Before you can do these kind of special computations you have to set the fluctuation properties of each layer that may have a thickness uncertainty. This is done by the menu command 'Fluctuation' which opens a dialog like this for the currently selected layer:

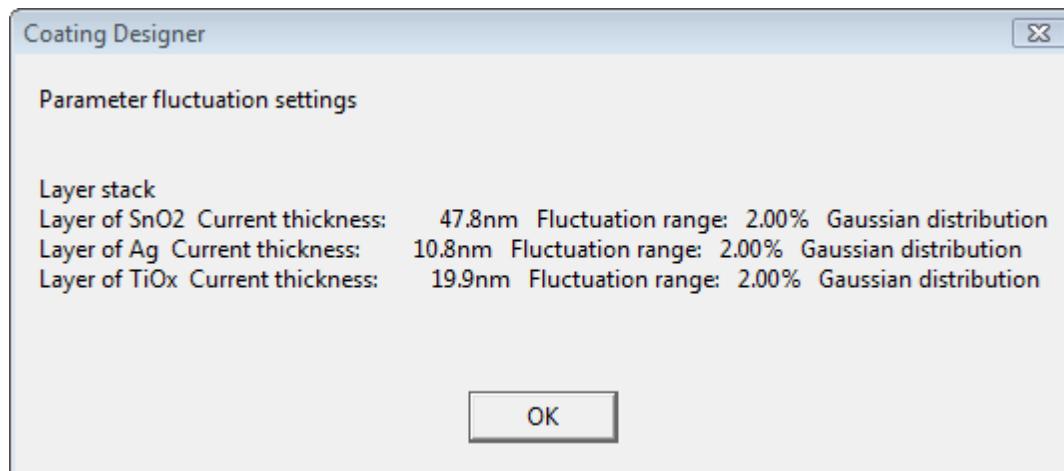


Select the type of distribution (equal distribution, Gaussian) and the 'amplitude' of the fluctuation in % or in absolute values:



If you check the option **Apply these settings to all thin films** the selected fluctuation parameters will be assigned to all thin films in the stack. Be sure that you really want to have the same fluctuation settings for all thin films if you activate this checkbox.

A summary of all fluctuation settings of the current configuration can be displayed using the menu command **File|Report|Parameter fluctuation settings** in the main window of SCOUT:



4.3.6 Wizards

4.3.6.1 Overview

If you are in a hurry and you have no time for detailed optical modelling, one of the following wizards may help you.

- The [Pane wizard](#) determines optical constants of thick, transparent substrates such as glass panes or pieces of plastic.
- The [Foil wizard](#) helps to determine the optical constants of plastic foils enclosed between two glass panes (safety glass).
- The [Glass coating wizard](#) extracts intensity transmission and reflection coefficients for coatings on glass panes. You can use these numbers to compute spectra of the coating on different types of glass - since this wizard does not compute optical constants, there are severe limitations of the application of glass coating wizard results.

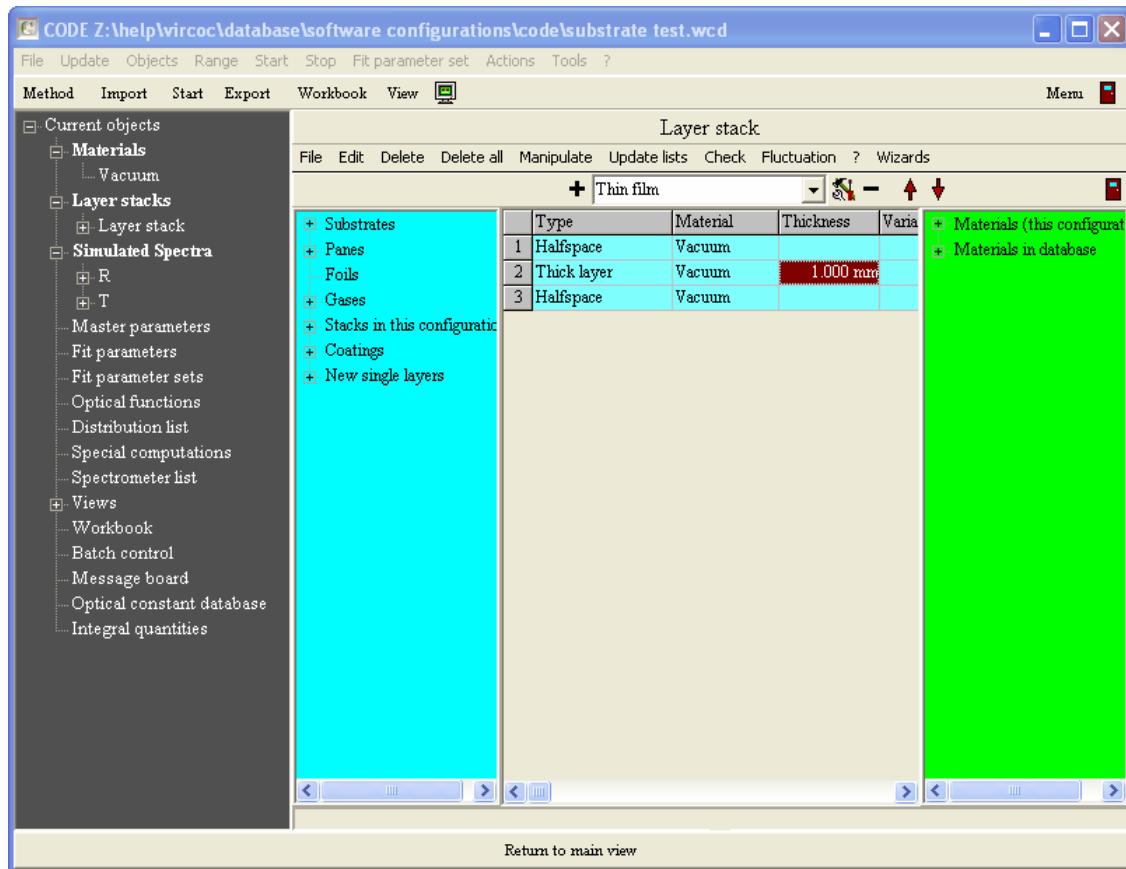
4.3.6.2 Pane wizard

This wizard determines the optical constants of transparent, thick substrates such as glass panes. You must provide a reflectance and a transmittance spectrum, both recorded at normal incidence of light. In the case of reflectance you can use a spectrum recorded at 8° angle of incidence if the measurement has been performed at random polarization. This is usually achieved by applying no polarizer in the measurement.

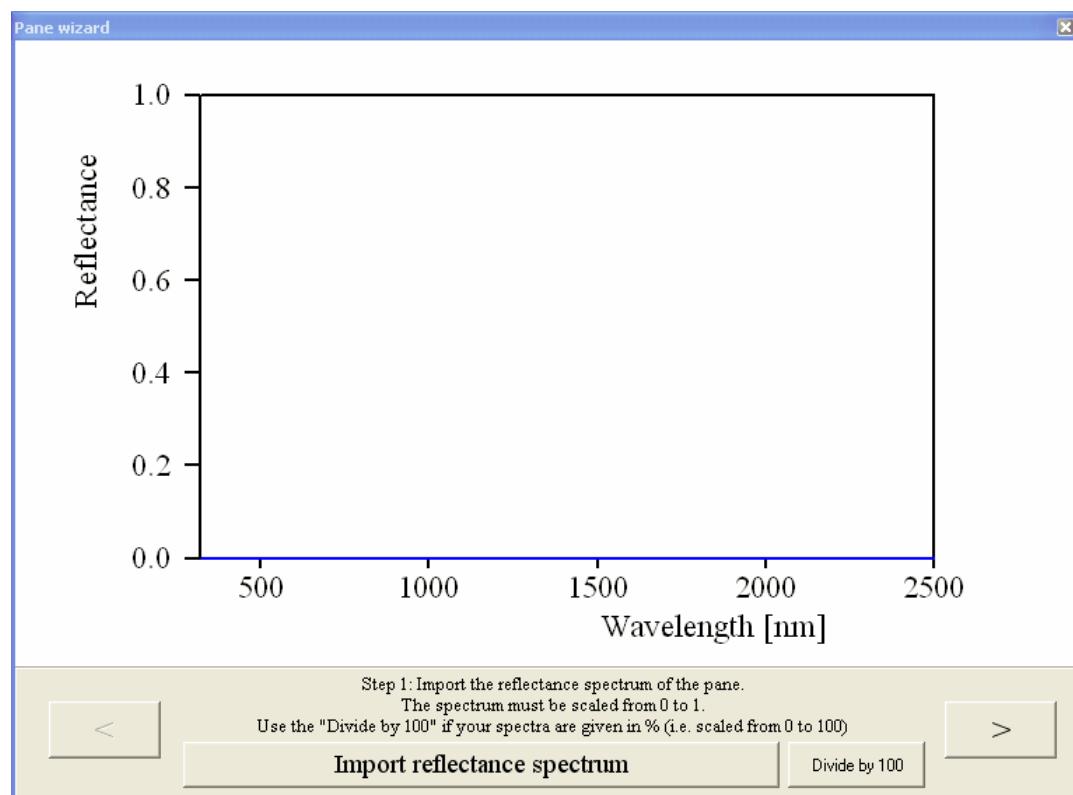
Before you apply the wizard by executing the layer stack command **Wizards|Pane wizard** you should check the following settings:

- Select the database (menu command in the main window: **File|Options|Database**) where the new pane should appear
- Clear the layer stack and create a new single layer of type **Thick layer**. The material assignment is not relevant at this time - it will be modified by the wizard. The thickness of the layer will also be set by the wizard.
- Be sure that you have data files with the reflectance and transmittance spectra which can be imported by SCOUT. Both the reflectance and transmittance must be larger than 0, and their sum should not exceed 1.

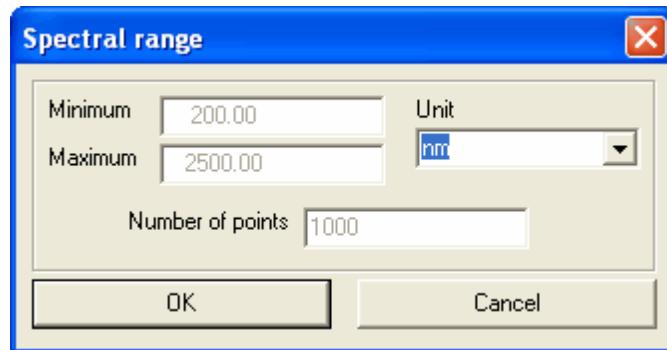
In the following starting situation



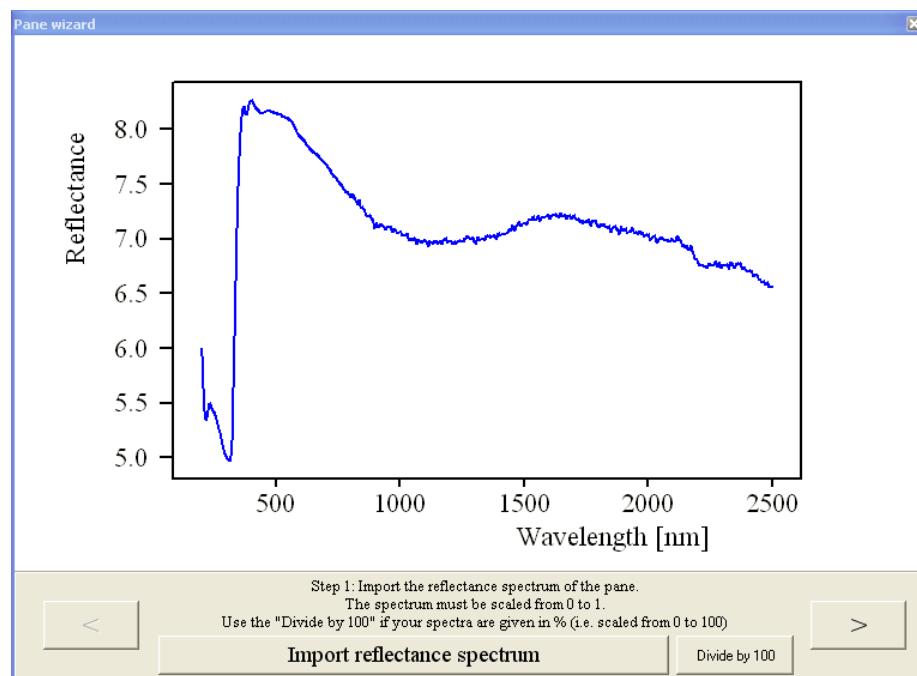
now execute the layer stack command **Wizards|Pane wizard**. The pane wizard window opens:



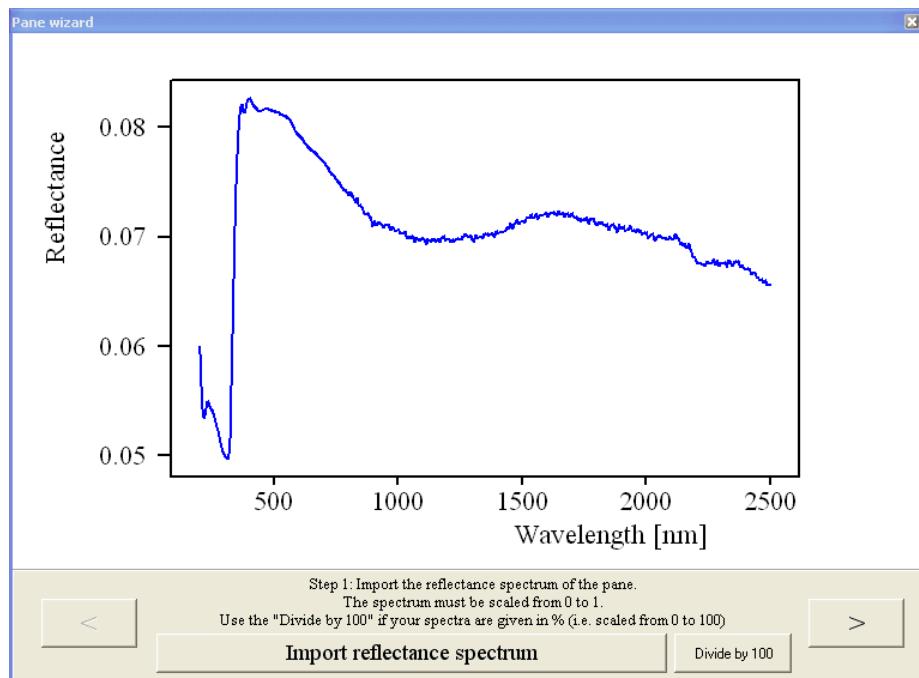
Now simply follow the instructions given by the wizard. First action is to import the reflectance spectrum (press the corresponding wizard button at the bottom). After you have selected the file make sure the spectral unit in the following dialog is set correctly:



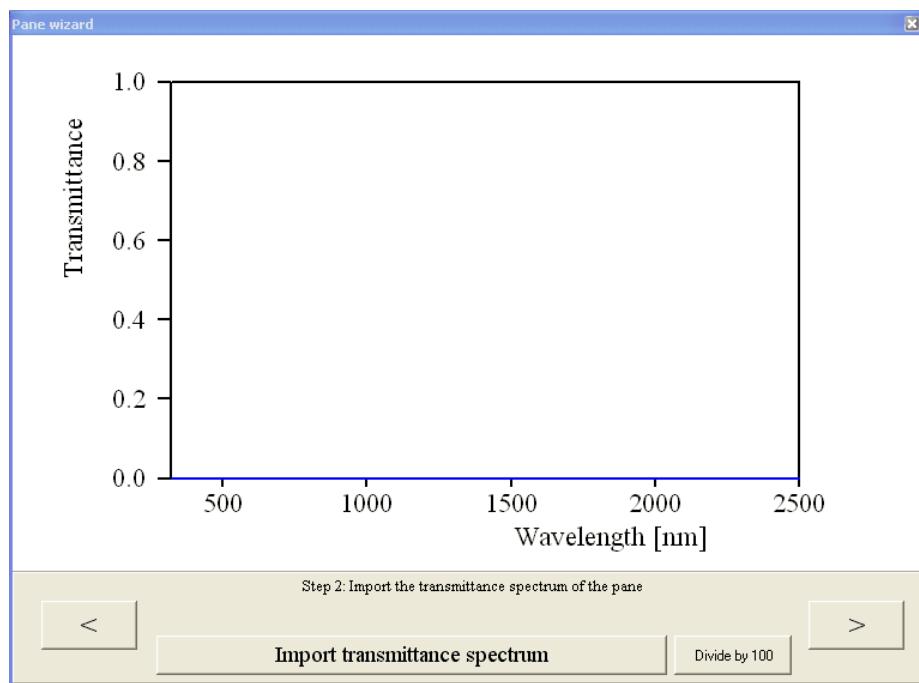
If the data are scaled in % (as in the example below)



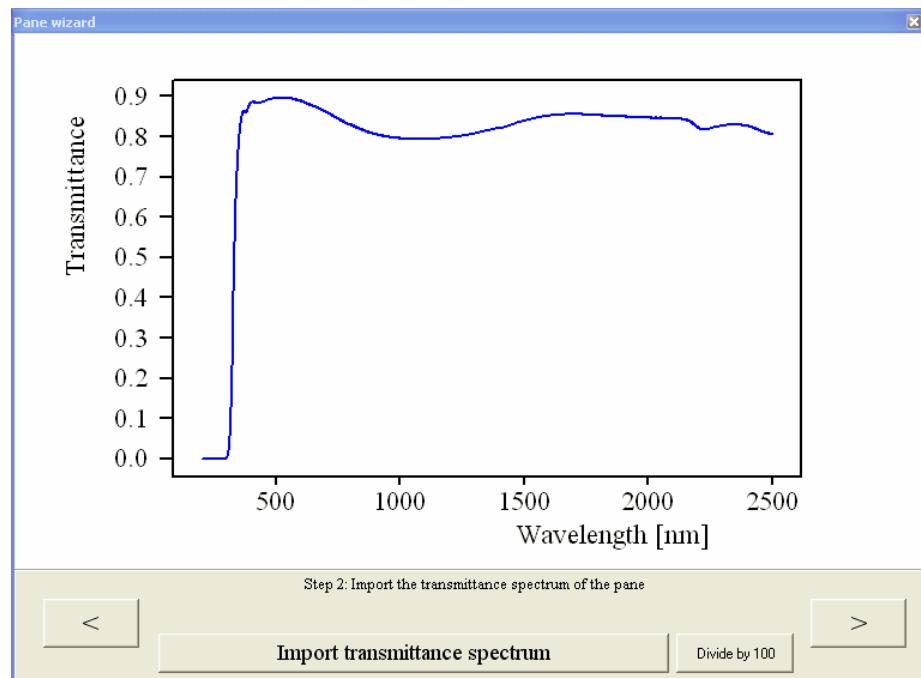
press the button 'Divide by 100':



Now click on the > button to go to the next step:

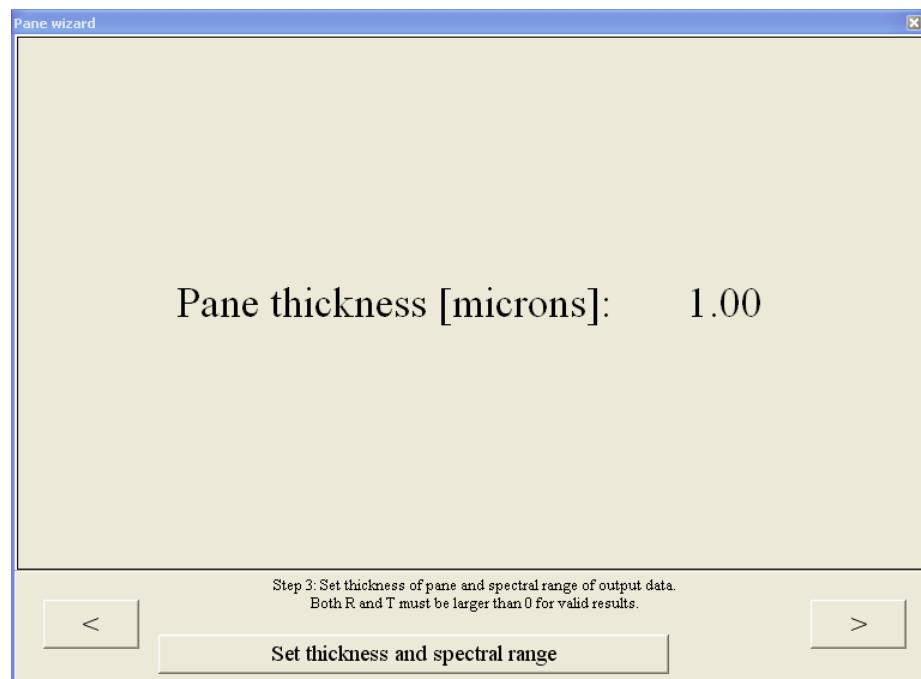


Import the transmittance spectrum and divide by 100 if necessary:

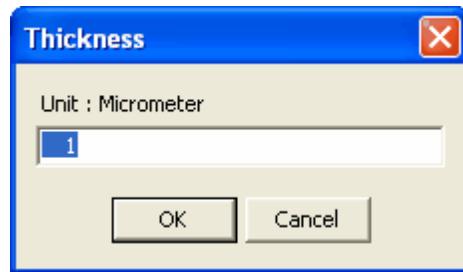


The method used by the wizard requires that the transmittance is larger than 0. Please check out in which wavelength range this condition is true.

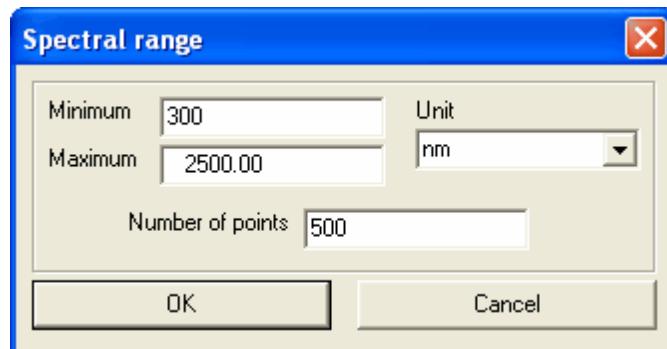
When you are done with transmission, click on >:



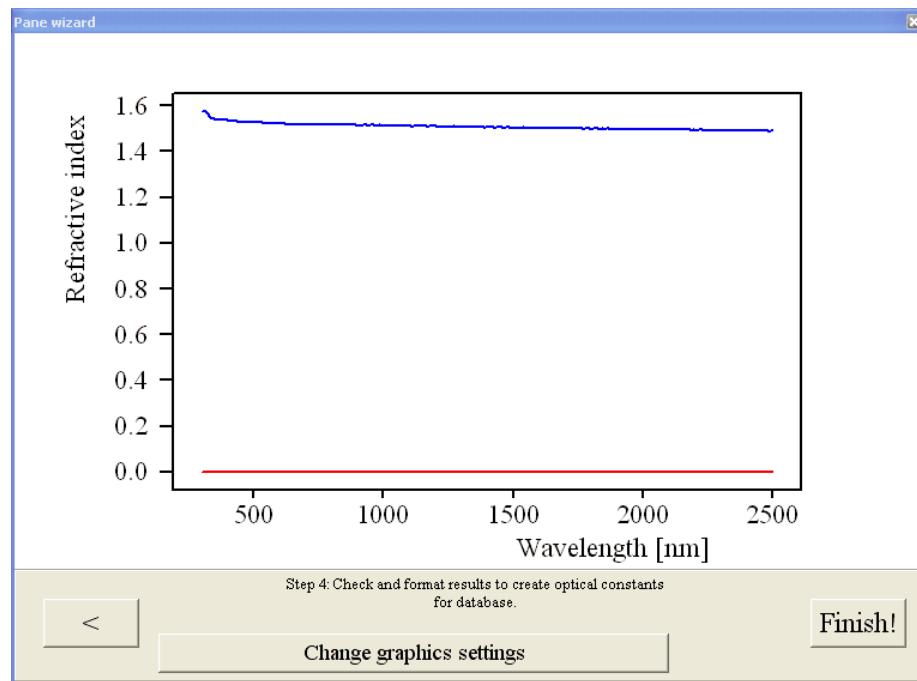
Now click the button 'Set thickness and spectral range' and enter the thickness in the following dialog (be aware of the unit micrometer):



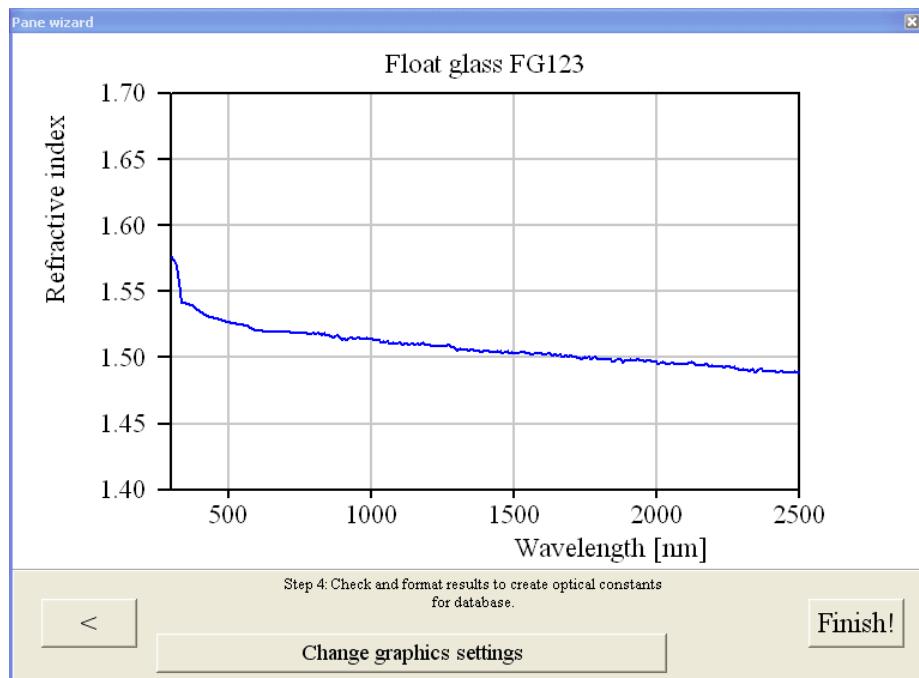
Finally set the spectral range for which you have valid reflectance and transmittance data ($T>0$, $R>0$, $1-R-T>=0$):



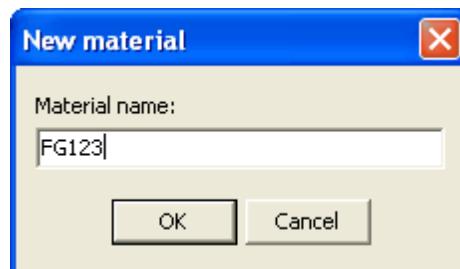
We are now almost through. Click on > again to activate the final computation:



The wizard window now displays the computed complex refractive index. Click the 'Change graphics settings' button if you want to modify the graphics settings. Note that this graph is the one that will be displayed in the database of optical constants. After your modification the graph may look like this:



Now click **Finish**. The wizard asks you for a name which will be used for the material in the database:



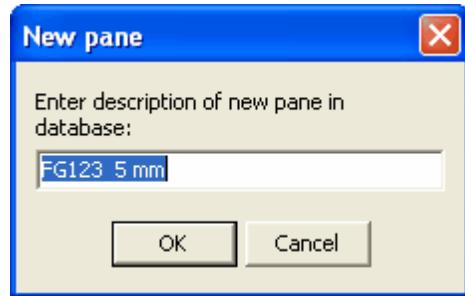
In the next dialog you should enter your name since you are responsible for the new database item:



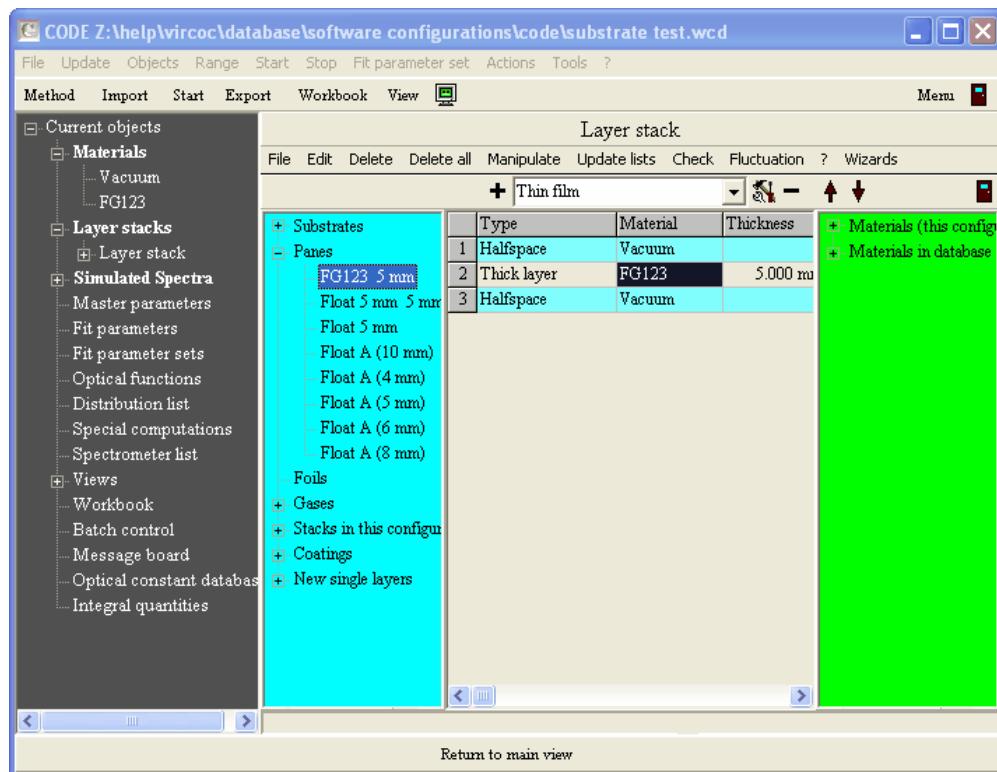
Next you can enter a comment for the new database material:



Finally a database item in the folder Panes is created using the new material and the thickness you have specified in the wizard dialog above. You can set the name of this pane item in the last dialog of the wizard:



The wizard now closes and you will find a new pane item (after execution of the menu item **Update lists**) in the panes subfolder of the database:

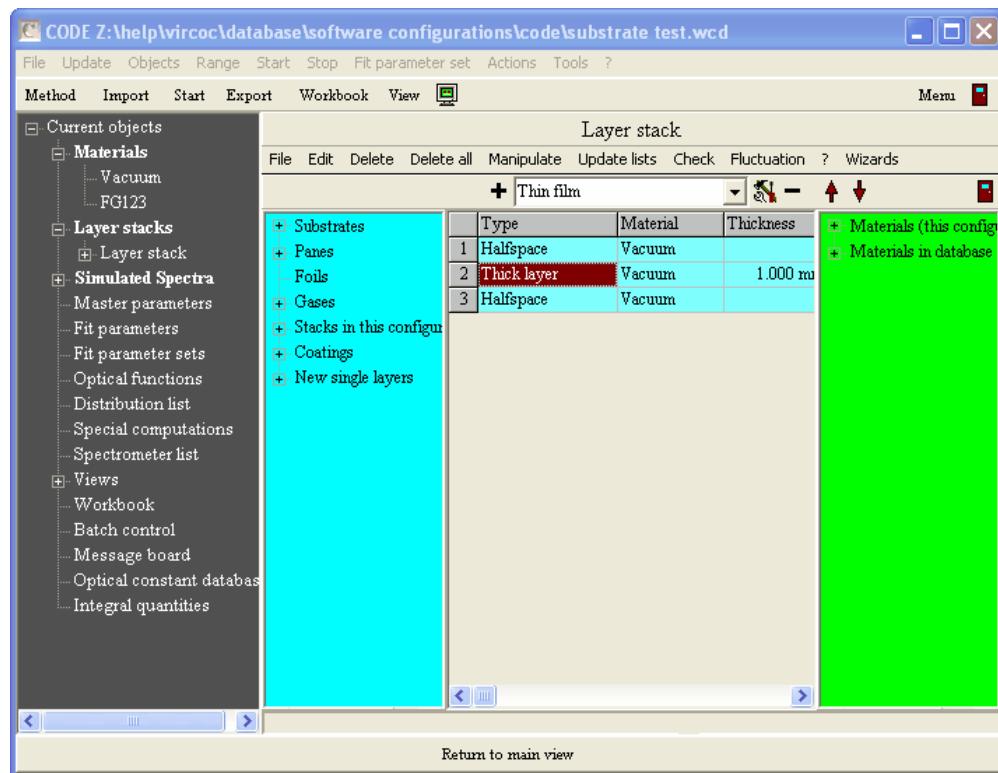


4.3.6.3 Foil wizard

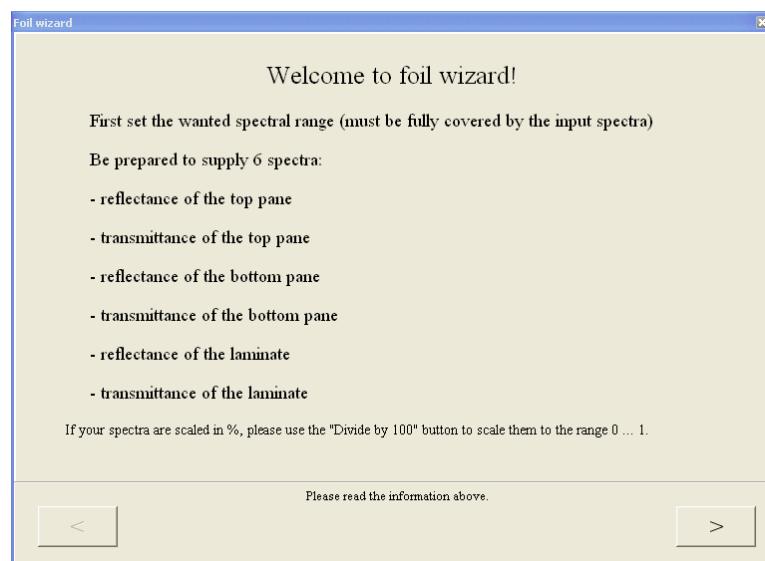
The foil wizard computes optical constants of plastic foils embedded between 2 glass panes. Before you apply the wizard by executing the layer stack command **Wizards|Foil wizard** you should go through the following preparational steps:

- Select the database (menu command in the main window: **File|Options|Database**) where the new foil should appear
- Clear the layer stack and create a new single layer of type **Thick layer**. The material assignment is not relevant at this time - it will be modified by the wizard. The thickness of the layer will also be set by the wizard.
- Be sure that you have all required data files which can be imported by SCOUT. You will need 6 spectra: Both the reflectance and transmittance of the two panes that enclose the foil, and reflectance and transmittance of the laminated package pane top / foil / pane bottom.

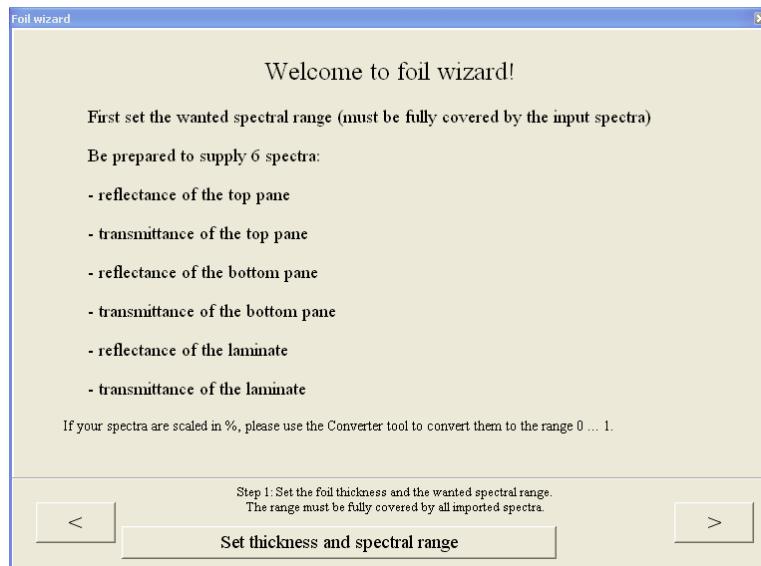
The starting situation is the same as in the case of the pane wizard:



Start the wizard with the command **Wizards|Foil wizard**:



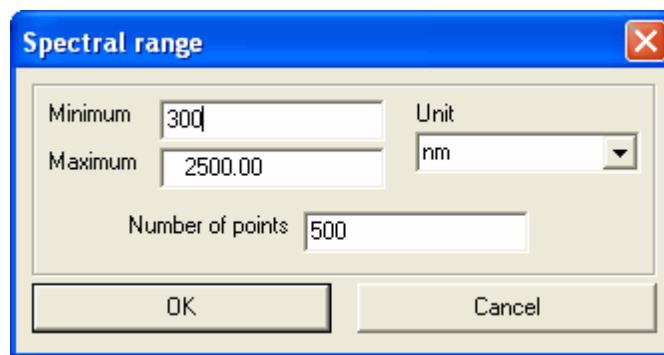
Click on > to start the work:



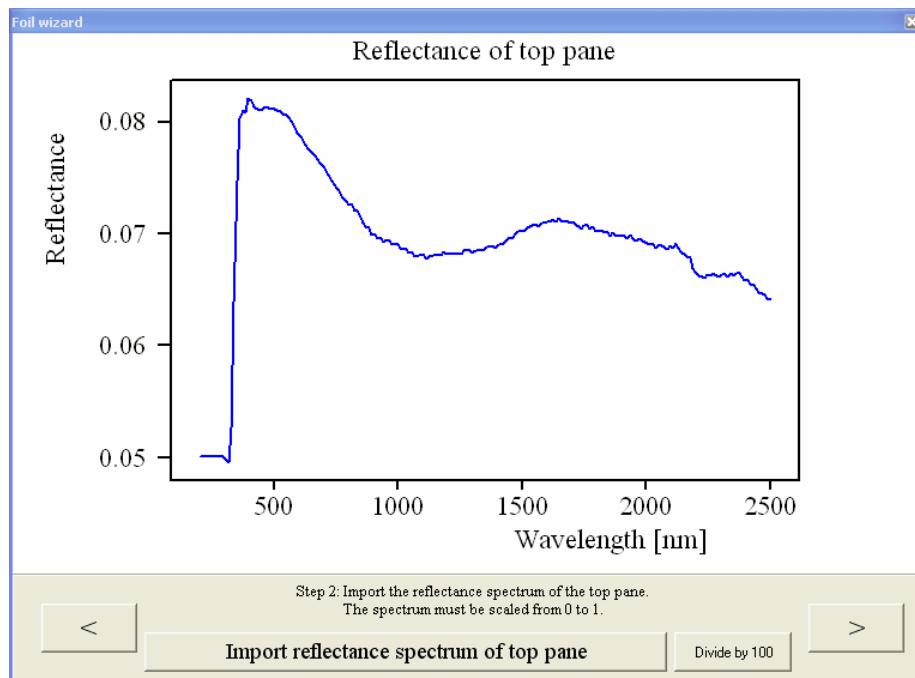
First click on **Set thickness and spectral range**. This will guide you through the following dialogs to enter the foil thickness



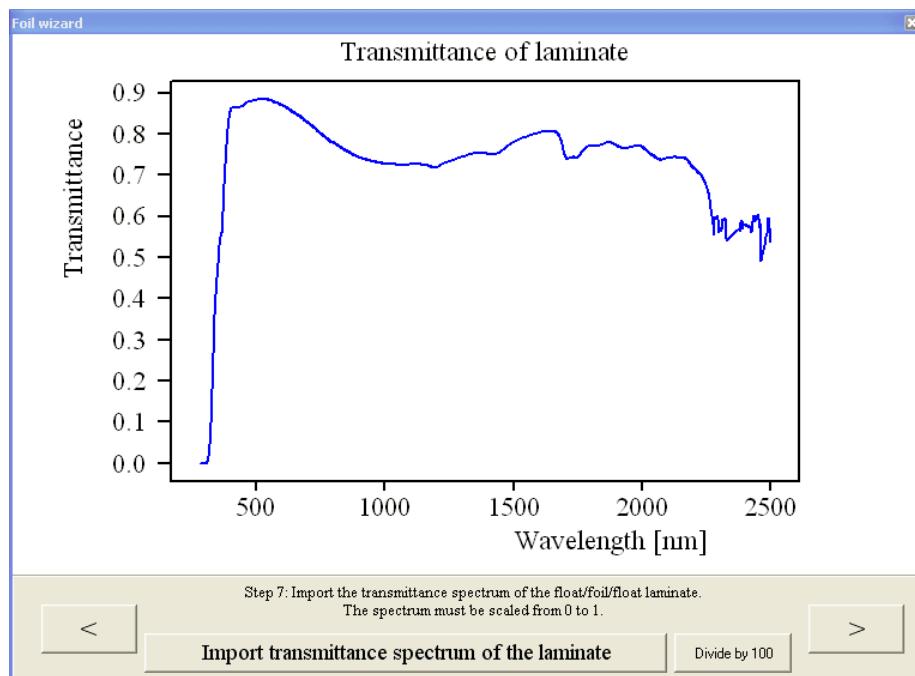
and to specify the spectral range of the analysis:



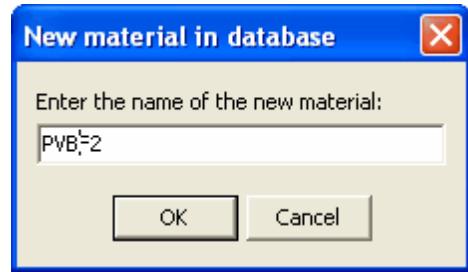
When you are done with these settings, press > to go on. You now have to import the 6 spectra, starting with the reflectance of the top pane:



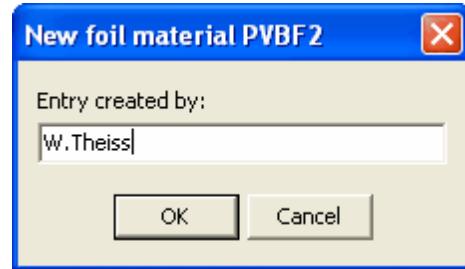
When you are ready with the import of the current spectrum, check if you have to divide by 100 (click the corresponding button, if necessary). Then press **>** to go to the next step. The last spectrum is the transmittance of the laminate:



Finally press the **Finish** button to enter the result in the database. You have to set a name for the new material



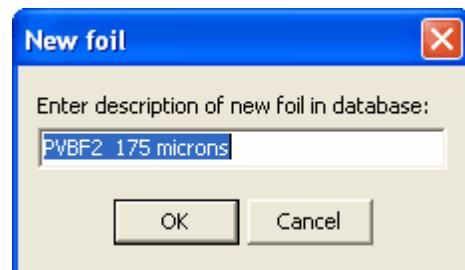
enter your name



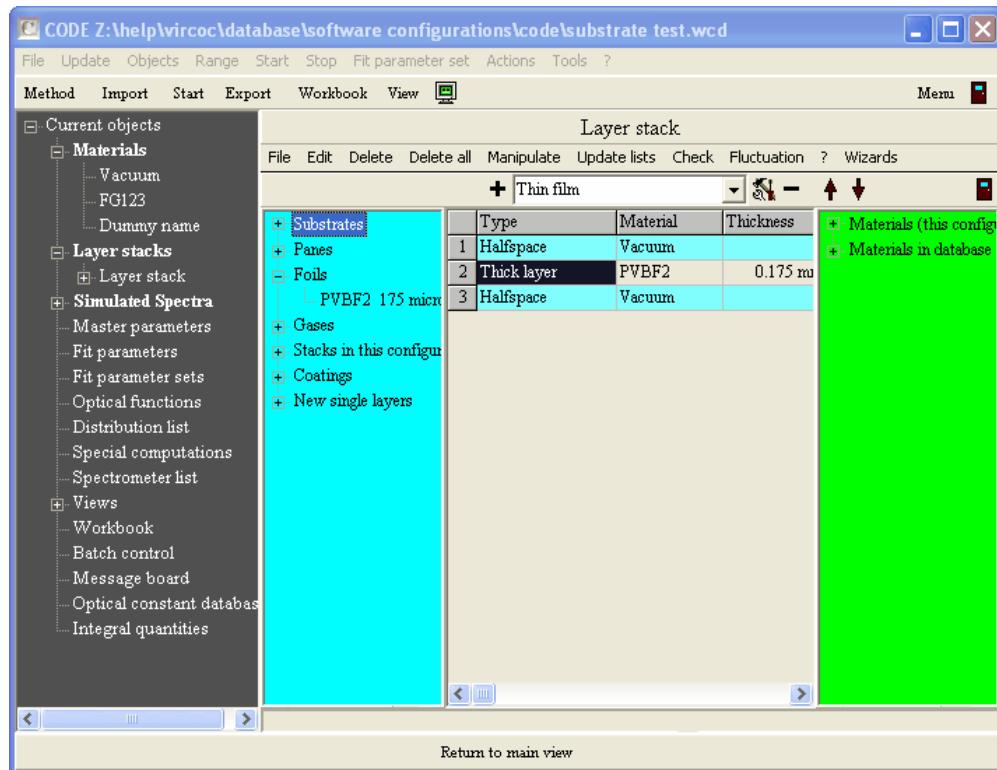
and a comment



After the material has been entered to the database, the wizard suggests a name for the new foil item in the foil subfolder. In the following dialog you get the chance to modify this identifier:



After the command Update lists you will find the new foil item in the Foils subfolder of the database:



4.3.6.4 Glass coating wizard

Coming soon ...

4.4 Spectra

4.4.1 Overview

After having defined the materials of your project and the structure of the layer stacks you have to specify which spectra are to be computed. This is done in the **spectra list** which collects all spectra that you want to calculate. Each spectrum is defined in its own window where you can also import an experimental spectrum for visual comparison and automatic fitting of the model. The goal of the automatic fitting is to minimize the deviation (mean squared difference) between simulated and measured data.

If you fit several spectra (Reflectance and transmittance, transmittance and ellipsometry, reflectance spectra at various angles of incidence or polarization) the total deviation is computed as the sum of the deviations of the individual spectra. The latter are multiplied with a weight factor that you can set for each spectrum in the list of spectra.

At present the following types of spectra can be simulated with SCOUT:

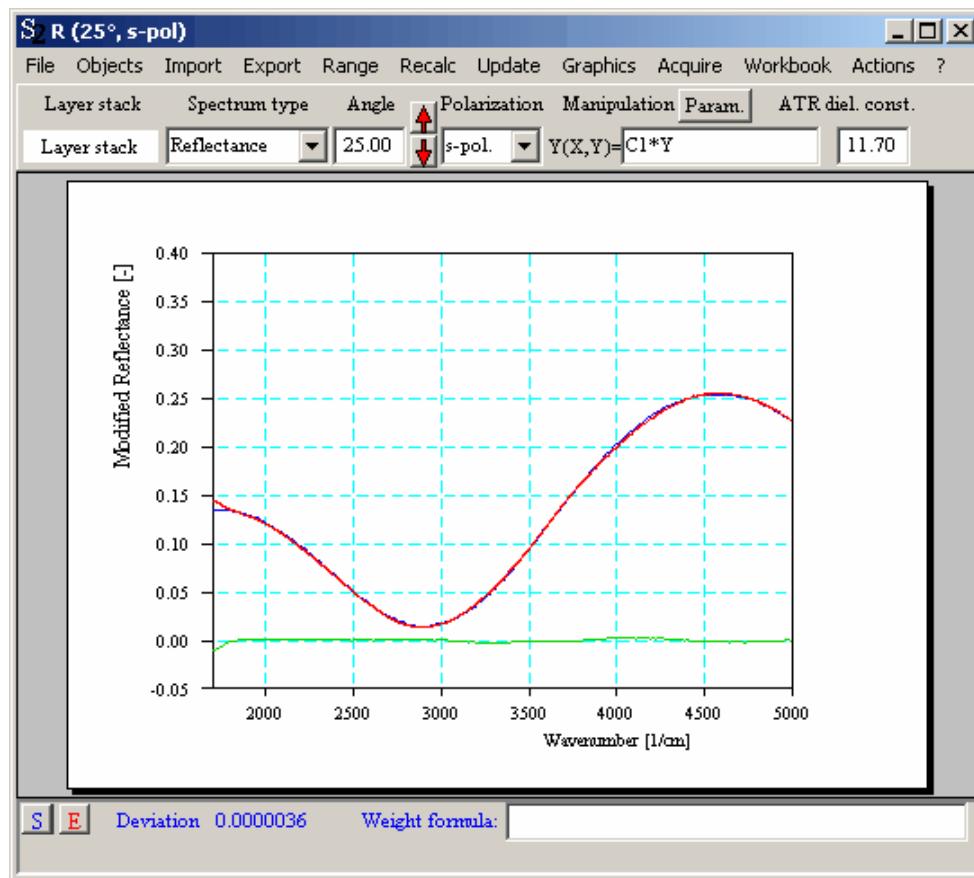
- [Reflectance](#)
- [Transmittance](#)

- [Absorbance](#)
- [ATR](#)
- [Layer Mix](#) (computes the average spectrum of a patterned sample)
- [Ellipsometry](#)
- [Ellipsometry stack average](#)
- [PL spectrum](#)
- [EL spectrum](#)
- [Layer absorption](#) (computes the absorbed fraction of incident light intensity in a certain layer in a stack)
- [Charge carrier generation](#) (computes the number of photon-generated charge carriers in a layer)

4.4.2 Reflectance, Transmittance, ATR

4.4.2.1 Introduction

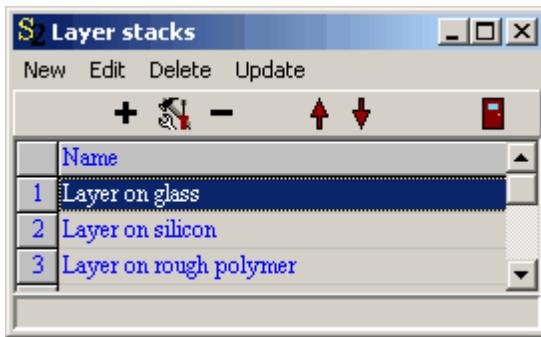
Reflectance, transmittance, ATR and related spectra can be computed using objects of type 'R,T, ATR'. Opening these objects with the **Edit** command in the spectra list you get access to a window like this:



You have to set several items to uniquely define the spectrum to be simulated:

Layer stack

If you work with several layer stacks you have to assign the layer stack for this spectrum. Open the list of layer stacks with **Objects|Layer stacks** in the main window:



Start a drag&drop operation from the layer stack you want to use. Drop it onto the white field below the label 'Layer stack' in the spectrum window. Convince yourself that the right layer stack has been assigned after this action.

If you want to compute reflectance and transmittance spectra for laterally structured samples you can use spectra of type '[Layer mix](#)' instead of the type 'R,T,ATR'.

Range

The **Range** command lets you define the spectral range of the simulated spectrum. As in previous SCOUT versions this range is not required to be the same as the one used for the dielectric function definitions.

Spectrum type

Here you can choose between different quantities that are computed. At present the following options are possible:

- Reflectance
- Transmittance
- ATR (attenuated total reflection)
- Absorbance ($= -\log(\text{transmittance})$)
- Refl. absorbance ($= -\log(\text{reflectance})$)
- 1-R-T

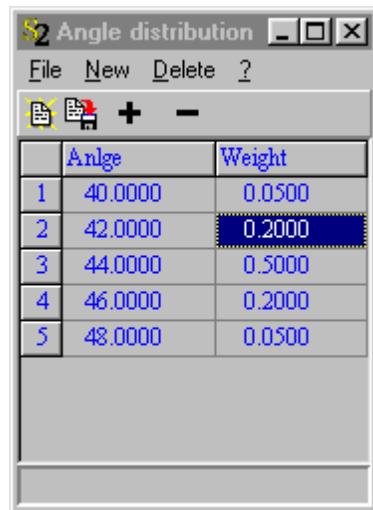
Angle

This is the angle of incidence being zero at normal incidence of light and 90° at oblique incidence. As mentioned above the light is incident from the top halfspace of the layer stack.

You can also describe experiments where the illumination comes from the bottom of the sample by setting the angle to a value between 90° and 180° . If you choose 172° , for example, the spectrometer beam comes from the sample's back with an angle of incidence of $180^\circ - 172^\circ = 8^\circ$.

Averaging over several angles of incidence

Sometimes it is important to average optical spectra for a distribution of angles of incidence. You can do this in SCOUT the following way. Open with **Object|Angle of incidence distribution** a list where you can define (by pushing the '+' button) as many angles of incidence as you like. Each angle gets a weight:



As soon as there is an entry in this list SCOUT ignores the angle setting in the spectrum window (see above) but computes for each angle in the list the spectrum. Each spectrum is multiplied by the weight of its angle of incidence. Then all spectra are added to the final spectrum which is displayed in the spectrum window. Note that the user, i.e. you, is responsible for the correct normalization of the weights (at least up to now). Of course, the sum of all the weight values should sum up to unity to represent a nice distribution function.

If you want to return to the normal setting of a single angle of incidence in the spectrum window you have to delete with the '-' button all entries in the list of angles of incidence.

Polarization

For the polarization the following settings can be used:

- s-pol: S-polarization
- p-pol: P-polarization
- mixed: A [user defined mixture of s- and p-polarization](#)

Manipulation

After the computation of one of the spectrum types listed above you can apply a [user-defined formula](#) to finally modify the spectrum. By using a special keyword in the formula you can refer to the reference spectrum mentioned below. See the section on spectrum normalization for details.

Target spectrum

In addition to the simulated and the measured spectrum, you can display a target spectrum. This is useful if you use SCOUT for production control: In this case the target spectrum would represent the spectrum of the ideal spectrum, i.e. it represents what you wish to see for the production all the time. You can set the target spectrum in a separate object that you open with **Objects|Target spectrum**.

With **File|Options|Target spectrum** you can control how the target spectrum is displayed: Activate the checked menu item **Show** to show it, set the **Pen** to determine which pen is used to draw it, and select **Line mode** in order to set the line mode for drawing the spectrum.

ATR diel. const.

If you compute ATR spectra, the incident beam is incident from a material with an almost real refractive index which is higher than the real part of the refractive index of the first material in the layer stack. You could replace the usual 'Vacuum' in the top halfspace of the [layer stack definition](#) by a suitable material and compute a regular reflectance spectrum. This is tedious and not very practical if you want to simultaneously analyze reflectance, transmittance and ATR spectra of one and the same sample. You would have to work with two different layer stacks in this case.

In order to avoid this complication, SCOUT lets you define the parameter **ATR diel. const** which is

used only if you compute ATR spectra. In this case the top halfspace material assignment (usually Vacuum, as mentioned above) is ignored and a material with a constant and real dielectric function is used instead. The value of the constant real part of the dielectric function is given by the parameter **ATR diel. const**. The square root of this number is the refractive index of the ATR element.

Typical values are 11.7 for silicon and 16 for germanium in the infrared, or 2.25 for glass in the visible.

Import of experimental spectra

Experimental spectra can be imported using the **Import** command. After data import the experimental data are multiplied automatically by the spectrum defined in the [Reference](#) window.

Computing the deviation

The deviation of the simulated and measured spectrum is computed the following way: For each data point of the simulated spectrum the value of the measured spectrum is computed by linear interpolation. The difference of the simulated and measured value is squared. If a weight function (at the bottom of the window) is defined the value of the weight function at the spectral point is evaluated and used to multiply the squared difference. Finally the average of these numbers for all data points is computed. This is the deviation.

Special actions

Objects of type 'R,T,ATR' can perform some special actions which may be useful in some situations:

- [Direct determination of layer thicknesses](#) by the Fourier transform technique
- [Data smoothing](#) to reduce the noise in measured spectra

4.4.2.2 Spectrum normalization

In general SCOUT computes absolute reflectance or transmittance spectra with values between 0 and 1. If you want to compare simulated and measured spectra you must have also experimental data that are absolute. If you do not have absolute experimental spectra but only relative ones there are two ways to proceed in SCOUT.

One way is to normalize the relative experimental data using an appropriate correction spectrum to convert the data to absolute ones. The imported experimental data are multiplied automatically by the reference spectrum defined in the window that opens after the command **Objects|Reference**. You can apply a user-defined [formula](#) to compute the reference spectrum or, alternatively, you can [import](#) the reference spectrum from a data file. If you have measured, for example, the reflectivity of a sample with respect to a reference mirror, you can import the absolute reflectivity of the reference mirror (which must be known from other experiments) into the reference window. Then, **the imported relative data are multiplied with the reflectance of the reference mirror** and converted to absolute reflectivities this way.

The second possibility is to compute relative spectra and compare them to relative experimental data. Of course, both quantities must be 'relative' in the same sense. An example is the following: In the experiment the transmission of a pure substrate is recorded as the reference, and then the transmission of a thin film deposited on that substrate is measured with the ratio of the two samples as experimental output. You might want to compare this ratio spectrum to the corresponding simulated ratio. To do this you have to proceed as follows: First, compute the absolute transmission spectrum of the blank substrate and store the spectrum to a data file, using the SpectraCalc format, for example. Then import this spectrum in the reference spectrum window. Set the content of the **Manipulation** edit box in the spectrum simulation window to ' $Y(X,Y)=Y/REFERENCE$ ' and activate **Recalc**. You should now obtain a simulated 1.0, since you are computing the ratio of the substrate transmission and the substrate transmission. Now modify the layer stack and compute relative transmission spectra which now can be compared to the measured ones.

If the **Manipulation** formula contains the keyword 'REFERENCE' it is assumed that you want to work with relative simulated spectra. In this case, **the experimental data are not multiplied with the reference spectrum after import!**

4.4.2.3 Polarization of the incident radiation

The mixture of s- and p-polarization is set by a user-defined function (giving the fraction of s-polarization) which is accessible in a separate window. This window is opened by the **Objects|Polarization** command. Here you can define the function using the elements given in the [technical notes](#). The default value for the fraction of s-polarization is 0.5, i.e. ideally unpolarized radiation is assumed.

Alternatively you can import data determined otherwise applying the [Import](#) command or simply use Drag&Drop to enter data from other sources, e.g. the *Data Factory*.

4.4.2.4 Manipulation of the computed spectrum

After the computation of the spectrum (Reflectance, Transmittance, ATR, Absorbance, ...) you can apply a [formula](#) to finally manipulate the simulated data before they are displayed and compared to the experimental values. The formula is entered in the **Manipulation** edit box.

If you want, for example, to calculate the square root of the reflectance as the final quantity of the simulation you have to set the spectrum type to reflectance and the formula to $Y(X,Y)=\text{SQRT}(Y)$.

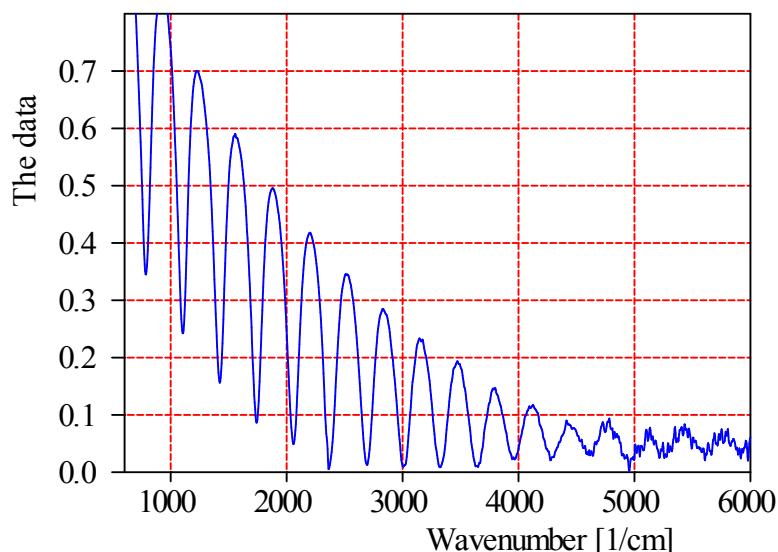
The symbol 'X' in the formula represents the spectral quantity displayed on the x-axis, i.e. the wavenumber, wavelength or energy, for example.

In your user-defined formula you can also refer (with the keyword 'REFERENCE') to the reference spectrum. This can be used to normalize the simulated spectrum. See the section on [normalization](#) for details.

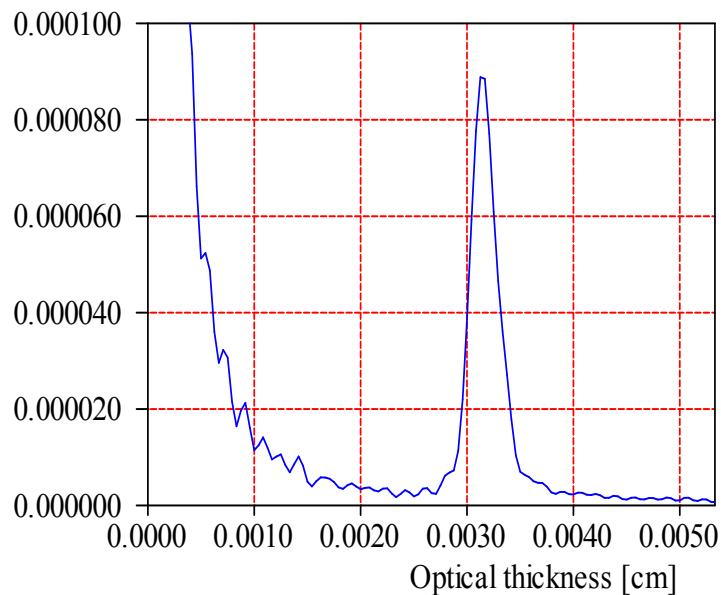
4.4.2.5 Thickness determination by Fourier transform

In the case of thickness determination by model fitting it is sometimes useful to obtain a good starting value for the thickness. This is in particular important in the case of thick layers.

If the spectrum exhibits pronounced interference patterns, and the refractive index of the deposited material is more or less constant, one can get a good approximation for the thickness from the Fourier transform of the spectrum. Here is an example:



If n is the refractive index of the layer, the peaks in the reflectance have a periodicity of $1/(2*n*d)$ where d is the layer thickness. The Fourier transform (power spectrum) of this spectrum shows a peak at the optical thickness $2*n*d$:



Knowing the value of n one can compute from the peak position of the optical thickness the geometric thickness d .

The menu command **Actions|FFT thickness analysis** of the spectrum window performs the analysis explained above. You are asked for the spectral range to be used for the periodicity analysis (wavenumber minimum and maximum, number of data points) and the expected geometrical thickness range (minimum, maximum) as well as for the refractive index (constant, real): For the choices 1000 ... 6000 1/cm with 2048 data points and a thickness range 3 ... 50 microns with $n=2.0$ SCOUT computes the power spectrum and searches for the highest peak in the specified thickness range. The obtained thickness is 7.88 microns.

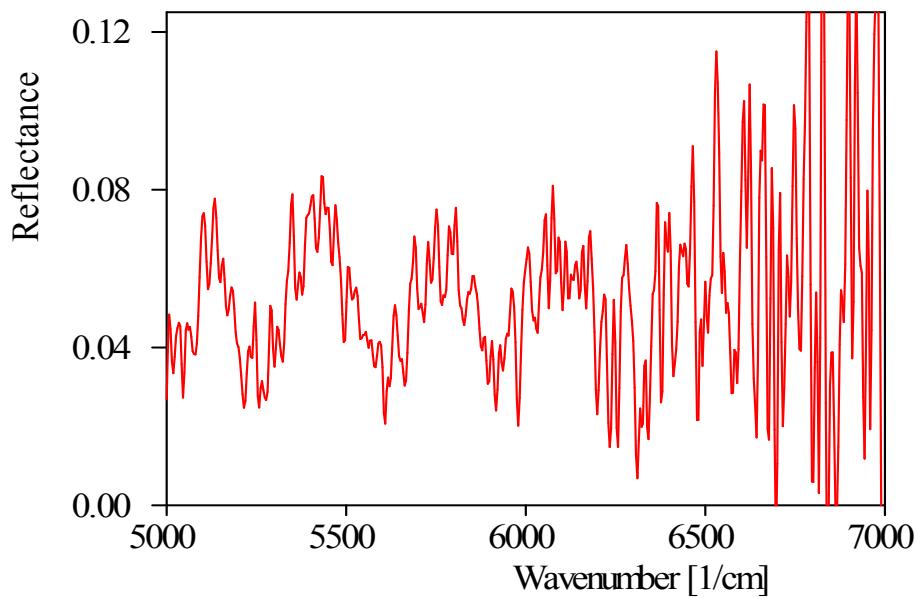
Once you have specified the parameters for the FFT thickness analysis, you can load new spectra and do the same analysis with the command **Actions|Repeat FFT thickness analysis**. In this case there are no questions about parameters but the previous ones are used once more.

4.4.2.6 Data smoothing

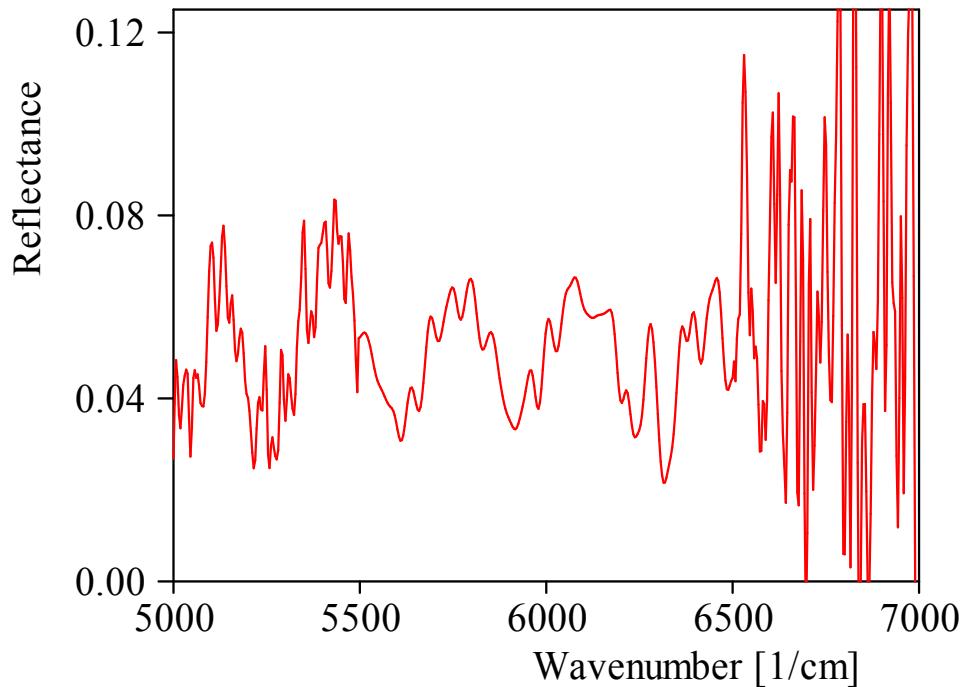
Noisy experimental data can be smoothed by the special action **Actions|Smooth experimental data**. You have to specify the spectral range for the operation and the number of times you want SCOUT to perform the smoothing action. A simple 3-point-smoothing is used (a point is replaced by the average of its value and those of the two neighbours in each iteration).

