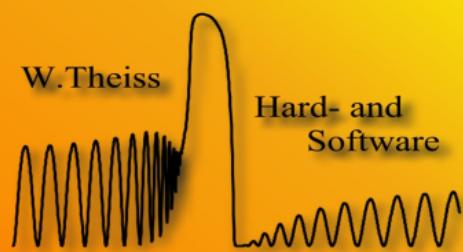
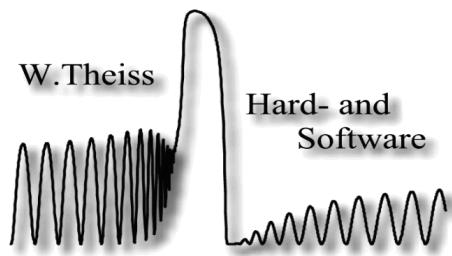


# CODE



## 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](mailto:theiss@mtheiss.com)      Web: [www.mtheiss.com](http://www.mtheiss.com)**



# CODE

## **Optical Spectrum Simulation for Coating Design**

---

*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: 11/28/2019, 12:24 AM in Aachen, Germany

# Table of Contents

<b>Foreword</b>	<b>0</b>
<b>Part I Overview</b>	<b>1</b>
1 About this document .....	2
2 Introduction .....	2
3 About the author .....	3
<b>Part II User guide</b>	<b>4</b>
1 Introduction .....	5
2 Integral quantities .....	5
The list of integral quantities .....	5
Color coordinates .....	8
Virtual office tower .....	13
Other quantities .....	15
Functions of integrals quantities .....	16
3 Integral quantities in main window views .....	17
4 Other differences to SCOUT .....	21
5 Presentations .....	22
Demo presentation .....	22
Generating presentations .....	23
Navigation .....	24
Publish .....	25
6 OLE automation .....	26
Introduction .....	26
Methods .....	26
acquire .....	26
add_layer_definition_on_top(stack_index : integer;definition : string).....	26
add_layer_on_top (stack_index : integer;	
layer_type,material:string;thickness:real number; thickness_unit : string).....	28
add_to_report(text : string).....	28
acquire_dark.....	28
acquire_reference.....	28
apply_factor_in_fit_parameter_transfers .....	29
automatic_spectrum_scaling(i : index).....	29
batch_add_sample (name: string).....	29
batch_clear_samples .....	29
batch_go .....	29
batch_restore_fit_results (sample : string).....	29
clear_fit_parameters .....	29
clear_layer_stack (index : integer).....	29
clear_material_list.....	30
clear_report.....	30
create_fit_parameter (list, object, subobject, subsubobject, parameter :	
integer) (copy).....	30
delete_fit_parameter (index : integer).....	32
df_load_from_file.....	32
df_load_from_file_2(n: Integer; const filename, unit_string:String);.....	32
df_load_from_x_n_k_table(n:integer; const filename : string);.....	33
df_load_from_x_n_k_table_2 (n:integer; const filename : string);.....	33
df_to_clipboard (n : integer).....	33

divide_all_measured_plate_spectra_by_100.....	33
do_no_apply_factor_in_fit_parameter_transfers.....	33
export_measured_spectrum(n:integer; filename:string; format : integer).....	33
export_simulated_spectrum(n:integer; filename:string; format : integer)_2.....	34
grid_test .....	34
hide .....	34
hide_statusbar.....	34
import_measured_data_via_variants(index : string; spectral_positions, spectral_data : variant; unit_string : string).....	34
function import_measured_plate_spectra(stack_index, layer_index : integer; t, rt, rb : string; file_format, option_t, option_rt, option_rb : integer) : integer .....	35
load_df(n : integer, filename : string).....	35
load_experiment (n:integer, filename: string, format:integer).....	35
load_material_from_database (name : string).....	35
pane_wizard_divide_r_by_100.....	36
pane_wizard_divide_t_by_100.....	36
pane_wizard_export_results (material : string; version : float; username, comment : string).....	36
pane_wizard_load_r (filename:string,file_format:integer,spectral_unit:string).....	36
pane_wizard_load_t (filename:string,file_format:integer,spectral_unit:string).....	36
pane_wizard_set_thickness (thickness : float).....	36
pane_wizard_spectral_range (min, max : float; points : integer;spectral_unit : string).....	36
rename_material (old_name, new_name : string).....	36
report_to_clipboard.....	36
report_to_file(filename:string).....	36
prepare_shutdown.....	37
print_fit(i : integer).....	37
save_configuration(filename : string).....	37
save_df(n : integer, filename : string).....	37
scout_normal.....	37
scout_on_top.....	37
searchfiles(dir,pattern : string).....	38
send_message(the_text : string).....	38
set_bottom_halfspace_material (i : integer; name : string).....	38
set_global_range(min, max : real number, no_points : integer, spectral_unit : string) ) .....	38
set_spectrum_simulation_range(n : integer; min, mix : real number; no_points : integer; spectral_unit : string).....	38
set_top_halfspace_material (i : integer; name : string).....	39
show .....	39
show_maximized.....	39
show_minimized.....	39
show_normal.....	39
show_statusbar.....	39
smooth_data.....	39
smooth_data_like_batch_control.....	39
spectrum_to_clipboard (n:integer).....	40
start .....	40
stop .....	40
update_data.....	40
update_plot.....	40
view_to_clipboard_wmf.....	40
view_to_clipboard_bmp.....	40
<b>Properties .....</b>	<b>40</b>
a_filename (string).....	40
absorption_coefficient .....	40

angle_distribution_no_points (spectrum_index:integer) (integer).....	40
angle_distribution_angle (spectrum_index, angle_index : integer) (float).....	41
angle_distribution_w eight (spectrum_index, angle_index : integer) (float).....	41
average_time (real number).....	42
batch_configuration_file (string).....	42
batch_fit_deviation (sample:string) (float).....	42
batch_fitparameter_name (index: integer) (string).....	42
batch_fitparameter_value (sample: string, index: integer) (float).....	42
batch_input_path(index : integer) (string).....	42
batch_name_of_spectrum (i : integer) (string).....	43
batch_number_of_spectra (integer).....	43
batch_prefit_result(sample : string) (string).....	43
batch_sample_parameter(sample : string) (float).....	43
batch_spectrum_filename (sample : string, index : integer) (string).....	43
batch_status (integer).....	43
color_coordinate(spectrum_index,simulated,color_type,illuminant,observati on_angle) (float).....	44
compute_emissivity (spectrum, index : integer) (integer).....	45
configuration_file.....	45
database .....	45
current_rating (string).....	45
delete_layer(stack_index,layer_index) (integer).....	45
df_real_part(n : integer; wavenumber : real number) (real number).....	45
df_real_part_nm.....	45
df_imag_part(n : integer; wavenumber : real number) (real number).....	46
df_imag_part_nm.....	46
dielectric_function_name(i:integer) (string).....	46
emissivity (spectrum, index : integer) (float).....	46
experimental_spectrum_comment(n:integer): string .....	46
experimental_spectrum_value.....	46
fft_thickness.....	47
fft_strength.....	47
file_format (integer).....	47
file_object_generation (real number).....	47
fit_deviation (real number).....	47
fit_parameter_name.....	47
fit_parameter_mode.....	48
fit_parameter_set.....	48
fit_parameter_value.....	48
fit_parameter_value_max.....	48
fit_parameter_value_min.....	48
fitting .....	48
foundfiles (integer).....	49
found_filename(i:integer) (string).....	49
get_sim_val1(n : integer; wavenumber : real number) (real number).....	49
get_sim_val2(n : integer; wavenumber : real number) (real number).....	49
grid_test_max (single).....	49
grid_test_min (single).....	49
grid_test_no_points (integer).....	49
grid_test_para (integer).....	49
import_measured_plate_spectra.....	50
import_glass_coating_spectra.....	50
incidence_angle.....	50
insert_layer_definition (stack_index, layer_index : integer, layer_definition : string) (integer).....	50
kappa(n : integer; wavenumber : real number) (real number).....	51
layer_material.....	51
layer_thickness .....	51
layer_type(n,m : integer) (string).....	51

main_w_inwindow_height.....	51
main_w_inwindow_left.....	51
main_w_inwindow_top.....	51
main_w_inwindow_width.....	51
material_is_in_database.....	51
material_is_in_material_list.....	51
measurement(i:integer) (string).....	51
multidimensional_gridfit (integer).....	52
number_of_dielectric_functions (integer).....	52
number_of_fit_parameters.....	52
number_of_fp_sets (integer).....	52
number_of_layers_in_stack (integer).....	52
number_of_layer_stacks (integer).....	52
number_of_optical_functions (integer).....	52
number_of_spectra (integer).....	53
number_of_tec_values (integer).....	53
object_generation (real number).....	53
optical_function(a_string : string) (real number).....	53
optical_function_by_index(i : integer) (real number).....	53
optical_function_text_by_index (index : integer) (string).....	53
parameter (i:integer) (real value).....	53
rating_threshold (level_index:integer) (float).....	53
rating_description (level_index:integer) (string).....	54
reference_spectrum(n:integer;formal:integer):string.....	54
refractive_index(n : integer; wavenumber : real number) (real number).....	54
sample_description: string.....	54
set_spectrum_y_axis(i : integer; min, max, tick_spacing : float).....	54
simulated_spectrum_value.....	54
simulated_spectrum_value_nm.....	54
simulation_exportfile(n:integer;formal:integer):string.....	55
spectrum_name(i:integer) (string).....	55
spectrum_weight(index) float.....	55
tec_value .....	55
tec_value_name .....	55
tec_value_nickname (tec_index : integer) (string).....	55
tec_value_optimize (tec_index : integer) (integer).....	56
tec_value_target_value (tec_index : integer) (integer).....	56
tec_value_weight (tec_index : integer) (float).....	56
tolerance (real number).....	56
w_weight_formula (spectrum_index, subindex : integer) (string).....	56
<b>Example .....</b>	<b>56</b>
<b>Using CODE in LabView .....</b>	<b>60</b>

## Part III Background 64

<b>1 Background information .....</b>	<b>65</b>
<b>2 Color coordinates XYZ .....</b>	<b>65</b>
<b>3 Color coordinates x,y,z .....</b>	<b>66</b>
<b>4 Color coordinates L*a*b* .....</b>	<b>67</b>
<b>5 Color coordinates Lab .....</b>	<b>67</b>
<b>6 Ra (color rendering index) .....</b>	<b>67</b>
<b>7 Light transmittance .....</b>	<b>68</b>
<b>8 Light reflectance .....</b>	<b>69</b>
<b>9 Solar direct transmittance .....</b>	<b>70</b>
<b>10 Solar direct reflectance .....</b>	<b>70</b>

11 Spectrum product .....	70
12 Photocurrent .....	71
13 g (total solar energy transmittance) .....	74
14 Emissivity .....	75
15 Sheet resistance .....	75
16 U (Thermal transmittance) .....	75
17 Sound insulation .....	76
18 NFRC support .....	78
NFRC VTc .....	78
NFRC 300-2004 solar average .....	78
Tdw-ISO .....	78
Tuv .....	78
NFRC U-value (winter) .....	78
NFRC U-value (summer) .....	79
NFRC SHGC .....	79
19 A spectrum .....	80
20 D65 spectrum .....	82
21 C spectrum .....	84
22 User-defined illuminants for color computation .....	86
23 Color-matching functions .....	91
24 Solar spectral distribution .....	94
25 AM 1.5 spectrum .....	95
<b>Index</b>	<b>96</b>

**CODE**

W. Theiss

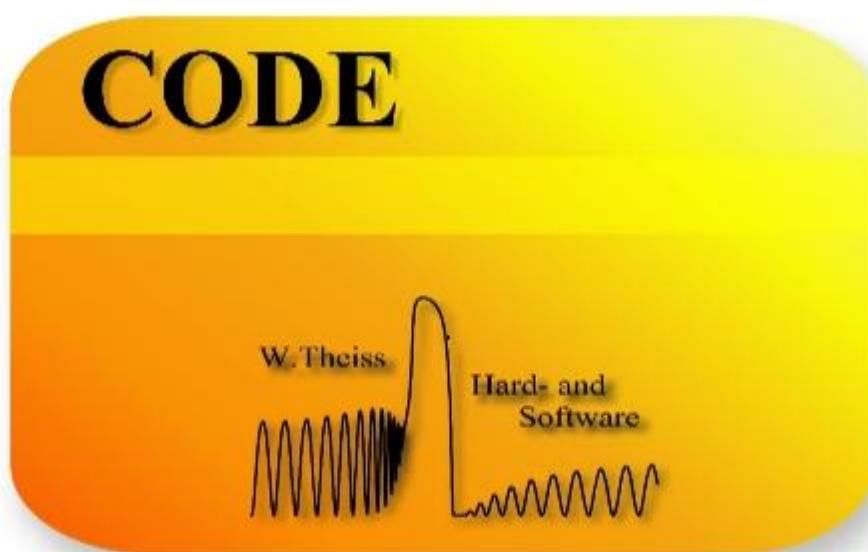
Hard- and  
Software

**Part**

I

## 1 Overview

### 1.1 About this document



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](mailto:theiss@mtheiss.com) web: [www.mtheiss.com](http://www.mtheiss.com)

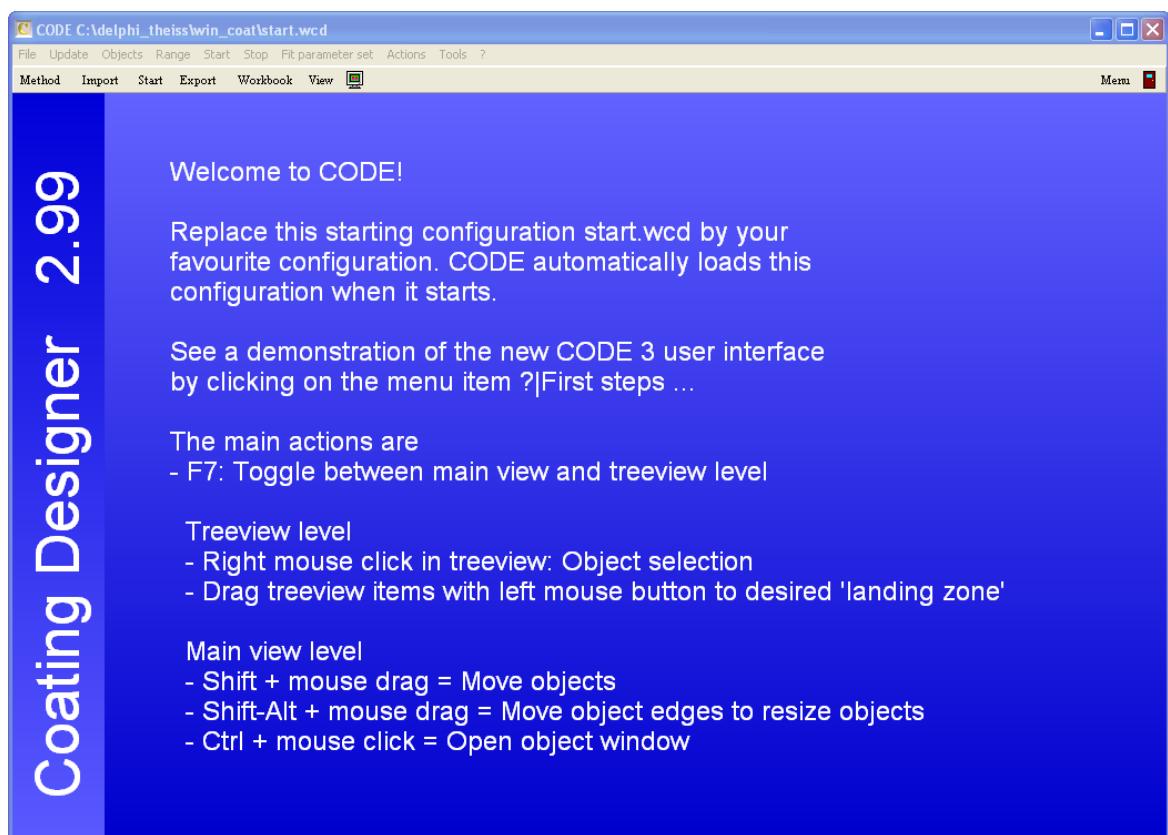
January 2012

### 1.2 Introduction

The Coating Designer (CODE) is an application to assist you in the optical design of thin film coatings. Originally designed for glass coatings it can be used quite generally whenever optical spectra have to be optimized for a certain application (such as optical filters, solar absorbers, ...). It is based on the SCOUT spectrum simulation software and adds some functions important for thin film design. In fact, Coating Designer contains a complete SCOUT. You can directly import SCOUT configurations by setting the import filter to the SCOUT file format.

This documentation is only about the **additional features** added to SCOUT. It is assumed that the reader is already an experienced user of SCOUT. Help on SCOUT is given in a separate document.

The CODE main window looks, except for the background picture, very much like that of SCOUT:



The introduction of the user guide gives an overview of the additional features of CODE with respect to SCOUT.

### 1.3 About the author

Responsible for all good and bad program features:

Wolfgang Theiss  
Dr.-Bernhard-Klein-Str. 110  
D-52078 Aachen  
Germany

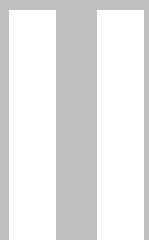
Phone: + 49 241 5661390  
Fax: + 49 241 9529100  
e-mail: theiss@mtheiss.com  
web: www.mtheiss.com

**CODE**

W. Theiss

Hard- and  
Software

**Part**



## 2 User guide

### 2.1 Introduction

This guide shows how the additional features of CODE are applied - background information on the computed quantities are described in a later chapter.