With the command **Actions|Repeat data smoothing** you can repeat the smoothing operation with the previous parameters.

Here is a short example. The original data



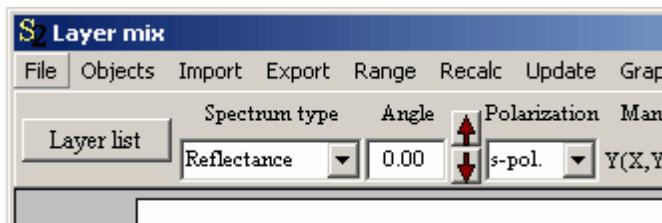
are smoothed in the range 5500 ... 6500 1/cm by applying the 3-point-averaging 10 times:



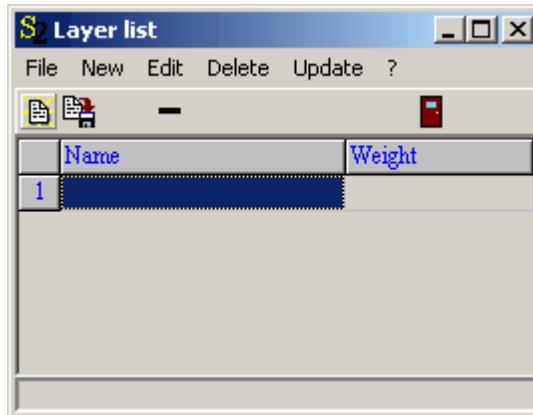
4.4.2.7 The spectrum type 'Layer mix'

If you want to compute reflectance and transmittance spectra for **laterally structured samples** you can use spectra of type '**Layer mix**' instead of the type 'R,T,ATR'. With objects of this type you can average spectra of different layer stacks.

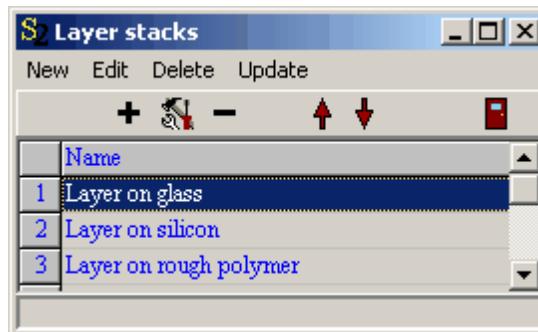
'Layer mix' spectra have a list of layer stacks rather than a single stack:



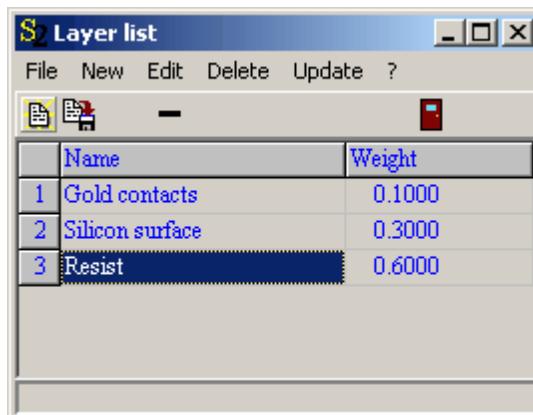
Press the **Layer list** button to open the list of the layer stacks used for this spectrum:



To add layers to this list drag&drop them from the list of layer stacks:



Each item in the list has a weight which should be proportional to the surface area that is covered by the corresponding layer stack. The sum of all weights should be 1.0 like in the following example:



For each layer stack in the list the spectrum is computed. The final spectrum is the sum of these

'sub-spectra' where each partial spectrum is multiplied by its weight factor.

4.4.3 Ellipsometry

4.4.3.1 Introduction

Besides reflectance, transmittance and ATR spectra (that are computed by 'R,T,ATR' objects in the spectrum list) you can also compute ellipsometry spectra. To do so, you have to create an object of type 'Ellipsometry' in the spectrum list.

The following items must be defined to uniquely set the spectrum to be simulated:

Range

The **Range** command lets you define the spectral range of the simulated spectrum. As in previous SCOUT versions this range is not required to be the same as the one used for the dielectric function definitions.

Spectrum type

Here you can choose between different quantities that are computed. At present the following options are possible:

- $\tan(\Psi)$, $\cos(\Delta)$
- Ψ , Δ (both in degrees)

Angle

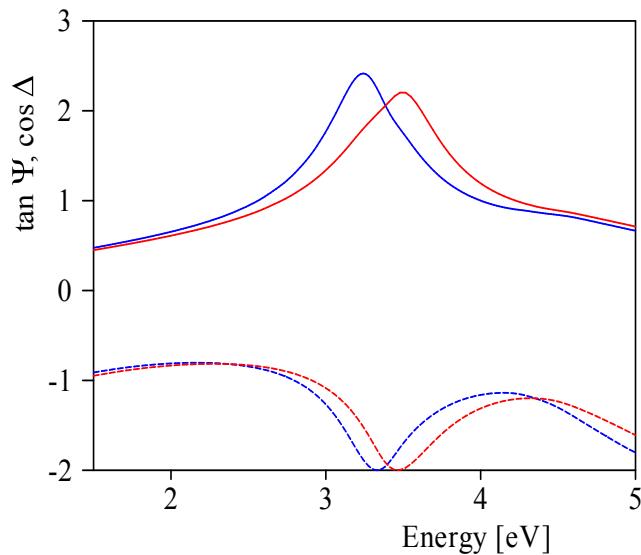
This is the angle of incidence being zero at normal incidence of light and 90° at oblique incidence. As mentioned above the light is incident from the top halfspace of the layer stack.

You can also describe experiments where the illumination comes from the bottom of the sample by setting the angle to a value between 90° and 180°. If you choose 172°, for example, the spectrometer beam comes from the sample's back with an angle of incidence of $180^\circ - 172^\circ = 8^\circ$.

4.4.3.2 Graphics

Since 'Ellipsometry' objects compute two quantities the graphics settings are more complicated as in the case of 'R,T,ATR' objects.

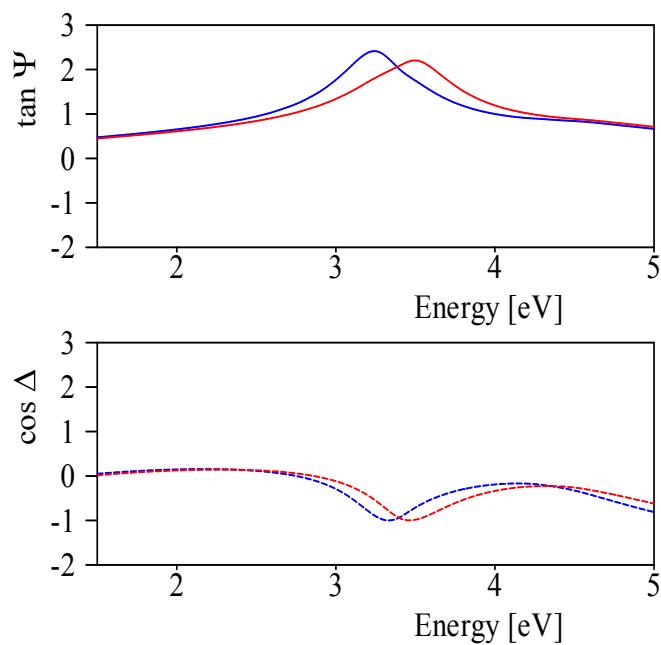
In principle, there are two independent graphs with corresponding graphics parameters, one for the quantity related to Ψ (Ψ itself, or $\tan(\Psi)$) and one for Δ (or $\cos(\Delta)$). However, the easiest handling of the data display is achieved with the default settings. These couple the two sets of graphics parameters to be the same except for the frame mode. This is set to 0 for the Δ graphics. A typical display is the following:



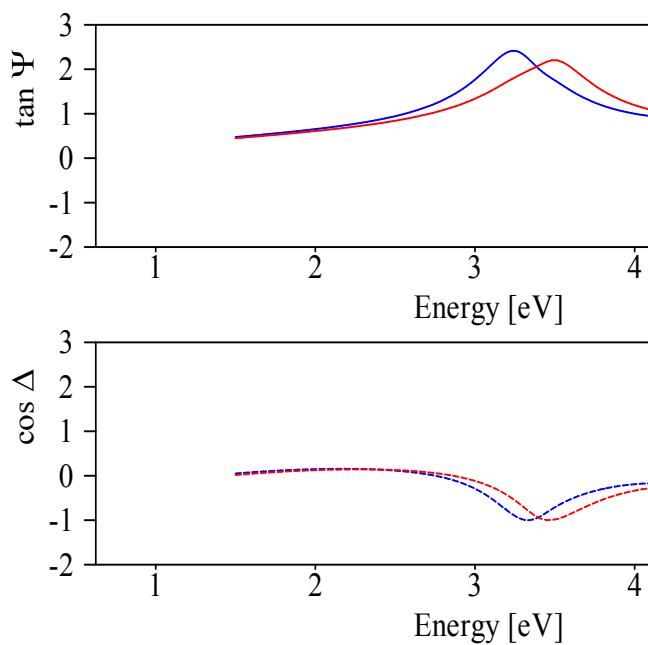
The simulated spectrum of $\tan(\Psi)$ is drawn in a blue solid line, the corresponding experimental data are in red (also using a solid line). The $\cos(\Delta)$ curves are dashed (blue for the simulation, red for the measured data).

The check box 'Lock graphics' is checked in the default configuration. This means that zoom and stretching operations affect both spectra, $\tan(\Psi)$ and $\cos(\Delta)$, the same way.

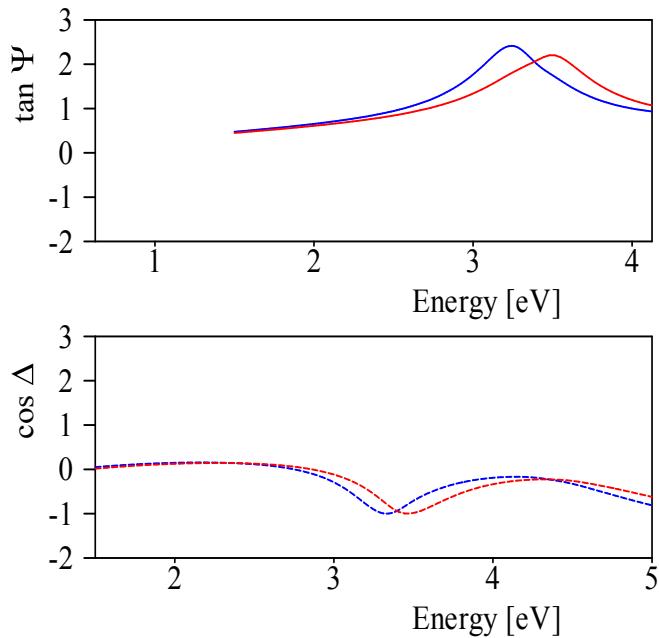
You can also edit the graphics parameters for Δ (command **Graphics|Delta**) and set the frame mode to a value larger than 0. This way you can generate two coordinate systems like in the following example:



If 'Lock graphics' is still checked the key and mouse commands to move through the graphics still apply to both graphs. For example, a pressing the left cursor key leads to this:



If you uncheck the 'Lock graphics' box all key and mouse commands apply to the Ψ curve only. The key command mentioned above would lead to the following appearance:



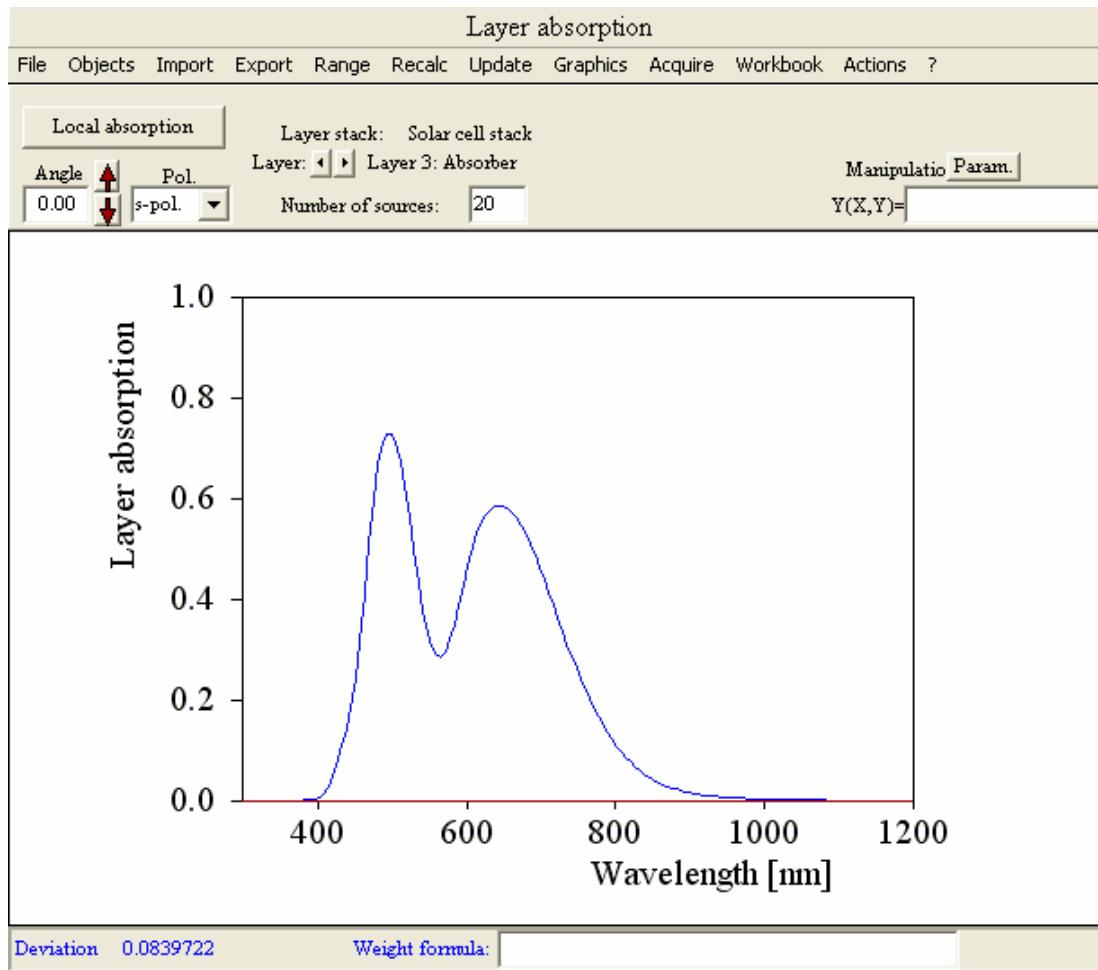
In this case, there are no quick graphics commands for the Δ spectrum in the present SCOUT version.

4.4.3.3 Stack averaging

A special type of ellipsometry spectrum is called 'Ellipsometry stack average'. You can use this object if the measured data can be considered as average spectra of different layer stacks. A typical case is a micro-structured sample. Please read the section about '['Layer mix' objects](#)' (in the chapter 'Reflectance, Transmittance, ATR') to learn how to work with this kind of object.

4.4.4 Layer absorption

This spectrum type lets you compute how much light is absorbed in a specific layer in a layer stack. The user interface looks like this:



In the upper left corner you specify the **angle of incidence** and the **polarization** of the illuminating beam. Right to the label 'Layer stack' you can drop the wanted layer stack from the list of layer stacks. Underneath the layer of interest in the stack is selected using the two little buttons . In order to compute the absorbed fraction in the layer, the layer is divided into sublayers for which the local absorption is computed. The number of sublayers is specified by '**Number of sources**'. The total light absorption in the layer is finally computed by integrating over all sublayers. Note that the computation of layer absorption can be quite slow if you work with many sublayers. After a computation, you can inspect the distribution of the local absorption in the layer for the last spectral point clicking the **Local absorption** button.

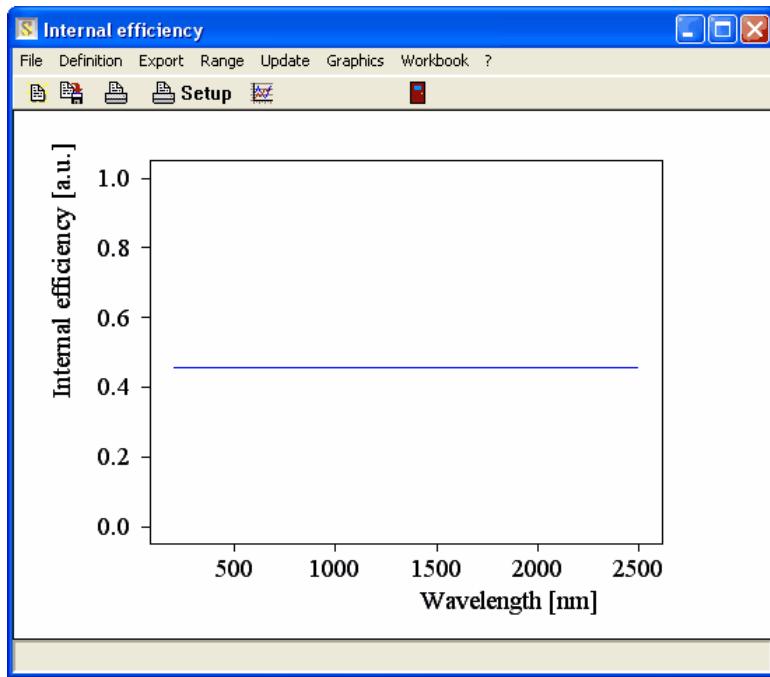
Like in all spectrum objects, the spectral range of the computation is set with the **Range** command.

4.4.5 Charge carrier generation

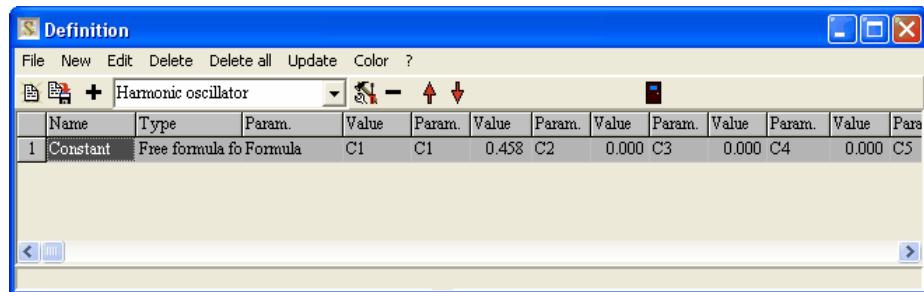
Objects of this type are based on [layer absorption](#) objects. The absorbed fraction of the selected layer (which is computed like in layer absorption objects) is multiplied with a user-defined internal efficiency. The efficiency gives the conversion probability for an absorbed photon to generate a charge carrier contributing to an electrical current through the layer.

The CODE software (which is an extension of SCOUT) contains an object type 'Photocurrent' to compute the current density in a solar cell layer stack. In order to do so, the object must be connected to an spectrum object of the type 'Charge carrier generation' discussed here.

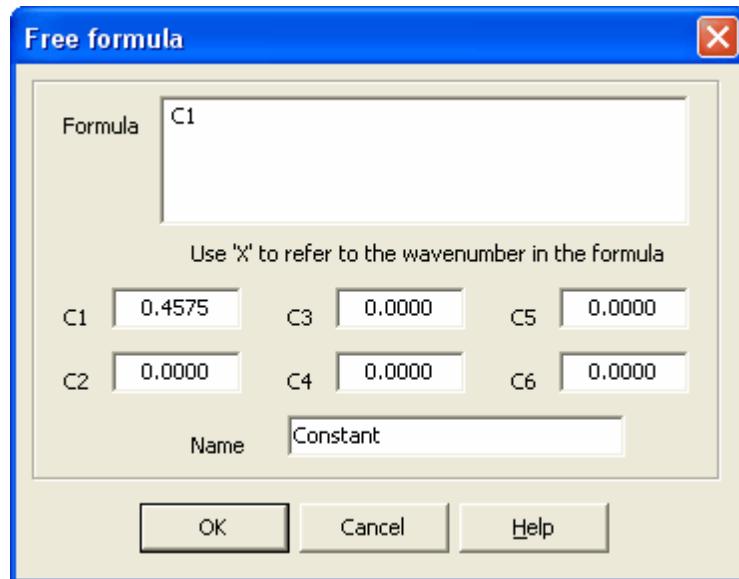
The internal conversion efficiency is specified in a subobject which can be opened by the menu command **'Objects|Internal efficiency'**:



Using the **Range** command you can set the spectral range for the efficiency. The curve is defined by a set of terms which are superimposed to the final curve. The terms are defined in another subwindow which opens using the **Definition** menu command:



The easiest way to generate a function is to use an object of type 'Free formula for imag. part' (Sorry, this name is a little misleading - we have 'borrowed' this object from another application). Using the **Edit** command the settings can be made in the following dialog:

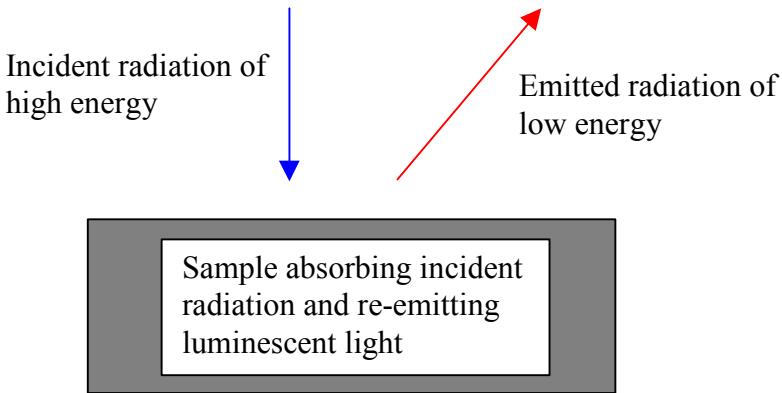


Here you can specify a user-defined formula for the conversion efficiency - in the given example the efficiency is just the constant C1. Note that the constants C1, C2, ..., C6 will show up as possible fit parameters. This allows you to adjust the conversion efficiency to measured data.

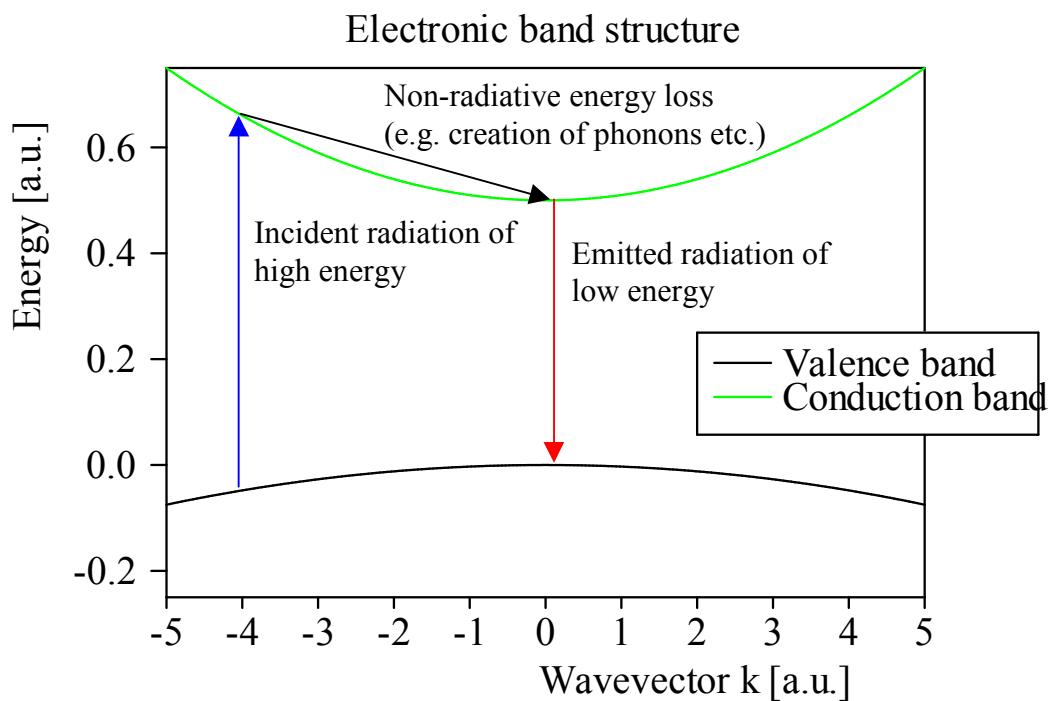
4.4.6 Photoluminescence

4.4.6.1 Introduction

Experimentally observed photoluminescence (PL) or fluorescence spectra are often considered to give a direct view of microscopic electronic transition phenomena. The experimental setup of such kind of experiments is simple - at least in principle:



A schematic view of the photoluminescence process in the sample is this:



From this simple picture simple interpretation rules could be deduced. For example, the energy of the emitted radiation (or the energy of the peak of a PL spectrum) is the gap energy of the material (which can be identified this way). Or: If samples differ in intensity of the emitted radiation the non-radiative energy loss mechanisms are different.

However, usually life is more complicated, and before a microscopic interpretation of the observed luminescence spectrum is tried one should take into account all external phenomena which may influence the shape of the luminescence spectrum. In particular in thin film systems reflectance, interference and re-absorption phenomena can modify the shape of the PL spectrum significantly, and an intuitive interpretation can be quite misleading.

Knowing the optical constants of all materials in a layer stack, the SCOUT software can be used to compute the influence of all these external effects on PL spectra. This way a solid base for the correct interpretation of PL spectra is provided.

The following sections describe

- [the algorithm to compute PL spectra](#)
- [the user interface](#)
- [simple simulated examples](#)

Examples of the **analysis of measured PL spectra** are given in a separate step-by-step PL tutorial (Tutorial 2, example 3).

4.4.6.2 Algorithm

4.4.6.2.1 Overview

The computation of PL spectra is done in the following three steps:

- [Calculation of the depth-profile of the absorbed radiation in the luminescent layer](#): This defines where the incident radiation is absorbed. It is assumed that the radiation is re-emitted at the same position. Hence the absorption depth-profile gives also the distribution of emitting light sources.
- [Multiplication with the microscopic \(internal\) efficiency](#): The so-called internal efficiency is what one would like to get from PL experiments. It gives the spectral distribution of the emission probability if a photon of the incident radiation is absorbed at a certain point in space. This is the wanted information about the microscopic process.
- [Computation of the escape probability for the emitted light](#): Once a photon is re-emitted it has to find its way through and out of the layer stack in order to be detected. The escape probability depends on reflection coefficients and re-absorption probabilities. These are computed using standard techniques to compute wave propagation effects.

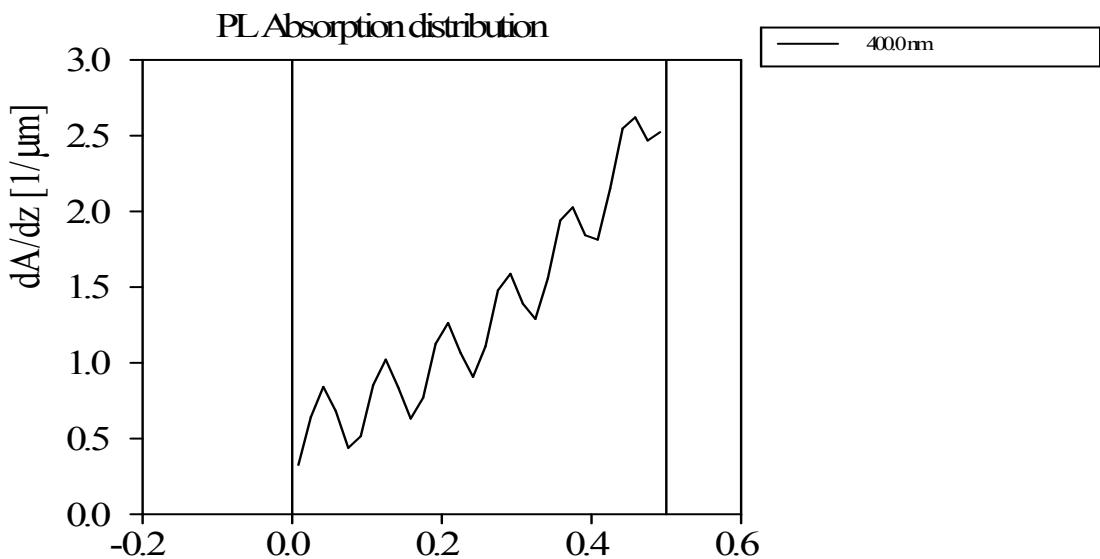
The final result of the computation is the expected PL spectrum. The calculation of the absorption profile and the escape probabilities (steps 1 and 3) is straightforward (more or less). The combination of these is called 'external efficiency' in the following. The internal efficiency (i.e. the microscopic process to convert high energy photons to low energy radiation, see step 2) is not known very precisely in most cases. The recommended strategy to proceed is as follows: Compute a guess for the internal efficiency spectrum by a user-defined function (which could be the superposition of several peaks, for example) and vary parameters of the formula (peak positions, widths, strengths) until the simulated PL spectrum matches the measured curve. Examples are discussed in a separate tutorial.

To demonstrate the algorithm a simple example of one luminescent layer (porous silicon, 500 nm thickness) on a silicon substrate is considered. The optical constants are taken from the SCOUT database without modification.

4.4.6.2.2 Absorption of the incident radiation

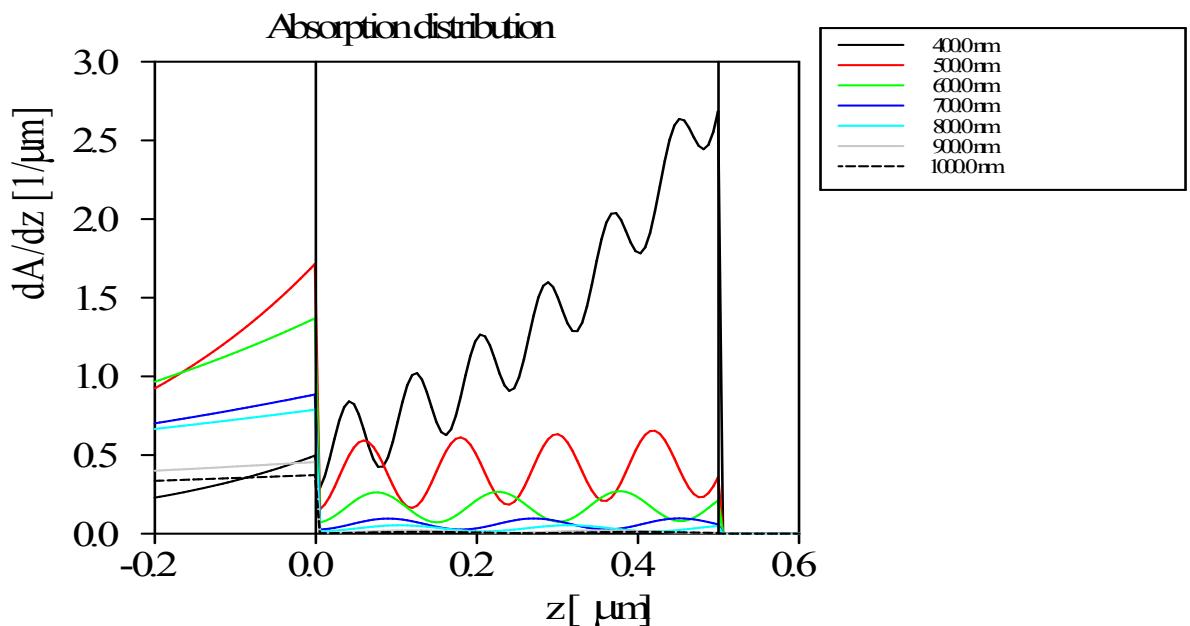
The first step of the PL spectrum computation is the determination of the light source distribution. To do this, the luminescent layer must be specified first. Then the incident radiation is characterized by its wavelength, angle of incidence and polarization. According to these settings, the depth profile of the absorption is calculated.

For the 500 nm porous silicon layer on silicon one gets for 400 nm wavelength (normal incidence of light) the following distribution (using 20 points):



The illumination is incident from the right in this picture. The silicon substrate is left to the point $z=0$. Note the narrow interference pattern.

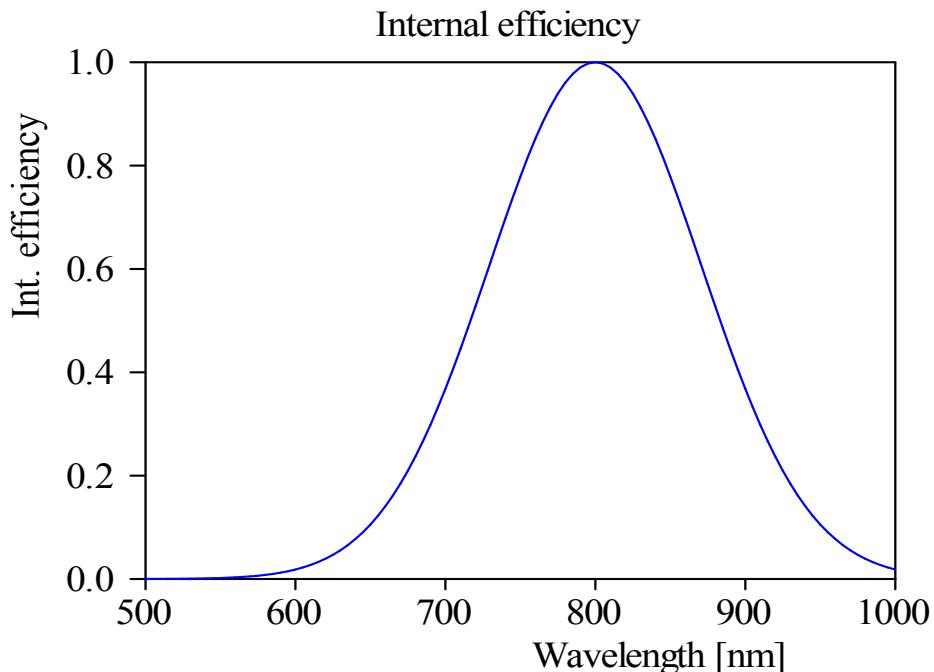
With the SCOUT object 'Absorption distribution' (see the help on distributions) the absorption pattern for higher wavelengths are computed as well. These exhibit even more pronounced interference structures in the local absorption:



Sampling the luminescent layer using 20 points means to work with 20 light sources in the following. Each light source emits proportional to the corresponding absorbed fraction of the incoming beam.

4.4.6.2.3 Internal efficiency

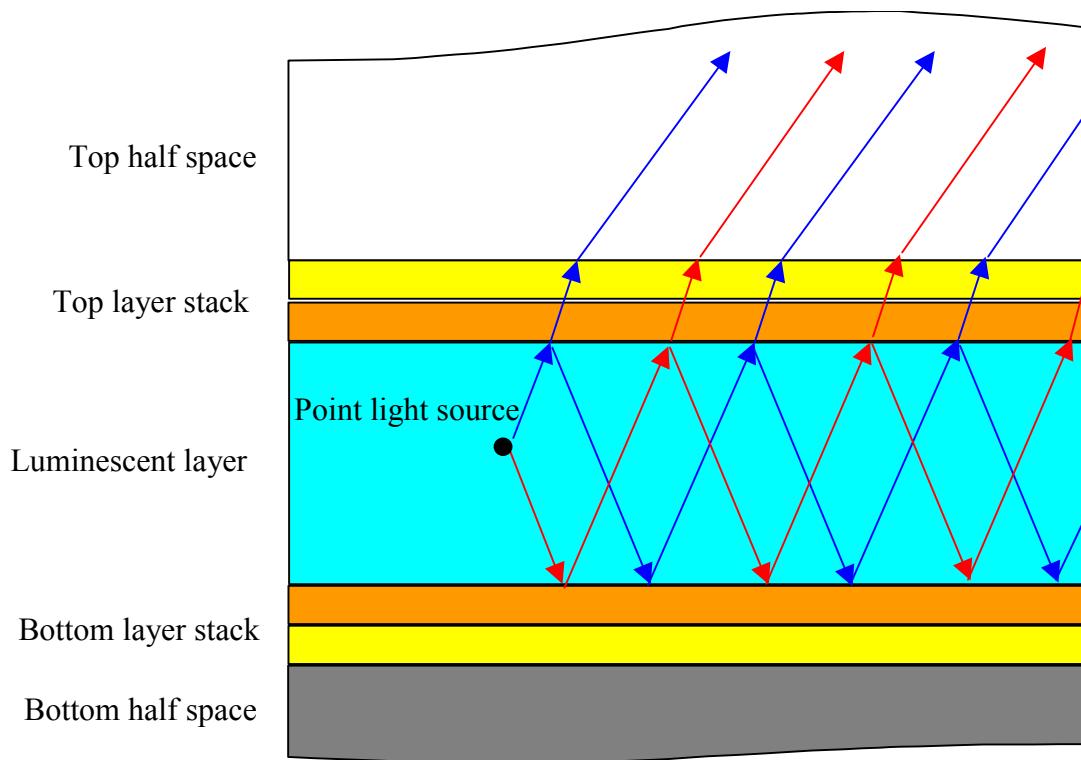
It is assumed that absorbed 'high energy photons' are converted to 'low energy photons' without any displacement in space, i.e. a possible re-emission takes place at the same position as the absorption. Furthermore, a complete isotropic emission with random polarization is assumed. The amount of created low energy photons is determined by the so-called 'internal efficiency' which gives the probability for emission at a certain wavelength. Internal efficiencies are defined by the superposition of oscillator terms, user-defined formulas or imported data sets. Here is an example. In the range 500 to 1000 nm (with 101 spectral points, i.e. 5 nm resolution) the formula $\text{EXP}(-\text{SQR}((X-800)/100))$ defines the following broad peak (note that x refers to the wavelength, in this case):



This means that most of the emitted photons have a wavelength around 800 nm. Almost no radiation at 500 nm is produced internally.

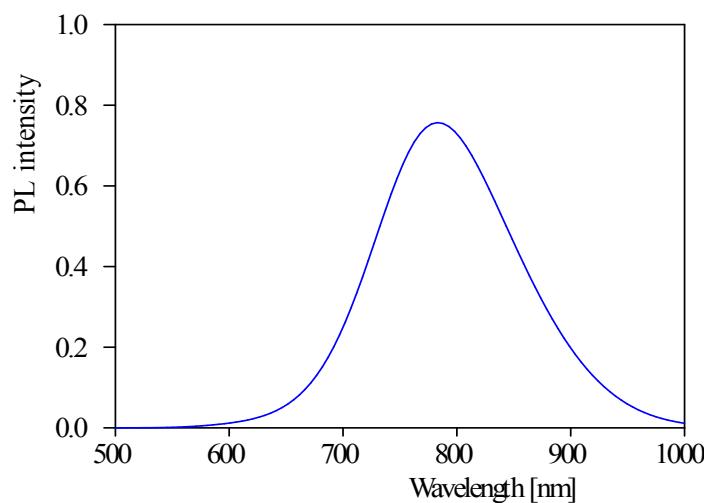
4.4.6.2.4 Emission

Finally the amount of radiation escaping in the direction of observation is computed. To do so, the observation angle and the polarization must be specified. For each light source in the luminescent layer, the waves that end up in the finally observed direction are superimposed (with phase relations) taking into account Fresnels' equations and summing up all multiply reflected partial waves:

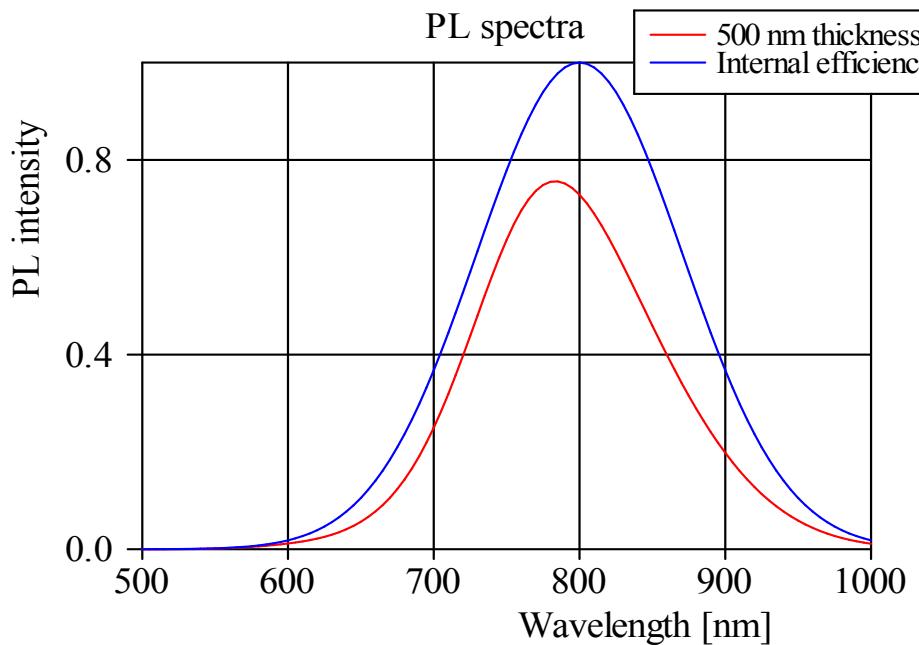


In the graph above, the multiply reflected partial waves within the top layer stack and the bottom layer stack are not drawn for simplicity. They are, however, included in the computation.

Squaring the electric field amplitudes the intensity for each point light source is computed. The contributions of all light sources are added up to the final value of the PL intensity. This is repeated for the whole spectral range (given by the settings of the internal efficiency). The result for the 500 nm porous silicon layer on silicon is this (400 nm excitation):



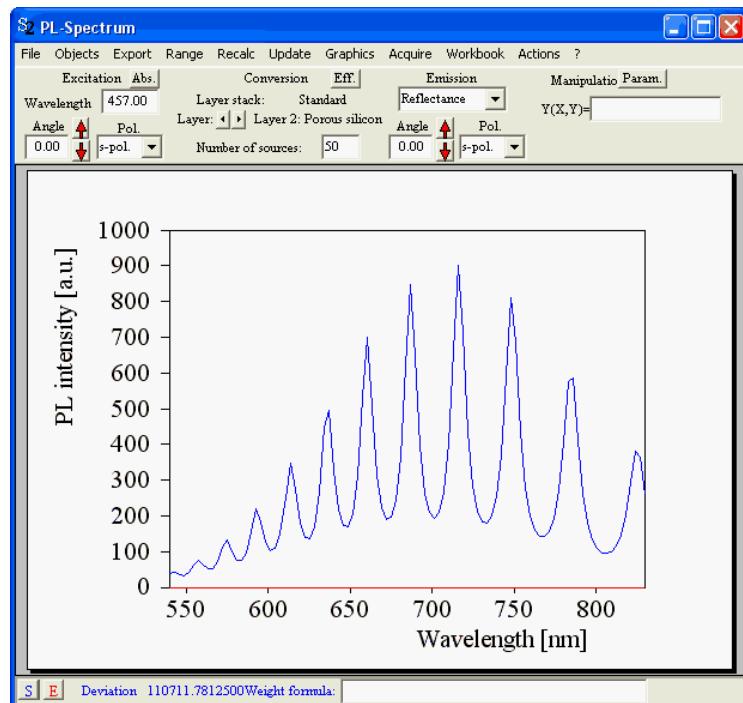
The comparison with the internal efficiency spectrum shows a shift of the emitted PL spectrum:



In contrast to an intuitive inspection of experimental spectra, in this case there is no doubt about the origin of the observed shift: Since the used internal efficiency is known and has a peak at 800 nm, any shift must be due to interference, reflection and re-absorption effects.

4.4.6.3 User interface

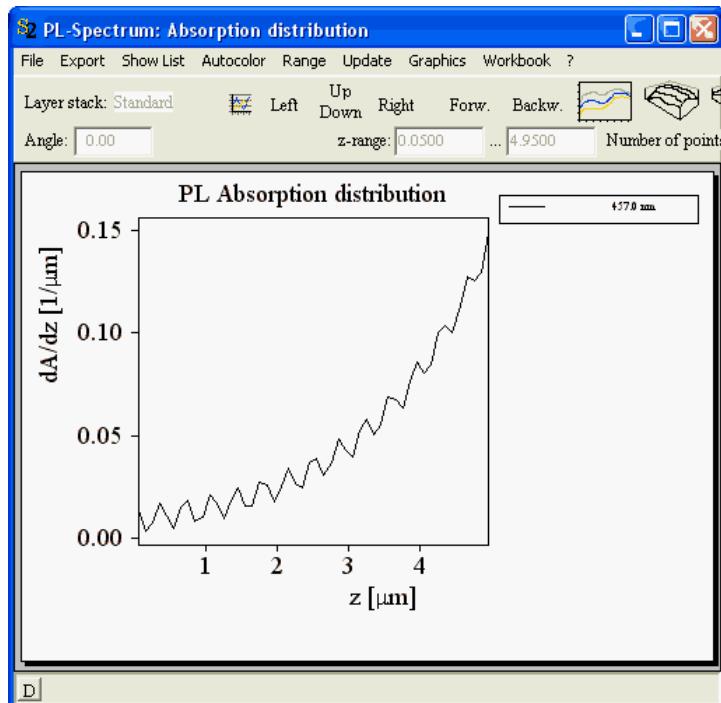
The main window of PL spectrum simulation objects looks like this:



The simulated PL spectrum depends on details of the excitation, the internal conversion and the emission (see the description of the algorithm above).

Excitation

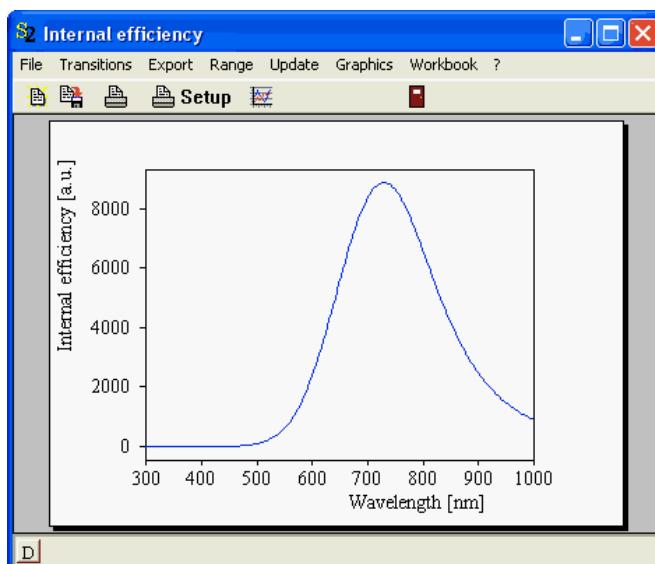
The excitation parameters are set on the upper left side of the window. You have to specify the wavelength of the exciting light (usually a laser), the angle of incidence and the polarization. Based on these data and the current layer stack, SCOUT will compute where and how much of the incident radiation is absorbed in the photoluminescent layer. After the computation of the PL spectrum, you can view the absorption distribution pressing the button labeled **Abs.**:



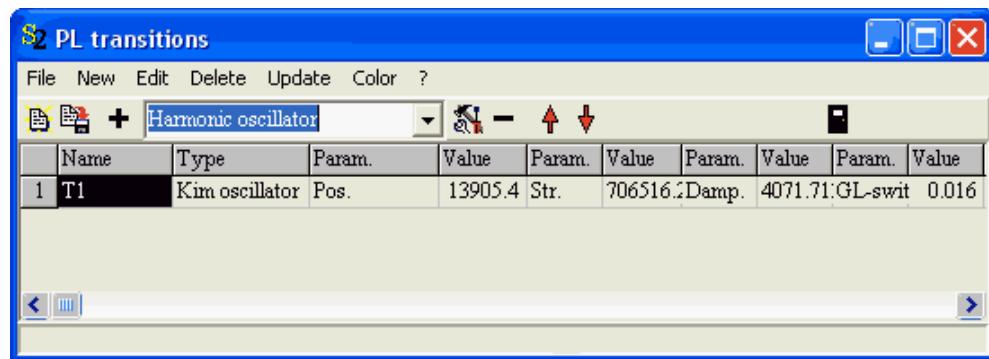
Conversion

In the parameter section called 'Conversion' you can define the layer stack (drag&drop it from the list of layer stacks) and the photoluminescent layer within the stack. Only one layer can be photoluminescent. Setting the parameter called 'Number of sources' you can define how many internal light sources are used within the PL active layer.

Pressing the button **Eff.** the window that defines the internal efficiency is opened:



Use the command **Transitions** to open a list similar to the susceptibility list for optical constant definition:



Use this list to define the internal efficiency spectrum as the superposition of several oscillator terms, user-defined formulas or imported data. The spectrum is computed exactly the same way as the imaginary part of the dielectric function. All parameters of the superimposed susceptibility terms appear as fit parameters that can be adjusted in SCOUT's fitting procedure.

Emission

In the section 'Emission' you have to specify on which side of the layer stack you observe the PL radiation: 'Reflectance' means that the PL light is observed on the side from which the excitation is done (this is the usual case), the selection 'Transmittance' should be made if you observe the emitted light on the other side of the layer stack. In addition, you have to set the angle of observation and the polarization.

4.4.6.4 Qualitative examples

4.4.6.4.1 Overview

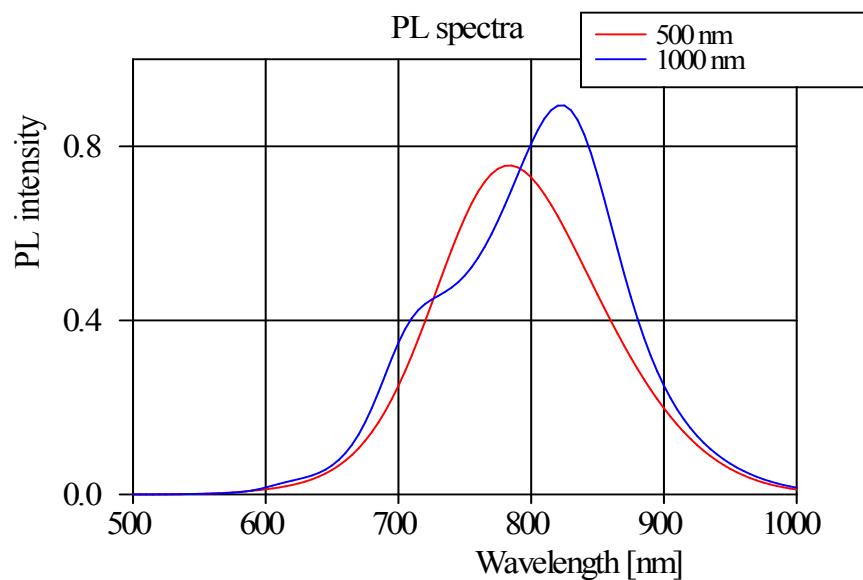
In this section PL spectra of some simple systems are computed in order to develop a feeling for the effects that may occur.

The following examples are discussed:

- [Thickness variations of luminescent layers](#) (porous silicon on silicon)
- [Additional layers](#) above the luminescent layer

4.4.6.4.2 Thickness variations

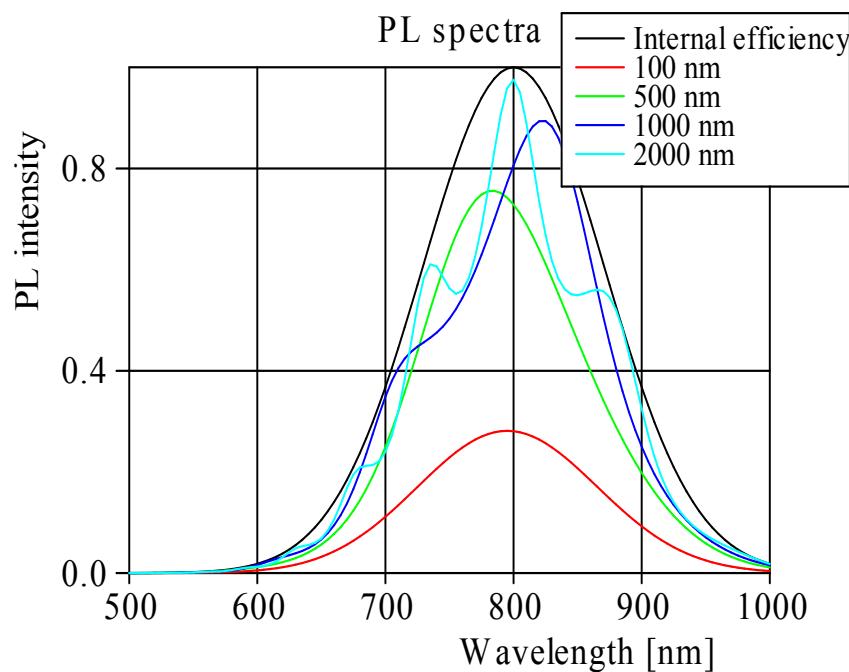
Suppose you prepared two samples consisting of a luminescent layer on a silicon substrate. The thicknesses differ by a factor of 2. The observed PL spectra would be the following:



What would you conclude from this? Did you prepare two very different luminescent systems with two broad, overlapping emission peaks in one case (the blue curve) and a single emission mechanism in the other (red)?

You should be careful with interpretations unless you checked the influence of the layer thickness on the shape of the PL spectrum! In fact, the two curves are once more porous silicon spectra computed with the same internal efficiency ([see above](#)). The layer thickness is the only difference between the two cases.

Here are some more spectra of this kind, including the two presented above:

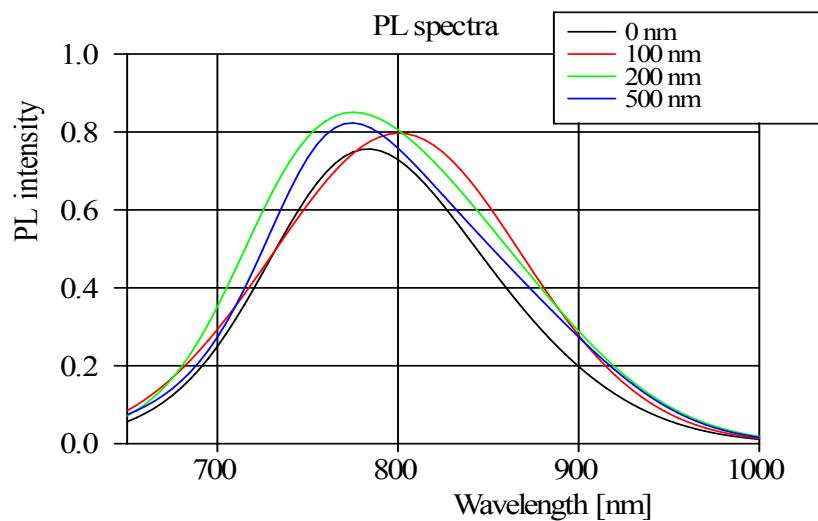


Interpretation problems arise more likely in the case of thin films where the external efficiency variations lead to spectral shifts and modifications which are not easily identified. In thicker layers the rich structure of the interference patterns clearly (more or less) show their origin.

4.4.6.4.3 Additional layers

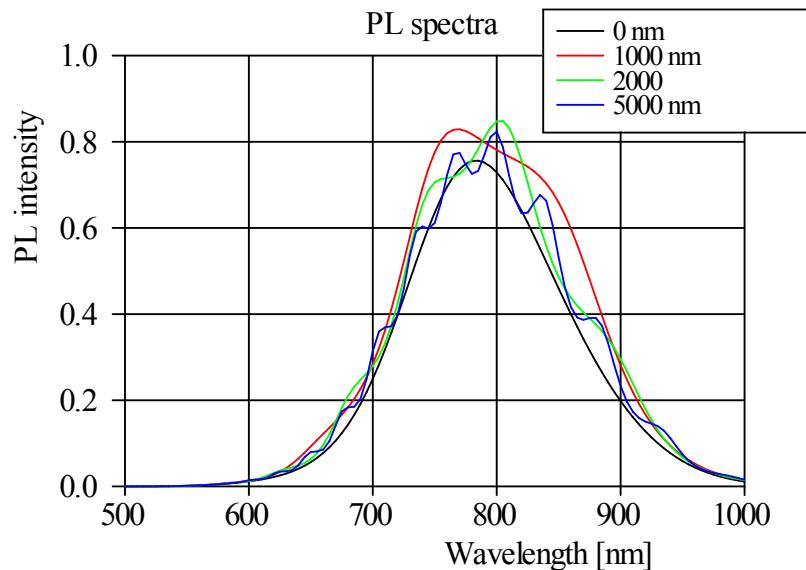
If you have to compare PL spectra of luminescent layers embedded in different layer stacks you should also be careful interpreting the observed phenomena. The influence of additional layers on PL spectra is demonstrated in this section for two simple systems.

The first one is a SiO₂ layer on top of the 500 nm luminescent porous silicon layer. The following plots show PL spectra computed for various SiO₂ layer thicknesses:

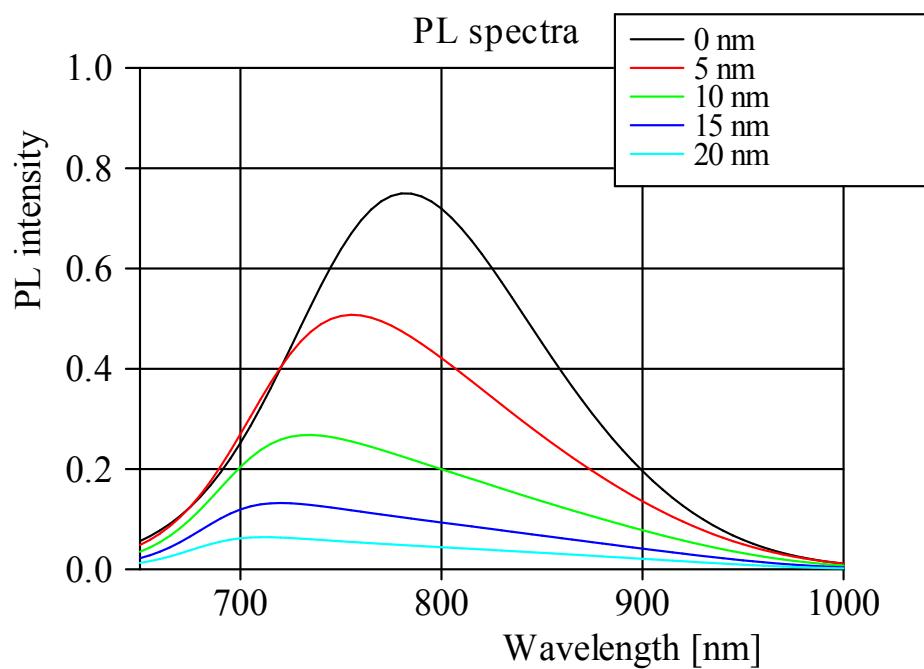


Peak position shifts of 40 nm (0.05 eV) and height differences of up to 25% occur, with identical internal electronic transitions! It is stressed once more that these spectra do not exhibit the origin of

these differences clearly. On the other hand, thick SiO₂ layers show pronounced interference fringes:



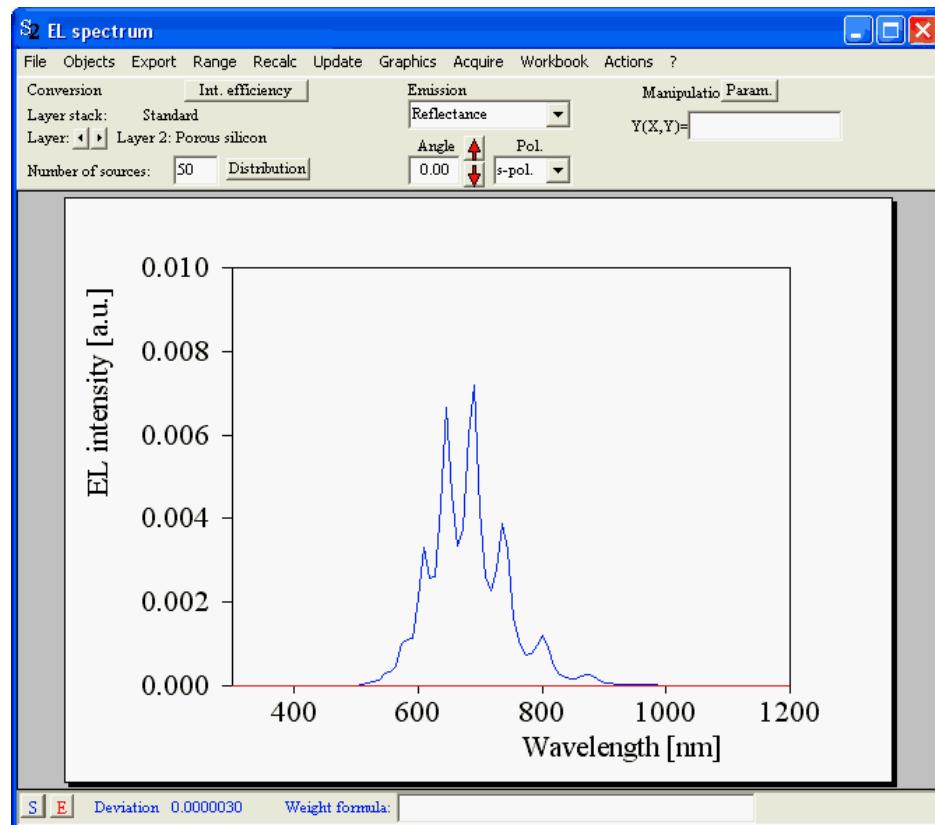
Metallic top layers have a quite different influence on the PL output. Here the case of a silver layer is considered. Like most metals, Ag is highly reflective in the infrared and (again, more or less) transparent in the UV. Hence the exciting high energy photons are transmitted to the underlying luminescent layers but the re-emitted luminescent radiation is blocked efficiently in the infrared and red spectral range. This results in a significant blue shift of the PL spectrum (although in all cases the internal electronic transitions are the same):



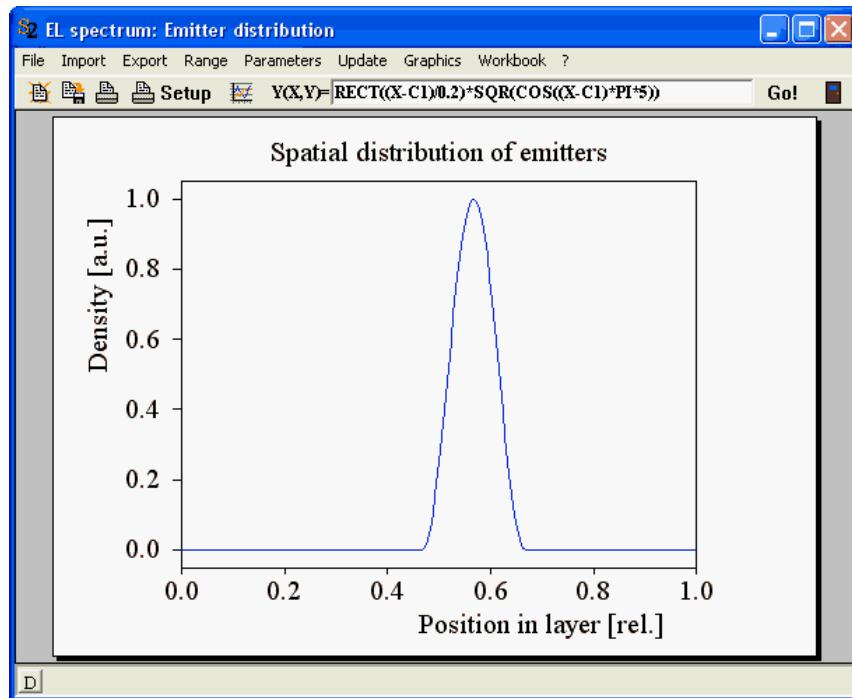
4.4.7 Electroluminescence

4.4.7.1 EL overview

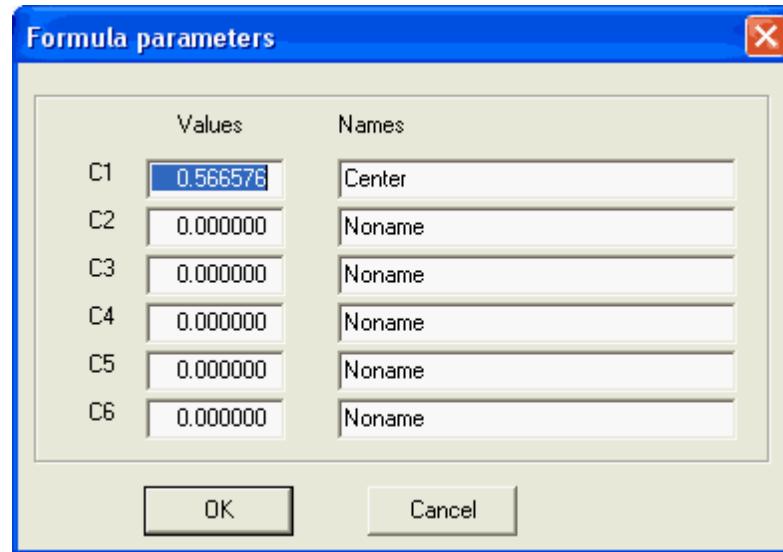
Electroluminescence spectra can be computed in SCOUT using objects of type 'EL spectrum' in the list of spectra. These objects are very similar to photoluminescence objects (please read the section [above](#) about photoluminescence). The main window is the following:



In contrast to photoluminescence objects there is no section about excitation, of course. The button **Distribution** opens a subwindow which defines the spatial distribution of the light emission within the electroluminescent layer:



The x-axis displays the relative position within the active layer: 0 is the bottom, 1 the top of the layer. In the edit box called 'Y(X,Y)' on the top of the window you can enter a formula for the distribution. Press **Go!** to compute the data using the entered expression. Within the formula, you can refer to up to 6 constants (called C1 to C6). Using the **Parameters** command you can assign names to these constants:



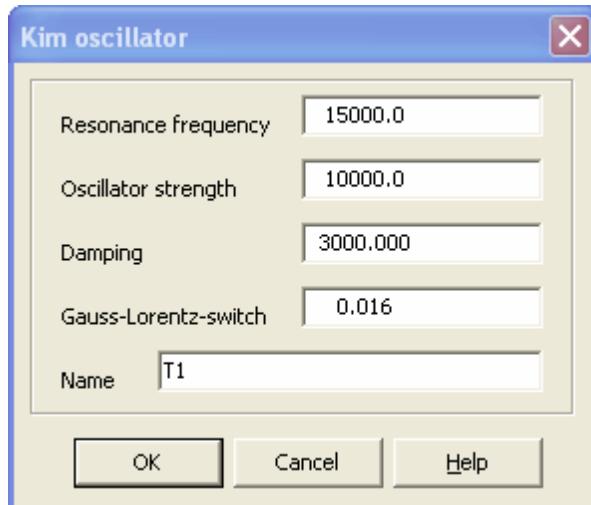
The parameters C1 ... C6 will appear as fit parameters in the list of fit parameters.