There is one additional list of objects, the list of so-called 'integral quantities'. These quantities characterize optical properties of layer stacks and are often used as the only characterizing numbers for products. Typical examples are color coordinates or averaged transmission coefficients in certain spectral ranges.

Integral quantities can be displayed in views (the main window of CODE). Details are described in a separate section.

Several minor differences are summarized in the section 'Other differences to SCOUT' below.

You can use CODE to generate presentations of your work. A CODE presentation is a sequence of CODE configurations with some mechanisms to navigate around. Details are described here.

In order to use OLE automation of both SCOUT and CODE on one machine, the OLE server name of the two programs is different. The programs also differ in the available OLE commands: CODE supports all SCOUT OLE automation features and adds some more. The additional commands and properties of the CODE OLE server are described in a separate section.

## 2.2 Integral quantities

### 2.2.1 The list of integral quantities

The list of integral quantities looks like this:

The screenshot shows a Windows-style dialog box titled "S2 Integral quantities". The menu bar includes "File", "New", "Edit", "Delete", "Update", "Tables", and "?". Below the menu is a toolbar with a "+" button, a dropdown menu, and several icons for sorting and filtering. The main area is a table with the following data:

	Value	Type	Spectrum	sim/exp	Optimize	Target value	Weight
1	93	Ra (Reflection)	R	Simulation	Off	0.0000	1.0000
2	94	Ra (Transmission)	T	Simulation	Off	0.0000	1.0000
3	7 %	Front pane absorption	R	Simulation	Off	0.0000	1.0000
4	0.7	Ug (EN 673)	R	Simulation	Off	0.0000	1.0000
5	0.54	g (EN 410)	R	Simulation	Off	0.0000	1.0000
6	0.51	g (DIN 67507)	R	Simulation	Off	0.0000	1.0000

Please see the technical reference of the SCOUT manual for a description of the general handling of lists like this.

As usual, you create items, i.e. rows, by selecting the desired item type in the drop down box (labeled in the situation above with 'Color coordinate L') and then press the '+' button.

The columns of the list have the following meanings:

#### Value

This is the actual, computed value of the integral quantity.

#### Type

Here the type of the quantity is indicated. There are several color coordinates that can be computed, and various other quantities of technical interest.

#### Spectrum

The name of the spectrum that is referred to when the integral quantity is computed. Certain types of integral quantities require spectral data in well-defined spectral ranges. The spectrum should cover the used spectral range completely. The usual interpolation and extrapolation is performed if the spectral data points in the spectrum do not match the points necessary for the computation of the integral quantity.

To select the spectrum you have to press the F4 function key. This lets you cycle through all entries of the spectrum list. Stop when you have reached the appropriate spectrum. By pressing the F5 function key you can move backwards in the spectrum list.

#### Sim/Exp

After the wanted spectrum is selected you can choose between the simulated spectrum and the experimental one for the computation of the integral quantity. Although usually you may be interested in the values for the simulated spectra only, there might be situations where you want to know the corresponding values for the experimental data also.

You toggle between the simulated and measured spectrum by pressing the F4 or F5 function key.

#### Optimize

This value can be switched between the states 'Off' and 'On'. If the setting is 'Off' nothing happens. If one of the entries is set to 'On' then WCD switches its fitting strategy: Whereas usually the goal is to decrease the deviation between simulated and measured data now the integral quantity is optimized with respect to its target value (see below). The fit deviation now is the sum of the squared differences between the 'value' and the 'target value' of those integral quantities that have their optimize column entry set to 'On'. Each square difference is multiplied by its individual weight (see below).

You can toggle between 'Off' and 'On' by pressing the F4 or F5 function key.

#### Target value

Here you set the target value for the fitting of the integral quantities.

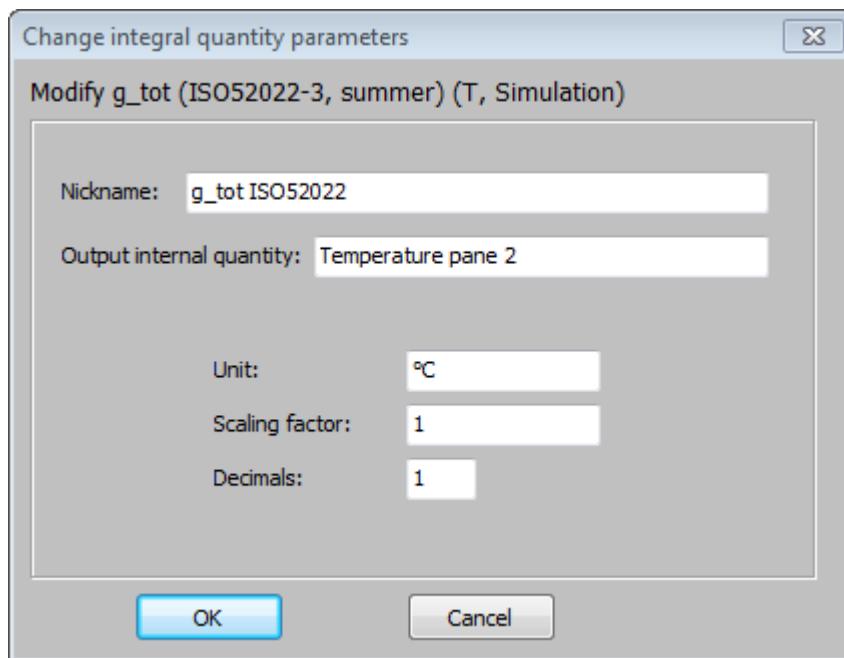
#### Weight

Here you specify the weight that multiplies the squared difference between the value and the target value of the integral quantity.

Special menu commands:

#### **Edit**

This command activates a sequence of dialogs which can be used to modify the properties of integral quantities. The sequence differs from object to object, but in most cases you will get the following:



Here you can change the text used to identify the integral quantity in views etc., i.e. the name of the object. Instead of the long name 'g\_tot (ISO52022-3, summer) (T, Simulation)' you could simply assign the nickname 'g\_tot ISO52022' to this quantity. If you leave the nickname blank the original default name will be used.

Some quantities offer several internal quantities which can be displayed as the result of the computation. You can specify the wanted quantity in the edit box 'Output internal quantity'. If you want to display the value of the integral unit together with a unit, you can enter the unit text in this dialog.

You can also set a scaling factor for the value of the integral quantity. If you want to display, for example, an averaged reflectance value (scaled between 0 and 1) in %, you should enter a scaling factor of 100 here. In addition, you should also use a proper unit (like % in this case) to indicate the scaling of the value.

Finally the dialog allows to set the number of decimals used to display the computed value.

The sequence of dialogs shown above will appear for every integral quantity. Some objects require more selections. In this case, additional dialogs will show up and allow the user to make choices. The individual dialogs are discussed below.

### Duplicate

This command simply duplicates the selected object.

### Export data

Select an item in the list and use this command to tell the item to export its data. This function is, for example, implemented for objects of type 'Color angle variation'. The angle dependence of L\*, a\* and b\* is exported to a comma separated text file (\*.csv) which can be read by Excel or similar spreadsheet programs.

### Penalty shape

Use this command to set the penalty shape function of the selected object. Penalty shape functions replace the squared difference of simulated and target value for integral quantities which is used to measure the quality of a fit. Read more about penalty shapes in the SCOUT technical manual.

### Tables

Use this command to create tables of quantities such as the applied color-matching functions or the spectral distributions of the various illuminants. The table values are written to the workbook which opens automatically.

## 2.2.2 Color coordinates

Color coordinates have several options which are set in the following dialog:



You access this dialog by selecting the color coordinate you want to change and activating the menu item **Edit** (or by clicking with the right mouse button on the color coordinate).

The following color coordinate types are implemented at present:

X,Y,Z

x,y,z

L\*, a\*, b\*

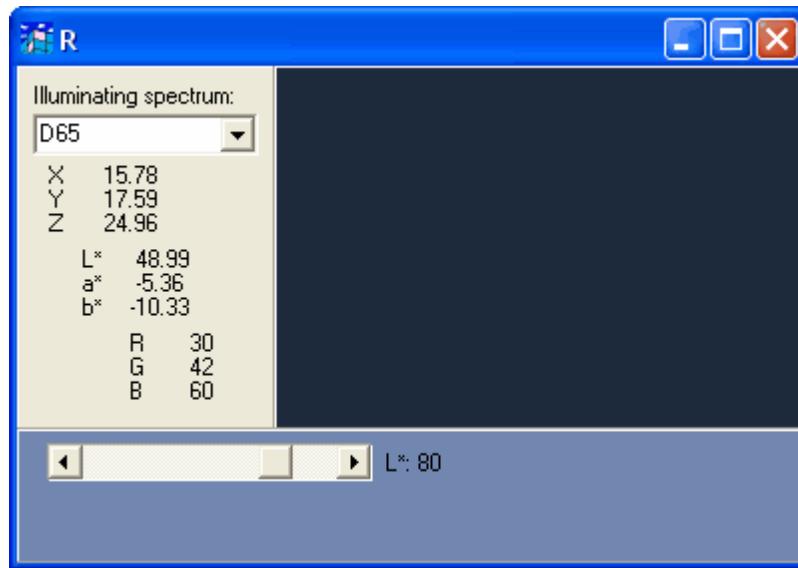
Lab

All are based on X,Y and Z which are computed using the selected illuminant and observation angle. As observation angle you can select either 2° or 10°. For these angles color matching functions are defined.

The illuminants called A, D65 and C are available for color computations. If you need more, you can import additional illuminants from an external file, including user-defined spectra. The file must be a text file with semicolon separated columns as it can be generated by Excel. An example of the import is given in the section Background/User-defined illuminants for color computation. The additional illuminants will show up in CODE color dialogs and can be selected in order to compute color coordinates. If you store a CODE configuration after you imported additional illuminants, the imported spectra will be stored as part of the configuration.

### Color view

Color view objects can be used to roughly visualize the 'color' of a spectrum. With the **Edit** command in the list of integral quantities the following window opens:



The window shows (after selection of an illuminating spectrum) the computed color coordinates X, Y, and Z and converts these to  $L^*$ ,  $a^*$ ,  $b^*$  and RGB coordinates (Red, Green, Blue). The latter are used to display the color in the upper right half of the window. Since the conversion from X, Y, Z to RGB values is not very accurate the color view should not be taken too seriously. Most PC screens are not calibrated properly anyway. So take the color impression just as a rough indication of the real color. For dark colors with low  $L^*$  values the visualization is quite useless since the color appears to be black on PC screens. For such cases the bottom rectangle can be used to display a brighter version of the color: It shows a visualization of a color with the same  $a^*$  and  $b^*$  values, but a user-defined  $L^*$  value. Use the slider position to vary  $L^*$  from 0 to 100. Note, that you must use the Update command in the list of integral quantities or in the main window of CODE in order to re-compute the color coordinates after you have changed the slider.

You can leave the color view window open while you change the optical model, e.g. while changing film thicknesses with sliders: The visualization is updated whenever the model is re-computed.

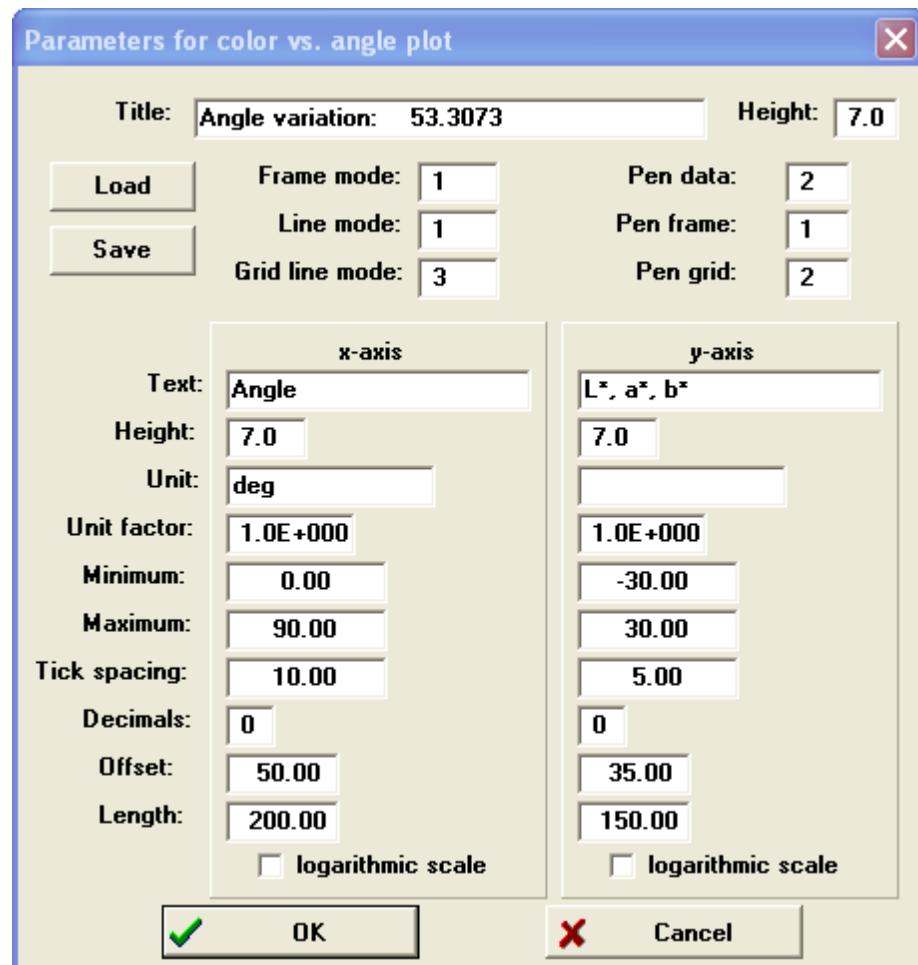
Color view objects can be displayed in a view in CODE's main window (see below).

### Color angle variation

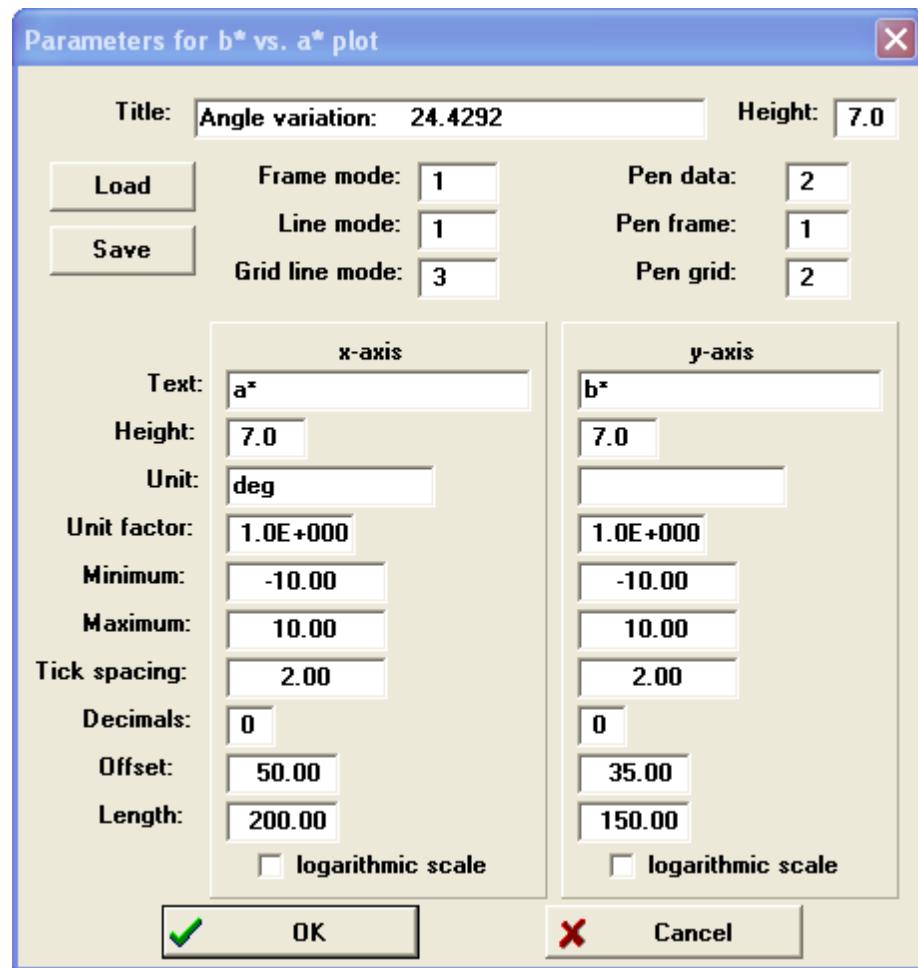
This object can be used to inspect and optimize the variation of a coating's color with angle of incidence. For a user-defined angle range, you can set target values for  $L^*$ ,  $a^*$  and  $b^*$ . For each angle, the object will compute  $L^*$ ,  $a^*$  and  $b^*$  using the actual coating. The mean square deviation between actual and target values is computed as the final number. You can set 0 as target for this deviation, and let CODE optimize parameters like film thicknesses in order to minimize the deviation, i.e. to achieve the wanted angular color variation. Be aware that for large angles the polarization of the incoming beam is important - in most cases the choice of unpolarized radiation in the corresponding spectrum object is most useful.

A view object showing a color angle variation object can display the angular dependence of  $L^*$ ,  $a^*$  and  $b^*$ . This is described below.