Alternatively to the computation using a formula, you can import a fixed distribution of light emitters from the workbook or using the **Import** command.

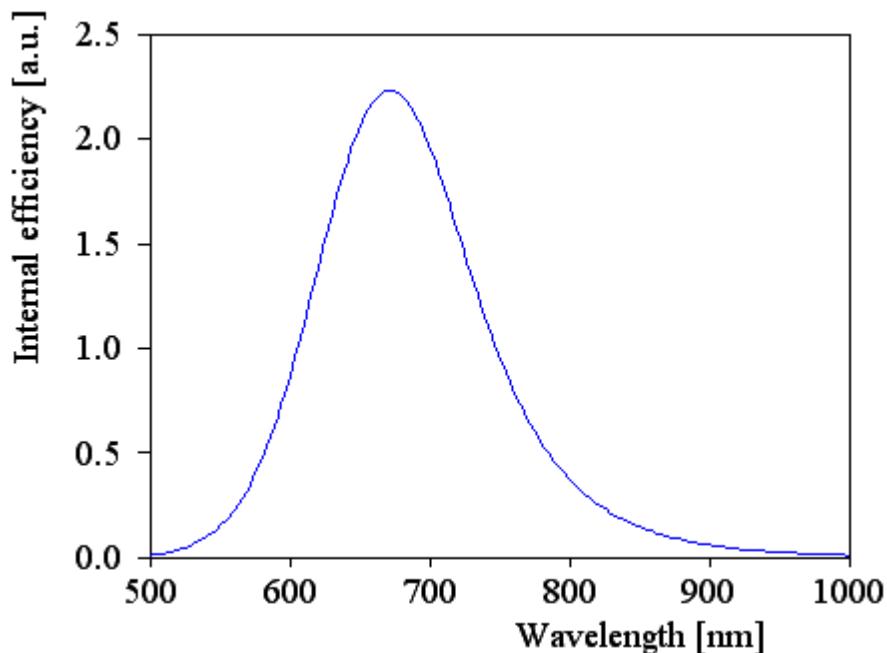
4.4.7.2 EL example

Here is a simple EL example. The emitting layer is a porous silicon layer on a silicon wafer, covered with a thin gold film.

The internal emission spectrum is modeled by a single Kim oscillator with the following parameters:



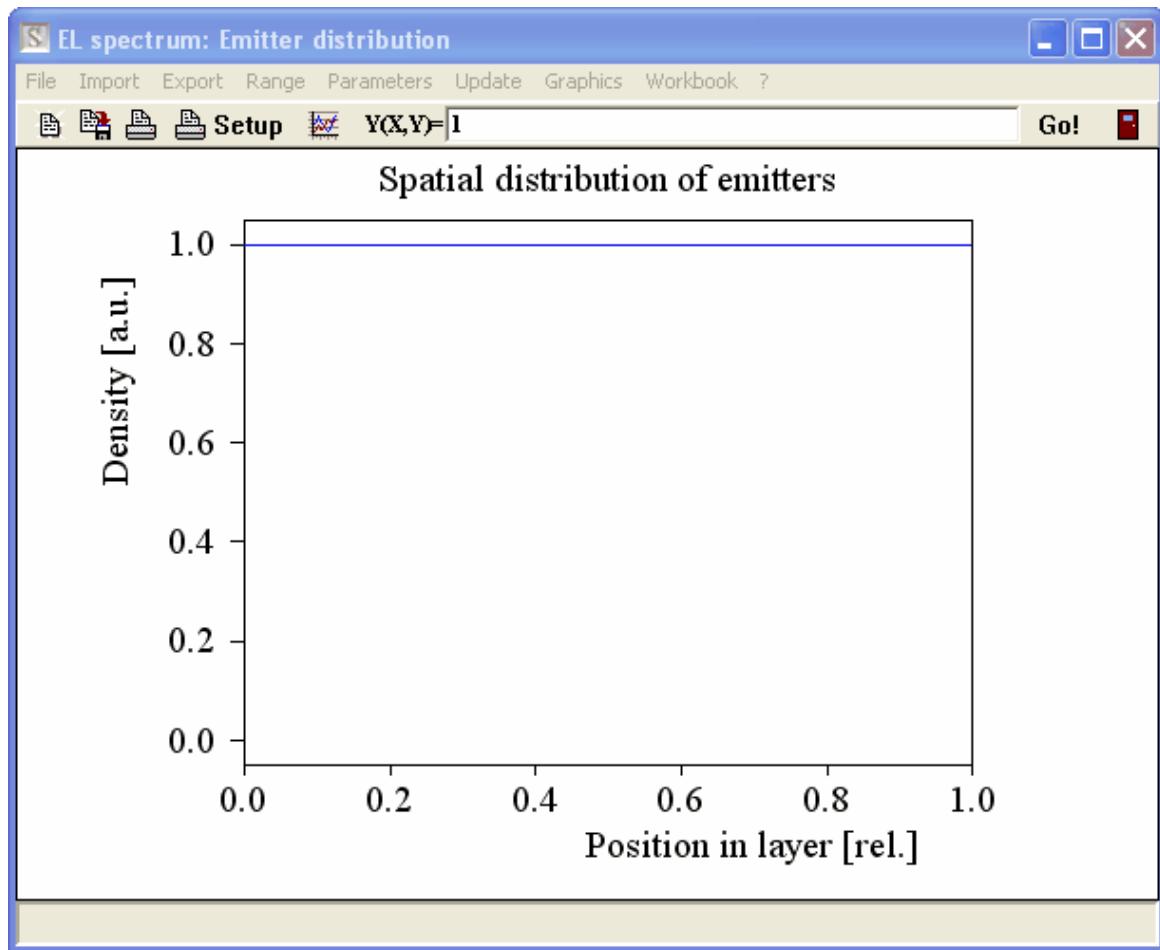
This leads to the following spectrum:



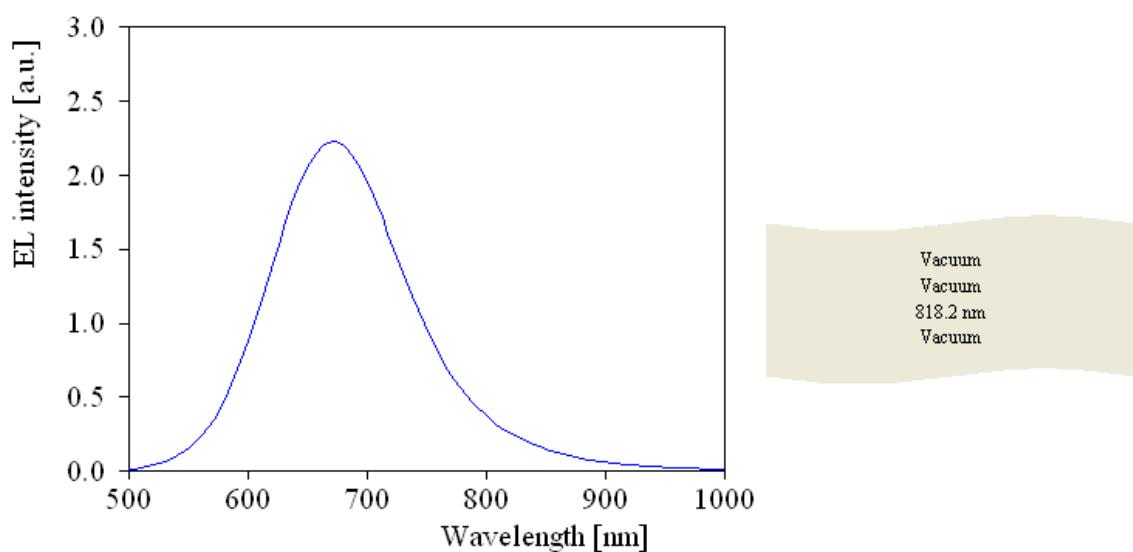
In order to make clear how the EL computation in SCOUT works, we will build up the stack step by step. Let's start with a single light-emitting thin film in between two halfspaces. All materials are vacuum - this is not really realistic, of course, but we can see the pure emission without any reflectance effects at the interfaces between the layers.

The depth distribution of the light emission within the EL active layer is sampled with 50 points - i.e. there are 50 light sources in the emitting layer, placed at equidistant positions. The light sources can have a depth-dependent strength of emission which is taken from a user-defined depth distribution. Click on the Distribution button to see this function. Let's start with the simple assumption that the light emission is distributed equally within the active layer, i.e. all points

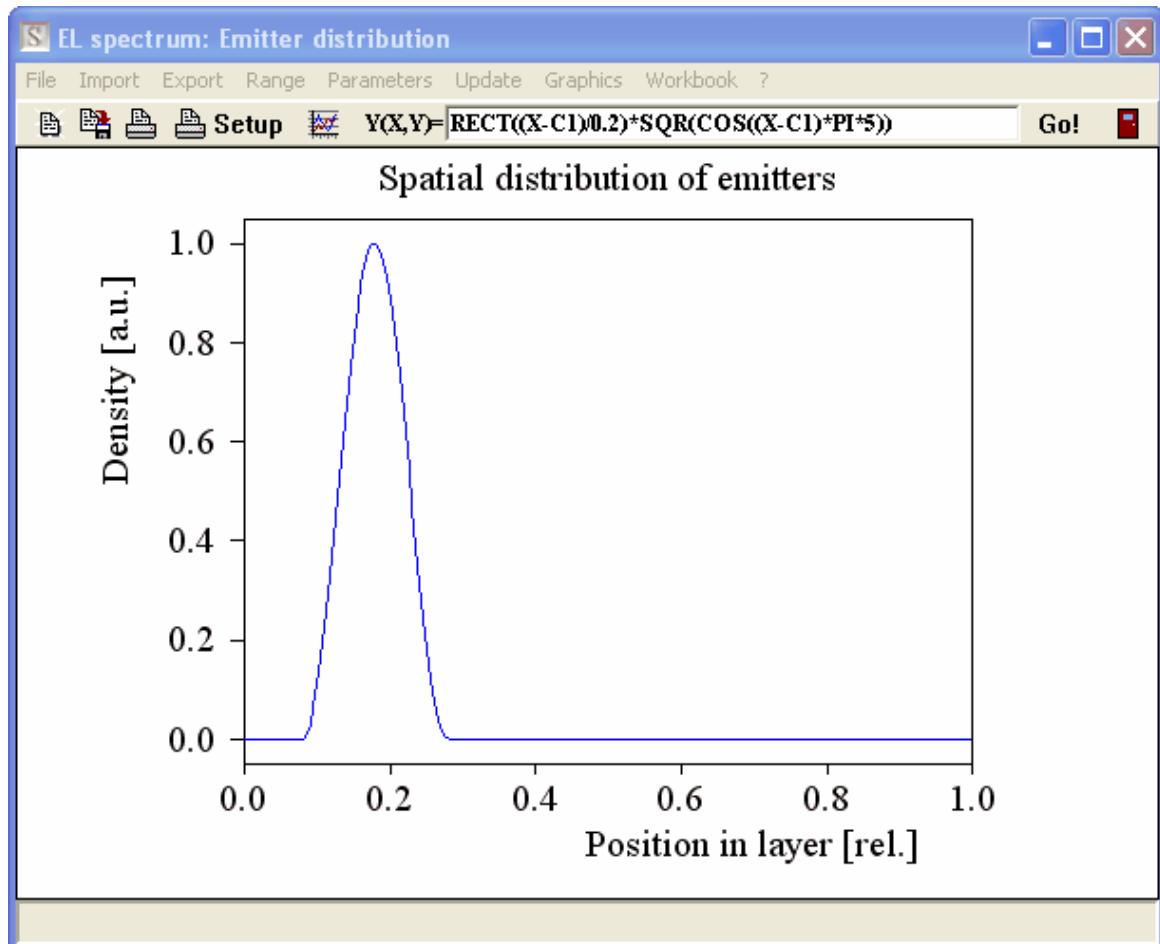
contribute with weight 1:



The EL spectrum computation now takes all the 50 points, sums up the output for each one and divides by the number of points. In the present case the outcome on the reflectance side of the sample is the original internal emission spectrum:

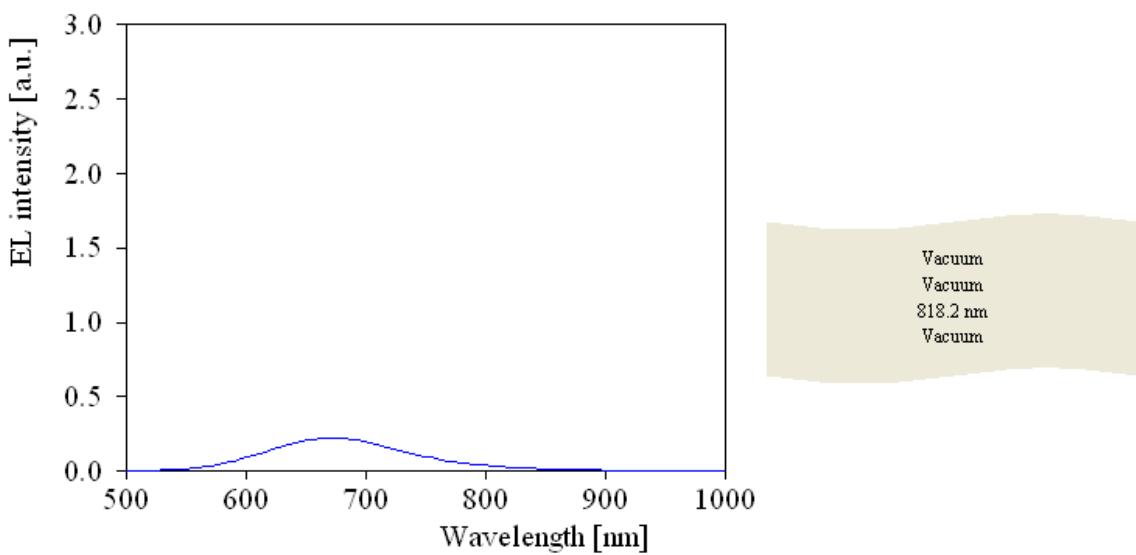


We assume now that the EL emission is not homogeneously distributed within the layer stack but has the following depth profile:



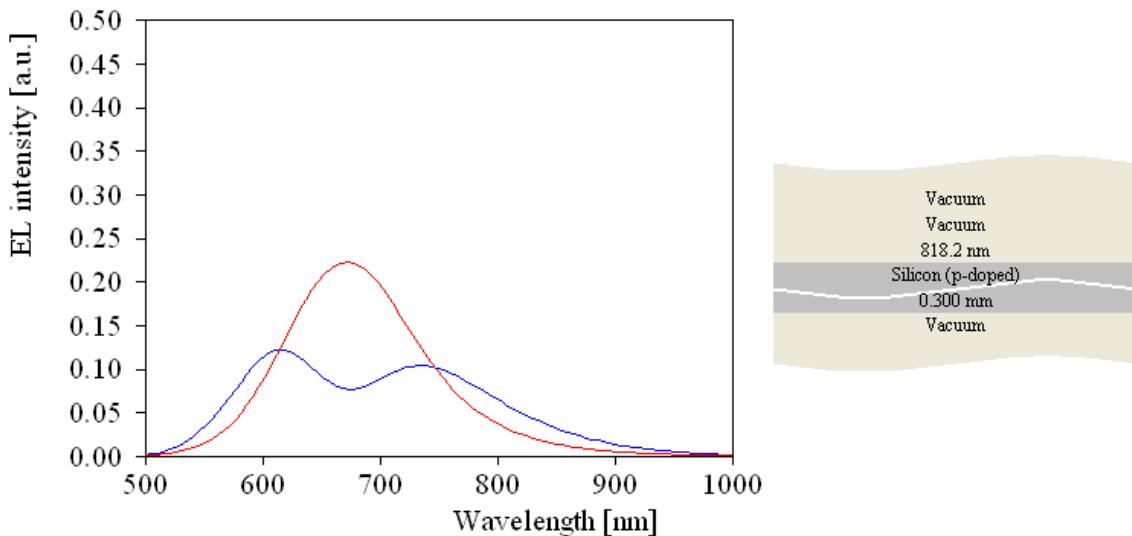
The parameter C1 in the formula allows to move the peak from the top to the bottom of the layer. You can do that, for example, with a slider (horizontal or vertical bar object) in a view. The parameters C1 to C6 of the user-defined distribution formula will appear as possible fit parameters in SCOUT.

The emitted spectrum is now the following:

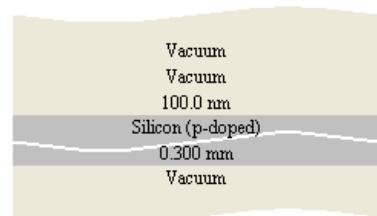
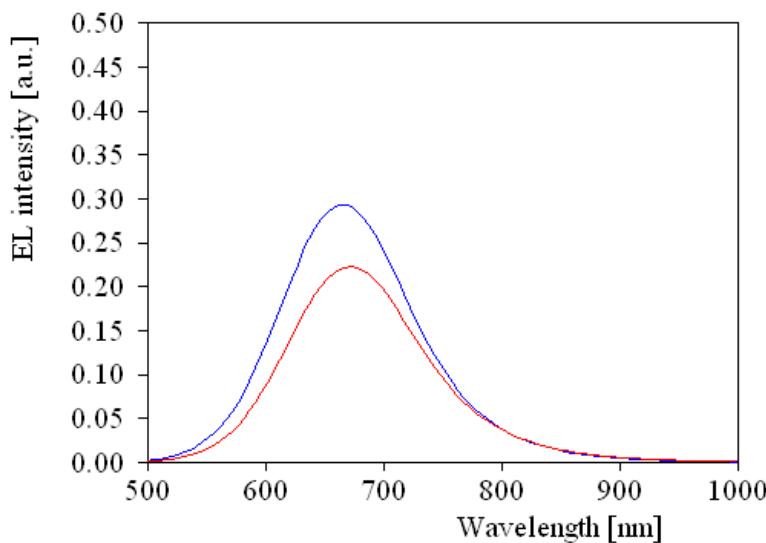


The average emission probability of the 50 points is now much less than 1 - hence the emission is much lower now.

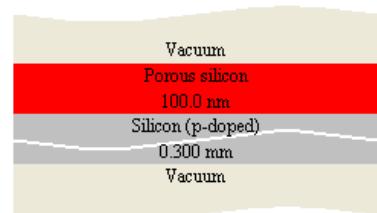
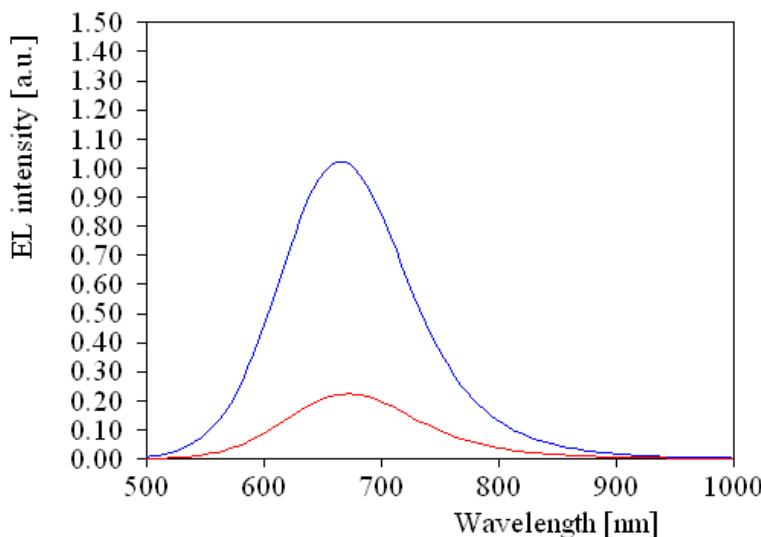
Note that the EL emission into a certain observation direction is computed adding up two contributions: One is the internal emission that aims directly into the wanted external direction (taking into account the change of direction due to refraction at the interfaces). The second one is the emission into the direction which is reflected into the wanted direction by the backside of the emitting layer. This second contribution should contribute to the spectrum if we place a silicon wafer below the emitting layer. The next graph compares the previous spectrum (vacuum backside, red curve) with the one obtained with a silicon wafer substrate (blue):



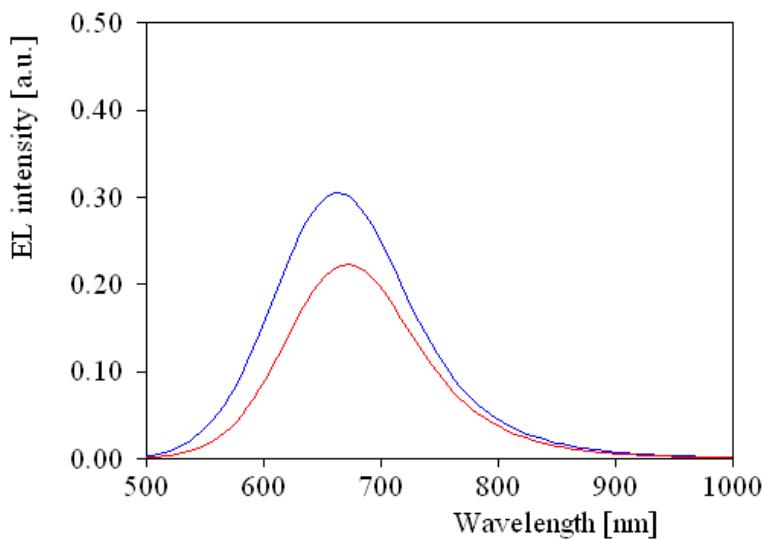
Well, in some spectral regions the backside reflected contribution leads really to an increase in EL emission, but in the main peak there is less light observed with silicon substrate. Assuming that the light emission is coherent in all directions, SCOUT superimposes the two contributions discussed above fully coherent. This leads to interference patterns in the spectra which strongly depend on the thickness of the EL emitting layer. Here is the result for a thinner emitting layer:



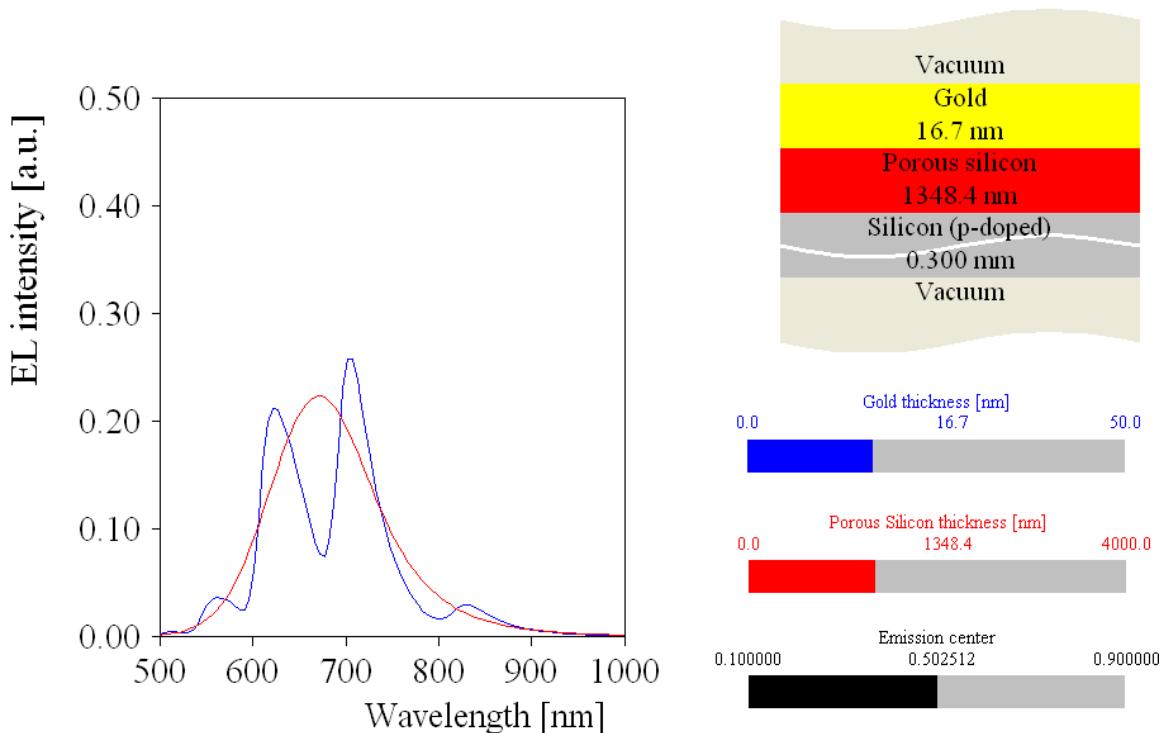
We can now replace the material 'Vacuum' of the active layer by porous silicon. The emission spectrum changes to the following:



Finally we add the gold layer on top which reduces the emission significantly:

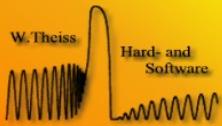


Placing some sliders in the main view gives you some freedom to manually change parameters:



Note that interference effects strongly modify the shape of the EL spectrum. The observed peaks cannot be directly interpreted as internal emission channels.

SCOUT



Part

V

5 Parameter fitting

5.1 Parameter fit overview

After the definition of the optical model (i.e. [materials](#), the [layer stacks](#) and the [spectra](#) including the import of the experimental data) the parameters that are going to be varied in order to optimize the agreement of model and measured data must be selected from all model parameters. These chosen parameters are called the fit parameters. The agreement of model and measured data is given by the so-called [fit deviation](#). Based on the fit deviation you can classify the agreement between model and experiment. This [rating](#) is described below.

Fit parameters are managed in the [list of fit parameters](#) (opened by a right-click of the corresponding treeview item).

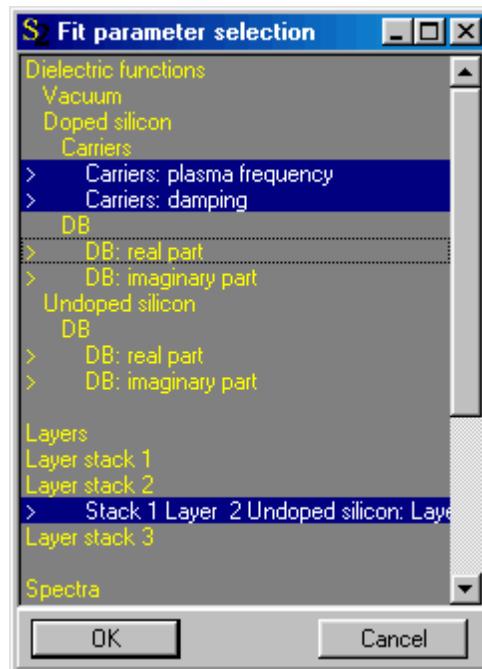
In order to inspect the influence of a parameter on the computed spectra the actions called [Parameter variation](#) and [Parameter fluctuation](#) are very useful. These are described in the section about the SCOUT [workbook](#).

5.2 Fit parameters, sliders and 'fit on a grid'

In the main window of SCOUT, switch with F7 to the treeview level and right-click the branch **Fit parameters** to open the list of fit parameters. At program start the list is empty. New parameters can be added using a dialog or by drag&drop from other lists of SCOUT.

Selecting fit parameters using the [Fit parameter selection](#) dialog

The **New** command (or the button labeled '+') of the fit parameter list opens a dialog with a list of all possible fit parameters from which you have to select. Dielectric function parameters (like resonance frequencies of oscillators or volume fractions of effective medium models), layer parameters such as thicknesses, fit parameters of the computed optical spectra (angle of incidence or constants that appear in modification formula) and [master parameters](#) are listed as follows:



As indicated, you can select many parameters simultaneously by pressing the Ctrl-key while

clicking on the various lines. Clicking an already selected line 'unselects' it. Only those lines beginning with '>' represent fit parameters. The other lines contain guiding information only.

Selecting fit parameters by **drag&drop**

Instead of using the fit parameter selection dialog, you can also select fit parameters by drag and drop operations. You have the following possibilities:

- Drag a dielectric function from the dielectric function to the fit parameter list: All the parameters of the dielectric function are added as fit parameters.
- Drag from a susceptibility of a dielectric function model to the fit parameter list: All the susceptibility parameters are added as fit parameters. If you start the drag operation in a column that represents a value of a parameter (these columns are labeled as 'value') only that particular fit parameter is added.
- Drag a layer from the layer stack to the fit parameter list: All the parameters of that layer are added.
- Drag a spectrum from the spectrum list to the fit parameter list: All the parameters of that spectrum are added.

The fit parameter list

The chosen parameters are added at the end of the fit parameter list. This may look like this:

	Value	Name	Variation	Low limit	High limit	Factor	Digits
1	1.4736	Stack 1 Layer 2 Undoped silicon: Layer thickness	Grid (100 points)	0.0500	2.0000	1.0000	4
2	0.980853	IR Reflectance Factor	Downhill simplex	0.950000	1.050000	1.0000	6
3	1283.6980	Doped silicon:Carriers: plasma frequency	Downhill simplex	0.0000	0.0000	1.0000	4
4	637.0527	Doped silicon:Carriers: damping	Downhill simplex	0.0000	0.0000	1.0000	4

The **name** column gives the fit parameter's name. Since the names of fit parameters assigned automatically by SCOUT are sometimes quite lengthy you can overwrite the current name with your own. Just type in the new name. However, be sure to remember which parameter it really is. At present, there is no way to get back to the automatically assigned name. In case you forgot what the parameter means, you have to delete it and create it new.

The **Value** column displays the current value of the parameter. If you would like to set bounds for the fit parameter value you can set an interval of allowed values using the **Low limit** and **High limit** values.

As **Factor** you can enter a number which is used to multiply the original value of the parameter: You can use this factor to scale up or down a parameter. If, for example, a thickness is given in micrometer you can multiply it by 1000 in order to see the value of the thickness in nm. If you change the factor, the current values of **Low limit** and **High limit** are multiplied by the factor as well.

Finally **Digits** specifies the number decimals used to display the current value of the parameter.

The **Variation** column allows to switch between different optimization options. In order to change the current option place the cursor on the cell and press the F5 key on your keyboard several times. By this action, you cycle through the available options. Stop where you want to stop. Pressing F4 you cycle backwards. At present, the following options are implemented in SCOUT:

Downhill simplex

The parameter is optimized in the automatic fit according to the [downhill simplex](#) scheme which is used at present as optimization technique in SCOUT.

Frozen

The parameter is not varied at all. This can be

used, if you want to keep the parameter's value fixed for the moment but you may need it soon again. In this case, you don't delete the parameter but set its variation option to 'Frozen'.

Grid (100 points)

If the variation option of a parameter is set to 'Grid (x points)' SCOUT will try x values of the parameter equally distributed in the range between **Low limit** and **High limit** before the automatic fit applying the downhill method is performed. For each value the model is re-computed, and the parameter value with the lowest deviation (i.e. difference between model and measurement) is taken as starting value for the automatic fit.

If several fit parameters are to be varied on a grid of test values, the option 'Multidimensional grid fit' (please check or un-check this option using the menu command 'File|Options|Fit|Multidimensional grid fit' in the main window) determines how the computations are performed.

Multidimensional grid fit switched off:

SCOUT will do one-dimensional grid variations in the sequence given by the fit parameter list, starting at the top. Note, that the result for the parameters' starting values will depend on the sequence. Hence, the choice of the grid variation must be done carefully.

Multidimensional grid fit switched on:

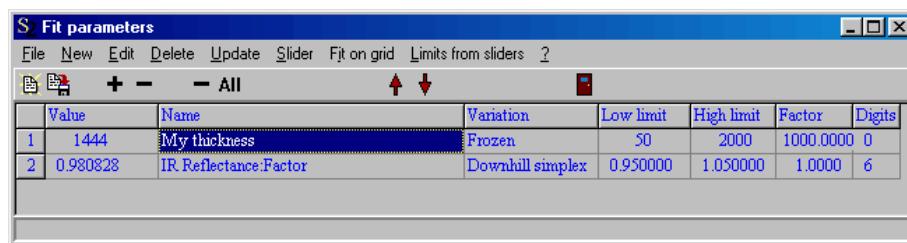
SCOUT will collect all fit parameters for which a grid variation is set in the variation column, and do a multidimensional grid fit taking into account all possible combinations of parameter values. Note that the amount of computational work can be huge: If you select 4 parameters with 100 grid values for each of them, there are 100000000 re-computations of the optical model to be done.

With multidimensional grid fit switched on, you can systematically scan a parameter space in several dimensions and hope to find the absolute minimum of the fit deviation. Setting the option 'File|Options|Fit|Export multidimensional grid fit data' you can log the fit deviation. For each point which is tried in the grid fit procedure the parameter values and the corresponding fit deviation is written to a text file. The filename is specified when you start the fit. The content of the text file can later be used to visualize the 'parameter landscape' which might be helpful in finding promising parameter

values.

Grid (50 points)	see above
Grid (20 points)	see above
Grid (10 points)	see above
Empty cell	Here you can enter a formula to compute the value of the fit parameter. In the formula, you can refer to the values of the master parameters (see below). This way you can introduce relations between parameters, or control several parameters of the model by one master parameter. If you know, for example, how the parameters of your optical constant models depend on temperature, you could introduce the temperature as master parameter and update the optical constants while the temperature is fitted.

The following example shows a frozen fit parameter and a factor different from 1.0:



Set fit parameters manually

You can do a 'manual fit' by typing new values for the fit parameters and selecting **Update** in the menu. This activates a new computation of the dielectric functions and all computed spectra as well as an update of all windows. This way you can immediately inspect how your changes influence the results.

Visual parameter variation: Sliders

A much more elegant way to visualize the parameter influence on the spectra is the use of sliders. Select a fit parameter in the list and click on the **Slider** menu item. This way a new window is created and displayed on the screen that allows you to 'move' a parameter value with the mouse and see instant updates of the model spectra:



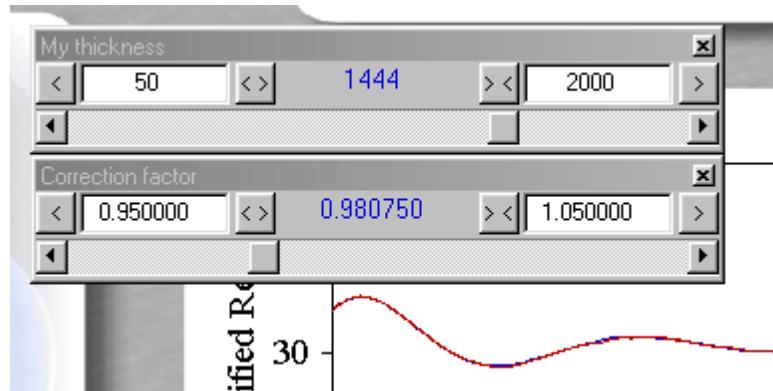
In the top of the window there are three values: The left and the right define the interval of the slider, the center one (in blue) is the actual value of the fit parameter. When the slider window is created the first time the slider range is the fit parameter interval **Low limit ... High limit** if you defined one. If you did not set a fixed interval for a fit parameter the initial range of the slider is 80 ... 120% of the current parameter value.

At any time you can overwrite the slider range by typing in new values for the left and right limit. If the slider position reaches the left limit, for example, you can also press the arrow labeled '<' to the

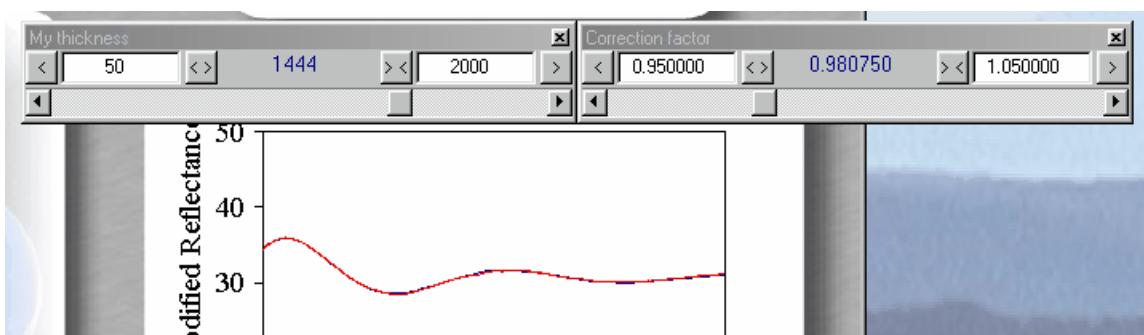
left. This centers the limits around the actual parameter value. This operation can be used to work in a large parameter space without switching between mouse and keyboard. It works in the analogous way on the right side of the parameter range.

Pressing the button labeled '<>' expands the current parameter range by a factor of 2 and centers around the current value. The '><' button reduces the range to 50% and centers around the current parameter value.

In addition to the **Slider** menu command you can also click on the fit parameter with the right mouse button to create the corresponding slider. You can open as many slider windows as you like if you want to investigate many parameters at a time. You can create regular blocks of slider windows by **Drag&Drop**: Create the first slider the regular way (as described above). To place a second slider window below the first one, 'drag' a fit parameter from the fit parameter list to the already existing slider and release the mouse cursor in the left half of the slider window. A new slider is created and placed directly below the first one:



If you release the mouse button in the right half of an existing slider the new slider window is created to the right of the existing one:



Special slider features:

- You can create a slider by dragging a fit parameter from the fit parameter list to a spectrum simulation window. In this case the upper left corner of the new slider window is at the mouse cursor position.
- You can drag any susceptibility parameter that appears in a susceptibility list of a dielectric function window to a spectrum simulation window. The parameter is added as fit parameter to the fit parameter list and a new slider is created at the cursor position.
- You can transfer the fit parameter ranges that you specified in the sliders to the fit parameter list by the menu command '**Limits from sliders**'.

Fit one parameter on a grid

The command **Fit on grid** can be used to optimize one of the fit parameters on a grid of test values. To do so, select the fit parameter that you want to optimize in the list. Specify a parameter range by setting the low limit and the high limit for this fit parameter appropriately. Then apply one of the commands **Fit on grid|with 10 points**, **Fit on grid|with 20 points**, **Fit on grid|with 50 points** or

Fit on grid|with 100 points. Now SCOUT scans the parameter range using the given number of points and returns the best value of the fit parameter on that grid.

The command **Fit on grid|Do all grid fits** will start the grid optimization taking into account all fit parameters for which a grid fit has been selected in the variation column.

This 'fit on a grid' is good for a rough parameter pre-selection, for example for finding film thicknesses in the case of interference patterns. Afterwards you should use the automatic fit for further optimization of the model.

5.3 Master parameters

With the command **Objects|Master parameters** (in the main window of SCOUT) you can open the list of so-called **master parameters** which collects parameters that can be used to control other parameters of the model. Master parameters have a name, a value and a number of decimals. The following restrictions apply to the name: It must not contain any spaces and its length must be below 30 characters. As all other parameters, master parameters can be [fit parameters](#). Just drag them to the fit parameter list or select them in the 'Fit parameter selection' dialog.

Any fit parameter in the fit parameter list can be computed from the values of the master parameters. In this case such a parameter is called **slave parameter**. To turn a fit parameter into a slave parameter move the cursor to the column called 'Variation' which usually shows the 'Downhill simplex' entry. Now press **F4** and see that the cell content changes between 'Downhill simplex', 'Frozen', various 'Grid' items and an empty content. Select the empty state and type in a formula which may contain the names of the master parameters. This way the parameter is computed from the master parameter values. It is not varied directly in the fit routine, and you cannot use a slider to change its value.

Applying master and slave parameters you can describe a coupling between model parameters. If, for example, the damping constant of a [Drude model](#) for free electrons in a thin metallic film depends on the film thickness, you can define a master parameter called 'Thickness'. This master parameter can control both the damping constant of the Drude model and the film thickness by appropriate expressions in the list of fit parameters.

5.4 Fit parameter sets

In the main window of SCOUT you can load and store so-called 'fit parameter sets' which can be very useful if you have to switch spectral ranges and selections of fit parameters quite often. Fit parameter sets store the following:

- spectral ranges and graphics parameters of all model dielectric functions
- spectral ranges and graphics parameters of all simulated spectra
- selection of fit parameters (including their variation options and ranges)

Note that actual values of fit parameters are not stored which means that their current values are preserved when you change fit parameter sets.

The situation for which the application of fit parameter sets is recommended is the following: You have to do the optical analysis for several samples. Some of the model parameters (e.g. oscillator parameters) have influence on the spectra only in limited spectral ranges. You can work with groups of parameters: Some are adjusted first in a certain spectral range, then others are optimized in a second step (probably in a different spectral range), and so on. Finally, you may want to optimize all relevant parameters in a large spectral range covering all the 'subranges'.

This procedure requires the execution of a lot of **Range** dialogs, fit parameter selections and deletions. If you use master and slave parameters, you would have to type in the required formulas for each sample again.

Here you can save a lot of typing work using fit parameter sets: As soon as you have set the spectral ranges and fit parameters for the first group of parameters and the first set of

measurements, you should save the settings with the command **Fit parameter set|Store** in the main window. When you need the same fit parameter selection the next time (for the second sample), you just recall it by **Fit parameter set|Load**.

Fit parameter sets can be loaded and used in OLE automation applications. Examples will be discussed in one of the tutorials.

In combination with automatic parameter fitting you can automatically apply sequences of fit parameter sets. See the section about [automatic fitting](#).

5.5 Automatic fitting

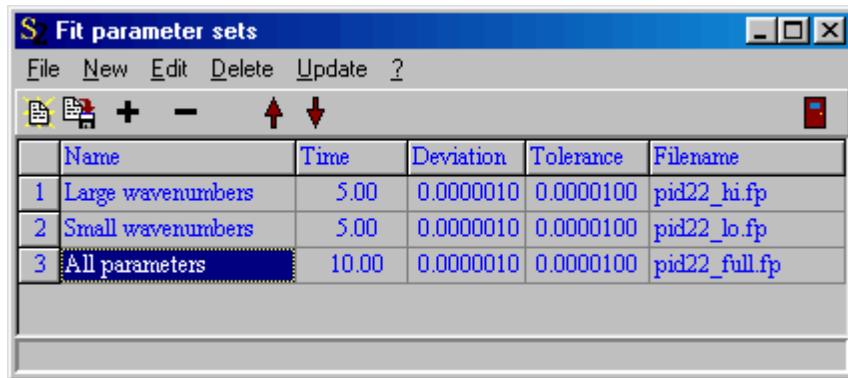
After you have found reasonable start values for the fit parameters by manual or visual variation you can activate the automatic fit by **Start** in the main window. The automatic parameter variation uses the methods described below to optimize the fit parameters, i.e. to minimize the [deviation](#). Whenever the model improves you'll get an update of the status bar at the bottom of the main window. If you have defined a view in the main window it will be re-drawn based on the improved model. In addition, all open dielectric function and spectrum windows are updated to keep you informed about the progress of the model.

With **Stop** you stop the fit activity. It is also possible that the fit algorithm automatically decides to stop the fit. This occurs if there seems to be no progress any more.

Alternatively to the **Start** and **Stop** menu commands you can use the Start button in the toolbar. This must be used if the menu has been hidden. After the fit has been started the button turns into a **Stop** button which can be used for manually stopping the fit activity.

List of fit parameter sets

Quite advanced fitting algorithms are possible using a sequence of fit parameter sets. These can be defined in the list of fit parameters sets which is opened by right-clicking the wanted element in the treeview branch **Fit parameters sets** in the main window. A list may look like this:



Name	Time	Deviation	Tolerance	Filename
1 Large wavenumbers	5.00	0.0000010	0.0000100	pid22_hi.fp
2 Small wavenumbers	5.00	0.0000010	0.0000100	pid22_lo.fp
3 All parameters	10.00	0.0000010	0.0000100	pid22_full.fp

Each fit parameter set has a user-defined **Name** to characterize it, a **Time** threshold (specified in seconds), a **Deviation** threshold, and a **Tolerance** threshold. The file containing the fit parameter set must be given by **Filename**. All fit parameter set files must be stored in the directory where the loaded SCOUT configuration file is located. Fit parameter set files are created using the **Fit parameter set|Store** command in the main window (see [above](#)).

SCOUT loads each set in the list and fits until the user-defined time threshold or the user-defined deviation threshold is reached. The third quantity that determines the termination of the automatic fitting is the tolerance which is an internal quantity of the downhill simplex method used to optimize the parameters (see below).

Then the next fit parameter set is loaded until the end of the list is reached.

Downhill simplex method

The automatic parameter fit for each fit parameter set is done by a routine that makes use of the so-called downhill simplex algorithm. This method is a simple, but stable algorithm to minimize a function (the fit [deviation](#)) that depends on many parameters. See [[Press 1992](#)] for a detailed description.

In almost all cases this method will find the next local minimum of the fit deviation. There is no guarantee, that this is the global minimum as well. It is the responsibility of the SCOUT user, i.e. you, to select good starting values close to the global minimum, or at least to inspect the final SCOUT solution carefully. In many cases you can achieve a very good fit performance by finding the right sequence of fit parameter sets and making use of 'fit on a grid' steps.

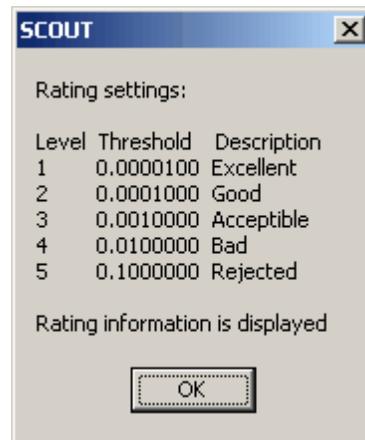
Automatic processing of many samples: Batch fit

In cases where an automatic fitting scheme is found, you can apply it to a sequence of input spectra. This SCOUT feature is described [below](#).

5.6 Rating

The quality of the agreement between the simulated and measured spectra is given by the fit deviation. If you want to work with a coarse classification of the fit quality you can compute and display rating information.

SCOUT has five rating levels with the following default settings:



For each level you can set the threshold and the descriptive text using the corresponding [menu commands](#). In the [options](#) menu there is also a switch to turn rating on or off. If rating is switched on you can display the current result in a view using a view object of type '[Rating view](#)'.

If you want to work with less than 5 rating levels you can set the threshold values for several levels to the same value. This way you can realize a simple 'good/bad' information, for example. However, you cannot have more than 5 rating levels at present.

5.7 Batch fit

5.7.1 Overview

If a fitting strategy has been developed that allows a completely automatic processing of input spectra you can make use of the batch fit feature of SCOUT. You have to enter the files to be processed into an appropriate table (which is created for you automatically), start the batch operation and wait. In the end you will receive a table with the results.

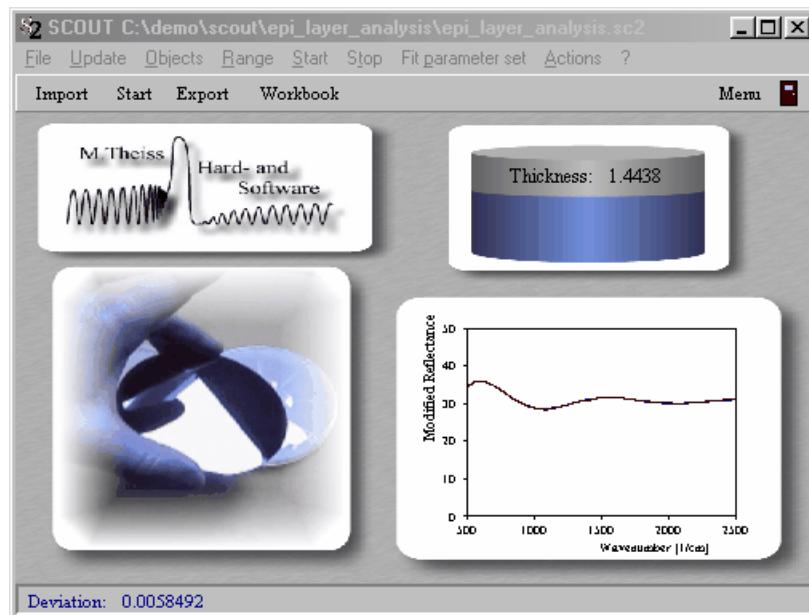
To do a batch fit, you have to [prepare the tables](#), activate the [automatic batch fit](#), and inspect and save the [final results](#). These steps are explained in the following.

5.7.2 Preparation of the tables

The fastest way to get going is this:

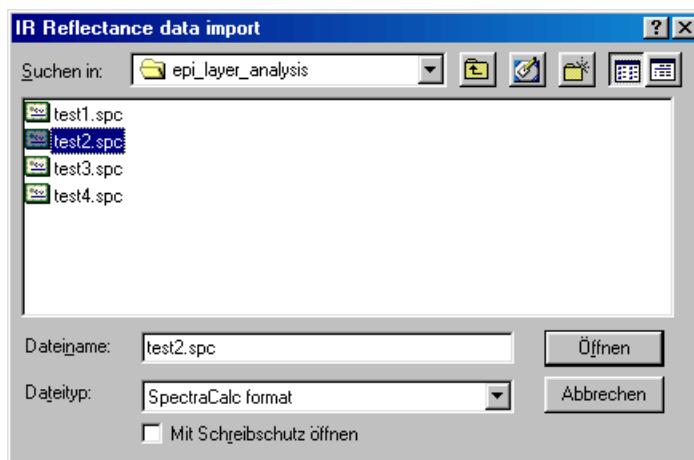
1.

It is assumed that you have a well-tested SCOUT model that can automatically handle the input spectra you want to analyze. Start SCOUT and load the corresponding configuration. Here is a simple example used to determine the thickness of an epitaxial silicon layer:



2.

Use the **Import** button in the toolbar of the main window of SCOUT. You are asked to select measured data for all spectra in the list of spectra, one after the other. Our simple example has only one spectrum, so we have to import one spectrum only:



3. Open the batch control window by **Actions>Show batch control** in the main window of SCOUT. The batch control window is very similar to the [workbook](#). It has three worksheets named **Results**, **Input options** and **Pre-fit**:

S: SCOUT batch fit control					
File Copy Paste Autosize cells Prepare Go! Actions					
A1					
	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					

4. Execute the **Prepare/New batch fit** menu command. The tables are now filled as far as it is possible. The results page now looks like this:

S: SCOUT batch fit control					
File Copy Paste Autosize cells Prepare Go! Actions					
A14		IR Reflectance:Factor			
	A	B	C	D	E
1	<i>Sample</i>				
2	Name				
3	Parameter				
4					
5	<i>Spectra</i>				
6	IR Reflectance				
7					
8	<i>Fit</i>				
9	Pre-fit set				
10	Fit deviation				
11					
12	<i>Parameters</i>				
13	Stack 1 Layer 2 Undoped silicon: Layer thickness				
14	IR Reflectance:Factor				
15					

5.

Before we enter items on the 'Results' page, we have to set some input options on the 'Input options' worksheet:

S: SCOUT batch fit control					
File Copy Paste Autosize cells Prepare Go! Actions					
E2		spc			
	A	B	C	D	E
1	Spectrum	Path		Input format	
2	IR Reflectance	C:\demo\scout\epi_layer_analysis\		4	1 spc
3					
4					
5					

For each spectrum in the SCOUT spectrum list you can set the path where the measured spectra are located. In the column 'Input format' you identify the file format by an integer number. Use one of the following:

- 1: [SCOUT standard format](#)
- 2: [xy-format](#)
- 3: [JCAMP format](#)
- 4: [SpectraCalc \(GRAMS\) format](#)
- 5: [Perkin-Elmer format](#)
- 6: [Zeiss format](#)
- 7: [Opus format](#)
- 14: [HunterLab UltraScan](#)

Some files (stored e.g. in the SCOUT standard format, the xy-format, the SpectraCalc and the HunterLab UltraScan format) may contain multiple spectra in one file. In this case the 'Input option' column gives the number of the spectrum in the file. The files are counted 1,2,3, ... If the data files all have the same file extension it is convenient to place it into the 'Extension' column. If not, leave this column empty.

6.

Now go back to the 'Results' page:

	A14	IR Reflectance:Factor	
	A	B	C
1	<i>Sample</i>		
2	Name		
3	Parameter		
4			
5	<i>Spectra</i>		
6	IR Reflectance		
7			
8	<i>Fit</i>		
9	Pre-fit set		
10	Fit deviation		
11			
12	<i>Parameters</i>		
13	Stack 1 Layer 2 Undoped silicon: Layer thickness		
14	IR Reflectance:Factor		
15			

Now it's time to enter the input data for the batch operation. You have to fill in items in the rows with black text. For each sample to be investigated you must specify a name. SCOUT will do batch fit from left to right and stop when it finds the first empty 'Name' cell.

For each sample you can specify a numerical parameter in the 'Parameter' row, but you don't have to. If you like to you can change the text 'Parameter' to something more specific. Here we take as an example the quantity 'Deposition time'.

Finally you have to enter a filename in the row labeled 'IR Reflectance'. If you have specified an input path and an extension in the 'Input options' worksheet, you can just put the pure filenames here. As an example, the table could now look like this:

The screenshot shows the SCOUT batch fit control window. The title bar reads "S SCOUT batch fit control". The menu bar includes "File", "Copy", "Paste", "Autosize cells", "Prepare", "Go!", and "Actions". The main area is a spreadsheet with rows numbered 1 to 15. Row 1 contains "Sample". Row 2 contains "Name" and values AB234-R, AB244-R, AB236-R, AB230-R. Row 3 contains "Deposition time" and values 1200, 600, 200, 100. Row 5 contains "Spectra". Row 6 contains "IR Reflectance" and values test1, test2, test3, test4. Row 8 contains "Fit". Row 9 contains "Pre-fit set". Row 10 contains "Fit deviation". Row 12 contains "Parameters". Row 13 contains "Stack 1 Layer 2 Undoped silicon: Layer thickness" and has a dropdown menu open. Row 14 contains "IR Reflectance:Factor". The bottom navigation bar includes "Results", "Input options", and "Pre-fit".

	A	B	C	D	E	F
1	<i>Sample</i>					
2	Name	AB234-R	AB244-R	AB236-R	AB230-R	
3	Deposition time	1200	600	200	100	
4						
5	<i>Spectra</i>					
6	IR Reflectance	test1	test2	test3	test4	
7						
8	<i>Fit</i>					
9	Pre-fit set					
10	Fit deviation					
11						
12	<i>Parameters</i>					
13	Stack 1 Layer 2 Undoped silicon: Layer thickness	▼				
14	IR Reflectance:Factor					
15						

If you have many filenames to enter, you can escape the typing work by using one of the following commands:

Place the cursor in the row of the spectrum for which you want to specify many input files, and select the menu command **Actions/Add multiple files**. You will get a file import dialog allowing you to select all files you want to process. Important for the right sequence of filenames in the batch control sheet is that you first select the last file and then, holding the Ctrl-key down, select the rest. If you have a continuous range of files in the dialog, select the last one first, hold the Shift-key down and then select the first file. This action will select all files between the last and the first. You can use wildcards like *R* in the filename editor in order to pre-select the files you see in the dialog. After you have selected the files, the path in the **Input option** page is adjusted if you have taken files from a different directory. It is a good idea anyway to check the setting on the input option sheet before you proceed: Selecting files with **Add multiple files** enters the full filenames including the file extension into the cells. In this case you should drop any entry for **Extension** on the Input option sheet.

You have to repeat this step for all spectra for which you have to specify input files.

Scan directory: This command will fill-in filenames completely automatic - you just have to specify a search directory. It can be used in situations where the filenames contain a text fragment that can be used to identify the corresponding spectrum in the SCOUT configuration. If, for example, there are reflectance and transmittance spectra in the search directory, and all reflectance spectra contain the text fragment 'RF' and all the transmittance spectra filenames contain 'T', you can call the reflectance spectrum in the optical model RF and the transmittance spectrum T. SCOUT will then collect all files that can be found with the wildcard "*RF*" as input files for the spectrum RF, and all files that can be found with "*T*" as input files for T.

7.

As a last preparational step you have to specify the so-called pre-fit sets which is done on the 'Pre-fit' worksheet:

S SCOUT batch fit control			
File Copy Paste Autosize cells Prepare Go! Actions			
B4	0.980828166007996		
A	B	C	D
1 Parameter set name	Set 1		
2			
3 Stack 1 Layer 2 Undoped silicon: Layer thickness	1.443779945		
4 IR Reflectance:Factor	0.980828166		
5			
6			
Results Input options Pre-fit			

In the first column you see the name of the fit parameters. SCOUT has already created a second column labeled 'Set 1'. If this worksheet has more than one column the following will happen: Before the actual fit for a sample is started, SCOUT tries all parameter sets defined on this 'Pre-fit' page and computes the fit deviation. The parameter set with the lowest deviation will be used to set the starting values of the parameters.

This means: If you have only one set like now, the fit will definitely start with these numbers. If you delete column B, the fit will start with its current fit parameters which may depend on the history of the batch operation, in particular on the last sample. This may be a wanted feature or not. It's your decision. If you specify more than one pre-fit set SCOUT will determine the best for each sample and use it as starting values.

In the present demonstration, three sets are defined as follows:

S SCOUT batch fit control			
File Copy Paste Autosize cells Prepare Go! Actions			
D4	1		
A	B	C	D
1 Parameter set name	Set 1	Set 2	Set 3
2			
3 Stack 1 Layer 2 Undoped silicon: Layer thickness	1.5	0.8	0.3
4 IR Reflectance:Factor	1	1	1
5			
6			
Results Input options Pre-fit			

Be sure to have a name for each set. SCOUT iterates through all sets until the first empty name cell is found.

8.

That's it! Return to the 'Results' page and save the settings for safety reasons with **File|SaveAs** using the *.vts file format. It's now time to [start the batch fit](#).

5.7.3 Go!

You are now ready for the batch fit operation - almost. It is a good idea to do a short test by placing the cursor into one of the sample columns on the **Results** sheet and execute the menu command **Fit current sample**. Instead of doing the batch fit for all entered samples, the analysis is done for the selected sample only. This way you can test if the loading of the spectra and the fit routine works as you think.

Execute the **Go!** command in the batch control window and watch the starting activity. SCOUT will load the input spectra, indicate the sample under investigation in the top of the batch control window, start the fit and enter the final result in the 'Results' page. If you have defined '[Optical functions](#)' their values are written to the batch control window underneath the results of the fit parameters.

The selected pre-fit set and the achieved fit deviation is entered as well:

S₂ SCOUT batch fit control

File Copy Paste Autosize cells Prepare Go! Actions

E12

	A	B	C	D	E
1	<i>Sample</i>				
2	Name	AB234-R	AB244-R	AB236-R	AB230-R
3	Deposition time	1200	600	200	100
4					
5	<i>Spectra</i>				
6	IR Reflectance	test1	test2	test3	test4
7					
8	<i>Fit</i>				
9	Pre-fit set	Set 1	Set 2	Set 3	Set 3
10	Fit deviation	0.0058492	0.0193696	0.0095141	0.0089941
11					
12	<i>Parameters</i>				
13	Stack 1 Layer 2 Undoped silicon: Layer thickness	1.4437759	0.6777128	0.2213485	0.1075605
14	IR Reflectance Factor	0.9808289	0.987103	0.9725202	0.9839448

Results / Input options / Pre-fit /

You should now [inspect the results](#) a little closer.

5.7.4 Inspecting the results

The batch control window including all results can be saved with the SCOUT configuration. You can select part of table with the mouse and use the **Copy** command to copy it to the clipboard. From there you can import the data in other programs like Excel, Word, Using the **File|SaveAs** command you can also store the table directly in Excel 5.0 format.

Look at the values of the fit deviation and check if they are small enough.

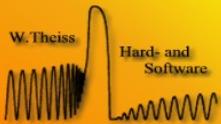
You can have a quick look at the fit results using the command **Actions|Loop through results**. In this case, starting with the first sample, the spectra and the obtained fit parameters are restored in SCOUT which now shows the status after having processed this sample. A small dialog lets you choose to continue with the next sample or quite the cycle through all fit results.

If you place the cursor in a sample's column, you can

- **Actions|Load fit parameters into SCOUT:** This loads the fit parameter values of this sample into SCOUT. Alternatively you can execute this command by the speed button labeled as 'FP --> SC2'.
- **Actions|Load spectra into SCOUT:** This loads the input spectra of this sample into SCOUT (Speed button: 'SP --> SC2').
- **Restore SCOUT (spectra and parameters):** This loads both the fit parameter values and the spectra of this sample into SCOUT (Speed button: 'FP+SP --> SC2').

It may happen that one of the fits was bad and, after reloading the poor results, you can improve the fit by some manual actions. In this case you can 'repair' the table and enter the improved values using the command **Actions|Enter fit parameters into table** (or the speed button 'SC2 --> WB'). Of course, the cursor must be in the column of the sample that has been revisited.

SCOUT



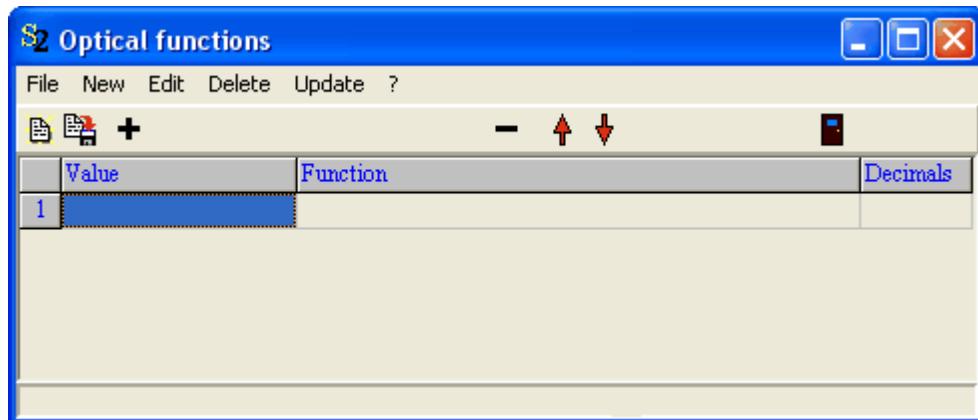
Part

VI

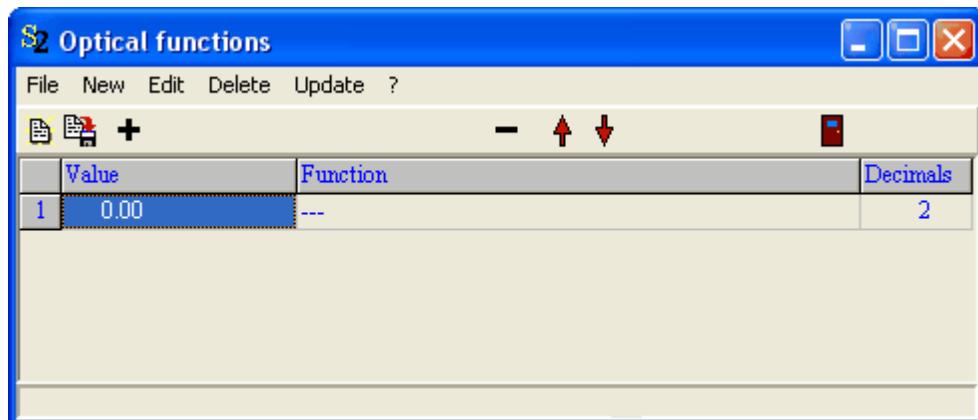
6 Optical functions

Definition of optical functions

Starting with version 2.2 you can access information about objects using so-called 'Optical functions'. Typical values are the refractive index of a material or reflectance and transmittance values of spectra at certain spectral positions. 'Optical functions' are defined in the list of optical functions. You can open this list in the main window with 'Objects|Optical functions':



Using the '+' button you can create a new entry in this list:



In order to define a function enter an expression in the column 'Function'. If you press **Update** SCOUT tries to evaluate the function (see below) and displays the result in the column 'Value' using as many digits as you specify in the 'Decimals' column. If the function cannot be evaluated a zero is returned.

Valid expressions

The expressions to be used to define optical functions are very simple: object name(argument). The object name must be exactly the same as it appears in your SCOUT configuration. There must be no space between the object's name and the argument. The argument is passed to the object which returns the result - if possible. If you have to specify a spectral position in the argument you can use all units that SCOUT knows, e.g. nm, 1/cm or eV.

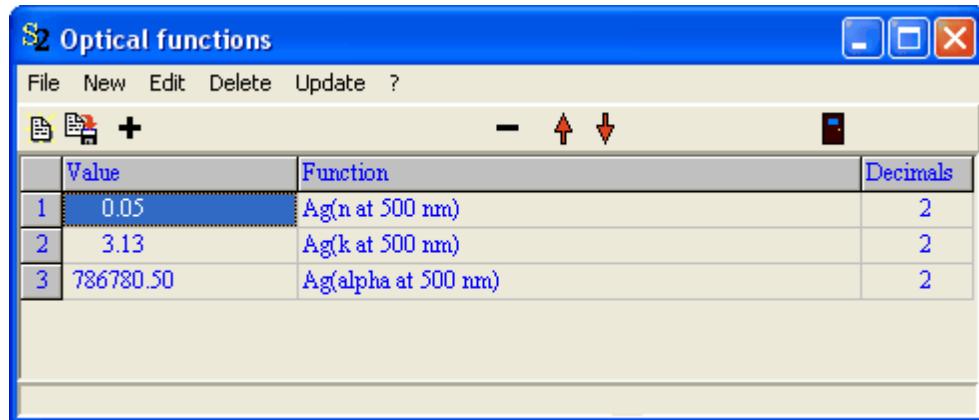
The search order for the object is the following

1. [material list](#)
2. [layer stack list](#)
3. [spectra list](#)

The CODE software which is based on SCOUT will search in the list of integral quantities as well.

Creating optical functions by drag&drop

The easy way to generate optical functions is by drag&drop. Drag a material (here: Ag) from the list of materials and drop it on the list of optical functions. The following items are generated automatically:



You can drag other objects like spectra (and integral quantities in CODE) as well to generate items in the list of optical functions.

Where do optical functions appear?

The values of all optical functions are written to the workbook after the [Export](#) command in the main window. They are also listed in the [Batch control window](#) when the results for a sample are reported.

6.1 Materials

Materials can handle the following arguments:

(n at 540 nm): this function returns the real part of the refractive index of the object at 540 nm. You can use alternative spectral units like 1/cm or eV if you like.

(k at 540 nm): this function returns the imaginary part of the refractive index of the object at 540 nm.

(alpha at 540 nm): this function returns the absorption coefficient (in 1/cm) of the object at 540 nm.

6.2 Layer stacks

Layer stacks respond to the following arguments:

(Reflectance at 540 nm): this function returns the reflectance of the layer stack at 540 nm (normal incidence of light)

(Transmittance at 540 nm): this function returns the transmittance of the layer stack at 540 nm (normal incidence of light)

(Absorption at 540 nm): this function returns the absorption (= 1-R-T) of the layer stack at 540 nm (normal incidence of light)

6.3 Spectra

Spectrum objects can compute the following quantities:

(simulated value at 540 nm): This function returns the value of the simulated spectrum at 540 nm.

(measured value at 540 nm): This function returns the value of the measured spectrum at 540 nm.

(measured average 2000 3000 50): This function returns the average of the measured spectrum between the spectral positions 2000 and 3000 (nm, eV, 1/cm or whatever spectral unit the object has at the moment), computed using 50 points.

(simulated average 2000 3000 50): This function returns the average of the simulated spectrum between the spectral positions 2000 and 3000 (nm, eV, 1/cm or whatever spectral unit the object has at the moment), computed using 50 points.

(measured integral 2000 3000 50): This function returns the integral of the measured spectrum between the spectral positions 2000 and 3000 (nm, eV, 1/cm or whatever spectral unit the object has at the moment), computed using 50 points.

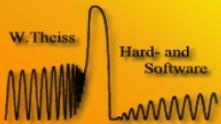
(simulated integral 2000 3000 50): This function returns the integral of the simulated spectrum between the spectral positions 2000 and 3000 (nm, eV, 1/cm or whatever spectral unit the object has at the moment), computed using 50 points.

(FFT thickness 5000 10000 2048 5 20 1.7): Computes and returns the [FFT thickness](#) using 2048 points between 5000 and 10000 1/cm (obtained by interpolation of the measured spectrum). The refractive index used in the method is 1.7, the thickness is searched between 5 and 20 microns.

(Thickness from fringes 1000 1600 0.05 20 3 2): Analyzes the maxima of interference patterns in the spectral range 1000 to 1600 (spectral unit taken from the spectrum). The function returns the layer thickness of layer 3 in the layer stack which is assigned to the spectrum. Pairs of interference maxima are taken into account only if their height is larger than 0.05 and if the peaks are separated by at least 20 (spectral unit taken from the spectrum). The function averages over 2 pairs of peaks. This function should be applied only in the case of a single layer causing the interference pattern.

(L* D65 2°): returns the color coordinate L*, computed for the illumination spectrum D65 and an observation angle of 2°. Possible values for the color coordinates are L*, a*, b*, X, Y, Z, x, y, z, l, a and b. Illumination spectra can be A, D65 and C, and observation angles of 2° and 10° may be used in the expression.

SCOUT

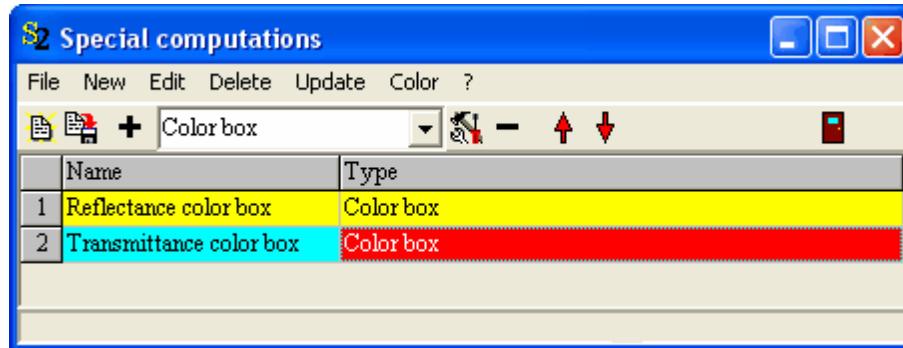


Part

VII

7 Special computations

Starting in the main window of SCOUT, the menu command **Objects|Special computations** opens the list of special computation objects. These objects perform (sometimes lengthy) computations that are not wanted with every update of the model.



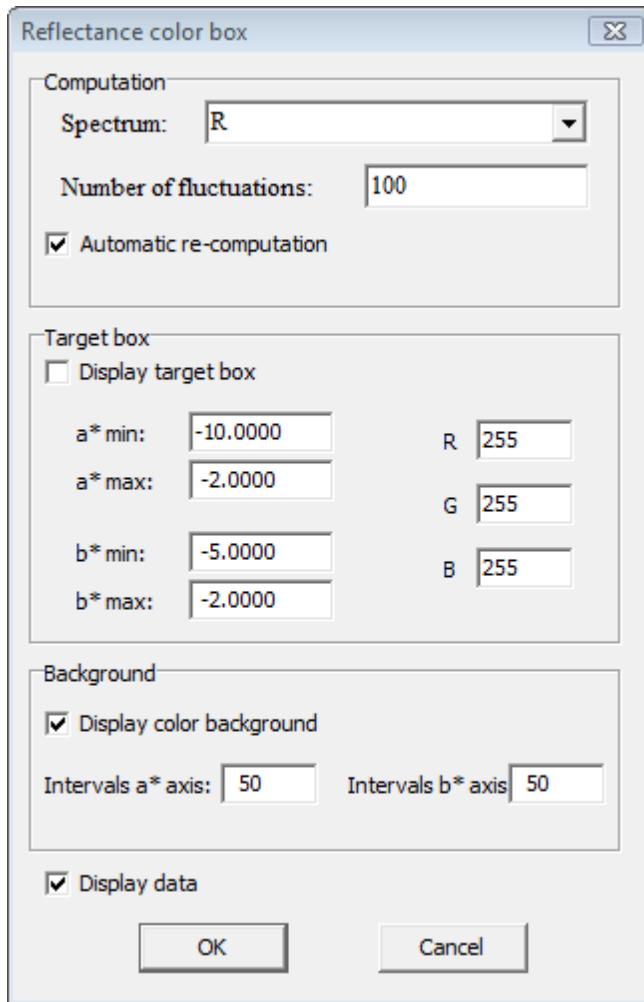
The **Edit** command shows the current object's parameter dialog, **Update** forces a re-computation of the object's data. If an object of this type is displayed in a view, a click on the display also starts the re-computation of the data.

At present the following special computation objects are available:

[ColorBox](#)

7.1 ColorBox

This object computes the variation of the color coordinates a^* and b^* due to [thickness fluctuations](#). The parameters of the computation (obtained with the **Edit** command) are the following:

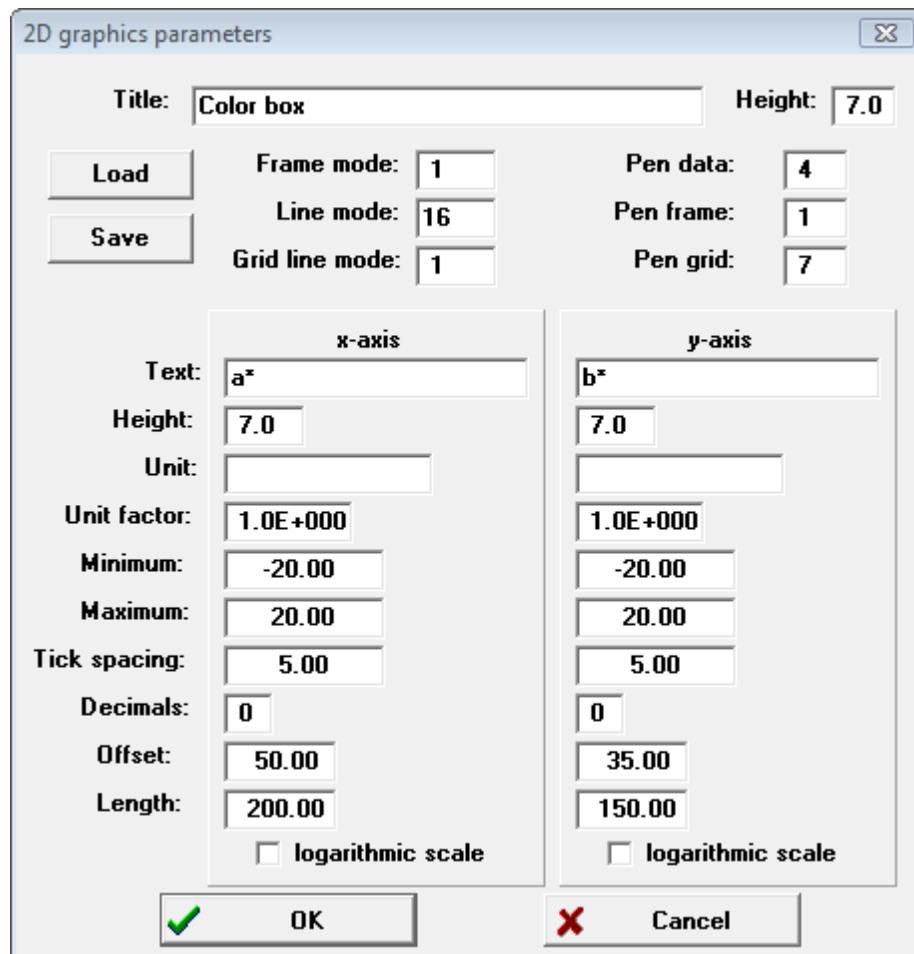


Select the spectrum the color of which you want to investigate and enter the number of iterations that are to be used in the computations - in this case you will get 100 values for a^* and b^* . If you activate the checkbox **Automatic re-computation** all 100 values are computed whenever you change a model parameter (by a slider movement, for example). Since this means to re-compute the whole model 100 times the program may react slow with this option activated. If disabled the data are computed on demand only, i.e. by a click on the corresponding view object (see below).

If you like you can define a rectangle in the a^*-b^* plane to visualize the tolerated area of color (target box). The rectangle is drawn in the color specified by the RGB values in the dialog.

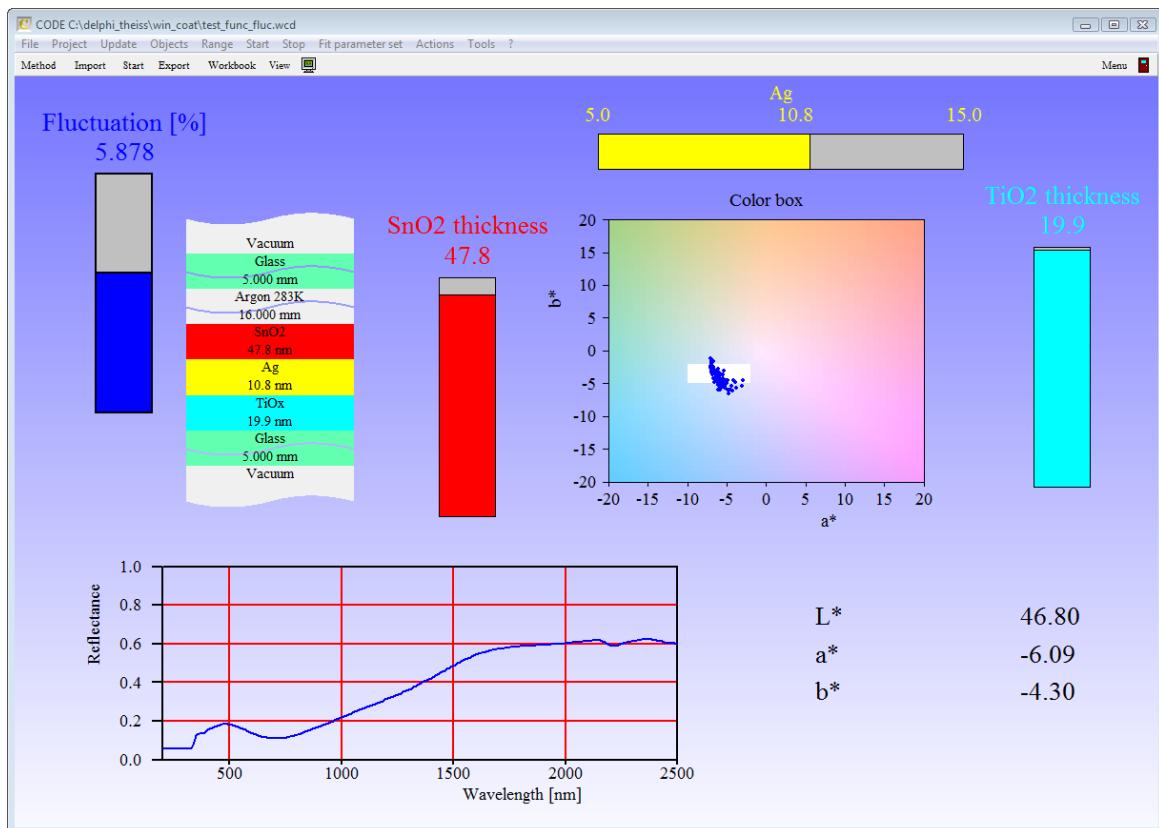
Activating the option **Display color background** draws a colored background indicating the color impression for the various positions in color space. With the settings shown above this background is drawn using 2500 little rectangles (50 by 50).

The ColorBox graph of b^* vs. a^* can be shown in a view object. The graphics parameters are set in the following dialog:



Since the color box graph will show a cloud of points you should use a 'line mode' for individual points (11, 12, 13, 14 or 16, see graphics course) but no continuous lines connecting the points.

In order to really see the color box drag the object from the list of special computations either directly to the main window of SCOUT or the list of the current view objects. In a view, a color box looks like this:

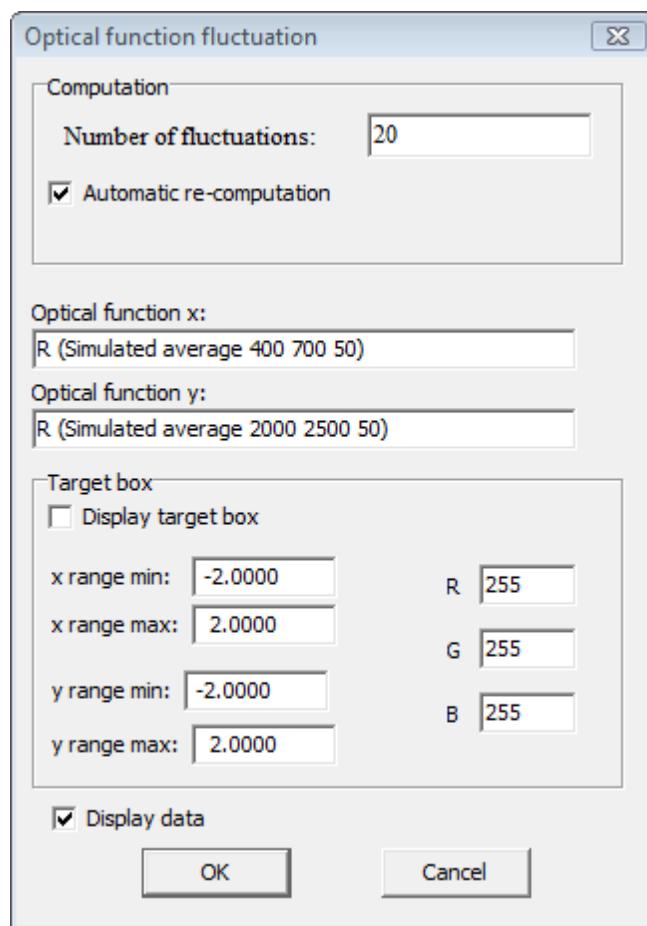


A mouse click on the object in a view activates the re-computation of the object's data (unless you activated the option of automatic re-computation).

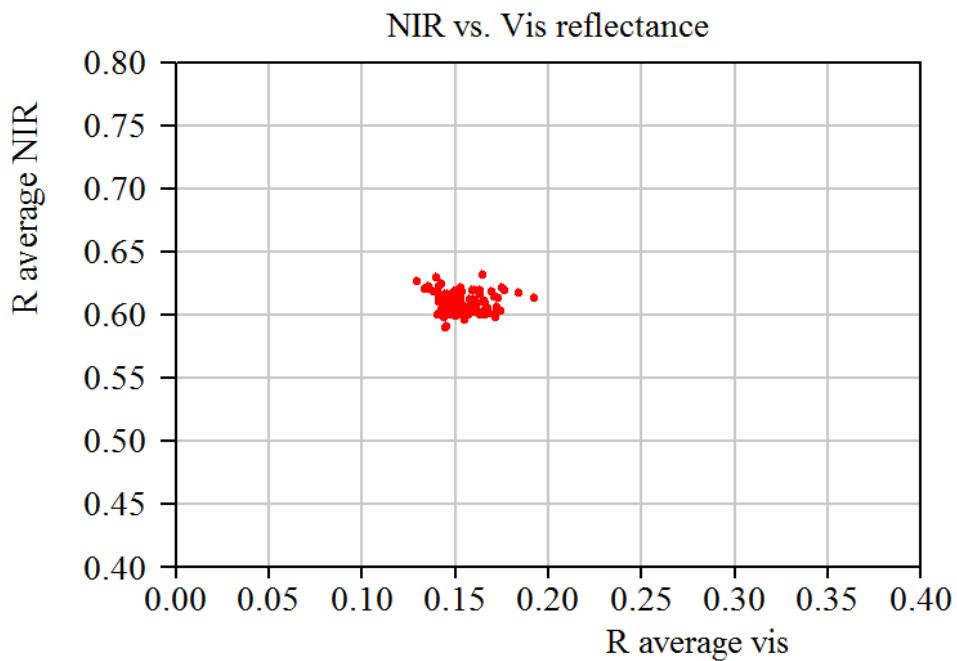
The example above shows a configuration which allows to regulate the strength of the thickness fluctuation of all layers by the slider on the left side. If automatic re-computation for the ColorBox object is selected, you can instantly watch the cloud of points while you change layer thicknesses and fluctuation strength. This way you can explore the relation of layer thicknesses and color stability.

7.2 Optical function fluctuation

Objects of this type work almost like ColorBox objects. Instead of being limited to the color coordinates a^* and b^* , you can use arbitrary optical functions for the x- and y-values of the cloud points. Here is an example:



The result in a view looks like this:

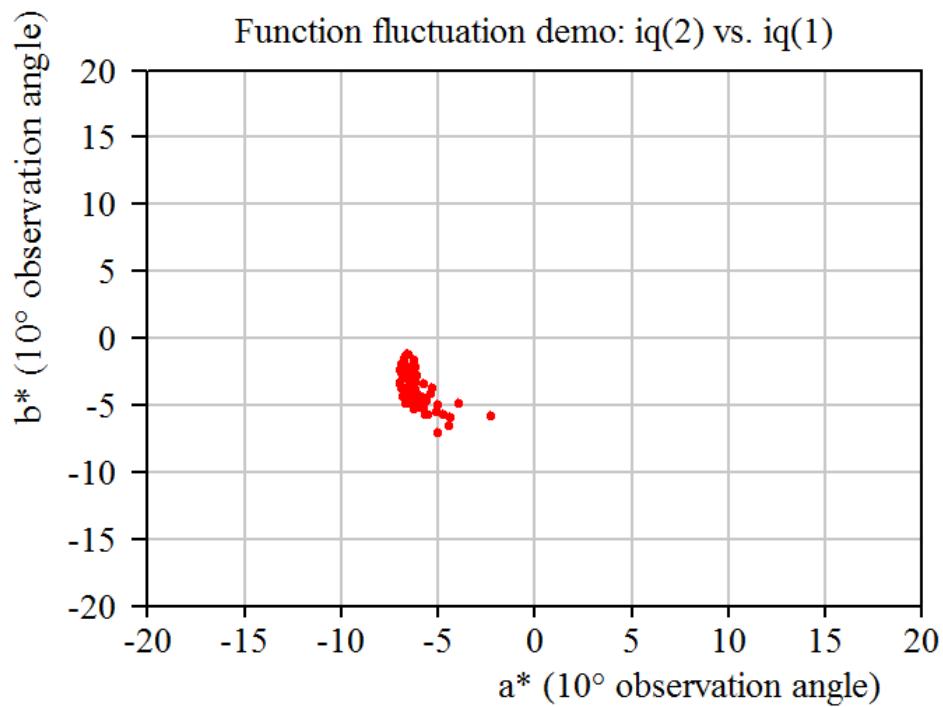
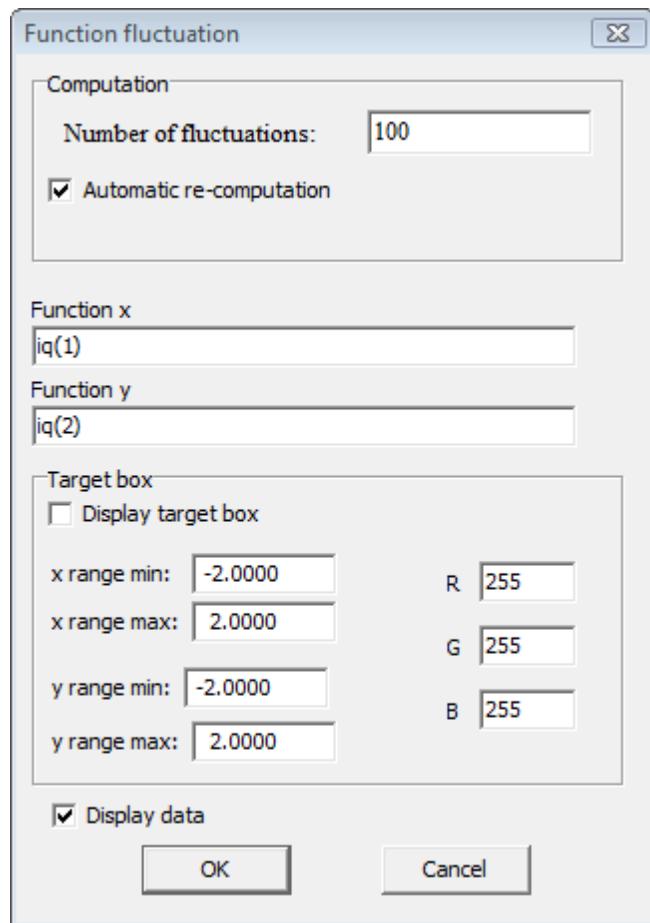


7.3 Function fluctuation

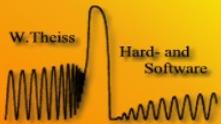
This object type is even more flexible than the **Optical function fluctuation** type described above. The x- and y-coordinates of the displayed points are computed according to user-defined functions. These may refer to values of objects in the list of optical functions (the term of(4) takes the 4th object in the list of optical functions) or objects in the list of integral quantities. The latter is available in the CODE software only (not in SCOUT!). You can refer to integral quantities by iq(xx) where xx denotes the position of the wanted object in the list. The term iq(23) takes the value of the 23rd integral quantity.

Since terms like of(5)-iq(12) refer to absolute positions of the wanted objects in their lists, you must not change the order of the optical function objects or integral quantities. If you did so, you should check if the expressions do still what you intend.

An example for a **Function fluctuation** object is given below:



SCOUT



Part

VIII

8 Data acquisition

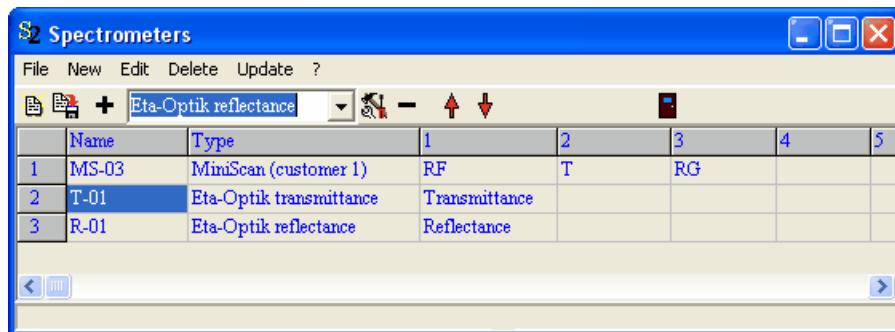
8.1 Overview

Starting with version 2.18 a new data acquisition concept has been introduced in SCOUT. It replaces the older mechanism of external OLE servers doing the measurements. The new way of data acquisition offers more flexibility, increased operational convenience and safety.

SCOUT contains a list called 'Spectrometers' which manages spectrometer objects. Each object in the list represents spectrometer hardware which delivers - after data acquisition - one or more measured spectra. Measured spectra can be assigned to simulated spectra in the [list of spectra](#). If a measured spectrum is assigned to a simulated spectrum, the new measured data are filled in the experiment object of the simulated spectrum whenever a data acquisition step has been processed. Each spectrometer object provides an **Edit** method for setting the relevant parameters for data acquisition. The spectrometer settings are saved as part of the SCOUT configuration.

The global **Acquire** command (use the SCOUT menu command **Actions|Acquire** or the  button in the toolbar) starts the data acquisition for each object in the spectrometer list. When all spectrometers have finished their measurements, the new experimental data are distributed to the simulated spectra according to the assignments (see below). New fit deviations are computed, and graphs are updated.

The spectrometer list looks like this:



	Name	Type	1	2	3	4	5
1	MS-03	MiniScan (customer 1)	RF	T	RG		
2	T-01	Eta-Optik transmittance	Transmittance				
3	R-01	Eta-Optik reflectance	Reflectance				

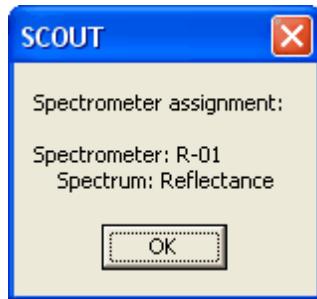
Each spectrometer object should be given a unique descriptive name. The columns labeled 1, 2, ... show the names of the spectra offered by the individual spectrometer objects. Most object types deliver one spectrum only (as the spectrometers T-01 and R-01 in the example). Spectrometer MS-03 exports 3 spectra as indicated by the names in columns 1, 2 and 3.

Measured spectra are assigned to simulated spectra by **Drag&Drop**. A drag operation must start with the mouse pointer positioned in the **row of the relevant spectrometer object**, and in the **column of the measured spectrum** to be used. The target of the Drag&Drop operation can be either the **row of the simulated spectrum in the list of spectra**, or the **window of the simulated spectrum itself**. After a successful Drag&Drop assignment you are informed about the new connection:



In order to check the spectrometer assignment for a simulated spectrum, use the menu command **Acquire|Show spectrometer assignment** in the simulated spectrum object. You will receive an

information like this:

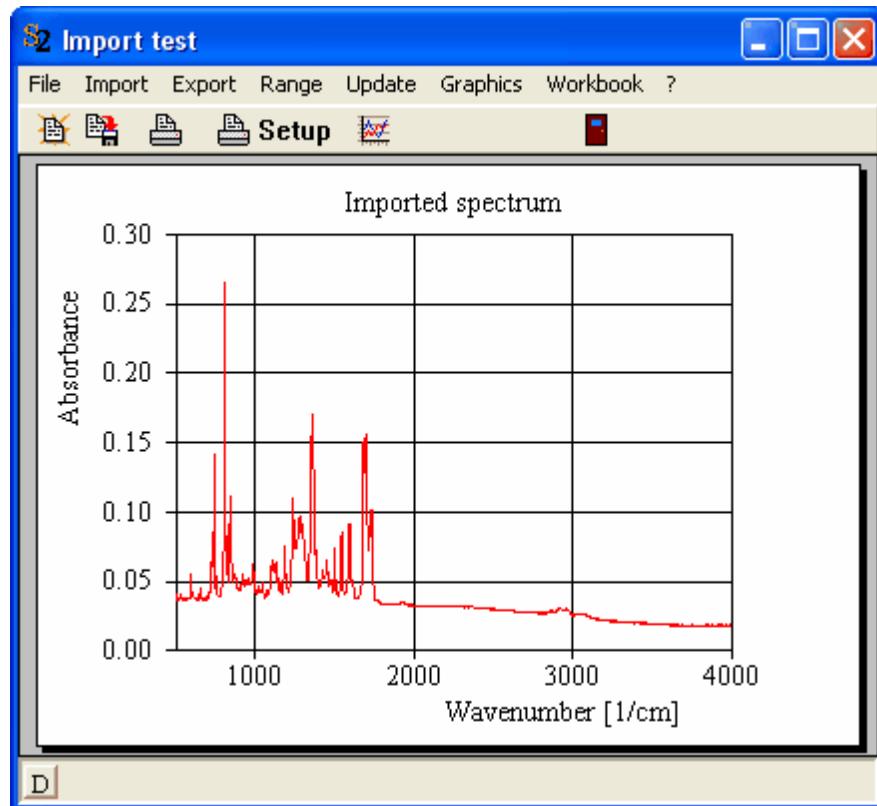


Use the command **Acquire|Reset spectrometer assignment** to delete the current assignment. Alternatively, you can do a new assignment in order to replace the spectrometer connection of a simulated spectrum object.

8.2 File import

This object type does not do measurements itself but imports a spectrum from a data file. This can be used if an external data acquisition program permanently writes its measured data to one and the same file. In order to connect this program to SCOUT you can use an object of the 'File import' type to import the measured data whenever the **Acquire** command of SCOUT is executed.

The object window looks like this:



Use the **Import** command to initially specify the filename and the file format of the data file that is used for the spectrum transfer. The measured data are automatically updated when the **Acquire** command in SCOUT is used.

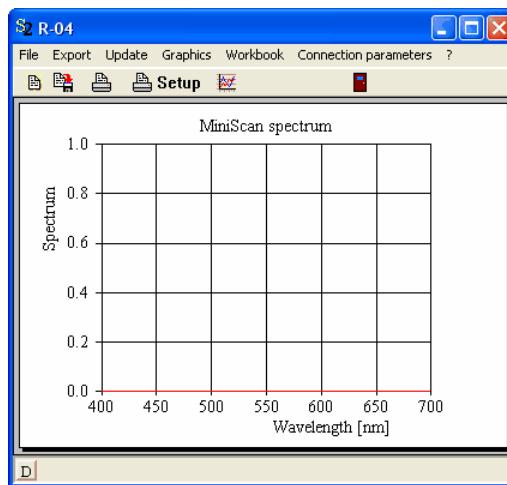
8.3 MiniScan

This object acquires spectra using a HunterLab MiniScan portable spectrometer (connected to the PC by a serial interface):

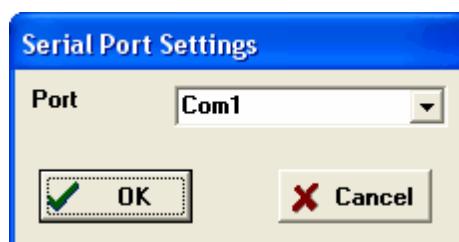


The current implementation assumes that you have calibrated the hardware manually and you just want to take reflectance spectra.

The object window is this:



There is only one parameter to be specified which is the serial port used for communication. Execute the menu command **Connection parameters** and select the serial port in the following dialog:

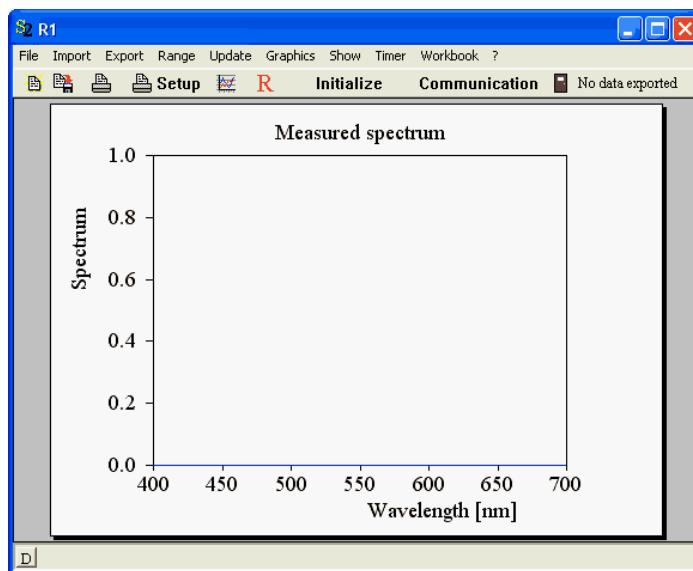


The serial port settings are stored in the SCOUT configuration. When the global **Acquire** command in SCOUT is executed all MiniScan spectrometers in the spectrometer list will record new spectra.

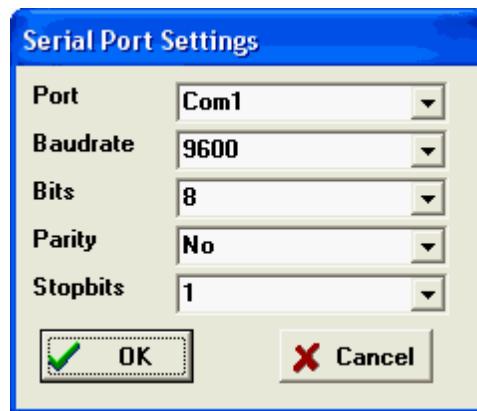
8.4 Minolta CM-2600d

This portable spectrometer is controlled through commands sent via the serial port. The SCOUT interface lets you send commands to initialize the spectrometer and to record a reflectance spectrum.

The Minolta window looks like this:



Use the **Communication** button to set the communication parameters in the following dialog:



The **Initializations** button sends the initializing commands to the spectrometer. If there is a file called 'minolta_initializations.ser' in the SCOUT program directory, SCOUT will read this file and send each line to the spectrometer. Below each command line you should place a text line saying 'Check error'. If this line is found SCOUT reads the spectrometer's response from the serial port and gives a warning if an error code is returned. Here is an example of a command file:

```
CPS,1,1
Check error
```

If SCOUT does not find the file 'minolta_initializations.ser' it sends the command 'CPS,0,0' in order to configure the spectrometer hardware.

In order to acquire a spectrum use the **R** button (or the Acquire command in the main window). In this case SCOUT sends the commands found in the file 'minolta_acquire.ser'. In this file there may be additional configuration commands. The last command should execute the measurement. Do not use a 'Check error' line after the command to do the measurement.

Here is an example for a valid file:

```
ATD  
Check error  
MES
```

After all lines have been sent, SCOUT waits for the measured data to be returned by the spectrometer.

8.5 OceanOptics spectrometer

Support for several OceanOptics spectrometers has been integrated into SCOUT. Before you can use OceanOptics hardware in SCOUT, you have to install the original OceanOptics software (for example OOIBase32) and make sure that the hardware is running properly. The file ooidrv.ini (located in the Windows directory) should contain the correct spectrometer and ADC type.

Up to now we have integrated support for the following spectrometer types:

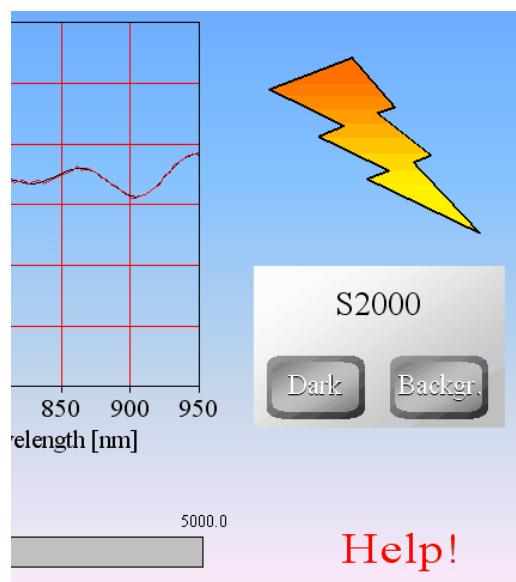
- ADC1000USB
- USB2000
- USB4000
- NIR512

In addition, the very flexible **Omnidriver** has been implemented which automatically scans for attached spectrometers and lets you select the one you want to work with.

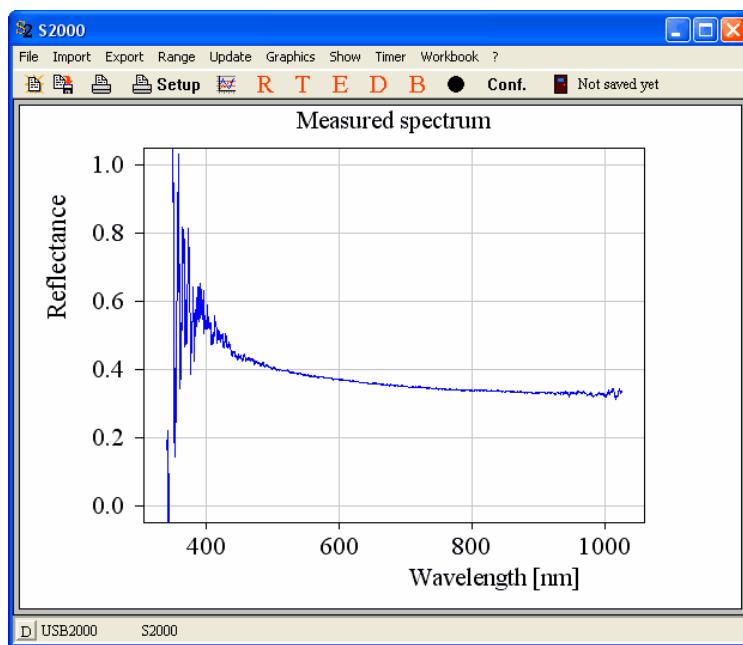
If your spectrometer is not listed above please contact us in order to find out if support for your hardware can be integrated into SCOUT as well.

In some cases the wavelength calibration for the spectrometer pixels could not be read from the hardware. In these cases it turned out to be necessary to create a text file with wavelength information in order to get the required wavelength assignment into SCOUT - in such a case you should contact us as well.

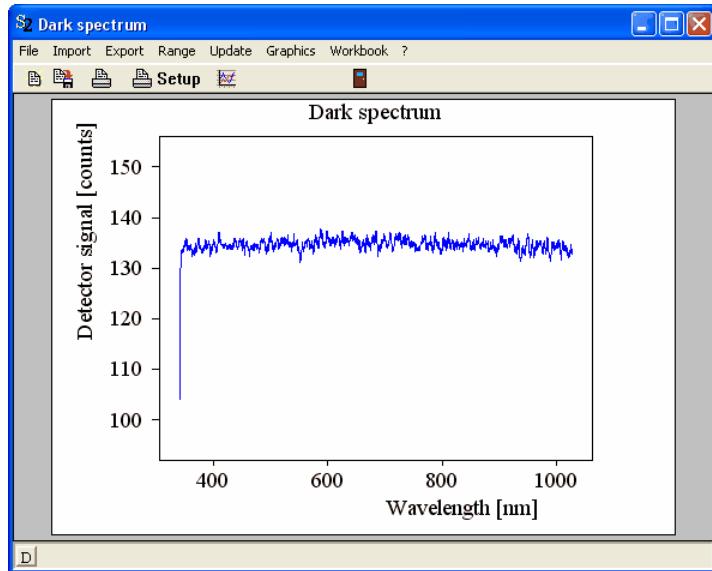
Like any other spectrometer object in SCOUT OceanOptics hardware can be represented in the main view as shown below. The gray rectangle with in the lower right corner has buttons for doing the **Dark** measurement and the **Background** measurement. The dark measurement should be performed first - using a 'black' sample such that no light from the sample stage reaches the detector. Secondly the background spectrum must be taken using a calibrated mirror in the case of reflectance or an empty sample stage in the case of transmittance. Having done the dark and background spectrum, you can execute the sample measurement pressing the **Acquire** button in the view, i.e. the arrow button in the example below:



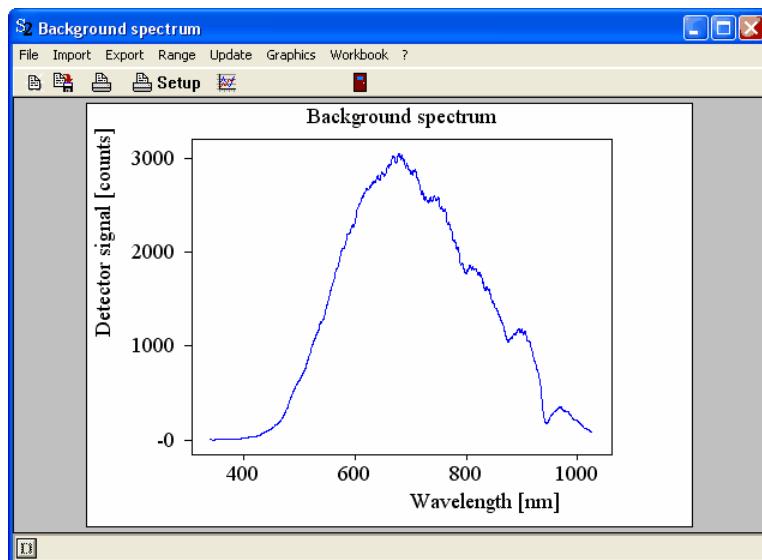
Holding the Ctrl-key down and clicking the name of the spectrometer (S2000 in this case) a window with access to all details of the spectrometer opens. The main graph shows the final sample spectrum (in this case: the absolute reflectance):



The menu command **Show|Dark spectrum** opens a subwindow displaying the recorded dark signal:



The background signal is shown in the subwindow that opens with the spectrometer window command **Show|Background spectrum**:

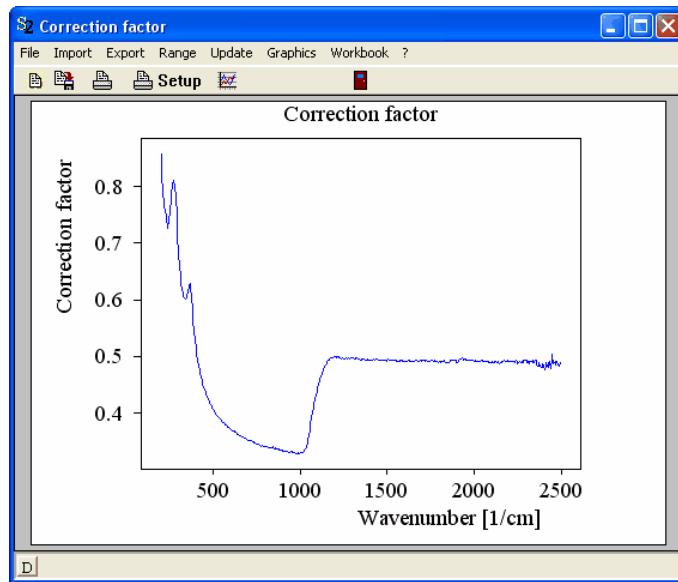


The absolute reflectance of a sample is computed as

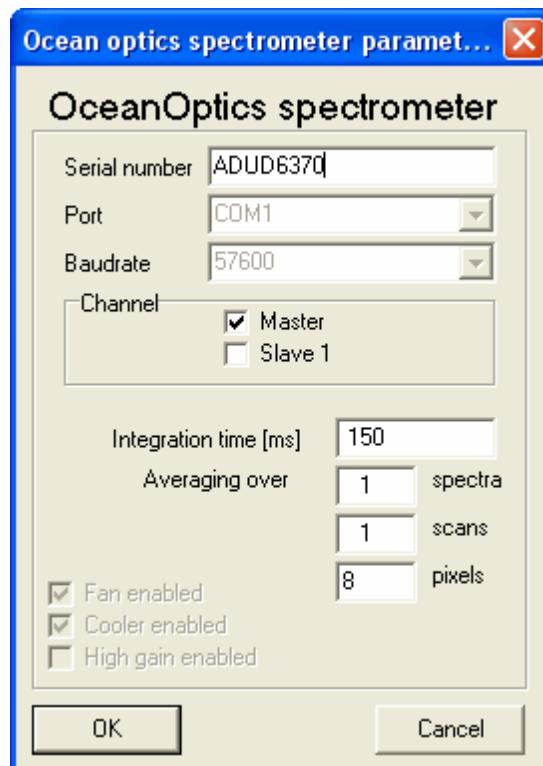
$$R = R_{\text{correction}} * (S_{\text{Sample}} - S_{\text{Dark}}) / (S_{\text{Background}} - S_{\text{Dark}})$$

Here S_{Sample} is the sample signal, S_{Dark} the dark signal and $S_{\text{Background}}$ the background signal.

The absolute reflectance of the reference mirror $R_{\text{correction}}$ can be entered in the subwindow that opens with the menu command **Show|Correction factor**. A silicon wafer (polished on both sides), for example, has a reflectance like this:



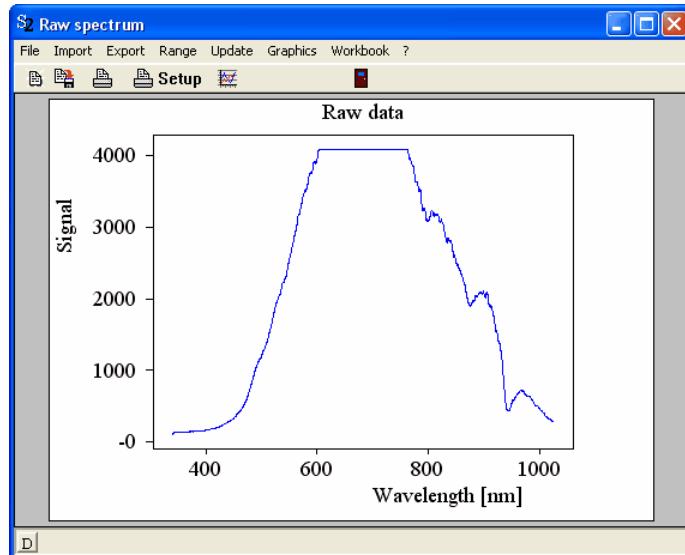
The button 'Conf.' opens the spectrometer configuration dialog:



In most cases it is sufficient to enter the serial number of the spectrometer, the integration time in ms and some averaging parameters: You can average the signal of several neighboured **pixels**, several **scans** within the spectrometer hardware and several final **spectra** in SCOUT. The integration time should be chosen such that the dynamic range of the spectrometer is used efficiently without reaching the upper limit. If the integration time is too large, an overexposure of one of the pixels may occur. In this case you might get a warning like the following



and the corresponding signal might look like this (in the case of a 12 bit detector which has a maximum number of counts of 4095):



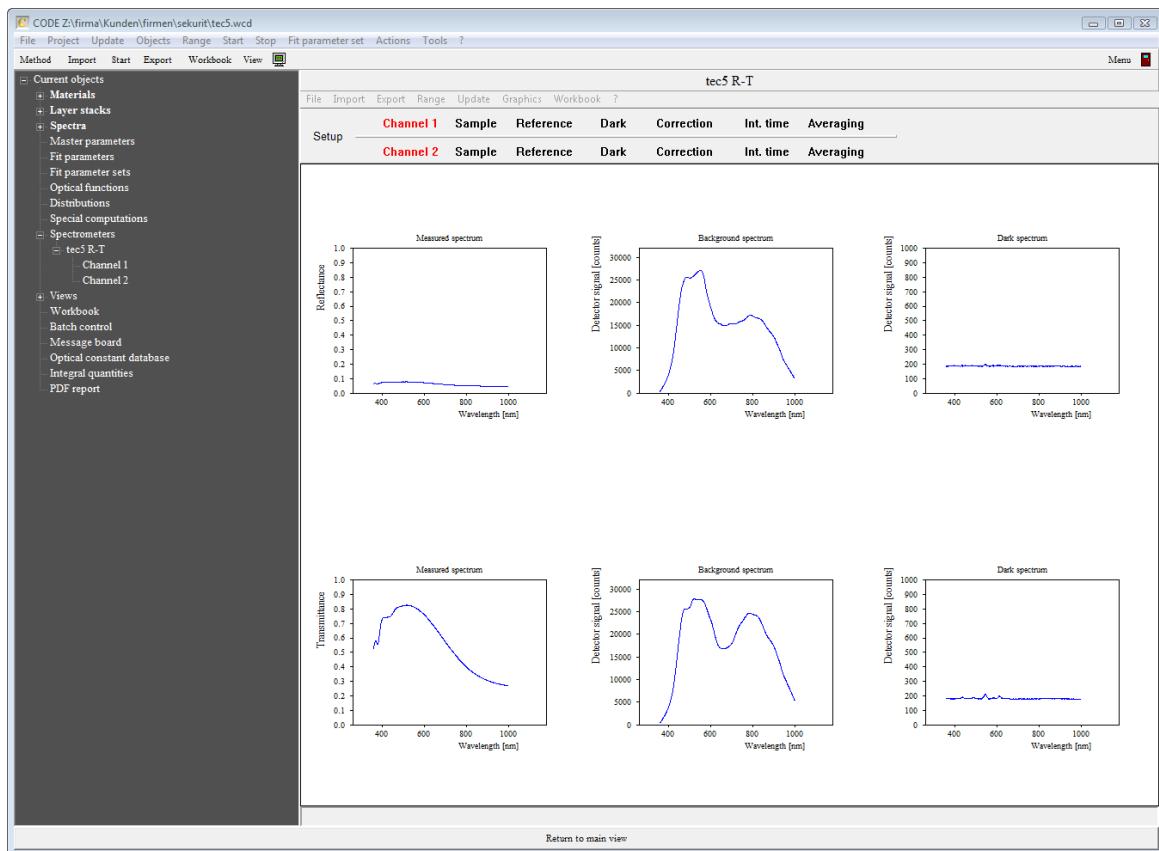
8.6 Avantes spectrometer

Various Avantes Vis and NIR spectrometers can be used to acquire spectra, including the new USB2 spectrometers.

8.7 tec5 spectrometer

In January 2011 we implemented tec5 hardware support. Single spectrometers and systems with 2 spectrometers are supported.

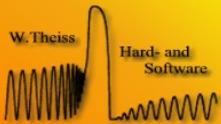
The object controlling a double spectrometer system looks like this:



For each spectrometer (called Channel 1 and Channel 2) you can easily set the integration time and the number of scans to average. Using the buttons **Dark**, **Reference** and **Sample** you can perform the dark measurement (this should be done first), the reference measurement and finally the sample measurement. In order to check the performance of the system, you have displays of the dark signals (to the right), the signals of the reference measurement (in the center) and the final sample spectra (left side) for both channels.

The example shown above is a system recording reflectance and transmittance of a thin film sample simultaneously. In a case like this, you should assign the measured reflectance (channel 1) to a spectrum object that simulates reflectance, and channel 2 should feed a spectrum simulation object computing transmittance.

SCOUT



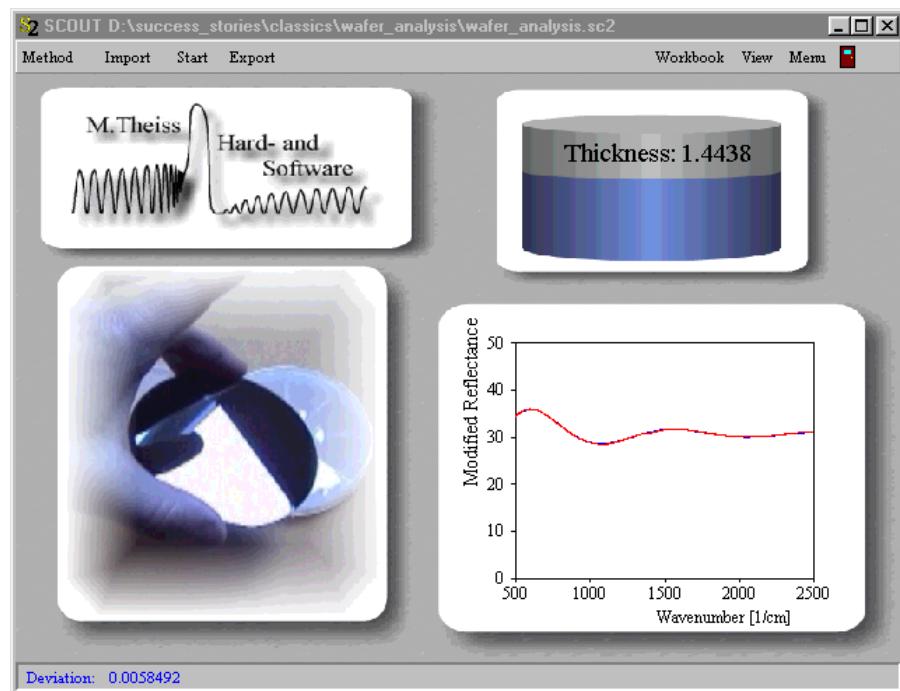
Part

IX

9 Views

9.1 Introduction

The main window of SCOUT can be used to give quick overviews on relevant parts of the SCOUT configuration without opening additional windows. Once the definition of the physical model is finished you can close all 'subwindows' and analyze series of spectra just using commands in the main window. The main window can be configured in a way such that also unskilled persons with little knowledge about spectroscopy can operate the SCOUT solution. Here is an example:



The display in the main window is defined in so-called 'Views' which can be configured automatically or user-defined. Views are managed in the [list of views](#) which is opened by the command **Objects|Views** in the main window.

Quick access to view elements

If you want to make changes to one of the elements of a view (e.g. to alter the graphics settings of a spectrum), you can **click on it while holding the Ctrl-key pressed down**. In this case the element's **Edit** function is called which gives you access to the element parameters. Doing this for the displayed reflectance spectrum you can open the spectrum definition window.

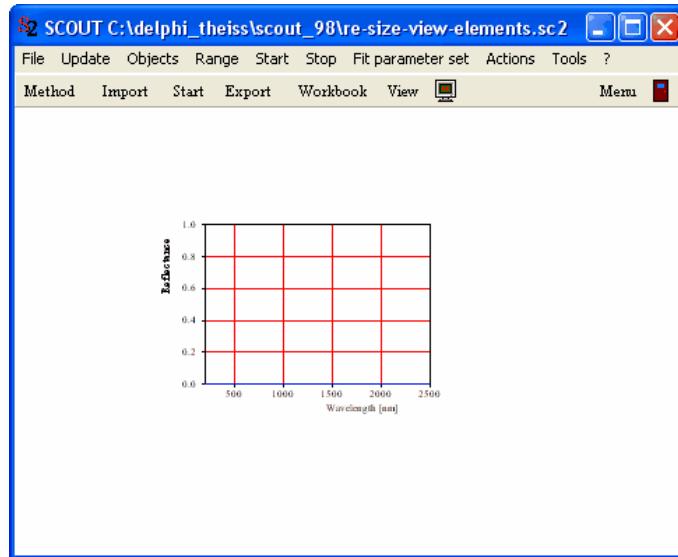
Moving view elements

View elements can be moved by mouse drags while the **Caps key** is pressed down.

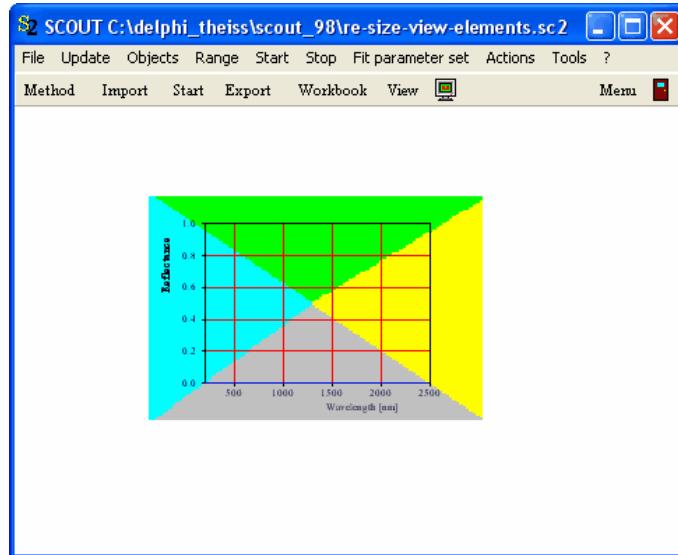
Re-sizing view elements

You can re-size elements by starting mouse drags (with the **right mouse button** hold down) at the center of the element. Moving the mouse cursor down to the right increases the size, moving up to the left makes the element smaller. Move slowly in order to give SCOUT time enough to re-draw the view. Stop the drag operation when the wanted size is achieved.

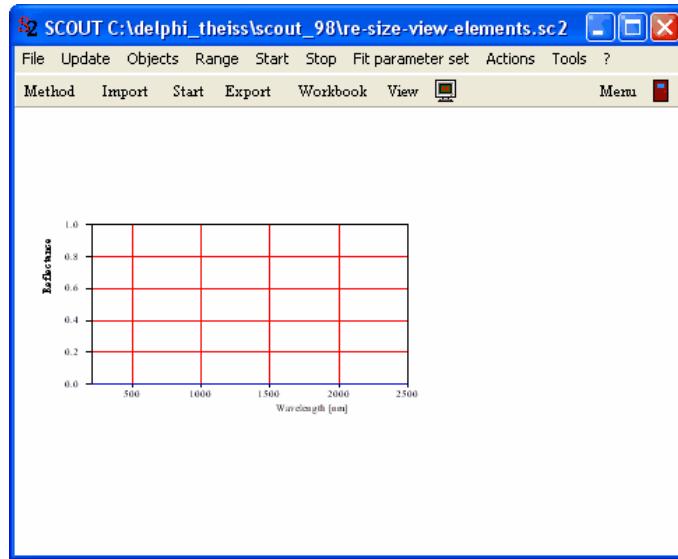
A more powerful re-size mechanism is explained using the example of a spectrum view object:



Suppose you want to move the left side of the graph to the left without changing the other upper, lower and right boundary of the view element's rectangle. IN order to do this, imagine the objects rectangle being divided into four sections like the colored triangles sketched below:



In order to move the left boundary of the spectrum object's rectangle, hold down the **Caps** and the **Alt** key at the same time, place the mouse cursor in the blue triangle and drag it to the left (pressing the left mouse button down). While you move the mouse, the view is constantly updated. Stop when the a satisfying position is reached:



In a similar way you can re-position the upper, lower or right boundary of a view element's rectangle.

Password protection

Since you can make changes to the model by accessing the view elements from the main window these actions are password protected.

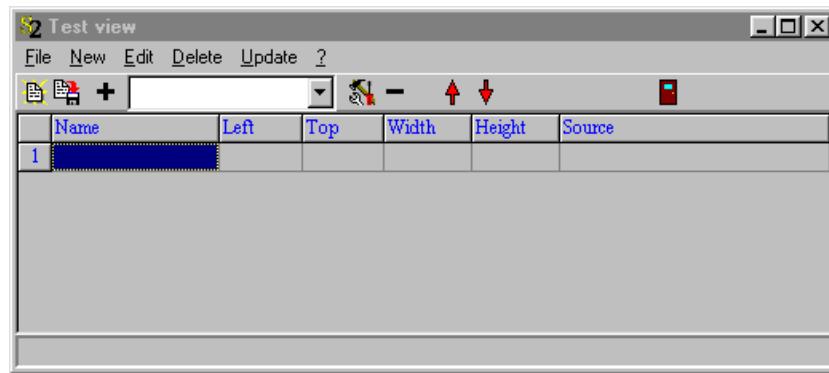
9.2 List of views

User-defined views are created, edited and deleted in the so-called 'List of views' which is opened in the main window of SCOUT using the menu item **Objects|Views**. Do not mix up this menu item with the toolbar button labeled '**View**' which is used to switch from one view to the next. If a view object has been created in this list it is displayed in the main window of SCOUT replacing the picture that is loaded at program start. If the list of views contains more than one item the toolbar button called '**View**' switches from one to the next. If the last view object has been reached a further click on the '**View**' button returns to the first view in the list. This way you can easily cycle through all defined views.

Some commands in the **Actions** submenu activate automatic view generation.

9.3 Defining views

To define the appearance of a view select it in the list of views and use the **Edit** command. This will open another list which contains the items that are to be displayed in the main window:



Here you can create several types of objects some of which are explained in the following. If no

objects are defined SCOUT will fill the main window with a white background.

If you define more than one object to be displayed in a view note that the objects are drawn in the sequence given by the list. The object on top of the list is drawn first, then the second one and so on. This will be important for the design of a view if objects overlap.

At present the following view elements are implemented:

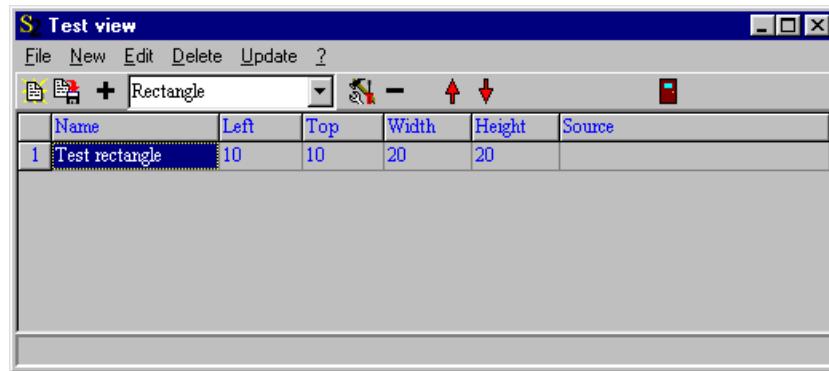
Rectangle	Displays a rectangle with user-defined color.
Text view	Displays a user-defined short text.
Bitmap view	Displays a bitmap picture.
Metafile view	Displays vector graphics.
Field view	Displays dielectric functions, spectra or distributions.
Parameter view	Displays a parameter value as text.
Parameter slider	Displays and modifies a parameter value by a slider.
Horizontal bar	Displays and modifies a parameter value by a horizontal bar graph.
Vertical bar	Displays and modifies a parameter value by a vertical bar graph.
Workbook display (columns)	Displays workbook data with one x- and multiple y-columns.
Workbook display (rows)	Displays workbook data with one x- and multiple y-rows.
Sample view	Displays the current sample identification.
Deviation view	Displays the current fit deviation.

9.4 View objects

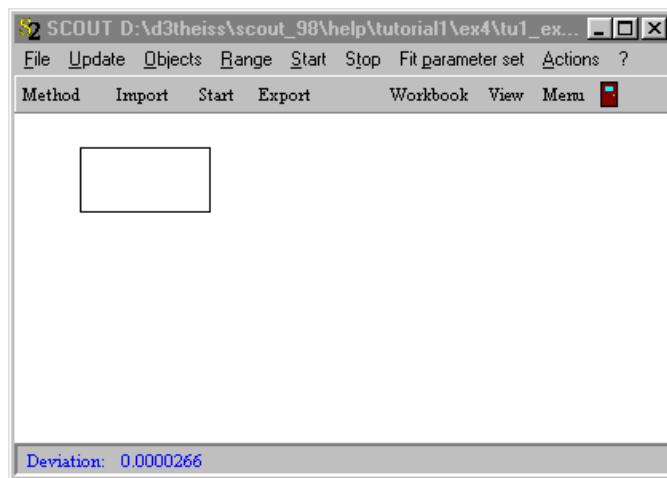
9.4.1 Display items

9.4.1.1 Rectangle

This object draws a rectangle in the view. Size and position of the rectangle are specified by the **Left**, **Top**, **Width** and **Height** entry in the list. All values have to be specified in percent of the total area of the main window. The entry



causes the following appearance of the main window:

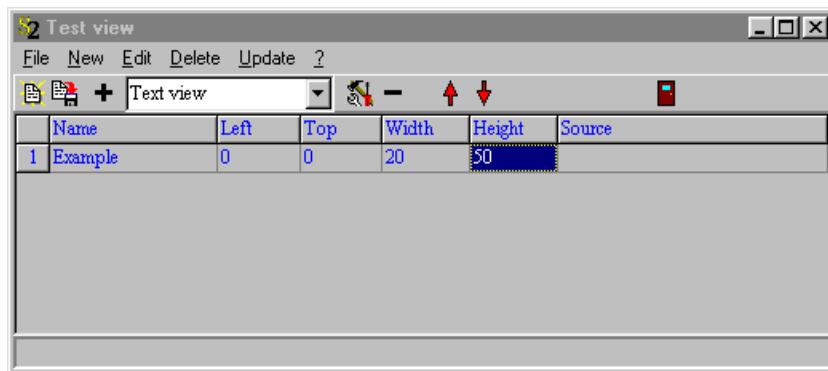


You can change the color of the rectangle using the **Edit** command (or a right click of the rectangle's row). You will get two color picking dialogs: The first one for the filling of the rectangle, the second one for the boundary.

If you want to fill the whole main window with a colored background you can use a rectangle object with Left = 0, Top = 0, Width = 100 and Height = 100.