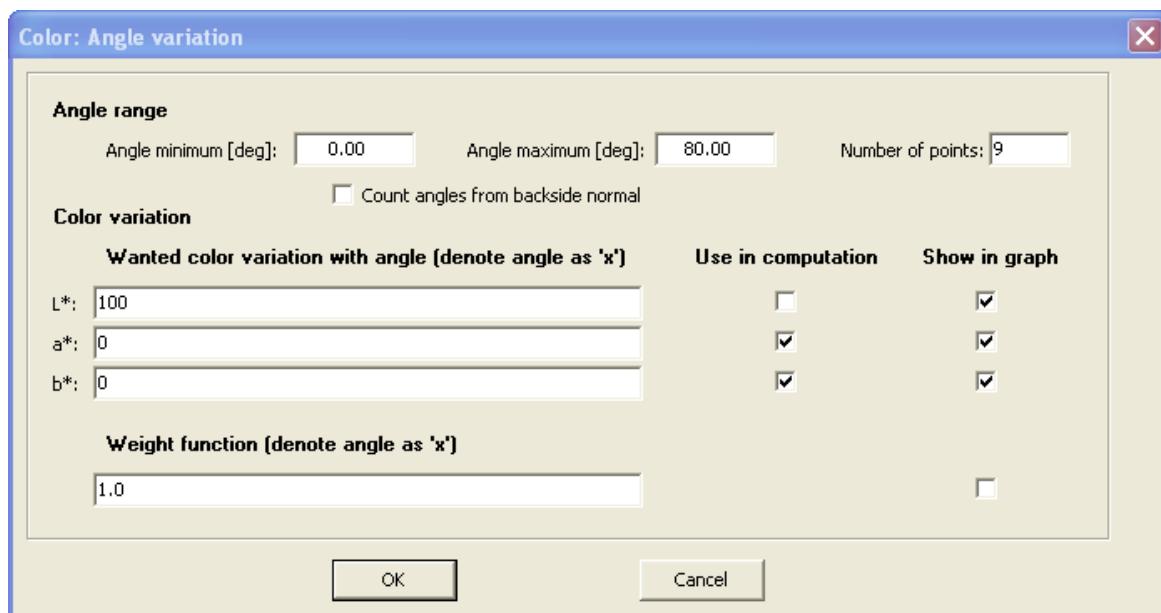
The **Edit** action for objects of this type first opens dialogs for the nickname, unit, scaling factor and decimals. Then two dialogs for graphics parameters are shown. These parameters are used by view objects that display objects of type 'Color angle variation' in the main view. The first dialog sets the parameters for displaying  $L^*$ ,  $a^*$  and  $b^*$  vs. angle of incidence (set mode=2 in the view object to show this graph):



The second dialog sets parameters for a plot showing  $b^*$  vs.  $a^*$  (mode = 3 in the view object):



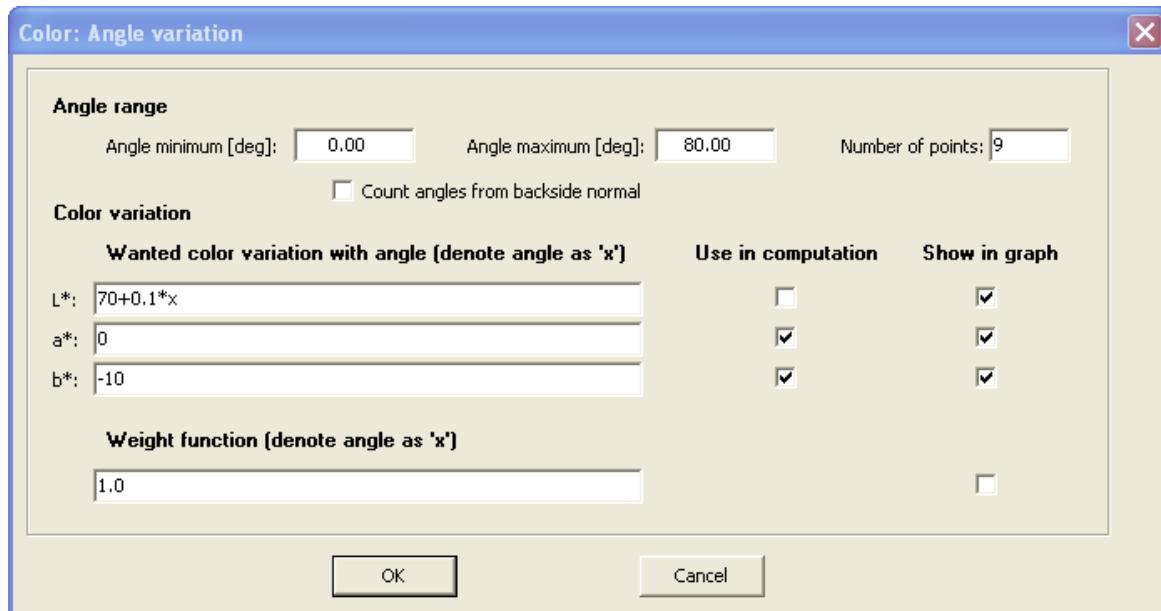
Then, in the final step, the following dialog lets you define the parameters of the computation:



In the top section the angular range is defined by setting the minimum and the maximum angle and the number of points used to scan the angle range. The example shown above means a scan of angles between 0° and 80° with a 10° resolution. If you check the option 'Count angles from backside

'normal' an angle of  $0^\circ$  means that the radiation is incident from the backside of the sample. In this case the scan from  $0^\circ$  to  $80^\circ$  covers incidence angles in the spectrum simulation object between  $180^\circ$  and  $100^\circ$ .

In the center section you can specify the wanted color variation with angle: For  $L^*$ ,  $a^*$  and  $b^*$  you have to enter an expression for the variation of the value with angle. The angle must be referred to as 'x'. The following example shows appropriate settings for a weak increase of  $L^*$  with angle and constant  $a^*$  and  $b^*$  values:



You must enter a formula for the target values in each edit field before you close the dialog. Otherwise you will get a warning message.

The checkboxes labeled 'Use in computation' can be used to control if this quantity ( $L^*$ ,  $a^*$  or  $b^*$ ) is taken into account for the computation of the deviation. In some cases it might be appropriate to leave  $L^*$  out of consideration, and optimize only  $a^*$  and  $b^*$ .

If some angles are more important for the color appearance of a coating than others, you can express this using an angular dependent weight function. The squared deviation between actual and target values for each angle is multiplied with the weight function at this angle.

The settings of the checkboxes under 'Show in graph' determine whether this quantity is displayed in the view graph.

### Color fluctuation

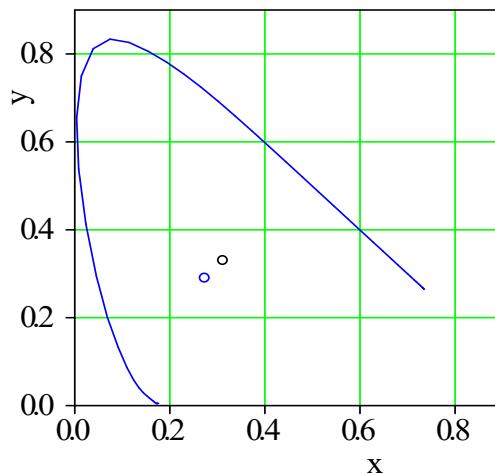
This quantity computes the variation of color in the case of fluctuating model parameters, such as thickness values. For each fluctuating quantity the color distance to the center color in  $L^*a^*b^*$  space is computed for the upper and lower boundary of the fluctuation. The final value is the average color distance of all variations. This very simple computational scheme is used as a compromise between speed and accuracy.

In the case of large layer stacks with many fluctuating layer thickness values the use of color fluctuation objects may lead to quite large computational times. On the other hand, the design of 'stable' coating properties is quite easy with color fluctuation objects.

### Dominant wavelength

The so-called dominant wavelength is computed by objects with the same name. Besides the computation of this quantity these objects also display the position of the color in the x-y-plane

which is used to compute the dominant wavelength. With **Edit** or clicking the right mouse button a new window opens showing the following graph:



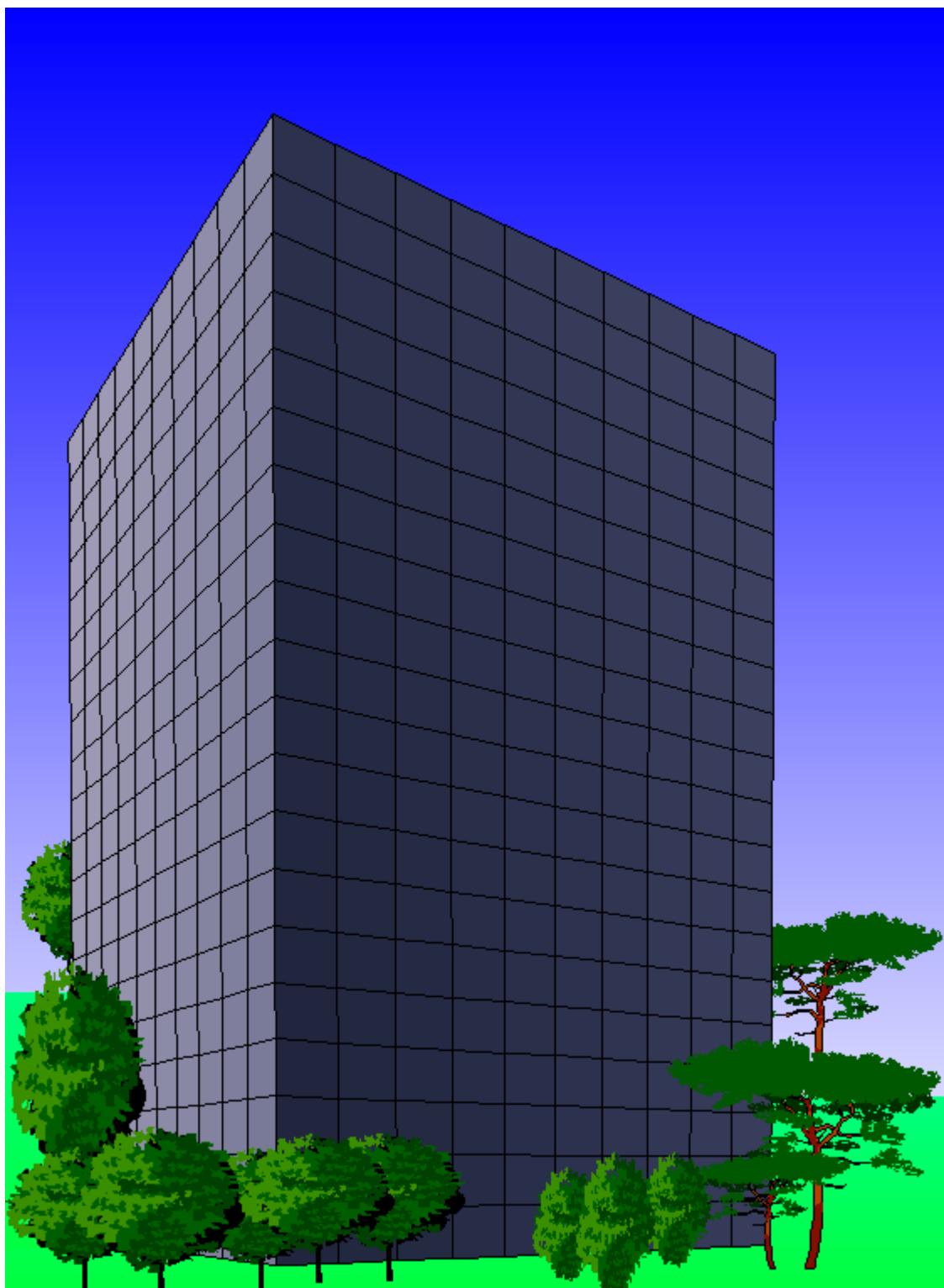
The blue line represents the position of pure spectral colors in the x-y-plane (380 to 700 nm wavelength). The black circle shows the position of 'white' whereas the blue circle shows the position of the current spectrum. If the model changes (e.g. by varying a layer thickness with a slider) the graphics is updated automatically: You can instantly see the blue circle moving through the x-y-plane. This is a very instructive tool for layer design.

### Purity

This quantity closely related to the dominant wavelength is calculated by objects named 'Purity'. Purity objects feature the same graphics as dominant wavelength objects.

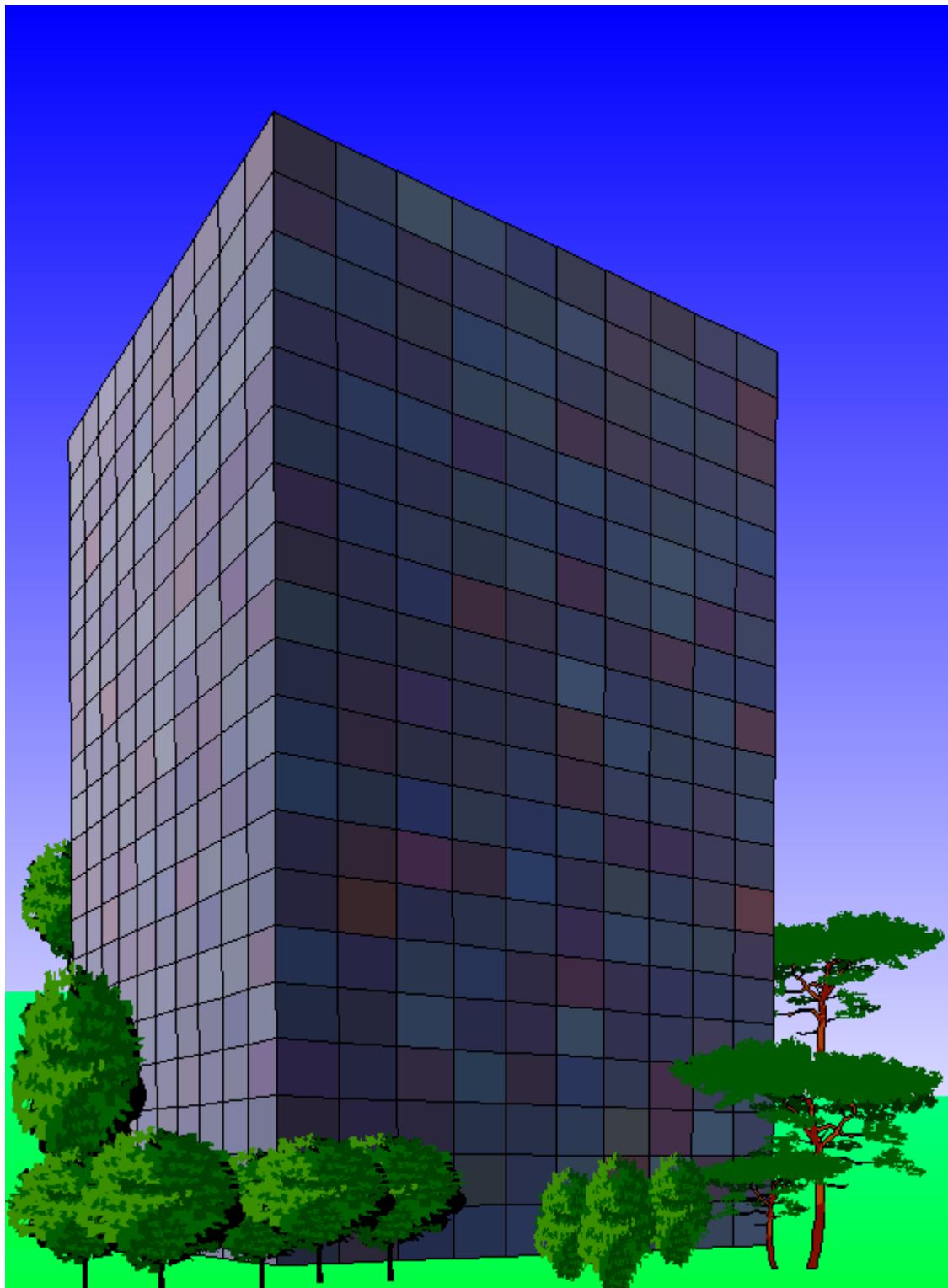
## 2.2.3 Virtual office tower

To inspect the appearance of coatings designed for architectural applications in an intuitive way you can equip a virtual office tower with your newly designed window panes. Create an object of type 'Virtual tower' and set the wanted spectrum. To be as realistic as possible this should be a reflectance spectrum using unpolarized radiation. The layer stack used for the computation of this spectrum should be the one you want to test, i.e. including all panes, air gaps and coatings ordered exactly as they will be mounted in the building. You as observer look at the building from the outside. Do not forget to set the 'sim/exp' option to 'Simulation'. Finally, use the **Edit** command to open another window where the image of the new office tower will be generated. Activate the **Build tower** command and Code starts to compute for each window of the building the angle of incidence and the corresponding reflectance spectrum. In the end the complete building is drawn:



Use the command **File|Save bitmap as ...** to store the image in a bitmap file.