9.4.1.2 Text view

This object type is used to display a text string in the view. The entry



causes the view to display the text 'Example'. The text appears in a rectangle whose upper left corner is determined by the values given in **Left** and **Top**. The height of the font used to draw the text is taken from the **Height** value. The **Width** parameter has no meaning. The color of the text can be specified in the color picking dialog that you get with the **Edit** command.

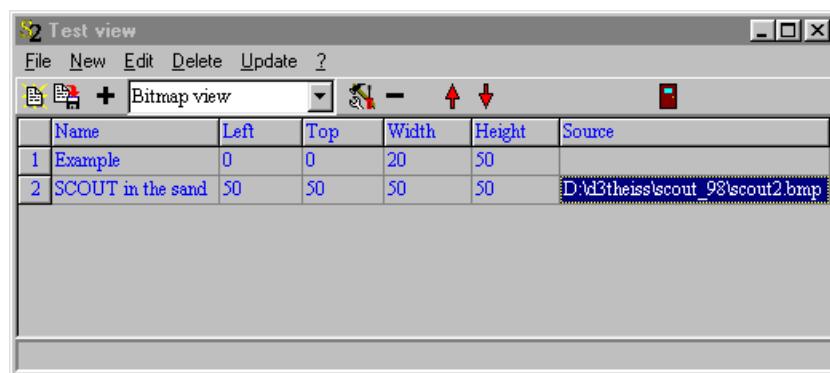
The example values above lead to the following view:



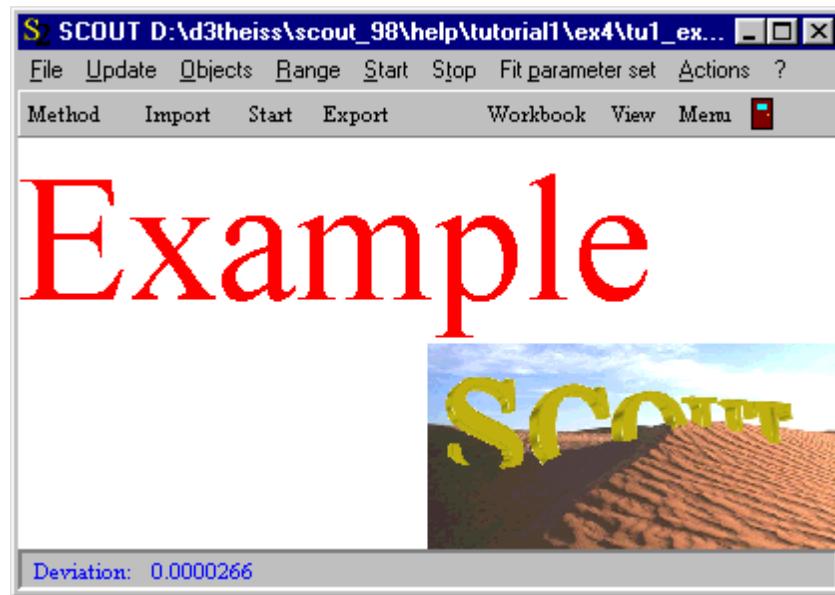
9.4.1.3 Bitmap view

Bitmap view objects display bitmaps in a rectangle (defined by the **Left**, **Top**, **Width** and **Height** parameters). The **Edit** command gives you access to a dialog to specify a Windows bitmap file (*.bmp). The **Source** value shows the location of the imported bitmap file.

The main purpose of using bitmaps is to illustrate the SCOUT method, i.e. indicate what is does, show a logo, give short explanations to the user or simply to make the SCOUT appearance nicer. The entry



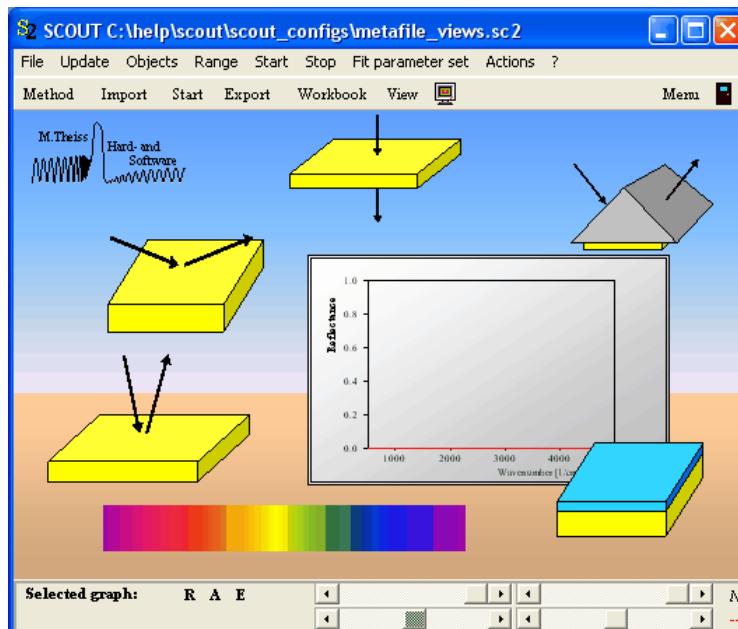
leads to this view:



Warning: After you imported a graph in a bitmap view object it becomes part of the SCOUT configuration and SCOUT does not need the original file any more. However, be aware of the fact that the SCOUT configuration file can get very large if it contains large bitmap files.

9.4.1.4 Metafile view

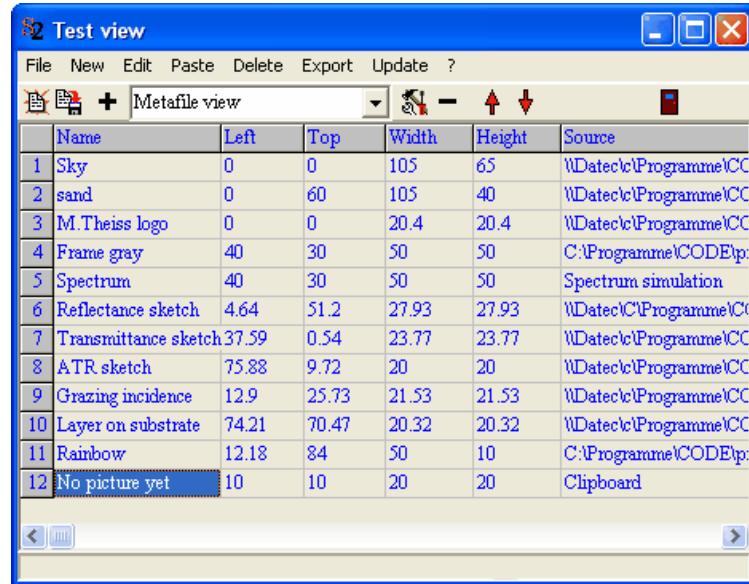
Metafile view objects display vector graphics in a rectangle (defined by the **Left**, **Top**, **Width** and **Height** parameters). The **Edit** command gives you access to a dialog to specify an Extended Windows Metafile (*.emf). The **Source** value shows the location of the imported metafile. The SCOUT installation copies some useful metafiles in the subfolder pictures. Some of them are sketches, others can be used as more or less colorful backgrounds. Here are some examples:



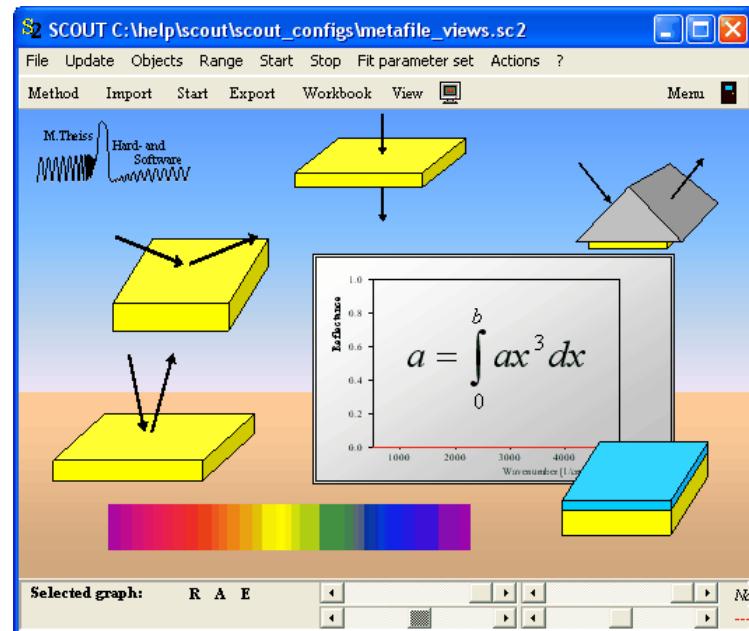
Creating metafile graphics

Extended metafiles are supported by only a few Windows graphics programs. The easiest way to create them is the following: Use Microsoft Word or other text processing programs which are able

to create drawings as vector graphics. Create a picture and copy it to the clipboard. Then create a metafile object in a SCOUT view like the one in the last row below:



Then use the **Paste** menu command which will paste the clipboard picture into the metafile view object (if the clipboard contains a metafile format). Update the main window of SCOUT to inspect the imported picture. If you use a formula editor you can even paste formulas in metafile objects:



The graph is now part of the SCOUT configuration. If you want to re-use the graphics for other configurations you can store the picture as extended metafile by selecting the metafile view object and applying the **Export** command in the view list.

The easy way to add sketches to a view

The main window of SCOUT has a menu command called **Actions|Add symbol to view**. This does the following: A metafile view object is created in the center of the current view and the corresponding Edit command is executed automatically. This opens the pictures subfolder of SCOUT where you can select a picture. Afterwards you can move or re-size the metafile picture in

the view.

Moving and re-sizing pictures

Like all view objects, metafile pictures can be moved in a view this way: Keep the **Caps** key pressed down, select the picture in the view with the **left mouse button** and **drag** it to the new position.

Re-size objects by the following action: Select the picture with the **right mouse button** in its upper left corner. Then **drag** the mouse cursor down to the right with the right mouse button kept down. Stop at the wanted new size.

Metafiles do not consume very much space in SCOUT configurations. You can use them freely to illustrate your SCOUT views.

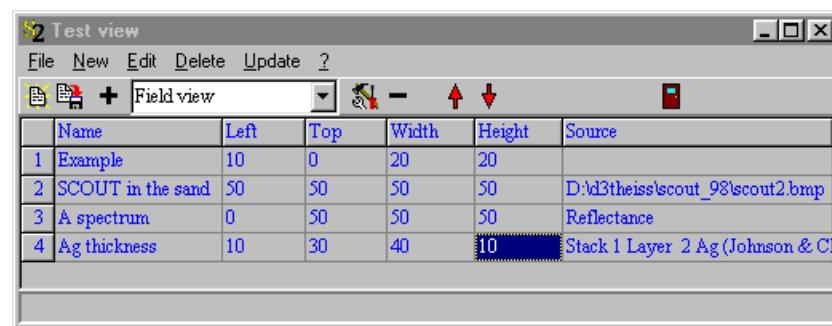
9.4.1.5 Parameter view

Objects of this type display values of model parameters in the view. They are created by Drag&Drop exclusively. **Do not create objects of this type using the '+' button of the list**.

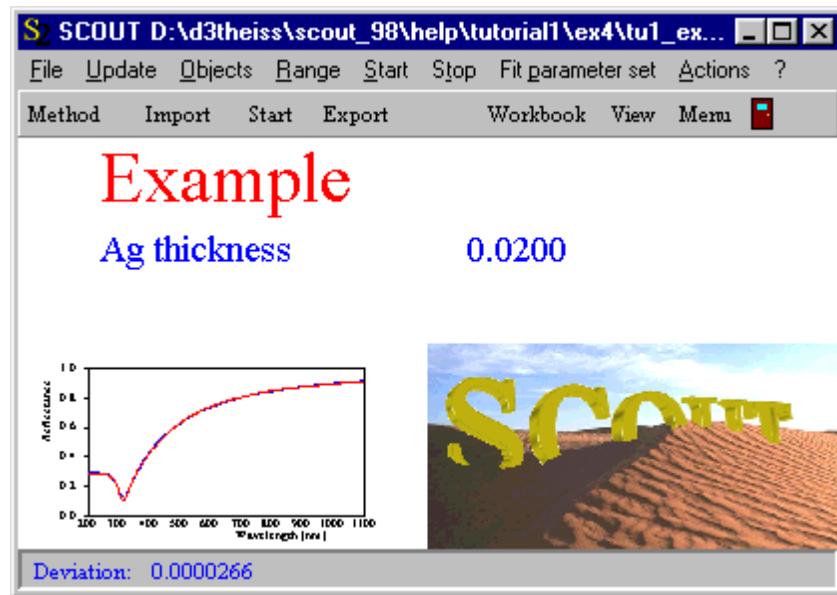
To display a layer thickness in the view, open the layer stack definition and the view definition simultaneously and drag the layer entry to the view definition list. After dropping the item a new object of type parameter view is created and connected automatically to the thickness parameter as indicated in the **Source** column.

If you drag a dielectric function model from the list of dielectric functions to the view definition list SCOUT creates a parameter view entry for all parameters of the dielectric function model. Generally SCOUT creates as many parameter view entries as the dropped object has fit parameters.

You can define in the **Name** column a name for the parameter of your choice. Beginning at the position defined by **Left** and **Top**, the text entered as **Name** is displayed. The color as well as the **Height** of the font are defined as for text view objects. At a distance given by the value of **Width** to the right of the Top position the parameter view object starts to output the numerical value of the parameter. Here is an example:

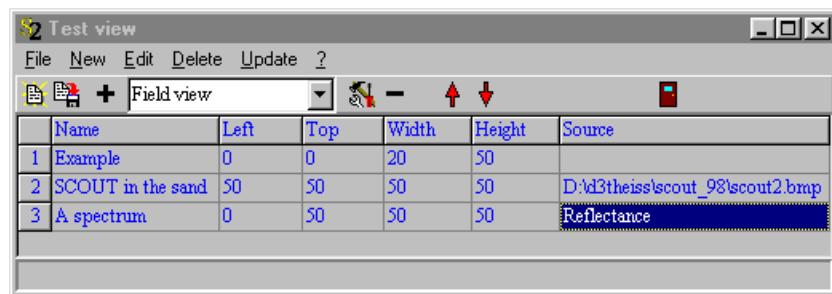


causes SCOUT to display the following output in the main window:

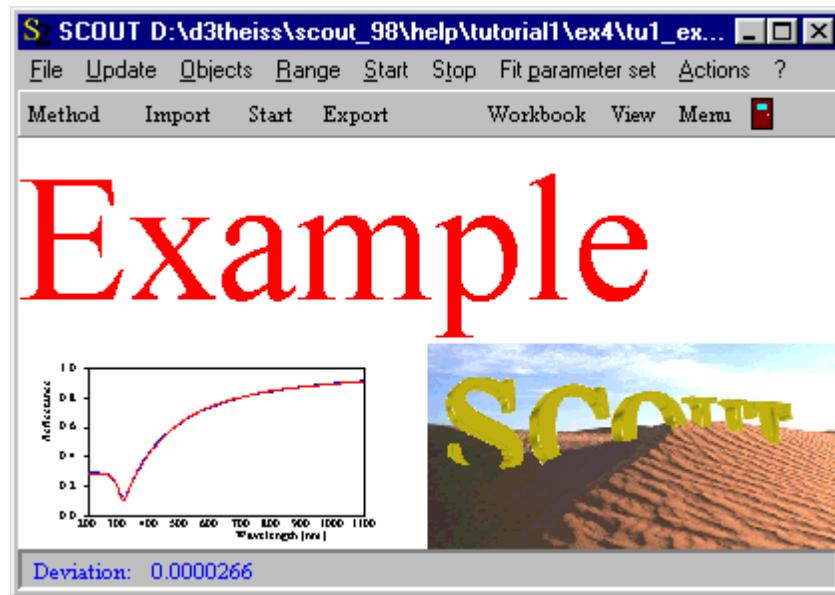


9.4.1.6 Field view

Field view objects display SCOUT [spectra](#), [dielectric functions](#) or [distributions](#). The graph is drawn within a rectangle defined by the **Left**, **Top**, **Width** and **Height** parameters. The data to be displayed have to be defined by **Drag&Drop**: To display a spectrum you have to open the list of spectra (from the main window: **Objects|Spectra**) and start in the row of the desired spectrum a drag operation. The dragging has to end with a drop above the row of the field view object in the view definition list. The example



creates the following view in the main window:



During a parameter fit SCOUT updates the view in the main window if an improvement has been achieved.

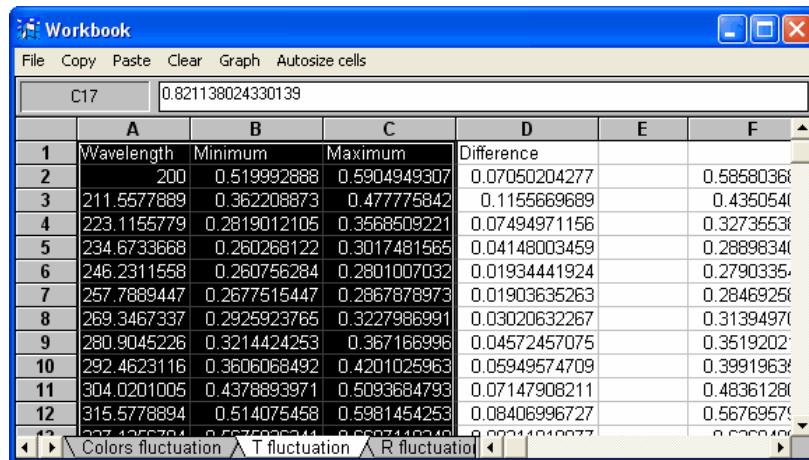
Besides spectra from the spectra list you can also drag dielectric function objects from the list of dielectric functions to a field view object if you want to display optical constants in a view.

9.4.1.7 Workbook display (columns)

This object displays workbook data in a view. It can be used in the case where you have a column of x-values and one or several columns of y-values and you want to display all y-values vs. the x-values. In order to use an object of this type in a view, proceed like this: Open the workbook and select the appropriate worksheet containing your data. Then select the cells that contain the x-values. The first row may contain text which will then appear in the graph labeling the x-axis. Then, in a second step, select the y-values. If you included explaining text for the x-values in the first row, you have to include text for each y-column as well. All y-values must be in a contiguous selection as shown in the example below:

S2 Workbook											
File Copy Paste Clear Graph Autosize cells											
Y101		0.882861912250519									
1	A	B	C	D	E	F	G	H	I	J	K
1	Wavelength	Minimum	Maximum	Difference		1	2	3	4	5	
2	200	0.2703654	0.3070233	0.0366579		0.3054698	0.2720242	0.2717028	0.2703654	0.2740775	0.308
3	209.09091	0.2688055	0.3087274	0.0399219		0.3068506	0.2705401	0.2702037	0.2688055	0.2726914	0.3081E
4	218.18182	0.2654737	0.3077937	0.04232		0.3056543	0.2672534	0.2669081	0.2654737	0.2894646	0.30714
5	227.27273	0.2605313	0.3049132	0.0443819		0.3025199	0.2623394	0.2619884	0.2605313	0.2645898	0.30417
6	236.36364	0.2578843	0.3051523	0.047268		0.3024338	0.2597468	0.259385	0.2578843	0.262069	0.30431
7	245.45455	0.2553825	0.3047330	0.0493505		0.3017643	0.2572792	0.2569106	0.2553825	0.259647	0.30381

If the x- and y-columns can be marked in one contiguous selection, you can do so like in this example:

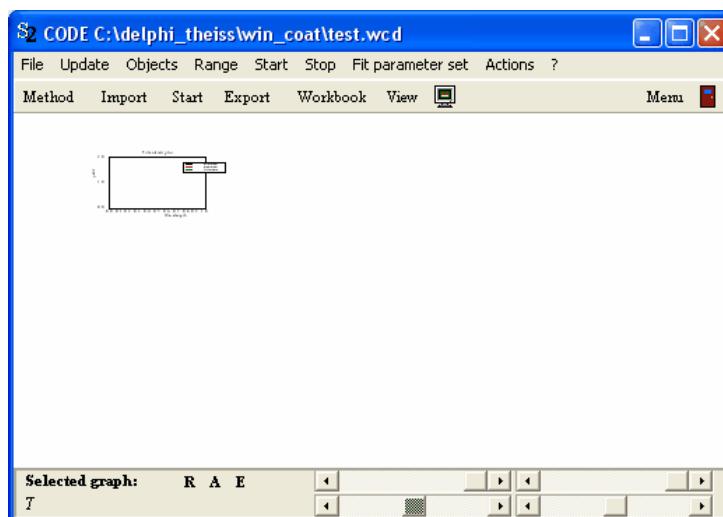


	A	B	C	D	E	F
1	Wavelength	Minimum	Maximum	Difference		
2	200	0.519992888	0.5904949307	0.07050204277		0.5858036
3	211.5577889	0.362208873	0.477775842	0.1155669689		0.4350540
4	223.1155779	0.2819012105	0.3568509221	0.07494971156		0.3273553
5	234.6733668	0.260268122	0.3017481565	0.04148003459		0.28898340
6	246.2311558	0.260756284	0.2801007032	0.01934441924		0.2790335
7	257.7889447	0.2677515447	0.2867878973	0.01903635263		0.2846925
8	269.3467337	0.2925923765	0.3227986991	0.03020832267		0.31394971
9	280.9045226	0.3214424253	0.367166996	0.04572457075		0.3519202
10	292.4623116	0.3606068492	0.4201025963	0.05949574709		0.3991963
11	304.0201005	0.4378893971	0.5093684793	0.07147908211		0.48361281
12	315.5778894	0.514075458	0.5981454253	0.08406996727		0.56769579
	227.1255721	0.5575926244	0.5607110245	0.00311040277		0.5220420

After the cells are selected, you have to open the list of views in the main window of SCOUT (**Object|Views**), select the view which shall contain the new graph and open its object list (**Edit** or **right-click**):

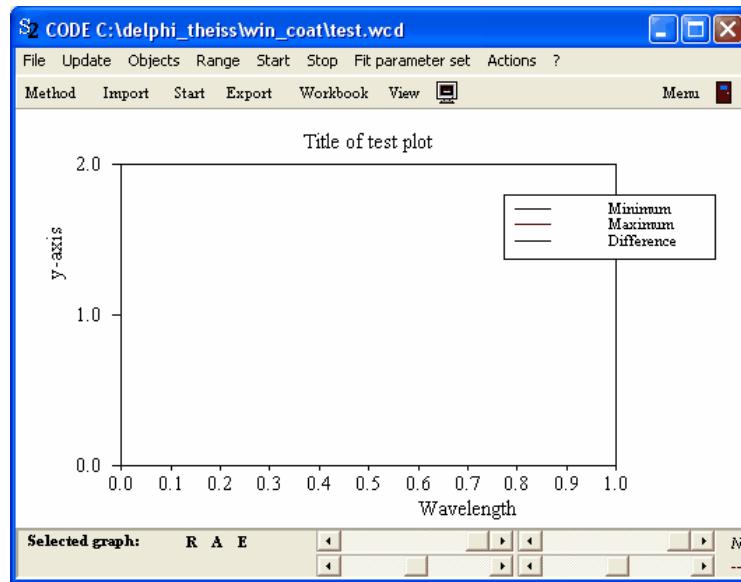


Now create in this list an object of type 'Workbook display (columns)' and **Update** the display in the main window of SCOUT:



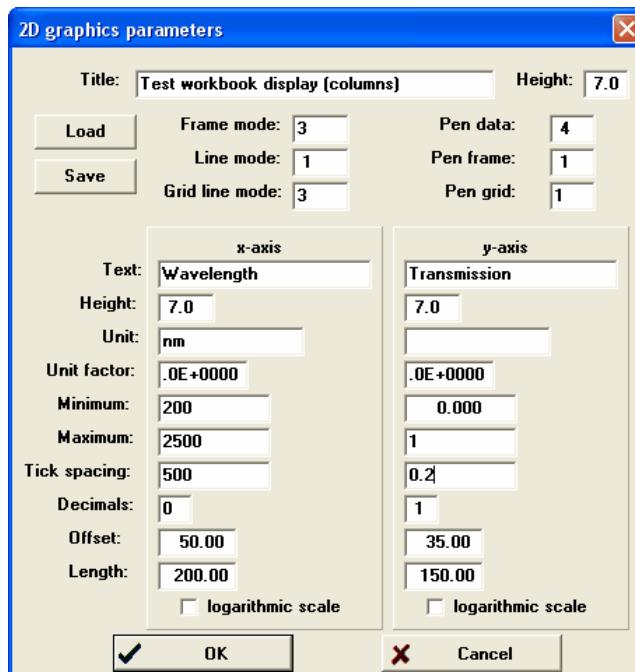
Obviously, this graph needs some fine-tuning!

First of all, you can change the size of the object by dragging while the **right mouse button** is pressed down. You can drag the object to a new position within the view using the mouse and pressing the **Caps key** and the **left mouse button** simultaneously. Try to manage something like this:

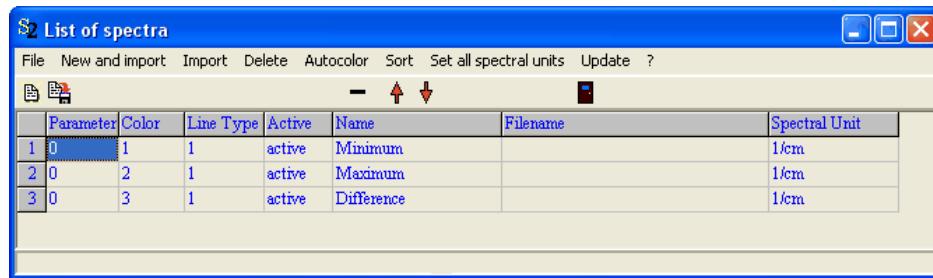


Now, click the graph while the **Ctrl-key** is pressed down in order to get access to all the graphics parameters. You will go through three dialogs if you have selected multiple columns of y-data. If there is only one column of y-data the last two dialogs are skipped.

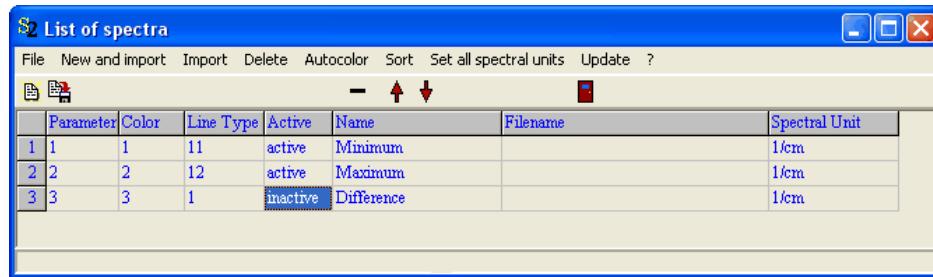
The first one gives you the chance to modify the graphics parameters like in many other SCOUT objects. See the SCOUT graphics course for the meaning of all the parameters. In this example, I have made the following settings:



When you have closed this dialog the next one opens which shows a list of all the columns of y-data:

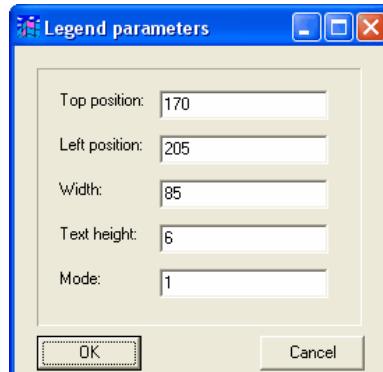


In this dialog (borrowed from Collect objects) you can ignore most of the items. In this context you should use the dialog to change the color and line type of each set of y-data (see the graphics course for available colors and line-types). With a **right-click** you can toggle the status '**Active**'. Only 'active' data sets are displayed in the graph. Try this assignment:



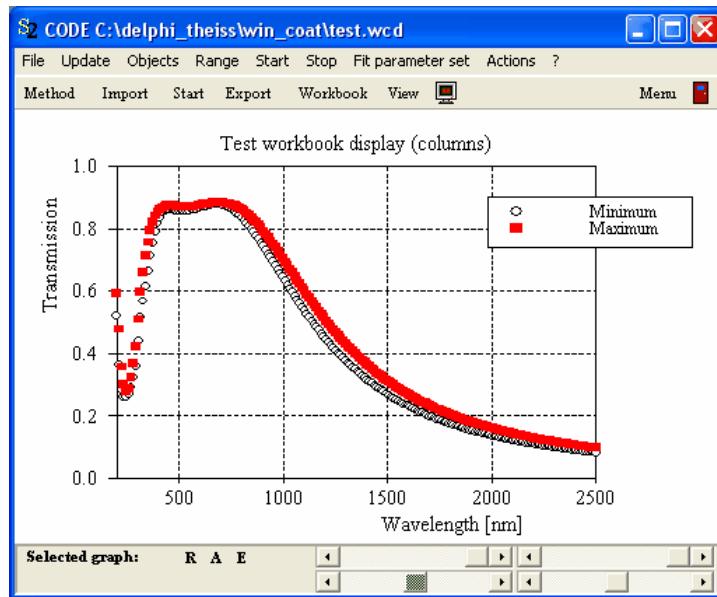
and then close the dialog.

Finally the dialog controlling the appearance of the legend is opened:



Here you can change the position and the width of the legend box. All numbers are to be given in mm, also the text height.

The mode parameter determines what is shown in the legend: A value of 0 suppresses the drawing of the legend, with 1 the legend will show the name of each data set, with 2 the numerical value assigned to the data sets are given, and with 3 both the names and the numerical values are shown. Once you have closed this dialog the main window re-draws the view according to the new settings:



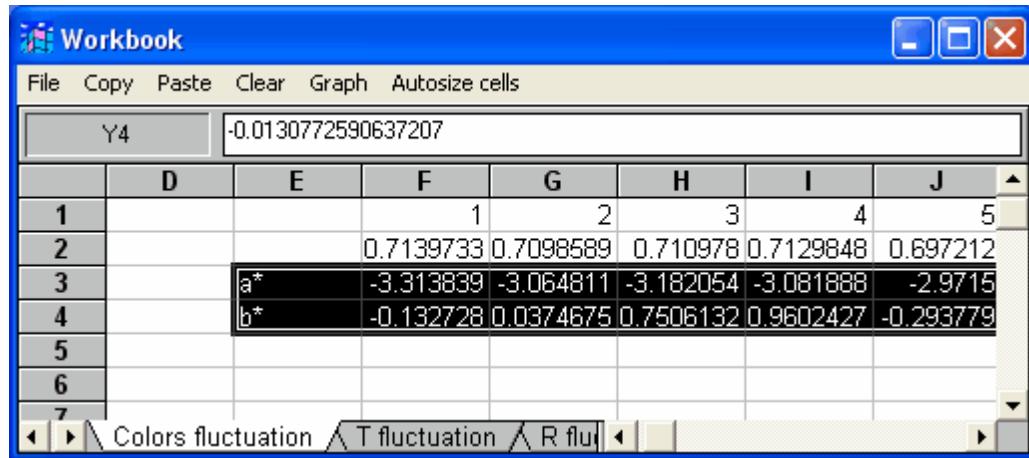
From now on the new 'Workbook display (columns)' object is tightly connected to the selected workbook cells. If the cell contents change due to recomputations an **update** in the main window will cause the display object to get and display the new data.

In the present version of this object type it is not possible to change the referenced workbook range afterwards. If you want to display the content of a different workbook range you have to create a new object as described in the top of this page.

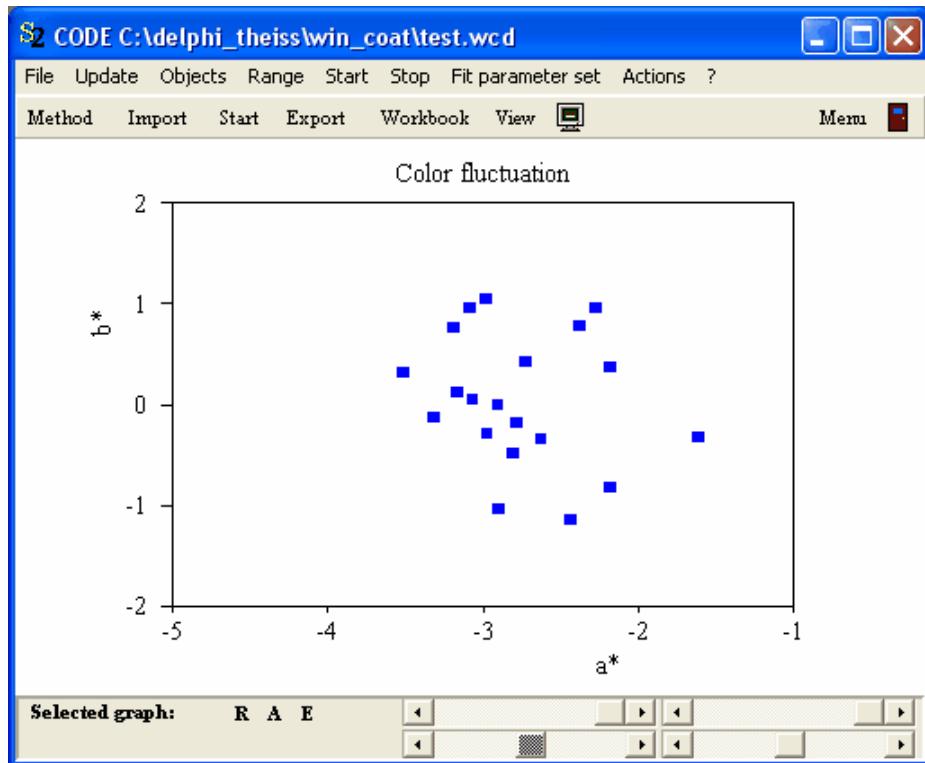
9.4.1.8 Workbook display (rows)

Objects of this type work exactly the same way as [Workbook display \(columns\)](#) objects, with the only difference that workbook rows instead of columns must be selected to deliver the data for the data sets.

Here is an example. After the following cells have been selected in the workbook



a creation of an object of type 'Workbook display (rows)' and appropriate graphics settings lead to the following picture:



9.4.1.9 Deviation view

This object type is used to display the current fit deviation in a view. It works exactly like [text view](#) objects, with the difference that the displayed text is given by the value of the fit deviation.

9.4.1.10 Rating view

This object type is used to display the current [rating](#). It works exactly like [text view](#) objects, with the difference that the displayed text is given by the current rating result.

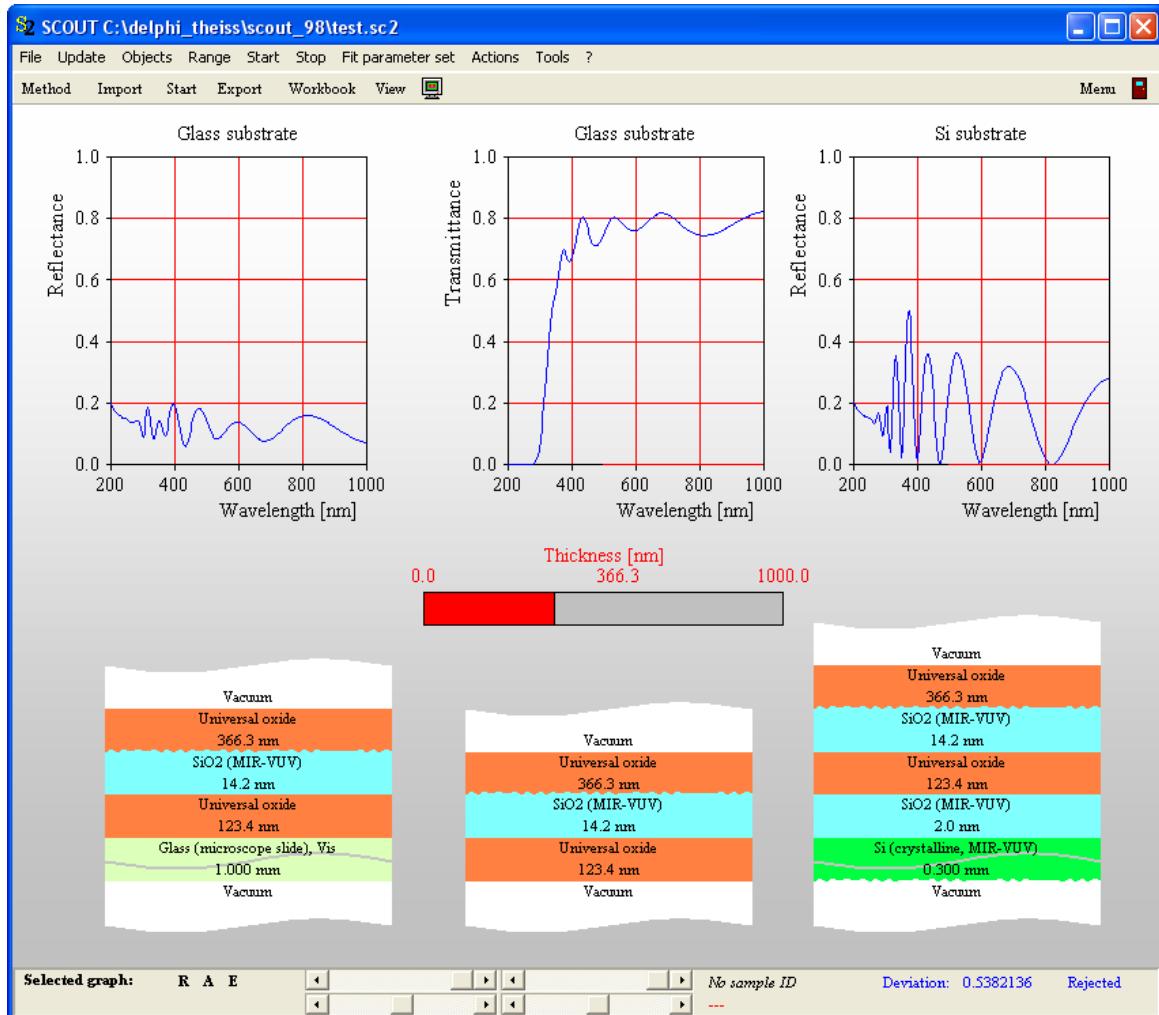
9.4.1.11 Sample view

This object type is used to display the current sample identification in a view. It works exactly like [text view](#) objects, with the difference that the displayed text is given by the name of the current sample.

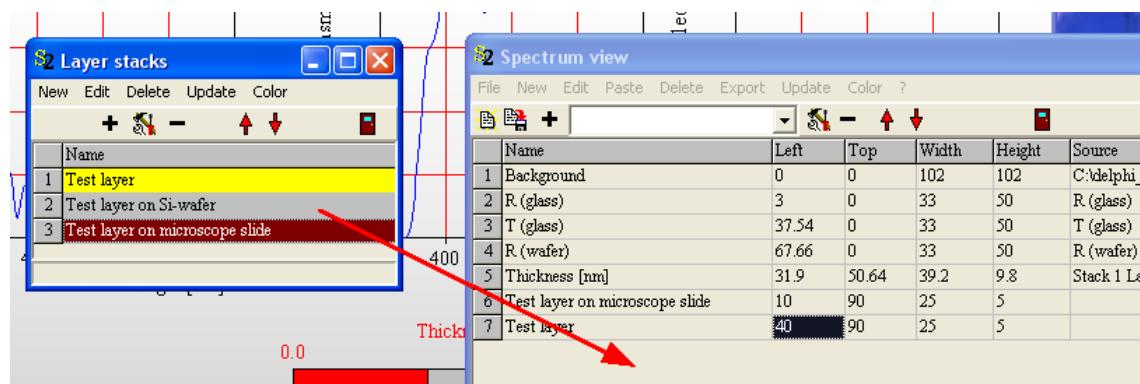
The sample identification is set when you execute the [Import command](#) in the speedbar of the main window, or in the **File|Options|Sample description** menu command.

9.4.1.12 Layer stack view

Starting with object generation 2.26, layer stacks can be drawn in views like in the following example:



In order to display a stack in a view, open both the list of view elements and the layer stack list. Then drag the wanted layer stack to the list of view elements and drop it there as indicated in the next graph:

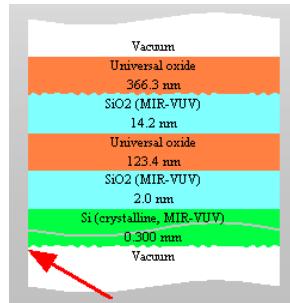


The layer stack view will be added to the list of view elements and you can set its size properly. The definition of position and size of layer stack view objects is a little different from other view elements.

Please read the following rules carefully before you try to use layer stack view objects.

Position

Most view objects are drawn in a rectangular box. The upper left corner of that box is usually defined by the 'Left' and 'Top' parameters. In order to properly position layer stack objects in a view, for this object type the position is defined differently: **The 'Left' and 'Top' parameters define the upper left corner of the bottom halfspace of the layer stack** as indicated below:



Size

The rectangle defined by 'Width' and 'Height' is not the rectangle including the whole view object as usual. Instead, for 'Layer stack view objects' it defines the size of a single layer in the layer stack. This is indicated by the black rectangle in the following graph:



Hotspot

The 'hotspot' to move, re-size, right- or double click a layer stack view object is indicated by the red rectangle in the area of the bottom halfspace shown below:



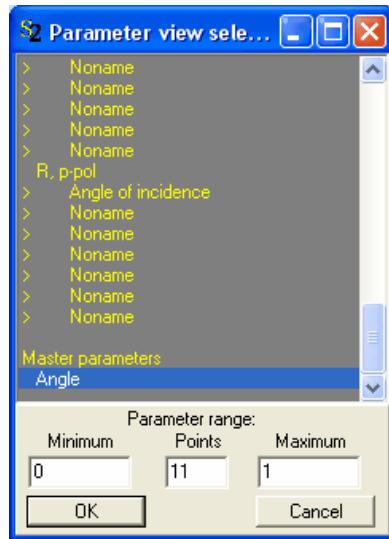
Please note that this is different from most other view objects. You can easily miss a layer stack view object working intuitively.

9.4.2 Action items

9.4.2.1 Parameter slider

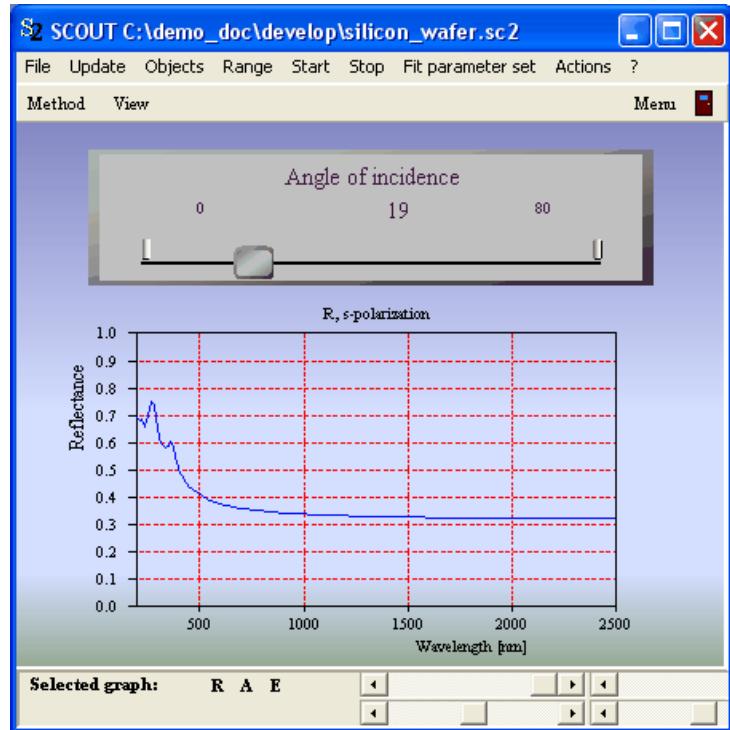
This object is used to display and modify the value of a model parameter in a view. The current parameter value is displayed as position of a vertically moving slider. You can move the slider with the mouse and modify the parameter value this way.

Once have created an object of this type, you have to select the model parameter in a dialog like the following (similar to the one for fit parameter selection):



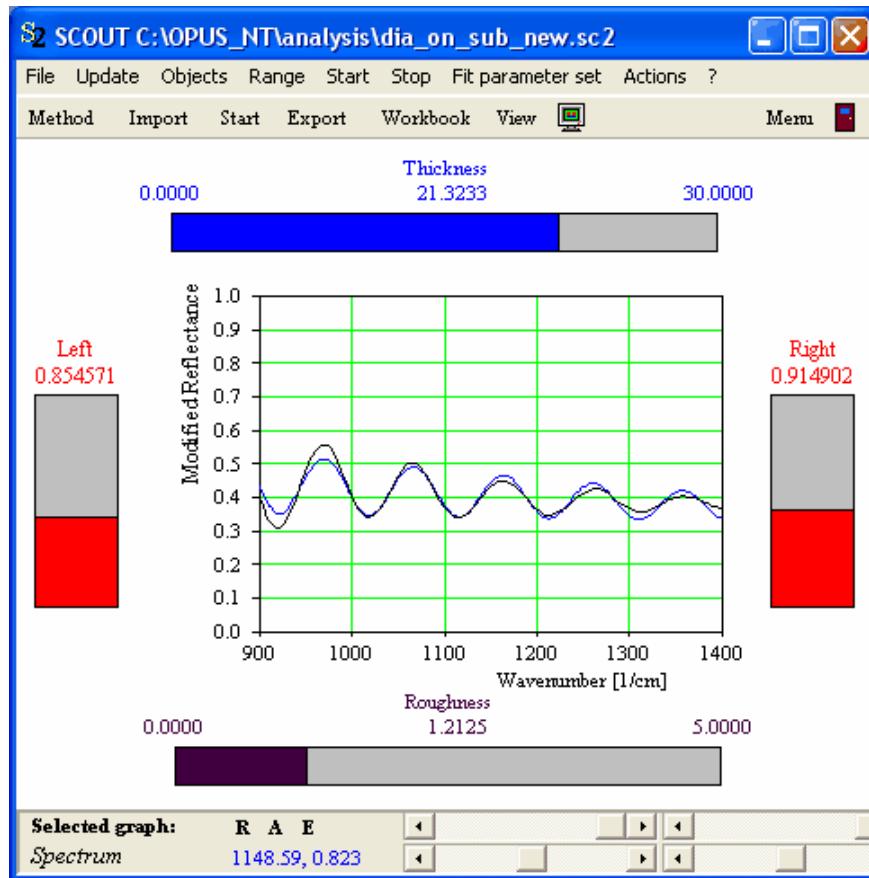
Please select one parameter only!

After you have closed this dialog, you have to select a color for the slider text (a dark color like black or blue is recommended) and the parameter variation range (minimum and maximum). Finally, you should enter an appropriate name for the slider object in the view list and adjust the size of the slider in the view to your needs. Here is an example (name of the object: 'Angle of incidence', parameter range: 0 ... 80 degrees, current value: 19 degrees):



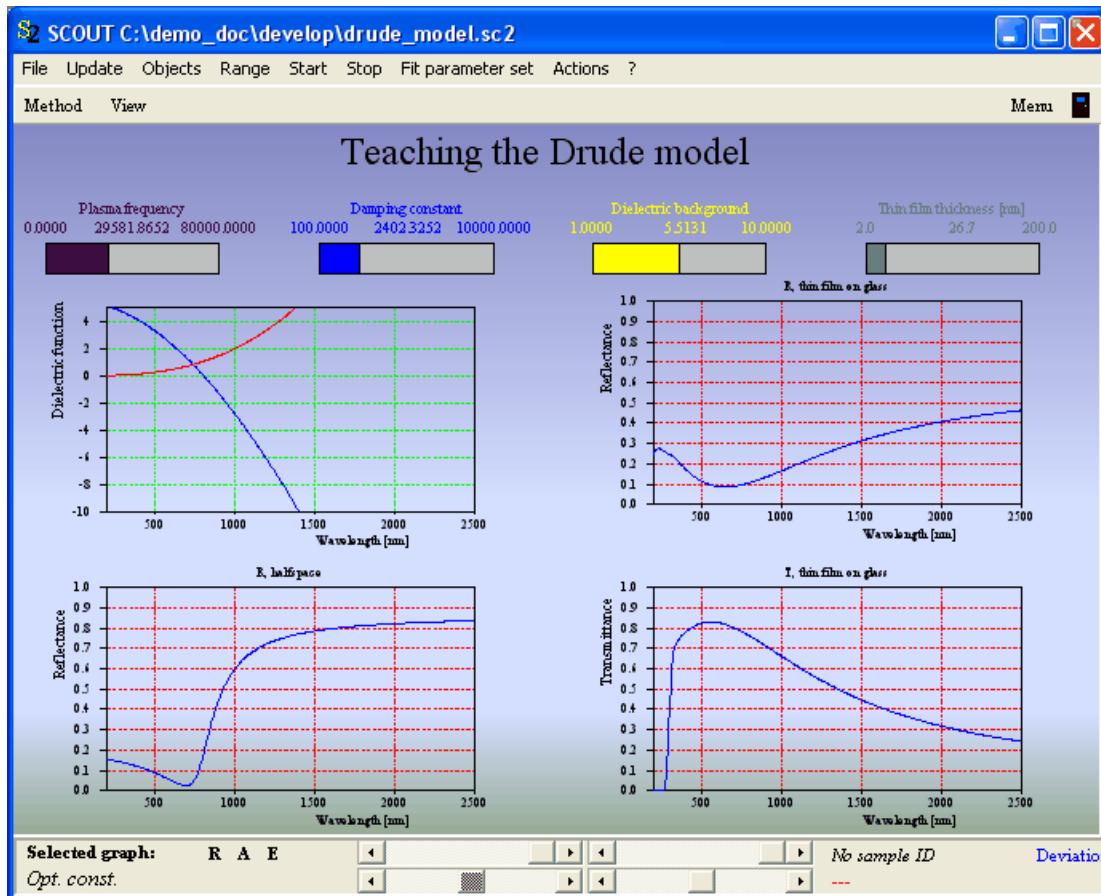
9.4.2.2 Vertical bar

Objects of this type are very similar to horizontal bars but they display the parameter value using a vertical bar. The example below shows two vertical and two horizontal bars in a view:



9.4.2.3 Horizontal bar

Objects of this type are created and work exactly as [Parameter slider](#) objects. The only difference is the display style in a view which is (no surprise) a horizontal bar. The parameter value can be 'moved' with the mouse. Bar displays are more compact than parameter sliders, so you can use more of them in a view like in this example:



9.4.2.4 Next view button

A click on this object will tell SCOUT to load the next view. The action is exactly the same as if the button [View](#) in the tool bar is pressed.

Besides the click action, objects of type 'Next view button' are like [Metafile view](#) objects. When they are created the graph called 'next_view_button.emf' in the subfolder pictures of the SCOUT directory is loaded into the object. You can change the displayed graph by **pasting** vector graphics from the clipboard or loading a different metafile graph using the **Edit** command (see the description of [metafile view objects](#) for details).

9.4.2.5 Previous view button

This object type behaves like '[Next view button](#)' objects, but switches to the previous instead of the next view. The default graph for the object is taken from the metafile 'previous_view_button.emf' in the subfolder pictures of the SCOUT directory.

9.4.2.6 Next configuration button

This object is a button which switches to the next configuration if SCOUT displays a [presentation](#). The default graph for the display of the button is taken from the metafile 'next_configuration_button.emf' in the subfolder pictures of the SCOUT directory.

9.4.2.7 Previous configuration button

This object is a button which switches to the next configuration if SCOUT displays a [presentation](#). The default graph for the display of the button is taken from the metafile 'next_configuration_button.emf' in the subfolder pictures of the SCOUT directory.

9.4.2.8 Acquire button

This object is a button which activates the acquiring of new spectra of all [spectrometers](#), like the menu command **Actions|Acquire**. The default graph for the display of the button is taken from the metafile 'acquire_button.emf' in the subfolder pictures of the SCOUT directory.

9.5 Printing views, clipboard copy

You can print the current view displayed in the SCOUT main window with the command **File|Print**. With **File|Printer setup** you can select and configure the printer that is going to be used for the next print action.

You can copy the currently selected view to the Windows clipboard using the command **Actions|Copy to clipboard** in the main window of SCOUT. The clipboard will contain the graph in the metafile format which can be pasted by many Windows programs, in particular text processing software.

Be prepared for the case that the graph will look quite small when you paste it e.g. into a Word document. Just resize it to the dimensions you want it to have.

9.6 Automatic view generation

The easiest way to create views is to activate one of the following automatic procedures that compose standard views. In most cases you can just work with them without further modifications. On the other hand, the automatically generated views can be a convenient basis for user-defined views with additional elements like explaining text, parameters or pictures.

The commands for automatic view generation are found in the **Actions** submenu of the [list of views](#). You can access this list by the command **Objects|Views** in the main window.

View of spectra

The command **Create view of spectra** does the following: First a [bitmap view](#) is created that contains a background image. SCOUT tries to find the file *view_spectra_background.bmp* and loads the image if the file exists. If we delivered a bitmap with this name and you don't like it, feel free to overwrite the picture with the one of your choice.

For each spectrum in the [list of spectra](#) SCOUT creates a [Field view](#) object and fills it with the right connection to the spectrum. The view objects are distributed in rows and columns. The numbers of rows and columns are computed from the number of spectra to be displayed.

Actual SCOUT versions create spectra views automatically when a configuration without any view is loaded.

View of optical constants

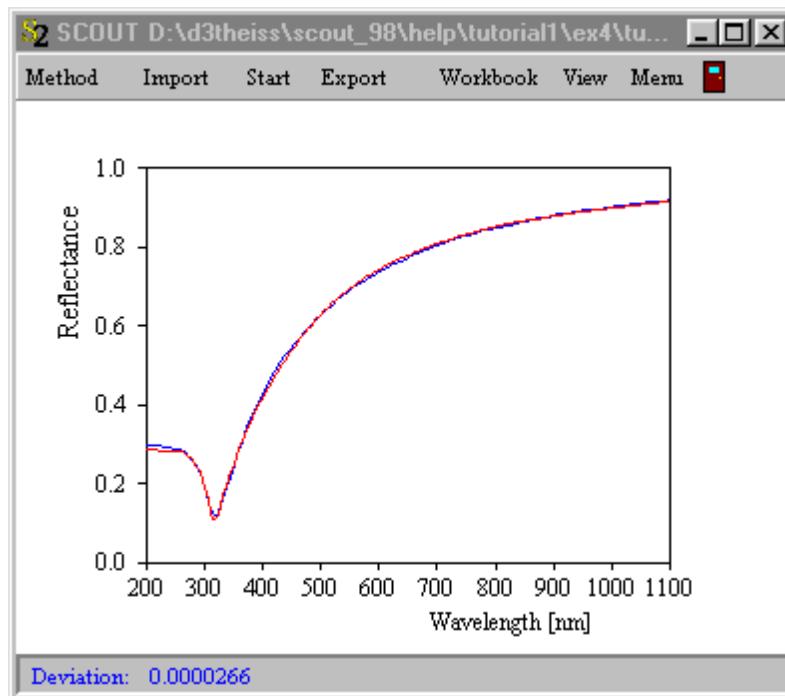
Very much like the view of spectra, you can create a view of the optical constants with the command **Create view of optical constants**. Again, first a [bitmap view](#) is created that contains a background image. SCOUT tries to find the file *view_opticalconstants_background.bmp* and loads the image if the file exists. If we delivered a bitmap with this name and you don't like it, feel free to overwrite the picture with the one of your choice.

For each item in the list of dielectric functions (with the exception of vacuum which is not displayed) SCOUT creates a [Field view](#) object and fills it with the appropriate connection. The view objects are distributed in rows and columns. The numbers of rows and columns are computed from the number of objects to be displayed.

9.7 View examples

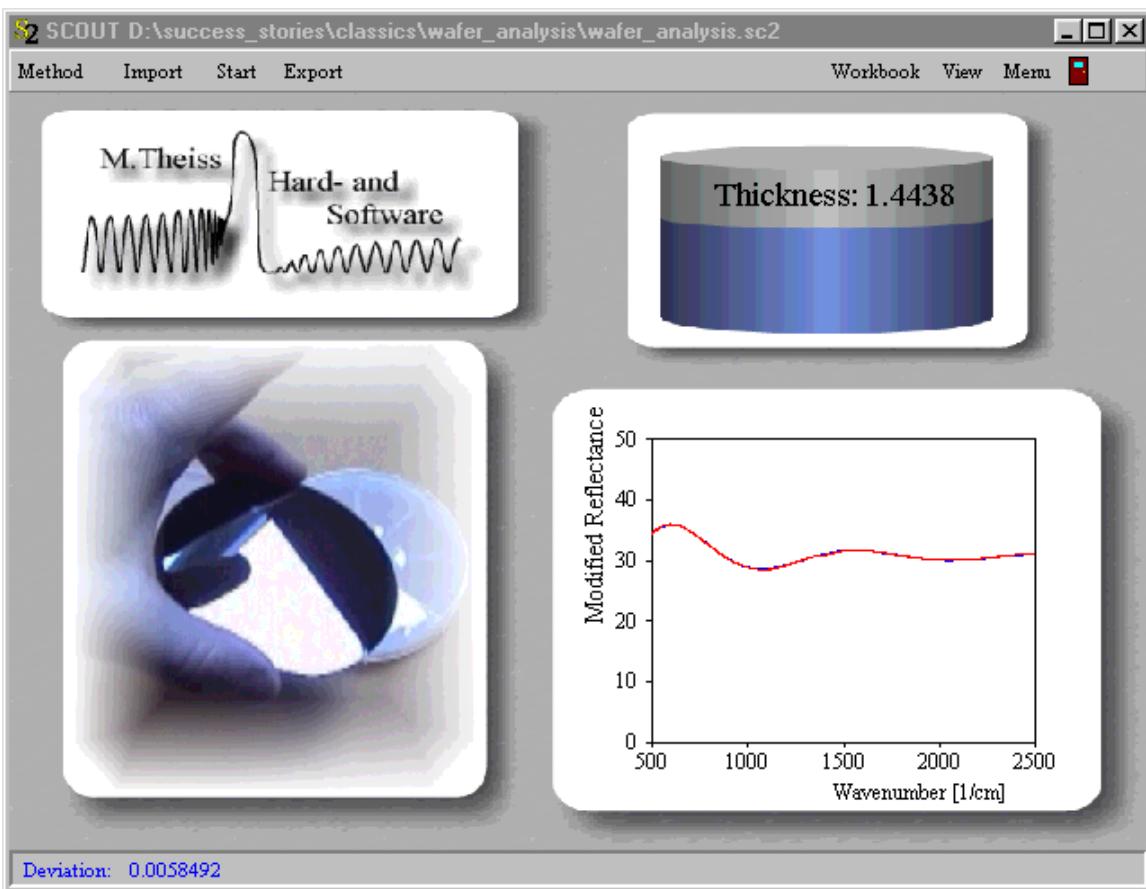
9.7.1 View example: Simple spectrum fit

The main window of SCOUT can be conveniently used to display a spectrum fit which is achieved by a single field view object. Note the difference of the graphics display if the data are updated: The main window update is much more smooth than the individual spectrum window which is due to a different graphics technique used in view objects.



9.7.2 View example: Fit and parameters

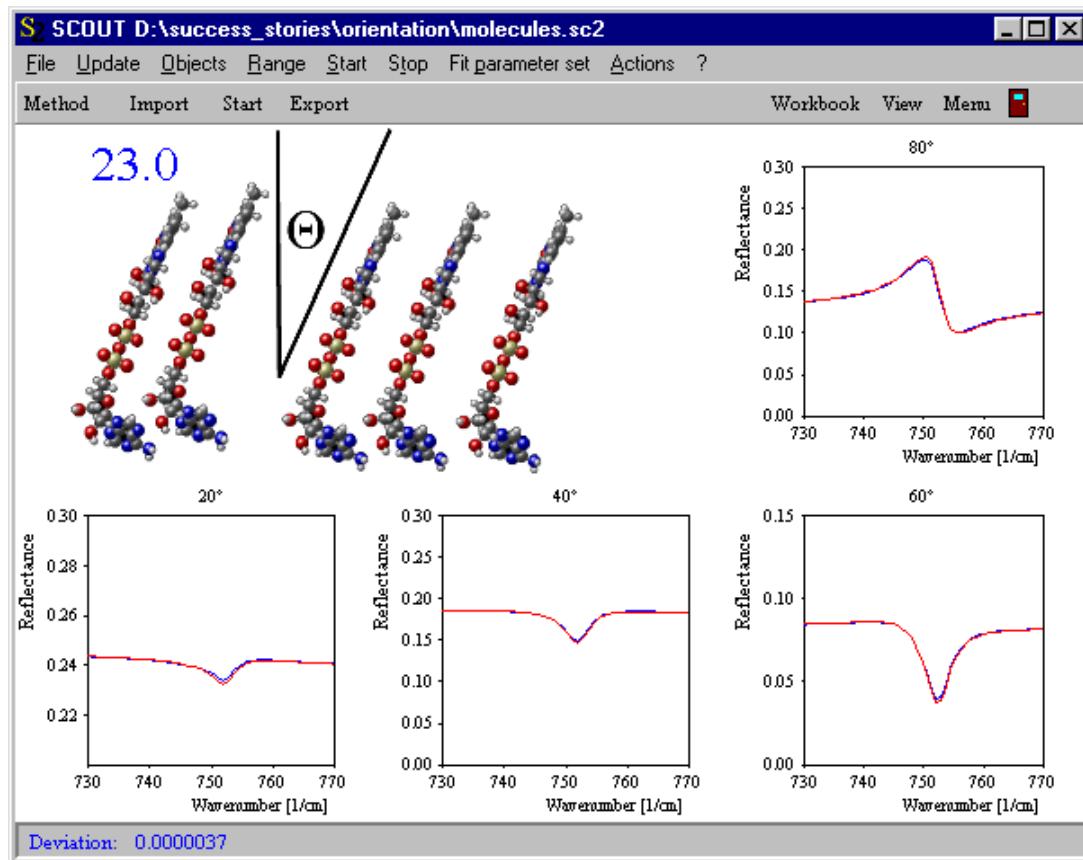
Here the main window of SCOUT shows the fit, i.e. the comparison of simulated and measured spectrum, through a field view object. The value of a model parameter (a layer thickness) is displayed using a parameter view object. The background is drawn using a bitmap view object.



With methods like the one sketched here you can solve routine problems without opening additional SCOUT windows: With the **Method** button you select the appropriate SCOUT configuration, the **Import** command lets you pick the experimental spectra (here you need just one), the **Start** button starts the parameter fit, and the **Export** button is finally used to save the results of the fit parameters in the SCOUT workbook.

9.7.3 View example: Display many spectra

Views are very important if your method requires the simultaneous analysis of several spectra. The following example shows how a view can display the simulation of 4 spectra and the values of some of the model parameters in one window:



9.8 Presentations

Views of several SCOUT configurations can be combined to presentations. A presentation is simply a sequence of SCOUT configurations. SCOUT will load one after the other and display the views in its main window.

Define the sequence of configurations

To prepare a presentation you have to copy all configurations to a folder. Then create a text file named **the_show.txt** which must contain - line by line - the names of the configurations to be loaded in the presentation. You can use the workbook to create this file. Type in the names of the configurations starting in cell A1 and then downwards like in the following example:

S2 Workbook				
File Copy Paste Clear Graph Autosize cells				
A4				
1	A	B	C	D
1	introduction.sc2			
2	optical constants.sc2			
3	final.sc2			
4				
5				
6				
7				

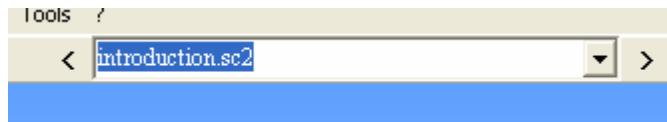
Use the SaveAs command and select the file format 'Tabbed text (*.txt)' to save the list to the text file **the_show.txt**.

Showing the presentation

To start the presentation use the menu command **Actions|Show presentation**. SCOUT will display a dialog which lets you select the folder that contains the presentation:



SCOUT will read the file `the_show.txt` in the folder and load the first configuration. In the speed button bar three additional items are displayed that let you navigate through the presentation:



The '`>`' command loads the next configuration, the '`<`' the previous one. You can switch to an arbitrary configuration using the dropdown box in the center. When SCOUT loads a presentation the dropdown box is filled with the content of the file `the_show.txt`.

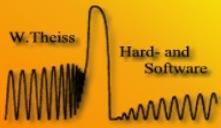
You can use the view buttons '[Next configuration button](#)' and '[Previous configuration button](#)' as navigation elements in views.

Tips and tricks

In order to achieve a nice presentation with smooth transitions from one page to the next, it is recommended that all configurations of a presentation have the same size. Full screen size is certainly the best choice.

If you just want to display the content of the view, you can hide the menu and the status bar at the bottom of the SCOUT window setting appropriate options (uncheck the menu commands **File|Options|Menu visible at start** and **File|Options>Status bar**).

SCOUT



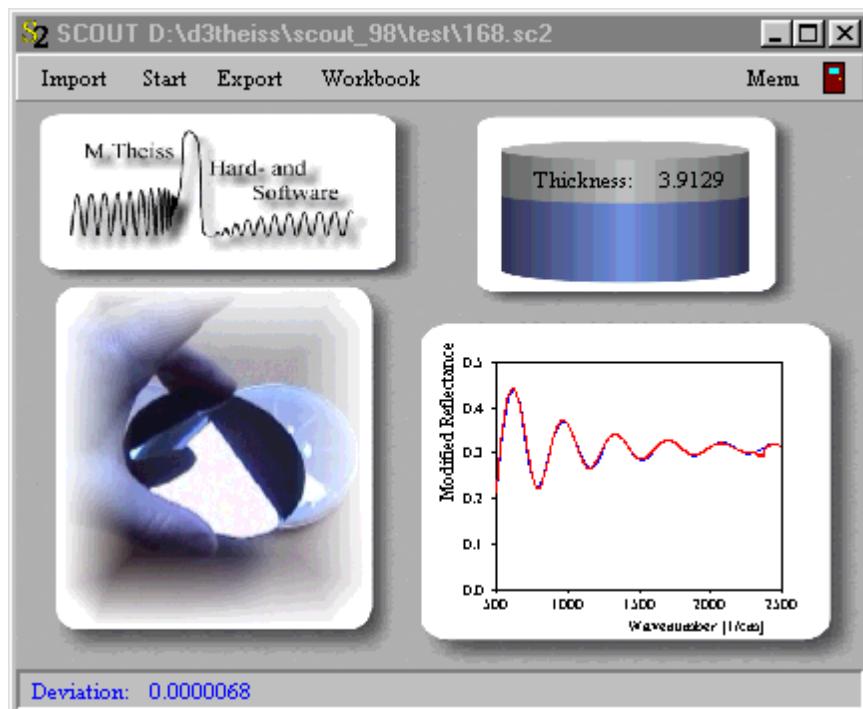
Part

X

10 Configuring SCOUT for routine work

After you have established an optical model for a certain class of samples and you are going to use SCOUT to analyze large amounts of samples in a routine manner, you may want to make the application of SCOUT for unexperienced users as easy as possible. To do so, you should work with the main window exclusively, showing what should be seen designing appropriate [views](#).