If you have defined thickness fluctuations in your layer stack (see SCOUT technical manual), for every window of the building a different realization of the fluctuating parameters is used. Strong fluctuations may lead to 'designs' like the following:



## 2.2.4 Other quantities

In addition to color coordinates the following integral quantities may be computed:

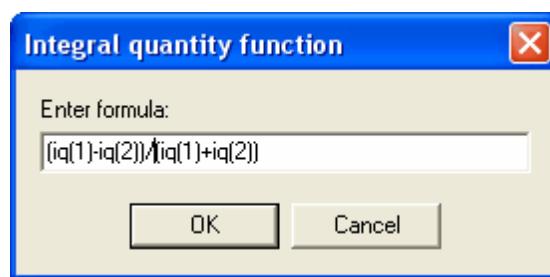
Light transmittance

Light reflectance  
 Solar direct transmittance  
 Solar direct reflectance  
 Spectrum product  
 Photocurrent  
 g (total solar energy transmittance)  
 U (Thermal transmittance)  
 NFRC VTc  
 NFRC 300 2003 solar average  
 Tdw-ISO  
 Tuv  
 NFRC U-value  
 NFRC SHGC  
  
 Emissivity (normal and effective)  
 Ra (Color rendering index)  
 Sheet resistance  
 Sound insulation

### 2.2.5 Functions of integrals quantities

Starting with object generation 2.86, the list of integral quantities may contain objects of type 'Function of int. quant.'. These objects are used in cases where you want to compute functions of other integral quantities, such as differences, sums or ratios.

If you edit an object of type 'Function of int. quant.' you are guided through the sequence of dialogs to set the nickname, unit, scaling factor and decimals. Finally a dialog to set the function is displayed:



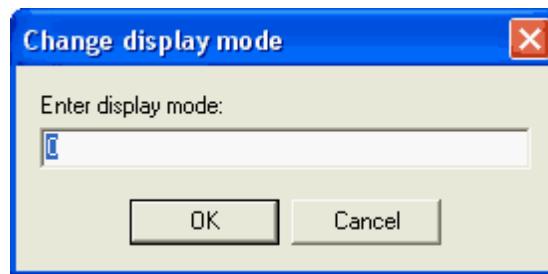
In this dialog you can refer to other integral quantities by the symbol iq(index). In the example, iq(1) refers to the first item of the list of integral quantities whereas iq(2) denotes the second item. You can refer to optical functions as well, using the symbol of(index). of(4), for example, refers to the 4th item of the list of optical functions.

The assignment of a spectrum to objects of type 'Function of int. quant.' is completely ignored in the computation.

Note that the order of integral quantities is important if you use functions of integral quantities: The computational routine in CODE computes the items of the integral quantity list from the top to the bottom. Functions that refer to items of this list should be placed underneath all the objects that are required for the computation. In the example above the function object may occur as the third item in the list or below.

## 2.3 Integral quantities in main window views

You can drag integral quantities like color coordinates from the list of integral quantities to a view list. If they carry a numerical value that value is displayed to the right of the name of the quantity. The format of the display is controlled by a mode parameter. If you select the view element and use the **Edit** command in the list of view elements, you are first asked to select the color for the text. Once the color dialog is closed, you are asked for the mode parameter in a dialog like this:



The name of the integral quantity (or the nickname, if you defined one) is shown, with left justification, inside the rectangle of the view element. If mode is set to 0, the numerical value is written to the right of the view object's rectangle, also left-justified. If mode equals 1, the numerical value is written right-justified with respect to the right side of the rectangle. Mode 1 is useful if you want to create tables of integral values in a view. The following example shows the difference between mode 0 and 1, with the gray rectangle indicating the width of the view objects' rectangles:

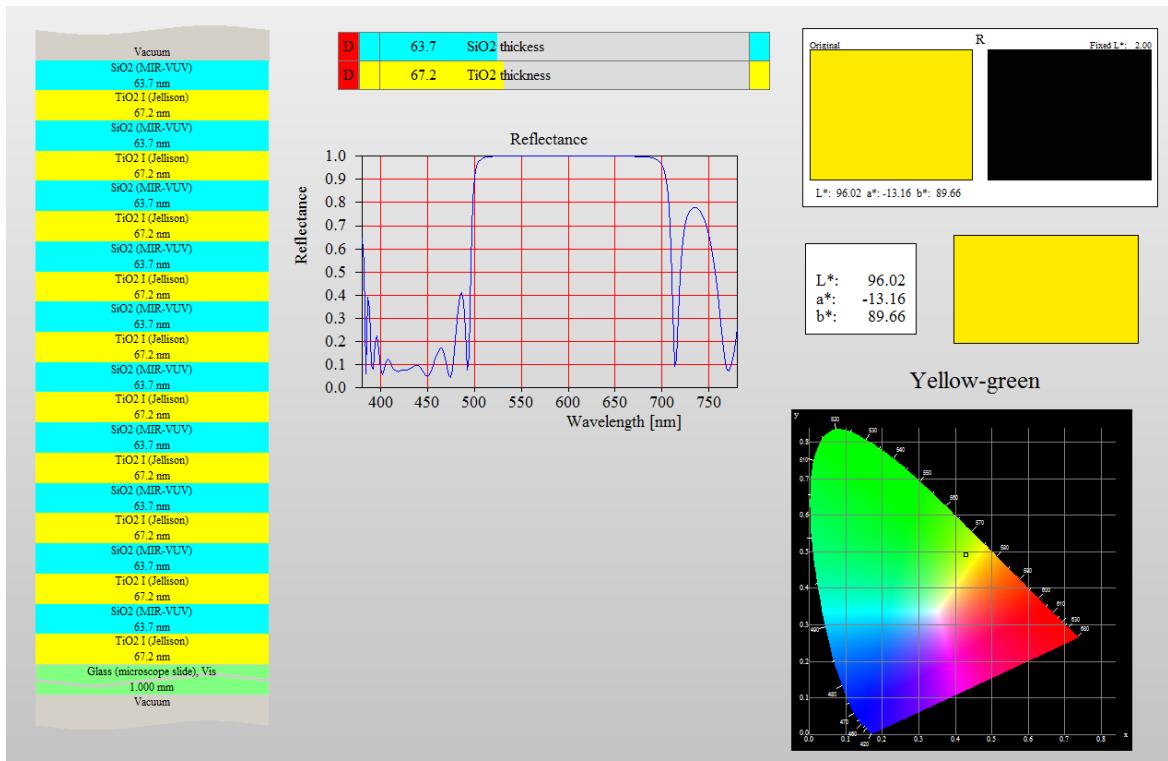
Mode 0	L*	48.9934
	b*	-10.3311
	a*	1.7144
Mode 1	L*	48.9934
	a*	1.7144
	b*	-10.3311

The following integral quantity objects are displayed in a special way when they are dragged into a view:

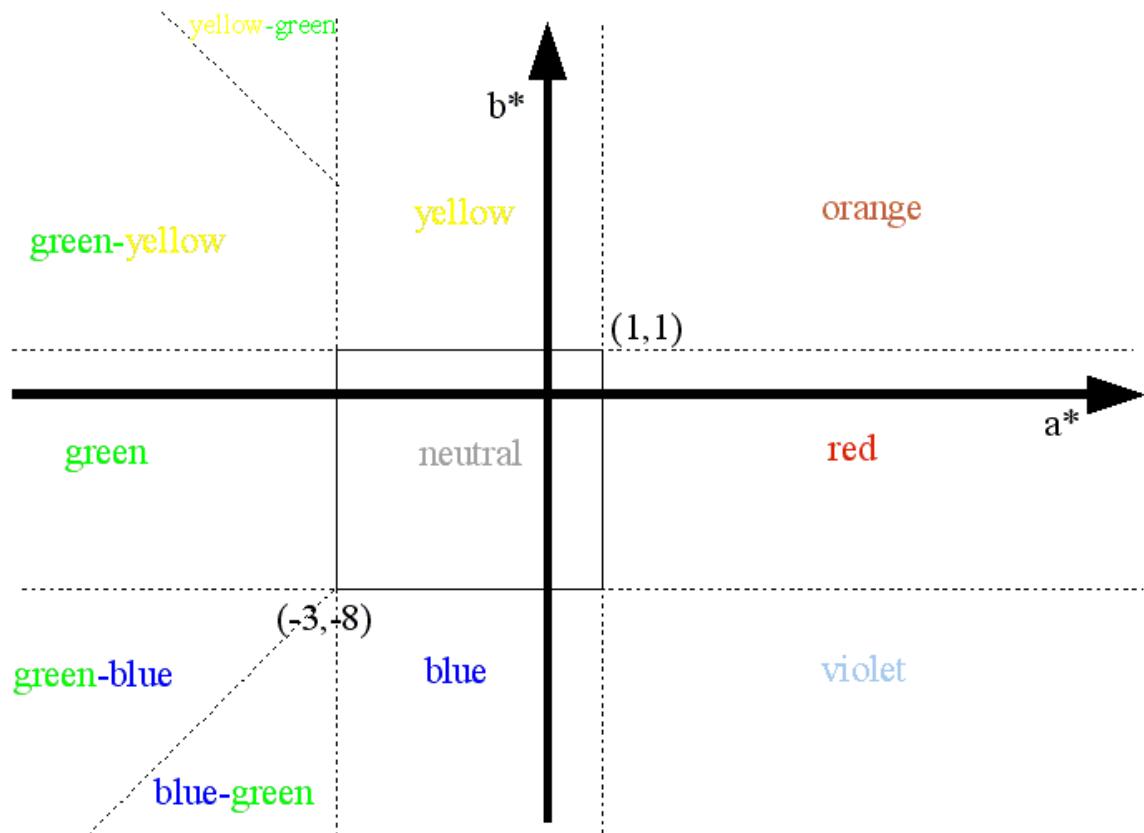
### Color view

These objects are displayed in various ways, depending on your choice. You can display a colored rectangle, the color coordinates as text, a mixture of colored rectangle and coordinates as text, the color impression as text or the position of the color in x-y-plane.

The following graph gives an overview of the options on the right side:

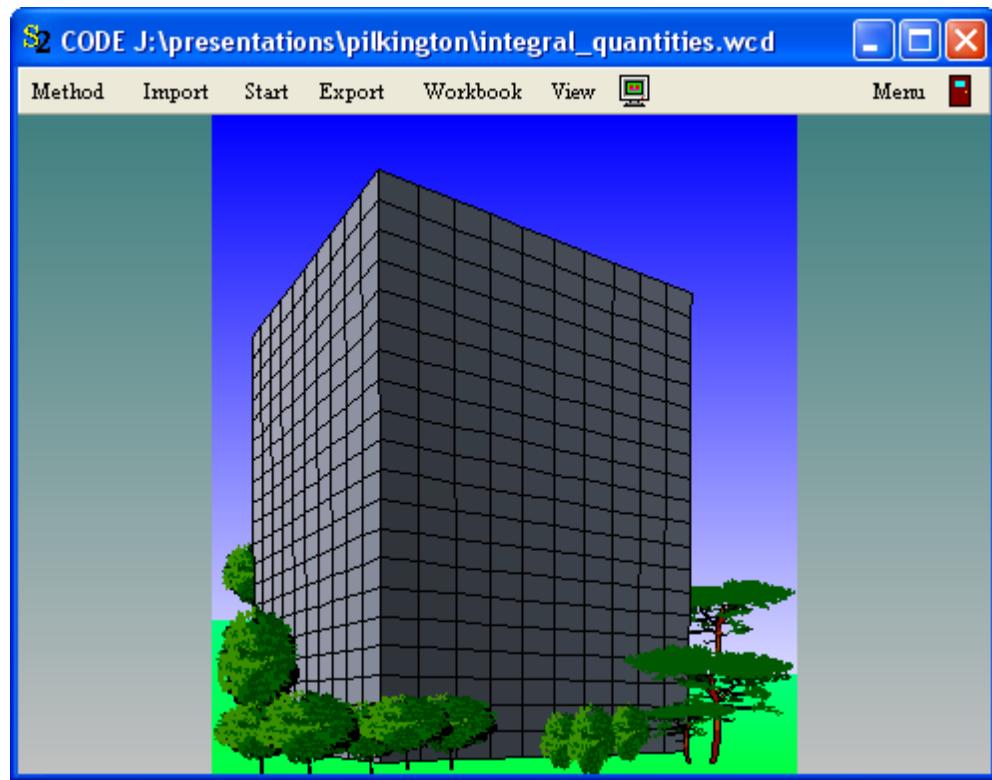


The "color impression as text" mechanism uses the coordinates  $a^*$  and  $b^*$ . They are converted to a color name according to the following map:



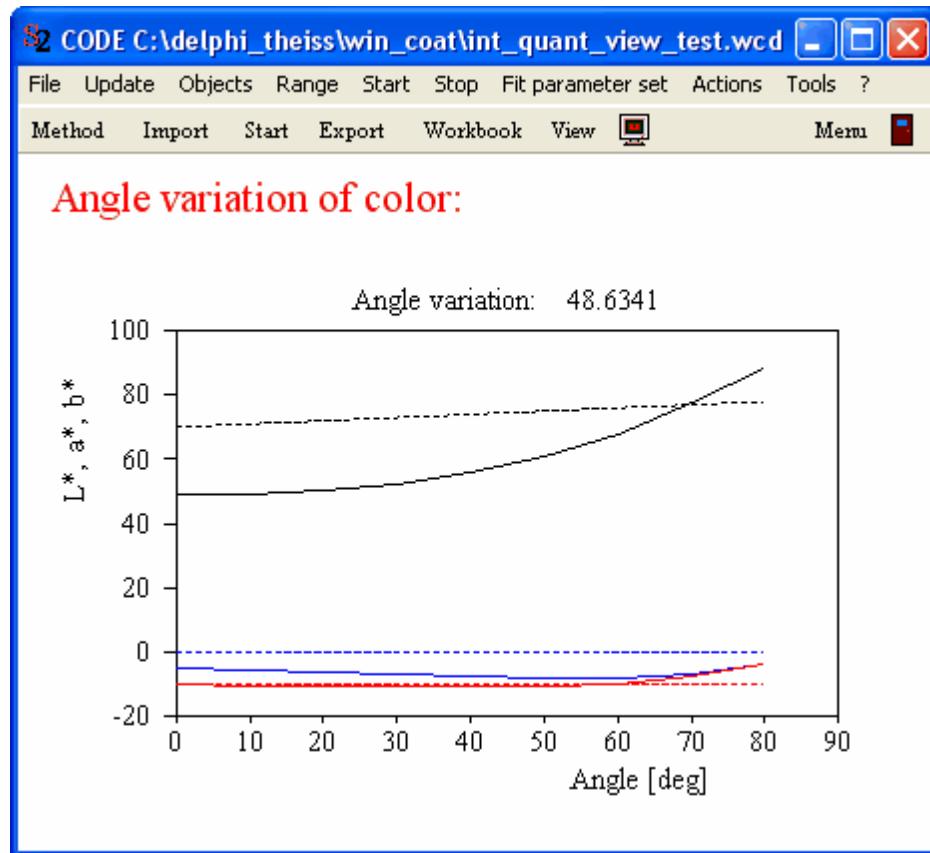
### Virtual Office Tower

If you drag objects of type 'Virtual Office Tower' to a view be aware of the large computational work that has to be done to draw the image. The build-up of the view may take quite a long time on slow computers. Tower objects are drawn in views like this:



### Color angle variation

Objects of this type are represented in views the same way as other integral quantities that produce a numerical value as output (see above, mode = 0 or mode = 1). If you set mode = 2 a graph is generated that fills the view element's rectangle:

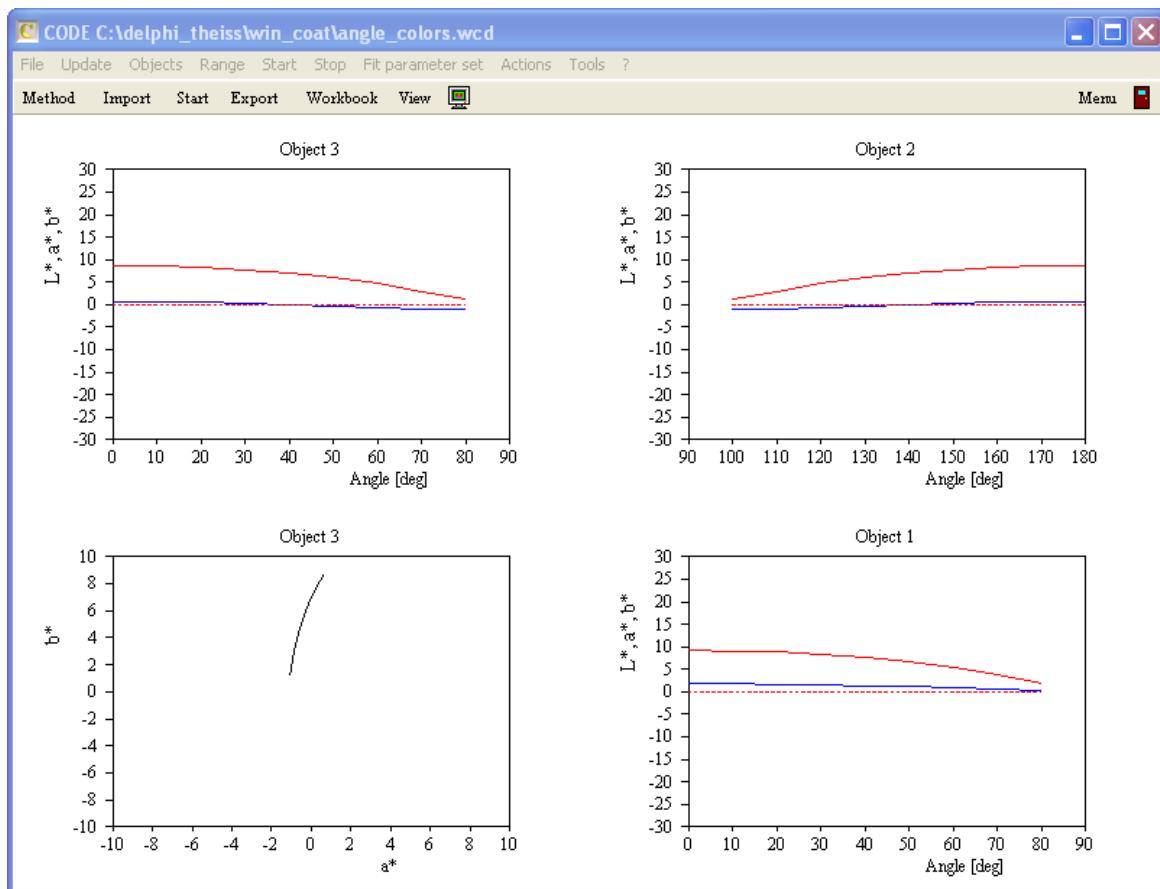


The graphics parameters used for the plot are set in the dialog that follows the objects **Edit** command (see above).

The solid lines in the graph represent the variation of L\* (black), a\* (blue) and b\* (red) with angle of incidence. The dashed lines show the corresponding target values. If you choose to display the weight function (check the appropriate checkbox in the object's dialog) it is drawn green.

Setting mode = 3 generates a 'b\*' vs. 'a\*' graph. The graphics parameters for this plot are also set in the **Edit** routine of the color angle variation object (see above).

You can generate several view objects that refer to the same color angle variation object like in the case below: The two graphs on the left show the same color angle variation object:



## 2.4 Other differences to SCOUT

In addition to the integral quantity list, there are some minor differences between CODE and SCOUT. These are summarized in this section.

### Automatic loading of a start configuration

After program start CODE looks in the program directory file for a configuration named 'Start.wcd'. If it exists it is loaded automatically.

### Automatic display of spectra and sliders

In versions of CODE created after 2001 the automatic display of spectra and sliders after loading a new configuration can be set as option of the configuration.

### Parameter variation

Like SCOUT, CODE computes the variation of optical spectra with a parameter variation. In addition, the variation of all integral quantities is computed as well. The additional data are stored in the worksheet 'Colors'.

### Parameter fluctuation

As in the case of parameter variations, CODE computes in addition to all SCOUT work the fluctuation of all integral quantities and stores the results in the worksheet 'Colors fluctuation'. No automatic graphical display is generated. In some cases it might be useful to create graphs using

objects of type 'Workbook display (rows)', e.g. in order to show color variations in a ' $b^*$  vs  $a^*$ ' plot. See the SCOUT manual for details about user-defined views.

Starting with object generation 2.55, we recommend the use of the new object type 'Color box' in the list of special computations (see SCOUT manual) in order to generate graphs of type ' $b^*$  vs.  $a^*$ '.

#### Additional optical functions

The values of integral quantities can be displayed in the list of 'Optical functions' (see SCOUT technical manual). They will then appear in the workbook when you export results using the **Export** command. In the **batch control window** the values of optical functions are listed as well for every sample.

Just drag the wanted items from the list of integral quantities to the list of optical functions.

## 2.5 Presentations

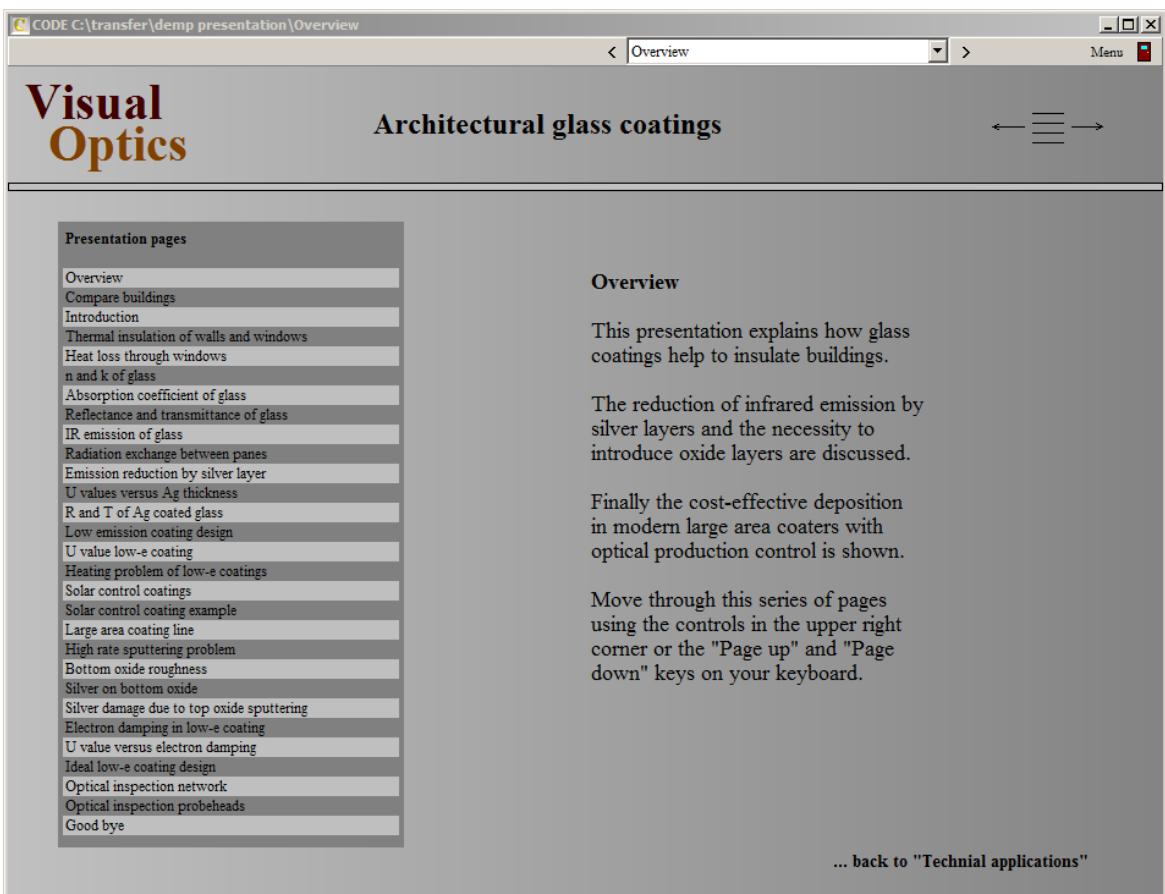
Presentations are sequences of CODE configurations that you can use to present and organize CODE work. You can easily navigate through the configurations like you are jumping from image to image in a slide show. The difference to a slide show, however, is that each configuration is a fully functional CODE configuration with active sliders or running animations.

Presentations have been introduced to CODE with object generation 3.98 (September 2014).

### 2.5.1 Demo presentation

On our website you can download a demo presentation. The link is [www.mtheiss.com/download/demo\\_presentation.zip](http://www.mtheiss.com/download/demo_presentation.zip). You will get a zip-file of about 11 MB which contains a folder 'demo presentation'. Extract the folder to your harddisc and open the presentation with the menu command 'Presentation/Load'. You will get a file dialog searching for a presentation file with extension \*.cpr. Pick the file the\_show.cpr in the folder of the demo presentation.

CODE will load the first page of the presentation and look like this:



The demo presentation has been taken from our interactive optics course VisualOptics. You can use this presentation to follow the instructions and explanations below.

## 2.5.2 Generating presentations

A new generation is generated by the menu command 'Presentation/New'. You are asked to input a new filename which will get the extension \*.cpr (for Code PResentation). It is recommended to generate the new file in a new, empty directory - all files of the presentation will be collected in this folder. The presentation file contains a list of the configurations that deliver the content of the pages of the presentation.

Initially the presentation is empty. As soon as the presentation file has been generated you can add configurations to the presentation. With the usual command 'File/open' you can load CODE configurations. In order to add the current configuration to the presentation, use the command 'Presentation/Add configuration to presentation'. You are asked for a name of the page - enter a useful and descriptive name without file extension. Be aware, however, that the configuration is saved in the presentation folder using this name - hence you cannot use characters which are not allowed to occur in filenames.

When a new presentation is created, CODE gets a new item in its treeview (this opens with F7) called 'Presentation'. This treeview item represents the list of configurations. With a right mouse click on the treeview item you can open this list. Using the red arrows you can change the sequence of pages if you need to correct this. After a change of the page sequence you have to execute the menu command 'Presentation/Save' to save your changes.

### Automatic saving of page modifications

If you open the presentation list in the treeview, you can check or uncheck its local menu command 'File/Options/Automatically save configurations when presentation pages change'. If checked, configurations are saved when you jump to a new page. This is useful while you are developing the presentation since it can save a huge number of mouse clicks. On the other hand, you can easily damage a carefully made configuration by some unwanted slider movements and save this undesired state accidentally. **To be on the safe side, uncheck the option of automatic saving.**

When you exit CODE and you have generated or modified a presentation, you will be asked if you would like to save the presentation.

### 2.5.3 Navigation

#### Keyboard commands

Within a presentation you can navigate from page to page (i.e. from CODE configuration to CODE configuration) with the **Page up** (jumps to the new page) and **Page down** (jumps to the previous page) keys on your keyboard.

#### Dropdown box

In addition, CODE shows a dropdown box on the button panel which represents a list of all pages:



Selecting an item in the dropdown box means to jump to this page immediately.

If the button panel is not visible you can show it by pressing 'b' on your keyboard. Pressing 'b' again hides the button panel again.

#### View elements

There are also some view elements for navigation through presentations. First of all, you can use an item of type 'List view' and assign the presentation to it. In this case you will see a list of the pages - clicking on a line means to jump to this page. Obviously it makes sense to place such a view element on an overview page of your presentation (either at the beginning or the end). If you have picked descriptive page names it can directly be used as a table of content. An example is shown on the left side of the graph below:



If you would like to have a visual navigation help on each page of the presentation you can use a view object of type 'Presentation control' in each configuration. When you generate an object like this, CODE picks a metafile graph named 'presentation\_control.emf' from its subdirectory 'Pictures' and uses this picture for the new view element. It looks like this:



There are 3 active regions that respond to mouse clicks. The left part of the graph triggers a jump to the previous page, the right section moves on the next page. A click on the center opens the first page of the presentation - in most cases this page gives an overview of all pages (see above) and can be used to quickly navigate to any page.

'Presentation control' objects are descendants of 'metafile view' elements. You can easily change the graph to a metafile image of your choice.

## 2.5.4 Publish

In order to send your presentation to someone else you have the following options.

If the receiver has CODE as well you can simply send the whole presentation folder. If the folder content is big you could pack it into a zip file and make the file available through download.

If the recipient does not have CODE you can export the presentation to a PDF file, using the command 'Presentation/Generate PDF document of whole presentation'. After you execute this command, CODE will show a file dialog for the PDF document. It will generate a new PDF file, and then load each configuration, following the sequence of the presentation. Each page of the

presentation is entered as a page in the PDF. If a configuration has more than one view it will output each view as a single PDF page. Finally the PDF is stored and you can send it to the wanted destination.

## 2.6 OLE automation

### 2.6.1 Introduction

#### Automatic installation

The installation of the OLE server is done by the CODE setup routine. It registers the application as OLE automation server named 'code.colors' which is after the setup procedure 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 CODE as OLE server, use the command

```
c:\program files\code\code.exe /unregserver
```

where the path to the executable file has to be adjusted, of course.

To manually register CODE please use the command

```
c:\program files\code\code.exe /regserver
```

CODE has all OLE commands and properties that our SCOUT software has, and some more in addition. This section lists all methods and properties, repeating all SCOUT items for completeness.

CODE as OLE server must be called as 'code.colors', like in the following line of a VisualBasic example:

```
Set wcd = CreateObject("code.colors")
```

The following sections describe the available OLE methods and properties. The application of OLE automation is demonstrated in an Excel VisualBasic example.

### 2.6.2 Methods

#### 2.6.2.1 acquire

This command executes the global acquire command in CODE: All spectrometers are told to record new spectra.

#### 2.6.2.2 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

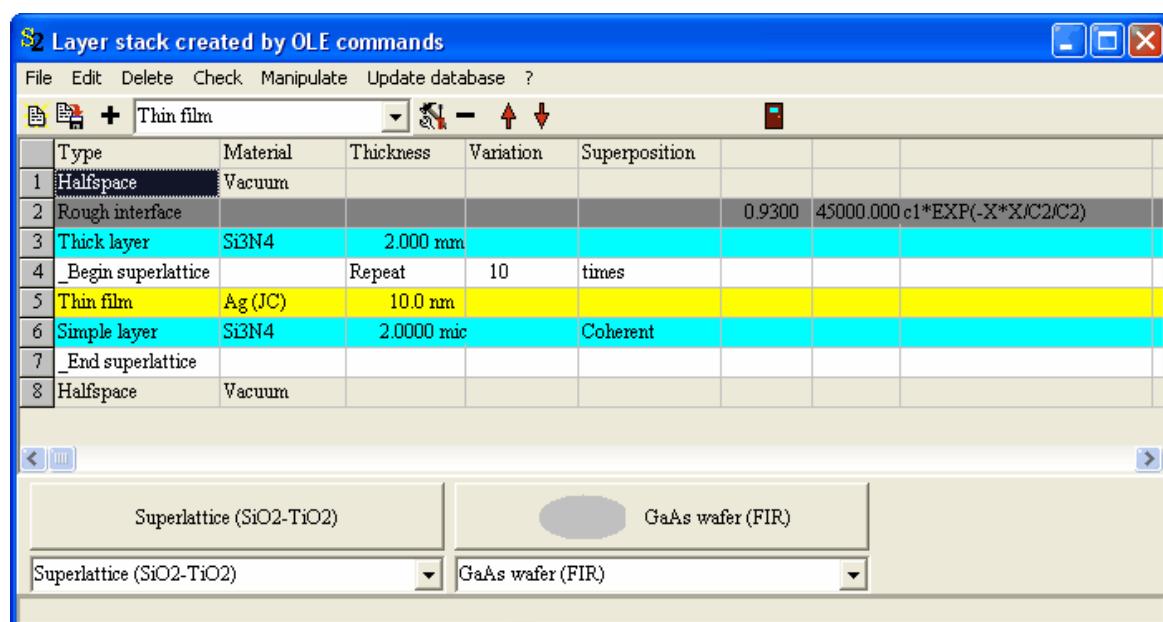
- "Simple layer" + tab + string specifying the material + tab + string specifying the thickness in cm(!) + tab + string specifying the partial wave superposition ("coherent" or "incoherent")
- "Thin film" + tab + string specifying the material + tab + string specifying the thickness in cm(!)
- "Thick 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)
- Starting with object generation 3.81: "Anisotropic layer" + tab + string specifying the material to be used for the x-y-plane + tab + string specifying the material to be used for the z-plane + tab + string specifying the thickness in cm(!) + tab + string specifying the partial wave superposition ("coherent" or "incoherent")

VisualBasic examples:

The VisualBasic code

```
wcd.clear_fit_parameters
wcd.clear_layer_stack (1)
wcd.clear_material_list
Call wcd.add_layer_definition_on_top(1, "End superlattice")
Call wcd.add_layer_definition_on_top(1, "Simple layer" + Chr(9) + "Si3N4" + Chr(9) + "0.0002" +
Chr(9) + "coherent")
Call wcd.add_layer_definition_on_top(1, "Thin film" + Chr(9) + "Ag (JC)" + Chr(9) + "0.000001")
Call wcd.add_layer_definition_on_top(1, "Begin superlattice" + Chr(9) + "10")
Call wcd.add_layer_definition_on_top(1, "Thick layer" + Chr(9) + "Si3N4" + Chr(9) + "0.2")
Call wcd.add_layer_definition_on_top(1, "Rough interface" + Chr(9) + "c1*EXP(-X*X/C2/C2)" +
Chr(9) + "0.93" + Chr(9) + "45000.0")
wcd.update_data
wcd.update_plot
```

creates the following layer stack:



Here is an VB example for the 'Variable thickness' layer type:

```
Call wcd.add_layer_definition_on_top(1, "Variable thickness" + Chr(9) + "Si3N4" + Chr(9) +
"0.0006" + Chr(9) + "0.00004" + Chr(9) + "2")
```

The following is an example for adding an anisotropic layer to the stack:

```
Call wcd.add_layer_definition_on_top(1, "Anisotropic layer" + Chr(9) + "material_x_y" + Chr(9) +
"material_z" + Chr(9) + "0.000003" + Chr(9) + "coherent")
```

#### **2.6.2.3 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 wcd.add_layer_on_top(1, "Thin film", "Si3N4", 0.45, "micron")
Call wcd.add_layer_on_top(1, "Thin film", "Ag (JC)", 19.4, "nm")
```

#### **2.6.2.4 add\_to\_report(text : string)**

Adds the submitted string to the CODE report window.

#### **2.6.2.5 acquire\_dark**

Use this command to tell all spectrometers to record their dark spectrum.

#### **2.6.2.6 acquire\_reference**

Use this command to tell all spectrometers to record their reference spectrum.

### 2.6.2.7 **apply\_factor\_in\_fit\_parameter\_transfers**

Execute this command if you want to apply the fit parameter factor if you export or import fit parameter values. Since the default setting is that factors are not used, you have to apply this command before you start to transfer fit parameters and you want the factor to be taken into account.

### 2.6.2.8 **automatic\_spectrum\_scaling(i : index)**

Executes automatic scaling of the graphics of the i-th spectrum.

### 2.6.2.9 **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:

```
wcd.add_sample("S413 #1")
```

### 2.6.2.10 **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:

```
wcd.batch_clear_samples
```

### 2.6.2.11 **batch\_go**

Starts the batch fit based on the current batch control workbook settings. CODE 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
```

### 2.6.2.12 **batch\_restore\_fit\_results (sample : string)**

Reloads the obtained batch fit parameters and the spectra for the specified sample.