In addition, you should hide the menu in the main window and show only a few commands in the toolbar below the usual menu as shown in the following example:



The appearance of the toolbar commands can be controlled in the main menu using **File|Options|Toolbar**. Check those items that you want to appear in the toolbar.

At present you can activate and de-activate the following items:

Method button

This is the same as the **File|Open** command. Execute it to load an analysis method (configuration) that contains the complete model, the required views and the fit strategy to obtain the wanted parameters from the fit.

Acquire button

This command activates the recording of new spectra if spectrometers are defined in the list of spectrometers. Refer to the section about [Data acquisition](#) for further details.

Import button

This command is used to import new experimental spectra for the analysis from datafiles. First you are asked for the sample name, i.e. a short text identifying the current sample. This text is used in the report of results and may be displayed in a view by a [sample view](#) object. Then the import dialog for selecting the input spectrum is executed for each spectrum in the spectrum list. SCOUT follows the order of the spectrum list, of course.

Start/Stop button

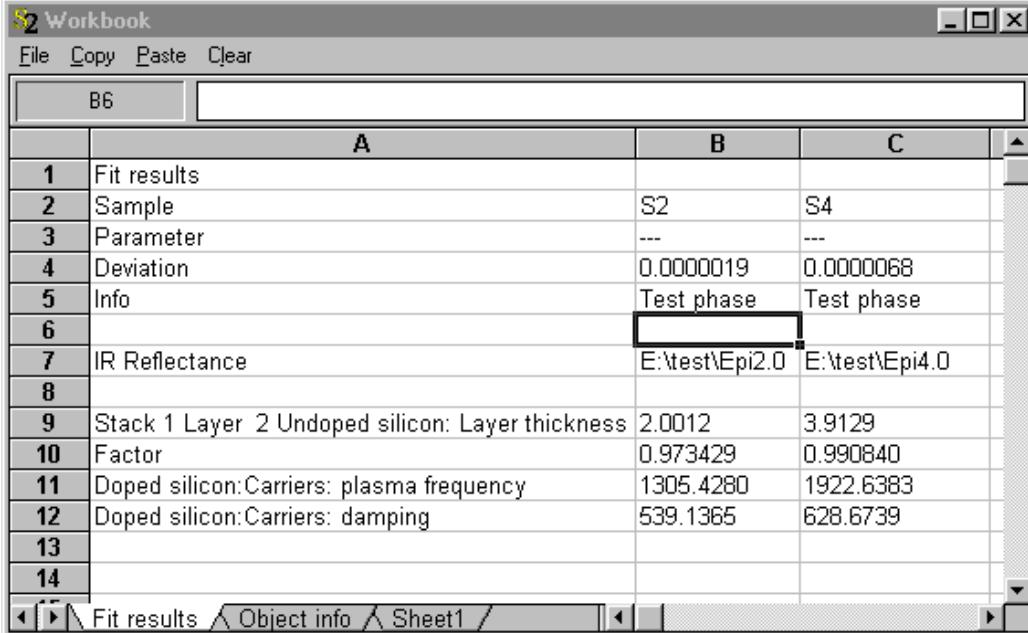
This starts the fit. After the start of the fit the text of the button is changed to 'Stop' and the same button is used to stop the fit.

Export button

This command writes the values of the current fit parameters to the table named 'Fit results' in the SCOUT [workbook](#). Before writing to the table SCOUT checks if the structure of the table matches the current configuration, in particular, the names of the fit parameters in the table and those in the current SCOUT configuration must agree. If not, SCOUT asks if it should erase the workbook content and start with a new page of results.

Workbook button

This command opens the workbook. After the analysis of some spectra the page 'Fit results' may look like this:



The screenshot shows a Windows-style application window titled "S2 Workbook". The menu bar includes "File", "Copy", "Paste", and "Clear". The main area is a spreadsheet-like table with columns labeled A, B, and C. Row 1 contains "B6" in column A and empty cells in B and C. Rows 2 through 14 contain various parameters and their values. Row 14 is empty. The last row, labeled "14", has a "Fit results" button in the first column and "Object info" and "Sheet1" buttons in the bottom navigation bar.

	A	B	C
1	Fit results		
2	Sample	S2	S4
3	Parameter	---	---
4	Deviation	0.0000019	0.0000068
5	Info	Test phase	Test phase
6			
7	IR Reflectance	E:\test\Epi2.0	E:\test\Epi4.0
8			
9	Stack 1 Layer 2 Undoped silicon: Layer thickness	2.0012	3.9129
10	Factor	0.973429	0.990840
11	Doped silicon:Carriers: plasma frequency	1305.4280	1922.6383
12	Doped silicon:Carriers: damping	539.1365	628.6739
13			
14			

Menu button

This command is used to switch on and off the main menu of SCOUT. This button is always visible, it cannot be hidden. However, you can protect the action of this button by setting a password if you want to avoid users switching on the menu.

Menu on/off

After loading the configuration the main menu can be switched on and off, according to the setting of the menu item called **File|Options|Menu visible at start**. Check or uncheck this menu item to tell SCOUT what it should do.

Password protection

If the main menu is switched off you can switch it on again pushing the **Menu** toolbar button (see above). Using **File|Options|Password** in the main window you can define a password that is required to display the menu. This may be used to avoid unauthorized configuration changes by unexperienced users.

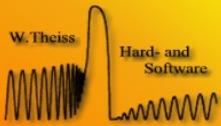
Automatic spectrum display after loading

Starting with object generation 1.9 (January 2001) SCOUT automatically opens all spectra in the spectrum list after loading a new configuration if the menu item **File|Options>Show spectra after loading** is checked.

Automatic slider display after loading

Starting with object generation 1.9 (January 2001) SCOUT automatically opens all fit parameter sliders after loading a new configuration if the menu item **File|Options>Show sliders after loading** is checked.

SCOUT



Part

XI

11 Spatial and time distributions, optical monitoring

11.1 Overview

SCOUT allows the computation of various **space and time distributions**, e.g. of the electric field amplitudes or the local absorption in layer stacks. Objects performing such kind of computations are collected in the list called 'Distributions'. You can open this list by a right-click in the treeview of SCOUT.

The list of distributions also provides objects for computing **optical monitoring charts**. This feature is not contained in the standard SCOUT program but must be purchased separately.

At present the following quantities are implemented:

[Spatial distribution of the electric field amplitude](#)

[Spatial distribution of the absorption of light](#)

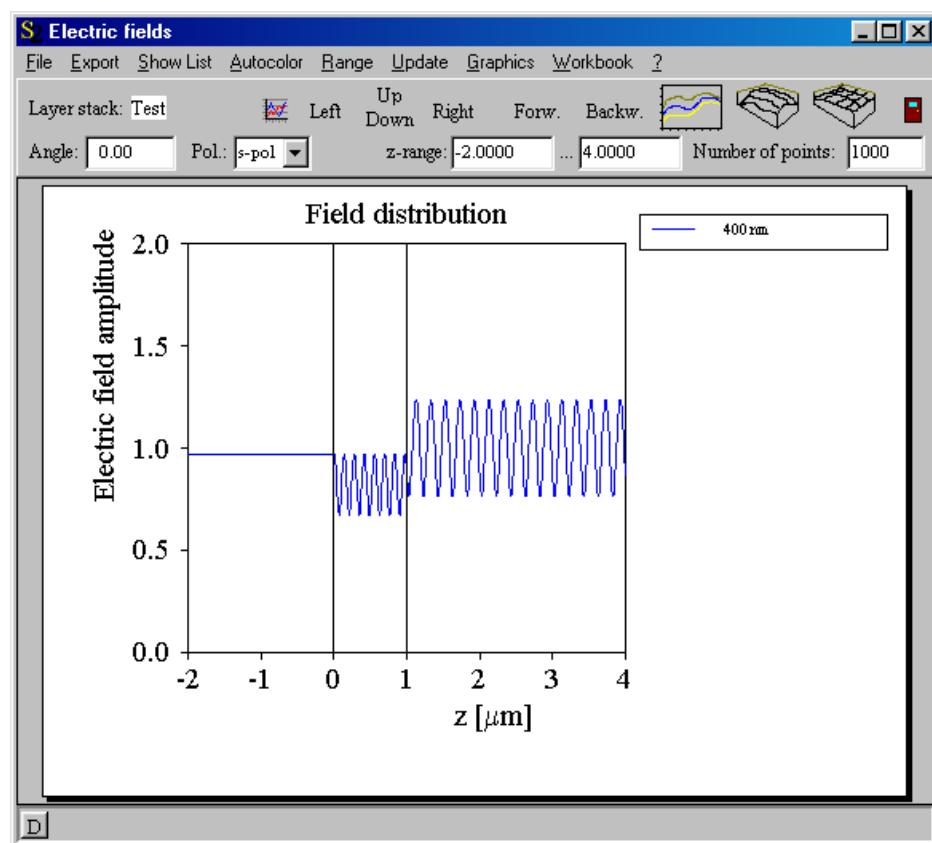
[Optical deposition control of layer growth: R vs. total thickness](#)

[Optical deposition control of layer growth: R vs. time](#)

[COME: Charts for optical monitoring of etching processes](#)

11.2 Electric field

Objects of type 'Electric field distribution' compute the spatial variation of the electric field amplitude in a layer stack. The main window of such an object looks like this:



The graph means the following: The horizontal (x-) axis displays the position in the layer stack. The last interface to the bottom halfspace is at zero. Starting at zero and going to the right, you move through all layers until you reach the top halfspace. Each interface is indicated by a vertical black line. The light is incident from the top, i.e. the incoming wave approaches from the right with an amplitude of 1.

The example picture given above shows the situation for a 1 micron film (SiO₂) and 400 nm wavelength: Above the layer (in the range 1 ... 4 microns) you see the superposition of the incoming and the reflected wave. In the layer (0 ... 1 micron) there is a superposition of all partial waves going from the right to the left with those moving from the left to the right. Below z = 0 there is only the transmitted radiation with a total amplitude that is constant in space.

The following parameters determine the result of the electric field computation:

Layer stack:

The white text field to the right of the label 'Layer stack' shows the name of the layer stack for which the computation is done. You can change the selection of the layer stack by a drag&drop operation: Make sure the list of layer stacks and its subbranches in the treeview are open. Open the electric field object with a right-mouse click (if it is not visible at the moment) and drag the wanted layer stack to the white text field where it is to be dropped.

Angle:

This is the angle of incidence of the incoming beam.

Pol.:

This is the polarization of the incident beam which can be s- or p-polarized.

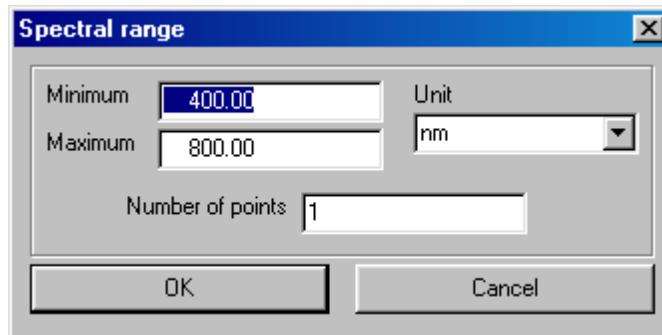
z-range:

Here you specify the spatial range for the electric field computation by entering the minimum and maximum z-values, and the number of points you want to use to sample the interval.

In the case of a 2D plot (like the one shown above) the z-range displayed may differ from the range in which values are computed. In all 3D plots the full computed range is shown always.

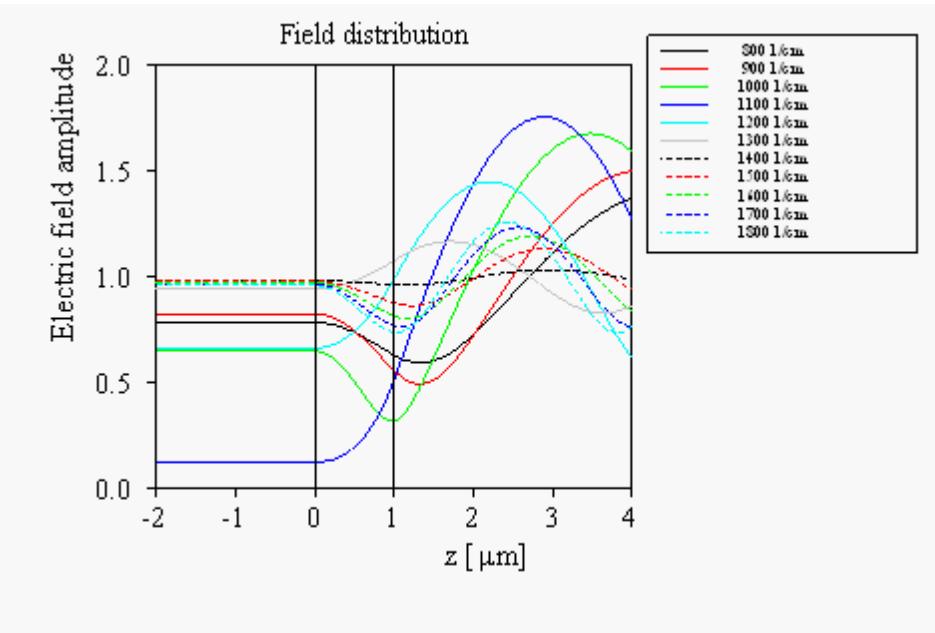
Range:

Using the menu command **Range** you can specify the spectral range for the computation. The usual dialog for spectral ranges is shown:



If the number of spectral points is set to 1 the electric field distribution is computed only for that spectral point. You can also use several spectral points to compute the field distribution for a whole sequence of spectral points.

The following example shows the result for a 1 micron glass plate in the infrared (800 ... 1800 1/cm) using 11 spectral points:

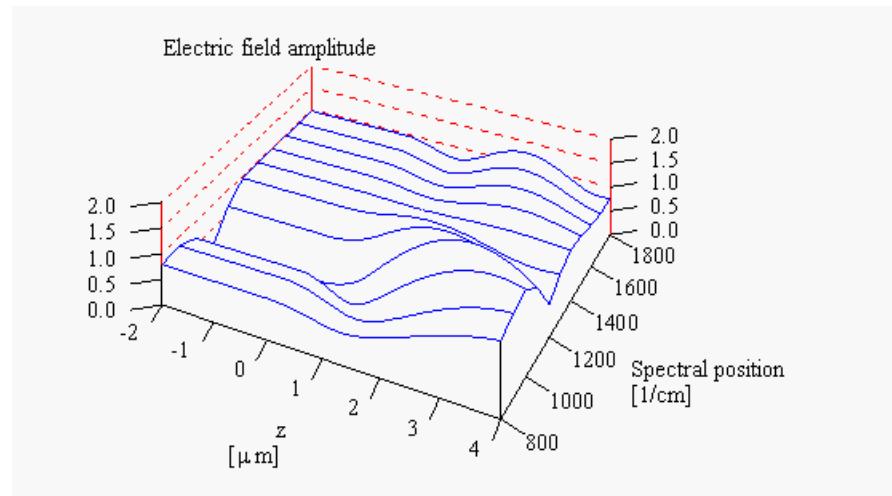


The individual lines are listed in a separate window which you can open using the **Show list** command:

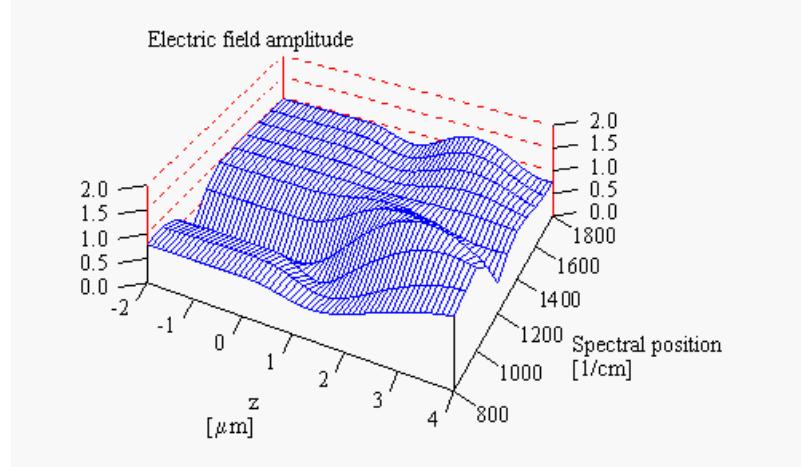
	Parameter	Color	Line Type	Active	Name	Filename
1	800	1	1	active	800 1/cm	800 1/cm
2	900	2	1	active	900 1/cm	900 1/cm
3	1000	3	1	active	1000 1/cm	1000 1/cm
4	1100	4	1	active	1100 1/cm	1100 1/cm
5	1200	5	1	active	1200 1/cm	1200 1/cm
6	1300	6	1	active	1300 1/cm	1300 1/cm
7	1400	7	1	active	1400 1/cm	1400 1/cm
8	1500	8	1	active	1500 1/cm	1500 1/cm
9	1600	1	1	active	1600 1/cm	1600 1/cm
10	1700	2	1	active	1700 1/cm	1700 1/cm
11	1800	3	1	active	1800 1/cm	1800 1/cm

For each graph you can set the color and the line type that are used for the plot. You can type in new values or cycle through all values by right mouse clicks.

If the distribution is computed for several spectral points you can display the results also in 3D pressing the appropriate buttons. You can use a line plot with lines along the x-axis



or a grid plot:



Automatic re-computation

If the menu item **File|Options|Automatic update** is checked SCOUT re-computes the electric field distributions if the model is changed. This can be used to elegantly inspect the influence of a model parameter like a layer thickness or an optical constant parameter on the field distribution: Just move a fit parameter slider and instantly see what changes in the fields occur.

On the other hand, electric field computations and in particular 3D plots can consume quite a bit of time, and automatic re-computation should definitely be switched off during parameter fitting.

Export of numerical values

The computed distributions can be exported to convenient tables using the **File|Report** menu command. The numbers are written to the workbook which you can open by the command **Workbook|Open workbook**. You'll find the table in the worksheet called 'Last distribution'.

Important restrictions:

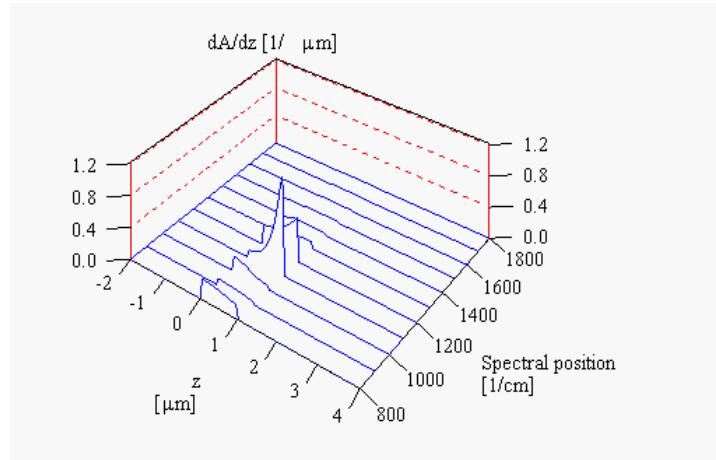
The present version of this object type requires that the transmission is slightly above zero. Otherwise you will get an error message and the computation is stopped. We are working on this problem.

Only coherent and isotropic layers are allowed at present.
No concentration gradient layers are allowed at present.

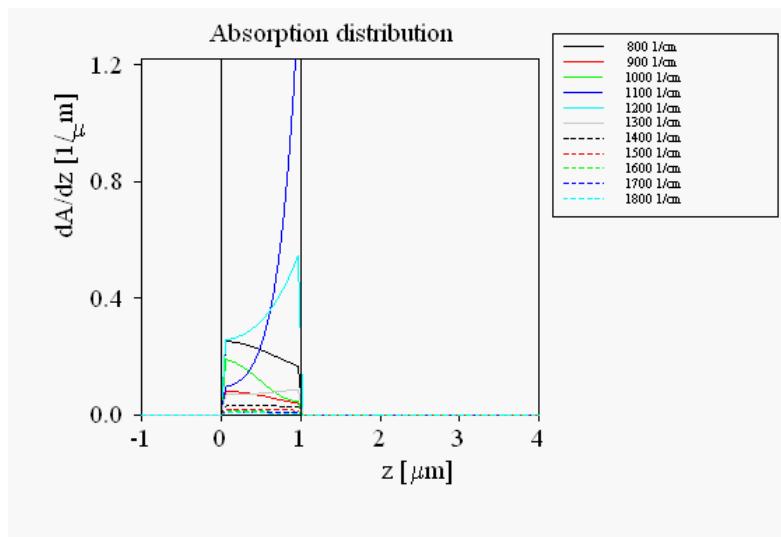
11.3 Absorption

This object computes the local absorption in the layer stack. It works exactly the same way as the objects that compute [electric field distributions](#).

The following picture shows the result for the infrared absorption of the 1 micron glass plate that was already used and shown in the [electric field distribution](#).



The same data in 2D:



Important restrictions:

The present version of this object type does not work properly for p-polarization. We are working on this problem.

11.4 Optical monitoring

11.4.1 R vs. total thickness during deposition

Optical monitoring charts displaying the reflectance of a layer stack vs. total layer stack thickness are computed by objects of type 'Optical growth monitor' that are created in the list of distributions. These objects are handled almost exactly like [Electric field distributions](#).

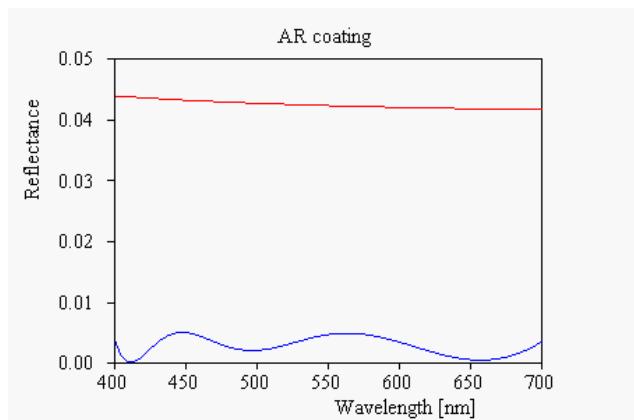
The difference is, of course, that the reflectance instead of the electric field is computed. In addition,

the meaning of the x-axis differs. Here x represents the total thickness of the layer stack. If x is negative, SCOUT computes the reflectance of the blank substrate. Starting at x=0 SCOUT computes how the reflectance of the layer stack changes while the layers are growing. Like in the deposition, SCOUT starts a new layer if the present one has reached its final thickness. As an example, an anti-reflection coating with 4 layers (note that this is just a demo to demonstrate the optical monitoring and not a professional filter design) and the following structure is used:

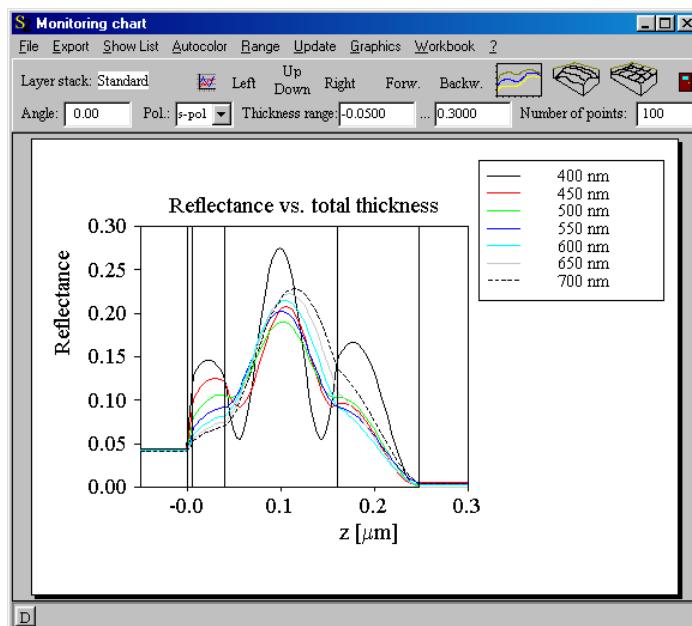
	Type	Material	Thickness
1	Halfspace	Vacuum	
2	Thin film	SiO ₂	87.6 nm
3	Thin film	ZrO ₂	121.2 nm
4	Thin film	SiO ₂	35.1 nm
5	Thin film	TiO ₂	4.7 nm
6	Halfspace	Glass Type BK7	

The optical constants are all taken from the SCOUT database.

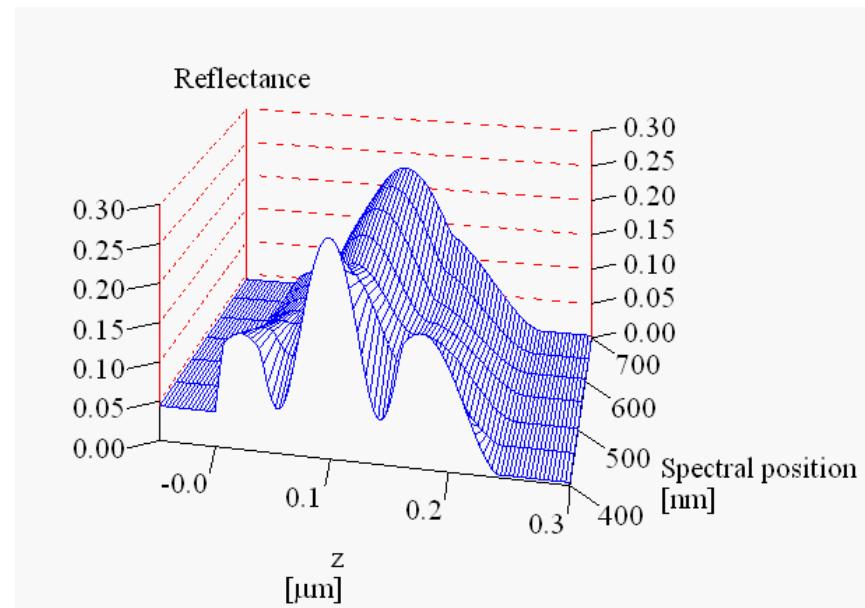
The reflectance spectra of the blank substrate (red curve) and the final coating are shown in the next graph:



The optical growth monitor shows the development from the blank substrate to the final reflectance. You can choose a 2D picture



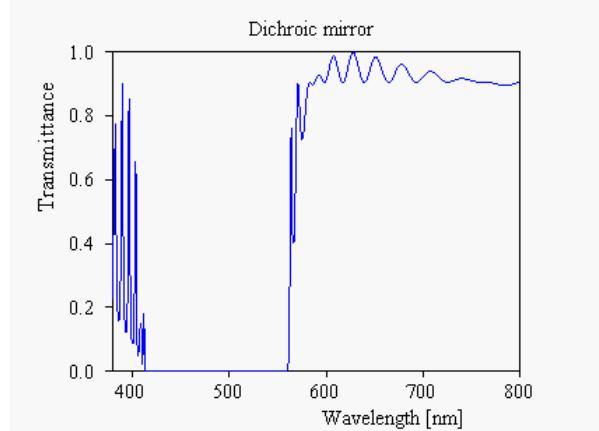
or 3D representations:



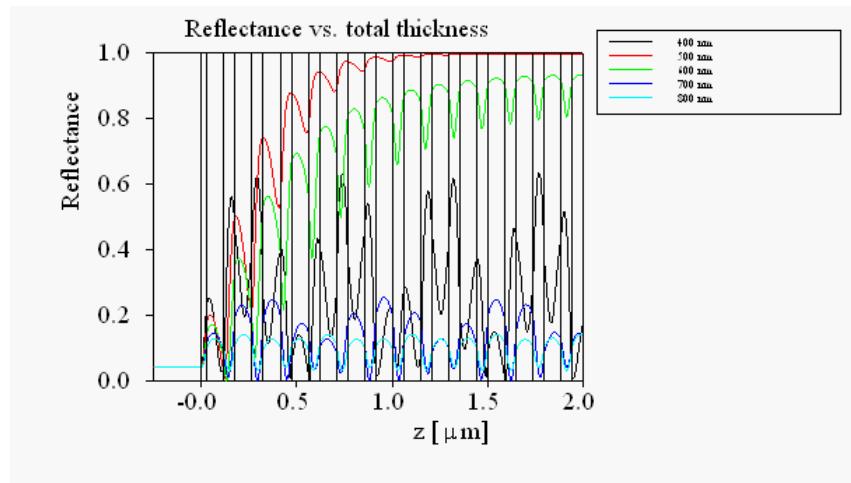
A more advanced example is a dichroic mirror (design published in [\[Park 1997\]](#)) of this structure:

1	Halfspace	Vacuum				
2	Thin film	ZrO ₂	29.2 nm			
3	_Begin superlattice		Repeat	27	times	
4	Thin film	SiO ₂	88.9 nm			
5	Thin film	ZrO ₂	58.5 nm			
6	_End superlattice					
7	Thin film	SiO ₂	88.9 nm			
8	Thin film	ZrO ₂	29.2 nm			
9	Halfspace	Glass Type BK7				

The final transmittance spectrum at 45 deg angle of incidence and s-polarisation is this:



Optical monitoring data for reflectance at normal incidence of light are shown in the next graph:



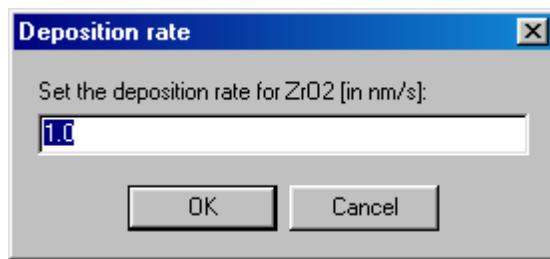
Based on data like this, the monitoring strategy is to be developed in order to find the right moment to stop the growth of a layer and to start with the next material. In addition to knowledge of the reflectance vs. total thickness, one may be interested also in the time dependent reflectance signal which is computed by a similar SCOUT object discussed [below](#).

11.4.2 R vs. time during deposition

Objects of type 'Opt. growth mon. (time)' (managed in the SCOUT list of distributions) compute the reflectance of a layer stack vs. time while the complete stack builds up the substrate. These objects are very similar to the previously discussed objects of type '[Optical growth monitor](#)'.

Deposition rates

To compute charts displaying the reflectivity of a stack vs. time during the layer deposition the deposition rate for each material in the stack must be known. SCOUT stores deposition rates of materials in a text file that must be in the same directory as the program file (scout_98.exe). At program start SCOUT reads the file deposition_rates.txt and enters the data decoded there into an internal table. If you try to compute monitoring data for a material with unknown deposition rate, SCOUT will ask you to type in that missing number in a dialog like this:



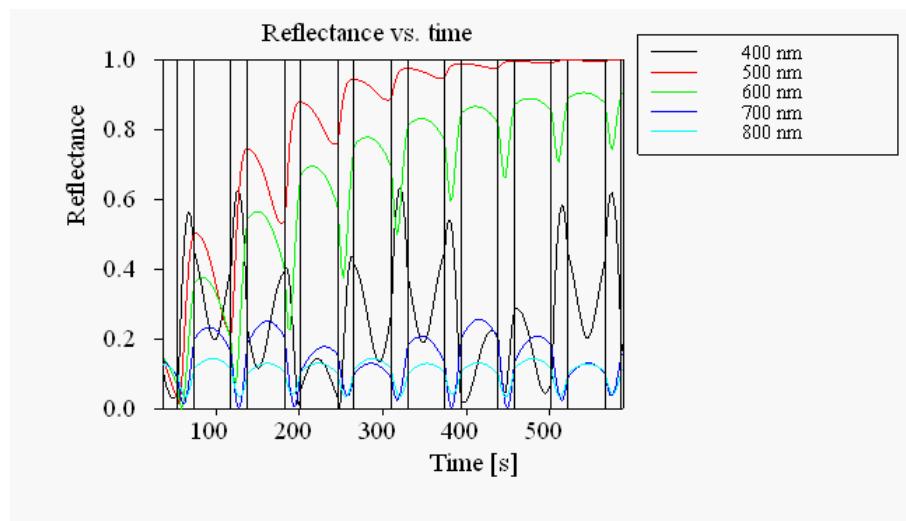
As you see in the dialog, deposition rates must be specified in nm/s. SCOUT will enter the new value into its table. Materials are identified by their name in the list of dielectric functions (which is also the caption of the dielectric function window). When you shut down the SCOUT program the current table values of deposition rates are stored in the file deposition_rates.txt for future use. Usually you do not have to inspect this file at any time. However, if you want to look or modify the current values of the deposition rates you can open the SCOUT table with the menu command 'Rates'. It looks like this:

S2 Workbook		
	A	B
1	Material	Deposition rate [nm/s]
2	(Al(0,6)Ga(0,4))0.5 In(0.5)P	1.
3	ZrO ₂	3.
4	SiO ₂	2.
5		
6		
7		
8		
9		
10		

Deposition rates ▾ Etch rates ▾

Here you can enter new values in the column labeled 'Depositon rate'. It is not recommended to enter new names in the 'Material' column because SCOUT identifies a material by its name which works reliably only in the case of perfectly correct spelling. So wait until SCOUT asks you to enter a deposition rate for a new material.

With the (fictitious) values given in the table above the reflectance vs. time curve for the deposition of the dichroic mirror example ([see above](#)) is shown below:



Note that SCOUT neglects any time for switching between the growth of the individual materials. Hence the SCOUT time for growing a stack is the minimum that you can have.

11.4.3 R vs. time during etching

Objects of type 'COME etching monitor' (managed in the SCOUT list of distributions) compute the reflectance of a layer stack vs. time while the complete stack is reduced by an etching process. COME stands for **Charts for Optical Monitoring of Etching**. These objects are very similar to the previously discussed objects of type '[Opt. growth mon. \(time\)](#)'. In fact, COME objects just do the reverse, starting the computation for the complete stack and finishing at the blank substrate.

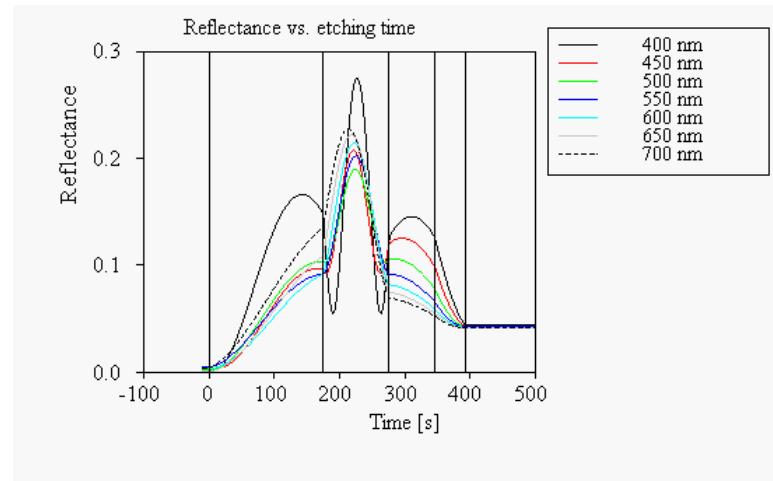
Etching rates

To compute charts displaying the reflectivity of a stack vs. time during etching the etching rate for each material in the stack must be known. Like in the case of deposition rates SCOUT stores etching rates of materials in a text file that must be in the same directory as the program file (scout_98.exe). The file holding etching rate information is called etch_rates.txt. As before, you do

not have to look at this file but you can wait until SCOUT asks you to provide etching rate information for a certain material.

Example

The COME result for the [anti-reflection coating](#) discussed above is this:



The following etch rates (based on pure imagination) have been used:

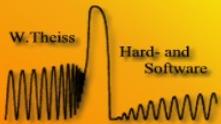
Material Etch rate [nm/s]

SiO₂ 0.5

ZrO₂ 1.2

TiO₂ 0.1

SCOUT



Part

XII

12 Working with the workbook

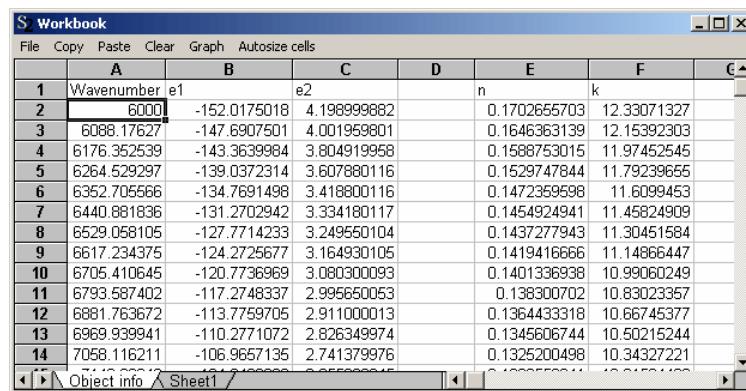
12.1 The SCOUT workbook

SCOUT is equipped with an internal spreadsheet called the workbook. The workbook may contain several worksheets (or workbook pages) each of which is a grid of cells that may contain text, numbers, or formulas that compute the cell content (using the content of other cells, in most cases). The workbook serves as temporary storage for many SCOUT operations, and it is a very convenient tool to exchange data with other applications, like e.g. Microsoft Excel. The workbook is saved as part of a SCOUT configuration, i.e. its content is stored if you save your SCOUT configuration in the main window with File|SaveAs.

If you want to keep the content of only the workbook for later use you can store it explicitly using the [file commands of the workbook](#) itself.

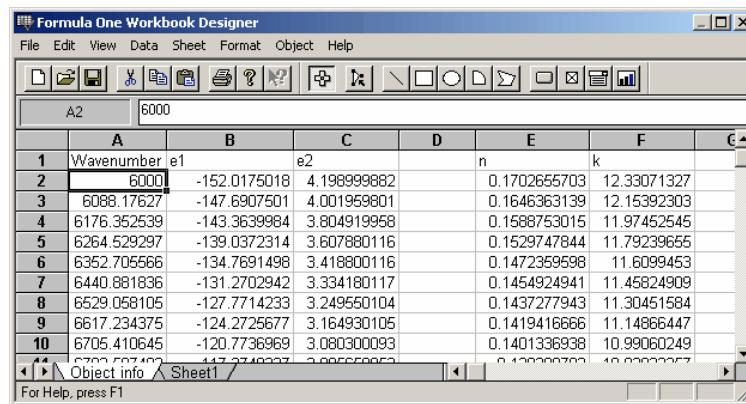
Handling the workbook

The normal appearance of the workbook is the following:



A screenshot of the SCOUT Workbook window. The window title is "Workbook". The menu bar includes File, Copy, Paste, Clear, Graph, and Autosize cells. The main area is a grid of cells labeled A through F and 1 through 14. Cell A2 contains the value "6000". Other visible data includes Wavenumber values (e.g., 6088.17627, 6176.352539, 6264.529297, etc.) and corresponding e1, e2, n, k values. The bottom of the window shows tabs for "Object info" and "Sheet1".

You can modify cell colors and formats, add, delete and print worksheets by right mouse button double clicks somewhere in the workbook. The following window opens:



A screenshot of the Formula One Workbook Designer window. The window title is "Formula One Workbook Designer". The menu bar includes File, Edit, View, Data, Sheet, Format, Object, and Help. The toolbar has various icons for cell selection, copy/paste, and format. The main area is identical to the Workbook window, showing the same grid of data with cell A2 containing "6000". The bottom of the window shows tabs for "Object info" and "Sheet1".

Using the commands Format|Cells and Formats|Pattern the following modifications where achieved:

	A	B	C	D	E	F
1	Wavenumber	e1	e2		n	k
2	6000.0	-152.02	4.20		0.17	12.33
3	6088.2	-147.69	4.00		0.16	12.15
4	6176.4	-143.36	3.80		0.16	11.97
5	6264.5	-139.04	3.61		0.15	11.79
6	6352.7	-134.77	3.42		0.15	11.61
7	6440.9	-131.27	3.33		0.15	11.46
8	6529.1	-127.77	3.25		0.14	11.30
9	6617.2	-124.27	3.16		0.14	11.15
10	6705.4	-120.77	3.08		0.14	10.99
11	6793.6	-117.27	3.00		0.14	10.83
12	6881.8	-113.78	2.91		0.14	10.67
13	6969.0	-110.28	2.83		0.13	10.50

12.2 Generating graphs with the workbook

In many cases it is convenient to visualize workbook data in graphs in order to inspect relations intuitively. This section shows how you can do that with the built-in Collect object.

The Collect object is able to display spectra (or other data which are series of x-y-data pairs) in 2D and 3D graphs. In the following I refer to the individual items as 'spectrum'. The spectra to be displayed are collected in a list where parameters like pen color or line style can be set for each spectrum. Working with Collect as a stand-alone program, you can transfer spectra from SCOUT or CODE to Collect by drag&drop. Of course, you can also import spectra from data files.

Copying data from the workbook to Collect

In order to display workbook data with the Collect object they must be copied into its list of spectra. This copy process can be initiated using several workbook commands, following always the same scheme:

- Select the workbook sheet from which you want to copy
- Select the cell where the copy process should start
- If you want to generate a new graph clear the Collect spectra list with the command **Graph|Clear graphics window**. If you do not clear the data the following copy action will add the new data to the ones that are already present in Collect.
- Activate the appropriate copy command in the **Graph** submenu
- Switch to the Collect object to set the graphics parameters that control the graph

To explain the different copy commands the data created in the [parameter variation example](#) are used:

Stack 1 Layer 2 SiO2 [fused]; Layer thickness									
A1	A	B	C	D	E	F	G	H	I
1	Stack 1 Ls	0	10	20	30	40	50	60	70
2	400	-0.996312	-0.823017	-0.530538	-0.26546	-0.075129	0.0305963	0.0390829	-0.091965
3	404	-0.99683	-0.825724	-0.534151	-0.269385	-0.078019	0.0311553	0.047607	-0.065074
4	408	-0.997018	-0.826637	-0.53578	-0.271673	-0.07979	0.0321264	0.0553397	-0.041655
5	412	-0.997328	-0.828435	-0.538447	-0.274867	-0.082352	0.03226	0.0618273	-0.020842
6	416	-0.997561	-0.829893	-0.540782	-0.277816	-0.084826	0.0321774	0.0674458	-0.00252
7	420	-0.997687	-0.830769	-0.542522	-0.280302	-0.087035	0.0320307	0.0723617	0.0136071
8	424	-0.997992	-0.833059	-0.545922	-0.284226	-0.090435	0.0308151	0.0760745	0.0277945
9	428	-0.998056	-0.833634	-0.547364	-0.286488	-0.092602	0.0303172	0.0797725	0.0405902
10	432	-0.998154	-0.834514	-0.549163	-0.28906	-0.095058	0.0294535	0.0828373	0.0520162
11	436	-0.998161	-0.834767	-0.550294	-0.291086	-0.097146	0.0287043	0.0855823	0.0622192
12	440	-0.998322	-0.836259	-0.552839	-0.294317	-0.100214	0.027106	0.0874508	0.0711895
13	444	-0.998451	-0.837602	-0.555248	-0.297449	-0.103251	0.0254007	0.0889277	0.0791338
14	448	-0.998512	-0.838416	-0.557063	-0.300078	-0.105926	0.023858	0.0902483	0.0863057
15	452	-0.998722	-0.840588	-0.560423	-0.30402	-0.109665	0.0214197	0.0907153	0.0924348
16	456	-0.998719	-0.840886	-0.561647	-0.306139	-0.111991	0.0199579	0.0916432	0.0982615
17	460	-0.998693	-0.841036	-0.562735	-0.308154	-0.114265	0.018451	0.0923329	0.1034494
18	464	-0.9986	-0.840638	-0.563226	-0.309667	-0.116165	0.0171535	0.0929897	0.1081782
19	468	-0.998591	-0.840959	-0.564501	-0.311828	-0.1188586	0.0154048	0.0931861	0.1123002
20	472	-0.998726	-0.842586	-0.567272	-0.31527	-0.122045	0.0127949	0.0926034	0.1155809
21	476	-0.999105	-0.846826	-0.573074	-0.321327	-0.127619	0.0084789	0.0906034	0.1175828
22	480	-0.999129	-0.847572	-0.574876	-0.323949	-0.130466	0.0062147	0.0899849	0.1201874
23	484	-0.999276	-0.849721	-0.578285	-0.327946	-0.134424	0.0030404	0.0885751	0.1220105
24	488	-0.999332	-0.850931	-0.580642	-0.331048	-0.137685	0.0003563	0.0874117	0.1237385

Ellipsometry spectrum Delta Ellipsometry spectrum Psi

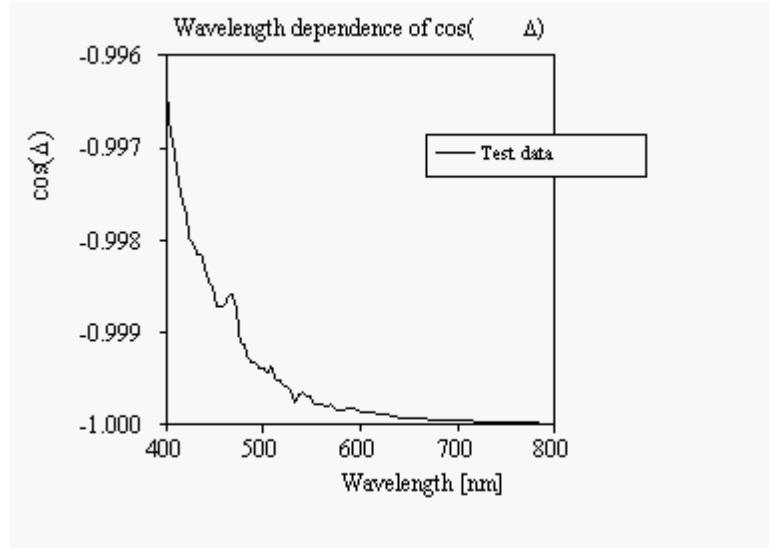
These copy commands can be used:

Graph|x,y downwards

This command is used to copy just a single spectrum starting at the selected cell. Placing the cursor in the cell A2 SCOUT will start reading the first x-value in that cell (400), then go to the cell B2 to the right of it and read the first y-value (-0.996312). Then it will inspect the cell A3. If it isn't empty the next pair of x- and y-values is taken from the cells A3 and B3. Afterwards cell A4 is tested for being empty, and so on. The reading process finishes when the first empty cell in column A occurs.

The collection of the found x-y-pairs is then treated as if it had been imported from a data file: SCOUT generates an equidistant set of x-values covering the same range of x as the input data. The corresponding set of y-values is obtained by linear interpolation from the original x-y-pairs. If, like in this example, the x-values are equidistant already, no changes to the spectrum occur. Finally the equidistant set of data is sent to the Collect spectrum list.

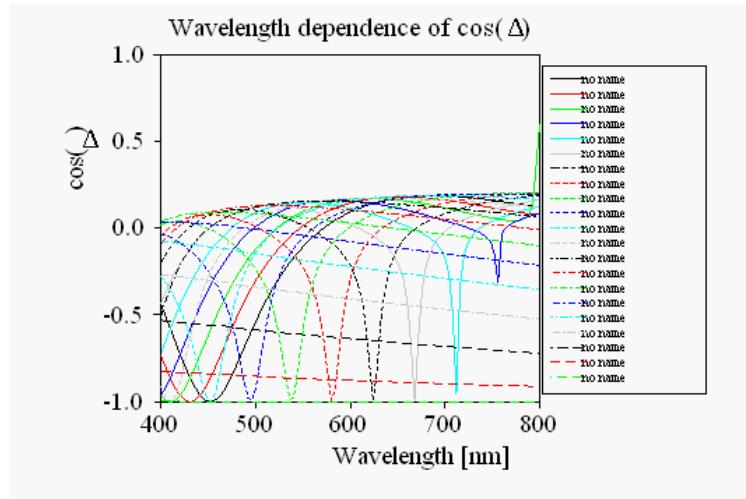
With appropriate graphics settings one would get from the sample data a graph like this:



Graph|x,y1, y2, ... downwards

After this command SCOUT starts at the cursor position and reads x-values downwards until an empty cell is found. Starting at the initial cursor position, it reads y-values moving to the right until an empty cell is found. If there are n rows of x-values and m columns of y-values the program expects the array of $n \times m$ cells to be filled with numerical data. It will pass to the Collect window m spectra being all defined on a grid of n spectral points.

Starting the copy process at cell A2 one could arrive here:

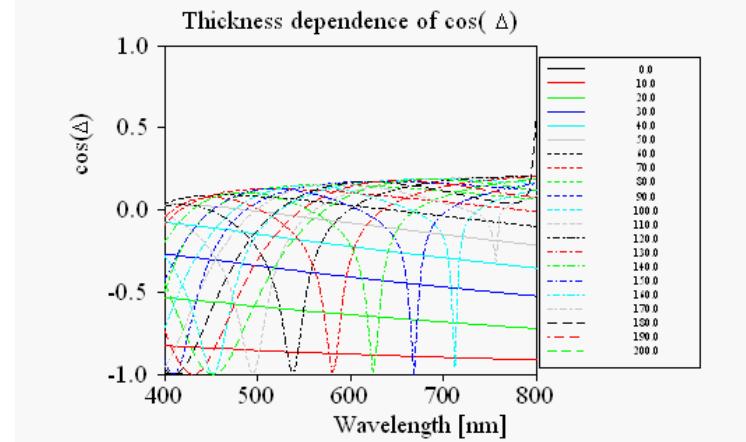


Obviously, the names of the individual spectra are not set automatically. You would have to do this manually which can be tedious if you have many spectra. If each spectrum is characterized by a number you can use the next command to achieve a reasonable automatic assignment.

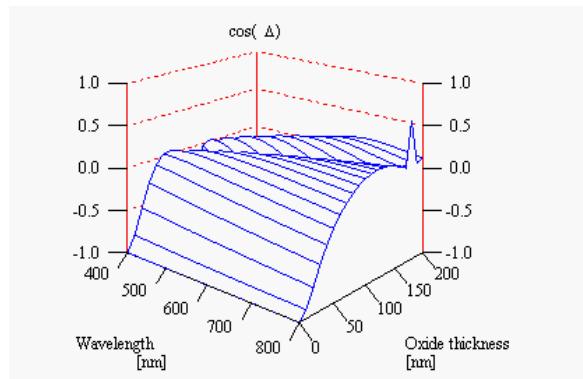
Graph|_, p1, p2, ..., x,y1, y2, ... downwards

After this command SCOUT starts at the cursor position and inspects the cells to the right until an empty cell is found. If there are m non-empty cells the entries of these cells are expected to be numerical values which are used as parameter value for the spectra. Starting in the cell directly below the selected cell SCOUT proceeds as described above, i.e. it reads n x-values downwards and m spectra to the right. The spectra and the corresponding parameter values are passed to the Collect window where the display can now be more complete since each spectrum has its own 'name' given by the parameter value.

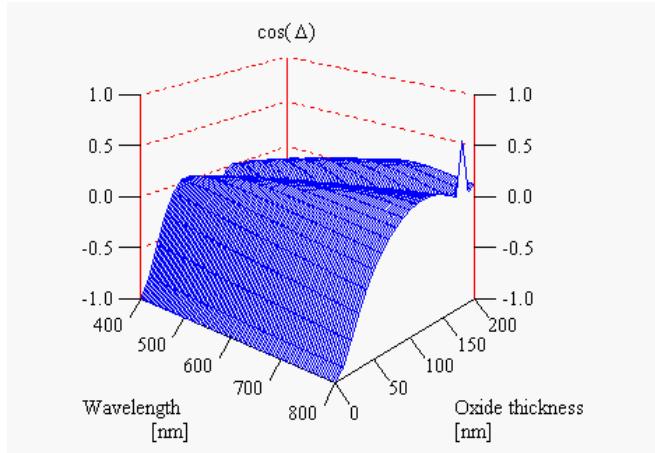
Starting at cell A1 in the example data shown above, one can achieve the following chart:



If, like in this case, each spectrum has a different parameter, and the list of spectra in the Collector window is sorted (which is done automatically for you if you apply the command **Graph|_, p1, p2, ..., x,y1, y2, ... downwards**) you can click on the buttons promising 3D plots. You get (after some plot parameter adjustments):



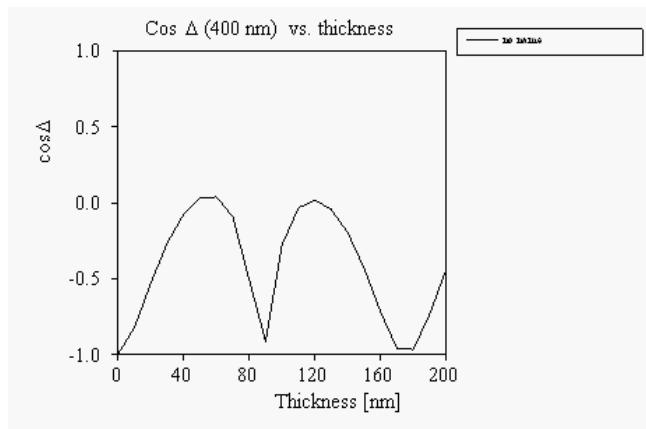
or



Graph|x,y from left to right

This command works like **Graph|x,y downwards**. The difference is that now the x-values are read from left to right, starting at the selected workbook cell. Below each x-value SCOUT expects a valid y-value.

Selecting cell B1 in the sample data, and applying this command one gets a plot of cos(delta) at 400 nm vs. SiO_x layer thickness:



Graph|x,y1, y2, ... from left to right

This command is used to read many spectra from left to right, like the 'downwards version' discussed above. Note, however, that you should avoid to import too many data sets. A number larger than 40 is not supported by SCOUT.

Graph|_, p1, p2, ..., x,y1, y2, ... from left to right

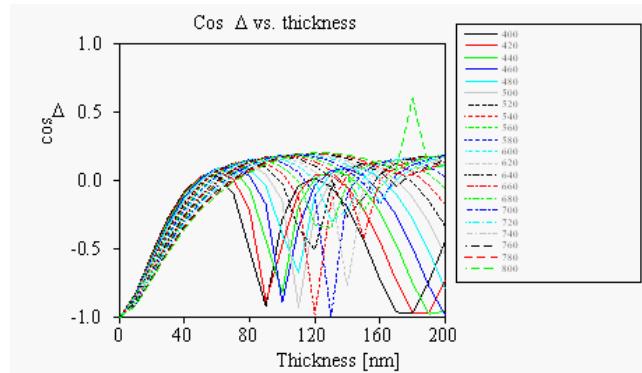
This is the 'left to right' version of the similar downwards command described above. Once more,

make sure that there are not too many rows to process. SCOUT will not work properly for more than 40 rows.

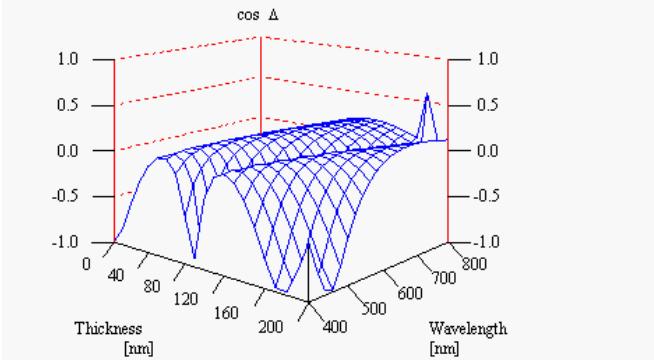
For sample data restricted to 21 spectral points in the range 400 ... 800 nm as shown here

S2 Workbook												
File Copy Paste Clear Graph												
V22		0.120543204247952										
1	Stack 1 L _a	A	B	C	D	E	F	G	H	I	J	K
2		0	10	20	30	40	50	60	70	80	90	9
3	400	-0.996312	-0.823017	-0.530538	-0.26546	-0.075129	0.0305963	0.0390829	-0.091965	-0.507433	-0.91506	
4	420	-0.997687	-0.830769	-0.542522	-0.280302	-0.087035	0.0320307	0.0723617	0.0136071	-0.213221	-0.88154	
5	440	-0.998322	-0.836259	-0.552839	-0.294317	-0.100214	0.027108	0.0874508	0.0711895	-0.054836	-0.43365	
6	460	-0.998693	-0.841036	-0.562735	-0.308154	-0.114265	0.018451	0.0923329	0.1034494	0.0349389	-0.1722	
7	480	-0.999129	-0.847572	-0.574876	-0.323949	-0.130466	0.0062147	0.0899849	0.1201874	0.0881162	-0.0334	
8	500	-0.999384	-0.853171	-0.586062	-0.338883	-0.146326	-0.007026	0.0838876	0.1277239	0.1201918	0.0473	
9	520	-0.99956	-0.858436	-0.596892	-0.353409	-0.162042	-0.020991	0.0752454	0.1292156	0.1392166	0.097673	
10	540	-0.999642	-0.862662	-0.606469	-0.366714	-0.176841	-0.034783	0.0653729	0.1269822	0.1499458	0.130121	
11	560	-0.999769	-0.868035	-0.617406	-0.3811	-0.192559	-0.049708	0.0536147	0.121314	0.1543434	0.150590	
12	580	-0.999846	-0.872913	-0.627779	-0.394907	-0.20781	-0.064533	0.0412138	0.1137065	0.1546056	0.163074	
13	600	-0.999854	-0.876362	-0.636443	-0.407123	-0.221727	-0.078458	0.0290194	0.1052222	0.1522921	0.170231	
14	620	-0.999883	-0.880333	-0.645657	-0.419708	-0.235897	-0.092707	0.0161811	0.0954494	0.1475871	0.173183	
15	640	-0.99992	-0.884603	-0.655168	-0.432452	-0.250142	-0.107095	0.0029395	0.0847568	0.1411007	0.172983	
16	660	-0.999932	-0.888139	-0.663711	-0.444236	-0.263527	-0.12082	-0.009969	0.0738844	0.1336689	0.170704	
17	680	-0.999946	-0.891706	-0.672219	-0.455894	-0.276743	-0.134428	-0.022942	0.0626058	0.1252816	0.166637	
18	700	-0.999953	-0.895226	-0.680843	-0.46773	-0.290124	-0.148221	-0.036216	0.0507734	0.1159067	0.160908	
19	720	-0.999961	-0.898617	-0.689147	-0.479149	-0.303072	-0.16164	-0.049261	0.0389243	0.1061438	0.154257	
20	740	-0.999967	-0.902043	-0.697607	-0.490762	-0.316194	-0.175228	-0.062535	0.0266925	0.0957253	0.1466526	
21	760	-0.999974	-0.905237	-0.705489	-0.50165	-0.328582	-0.188154	-0.075281	0.014786	0.0853527	0.138464	
22	780	-0.99998	-0.908442	-0.713466	-0.51267	-0.341089	-0.20119	-0.088173	0.0026337	0.0745497	0.129685	
23	800	-0.999984	-0.911296	-0.720641	-0.522706	-0.352607	-0.213321	-0.100295	-0.008929	0.0641054	0.120972	

one gets the following picture (starting with cell A1 selected initially):

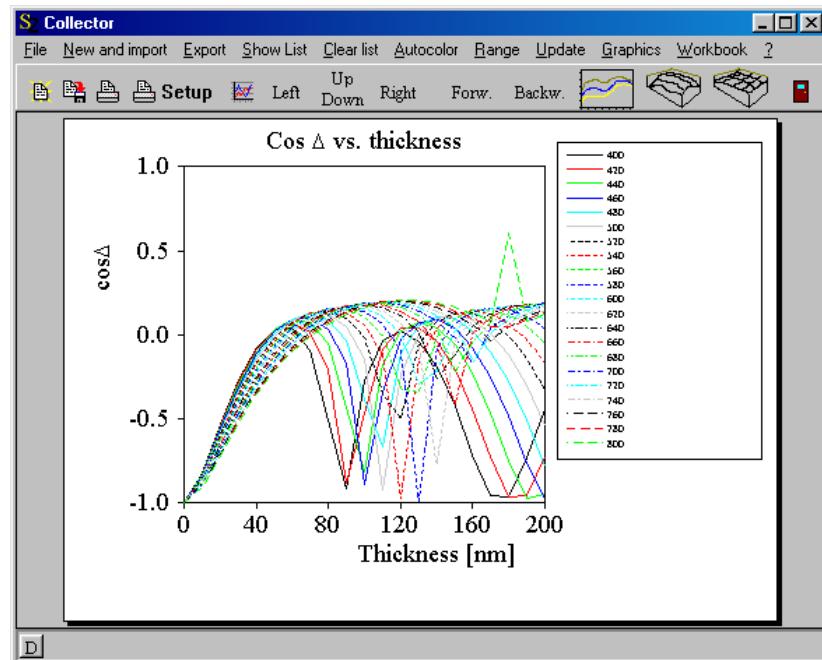


Plots with well-defined and different parameters for each data set can also be done in 3D:



Displaying the data in the Collect window

Once the data are copied by one of the commands above the Collect window is opened:



The individual spectra are collected in a list which is opened by the **Show List** command:

The figure shows the SCOUT List of spectra window. It displays a table of collected spectra, with each row representing a spectrum. The columns are labeled: Parameter, Color, Line Type, Active, Name, Filename, and Spectral Unit. The table lists 18 rows, each corresponding to a different wavelength or parameter value. The 'Active' column indicates which spectra are currently active. The 'Name' column lists the names assigned to each spectrum, such as 400, 420, 440, etc. The 'Filename' column lists the filenames for each spectrum, such as Row 1, Row 2, etc. The 'Spectral Unit' column lists the spectral units for each spectrum.

	Parameter	Color	Line Type	Active	Name	Filename	Spectral Unit
1	400	1	1	active	400	Row 1	
2	420	2	1	active	420	Row 2	
3	440	3	1	active	440	Row 3	
4	460	4	1	active	460	Row 4	
5	480	6	1	active	480	Row 5	
6	500	7	1	active	500	Row 6	
7	520	1	3	active	520	Row 7	
8	540	2	3	active	540	Row 8	
9	560	3	3	active	560	Row 9	
10	580	4	3	active	580	Row 10	
11	600	6	3	active	600	Row 11	
12	620	7	3	active	620	Row 12	
13	640	1	5	active	640	Row 13	
14	660	2	5	active	660	Row 14	
15	680	3	5	active	680	Row 15	
16	700	4	5	active	700	Row 16	
17	720	6	5	active	720	Row 17	
18	740	7	5	active	740	Row 18	

Each spectrum has a parameter, a color, a line type, a name and a spectral unit.

12.3 Display graphs of workbook data in a view

Workbook data can be displayed in the main window of SCOUT using objects of type [Workbook display \(columns\)](#) or [Workbook display \(rows\)](#) in a view. The description of these objects contain detailed examples demonstrating the required steps.

12.4 Workbook file formats

You can save the workbook to a file using the File|SaveAs command. You can chose between different file formats:

Formula One (*.vts)

This is a binary format which should be used if the saved workbook content will only be read by SCOUT workbooks again. Hence this format is not very useful if you want to exchange data with other programs by file transfer. However, since this is the native format of the workbook component, it is the most complete storage method with no loss of information.

Excel 5.0 format (*.xls)

This file type can be used to store the workbook in Excel binary format. Obviously, this can be used to transfer data to Excel or other programs that can import Excel data.

Text file

Alternatively, you can save the workbook content using this text file format which can also be an attractive way to make data accessible to other programs.

12.5 Actions involving the workbook

12.5.1 Parameter variation

This very powerful action shows the influence of one of the model parameters on optical spectra. It works like this: In the main window of SCOUT, the command **Actions|Parameter variation** is executed. In a dialog you can pick one of the model parameters and set the range of the variation by specifying the minimum and the maximum value and the number of parameter values that you want to consider. For each particular value of the parameter range, SCOUT computes all spectra and stores them appropriately in worksheets in the workbook.

In most cases you want to inspect the results graphically. This can be done by copying the workbook data to the clipboard and from there to Excel or any other software of your choice, or you can do the visualization using the Collect object that is built in SCOUT.

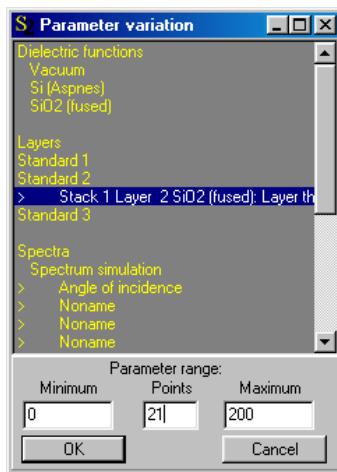
Here is a step by step example. The goal is to investigate the dependence of a reflectance and an ellipsometry spectrum on a layer thickness. In order to concentrate on the demonstration of the parameter variation technique, a very simple example of a silicon oxide layer (layer type: thin film. This type of layer is automatically treated with coherent superposition of partial waves and the unit for the thickness is nm) on a silicon substrate (halfspace) is used. A reflectance spectrum (normal incidence of light) and ellipsometry data (70 degrees angle of incidence) are computed in the spectral range 400 ... 800 nm (100 spectral data points).

The individual steps are these:

Activate - in the main window of SCOUT - the command **Actions|Parameter variation** and observe that you enter this dialog:



Select the layer thickness and a parameter range from 0 to 200 nm using 21 points (this means you will have 10 nm steps in the variation) according to the following settings:



If a parameter has a unit (like thicknesses) you have to specify the parameter range using exactly the same units. In the present example the unit of the layer thickness is nm.