### 2.6.2.13 **clear\_fit\_parameters**

This command deletes all items from the list of fit parameters.

VisualBasic example:

```
wcd.clear_fit_parameters
```

### 2.6.2.14 **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:  
wcd.clear\_layer\_stack (1)

#### 2.6.2.15 clear\_material\_list

Clears the list of materials: All items are deleted, with the exception of 'Vacuum'.

VisualBasic example:  
wcd.clear\_material\_list

#### 2.6.2.16 clear\_report

Clears the CODE report window to prepare it for subsequent entries.

#### 2.6.2.17 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 CODE 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 wcd As Object

Public Sub fit()  
'Create the OLE server  
Set wcd = CreateObject("code.colors")  
'Load a pre-configured configuration with a suitable reflectance spectrum and spectral range  
wcd.configuration\_file = "c:\test\ole\_test.wcd"

```
'Show SCOUT on the screen
wcd.Show

'Clear the layer stack - we will build up a new one in a moment
wcd.clear_layer_stack (1)
'Clear the list of materials - materials will be taken from the database
wcd.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 wcd.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 wcd.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 wcd.load_experiment(1, "C:\test\ag_10.std", 1)

'Clear the list of fitparameters
wcd.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 wcd.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 wcd.create_fit_parameter(1, 3, 1, 0, 2)

'Set boundaries for the damping constant which is fit parameter 2 now (after the layer thickness).
wcd.fit_parameter_value_min(2) = 100
wcd.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 wcd.create_fit_parameter(1, 3, 2, 0, 2)
Call wcd.create_fit_parameter(1, 3, 3, 0, 2)
Call wcd.create_fit_parameter(1, 3, 4, 0, 2)
Call wcd.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 wcd.create_fit_parameter(3, 1, 0, 0, 1)

'Select - also for demonstration only - the first master parameter as fitparameter.
Call wcd.create_fit_parameter(4, 1, 0, 0, 1)

'Re-compute the model and update the main view
wcd.update_data
```

```
wcd.update_plot

'Start the fit
wcd.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 wcd.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) = wcd.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) = wcd.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) = wcd.kappa(3, 10000000 / 1000)

End Sub
```

#### 2.6.2.18 delete\_fit\_parameter (index : integer)

Deletes the fit parameter with the given index. Fit parameters are counted 1, 2, 3, ...

#### 2.6.2.19 df\_load\_from\_file

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).

#### 2.6.2.20 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

### **2.6.2.21 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).

### **2.6.2.22 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

### **2.6.2.23 df\_to\_clipboard (n : integer)**

Tells the n-th material in the list of materials to copy its graphics to the clipboard.

### **2.6.2.24 divide\_all\_measured\_plate\_spectra\_by\_100**

This command tells all layer stacks to tell all layers of type 'Measured plate' to divide their spectra by 100. Use this command after you imported spectra that were given in percent.

### **2.6.2.25 do\_no\_apply\_factor\_in\_fit\_parameter\_transfers**

Execute this command if you do not want to apply the fit parameter factor if you export or import fit parameter values. This is the default setting.

### **2.6.2.26 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:

```
wcd.export_measured_spectrum(1,"c:/temp/dummy2.spc",4)
```

##### **2.6.2.27 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:

```
wcd.export_measured_spectrum(1,"c:/temp/dummy2.spc",4)
```

##### **2.6.2.28 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.

##### **2.6.2.29 hide**

Hides CODE from the screen.

##### **2.6.2.30 hide\_statusbar**

Hides the status bar.

##### **2.6.2.31 import\_measured\_data\_via\_variants(index : string; spectral\_positions, spectral\_data : variant; unit\_string : string)**

Use this command to pass experimental spectra to a CODE 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).

### 2.6.2.32 **function import\_measured\_plate\_spectra(stack\_index, layer\_index : integer; t, rt, rb : string; file\_format, option\_t, option\_rt, option\_rb : integer) : integer**

This function is used to load measured spectra into a layer of type 'Measured plate'. The layer is specified by the index of the stack in which it exists (counting 1, 2, 3, ...) and the layer index (also counted like 1, 2, 3, ...).

The strings t, rt and rb are used to pass the filenames for the transmittance, top side reflectance and bottom side reflectance.

File\_format sets the format of the file.

The integers option\_t, option\_rt and option\_rb define file import options that are used if you are reading from a file containing several spectra. option\_t=7 means that the transmittance spectra is the seventh spectrum in the file, for example.

The return value is 0 if the spectra have been successfully loaded. Otherwise a value larger than 0 is returned.

VisualBasic call:

```
dim r as integer
r = wcd.import_measured_plate_spectra(1,2,"c:\temp\t.spc","c:\temp\r_top.spc", "c:
\temp\r_bottom.spc", 4, 1, 1, 1)
if r<>0 then
    msgbox "Error importing measured plate spectra!"
endif
```

### 2.6.2.33 **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 wcd.load_df(1, "c:/temp/df2.dfm")
```

### 2.6.2.34 **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'.

### 2.6.2.35 **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:

```
wcd.load_material_from_database ("Si3N4")
```

**2.6.2.36 pane\_wizard\_divide\_r\_by\_100**

Divides the reflectance spectrum of the pane wizard by 100.

**2.6.2.37 pane\_wizard\_divide\_t\_by\_100**

Divides the reflectance spectrum of the pane wizard by 100.

**2.6.2.38 pane\_wizard\_export\_results (material : string; version : float; username, comment : string)**

The pane wizard analysis is done and the results are exported to the database. The filename is given by material, the file version by version. The database item is stored using the username and comment specified by the username and comment parameters.

**2.6.2.39 pane\_wizard\_load\_r (filename:string,file\_format:integer,spectral\_unit:string)**

Loads a spectrum into the reflectance field of the pane wizard. The data are read from the file given by filename, using the file format specified with file\_format. The spectral unit of the data must be given in spectral\_unit.

**2.6.2.40 pane\_wizard\_load\_t (filename:string,file\_format:integer,spectral\_unit:string)**

Loads a spectrum into the transmittance field of the pane wizard. The data are read from the file given by filename, using the file format specified with file\_format. The spectral unit of the data must be given in spectral\_unit.

**2.6.2.41 pane\_wizard\_set\_thickness (thickness : float)**

Use this command to pass the thickness of the pane to the pane wizard. The thickness must be specified in micrometers.

**2.6.2.42 pane\_wizard\_spectral\_range (min, max : float; points : integer;spectral\_unit : string)**

Sets the spectral range of the pane wizard data analysis. The spectral unit is given by spectral\_unit. The number of data points is specified by points, whereas the minimum and the maximum of the spectral range is set by min and max, respectively.

**2.6.2.43 rename\_material (old\_name, new\_name : string)**

Renames the material called old\_name to new\_name.

**2.6.2.44 report\_to\_clipboard**

Copies the content of the CODE report window to the clipboard.

**2.6.2.45 report\_to\_file(filename:string)**

Saves the content of the CODE report window to a text file the name of which is the string passed to the method.

#### 2.6.2.46 **prepare\_shutdown**

This method performs internal clean-up actions which are required before you shutdown the OLE server.

There are 2 important rules applying this command:

**Do never call this command when you want to continue to use the OLE server!**

**Do call this command before you shutdown the OLE server!**

#### 2.6.2.47 **print\_fit(i : integer)**

Tells CODE 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)
```

#### 2.6.2.48 **save\_configuration(filename : string)**

Saves the complete CODE configuration to the specified file.

**Warning: An already existing file it will be overwritten without warning.**

VisualBasic example:

```
wcd.save_configuration ("c:/temp/fit_sample1.wcd")
```

#### 2.6.2.49 **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 wcd.save_df(2, "c:/temp/df2.df")
```

#### 2.6.2.50 **scout\_normal**

Show the program window like a normal window which can be hidden by other windows.

#### 2.6.2.51 **scout\_on\_top**

Show the program window on top of other windows on the screen.

### 2.6.2.52 `searchfiles(dir,pattern : string)`

Executes a dialog to select multiple files that can be used afterwards for CODE 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'.

### 2.6.2.53 `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.

### 2.6.2.54 `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.

### 2.6.2.55 `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. CODE reacts as if you had executed the **Range** command in the main window.

The new spectral range is defined by the minumum 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 CODE
'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
' m m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

### 2.6.2.56 `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. CODE 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 minumum 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 CODE
--------------------	--	----------------------------

'cm^-1' or '1/cm'	>	wavenumbers
'nm'	>	nanometers
'm m' or 'micron'	>	microns
'eV'	>	electron volts
'THz'	>	THz

#### 2.6.2.57 **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.

#### 2.6.2.58 **show**

Displays CODE on the screen.

#### 2.6.2.59 **show\_maximized**

Shows the main window maximized on the screen.

#### 2.6.2.60 **show\_minimized**

Minimizes the main window on the screen.

#### 2.6.2.61 **show\_normal**

Shows the window in its normal position and size, i.e. not maximized and not minimized.

#### 2.6.2.62 **show\_statusbar**

Shows the status bar.

#### 2.6.2.63 **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

#### 2.6.2.64 **smooth\_data\_like\_batch\_control**

Smooth the experimental data using the settings of the batch control window for each spectrum.

**2.6.2.65 spectrum\_to\_clipboard (n:integer)**

Tells the n-th spectrum object to copy its graphics to the clipboard.

**2.6.2.66 start**

Starts the parameter fit using the current configuration.

**2.6.2.67 stop**

Stops the parameter fit.

**2.6.2.68 update\_data**

Forces the re-computation of the CODE configuration. This command should be used after parameter changes to recalculate the optical model.

**2.6.2.69 update\_plot**

Forces the re-drawing of all visible windows of CODE, including the main window displaying the current view.

**2.6.2.70 view\_to\_clipboard\_wmf**

Sends the current view as metafile (vector graphics) to the clipboard.

**2.6.2.71 view\_to\_clipboard\_bmp**

Sends the current view as bitmap to the clipboard.

## 2.6.3 Properties

**2.6.3.1 a\_filename (string)**

Executes the standard dialog to select a filename. Returns the selected name. VBA example:

```
Range("results").Offset(4 + i, 1) = wcd.a_filename;
```

Writing this property has no effect.

**2.6.3.2 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.

**2.6.3.3 angle\_distribution\_no\_points (spectrum\_index:integer) (integer)**

Returns or sets the number of angles of the angle of incidence distribution of the spectrum which is specified by the spectrum\_index. Spectra are counted 1, 2, 3, ...

If the number of angles is set to a new value, nothing happens if the new number of angles agrees to the current number of angles. If the new number is larger than the current one, angles are added at

the end of the list. If the new number is smaller, angles are deleted from the end of the list until the desired number of angles is reached.

VBA examples:

```
wcd.angle_distribution_no_points(1) = 5
```

```
MsgBox wcd.angle_distribution_no_points(1)
```

#### **2.6.3.4 angle\_distribution\_angle (spectrum\_index, angle\_index : integer) (float)**

Returns or sets an angle of the angle of incidence distribution of the spectrum which is specified by the spectrum\_index. Spectra are counted 1, 2, 3, ...

The angle which is returned or set is specified by the angle\_index. Angles in the distribution are counted 1, 2, 3, ...

VBA code examples:

```
wcd.angle_distribution_angle(1, 1) = 15  
wcd.angle_distribution_angle(1, 2) = 25  
wcd.angle_distribution_angle(1, 3) = 35  
wcd.angle_distribution_angle(1, 4) = 45  
wcd.angle_distribution_angle(1, 5) = 55
```

```
MsgBox wcd.angle_distribution_angle(1, 2)
```

#### **2.6.3.5 angle\_distribution\_weight (spectrum\_index, angle\_index : integer) (float)**

Returns or sets a weight of the angle of incidence distribution of the spectrum which is specified by the spectrum\_index. Spectra are counted 1, 2, 3, ...

The weight which is returned or set is specified by the angle\_index. Angles in the distribution are counted 1, 2, 3, ...

Please take into account that there is no automatic normalization of the weights of a distribution. It is your responsibility to properly normalize angular weights.

VBA code examples:

```
wcd.angle_distribution_weight(1, 1) = 0.1  
wcd.angle_distribution_weight(1, 2) = 0.3  
wcd.angle_distribution_weight(1, 3) = 0.25  
wcd.angle_distribution_weight(1, 4) = 0.15  
wcd.angle_distribution_weight(1, 5) = 0.2
```

```
MsgBox wcd.angle_distribution_weight(1, 2)
```

### 2.6.3.6 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.

### 2.6.3.7 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:

```
wcd.batch_configuration_file = "f:\examples\batch_parameters.xls"
```

This statement causes the loading of the specified batch configuration file.

### 2.6.3.8 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 = wcd.batch_fit_deviat ion("s413 #3")
```

### 2.6.3.9 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 = wcd.batch_fitparameter_name(5)
```

### 2.6.3.10 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 = wcd.batch_fitparameter_value("s413 #3", 5)
```

### 2.6.3.11 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:

```
wcd.batch_input_path(1) = "C:\test\data\trans\"  
wcd.batch_input_path(2) = "C:\test\data\ref\"
```

#### 2.6.3.12 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 = wcd.batch_name_of_spectrum(2)
```

Writing this property has no effect.

#### 2.6.3.13 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 = wcd.batch_number_of_spectra
```

Writing this property has no effect.

#### 2.6.3.14 batch\_prefit\_result(sample : string) (string)

Returns the pre-fit set found for the specified sample during the batch fit.

VBA example:

```
start_set = wcd.batch_prefit_result("s413 #3")
```

#### 2.6.3.15 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:

```
wcd.batch_sample_parameter("s413 #3") = 13.56
```

#### 2.6.3.16 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:

```
wcd.batch_spectrum_filename("s413 #3",1) = "c:\test\s413_3_r"
```

#### 2.6.3.17 batch\_status (integer)

Inspect this value to find out if CODE 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.

```
wcd.batch_go
While wcd.batch_status <> 0
    Application.Wait Now + TimeSerial(0, 0, 3)
Wend
msgbox "Batch finished"
```

#### 2.6.3.18 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 wcd.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.

### 2.6.3.19 compute\_emissivity (spectrum, index : integer) (integer)

Returns or sets the 'emissivity computation flag' of the interface given by index in the spectrum object given by spectrum. A value of 1 indicates that the emissivity is to be computed, a value of 0 means the assigned fixed value is used (no computation is done).

Please note:

Spectra are counted 1, 2, 3, ...

Emissivity positions are counted according to the following scheme:

- 0 (external surface of the glazing)
- 1 (surface of the first pane pointing to the interior)
- 2 (surface of the second pane pointing to the exterior)
- 3 (surface of the second pane pointing to the interior)
- .
- .
- .

### 2.6.3.20 configuration\_file

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:

```
wcd.configuration_file = "f:\examples\test.wcd"
```

This statement causes the loading of the specified configuration file.

Reading this property returns the name of the configuration file that was loaded by CODE in the last 'load configuration' action.

### 2.6.3.21 database

Sets or returns the current database path.

### 2.6.3.22 current\_rating (string)

Returns the current rating as string.

### 2.6.3.23 delete\_layer(stack\_index,layer\_index) (integer)

Deletes the layer with index layer\_index in the stack specified by stack\_index. Returns 0 if successful and a number <>0 if a problem has occurred.

### 2.6.3.24 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.

### 2.6.3.25 df\_real\_part\_nm

Returns the value of the real part of the n-th dielectric function in the dielectric function list at the wavelength specified by the 'wavelength' property (in nm).

**2.6.3.26 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.

**2.6.3.27 df\_imag\_part\_nm**

Returns the value of the imaginary part of the n-th dielectric function in the dielectric function list at the wavelength specified by the 'wavelength' property (in nm).

**2.6.3.28 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) = wcd.dielectric_function_name(5)
```

Setting this property has no effect.

**2.6.3.29 emissivity (spectrum, index : integer) (float)**

Returns or sets the emissivity of the interface given by index in the spectrum object given by spectrum.

Please note:

Spectra are counted 1, 2, 3, ...

Emissivity positions are counted according to the following scheme:

- 0 (external surface of the glazing)
  - 1 (surface of the first pane pointing to the interior)
  - 2 (surface of the second pane pointing to the exterior)
  - 3 (surface of the second pane pointing to the interior)
- .
- .
- .

**2.6.3.30 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.

**2.6.3.31 experimental\_spectrum\_value**

Returns the value of the experimental spectrum that belongs to the n-th spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.

### 2.6.3.32 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: wcd.fft\_thickness(1,650,7000,2048,5,50,2.0)

### 2.6.3.33 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.

### 2.6.3.34 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

### 2.6.3.35 file\_object\_generation (real number)

Returns the object generation of the last configuration file that has been loaded into CODE.

### 2.6.3.36 fit\_deviation (real number)

Returns the current fit deviation.

```
Dim this_deviation as single
this_deviation = wcd.fit_deviation
```

Setting this property has no effect.

### 2.6.3.37 fit\_parameter\_name

Returns the name of the n-th fit parameter. The following VBA command fills a certain Excel cell with the name of the selected fit parameter:

```
Range("results").Offset(4 + i, 1) = wcd.fit_parameter_name(i)
```

### 2.6.3.38 fit\_parameter\_mode

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

### 2.6.3.39 fit\_parameter\_set

Setting the `fit_parameter_set` property you specify the file from which a fit parameter set is to be loaded.

Reading this property returns the name of the currently loaded fit parameter set.

VBA example:

```
wcd.fit_parameter_set = "f:\examples\thickness_fit.fps"
```

### 2.6.3.40 fit\_parameter\_value

Returns the present value of the fit parameter n. The fit parameters are labeled starting at 1.

```
Dim my_value as single
my_value = wcd.fit_parameter_value(12)
```

Set this property to pass a value to a fit parameter in CODE:

```
Dim my_value as single
my_value=3.4
wcd.fit_parameter_value(12)=my_value
```

### 2.6.3.41 fit\_parameter\_value\_max

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.

### 2.6.3.42 fit\_parameter\_value\_min

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.

### 2.6.3.43 fitting

Returns information on the fit activity of CODE.

The value of fitting is 0 if there is no fit running, fitting is equal to 1 during the fit.

VBA example:

```
if wcd.fitting=1 then ...
```

**2.6.3.44 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.

**2.6.3.45 found\_filename(i:integer) (string)**

This function returns the complete filename of the i-th selected file in the last searchfile action.

**2.6.3.46 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.

**2.6.3.47 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.

**2.6.3.48 grid\_test\_max (single)**

This property sets or gets the minimum value of a parameter variation on a fixed grid.

**2.6.3.49 grid\_test\_min (single)**

This property sets or gets the maximum value of a parameter variation on a fixed grid.

**2.6.3.50 grid\_test\_no\_points (integer)**

This property sets the number of points used in the parameter variation on a fixed grid.

**2.6.3.51 grid\_test\_para (integer)**

This property sets the fit parameter varied in a parameter variation on a fixed grid.

### 2.6.3.52 import\_measured\_plate\_spectra

Loads measured spectra into a layer of type 'Plate measured'.

You have to specify the following parameters:

stack\_index: Index of the layer stack (integer)  
layer\_index: Index of the layer in the stack (integer)  
T : filename (including path) of the transmittance spectrum  
R-Top: filename (including path) of the reflectance spectrum (top side)  
R-Bottom: filename (including path) of the reflectance spectrum (bottom side)  
file\_format: File format as integer number (see file format specification)  
option\_t : file option for transmittance data  
option\_r\_top : file option for reflectance data (top side)  
option\_r\_bottom : file option for reflectance data (bottom side)

If you read the spectra from individual files which contain one spectrum only, all file options are 1.  
If you read all spectra from a single file which contains all required data, you have to use the same filename for T, R-Top and R-Bottom and specify different values for option\_t, option\_r\_top and option\_r\_bottom, depending of the sequence of the data in the file.

### 2.6.3.53 import\_glass\_coating\_spectra

Loads measured spectra into a layer of type 'Glass coating'.

You have to specify the following parameters:

stack\_index: Index of the layer stack (integer)  
layer\_index: Index of the layer in the stack (integer)  
T : filename (including path) of the transmittance spectrum  
R-air: filename (including path) of the reflectance spectrum (air side)  
R-glass: filename (including path) of the reflectance spectrum (glass side)  
file\_format: File format as integer number (see file format specification)  
option\_t : file option for transmittance data  
option\_r\_air : file option for reflectance data (air side)  
option\_r\_glass : file option for reflectance data (glass side)

If you read the spectra from individual files which contain one spectrum only, all file options are 1.  
If you read all spectra from a single file which contains all required data, you have to use the same filename for T, R-Top and R-Bottom and specify different values for option\_t, option\_r\_top and option\_r\_bottom, depending of the sequence of the data in the file.

### 2.6.3.54 incidence\_angle

Returns the angle of incidence of the n-th spectrum. The following VBA command fills a certain Excel cell with the angle of incidence of the 2nd spectrum:

```
Range("results").Offset(2, 1) = wcd.incidence_angle(2)
```

Setting this property sets the value of the incidence angle of the n-th spectrum in the spectrum list.

### 2.6.3.55 insert\_layer\_definition (stack\_index, layer\_index : integer, layer\_definition : string) (integer)

Insert a new layer in the stack given by stack\_index. The new layer is added below the one specified by layer\_index. The layer is generated using the information given in layer\_definition.  
Returns 0 if no problem is found. Otherwise a value <> 0 is returned.

**2.6.3.56 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.

**2.6.3.57 layer\_material**

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, ...).

**2.6.3.58 layer\_thickness**

Sets 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.

**2.6.3.59 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, ...).

**2.6.3.60 main\_window\_height**

Height of the main window.

**2.6.3.61 main\_window\_left**

Screen coordinate (x) of the main window's upper left corner.

**2.6.3.62 main\_window\_top**

Screen coordinate (y) of the main window's upper left corner.

**2.6.3.63 main\_window\_width**

Width of the main window.

**2.6.3.64 material\_is\_in\_database**

Returns 1 if material\_name is an object in the current database, 0 if not.

**2.6.3.65 material\_is\_in\_material\_list**

Returns 1 if material\_name is an object in the current list of material, 0 if not.

**2.6.3.66 measurement(i:integer) (string)**

Setting this string to an appropriate filename will cause CODE 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.

### 2.6.3.67 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:

```
wcd.multidimensional_gridfit=1  
. . .  
check = wcd.multidimensional_gridfit
```

### 2.6.3.68 number\_of\_dielectric\_functions (integer)

Returns the number of dielectric functions in the present CODE configuration. The dielectric functions are counted from 1 to number\_of\_dielectric\_functions.

```
Dim np as integer  
np = wcd.number_of_dielectric_functions
```

Setting this property has no effect.

### 2.6.3.69 number\_of\_fit\_parameters

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 = wcd.number_of_fit_parameters
```

### 2.6.3.70 number\_of\_fp\_sets (integer)

Geben Sie hier den Text ein.

### 2.6.3.71 number\_of\_layers\_in\_stack (integer)

Returns the number of layers in the n-th layer stack (stacks are counted 1,2,3, ...).

### 2.6.3.72 number\_of\_layer\_stacks (integer)

Returns the number of layer stacks.

### 2.6.3.73 number\_of\_optical\_functions (integer)

Returns the number of items in the list of optical functions.

```
Dim nof as integer  
nof = wcd.number_of_optical_functions
```

**2.6.3.74 number\_of\_spectra (integer)**

Returns the number of spectra in the present configuration. The spectra are counted from 1 to number\_of\_spectra.

*Dim np as integer  
np = wcd.number\_of\_spectra*

**2.6.3.75 number\_of\_tec\_values (integer)**

This integer gives you the number of technical values in the present CODE configuration, i.e. number of rows in the list of integral quantities. Use this number to iterate through all values in a loop.

**2.6.3.76 object\_generation (real number)**

Returns the object generation of the CODE program.

**2.6.3.77 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 = wcd.optical_function("SiN(n at 400 nm)")`

Setting this property has no effect.

**2.6.3.78 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 = wcd.optical_function_by_index(3)`

Setting this property has no effect.

**2.6.3.79 optical\_function\_text\_by\_index (index : integer) (string)**

Returns the text of the optical function with the given index. Optical functions are counted 1, 2, ...

**2.6.3.80 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 CODE.

**2.6.3.81 rating\_threshold (level\_index:integer) (float)**

Returns or sets the rating threshold for the rating level specified by level\_index.

**2.6.3.82 rating\_description (level\_index:integer) (string)**

Returns or sets the string for the rating level specified by level\_index.

**2.6.3.83 reference\_spectrum(n:integer;format:integer):string**

Setting this string to an appropriate filename will cause CODE 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:

```
wcd.reference_spectrum(1, 4) = "c:/temp/dummy2.spc"
```

**2.6.3.84 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.

**2.6.3.85 sample\_description: string**

Read and write the sample description which is used in some views.

**2.6.3.86 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'.

**2.6.3.87 simulated\_spectrum\_value**

Returns the value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.

**2.6.3.88 simulated\_spectrum\_value\_nm**

Returns the value of the n-th simulated spectrum in the spectrum list at the wavelength specified by the 'wavelength' parameter (in nm).

### 2.6.3.89 simulation\_exportfile(n:integer;format:integer):string

Assigning a filename to simulation\_exportfile will cause CODE 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:

```
wcd.simulation_exportfile(1, 4) = "c:/temp/dummy2.spc"
```

### 2.6.3.90 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) = wcd.spectrum_name(2)
```

Setting this property has no effect.

### 2.6.3.91 spectrum\_weight(index) float

Returns the weight of the spectrum with the given index. Spectra are counted like 1, 2, 3, ...

The following VBA command fills a certain cell with the weight of the 2nd spectrum:

```
Range("results").Offset(2, 1) = wcd.spectrum_weight(2)
```

Setting this property changes the weight of the indexed spectrum.

### 2.6.3.92 tec\_value

Returns the present value of the i'th integral quantity.

### 2.6.3.93 tec\_value\_name

Returns the name of the i'th integral quantity.

### 2.6.3.94 tec\_value\_nickname (tec\_index : integer) (string)

Returns the nickname of the integral quantity with index tec\_index. The index is counted 1, 2, 3, ...

If the integral quantity has no nickname its name is given as result of this function.

### 2.6.3.95 tec\_value\_optimize (tec\_index : integer) (integer)

Returns or sets the status of the optimization\_flag of the technical value given by tec\_index. A value of 1 means optimization is on, 0 stand for optimization off.

### 2.6.3.96 tec\_value\_target\_value (tec\_index : integer) (integer)

Returns or sets the target value of the technical value given by tec\_index.

### 2.6.3.97 tec\_value\_weight (tec\_index : integer) (float)

Returns or sets the weight of the technical value given by tec\_index.

### 2.6.3.98 tolerance (real number)

Use this property to read and to set the tolerance which is used for terminating the downhill simplex fitting method.

### 2.6.3.99 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.

## 2.6.4 Example

### OLE example: Excel worksheet computes technical data for different layer thicknesses

This example shows a rather common task: You want to investigate how some technical values of a coating depend on the individual layer thicknesses. What you need to do is to compute for each set of thickness values all the wanted quantities and collect them in appropriate tables. You can do this quite easily with Excel which provides the required table structures, a macro language to express your wishes and the ability to control CODE as OLE automation server.

Before you start the Excel part of the work you have to setup a CODE configuration containing a complete model (optical constants, definition of the layer stack, definition of the spectra that are to be computed). Select those model parameters as fit parameters which shall be modified from Excel. In our example these are three layer thicknesses.

The Excel file containing the prepared macros (see below) looks like this (in the beginning):

<i>Window Coating Designer</i>	Configuration:	d:/d3theiss/win_coat/dw_test.wcd		
WCD names	Comments			

The cell named configuration\_file will be used by the macro that starts CODE to load the CODE configuration that we are going to work with. To start CODE and load the configuration you can use the CODE menu item **WCD|Start and load configuration** which has been added to the Excel sheet. After the configuration loading the Excel enters the names of the CODE fit parameters and the CODE technical values in the first column called **WCD names**. Now the sheet looks the following way:

<u>Window Coating Designer</u>	Configuration:	d:/d3theiss/win_coat/dw_test.wcd		
WCD names	Comments			
Layer no. 1 Oxide: Layer thickness				
Layer no. 2 Ag model: Layer thickness				
Layer no. 3 Oxide: Layer thickness				
Purity (R, Simulation)				
Dominant wavelength (R, Simulation)				
Light reflectance (D65) (R, Simulation)				
L* / D65 / 2° (R, Simulation)				
a* / D65 / 2° (R, Simulation)				
b* / D65 / 2° (R, Simulation)				
Solar direct reflectance (R, Simulation)				
Light transmittance (D65) (T, Simulation)				
Solar direct transmittance (T, Simulation)				

There are 3 fit parameters (thicknesses) and 9 technical values. In the second column you can enter comments that describe the quantities in your way. For example, you could place here internal terms for the layers or quantities of interest:

<u>Window Coating Designer</u>	Configuration:	d:/d3theiss/win_coat/dw_test.wcd		
WCD names	Comments			
Layer no. 1 Oxide: Layer thickness	Top			
Layer no. 2 Ag model: Layer thickness	Metal			
Layer no. 3 Oxide: Layer thickness	Bottom			
Purity (R, Simulation)	Pur			
Dominant wavelength (R, Simulation)	DW R			
Light reflectance (D65) (R, Simulation)	LR (front)			
L* / D65 / 2° (R, Simulation)	L* R			
a* / D65 / 2° (R, Simulation)	a* R			
b* / D65 / 2° (R, Simulation)	b* R			
Solar direct reflectance (R, Simulation)	Solar front			
Light transmittance (D65) (T, Simulation)	LT			
Solar direct transmittance (T, Simulation)	Solar through			

Now it's time to enter the values for the thicknesses that you want to include in your computation. Give a name (which can be just a number) for the set in the cells to the right of the term **Comments** and enter below the parameter values. The macro that performs the computation looks if there is a name in the third row of the worksheet. If so, it expects thickness values below that name. The macro continues to work through the columns from left to right until it cannot find a name anymore, i.e. if it reaches an empty name cell. For large series of values it would be reasonable to write a short VisualBasic macro that creates the input values for you. A prepared worksheet could be this one:

<u>Window Coating Designer</u>	Configuration:	d:/d3theiss/win_coat/dw_test.wcd		
WCD names	Comments	Set 1	Set 2	Set 3
Layer no. 1 Oxide: Layer thickness	Top	30.000	30.000	30.000
Layer no. 2 Ag model: Layer thickness	Metal	5.000	10.000	15.000
Layer no. 3 Oxide: Layer thickness	Bottom	30.000	30.000	30.000
Purity (R, Simulation)	Pur			
Dominant wavelength (R, Simulation)	DW R			
Light reflectance (D65) (R, Simulation)	LR (front)			
L* / D65 / 2° (R, Simulation)	L* R			
a* / D65 / 2° (R, Simulation)	a* R			
b* / D65 / 2° (R, Simulation)	b* R			
Solar direct reflectance (R, Simulation)	Solar front			
Light transmittance (D65) (T, Simulation)	LT			
Solar direct transmittance (T, Simulation)	Solar through			

Now execute the computation macro by the menu item **WCD|Compute data** and watch how Excel fills up the table:

<u>Window Coating Designer</u>	Configuration:	d:/d3theiss/win_coat/dw_test.wcd		
WCD names	Comments	Set 1	Set 2	Set 3
Layer no. 1 Oxide: Layer thickness	Top	30.000	30.000	30.000
Layer no. 2 Ag model: Layer thickness	Metal	5.000	10.000	15.000
Layer no. 3 Oxide: Layer thickness	Bottom	30.000	30.000	30.000
Purity (R, Simulation)	Pur	0.059	0.165	0.055
Dominant wavelength (R, Simulation)	DW R	490.952	485.919	-509.679
Light reflectance (D65) (R, Simulation)	LR (front)	0.185	0.081	0.042
L* / D65 / 2° (R, Simulation)	L* R	50.061	34.125	24.322
a* / D65 / 2° (R, Simulation)	a* R	-4.749	-5.920	7.262
b* / D65 / 2° (R, Simulation)	b* R	-1.854	-6.252	-2.093
Solar direct reflectance (R, Simulation)	Solar front	0.156	0.197	0.286
Light transmittance (D65) (T, Simulation)	LT	0.808	0.907	0.940
Solar direct transmittance (T, Simulation)	Solar through	0.832	0.785	0.694

The Visual Basic macros used in this example:

```
' The wcd object is the global object representing the Windows Coating Designer OLE server
Dim wcd As Object

' This macro creates the wcd object and loads the configuration file that is specified in the
' Excel worksheet "colors" (cell named "configuration_file")
' This configuration must have been created before for the layer stack that is going to be inspected
```

---

```
Sub start_wcd()
    'Create the wcd object
    Set wcd = CreateObject("code.colors")
```

```
'Load the configuration file
wcd.configuration_file = Range("colors!configuration_file").Value
'Show the WCD program on the screen
wcd.Show
row_offset = 2

'Get the number of fit parameters (i.e. layer thicknesses) from WCD
no_paras = wcd.number_of_fit_parameters
'Get the number of integral values (e.g. color coordinates) from the WCD list of integral values
no_tec_values = wcd.number_of_tec_values

'Enter the names of the fit parameters and integral quantities to be computed into the Excel
worksheet
'in column 1 (A)
For i = 1 To no_paras
    Range("colors!origin").Offset(row_offset + i, 0).Value = wcd.fit_parameter_name(i)
Next i
For i = 1 To no_tec_values
    Range("colors!origin").Offset(row_offset + no_paras + 1 + i, column_count).Value =
wcd.tec_value_name(i)
Next i

End Sub
```

---

```
'This macro deletes the OLE server from memory
Sub delete_wcd()
    wcd.prepare_shutdown
    Set wcd = Nothing
End Sub
```

---

```
'This macro displays the WCD main window if it was hidden
Sub show_wcd()
    wcd.Show
End Sub
```

---

```
'This macro hides the WCD main window
Sub hide_wcd()
    wcd.Hide
End Sub
```

---

```
'This macro computes the wanted integral quantities
Sub compute_values()
    Dim fitpara_count As Integer
    Dim tecdata_count As Integer
    Dim no_paras As Integer
    Dim no_tec_values As Integer

    row_offset = 2
```

```
' Get the number of fit parameters from WCD
no_paras = wcd.number_of_fit_parameters
' Get the number of integral quantities from WCD
no_tec_values = wcd.number_of_tec_values

column_count = 2
' Check if there is another set of thicknesses to process
While Range("colors!origin").Offset(row_offset, column_count).Value <> ""
    'If so take the values for the fit parameter from the Excel worksheet
    For i = 1 To no_paras
        'If the fit parameter is a layer thickness (fit parameter mode =2), convert the value in nanometers
        'used in the Excel sheet to microns which is the
        'unit expected by WCD
        If wcd.fit_parameter_mode(i) = 2 Then
            wcd.fit_parameter_value(i) = 0.001 * Range("colors!origin").Offset(row_offset + i,
            column_count).Value
        Else
            wcd.fit_parameter_value(i) = Range("colors!origin").Offset(row_offset + i, column_count).Value
        End If
    Next i
    'Recompute everything in WCD with the new set of parameters
    wcd.update_data
    ' Get from WCD the new values of the integral quantities
    For i = 1 To no_tec_values
        Range("colors!origin").Offset(row_offset + no_paras + 1 + i, column_count).Value =
        wcd.tec_value(i)
    Next i
    'Proceed to the next column
    column_count = column_count + 1
Wend