After pressing OK SCOUT computes all the required spectra and automatically opens the workbook:

Stack 1 Layer 2 SiO ₂ (fused); Layer thickness										
	A	B	C	D	E	F	G	H	I	
1	Stack 1	λ	0	10	20	30	40	50	60	70
2	400	-0.996312	-0.823017	-0.530538	-0.26546	-0.075129	0.0305963	0.0390829	-0.091965	-0.1
3	404	-0.996683	-0.825724	-0.534151	-0.269385	-0.078019	0.0311553	0.047607	-0.0865074	-0.
4	408	-0.997018	-0.826637	-0.53578	-0.271673	-0.07979	0.0321264	0.0553397	-0.041655	-0.
5	412	-0.997328	-0.828435	-0.538447	-0.274867	-0.082352	0.03226	0.0618273	-0.020842	-0.
6	416	-0.997561	-0.829893	-0.540782	-0.277816	-0.084826	0.0321774	0.0674458	-0.00252	-0.
7	420	-0.997687	-0.830769	-0.542522	-0.280302	-0.087035	0.0320307	0.0723617	0.0136071	-0.
8	424	-0.997792	-0.833059	-0.545922	-0.284226	-0.090435	0.0308151	0.0760745	0.0277945	-0.
9	428	-0.998056	-0.833634	-0.547364	-0.286488	-0.092602	0.0303172	0.0797725	0.0405902	-0.
10	432	-0.998154	-0.834514	-0.549163	-0.28906	-0.095058	0.0294535	0.0828373	0.0520162	-0.
11	436	-0.998161	-0.834767	-0.550294	-0.291086	-0.097146	0.0287043	0.0855823	0.0622192	-0.
12	440	-0.998322	-0.836259	-0.552839	-0.294317	-0.100214	0.027106	0.0874508	0.0711895	-0.
13	444	-0.998451	-0.837602	-0.555248	-0.297449	-0.103251	0.0254007	0.0889277	0.0791338	-0.
14	448	-0.998612	-0.838416	-0.557063	-0.300078	-0.105926	0.023868	0.0902483	0.0863057	-0.
15	452	-0.998722	-0.840588	-0.560423	-0.30402	-0.109666	0.0214197	0.0907153	0.0924348	0.01
16	456	-0.998719	-0.840886	-0.561647	-0.306139	-0.111991	0.0199579	0.0916432	0.0982615	0.0
17	460	-0.998693	-0.841036	-0.562735	-0.308154	-0.114265	0.018451	0.0923329	0.1034494	0.0
18	464	-0.9986	-0.840638	-0.563226	-0.309667	-0.116165	0.0171535	0.0929897	0.1081782	0.0
19	468	-0.998591	-0.840959	-0.564501	-0.311828	-0.118586	0.0154048	0.0931861	0.1123002	0.0
20	472	-0.998726	-0.842586	-0.567272	-0.31527	-0.122045	0.0127949	0.0926034	0.1155809	0.0
21	476	-0.999105	-0.846826	-0.573074	-0.321327	-0.127619	0.0084789	0.0906034	0.1175828	0.0
22	480	-0.999129	-0.847572	-0.574876	-0.323949	-0.130466	0.0062147	0.0899849	0.1201874	0.0
23	484	-0.999276	-0.849721	-0.578285	-0.327946	-0.134424	0.0030404	0.0885751	0.1220105	0.0
24	488	-0.999332	-0.850931	-0.580642	-0.331048	-0.137685	0.0003563	0.0874117	0.1237385	0.1

SCOUT has created a new worksheet for the reflectance data, and two new worksheets for the cos(delta) and tan(psi) values. Each worksheet has in its first row the name of the varied parameter and the parameter values. The first column downwards contains the spectral position (here the wavelength in nm). Below each parameter value there is a column with the corresponding spectrum. Clicking in the gray field to the left of 'A' and above '1' selects the whole worksheet. The **Copy** command transfers the selected cells into the clipboard. From the clipboard you can get the data with **Paste** in Excel (or what ever program you prefer for data juggling).

If you like you can also generate 2D or 3D graphs with SCOUT which provides a built-in Collect object for doing this. The operation of this part of SCOUT is described in a [separate section](#).

After you have done a parameter variation, you can apply the SCOUT command **Actions|Repeat parameter variation** to do it once more. SCOUT will empty all relevant worksheets and fill them with new data. You do not have to select the variation parameter and the range any more. This way to work is very useful in optimizing a model where you manually make frequent changes and you have to inspect the parameter variation after each modification of the model.

12.5.2 Parameter fluctuation

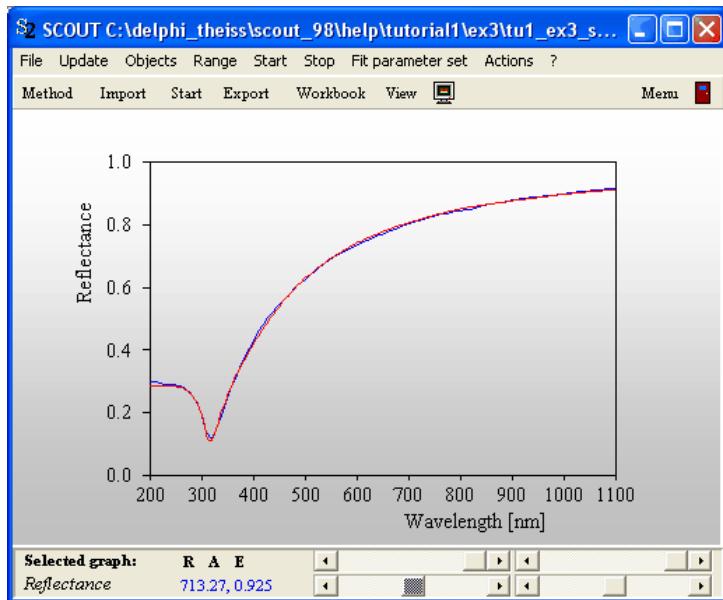
Similar to [Parameter variations](#) this action visualizes the consequence of fluctuations of model parameters to optical spectra. It is activated by the main window command **Actions|Parameter fluctuation**.

Preparation

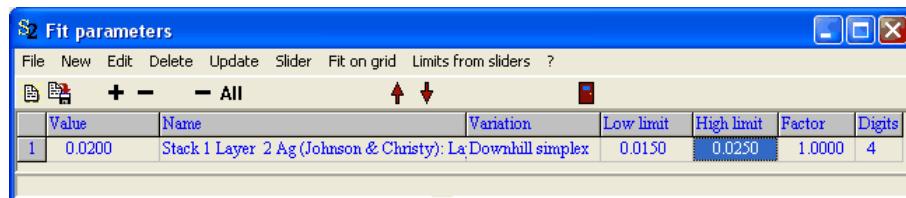
Before you use this SCOUT action you should have a working model that computes optical spectra. The list of fit parameters should contain only those parameters for which you want to study the influence on the spectra. For each parameter you have specified the range of fluctuation by setting appropriate values for its minimum and maximum.

An example could be the following: For a layer stack you have optimized the film thicknesses. Now you want to know what spectral variations you can expect when the thicknesses are realized in a production with some tolerances, i.e. random fluctuations around their design values. In that case, all relevant thicknesses should be selected as fit parameters, and the range of fluctuation must be entered in the list of fit parameters.

In order to demonstrate the application of the 'Parameter fluctuation' action, start SCOUT and load the final configuration of SCOUT tutorial 1, example 3. This has been saved in the file tu1_ex3_step3.s98 (you need to select the SCOUT 98 file filter in the file dialog). After loading it with SCOUT 2 you should see the following main window:

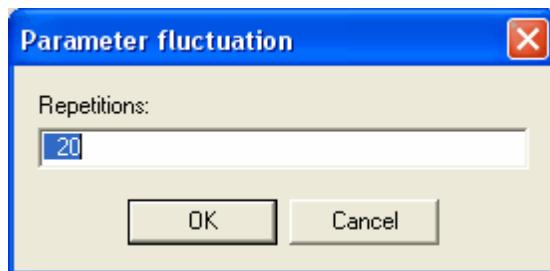


The configuration computes a reflectance spectrum for a single Ag layer on glass. The Ag layer thickness is the only fit parameter in the list of fit parameters. Set its range to 0.015 ... 0.025 (thickness in microns):



Activation

Execute the command **Actions|Parameter fluctuation** in the main window of SCOUT. You are asked for the number of repetitions, i.e. the number of realizations of a random parameter value and subsequent spectrum computations:



At present, values between 1 and 250 are allowed here. Leave the suggested value of 20 and press OK. That's all you have to do!

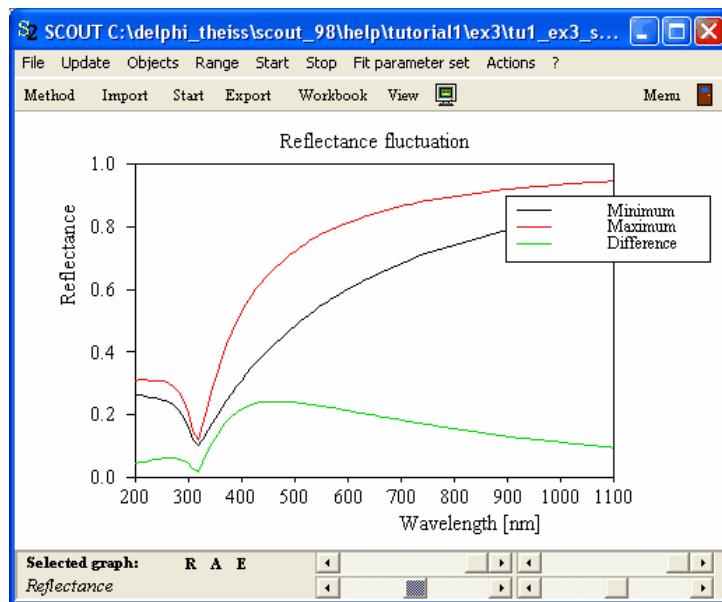
What happens

For each spectrum in the list of spectra (in the current example there is only one, the reflectance spectrum) a new worksheet in the workbook is created. The worksheet gets the name of the spectrum with the extension 'fluctuation', in the example the name is 'Reflectance fluctuation'. SCOUT will then draw 20 times a random number between 0.015 and 0.025, assign the value to the Ag layer thickness and re-compute the reflectance spectrum. The reflectance data are stored in workbook columns, and SCOUT will compute the minimum and maximum (and their difference) of the reflectance for each spectral point. The worksheet looks like this:

S2 Workbook								
File Copy Paste Clear Graph Autosize cells								
D101	0.0948584675788879							
A	B	C	D	E	F	G	H	I
1	Wavelength	Minimum	Maximum	Difference		1	2	3
2	200	0.2653151	0.3086021	0.043287	0.2653151	0.2680086	0.307772	0.281058
3	209.09091	0.2635426	0.3107365	0.0471939	0.2635426	0.2663463	0.309662	0.2800474
4	218.18182	0.2600889	0.3101568	0.050068	0.2600889	0.2629548	0.3088806	0.2770614
5	227.27273	0.2550761	0.3076255	0.0525493	0.2550761	0.2579767	0.3061494	0.2723575
6	236.36364	0.2522813	0.3083047	0.0561284	0.2522813	0.2552675	0.3065789	0.2701241

In addition, SCOUT creates a new [view](#) called 'Reflectance fluctuation' and displays the minimum, maximum and difference of the fluctuating reflectance data. In the case of several spectra in the list of spectra, SCOUT will create several views with corresponding names. Click the **View** button in the main window's speedbar to cycle through all views.

The data display in the view is done by an object of type '[Workbook display \(columns\)](#)'. In the example, the view is the following:



After the creation of the view object, you can modify (**Click** on the object while the **Ctrl-key** is pressed) its appearance (change the graphics parameters, line types, colors and names of the individual curves, legend settings) as described [above](#). However, if you do a parameter fluctuation computation again the graphics parameters are copied from the original spectrum object and some of your changes may be lost.

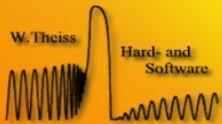
Doing it again

If you have changed the model (e.g. the range of fluctuations of some of the parameters) you can redo the fluctuation computation by **Actions|Repeat parameter fluctuation**. After this command, you are not asked for the number of repetitions but the value set previously is used again.

Exporting the fluctuation data

Since the spectral data of the computation are stored in the workbook you can use the [workbook mechanisms for data export](#) to save the data to files or copy&paste to other programs like Excel.

SCOUT



Part

XIII

13 The database

13.1 Introduction

With SCOUT 2.1 a new structure of the database has been introduced. SCOUT does not require the Borland Database Engine any more. Instead, all database records are realized as files in a well-defined folder structure. Initially, we have made these changes in order to prepare SCOUT for the Linux operating system (for which the Borland Database Engine is not available). While the step to Linux is still far away, the new database concept turned out to be quite useful already.

The structure of the database is this: The main database folder (default: 'Database' in the SCOUT program directory) contains several subfolders. The subfolders contain files with the data of the various database objects. At present the following subfolders are implemented:

<u>Materials</u>	This folder contains material definitions
<u>Coatings</u>	Here you find layer stack definitions
<u>Substrates</u>	Layer stack definitions of typical substrates

Creating a new database

You can create a new database anywhere in your file system. The full setup of SCOUT 2.1 will create a small database named 'My_database' in the SCOUT program directory. Just copy this folder (including its subfolders) to the desired new location and rename it appropriately.

Connecting SCOUT to a new database

In order to connect to a database different from the default database you have to use the **File|Options|Database** command in the main window. Specify the new database folder in the subsequent dialog and you are connected to the new database.

The database path is stored in a SCOUT configuration if you use the version 2.1 file format.

Handling database objects

SCOUT itself does not provide mechanisms to delete or move database entries. Since all database objects are simple files you can delete and move them with Windows Explorer. Note, however, that SCOUT reads the content of the database at program start or during re-connection only. Hence after a database modification you have to re-connect SCOUT to the database in order to update the lists. If you want to delete, copy or move materials you have to keep in mind that the properties of each material are stored in two files. These files have the same filename, but different extensions. In the case of copying stored effective dielectric functions, you as the database manager are responsible for copying the relevant host and particle materials as well to the new location.

Converting your old optical constant database to the new format

You can convert your old optical constant database to the new format using the DF_DB tool which is delivered with the SCOUT 2.1 setup files. This program is described in the section about tools [below](#).

13.2 Materials

Materials are stored in the subfolder 'Materials' (no surprise!). The properties of each material are stored in two files. The name of the material is used as filename. The two files have the extensions *.oc and *.b . Since a material's name is used to create filenames you cannot use symbols like '\', '/', '<', '>' or '%' in the name.

If a material is a composite (described as effective medium) the database contains links to the related host and particle materials which must be in the database as well. If you drag such an item from the database to the material list in SCOUT the entries for the host and the particle material are automatically copied as well.

The materials available in the database are accessible in the [material list](#).

13.3 Layer stacks

Layer stacks are stored in the 'Coatings' and 'Substrate' subfolders. Use the save mechanism of a [layer stack](#) to create new entries in these database folders. You must use the recommended extension *.ls for a valid database object. At program start (or after a database connection) SCOUT will read all entries with the *.ls extension from the database folders. The found names are used to to fill the dropdown boxes for coatings and substrates in layer stack definition objects.

In order to illustrate layer stacks graphically you can generate bitmap graphs and store them in the database folders as well. For each entry with *.ls extension SCOUT will look for a bitmap file with the same name and the extension *.bmp. If such a file is found it will be displayed for the selected coating or substrate in layer stack definition objects.

SCOUT



Part

XIV

14 Technical notes

14.1 General features of SCOUT objects

SCOUT objects have some general features which are documented in this section.

There are some common menu commands and also some symbols on the 'speed bar' (called speed buttons) below the menu which are shared by many objects. Most of the speed buttons cause the same action than menu commands which is indicated in the following.

File

The file menu with **New**, **Open** (), **Save** and **Save As** () commands for loading and storing the object to files (binary). Also a **print** and **printer setup** command as well as an **option** item is provided generally. The **report** command writes the object report to a small text editor from where it can be saved to a text file or copied as text into the clipboard. Usually an **Exit** () action is also provided.

Update

This command is used to update the object. Most objects will respond by re-drawing the object's graphics, some will also do some re-computation if necessary.

Help

Show this help.

14.2 List properties

Many SCOUT objects use lists to manage a collection of objects. In some cases the objects are all of the same type, in others one list manages various types of objects. A typical list window looks like this:



On top are the menu items of the list. Besides the File submenu which is a feature of almost all [SCOUT objects](#) the above example shows the items that are common to most list objects:

New ()

This command creates a new list object. If the list supports only one type of objects a new one is created and appears in the list. Usually you have to choose one of several possible object types. This is done by a selection box like the following



which shows the object types that may be created in this particular list. An object of the selected type is created when the **New** command is activated the next time.

If there are already objects in the list and one is selected (simply place the cursor on the object's line to select it) most lists insert the new object before (i.e. above) the selected one. Some lists simply append new objects at the end of the list.

Edit ()

This activates some mechanism to edit the currently selected object. For objects which are represented by a window this window will be opened giving access to all the objects settings. Objects which are not represented by a window may show a dialog to edit their properties. Some objects may not respond at all to the edit command if their properties can be set directly in the list (see below).

An alternative way to activate the 'edit action' is to select an object's line in the list and click on it with the right mouse button.

Delete ()

This deletes the currently selected item, usually without any warning.

Delete all

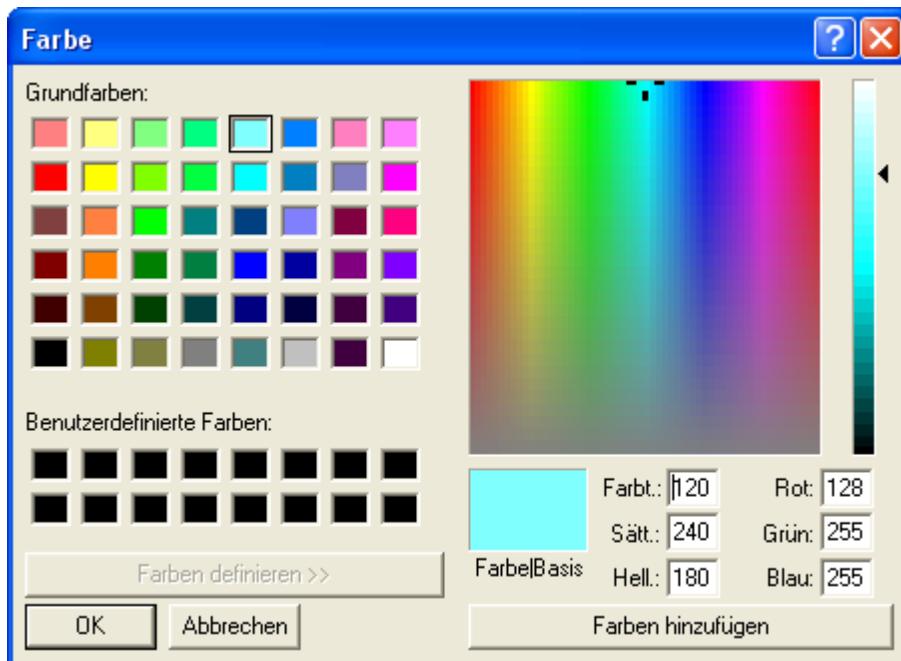
Use this command to delete all objects in the list.

Move up () and move down ()

For cases where the order of the list entries is important you can move up and down the selected object using the shown speed buttons. These actions cannot be activated by menu items.

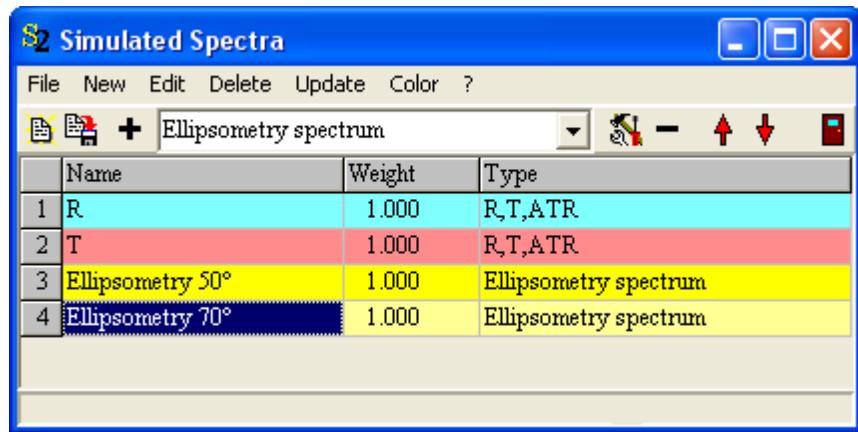
Color

This menu command is available in almost all lists. It opens the standard windows dialog to select a color (depending on your windows version you may get this dialog in your language, of course) for the selected object in the list:



The selected color is used to draw the background of the object in the list. Since the text is

displayed in black you should avoid dark colors in order to maintain readability. Here is an example of a list with colored objects:



A double click on an object in a list activates the color dialog as well.

Direct editing

Most text entries in a list can be edited directly by simply placing the cursor on the text and typing in something new. In this case the selected 'text cell' is turned into edit mode which can be seen by the white background color of the cell. Instead of simply starting to type you can switch to edit mode by hitting the return key on your keyboard. A second return key hit shuts off edit mode again.

Cell selected, no editing:

name	we
Spectrum simulation	1

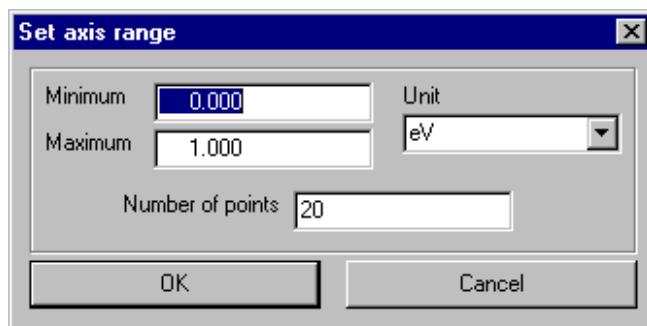
Cell selected and in edit mode:

name	we
Spectrum simulation	1.00

Most objects offer some of their basic parameters (like their name, e.g.) for direct editing which is very convenient if you have to modify many objects in a list. When you finish editing by moving to another cell or by hitting the return key the object decides if it accepts the text entered by you. If not the cell content is restored that was there before you started editing.

14.3 Spectral range selection

Most objects where a spectral range must be defined use the following dialog to set the required values specifying the range:



The example defines 20 points (with equal spacing in between the points) covering the range from 0 to 1 eV. The spectral units supported by SCOUT are

Quantity	Unit
Wavenumber	1/cm
Energy	eV
Frequency	THz
Wavelength	nm
Wavelength	micron

14.4 User defined functions

Some SCOUT objects use a formula interpreter to create data. To do this the user usually specifies in a dialog the formula by editing a string. Also the range over which the data will be calculated can be set. When you leave the dialog a formula interpreter checks the formula string and fills the data of the object with new values calculated according to the formula.

In a formula string you can use the following functions:

Function	Explanation
abs(...)	the absolute value of the argument
sin(...)	
cos(...)	
tan(...)	trigonometric functions (argument in rad)
cot(...)	
asin(...)	
acos(...)	inverse trigonometric functions
atan(...)	
sinh(...)	
cosh(...)	hyperbolic functions
tanh(...)	
exp(...)	the exponential
ln(...)	the natural logarithm
log(...)	the logarithm with base 10
sqr(...)	the square of the argument
sqrt(...)	the square root of the argument
frac(...)	returns the fractional part of the argument
int(...)	returns the integer part of the argument
step(...)	the step or 'heavyside' function
rect(...)	the rect function
rnd(...)	returns random numbers centered around 0 with amplitude given by the argument
si(...)	sin(...)/(...)
pi	3.14159

Expressions of the form $\text{number1}^{\text{number2}}$ are allowed and return the number²'th power of number1. Of course the usual operators +,-,*,/ and () can be used to combine expressions.

The variables that can be used in a formula string vary from object to object and are given in the individual formula dialogs.

14.5 Import formats

14.5.1 Overview

The formats for data import supported at present are the following:

- [SCOUT standard format](#)
- [xy-format](#)
- [JCAMP format](#)
- [SpectraCalc format](#)
- [Perkin-Elmer format](#)
- [Perkin-Elmer ASCII format](#)
- [Zeiss format](#)
- [Opus format](#)
- [HunterLab UltraScan](#)
- [Hitachi ASCII format](#)
- [Woollam WVASE32 ASCII format](#)

If your favourite format is not supported please ask us to implement it.

Data from Excel tables can be imported using the SCOUT workbook. A description of this procedure is given [below](#).

14.5.2 SCOUT standard format

For easy data exchange between SCOUT objects and other programs SCOUT objects can read and write data in ASCII files using a simple, column based data format. This is explained at best in an example. A file containing data columns (y-values) defined on **equidistant x-values** would look like this:

Standard data file	Comment
0	First row: minimum of the x-range
100	Second row: maximum of the x-range
7 3	Third row: Number of rows of data points and separated with a blank the number of data columns. Here: 7 rows in 3 columns.
Data created by M.D. 1.00 232.4 4343.97 1.30 242.3 4545.23 0.34 234.4 4546.45 4.56 232.6 4543.40 1.45 232.4 4545.56 1.43 232.8 4545.63 1.40 232.9 4545.99 Hello user! In this section anything can be added but it is useless since it	Fourth row: one line to comment the data Data rows and columns Anything else in the file is not read usually.

is not read by SCOUT objects.

The format for non-equidistant x-values (i.e. for each y-value there is an x-value) is the following:

Standard data file	Comment
0	First row: 0
0	Second row: 0 To indicate the format for nonequidistant data the first and second row must contain the same number, usually a 0
7 3	Third row: Number of rows of data points and separated with a blank the number of columns of y-data. Here: 7 rows in 3 columns.
Data created by M.D. 0.01 1.00 232.4 4343.97 0.02 1.30 242.3 4545.23 0.05 0.34 234.4 4546.45 0.04 4.56 232.6 4543.40 0.03 1.45 232.4 4545.56 0.09 1.43 232.8 4545.63 0.12 1.40 232.9 4545.99	Fourth row: one line to comment the data Data rows and columns one row of x-data three rows of y-data
Hello user! In this section anything can be added but it is useless since it is not read by SCOUT objects.	Anything else in the file is not read usually.

14.5.3 xy-format

Some SCOUT objects can read or write data in a very simple format without any header, namely by just using pairs of x- and y-values stored in ASCII files.

An example is the following:

1.2	0.234
1.5	0.302
1.9	0.452
0.7	0.332

Since the x-values must not necessarily have equal spacings (as in the example above) a problem arises when an object tries to import such data which requires equidistant x-values (such as object of type 'single data column field'). In this case the following is done:

The pairs of x-y-data are read from the file, then they are sorted by the x-values (starting with the smallest). A 'grid' of equally spaced x-values is created covering the same x-range as the original data and using the same number of data points. Finally the corresponding y-values are calculated by

interpolating between the original data.

14.5.4 JCAMP format

The JCAMP-format is a text file (ASCII files) format which has been established as a standard for data transfer between spectroscopic software packages. Unfortunately there exist by now many different JCAMP-formats with different data compression techniques (there are even binary JCAMP-formats). SCOUT supports only a very few of the simple ASCII-versions.

14.5.5 SpectraCalc format (GRAMS)

Many optical spectra are stored using the very compact binary SpectraCalc-format (usually identified by the extension *.SPC). SCOUT can import such files as long they contain just one spectrum - multiple spectra files are not supported up to now.

14.5.6 Perkin-Elmer format

Binary format used by Perkin-Elmer spectrometers (usual file extension: *.sp).

If the import using this routine does not work, please try the import routine for [Perkin-Elmer ASCII format](#).

14.5.7 Perkin-Elmer ASCII format

ASCII format created by Perkin-Elmer spectrometers. Unfortunately the default file extension is the same (*.sp) as for the binary format [above](#). So be sure which format you have before you try to import data using this import routine.

14.5.8 Zeiss format

Format used by Zeiss spectrometers for spectrum export. Typical file extensions: *.000, *.001, ...

14.5.9 Opus format

Used by the Bruker Opus software to export data in binary files. Typical file extension: *.0, *.1, ...

14.5.10 HunterLab UltraScan

This format is text file based (default extension: *.CSV). A file may contain several spectra, each one in a separate row with its spectral values separated by commas. A header line contains the description of the items in each row.

SCOUT will display the first four values found for the individual spectra in a dropdown list and give the user the opportunity to select a spectrum from the list. It is assumed that the first four items contain a description of the spectrum, giving enough information for identification.

14.5.11 Hitachi ASCII format

This is a simple ASCII format created by Hitachi spectrometers. SCOUT will only import data from the file if line 4 in the file contains a '3' - this indicates a spectral scan in nanometers.

SCOUT will not check if the spectral type denoted in line 5 (Reflectance, Transmittance, ...) agrees with the settings in the importing object. This is your responsibility.

14.5.12 WVASE32 ASCII format

Output format of the Woollam WAVSE32 software.

This format is for multiple files. SCOUT will handle the data according to the following rules:

Ellipsometry:

- If the importing object is an ellipsometry spectrum it will accept data only from lines starting with a number in the first column (This indicates ellipsometry data). Secondly, the ellipsometry object will check if the angle of incidence given in the second file column agrees with its own angle. Only in the case of positive identification of spectrum type (ellipsometry) and angle of incidence it will read psi and delta.
- Ellipsometry objects will not do any automatic correction of the type of computed quantities - please check yourself if the model computes Psi and Delta when you import Psi and Delta values. If the imported data are trigonometric functions of Psi and Delta you have to do appropriate settings in the ellipsometry object to compute the same quantities in the model.

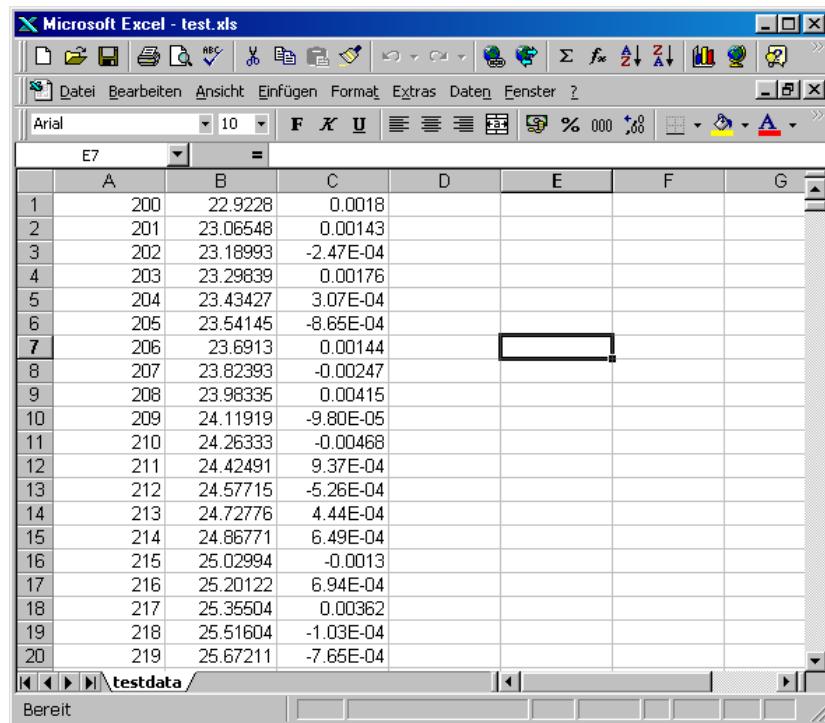
Reflectance and transmittance:

- In the case of reflectance or transmittance spectra, SCOUT will scan the WVASE32 file for lines starting with identifiers like 'uT' or 'sRr'.
- In the case of non-normal incidence of light, SCOUT will only accept data lines with a polarization state matching the polarization settings of the model: The first letter of the data line indicates the polarization: 's' denotes s-polarization, 'p' stands for p-polarization and 'u' is for unpolarized radiation.
- In the case of normal incidence (exactly 0 degree or 180 degree), SCOUT will scan for data lines starting with 'u' independent of the polarization settings of the model (If you would set the polarization in the model to 'mixed' SCOUT would compute both s- and p-polarized spectra and do the averaging. This would take twice as long as the computation of a single spectrum).
- In the case of a computed reflectance spectrum, SCOUT will only accept lines for which the 2nd letter is 'R'.
- In the case of a computed transmittance spectrum, SCOUT will only accept lines for which the 2nd letter is 'T'.
- If the angle of incidence in the model is larger than 90 degree, only data lines with the 3rd letter equal to 'r' are considered.
- If the angle of incidence is below 90 degrees, the third letter must be 'b' for reflectance (i.e. including backside reflection).
- In the case of transmittance and angles below 90 degrees, the 3rd letter must be blank. The Woollam software does not create different notations for transmittance spectra with or without multiple backside reflections.

If your WVASE32 files contain all the spectra you need for your SCOUT configuration, you can use the following shortcut for data import: Use the command **Actions|Import all spectra from a single WVASE32 ASCII file**, select the file and SCOUT will try to import measured spectra for all objects in your list of spectra that import measured spectra.

14.5.13 Import from Excel

Spectra from Excel can be imported if they are available in the following structure: There must be a column with the spectral positions, followed by a column with the reflectance or transmittance data. An example is shown below:



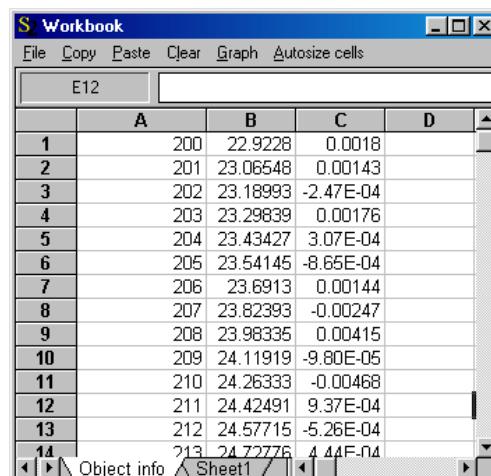
The screenshot shows a Microsoft Excel window titled "Microsoft Excel - test.xls". The spreadsheet contains a single sheet named "testdata". Column A (wavelength) ranges from 200 to 219 nm. Column B (transmittance) ranges from 22.9228% to 25.67211%. Row 7 is selected, indicated by a black border around the cells in columns A through F.

	A	B	C	D	E	F	G
1	200	22.9228	0.0018				
2	201	23.06548	0.00143				
3	202	23.18993	-2.47E-04				
4	203	23.29839	0.00176				
5	204	23.43427	3.07E-04				
6	205	23.54145	-8.65E-04				
7	206	23.6913	0.00144				
8	207	23.82393	-0.00247				
9	208	23.98335	0.00415				
10	209	24.11919	-9.80E-05				
11	210	24.26333	-0.00468				
12	211	24.42491	9.37E-04				
13	212	24.57715	-5.26E-04				
14	213	24.72776	4.44E-04				
15	214	24.86771	6.49E-04				
16	215	25.02994	-0.0013				
17	216	25.20122	6.94E-04				
18	217	25.35504	0.00362				
19	218	25.51604	-1.03E-04				
20	219	25.67211	-7.65E-04				

Column A contains the wavelength in nm, column B the transmittance data in %.

To import this spectrum to SCOUT the following steps are to be taken:

- Select column A and B or the whole worksheet and copy the selection to the clipboard.
- Open the SCOUT spectrum window where you want to import the spectrum.
- Use the **Workbook|Open workbook** command to open the workbook.
- Select cell A1 and activate the Paste command to paste the clipboard data to the SCOUT workbook. Use the **Autosize cells** command to resize the cell widths if necessary.



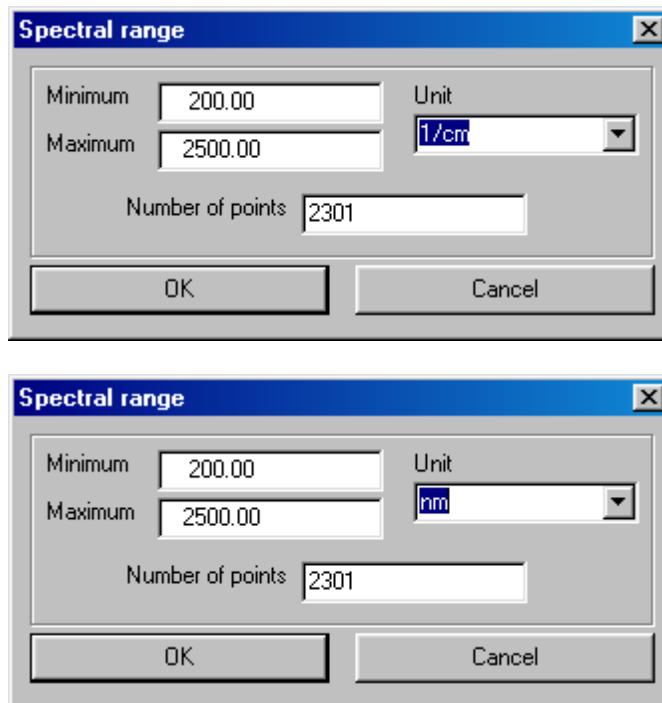
The screenshot shows the SCOUT "Workbook" window. The "testdata" sheet is active, displaying the same spectrum data as the Excel spreadsheet. The data is pasted into cells A1 to D14. The "File" menu is visible at the top.

	A	B	C	D
1	200	22.9228	0.0018	
2	201	23.06548	0.00143	
3	202	23.18993	-2.47E-04	
4	203	23.29839	0.00176	
5	204	23.43427	3.07E-04	
6	205	23.54145	-8.65E-04	
7	206	23.6913	0.00144	
8	207	23.82393	-0.00247	
9	208	23.98335	0.00415	
10	209	24.11919	-9.80E-05	
11	210	24.26333	-0.00468	
12	211	24.42491	9.37E-04	
13	212	24.57715	-5.26E-04	
14	213	24.72776	4.44E-04	

- Now place the cursor in the workbook cell A1 (which contains the number 200)
- Go back to the spectrum window (you can close the workbook if you like) and import the data with

the command **Workbook|Import xy**.

- In the following range dialog, check the unit and change it if it's wrong.



- That's it. The data are imported.

14.6 Export formats

14.6.1 Overview

Some SCOUT objects can export computed numbers to data files. Depending on the type of format the user has to specify some parameters such as the size of a data column or the number of decimals used in text file export.

The following formats are used:

[SCOUT standard format](#)
[SpectraCalc format](#)
[Perkin-Elmer format](#)

14.7 Options

You can set several program options to influence the behaviour of SCOUT. These are accessible using the **File|Options** submenu.

Here is a list of implemented options:

Password

You can protect a SCOUT configuration by setting a password. However, this protection only works with the main menu switched off. To turn the main menu on or to modify view elements, the user needs to know the password.

If the password is empty (this is the default setting at program start) SCOUT will not ask for a password verification.

Username

If you specify a username SCOUT will enter it automatically if you add a material to the database.

Database

In order to connect to a database different from the default database you have to use the **File|**

Options|Database command in the main window. Specify the new database folder in the subsequent dialog and you are connected to the new database.

Sample description

Use this command to enter the description of the sample. The sample description can be displayed in a view using the corresponding [view object](#).

Sample descriptions are usually used in combination with [batch control](#).

Toolbar

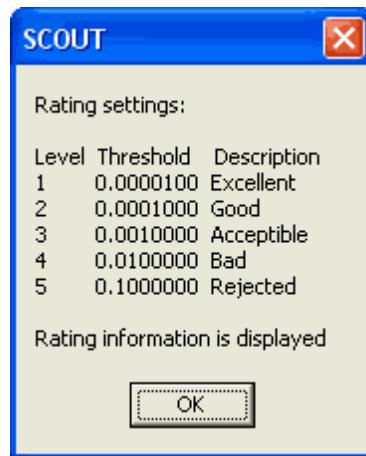
This is a submenu with switches that control the visibility of the various toolbar buttons.

See the section about [configuring SCOUT for routine work](#).

Rating

This submenu gives access to the rating settings. The following submenu commands are available:

- **Show:** check or uncheck this option to show (or not show) rating information
- **Change level threshold:** Use this command to modify the threshold of a quality level
- **Change level description:** Use this command to modify the description of a quality level
- **Show settings:** This command opens a short dialog like the following which displays the current settings:
-

**Batch control**

Submenu related to the batch control window with the following commands:

- **Synchronize batch control window: Right side:** If you check this option the batch control window will appear to the right side of the main window and fill the rest of the screen whenever you re-size the main window
- **Synchronize batch control window: Bottom:** If you check this option the batch control window will appear at the bottom of the main window and fill the rest of the screen whenever you re-size the main window
- **Batch control always on top:** When this option is checked the batch control window will always be on top of other windows

Appearance

Submenu with the following items

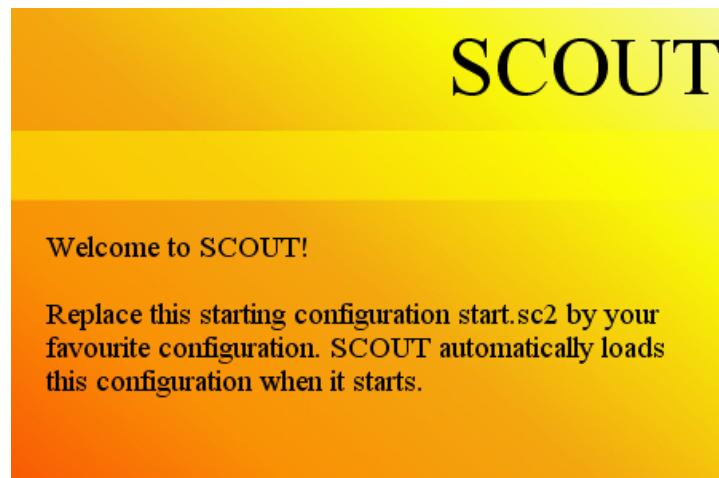
- **Main window dimensions:** Use this command to set the size of the main window in 4 short dialogs. You are asked for the position of the upper left corner on the screen and the window

width and height.

- **Laptop dimensions:** Use this command to fill the typical screen of a laptop computer with the main window (1024 by 768 pixels). This command is convenient if you prepare [presentations](#) for beamers with this resolution.
- **Statusbar:** This menu item can be switched on and off. The statusbar at the bottom of the main window is displayed or not depending on the setting.
- **Menu visible at start:** Uncheck this option if you want to hide the menu after loading the configuration. This is useful for [presentations](#)
- **Presentation style:** If you check this option the main window is shown without title bar, surrounding frame and Windows control elements:



With switched-off statusbar and menu it looks like this:



- **Laptop presentation style:** The same as the 'Presentation style' command above, but with simultaneous re-scaling to the 1024*768 resolution
- **Show sliders after loading:** This is a switch that controls if sliders are automatically displayed at program start. See the section about [configuring SCOUT for routine work](#).
- **Show spectra after loading:** This is a switch that controls if the spectra windows are automatically displayed at program start. See the section about [configuring SCOUT for routine work](#).

Data acquisition

Submenu with items related to [spectrum recording](#):

- **Save acquired spectra:** Check this item if you want to save the measured spectra to data files

after [data acquisition](#)

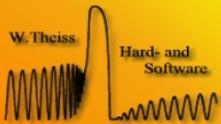
- **Start fit after acquiring new spectra:** If checked, a [parameter fit](#) is automatically started when the [data acquisition](#) has finished

Animation

Submenu to control parameter animation:

- **Parameter animation interval:** Use this command to set the delay between two subsequent states of the animation

SCOUT



Part

XV

15 Tools

15.1 Overview

SCOUT communicates with some tools by easy drag&drop exchange of optical spectra. The programs are explained shortly in the following. All tools mentioned below are contained in your SCOUT licence.

15.2 Data Factory

Data factory is a utility to create and manipulate data.

With Data factory you can

- Compute new data or manipulate existing ones by applying user-defined formulas
- Use up to three input spectra in your formula
- Create data by drawing using spline interpolation
- Reduce the number of data points by averaging
- Smooth curves
- Fourier transform data

15.3 Collect

Collect is used to display many spectra in one graph. The present version of Collect supports 2D and 3D plots.

15.4 DirectDF

The functionality of DirectDF has been integrated into SCOUT when the [Pane wizard](#) has been developed. If you have SCOUT or CODE you do not need DirectDF any more - just use the pane wizard!

This is a small tool for the direct determination of dielectric functions. It can be applied if you have recorded a transmittance as well as a reflectance spectrum on the same sample spot, and if, in addition, the sample is just a thick layer of the material under investigation. No interference fringes must appear in the spectra, although multiply reflected partial waves are allowed. Both spectra have to be recorded at almost normal incidence of light. From the two input spectra and the sample thickness (which must also be known before) the dielectric function is computed. The transmission must be significantly different from zero.

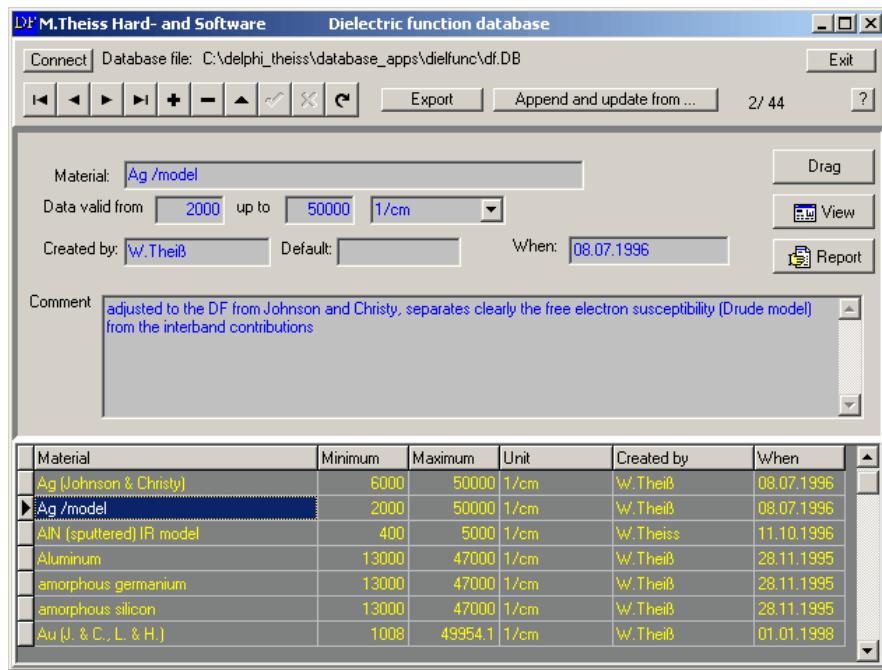
This method applies to thick substrates with both sides polished such as microscope slides, silicon wafers polished on both sides or KBr plates.

15.5 Digit

This tool is used to digitize curves manually or semi-automatic from literature sources. You must have a Windows-bitmap (*.bmp) obtained from a scanner or a video frame grabber.

15.6 DF_DB

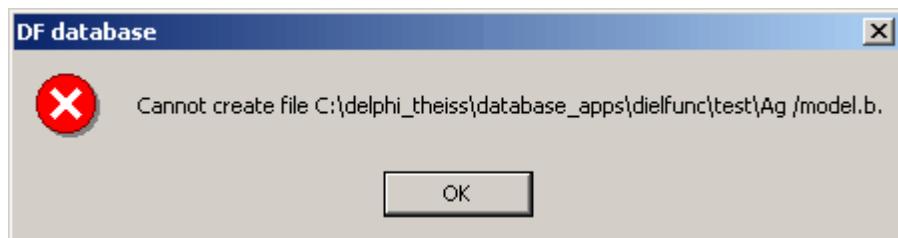
This tool converts a database of optical constants from previous SCOUT versions (up to version 2) to version 2.1 which is necessary because we have changed the structure of the [database](#) completely. In order to convert your dielectric function database to the new database system start the program df_db which is contained in the complete setup of SCOUT 2.1. The program looks like the old database window:



Use the **Connect** button in the upper left corner to connect to the database you want to convert. Then press the **Export** button in the upper center and select the folder where you want the new database entries be created. Select the subfolder 'Materials' in the new database folder structure. Once you have selected the target folder you are asked if you want to overwrite existing entries automatically:

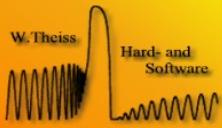


Having received the answer to this question, DF_DB starts to convert the data. The material names of the old database are used to create the required datafiles of the new database. If one of these names contains an illegal character such as >, <, %, \ or / you will get an error message and the conversion is stopped. In the example above the name 'Ag /model' is not allowed as filename and the following error message pops up:



In this case you have to rename the material and start the export again.

SCOUT



Part

XV

16 Ole automation

16.1 Introduction

16.1.1 Introductory remarks

The SCOUT software developed for optical analysis by spectrum simulation is a Windows based stand-alone solution. The typical SCOUT user works in a research environment, has some background in optical spectroscopy and is able to setup new models for optical constants and layer stacks.

Communication with other programs is possible by file import and export and some clipboard transactions.

For routine spectroscopy (product and process control) simplified user interfaces have been developed. While there is an almost complete SCOUT program doing its work in the background the 'front end window' has only a few buttons being operated by users without spectroscopic background.

A typical sequence is the following: First a so-called 'method' is loaded (i.e. a complete configuration containing all necessary models for a given problem) which is then used to analyze many spectra of a certain type of samples. A spectrum is loaded (or recorded with direct spectrometer control from within the front end application), the parameter fitting is started and stopped after a certain level of quality has been reached, and finally the obtained parameters are stored in a proper way.

Following increasing demands for external control we have turned SCOUT into an OLE automation server accessible by all Windows programs supporting OLE automation. The SCOUT OLE automation server is described in this user guide.

After a description of the installation procedure the public properties and methods of SCOUT OLE automation objects are discussed. Then we show how you can automate your SCOUT work using Visual Basic for Applications (VBA) macros in Excel. Programs providing easy access to tables of data like spreadsheet applications are certainly best suited for SCOUT work. This will be clear from the shown examples. Nevertheless, you can use any application that can act as an OLE automation client (or controller) to work with the SCOUT server.

16.1.2 Installation

Automatic installation

The installation of the OLE server is done by the SCOUT setup routine. It registers the application as OLE automation server named 'scout.scoutole' which is from now on available to all OLE automation clients like LabView, Excel, C++,

Manual registration

You can unregister and register OLE servers manually, using the Windows command line with administrator rights. In order to unregister SCOUT as OLE server, use the command

```
c:\program files\scout\scout.exe /unregserver
```

where the path to the executable file has to be adjusted, of course.

To manually register SCOUT please use the command
`c:\program files\scout\scout.exe /regserver`

Using OLE automation

During your work with the automation server you will have to use configuration files and – eventually

– files that contain sets of fit parameters. It is recommended that you collect those and the experimental data files that are going to be analyzed in appropriate directories.

After the automation server is registered in the Windows registry (see the installation section) OLE automation clients can create automation objects and control these by their properties and methods. To give an example, the following two lines of Visual Basic code generate the new automation object 'scout' whose methods and properties can be used afterwards:

```
Public scout As Object  
Set scout = CreateObject("scout.scoutole")
```

While methods are called to induce some server activity object properties are used to set or recall some object variables. This section describes the SCOUT OLE automation object properties and their use.

16.2 Automation object properties

16.2.1 a_filename (string)

Executes the standard dialog to select a filename. Returns the selected name. VBA example:

```
Range("results").Offset(4 + i, 1) = scout.a_filename;
```

Writing this property has no effect.

16.2.2 absorption_coefficient ...

Returns the absorption coefficient of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property. The unit of the absorption coefficient is 1/cm.

Writing this property has no effect.

16.2.3 average_time (real number)

Reading this number you obtain the average time (in seconds) used for one iteration of the fitting procedure.

VBA example:

```
Range("results").Offset(4 + i, 1) = scout.average_time;
```

Writing this property has no effect.

16.2.4 batch_configuration_file (string)

Setting the batch_configuration_file property you can load the content of the batch operation worksheets. You have to pass the name of an existing file (including the extension *.xls) as shown in the following VBA example:

```
scout.batch_configuration_file = "f:\examples\batch_parameters.xls"
```

This statement causes the loading of the specified batch configuration file.

16.2.5 batch_fit_deviat ion (sample:string) (float)

After the performance of a batch fit you can use this command to get the obtained fit deviation for the specified sample.

VBA example:

```
quality = scout.batch_fit_deviat ion("s413 #3")
```

16.2.6 batch_fitparameter_name (index: integer) (string)

Returns as a string the name of the batch fit parameter specified by the index parameter.

VBA example:

```
this_name = scout.batch_fitparameter_name(5)
```

16.2.7 batch_fitparameter_value (sample: string, index: integer) (float)

Returns as floating point number the value of the batch fit parameter specified by index, obtained in the batch fit for the sample specified by the sample parameter.

VBA example:

```
this_value = scout.batch_fitparameter_value("s413 #3", 5)
```

16.2.8 batch_fp_set_stop_reason (sample : string, fp_set_index: integer) (string)

Returns the reason why the batch fit of sample samplename has been finished for the fit parameter set given by fp_set_index.

16.2.9 batch_input_path(index : integer) (string)

This property is the network path to the input spectra of the spectrum specified by the index parameter. The filenames of the input files are composed from the batch_input_path, the batch_spectrum_filename and the extension defined in the 'Input options' worksheet of the batch control workbook.

You can read or write the batch_input_path.

VBA example:

```
scout.batch_input_path(1) = "C:\test\data\trans\"  
scout.batch_input_path(2) = "C:\test\data\ref\"
```

16.2.10 batch_name_of_spectrum (i : integer) (string)

Returns (as string) the name of the i-th spectrum in the batch control workbook. The spectra are labeled 1,2,..., n if there are n spectra for each sample.

VBA example:

```
my_spectrum_name = scout.batch_name_of_spectrum(2)
```

Writing this property has no effect.

16.2.11 batch_number_of_spectra (integer)

Returns the number of spectra defined in the current batch control workbook. Use this number to iterate through all spectra in a loop.

VBA example:

```
nbs = scout.batch_number_of_spectra
```

Writing this property has no effect.

16.2.12 batch_prefit_result(sample : string) (string)

Returns the pre-fit set found for the specified sample during the batch fit.

VBA example:

```
start_set = scout.batch_prefit_result("s413 #3")
```

16.2.13 batch_sample_parameter(sample : string) (float)

Use this property to set or read the numeric parameter of a sample in batch control workbooks. Specify the sample by its name, and pass the numerical parameter as float.

VBA example:

```
scout.batch_sample_parameter("s413 #3") = 13.56
```

16.2.14 batch_spectrum_filename (sample : string, index : integer) (string)

Specifies the input spectrum for the spectrum specified by the index parameter and the sample specified by the sample parameter. You can read or write this item. The input spectra are labeled 1, 2, ... n if there are n spectra for each sample.

VBA example:

```
scout.batch_spectrum_filename("s413 #3", 1) = "c:\test\s413_3_r"
```

16.2.15 batch_status (integer)

Inspect this value to find out if SCOUT is busy with a batch fit operation.

If the value of batch_status is 1 the batch fit is still active.

If the value is 0 the batch fit is finished and you can pick up the results.

VBA example:

This example starts the batch fit and then waits in a loop until the batch fit is done. Every 3 seconds the batch status is checked. Finally a message box indicates the termination of the batch fit.

```
scout.batch_go
While scout.batch_status <> 0
    Application.Wait Now + TimeSerial(0, 0, 3)
Wend
msgbox "Batch finished"
```

16.2.16

color_coordinate(spectrum_index,simulated,color_type,illuminant,observation_angle) (float)

Computes the color coordinate of a spectrum.

The index of the spectrum in the list of spectra is given by the parameter 'spectrum_index'. The spectra are counted 1, 2, 3, ...

Depending on the parameter 'simulated' the color coordinate of the simulated or the measured spectrum is computed:

simulated = 0: The color coordinate of the measured spectrum is computed
 simulated = 1: The color coordinate of the simulated spectrum is computed

The 'color_type' parameter selects the color coordinate to be computed. The following choices are possible:

0:	L*
1:	a*
2:	b*
3:	X
4:	Y
5:	Z
6:	x
7:	y
8:	z
9:	L
10:	a
11:	b

'Illuminant' selects one of the following illumination spectra:

0:	A
1:	D65
2:	C

With 'observation angle' you can set the observation angle for which the color coordinate is computed:

0:	2°
1:	10°

Example:

Calling `scout.color_coordinate(1, 1, 1, 0, 1)` returns a* of the first simulated spectrum in the list of spectra. It is computed for illuminant A and 10° observation angle.

16.2.17 configuration_file (string)

Setting the `configuration_file` property you specify the optical analysis method that is stored in a complete configuration file. You have to pass the name of an existing file as shown in the following VBA example:

```
scout.configuration_file = "f:\examples\s-pol.sio"
```

This statement causes the loading of the specified configuration file.
Reading this property returns the name of the configuration file that was loaded by SCOUT in the last 'load configuration' action.

16.2.18 dielectric_function_name(i:integer) (string)

Returns the name of the i th dielectric function. The following VBA command fills a certain cell with the name of the 5th dielectric function:

```
Range("results").Offset(5, 1) = scout.dielectric_function_name(5)
```

Setting this property has no effect.

16.2.19 df_real_part(n : integer; wavenumber : real number) (real number)

Returns the value of the real part of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.20 df_imag_part(n : integer; wavenumber : real number) (real number)

Returns the value of the imaginary part of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.21 experimental_spectrum_comment(n:integer): string

Returns the comment of the n-th measured spectrum in the spectrum list.

Writing this property has no effect.

16.2.22 experimental_spectrum_value(n:integer;wavenumber:real number):real number

Returns the value of the n-th measured spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.23 fft_thickness

Performs the FFT thickness analysis and returns the obtained thickness.

Parameters:

n_spec: number of the spectrum in the spectrum list
wn_min: wavenumber minimum of the used spectral range
wn_max: wavenumber maximum of the used spectral range
fft_np: number of data points used to sample the spectral range
d_min: expected thickness minimum
d_max: expected thickness maximum
n: refractive index of the layer

VisualBasic example: `scout.fft_thickness(1,650,7000,2048,5,50,2.0)`

16.2.24 fft_strength

Returns the height of the peak determined in the last `fft_thickness` routine for spectrum `n_spec`. This number can be used to get information on the quality of the analysis.

16.2.25 file_format (integer)

Sets the file format for subsequent [import actions](#).

The following values are implemented up to now:

format	File format
1	Standard
2	xy-format
3	JCAMP
4	SpectraCalc
5	Perkin-Elmer
6	Zeiss
7	Opus

16.2.26 file_object_generation (real number)

Returns the object generation of the last configuration file that has been loaded into SCOUT.

16.2.27 fit_deviations (real number)

Returns the current fit deviation.

Dim this_deviations as single
this_deviations = scout.fit_deviations

Setting this property has no effect.

16.2.28 fit_parameter_mode(i:integer) (integer)

Sets (or reads) the variation mode of the n-the fit parameter. The following values are implemented at present:

mode	meaning
-1	Frozen
0	Downhill simplex
1	Computed from master parameters by formula
2	Fit on a grid with 10 points
3	Fit on a grid with 20 points
4	Fit on a grid with 50 points
5	Fit on a grid with 100 points

16.2.29 fit_parameter_name(i:integer) (string)

Returns the name of the fit parameter. The following VBA command fills a certain cell with the name of the selected fit parameter:

```
Range("results").Offset(4 + i, 1) = scout.fit_parameter_name(i)
```

Setting this property has no effect.

16.2.30 fit_parameter_set (string)

Setting the `fit_parameter_set` property you specify the file from which a fit parameter set is to be loaded. Reading this property has no effect.

Reading this property returns the name of the currently loaded fit parameter set.

VBA example:

```
scout.fit_parameter_set = "c:\buffer\test.fps"
```

16.2.31 fit_parameter_value(i:integer) (real number)

Returns the present value of the fit parameter i. The n fit parameters are labeled from 1 to n.

```
Dim my_value as single
my_value = scout.fit_parameter_value(12)
```

Set this property to pass a value to a fit parameter in SCOUT:

```
Dim my_value as single
my_value=3.4
scout.fit_parameter_value(12)=my_value
```

16.2.32 fit_parameter_value_min(i:integer) (real number)

Sets the lower boundary (minimum) of the range of variation of the n-th fit parameter. Reading this quantity returns the current value of the lower boundary.

16.2.33 fit_parameter_value_max(i:integer) (real number)

Sets the upper boundary (maximum) of the range of variation of the n-th fit parameter. Reading this quantity returns the current value of the upper boundary.

16.2.34 fitting (integer)

Returns information on the fit activity of SCOUT.

The value of fitting is 0 if there is no fit running, fitting is equal to 1 during the fit.

VBA example:

```
if scout.fitting=1 then ...
```

Writing this property has not effect.

16.2.35 foundfiles (integer)

Returns the number of files that were selected in the last 'searchfiles' action.

This number can be used to iterate through all the found filenames. The latter can be obtained by the [found_filename](#) function.

16.2.36 found_filename(i:integer) (string)

This function returns the complete filename of the i-th selected file in the last [searchfile](#) action.

16.2.37 fp_set_name (index : integer) (string)

Returns the name of the fit parameter set with index 'index'.

16.2.38 get_sim_val1(n : integer; wavenumber : real number) (real number)

Returns the 1st simulated value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property. If the spectrum is an ellipsometry spectrum, there are two simulated quantities: Psi and delta, or cos(psi) and tan(delta). In that case the 1st value is the psi-related quantity. The second quantity, related to delta, can be obtained using the function [get_sim_val2](#).

Writing this property has no effect.

16.2.39 get_sim_val2(n : integer; wavenumber : real number) (real number)

Returns the 2nd simulated value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property. This is used in cases like ellipsometry where there are two simulated quantities: Psi and delta, or cos(psi) and tan(delta). In that case the 1st value (obtained by the function [get_sim_val1](#)) is the psi-related quantity. The second quantity is the one related to delta.

Writing this property has no effect.

16.2.40 grid_test_max (single)

This property sets or gets the minimum value of a [parameter variation on a fixed grid](#).

16.2.41 grid_test_min (single)

This property sets or gets the minimum value of a [parameter variation on a fixed grid](#).

16.2.42 grid_test_no_points (integer)

This property sets the number of points used in the [parameter variation on a fixed grid](#).

16.2.43 grid_test_para (integer)

This property sets the fit parameter varied in a [parameter variation on a fixed grid](#).

16.2.44 incidence_angle(i:integer) (real number)

Returns the angle of incidence of the i th spectrum. The following VBA command fills a certain cell with the angle of incidence of the 2nd spectrum:

```
Range("results").Offset(2, 1) = scout.incidence_angle(2)
```

Setting this property sets the value of the incidence angle of the i th spectrum in the SCOUT spectrum list.

16.2.45 kappa(n : integer; wavenumber : real number) (real number)

Returns the imaginary part of the refractive index of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.46 layer_material(n,m: integer) (string)

This function returns the name of the material assigned to layer m in the layer stack n (stacks and layers are counted 1,2,3, ...).

16.2.47 layer_thickness(n,m : integer) (float)

Sets or reads the thickness of the m-th layer in the n-th layer stack to the value passed to the property. Both the layer stacks and the layers in a layer stack are counted starting at 1. Reading this property returns the current value of the layer thickness.

16.2.48 layer_type(n,m : integer) (string)

This function returns the type of layer m in the layer stack n (stacks and layers are counted 1,2,3, ...).

16.2.49 load_experiment(n: Integer; const filename: String; format: Integer);

Loads for the n-th spectrum in the spectrum list experimental data from the file specified by 'filename'. The format used is given by the 'format' parameter:

format	File format
1	Standard
2	xy-format
3	JCAMP
4	SpectraCalc
5	Perkin-Elmer
6	Zeiss
7	Opus

VBA example:

```
scout.load_experiment(5,"c:\buffer\test.spc",4);
```

16.2.50 main_window_height

Height of the main window.

16.2.51 main_window_left

Screen coordinate (x) of the main window's upper left corner.

16.2.52 main_window_top

Screen coordinate (y) of the main window's upper left corner.

16.2.53 main_window_width

Width of the main window.

16.2.54 material_is_in_material_list (material_name : string) (integer)

Returns 1 if a material named material_name is in the list of materials, 0 otherwise.

16.2.55 material_is_in_database (material_name : string) (integer)

Returns 1 if a material named material_name is in the database, 0 if not.

16.2.56 measurement(i:integer) (string)

Setting this string to an appropriate filename will cause SCOUT to import the experimental spectrum of the i-th spectrum in the spectrum list from the specified file. The expected format of the measured data must be set before using the [file_format](#) property.

16.2.57 multidimensional_gridfit (integer)

Returns or sets the option 'multidimensional gridfit'.

A value of 0 means 'multidimensional gridfit' is switched off, a value of 1 stands for 'multidimensional gridfit' is switched on.

VBA examples:

```
scout.multidimensional_gridfit=1
.
.
.
check = scout.multidimensional_gridfit
```

16.2.58 number_of_dielectric_functions (integer)

Returns the number of dielectric functions in the present SCOUT configuration. The dielectric functions are counted from 1 to number_of_dielectric_functions.

```
Dim np as integer
np = scout.number_of_dielectric_functions
```

Setting this property has no effect.

16.2.59 number_of_fit_parameters (integer)

Returns the number of fit parameters used at present. This can be used to write loops with some action executed for all fit parameters. For example, you could create a table with the fit parameter names and current values.

```
Dim np as integer
np = scout.number_of_fit_parameters
```

Setting this property has no effect.

16.2.60 number_of_fp_sets (integer)

Returns the number of fit parameter sets.

16.2.61 number_of_layer_stacks

Returns the number of layer stacks.

16.2.62 number_of_layers_in_stack(n:integer)

Returns the number of layers in the n-th layer stack (stacks are counted 1,2,3, ...).

16.2.63 number_of_spectra (integer)

Returns the number of spectra in the present SCOUT configuration. The spectra are counted from 1 to number_of_spectra.

*Dim np as integer
np = scout.number_of_spectra*

Setting this property has no effect.

16.2.64 object_generation (real number)

Returns the object generation of the SCOUT program.

16.2.65 optical_function(a_string : string) (real number)

Returns the value of the optical function defined by the text passed in the string a_string. If the optical function cannot be evaluated a value of 0 is returned.

Example:
x1 = scout.optical_function("SiN(n at 400 nm)")

Setting this property has no effect.

16.2.66 optical_function_by_index(i : integer) (real number)

Returns the value of the i-th optical function in the current list of optical functions in SCOUT.

Example:
x1 = scout.optical_function_by_index(3)

Setting this property has no effect.

16.2.67 parameter (i:integer) (real value)

Sets or reads the value of the i th parameter.

Defined for future use. Until now no parameters are used in SCOUT.

16.2.68 rating (string)

Returns the current rating.

16.2.69 rating_threshold (level_index:integer) (float)

Returns or sets the rating threshold for the rating level specified by level_index.

16.2.70 rating_description (level_index:integer) (string)

Returns or sets the string for the rating level specified by level_index.

16.2.71 reference_spectrum(n:integer;format:integer):string

Setting this string to an appropriate filename will cause SCOUT to import the reference spectrum of the i-th spectrum in the spectrum list from the specified file. The format of the measured data must be set by the format parameter.

The following values for the format are implemented up to now:

format	File format
1	Standard
2	xy-format
3	JCAMP
4	SpectraCalc
5	Perkin-Elmer
6	Zeiss
7	Opus

VisualBasic example:

```
scout.reference_spectrum(1, 4) = "c:/temp/dummy2.spc"
```

16.2.72 refractive_index(n : integer; wavenumber : real number) (real number)

Returns the refractive index of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.73 sample_description: string

Read and write the sample description which is used in some views.

16.2.74 simulated_spectrum_value(n:integer;wavenumber:real number):real number

Returns the value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

16.2.75 simulation_exportfile(n:integer;format:integer):string

Assigning a filename to simulation_exportfile will cause SCOUT to save the simulated data of the n'th spectrum in the spectrum list to the specified file. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

format	File format
1	Standard
2	xy-format
4	SpectraCalc

VisualBasic example:

```
scout.simulation_exportfile(1, 4) = "c:/temp/dummy2.spc"
```

16.2.76 spectrum_graphics_xmax(n : integer) (real number)

Returns the maximum of the displayed x-range in the graphics of the n-th spectrum in the list of spectra.

16.2.77 spectrum_graphics_xmin(n : integer) (real number)

Returns the minimum of the displayed x-range in the graphics of the n-th spectrum in the list of spectra.

16.2.78 spectrum_graphics_xticks(n : integer) (real number)

Returns the distance between tick marks on the x-axis in the graphics of the n-th spectrum in the list of spectra.

16.2.79 spectrum_graphics_xtitle(n : integer) (real number)

Returns the title of the x-axis in the graph of the n-th spectrum in the list of spectra.

16.2.80 spectrum_graphics_xunit(n : integer) (real number)

Returns the unit of the x-axis in the graph of the n-th spectrum in the list of spectra.

16.2.81 spectrum_graphics_ymax(n : integer) (real number)

Returns the maximum of the displayed y-range in the graphics of the n-th spectrum in the list of spectra.

16.2.82 spectrum_graphics_ymin(n : integer) (real number)

Returns the minimum of the displayed y-range in the graphics of the n-th spectrum in the list of spectra.

16.2.83 spectrum_graphics_yticks(n : integer) (real number)

Returns the distance between tick marks on the y-axis in the graphics of the n-th spectrum in the list of spectra.

16.2.84 spectrum_graphics_ytitle(n : integer) (real number)

Returns the title of the y-axis in the graph of the n-th spectrum in the list of spectra.

16.2.85 spectrum_name(i:integer) (string)

Returns the name of the i th spectrum.

The following VBA command fills a certain cell with the name of the 2nd spectrum:

```
Range("results").Offset(2, 1) = scout.spectrum_name(2)
```

Setting this property has no effect.

16.2.86 spectrum_range_max(n : integer) (real number)

Returns the maximum of the spectral range of the n-th spectrum in the list of spectra.

16.2.87 spectrum_range_min(n : integer) (real number)

Returns the minimum of the spectral range of the n-th spectrum in the list of spectra.

16.2.88 spectrum_range_points(n : integer) (integer)

Returns the number of points of the n-th spectrum in the list of spectra.

16.2.89 spectrum_range_unit(n : integer) (string)

Returns the spectral unit of the n-th spectrum in the list of spectra.

16.2.90 tolerance (real number)

Use this property to read and to set the tolerance which is used for terminating the downhill simplex fitting method.

16.2.91 weight_formula (spectrum_index, subindex : integer) (string)

Returns or sets the weight formula of the spectrum given by spectrum_index. If the spectrum object is an ellipsometry object you have to set subindex to 1 to refer to the weight_formula for psi. Using a value of 2 for subindex selects the weight formula for delta.

16.3 Automation object methods

16.3.1 acquire

This command executes the global acquire command in SCOUT: All spectrometers are told to record new spectra.

16.3.2 acquire_dark

Use this command to tell all spectrometers to record their dark spectrum.

16.3.3 acquire_reference

Use this command to tell all spectrometers to record their reference spectrum.

16.3.4 add_layer_definition_on_top(stack_index : integer;definition : string)

A new layer is added on top of the current layer stack with index 'stack_index' in the list of layer stacks (stacks are counted 1, 2, 3, ...).

The new layer is specified by the string 'Definition'. The string may be composed by several terms, separated by tab characters. A tab character has the ASCII code 9 (In VisualBasic, for example, this is obtained by Chr(9)).

Possible layer definitions are

- "Simple layer" + tab + string specifying the material + tab + string specifying the thickness in cm (!) + tab + string specifying the partial wave superposition ("coherent" or "incoherent")
- "Thin film" + tab + string specifying the material + tab + string specifying the thickness in cm(!)
- "Thick layer" + tab + string specifying the material + tab + string specifying the thickness in cm(!)
- "Variable thickness" + tab + string specifying the material + tab + string specifying the thickness in cm(!) + tab + string specifying the thickness variation in cm (!) + tab + string specifying the thickness profile ("1" rectangular, "2" Gaussian)
- "Rough interface" + tab + string specifying the roughness formula + tab + string specifying the value of the c1 parameter + tab + string specifying the value of the c2 parameter
- 'End superlattice' (no parameter)
- 'Begin superlattice'+tab+'10' (the last partial string specifies the number of repetitions of the superlattice)

VisualBasic examples:

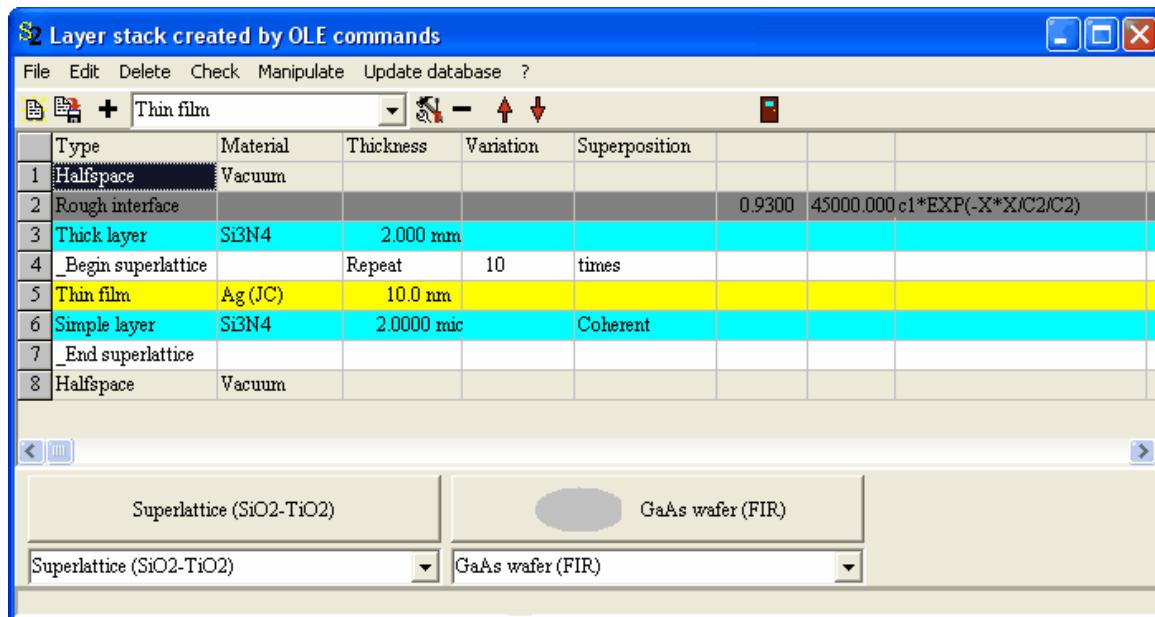
The VisualBasic code

```

scout.clear_fit_parameters
scout.clear_layer_stack (1)
scout.clear_material_list
Call scout.add_layer_definition_on_top(1, "End superlattice")
Call scout.add_layer_definition_on_top(1, "Simple layer" + Chr(9) + "Si3N4" + Chr(9) + "0.0002" +
Chr(9) + "coherent")
Call scout.add_layer_definition_on_top(1, "Thin film" + Chr(9) + "Ag (JC)" + Chr(9) + "0.000001")
Call scout.add_layer_definition_on_top(1, "Begin superlattice" + Chr(9) + "10")
Call scout.add_layer_definition_on_top(1, "Thick layer" + Chr(9) + "Si3N4" + Chr(9) + "0.2")
Call scout.add_layer_definition_on_top(1, "Rough interface" + Chr(9) + "c1*EXP(-X*X/C2/C2)" +
Chr(9) + "0.93" + Chr(9) + "45000.0")
scout.update_data
scout.update_plot

```

creates the following layer stack:



Here is an VB example for the 'Variable thickness' layer type:

```
Call scout.add_layer_definition_on_top(1, "Variable thickness" + Chr(9) + "Si3N4" + Chr(9) +
"0.0006" + Chr(9) + "0.00004" + Chr(9) + "2")
```

16.3.5 add_layer_on_top (stack_index : integer; layer_type,material:string;thickness:real number; thickness_unit : string)

A new layer is added on top of the current layer stack with index 'stack_index' in the list of layer stacks (stacks are counted 1, 2, 3, ...).

The type of the layer is specified by the string 'layer_type'. Possible types are

- 'Thin film' (coherent superposition of partial waves)
- 'Thick layer' (incoherent superposition of partial waves)

The material assignment is given by the 'material' parameter. If the wanted material is found in the list of materials, it is taken from there. If not, the database is searched for the requested material. In the case of success, the material is transferred from the database to the list of materials and used

for the layer stack assignment.

The thickness of the new layer is specified by the parameters 'thickness' (a real number) and 'thickness_unit' (a string). Possible values for the thickness unit are

- 'nm'
- 'micron'
- 'mm'
- 'm'

VisualBasic examples:

```
Call scout.add_layer_on_top(1, "Thin film", "Si3N4", 0.45, "micron")
Call scout.add_layer_on_top(1, "Thin film", "Ag (JC)", 19.4, "nm")
```

16.3.6 add_to_report(text : string)

Adds the submitted string to the SCOUT report window.

16.3.7 automatic_spectrum_scaling(i : index)

Executes automatic scaling of the graphics of the i-th spectrum.

16.3.8 batch_add_sample (name: string)

A new sample column is added to the 'Results' worksheet of the batch control workbook. In subsequent calls you have to [specify the input spectra](#) for this sample.

VBA example:

```
scout.add_sample("S413 #1")
```

16.3.9 batch_clear_samples

Use this command to clear the 'Results' page of the batch control workbook, with the exception of the first column. All input spectra and results are deleted. This is used to prepare a new batch run. Using the [batch_add_sample](#) command you can add new samples afterwards.

VBA example:

```
scout.batch_clear_samples
```

16.3.10 batch_restore_fit_results (sample : string)

Reloads the obtained batch fit parameters and the spectra for the specified sample.

16.3.11 batch_go

Starts the batch fit based on the current batch control workbook settings. SCOUT will be busy with the batch fit activity for a while, but still respond to OLE commands. You can inspect the value of the property [batch_status](#) to find out if the batch fit operation is still active.

VBA example:

scout.batch_go

16.3.12 clear_fit_parameters

This command deletes all items from the list of fit parameters.

VisualBasic example:

```
scout.clear_fit_parameters
```

16.3.13 clear_layer_stack (index : integer)

Deletes all layers of a layer stack and assigns 'Vacuum' to the top and bottom halfspaces.

The position of the layer stack in the list of layer stacks is given by index (stacks are numbered 1, 2, 3, ...).

VisualBasic example:

```
scout.clear_layer_stack (1)
```

16.3.14 clear_material_list

Clears the list of materials: All items are deleted, with the exception of 'Vacuum'.

VisualBasic example:

```
scout.clear_material_list
```

16.3.15 clear_report

Clears the SCOUT report window to prepare it for subsequent entries.

16.3.16 create_fit_parameter (list, object, subobject, subsubobject, parameter : integer) (copy)

Creates a new fit parameter in the list of fit parameters. The parameters passed to this procedure have the following meaning:

list: the SCOUT list with the object the fit parameter of which is to be created

- 1 Materials
- 2 Layer stacks
- 3 Spectra list
- 4 Master parameter list

object: the index of the object in the list. The objects are counted 1, 2, 3, ...

subobject: if the object has subobjects this parameter specifies the index of the wanted subobject. This can be, for example, a susceptibility in the list of susceptibilities of a dielectric function model.

subsubobject: if the subobject has again subobjects of its own, this is the index of the wanted subobject.

parameter: the index of the parameter to be selected as fit parameter. If you are not sure about the index of a parameter you can drag the object to the fit parameter list and see which parameters are

created in which order. The 'create_fit_parameter' OLE command uses exactly the same number and sequence of parameters.

Here is a simple example making use of the create_fit_parameter command:

```
'Excel VisualBasic demo of a remote fitting procedure using OLE automation
```

```
'This object is the SCOUT automation object  
Public scout As Object
```

```
Public Sub fit()
```

```
    'Create the OLE server
```

```
    Set scout = CreateObject("scout_98.scoutole")
```

```
    'Load a pre-configured configuration with a suitable reflectance spectrum and spectral range  
    scout.configuration_file = "c:\test\ole_test.sc2"
```

```
    'Show SCOUT on the screen
```

```
    scout.Show
```

```
    'Clear the layer stack - we will build up a new one in a moment
```

```
    scout.clear_layer_stack (1)
```

```
    'Clear the list of materials - materials will be taken from the database
```

```
    scout.clear_material_list
```

```
    'Create a thick layer with 2 mm thickness and fill it with BK7 glass, a material named "Glass Type BK7" must be in the database
```

```
    Call scout.add_layer_definition_on_top(1, "Thick layer" + Chr(9) + "Glass Type BK7" + Chr(9) +  
    "0.2")
```

```
    'Create a thin silver filme on top of the glass substrate, take the material "Ag model" from the  
    database
```

```
    Call scout.add_layer_definition_on_top(1, "Thin film" + Chr(9) + "Ag model" + Chr(9) + "0.000001")
```

```
    'Load a measured spectrum into the first spectrum object, using standard format
```

```
    Call scout.load_experiment(1, "C:\test\ag_10.std", 1)
```

```
    'Clear the list of fitparameters
```

```
    scout.clear_fit_parameters
```

```
    'From the list of layer stacks, take the first object
```

```
    'and then the second subobject. This is the silver layer.
```

```
    'Select parameter 1 as fit parameter (this is the thickness)
```

```
    Call scout.create_fit_parameter(2, 1, 2, 0, 1)
```

```
    'From the list of materials, take the 3rd object which is the silver model
```

```
    '(initially there was vacuum, then glass has been taken from the database,
```

```
    'and finally, as 3rd object, the Ag model was loaded).
```

```
    'Take the first susceptibility which is the Drude model.
```

```
    'Take the 2nd parameter of the Drude model which is the damping constant and select
```

```
    'it as fit parameter.
```

```
    Call scout.create_fit_parameter(1, 3, 1, 0, 2)
```

```
    'Set boundaries for the damping constant which is fit parameter 2 now (after the layer thickness).
```

```
    scout.fit_parameter_value_min(2) = 100
```

```
    scout.fit_parameter_value_max(2) = 10000
```

```

'The silver model contains several Brendel oscillators in the list of susceptibilities.
'Take the 2nd, 3rd, 4th and 5th susceptibility and select for each one parameter
'2 as fit parameter, i.e. the oscillator strength.
'This is just for demonstration - it is not really reasonable to vary these parameters in the fit.
Call scout.create_fit_parameter(1, 3, 2, 0, 2)
Call scout.create_fit_parameter(1, 3, 3, 0, 2)
Call scout.create_fit_parameter(1, 3, 4, 0, 2)
Call scout.create_fit_parameter(1, 3, 5, 0, 2)

>Select - just for demonstration - parameter #1 of the first spectrum in the list of spectra.
'This is the angle of incidence of the reflectance spectrum.
Call scout.create_fit_parameter(3, 1, 0, 0, 1)

>Select - also for demonstration only - the first master parameter as fitparameter.
Call scout.create_fit_parameter(4, 1, 0, 0, 1)

'Re-compute the model and update the main view
scout.update_data
scout.update_plot

'Start the fit
scout.start
'Start a waiting loop which will finish when SCOUT stops its fit (or you as a user
'press the STOP button in SCOUT.
While scout.fitting = True
    'Wait for 1 second
    Application.Wait Now + TimeSerial(0, 0, 1)
    Wend

'Write results to a worksheet range
'fit parameter #1 is the thickness
Range("results").Offset(1, 1) = scout.fit_parameter_value(1)
'export the real part of the refractive index of the Ag model at 1000 nm wavelength
Range("results").Offset(1, 2) = scout.refractive_index(3, 10000000 / 1000)
'export the imaginary part of the refractive index of the Ag model at 1000 nm wavelength
Range("results").Offset(1, 3) = scout.kappa(3, 10000000 / 1000)

End Sub

```

16.3.17 df_load_from_file(n: Integer; const filename:String);

Using this command the n-th dielectric function is told to import data from a standard file specified by the 'filename' parameter.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

16.3.18 df_load_from_file_2(n: Integer; const filename, unit_string:String);

Using this command the n-th dielectric function is told to import data from a standard file specified by the 'filename' parameter.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

You can specify the spectral unit of the file data by setting the unit_string parameter to one of the following:

String transmitted		Spectral unit used
'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
'm m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

16.3.19 df_load_from_x_n_k_table(n:integer; const filename : string);

Using this command the n-th dielectric function is told to import data from a testfile specified by the 'filename' parameter. The file must contain a table with three columns. The first column has to contain the spectral position, followed by n and k.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

16.3.20 df_load_from_x_n_k_table_2 (n:integer; const filename : string);

Using this command the n-th dielectric function is told to import data from a testfile specified by the 'filename' parameter. The file must contain a table with three columns. The first column has to contain the spectral position, followed by n and k.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

You can specify the spectral unit of the file data by setting the unit_string parameter to one of the following:

String transmitted		Spectral unit used
'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
'm m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

16.3.21 export_simulated_spectrum(n:integer; filename:string; format : integer)

Use this command to save the simulated data of the n'th spectrum in the spectrum list to the file specified by filename. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

format	File format
1	Standard
2	xy-format
4	SpectraCalc

VisualBasic example:

```
scout.export_simulated_spectrum(1,"c:/temp/dummy2.spc",4)
```

16.3.22 df_to_clipboard (n : integer)

Tells the n-th material in the list of materials to copy its graphics to the clipboard.

16.3.23 export_measured_spectrum(n:integer; filename:string; format : integer)

Use this command to save the measured data of the n'th spectrum in the spectrum list to the file specified by filename. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

format	File format
1	Standard
2	xy-format
4	SpectraCalc

VisualBasic example:

```
scout.export_measured_spectrum(1,"c:/temp/dummy2.spc",4)
```

16.3.24 export_simulated_spectrum(n:integer; filename:string; format : integer)_2

Use this command to save the simulated data of the n'th spectrum in the spectrum list to the file specified by filename. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

format	File format
1	Standard
2	xy-format
4	SpectraCalc

VisualBasic example:

```
scout.export_measured_spectrum(1,"c:/temp/dummy2.spc",4)
```

16.3.25 grid_test

This procedure does the following: The fit parameter specified by the integer `grid_test_para` is varied from the value `grid_test_min` to the value given by `grid_test_max`. `Grid_test_no_points` are used to cover this parameter range. For each value the fit deviation is computed. After the procedure the fit parameter is set to the best value found on this grid of tried values. This feature can be used before an automatic fit to find good starting values for some of the fit parameters.

16.3.26 hide

Hides the SCOUT window.

16.3.27 hide_statusbar

Hides the status bar.

16.3.28 import_measured_data_via_variants(index : string; spectral_positions, spectral_data : variant; unit_string : string)

Use this command to pass experimental spectra to a SCOUT reflectance or transmittance spectrum.

The 'Index' parameter gives the index of the spectrum in the list of spectra (counted like 1, 2, 3, ...). The spectral positions (wavelengths, frequencies, wavenumbers, energies) and data (reflectance or transmittance) are passed through variant arrays - they must be 1-dimensional and of the same length. The parameter 'unit_string' sets the spectral unit (eV, 1/cm, THz, nanometer, wavenumber).

16.3.29 load_df(n : integer, filename : string)

Loads the n-th dielectric function from the specified file. The entries in the list of dielectric functions are numbered starting with 1.

Warnings:

- Be sure that the reading object and the file format are compatible.
- The content of the dielectric function object will be replaced by the data in the file.

VisualBasic example:

```
Call scout.load_df(1, "c:/temp/df2.dfm")
```

16.3.30 load_material_from_database (name : string)

This command loads a material from the database into the list of materials. The material is specified by the name parameter which must exactly match an item in the database.

VisualBasic example:

```
scout.load_material_from_database ("Si3N4")
```

16.3.31 load_experiment (n:integer, filename: string, format:integer)

Loads the file specified by filename into the n-th spectrum object, using the file format given by 'format'.

16.3.32 prepare_shutdown

In some cases it is not sufficient just to delete SCOUT as OLE server when you have finished your work. You might get access violation errors and corresponding message dialogs which prevent SCOUT from proper closing. In order to avoid this you should call the method `prepare_shutdown` as last OLE command before you shutdown the server.

16.3.33 print_fit(i : integer)

Tells SCOUT to print the graphics of spectrum i in the list of spectra. The purpose, of course, is to get information on the fit quality.

VBA example:

```
scout.print_fit(2)
```

16.3.34 rename_material (old_name, new_name : string)

Renames the material called old_name to new_name.

16.3.35 report_to_clipboard

Copies the content of the SCOUT report window to the clipboard.

16.3.36 report_to_file(filename:string)

Saves the content of the SCOUT report window to a text file the name of which is the string passed to the method.

16.3.37 save_configuration(filename : string)

Saves the complete SCOUT configuration to the specified file.

Warning: An already existing file it will be overwritten without warning.

VisualBasic example:

```
scout.save_configuration ("c:/temp/fit_sample1.sc2")
```

16.3.38 save_df(n : integer, filename : string)

Saves the n-th dielectric function to the specified file. The entries in the list of dielectric functions are numbered starting with 1.

Warning: An already existing file it will be overwritten without warning.

VisualBasic example:

```
Call scout.save_df(2, "c:/temp/df2.dfm")
```

16.3.39 scout_normal

Show the program window like a normal window which can be hidden by other windows.

16.3.40 scout_on_top

Show the program window on top of other windows on the screen.

16.3.41 searchfiles(dir,pattern : string)

Executes a dialog to select multiple files that can be used afterwards for SCOUT actions.
In 'dir' you can provide the intial path for the search, 'pattern' specifies a filename search pattern.

The selected filenames can be used afterwards using the functions ['foundfiles'](#) and ['found_filename'](#).

16.3.42 send_message(the_text : string)

Use this command to send a message to SCOUT. SCOUT will respond to the message with an action corresponding to the message.

Up to now this feature is not used.

16.3.43 set_bottom_halfspace_material (i : integer; name : string)

Assigns the material called name to the bottom halfspace of the i-th layer stack.

Stacks are counted 1, 2, 3, ...

A material called name must be in the list of materials. If it is not you can use the command [load_material_from_database\(name\)](#) in order to load the material from the database into the list of materials.

16.3.44 set_global_range(min, max : real number, no_points : integer, spectral_unit : string)

Use this command to change the range of all spectral objects such as dielectric functions and computed spectra in one step. SCOUT reacts as if you had executed the **Range** command in the main window.

The new spectral range is defined by the mininum value min, the maximum value max, the number of data points no_points and the spectral unit. The spectral unit must be specified as one of the following strings:

String transmitted		Spectral unit used by SCOUT
'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
' m m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

16.3.45 set_spectrum_simulation_range(n : integer; min, mix : real number; no_points : integer; spectral_unit : string)

Use this command to change the range of one of the simulated spectra. SCOUT reacts as if you had executed the **Range** command in the window of the n-th spectrum. Spectra are counted starting with 1, i.e. 1, 2, 3, ...

The new spectral range is defined by the minimum value min, the maximum value max, the number of data points no_points and the spectral unit. The spectral unit must be specified as one of the following strings:

String transmitted		Spectral unit used by SCOUT
'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
' m m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

16.3.46 set_spectrum_x_axis(i : integer; min, max, tick_spacing : float)

Use this command to set the range of the x-axis of the i-th spectrum (counted 1,2,3,...): min will be the minimum, max the maximum, and tick_spacing will be used as interval between the 'ticks'.

16.3.47 set_spectrum_y_axis(i : integer; min, max, tick_spacing : float)

Use this command to set the range of the y-axis of the i-th spectrum (counted 1,2,3,...): min will be the minimum, max the maximum, and tick_spacing will be used as interval between the 'ticks'.

16.3.48 set_top_halfspace_material (i : integer; name : string)

Assigns the material called name to the top halfspace of the i-th layer stack. Stacks are counted 1, 2, 3, ...

A material called name must be in the list of materials. If it is not you can use the command [load_material_from_database\(name\)](#) in order to load the material from the database into the list of materials.

16.3.49 show

Displays the SCOUT window.

16.3.50 show_statusbar

Shows the status bar.

16.3.51 show_maximized

Shows the main window maximized on the screen.

16.3.52 show_minimized

Minimizes the main window on the screen.

16.3.53 show_normal

Shows the window in its normal position and size, i.e. not maximized and not minimized.

16.3.54 smooth_data

Smoothing of the experimental data using 3-point-averaging.

Parameters:

n_spec: number of the spectrum in the spectrum list
wn_min: wavenumber minimum of the spectral range
wn_max: wavenumber maximum of the spectral range
nt: number of smoothing iterations

16.3.55 smooth_data_like_batch_control

Smooth the experimental data using the settings of the batch control window for each spectrum.

16.3.56 spectrum_to_clipboard (n:integer)

Tells the n-th spectrum object to copy its spectrum to the clipboard.

16.3.57 start

Starts the parameter fit using the current configuration.

16.3.58 stop

Stops any fitting activity.

16.3.59 update_data

Forces the re-computation of the SCOUT_98 configuration. This command should be used after parameter changes to calculate updated spectra and fit deviation.

16.3.60 update_plot

Forces the re-drawing of the SCOUT_98 configuration.

16.3.61 view_to_clipboard_wmf

Sends the current view as metafile (vector graphics) to the clipboard.

16.3.62 view_to_clipboard_bmp

Sends the current view as bitmap to the clipboard.

16.4 Controlling SCOUT from Excel

16.4.1 Introduction

This example shows how SCOUT work can be done using Visual Basic for Applications (VBA) and Excel tables. The required tables, VBA programs and menu entries are stored in the Excel file Scout.xls .

The following ingredients are needed and described:

- [A set of pre-defined tables](#)
- [A set of VBA procedures](#) (also called 'macros' in this document)
- [Excel menu items](#) to execute the VBA routines

16.4.2 Table structure

The Excel files prepared for SCOUT control contain the following sheets:

Fit strategy

This table stores information on the SCOUT_98 configuration file which contains the optical modelling for the particular problem. The SCOUT configuration covers the optical constants of all materials, the layer structure, the spectra that are computed and fitted and the list of fit parameters. A convenient way to enter the configuration filename is to select the cell and the activate the menu item 'Get arbitrary filename' from the SCOUT menu in Excel.

In addition a list of fit parameter sets is specified which defines the 'fit strategy' followed during the automatic batch processing of many input spectra (see below). The names of the fit parameter set files can be entered the same way as the configuration file.

A simple example of a 'Fit strategy' table (only one fit parameter set is used in the batch fit process) is given below:

Method file:	G:\theiss\solutions\sog\sing_am4.s98			
Fit parameter sets:			Time	Deviation threshold
Test	G:\theiss\solutions\sog\sing_am4.fp		00:00:30	1.00E-06

Pre-fit parameters

The success and the speed of an automatic fit depends strongly on the starting values of the

parameters. In cases where there are large differences between successive spectra it may be helpful to do a comparison to a limited set of typical parameter sets before the actual fitting work starts. The most successful set of parameters can then be used as the starting set for the automatic fit.

On the 'Pre-fit parameter' sheet you can define an arbitrary number of parameter sets. The corresponding batch fit macro looks up the entries in the row labeled as 'description' starting in the column named 'Set 1'. The parameters specified below 'Set 1' are tried first. The fit deviation is entered in the row named 'deviation'. If there is an entry to the right of the cell 'Set 1' (as is the case here since we have the 'Set 2' label) the next set is tested. This is continued until the first empty description cell is found. Of course, the best fit parameter set is loaded as starting set in the subsequent optimization procedure.

For the pre-fit a certain fit parameter set is used which must be specified in the second row of the sheet. This fit parameter set must match the selection of fit parameters in the starting SCOUT configuration. Usually this set contains all possible fit parameters. When the SCOUT configuration is loaded the fit parameter names are entered into the 'pre-fit parameters' sheet.

Pre-fit parameters				
Parameter set:	G:\theiss\solutions\sog\sing_am4.fp			
Description		Set 1	Set 2	Set 3
	Deviation			
Parameter no.	Parameter name			
1	SiNx:UV: oscillator strength	85442.2813	68551.2188	58751.8203
2	SiNx:UV: damping	10.7583	10.0413	27.6009
3	SiNx:Interband: resonance frequency	21808.9199	23282.6641	22829.4316
4	SiNx:Interband: oscillator strength	19477.9766	17552.2695	13149.2422
5	SiNx:Interband: distribution width	3775.1897	3892.2424	3996.9253
6	SiNx:Noname: real part	0.3115	1.2370	1.5319
7	Layer no. 1 SiNx: Layer thickness	0.3973	0.4324	0.6232

Results

This sheet is used for the batch processing of a series of input spectra. An example is given below:

Results				
Pathname:	G:\theiss\solutions\am4_file\	Format:	5	
Parameter		9.0000	12.0000	15.0000
Deviation				
Files	Transmittance	sso1142.sp	sso1143.sp	sso1144.sp
	Reflectance	sso1156.sp	sso1157.sp	sso1158.sp
1	SiNx:UV: oscillator strength	85442.2813	68551.2188	58751.8203
2	SiNx:UV: damping	10.7583	10.0413	27.6009
3	SiNx:Interband: resonance frequency	21808.9199	23282.6641	22829.4316
4	SiNx:Interband: oscillator strength	19477.9766	17552.2695	13149.2422
5	SiNx:Interband: distribution width	3775.1897	3892.2424	3996.9253
6	SiNx:Noname: real part	0.3115	1.2370	1.5319
7	Layer no. 1 SiNx: Layer thickness	0.3973	0.4324	0.6232
	Refr. index at 1000 nm	2.9617	2.5921	2.3339
	Refr. index at 2000 nm	2.8282	2.4980	2.2606

When the SCOUT configuration is loaded several entries are made in the 'Results' sheet which help you to fill in the spectrum filenames. The names of the simulated spectra are put to the right of the cell 'Files'. In addition the fit parameter names are listed below the spectrum names. During the batch process the resulting fit parameter values for the individual samples will be filled into the table. In the given example two spectra (transmittance and reflectance) are computed in SCOUT. Hence

you must specify the corresponding number of spectrum filenames for the batch mode. All spectra must be in one and the same directory which is given in the cell to the right of 'Pathname'. To enter a pathname you can select the cell and activate the menu item 'Get spectrum path'. In the following file import dialog you have to select any file in the desired directory and leave the dialog by 'Ok'. You will observe that the filename is skipped and only the pathname is filled in the cell.

In addition you must also specify the file format to be used by SCOUT for import actions. The cell to the right of 'Format' contains a number which specifies the format according to the following assignments:

Format	File format
1	Standard
2	xy-format
3	JCAMP
4	SpectraCalc
5	Perkin-Elmer
6	Zeiss
7	Opus

Finally you have to specify all the filenames that are to be 'fitted'. In the shown example the first set of spectra is the transmission spectrum sso1142.sp in combination with the reflectance file sso1156.sp. Both are in the directory g:\theiss\solutions\am4_file\ and are stored in Perkin-Elmer format. To enter files conveniently you can use the 'Get spectrum filename' menu command and select the wanted file in the subsequent dialog. The path is subtracted and only the pure filename is entered in the cell that is presently selected.

SCOUT status

The SCOUT status sheet is used to check the present SCOUT configuration. Activating the menu command 'Show status' fills in the present list of dielectric functions and the list of the computed spectra:

Status of the present SCOUT configuration	
Dielectric functions	Spectra
Vacuum	Transmittance
SiNx	Reflectance
Glass KKR	

VBA macros

This sheet contains the VBA macros which are explained in detail below.

Of course you are invited to program your own VBA macros to customize Excel to your needs. The SCOUT OLE automation commands and properties that you can use are explained in a separate document.

16.4.3 SCOUT menu items in Excel

The Excel files add to Excel a menu called 'SCOUT' which has the following subitems:

Create SCOUT

Use this command to start SCOUT as OLE automation server running invisibly in the back.

Immediately after startup SCOUT loads the configuration specified on the 'fit strategy' page.

Start fit

Starts the SCOUT fit activity as if you had pressed the Start button in SCOUT.

Stop fit

Stops the current fit activity of SCOUT as if you had pressed the Stop button in SCOUT.

Delete SCOUT

This removes SCOUT from memory. SCOUT is automatically shut down if you quit Excel.

Start batch fit

Starts the batch fit macro which processes all spectra in the input list (located on the 'Results' page).

No pre-fit is done which is recommendable if the input spectra do not differ from each other very much.

Start batch fit using pre-fit

This command starts the batch fit macro with pre-fit activity. This is recommended if the input spectra may differ significantly from each other.

Show status

With this command you can create entries on the SCOUT status page giving information on the SCOUT model. A list with the names of the used dielectric functions is given. The names of the computed spectra are also shown. This information is useful if you want to write macros which have to refer to certain dielectric functions (for example, to extract refractive indices or absorption coefficients).

Enter fit parameters at cursor

A (vertical) list of the current values of the SCOUT fit parameters is created starting at the current cursor position in Excel. This command can be used to manually create collections of fit results. This command can also be useful for the creation of new pre-fit parameter sets.

Restore fit results from cursor column

After a batch fit you may want to inspect visually the fitting result for a certain sample. To do this, simply place the cursor on the 'Results' page in the column that contains the fit parameters obtained for that sample and activate the 'Restore ...' command. ...

Get arbitrary filename

This command is used to enter a filename in the active cell. It activates a standard file import dialog. If the dialog does not appear on the screen it might be hidden by some other windows. You can use the Alt-Tab key combination to bring it to the front. The filename selected in the dialog is entered in the cell after the dialog is closed.

Get spectrum filename

This command is used to select a file that holds a spectrum for the batch process. It is useful only on the 'Results' sheet. After you selected a filename Excel subtracts the path specified in the cell 'Pathname' and returns the pure filename only. During the batch mode this filename is composed with the path given set in the 'Pathname' cell to result in the complete filename for data import.

Get spectrum path

This command is used to fill in a pathname in the cell 'Pathname'. Select this cell and activate the 'Get spectrum path' menu item. In the following file import dialog pick any file in the directory that holds your batch process input spectra. After the dialog Excel subtracts the pure filename and returns to the 'Pathname' cell only the path. During the batch mode this path is used to find the experimental spectra.

16.4.4 Excel macros

16.4.4.1 Overview

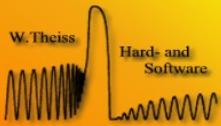
The following macros are explained in this section:

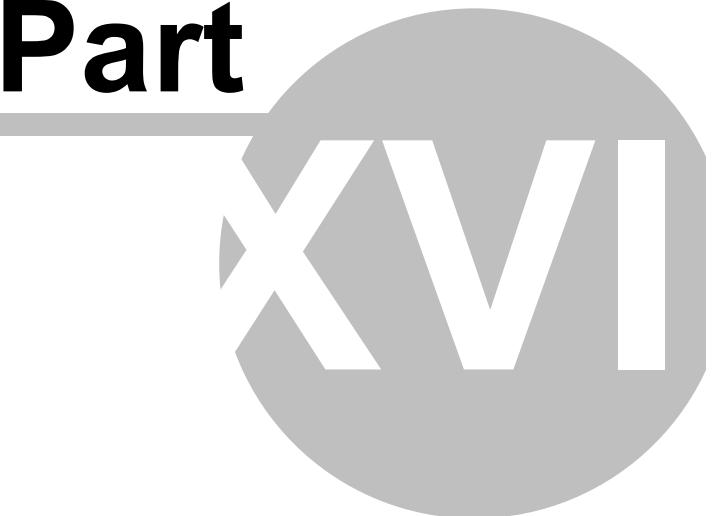
... Coming sooner or later ...

16.4.5 Application example

...coming sooner or later ...

SCOUT



Part  **XVI**

17 References

17.1 Articles related to optical spectroscopy

[Brendel 1990A]

R.Brendel, Appl. Phys. **A50**, 587 (1990)

[Bergman 1978 A]

D.Bergman, Phys. Rep. **C 43** (1978), 377

[Bruggeman 1935A]

D.A.G. Bruggeman, Ann. Phys. **24** (1935), 636

[Campi 1988A]

D.Campi, C. Coriasso: 'Prediction of optical properties of amorphous tetrahedrally bonded materials', J. Appl. Phys. **64** (8) (1988), 4128-4134

[Eickhoff 1992A]

T.Eickhoff, P.Grosse, S.Henkel, W.Theiß: 'Determination of the effective dielectric function of silver powders produced by inert gas evaporation', Z.Phys.B. - Condensed Matter **88** (1992), 17-23

[Evenschor 1990A]

M.Evenschor, P.Grosse, W.Theiß, 'Optics of two-phase composites', Vibrational Spectroscopy **1** (1990), 173-177

[Gervais 1974]

F.Gervais, B.Pirou, J.Phys.C: Solid State Phys., Vol. 7 (1974), 2374

[Grosse 1991A]

P.Grosse, 'Characterization of Thin Solid Films and Surfaces by Infrared Spectroscopy', in Festkörperprobleme **31**, ed. by R. Helbig (Vieweg, Braunschweig, Wiesbaden 1991), 77-97

[Harbecke 1986A]

B.Harbecke, 'Coherent and Incoherent Reflection and Transmission of Multilayer Structures', Appl. Phys. B **39** (1986), 165-170

[Jellison 1998A]

G.E.Jellison, Jr., 'Spectroscopic ellipsometry data analysis: measured versus calculated quantities', Thin Solid Films **313-314** (1998), 33-39

[Kim 1992]

C.C. Kim, J.W. Garland, H. Abad, P.M. Raccah, Phys. Rev. B **45** (20) 1992, 11749-11767

[Looyenga]

H.Looyenga, Physica **31** (1965), 401

[Maxwell Garnett 1904A]

J.C.Maxwell Garnett, Philos. Trans. R. Soc. London **203** (1904), 385

[O'Leary 1997A]

S.K.O'Leary, S.R.Johnson, P.K.Lim, J.Appl. Phys. Vol. **82**, No. 7 (1997), 3334-3340

[Park 1997]

J.H.Park, Y.J.Park, Y.M.Hwang , 'Design and fabrication of dichroic mirrors for color separation and recombination of Kr-Ar laser (white laser)', SPIE Vol. **3133** (1997), 288

[Parikh 1992A]

A.N.Parikh, D.L.Allara: 'Quantitative determination of molecular structure in multilayered thin films of biaxial and lower symmetry from photon spectroscopies. I.Reflection infrared vibrational spectroscopy', *J.Chem.Phys* **96** (2) (1992), 927-945

[Press 1992]

W.H.Press, S.A.Teukolsky, W.T.Vettering, B.P.Flannery, 'Numerical recipes in C', Cambridge University Press, New York, 1992; 408-412

[Sturm 1991A]

J.Sturm, P.Grosse, W.Theiß: 'Effective dielectric functions of samples obtained by evaporation of alkali halides', *Z.Phys.D. - Atoms, Molecules and Clusters* **20** (1991), 341-343

[Sturm 1991B]

J.Sturm, P.Grosse, W.Theiß: 'Effective dielectric functions of alkali halide composites and their spectral representation', *Z.Phys.B. - Condensed Matter* **83** (1991), 361-365

[Theiss 1994A]

W.Theiß, 'The Use of Effective Medium Theories in Optical Spectroscopy', in *Festkörperprobleme* 3 3, ed. by R. Helbig (Vieweg, Braunschweig, Wiesbaden 1994), 149-176

[Theiss 1994B]

W.Theiß, S.Henkel, M.Arntzen, "Connecting microscopic and macroscopic properties of porous media: choosing appropriate effective medium concepts", *Thin Solid Films* **255** (1995), 177-180

[Theiss 1995A]

W.Theiß, M.Wernke, V.Offermann, "Depth profiling of porous silicon layers by ATR spectroscopy", *Thin Solid Films* **255** (1995), 181-184

[Theiss 1995B]

W.Theiß, S.Henkel, M.Arntzen, "Connecting microscopic and macroscopic properties of porous media: choosing appropriate effective medium concepts", *Thin Solid Films* **255** (1995), 177-180

[Theiss 1997A]

W.Theiß, "Optical properties of porous silicon", *Surface Science Reports* **29**, No. 3/4 (1997), 91-192

[Valk 1998A]

J.Valk, S.van Enckevort, W.Theiß, "Infrared analysis of thin films deposited on rough and curved substrates", *AIP conference proceedings* **430** (1998), 647

Index

- 3 -

3D 223
3-point-averaging 295
3-point-smoothing 123

- 7 -

7 227

- A -

A 169
 a^* 169, 172, 271
abs 253
absolute minimum 152
absolute reflectance 184
Absorbance 118, 122
absorbed fraction 129
absorption 67, 131, 133, 223, 227
absorption coefficient 44, 45, 168, 268
Absorption distributio 133
access violation errors 292
acos 253
acquire 180, 213, 283
Acquire bu 184
Acquire button 213, 219
Actions 122, 213, 257
ADC type 184
ADC1000USB 184
Add multiple files 160
Add points 74
Add symbol to view 197
add_layer_definition_on_top 283
add_layer_on_top 284
add_to_report 285
algorithm 158
Alt key 191
amorphous film 64
amorphous oxide 68
amplitude 223
analysis 33
analysis method 219
angle 14, 118, 223, 257

angle of incidence 118, 126, 129, 133, 257, 276
anisotropic layer 96, 101
anisotropic layers 101
anti-reflection coating 227, 231
ASCII files 254, 255, 256
asin 253
assignment 180
assignment of dielectric functions 72
asssign materials 101
atan 253
atomic cores 43
ATR 118, 122
ATR element 118
attenuated total reflection 118
author 8
automatic fit 291
automatic fit routine 14
Automatic fitting 118, 158
Automatic re-computation 223
Automatic scaling 36
Automatic spectrum display 219
Automatic view update 44
automation objects 267
autoscale button 23
Avantes 188
average 123, 169
average spectra 124
average thickness 82
average time 268
averaging 184

- B -

b^* 169, 172, 271
background 193, 194, 214, 250
background picture 11
background spectrum 184
backgrounds 197
band gap 64
bar graph 33
bars 211
batch control 33
batch control window 38, 160, 164
Batch fit 158, 159, 165
batch fit macro 298
batch fit preparation 160
batch mode 298
batch operation 160
batch processing 296

Batch results 165
 Begin superlattice 89, 283
 Bergman 72
 Bergman representation 74
 bitmap view 196, 214
 bitmaps 196
 blue shift 141
 bonding 54
 bottom 293
 boundary 191, 275
 Brendel oscillator 58
 Brendel oscillators 57
 Bruggeman 72, 74, 90

- C -

C 169
 Campi 66
 Campi-Coriasso model 66
 Caps key 191
 Carrier concentration 54, 68
 cell 205, 234
 charge 54
 charge carrier 129
 Charge carrier generation 129
 charge carriers 43, 54
 charged impurities 55
 charts 223, 230, 231
 Check 77
 Check error 183
 classification 159

Clear layer stack 77
 clear_fit_parameters 286
 clear_layer_stack 286
 clear_material_list 286
 clear_report 286
 click 250
 clipboard 33, 197, 213, 292
 clipboard transactions 267
 coating 97
 CODE soft 176
 coherent 80, 283
 coherent superposition 3, 82, 284
 color 172, 194, 199, 209, 250, 271
 color box 172
 color coordinate 103, 169, 271
 color coordinates 172
 color picking dialog 194, 195
 color selection 11

ColorBox 103, 174
 column 201
 COME 231
 Communication 183
 complex refractive index 2
 composites 74
 composition 90
 computer simulation 3
 concentration gradient 90
 conduction band 64
 configuration 11, 184, 196, 214, 216, 219, 292
 configuration file 271, 296
 configurations 3
 Connect 247, 264
 Connection parameters 182
 connectivity 74
 constant dielectric function 53
 Constant refractive index 54
 constant susceptibility 53
 Continuous update 74
 conversion 129, 264
 Copy to clipboard 213
 Coriasso 66

cos 253
 cos(Delta) 126
 cosh 253
 cot 253
 coupling 157
 Creeate SCOUT 298
 critical points 53
 crossover frequency 55
 CSV 256
 Ctrl key 191
 Ctrl-key 14, 23
 cursor position 298

- D -

D65 169
 Damping 54, 66, 67
 damping constan 58
 damping constant 54, 55, 57, 58, 64, 157
 Dark 188
 dark measurement 184
 data acquisition 180, 181
 data exchange 254
 Data Factory 122
 data files 259
 data format 254

- data points 294
 data transfer 256
 database 3, 14, 46, 76, 77, 259, 264, 284, 291
 database file 247
 DDE 267
 default database 247
 Definition 129
 definition points 74
 Delete 250
 delete all 11
 Delete all points 74
 Delete points 74
 Delete SCOUT 298
 delta 126, 257, 276
 density 54
 density of states 64
 Deposition rates 230
 depth profile 133
 Design 103
 deviation 152, 158
 Deviation threshold 158
 deviation view 23
 DF_DB 264
 dialog 250
 dichroic mirror 227, 230
 dielectric background 53, 67
 dielectric function 2, 43, 44, 45, 46, 52, 53, 58, 61, 64, 96, 152, 200, 264, 268, 272, 278, 288, 289, 291, 293
 dielectric function assignment 96
 dielectric function database 76, 247
 dielectric function model 68, 199
 dielectric functions 101
 Direct editing 250
 direct evaluation 46
 directory 160
 disordered systems 57
 display 118, 197, 199, 209, 211
 display text 195
 distribution 227
 distributions 133, 223, 227
 doped semiconductors 54
 double click 207, 250
 double spectrometer 188
 Downhill simplex 152, 158, 274
 drag 191, 247
 drag&drop 14, 23, 72, 76, 77, 96, 101, 118, 122, 124, 152, 180, 199, 200, 223, 247
 drawing area 3
 drawings 197
 dropdown box 11
 dropdown list 256
 Drude 68
 Drude model 54, 55, 157
 dynamic data exchange 267
- E -**
- Edit 191, 250
 Edit graphics parameters 36
 edit mode 250
 effective dielectric function 73, 90
 effective dielectric functions 72, 74
 effective mass 54
 effective medium 72, 74
 efficiency 129
 electric field 43, 135, 223
 Electric field distribution 223
 electric susceptibility 43
 electrical current 129
 electronic interband transitions 53
 electronic system 43
 electronic transition 131
 ellipsometry 46, 126, 257, 275, 276
 EMA 74
 embedded particles 72
 emf 197
 Emission 135
 emission peak 140
 emitted radiation 131
 End superlattice 89, 283
 energy loss function 46
 epi-layer 33
 epitaxial silicon layer 160
 equidistant x-values 254
 equidistant x-values 255
 error message 264
 etching monitor 231
 etching process 231
 etching rate 231
 Excel 33, 165, 234, 241, 296
 Excel files 296
 Excel menu 298
 excitation 135
 Exit 250
 exp 253
 experimental data 123, 277
 experimental spectra 118, 219, 291, 298
 Export 33, 197, 214, 259, 264

- Export button 219
 export format 259
 Export multidimensional grid fit data 152
 Extended Drude model 55
 Extended Windows Metafile 197
 Extension 160
 external efficiency 140
 external OLE servers 180
 extrapolation 46
- F -**
- F4 152, 157
 F5 152
 F7 23
 Factor 152
 Fast Fourier Transforms 62
 fft 62, 122, 273
 FFT thickness 169
 FFT thickness analysis 273
 fft_strength 273
 fft_thickness 273
 field view 200, 214
 file extension 160
 file format 160, 296
 file import 160, 181, 267
 File|Options|Automatic update 223
 'fill' a layer 101
 film thickness 2
 filter design 227
 finish 292
 first moment 74
 fit 10, 219, 295
 fit activity 158, 275
 fit algorithm 158
 fit deviation 152, 159, 165, 180, 273
 Fit on a grid 152, 158, 274
 fit parameter 60, 219, 274, 275, 278, 286, 298
 fit parameter list 152
 fit parameter set 274
 fit parameter sets 3, 157, 158
 fit parameters 3, 14, 63, 69, 72, 152, 157, 199, 214, 278, 286
 fit parameters at cursor 298
 fit quality 292
 Fit results 219
 Fit strategy 296
 fit_parameter_mode 274
 fit_parameter_value_max 275
- fit_parameter_value_min 275
 fitting 275
 Fixed data 46
 Flip layer stack 77
 fluctuations 103, 172
 fluorescence 131
 font 199
 format 257
 formats 241
 formula 60, 122, 135, 152, 234, 253, 274
 formula editor 197
 formulas 69
 found_filename 275
 foundfiles 275
 Fourier transform 122
 FP --> SC2 165
 FP+SP --> SC2 165
 frac 253
 fraction of s-polarization 122
 frame mode 36
 Free carriers 53
 Free formula 63, 129
 frequency dependence 55
 Frequency dependent expressions 60
 Fresnels' equations 135
 fringes 169
 front end 267
 Frozen 152, 274
 Full screen 216
 Function 63, 129, 253
 Function fluctuation 176
- G -**
- g(n
 f) 74
 Gap 66, 67
 gap energy 64, 68, 131
 Gauss-distribution 57
 Gaussian 57, 58, 103
 Gauss-Lorentz-switch 58
 General features 250
 Gervais 58
 Get arbitrary filename 298
 Get spectrum filename 298
 Get spectrum path 298
 get_sim_val1 275
 get_sim_val2 276
 glass 11

glass plate 223, 227
 glass substrate 14
 Glass Type BK7 227
 global minimum 158
 Go 164
 Go! 38
 graph 196
 Graph selection 36
 graphics 126, 191
 Graphics course 36
 graphics parameters 14, 157, 201
 graphics setting 36
 graphs 36
 grid 291
 grid variation 152
 grid_test 291
 grid_test_max 276
 grid_test_min 276
 grid_test_no_points 276
 grid_test_para 276

- H -

halfspace 14, 77, 80, 97, 118, 207, 223, 286, 293, 294
 hardware 180, 184
 harmonic oscillator 54, 66
 header 256
 Height 11, 194, 195, 196, 199, 200, 277
 Help 250
 Hide 291
 High limit 152
 Hilbert transformation 62
 Horizontal bar 211
 host material 72
 Hotspot 207
 HunterLab 182, 256

- I -

ideal spectrum 118
 ight source distribution 133
 illuminant 271
 Illumination 169
 import 33, 181, 214, 219, 291
 import actions 296
 Import button 160, 219
 import data 288, 289
 import dialog 219, 298

Imported data 61
 imported dielectric function 288, 289
 impurities 55
 impurity concentrations 2
 incidence 118
 incidence angle 276
 incident radiation 133
 Included layer stack 97
 incoherent 80, 96, 283
 incoherent superposition 3, 82, 284
 infrared 53, 58, 141
 infrared region 54
 inhomogeneous materials 72
 inhomogeneous thickness 82
 Initializations 183
 Input format 38, 160
 Input option 160
 Input options 38, 160
 input spectra 38, 158, 296
 installation 267
 int 253
 integral 169
 integral quantities 176
 integration time 184, 188
 interband transition 54, 66, 67
 interband transitions 53, 61, 62, 64
 interface 80, 82, 85, 223
 interface roughness 85
 interference 131
 interference artefacts 96
 interference fringes 80, 82, 141
 interference pattern 33, 96, 122, 133, 169
 interference patterns 140
 interference structure 80
 internal efficiency 129, 135, 140
 interpolation 46, 61
 interval 152
 Invert layer stack 77

invisibly 298
 isotropic 135
 isotropy 74
 iteration 268

- J -

JCAMP format 256
 joint density of states 53, 64

- K -

kappa 276
key commands 10
Kim oscillator 58
KKR 62
KKR analysis 46
KKR dielectric function model 61
KKR susceptibility 61, 62, 63
Kramers-Kronig-Relation 62

- L -

L^* 169, 271
laterally structured samples 118
layer 82, 85, 129, 277, 283, 284
layer absorption 129
layer definitions 283
layer deposition 230
Layer list 124
Layer mix 118, 124
layer stack 14, 23, 80, 96, 118, 124, 129, 152, 223, 277, 283, 284, 286, 293, 294
layer stack definition 77, 97, 199
Layer stack view 207
Layer stacks 3, 77, 168, 279, 286
layer thickness 140, 169, 199, 214
layer_thickness 277
layers 3, 77, 101, 279, 286
left 11, 194, 195, 196, 199, 200, 277
level 159
light scattering 72
light source 135
light sources 133
limit 152
limited resolution 80
Limits from sliders 152
Line mode 118
linear interpolation 46, 61
list object 250
list of dielectric functions 200
List of fit parameter sets 158
list of fit parameters 152, 286
list of layer stacks 118, 223, 283, 284
list of materials 284, 286, 291
list of spectra 160, 200
list of view elements 207
list of views 193, 201

List properties 250
literature 46
In 253
LO modes 58
load configuration 271
load dielectric function 291
Load layer stack 77
load_material_from_database 291
local absorption 133, 223, 227
local minimum 158
log 253
long wavelength limit 72
lookup table 69
Loop through results 165
Looyenga 72, 74, 90
Lorentzian 57, 58
Low limit 152
lower boundary 275
luminescent layer 133, 135, 140
luminescent layers 141

- M -

m 275
main menu 38, 219
main view 3, 10, 11, 23
main view level 3
main window 3, 33, 191, 213, 277
Manipulate layer stack 77
Manipulation 122
manual parameter variation 23
manual thickness variation 23
master model 68, 69, 72
Master model objects 69
Master parameter 274, 286
Master parameter list 286
Master parameters 68, 69, 152, 157
material 101, 230, 264, 277, 291, 292, 293, 294
material identification 54
Materials 14, 286
maxima 169
Maxwell Garnett 72, 73, 90
measured data 14, 181
measured spectrum 169, 272
measurements 180, 219
menu 3, 10, 216
Menu button 219
menu commands 250
Menu items 296

Menu visible at start 219

message 293

metafile 212, 213

metafile format 197

metafile graphics 11

Metafile view 11, 197, 212

metal blacks 74

metal-insulator composites 74

Metallic layers 141

metals 54

method 214, 219, 267

Method button 219

microgeometry 72, 74

micron 284

microns 80

microscope slide 97

microscopic 131

microscopic quantities 2

microscopic vibrations 54

microstructures 72

microtopology 74

minimum 275

MiniScan 182

Minolta 183

mixing formulas 72

mm 284

model parameter 199, 214, 215

model parameters 2

monitoring 230, 231

monitoring charts 223

move down 250

Move points 74

Move up 250

Moving 197

Multidimensional grid 152

multiple reflection 80

multiple spectra files 256

- N -

Name 199

navigation bar 247

network 74

New 250

next configuration button 212

next view button 212

NIR512 184

nm 284

non-equidistant x-values 254

non-radiative energy loss 131

normalization 121, 122

numbers 234

- O -

object menu 3

object types 250

objects 3

observation angle 135, 169, 271

OceanOptics 184

OJL 68

OJL model 64

OLE 267

OLE automation 3, 157

OLE automation clients 267

OLE server 267, 292

Omnidriver 184

OOIBase32 184

Open 250

optical constants 3, 43, 44, 131, 264, 291, 293

optical function 279

Optical function fluctuation 174

optical functions 174, 176

Optical growth monitor 227

optical model 14, 33

optical models 3

optical monitoring 223, 227, 230, 231

optical phonons 54

optical spectroscopy 2, 267

optical thickness 122

optical transition 64

optimization 152

option 152, 250

options 247, 259

Opus format 256

oscillator 58

oscillator model 58

Oscillator strength 54, 66

overlap 193

oxygen content 68

- P -

Parabolic bands 64

parameter change 295

parameter fit 10, 152, 214

parameter fluctuation 3

parameter selection 23, 152, 209

parameter set 160
 Parameter slider 209, 211
 parameter space 152
 parameter value 211
 parameter variation 3
 parameter view 23, 199, 214
 Parikh 96
 partial wave 283
 partial waves 135
 partially reflected waves 82
 Password 219, 259
 Password protection 219
 Paste 197
 path 38
 pathname 296, 298
 peak 131, 135, 169
 Peak shift 141
 Pen 118
 percent 194
 percolation 74
 percolation strength 74
 periodicity 122
 Perkin-Elmer format 256
 phonons 54, 58
 Photocurrent 129
 photoluminescence 131
 pi 253
 pictures 11, 197, 212
 PL 131
 PL intensity 135
 PL spectra 140, 141
 PL spectrum 133
 plasma frequency 54, 68, 157
 points 74, 176
 polarization 43, 118, 122, 129, 133, 135, 223, 257
 poles 58
 porous silicon 133, 135, 140, 141
 portable spectrometer 182, 183
 position 207, 295
 power spectrum 122
 pre-defined layer stacks 77
 pre-fit 160, 298
 Pre-fit parameters 296
 Preparation 160
 Prepare 160
 prepare_shutdown 292
 presentation mode 10
 Presentations 216
 previous view button 212

print 250, 292
 printer setup 250
 procedures 296
 product 58
 production 103
 production control 118
 Psi 126, 257, 275
- Q -
 quality 159
 quantum mechanics 62
 quasistatic approximation 72
- R -
 range 129, 294
 Range command 14
 range of cells 33
 rates 230
 rating 280
 rating level 159
 re-absorption 131
 re-computation 295
 rect 253
 rectangle 23, 191, 194, 200
 rectangles 11
 re-drawing 296
 re-emission 135
 Reference 122, 188
 reference spectrum 121
 reflectance 11, 14, 118, 122, 124, 131, 183, 188, 227, 257, 291
 reflectance spectrum 14, 23
 refractive index 2, 44, 118, 122, 169, 273, 276, 280
 register 267
 rename_material 292
 report 250
 report window 286, 292
 report_to_clipboard 292
 report_to_file 292
 Reset 180
 re-size 207
 re-sizing 197
 resolution 61, 80
 resonance frequencies 2, 54, 57, 58
 resonance frequency 66, 67
 Restore fit results 298
 Restore graphics settings 36

Results 33, 38, 160, 296

retardation effects 72

right mouse button 250

rnd 253

Rough interface 85, 97, 283

roughness 85

routine analysis 219

Routine work 33

row 201

- S -

Sample 33, 188

sample description 33

sample names 38

sample view 23

Save 250

Save As 250

save configuration 292

save dielectric function 293

Save layer stack 77

SC2 --> WB 165

Scan directory 160

scans 188

scattering 55, 85

scattering losses 85

SCOUT 3

SCOUT applications 2

SCOUT control 296

SCOUT model 160, 298

SCOUT objects 250

screen 277, 295

searchfiles 293

select a filename 268

selected graph 36

selection 201

selections of fit parameters 157

semiconductor 53

semiconductors 54

semi-infinite 80

sequence 216

sequence of fit parameter sets 158

serial number 184

serial port 182, 183

server activity 267

set_bottom_halfspace_material 293

set_spectrum_simulation_range 294

set_top_halfspace_material 294

shape 131

sheets 296

shift 135

Shift key 23

show 294

Show presentation 216

Show previous configuration 213

Show status 298

show_maximized 295

show_minimized 295

show_normal 295

shut down 298

si 253

signals 188

silicon 135

silicon substrate 133, 140

silicon wafer 97

silver 14, 141

silver layer 11

Simple layer 80, 283

simulated quantities 275, 276

simulated spectra 14

simulated spectrum 169, 180, 281

simulation 2

sin 253

sinh 253

SiO₂ 227

SiO₂ layer 141

size 207, 295

sketches 197

slave parameter 72, 157

slave parameters 68, 69

slider 23, 33, 209

slider motion 36

slider position 36

slider range 23

Sliders 3, 152

Smoothing 123, 295

smoothing iterations 295

solar cell 129

Source 196, 199

SP --> SC2 165

spatial distribution 223

spectra 3, 118, 200, 279

Spectra list 286

SpectraCalc format 256

spectral density 74

spectral position 169

Spectral range 14, 123, 135, 157, 223, 252

spectral resolution 61

spectral unit 294
spectrometer 184, 188, 219
spectrometer assignment 180
spectrometer control 267
spectrometer list 180
spectrometer types 180
spectrometers 180, 188, 213, 283
spectroscopic ellipsometry 46
spectrum 122, 257, 271, 275, 276, 282, 291, 292, 294
spectrum filename 296
spectrum fit 214
spectrum list 152
Spectrum normalization 121
spectrum simulation 3, 267
spectrum simulations 118
spectrum transfer 181
speed bar 250
speed button 250
spreadsheet 234
sqr 253
sqrt 253
stack 277
stacks 279
stand-alone solution 267
standard format 254
standard format' 14
Start 14, 158, 214, 219, 295
Start batch fit 298
Start batch fit using pre-fit 298
Start button 33
Start fit 298
Start/Stop button 219
starting set 296
starting values 291, 296
startup 298
static dielectric constant 62
status bar 3, 158, 216
Statusbar 10, 33, 36, 259
step 253
Stop 14, 158, 219, 295
Stop fit 298
strategy 227
Strength 64, 67
structured samples 118
sublayers 90, 129
substrate 227
substrates 97
superlattice 89, 227, 283
superposition 283
superposition of partial waves 80
surface area 124
susceptibility 3, 43, 52, 60, 66, 67, 72, 152
susceptibility types 52
symbol 197

- T -

tab character 283
table 219
tables 296
tail states 64
tan 253
tan(Psi) 126
tanh 253
Target spectrum 118
Tauc-Lorentz model 67
tec5 188
text 199, 234
text file 33, 241, 292
text string 195
text view 11
the speed 216
Thick layer 82, 283, 284
'Thick layer' 14
thickness 14, 80, 85, 90, 169, 172, 277, 284
thickness analysis 122
Thickness averaging 85
thickness determination 122
thickness distribution function 82
thickness fluctuations 103, 172
thickness uncertainties 103
thickness unit 284
thickness variation 82, 283
thickness 'variation' 82
Thin film 82, 283, 284
'Thin film' 14
thin film system 131
thin films 140
threshold 159
time evolution 10
Time threshold 158
TiO₂ 227
TO modes 58
Tolerance threshold 158
TO-modes 54
tool bar 212
toolbar 3, 10, 23, 158, 160, 180, 219

top 11, 194, 195, 199, 200, 294
 top halfspace 118
 topology 74
 transition region 55
 transmittance 118, 122, 124, 188, 257, 291
 transparent substrate 96
 transverse optical phonons 54
 tree view level 3
 treeview 3, 10, 11, 14, 23
 trigonometric 257
 tutorial 33
 two-phase composite 72
 two-phase composites 74
 typical parameter sets 296

- U -

UltraScan 256
 ultraviolet 53
 unexperienced users 219
 uniaxial layer 96
 unit 284
 Update 250
 upper boundary 275
 USB2 188
 USB2000 184
 USB4000 184
 user interface 23
 user-defined formula 68, 69, 90, 122, 129
 user-defined function 122
 user-defined parameters 68
 user-interface 14, 33
 Username 259
 UV 53
 UV spectroscopy 62

- V -

Vacuum 286
 valence band 64
 Value 152
 Variable thickness 283
 variant 291
 variant arrays 291
 Variation 152
 variation mode 274
 VBA 296
 VBA macros 296
 vector graphics 197

Vertical bar 23, 33, 211
 vibrational modes 57, 58
 vibrations 54
 view 193, 194, 195, 196, 199, 200, 201, 209, 211, 213, 215, 219
 view button 23
 view definition 199, 200
 view design 193
 view element 11
 moving 191
 re-sizing 191
 view elements 23, 191
 View example 214
 views 11, 44, 191, 193, 216
 visible 53
 Visual Basic 267
 Visual Basic for Applications 296
 visual comparison 118
 Visual parameter variation 152
 VisualBasic 283
 volume fraction 72, 73, 74, 90

- W -

wave 223
 wave propagation 77, 80, 96
 wavelength 133
 wavenumber 268, 276, 280
 WAVSE32 257
 weight 74, 124
 Width 11, 57, 194, 195, 196, 199, 200, 277
 window 295
 Wolfgang Theiss 8
 Word 165
 workbook 33, 205, 214, 219, 234
 Workbook button 219
 workbook data 201
 Workbook display (columns) 201
 Workbook display (rows) 205
 worksheet 201
 worksheets 234

- X -

X 169, 271
 x-axis 294
 x-y-data 255
 xy-format 255

- Y -

Y 169, 271

- Z -

Z 169, 271

Zeiss format 256

z-range 223

ZrO₂ 227