End Sub
```

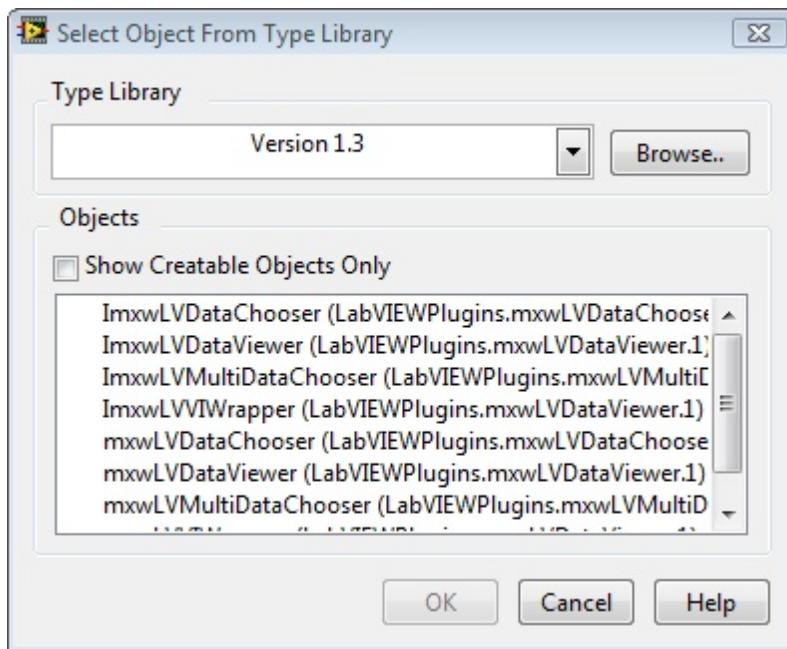
## 2.6.5 Using CODE in LabView

**You must have object generation 3.60 for the mechanism described below!**

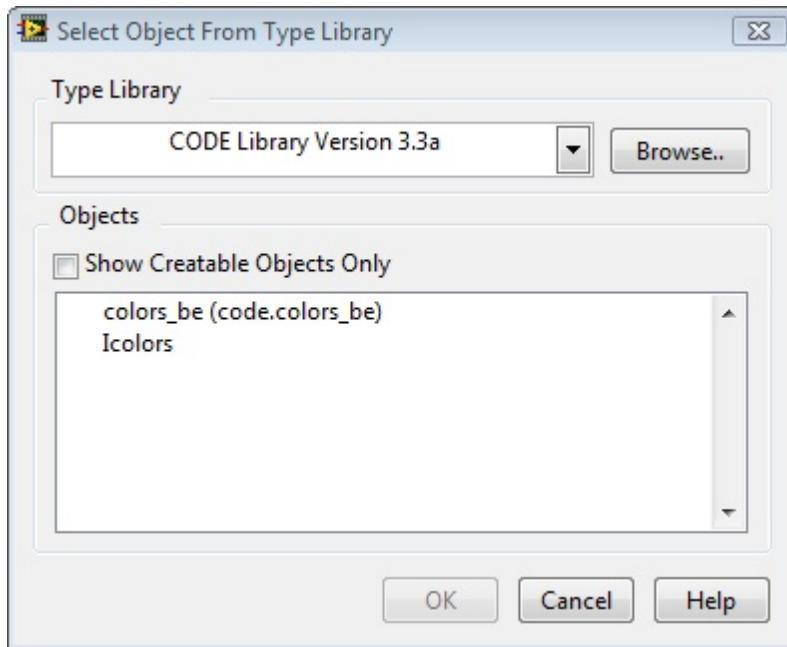
If you develop automation solutions with LabView you can access CODE in your LabView projects. A typical application would be a scanning spectrometer system, used to obtain one- or two-dimensional thickness profiles. If LabView is used to control the mechanical motion and the data acquisition of the spectrometer hardware, it would be advantageous to control CODE by LabView as well. After a measurement, LabView could pass the recorded spectra to CODE, start the thickness determination, and pick-up the results in order to generate appropriate operator displays.

Here is how you get access to CODE within LabView:

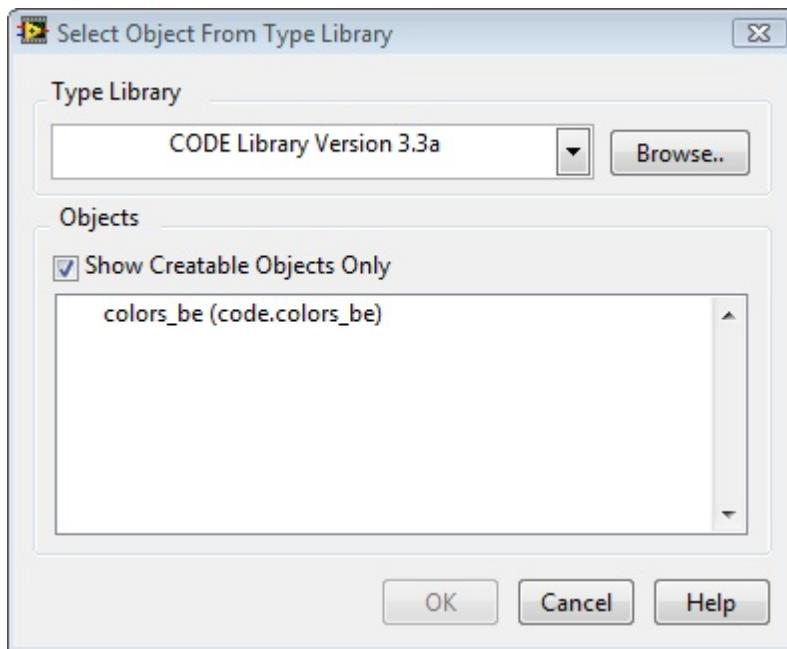
- Generate a control of type 'Automation refnum'
- Right-click on the control and select the command 'Select ActiveX class/Browse'
- A dialog for the selection of a type library opens



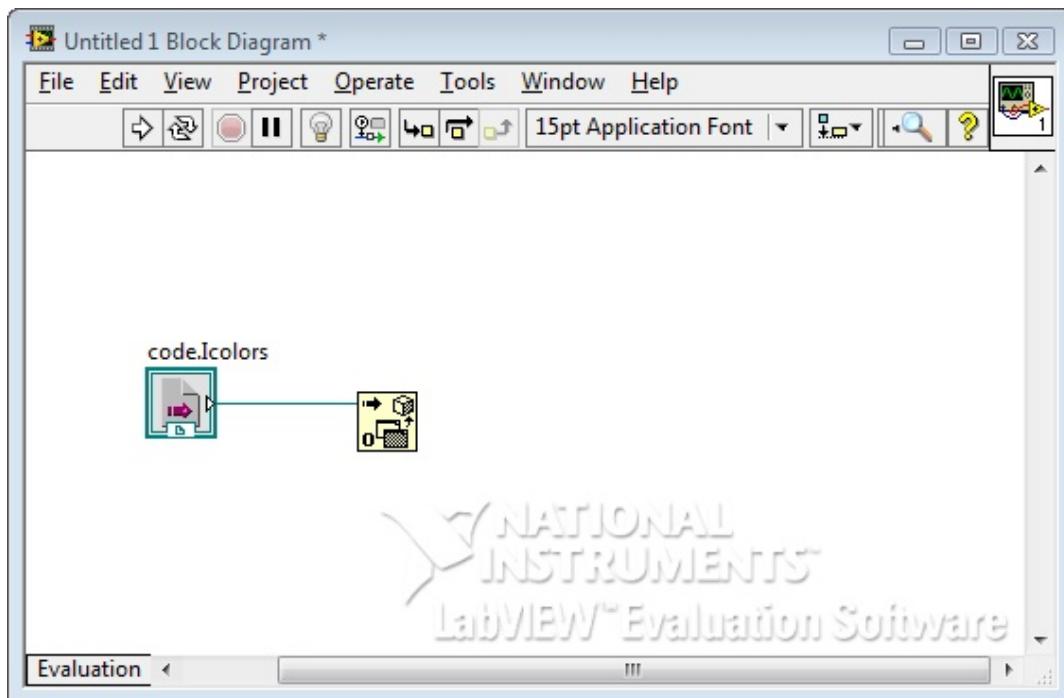
- Search for the type library 'CODE library' in the dropdown list
- If you do not find it click on the 'Browse' button to the right and select the CODE program file code.exe (in order to see the program file you may have to set the file type to 'All files (\*.\*)' in the file dialog)
- The dialog should now look like this:



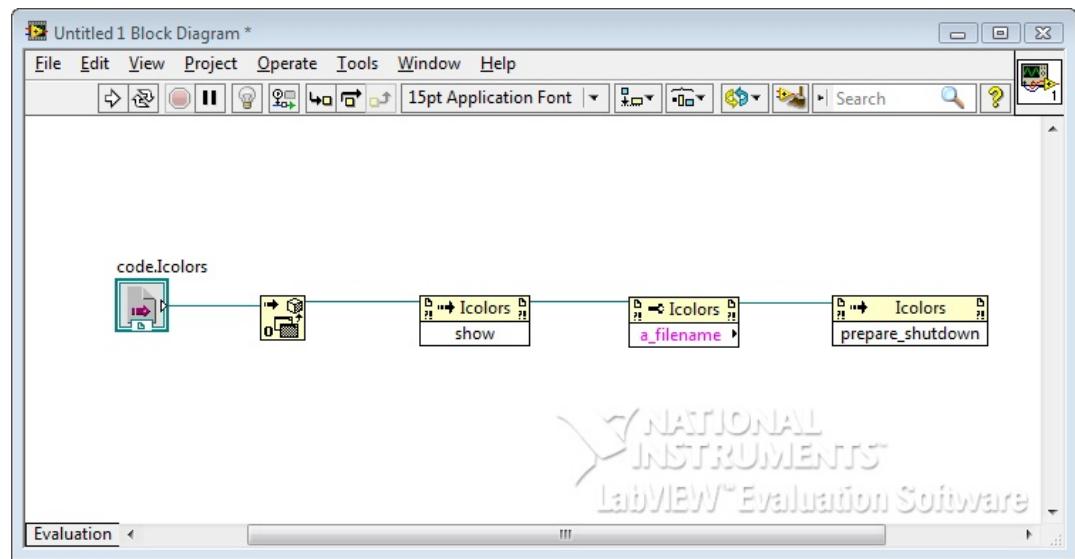
- Check the checkbox 'Show Creatable Objects Only' and select the only remaining line 'scoutole (scout.scoutole)':



- Then press OK to leave this dialog.
- In the LabView Block diagram, right-click the new object and select the command 'ActiveX palette' to generate a new object of type 'Automation open'. Connect the 2 objects like shown below:



- From now on you can use all OLE properties and procedures of CODE. In order to check if you can really access CODE commands generate a small test sequence: Show CODE on the screen ('show' command), execute a file selection dialog (by calling the property 'a\_filename') and finally call the routine 'prepare\_shutdown' to prepare the death of the CODE server.



Before you do any real work in LabView, please test if you can get this sequence to execute properly.

**CODE**

W. Theiss

Hard- and  
Software

**Part**

III

## 3 Background

### 3.1 Background information

This section gives background information (theory, formulas, tables) for some computed quantities.

At present you find information about

Color coordinates XYZ

Color coordinates xyz

Color coordinates L\*a\*b\*

Color coordinates Lab

Color-matching functions

Ra (color rendering index)

Light transmittance

Light reflectance

Solar direct transmittance

Solar direct reflectance

Spectrum product

Photocurrent

g (total solar energy transmittance)

Emissivity

Sheet resistance

U (Thermal resistance)

A spectrum

D65 spectrum

C spectrum

Solar spectral distribution

AM 1.5 spectrum

### 3.2 Color coordinates XYZ

The color coordinates X,Y,Z are computed according to DIN 5033.

The following steps are taken:

In case of reflectance or transmittance spectra the illuminant D65 with relative spectral distribution S (as given in the table below) is used to compute the intermediate quantity k:

$$k = \frac{100}{\sum_{\lambda=380nm}^{780nm} S_{\lambda} \bar{y}(\lambda) \Delta \lambda} \quad \bar{y} : \text{Color - matching function}$$

The tristimulus values X, Y and Z are calculated according to the following equations:

$$X = k \sum_{\substack{\lambda=380\text{nm} \\ 5\text{nm steps}}}^{780\text{nm}} S_\lambda \bar{x}(\lambda) \tau(\lambda) \Delta\lambda$$

$$Y = k \sum_{\substack{\lambda=380\text{nm} \\ 5\text{nm steps}}}^{780\text{nm}} S_\lambda \bar{y}(\lambda) \tau(\lambda) \Delta\lambda$$

$$Z = k \sum_{\substack{\lambda=380\text{nm} \\ 5\text{nm steps}}}^{780\text{nm}} S_\lambda \bar{z}(\lambda) \tau(\lambda) \Delta\lambda$$

$\bar{x}$  : Color - matching function

$\bar{y}$  : Color - matching function

$\bar{z}$  : Color - matching function

where  $\tau$  is the transmittance of the layer stack. The color-matching functions are given below.

The given expressions are for  $2^\circ$  observation. For  $10^\circ$  the corresponding values for the color-matching functions are taken. These are also given below.

### 3.3 Color coordinates x,y,z

The chromaticity coordinates x,y, and z are computed from X,Y, and Z using the relations

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

### 3.4 Color coordinates L\*a\*b\*

The color coordinates L\*,a\*,b\* (CIELAB) are computed according to DIN 5033. They are based on X,Y,Z (see above).

The following equations are used:

$$L^* = 116 \sqrt[3]{\frac{Y}{Y_N}} - 16$$

$$a^* = 500 \left[ \sqrt[3]{\frac{X}{X_N}} - \sqrt[3]{\frac{Y}{Y_N}} \right] \quad \text{where} \quad \begin{aligned} X_N &= 95.05 \\ Y_N &= 100.00 \quad \text{for D65 illuminant} \\ Z_N &= 108.90 \end{aligned}$$

$$b^* = 200 \left[ \sqrt[3]{\frac{Y}{Y_N}} - \sqrt[3]{\frac{Z}{Z_N}} \right]$$

If the arguments of the cubic roots are smaller than 0.008856 then the cubic root is replaced by the expression 7.787\*argument+16/116 .

### 3.5 Color coordinates Lab

The quantities L, a, and b are computed according to

$$L = 100 \left( \frac{Y}{Y_N} \right)^{1/2}$$

$$a = 175 \left( \frac{0.0102 * X_N}{(Y/Y_N)} \right)^{1/2} \cdot \left[ \left( \frac{X}{X_N} \right) - \left( \frac{Y}{Y_N} \right) \right] \quad \text{where} \quad \begin{aligned} X_N &= 95.05 \\ Y_N &= 100.00 \quad \text{for D65 illuminant} \\ Z_N &= 108.90 \end{aligned}$$

$$b = 70 \left( \frac{0.00847 * Z_N}{(Y/Y_N)} \right)^{1/2} \cdot \left[ \left( \frac{Y}{Y_N} \right) - \left( \frac{Z}{Z_N} \right) \right]$$

The coordinates X,Y, and Z are defined above.

### 3.6 Ra (color rendering index)

This quantity is computed according to standard EN 410. It requires the range of the spectrum to cover 380 ... 780 nm.

### 3.7 Light transmittance

The light transmittance is computed according to the following equation (DIN 67 504):

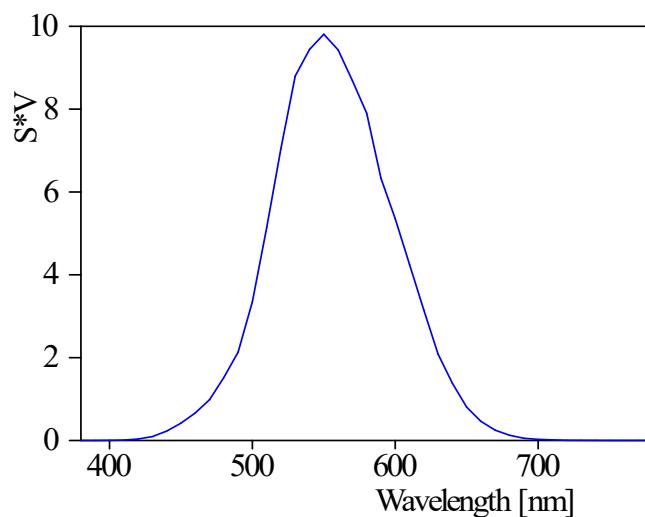
$$\tau = \frac{\sum_{\substack{\lambda=380\text{ nm} \\ 5\text{ nm steps}}}^{780\text{ nm}} S_\lambda \tau(\lambda) V(\lambda) \Delta\lambda}{\sum_{\substack{\lambda=380\text{ nm} \\ 5\text{ nm steps}}}^{780\text{ nm}} S_\lambda V(\lambda) \Delta\lambda}$$

$S_\lambda$ : relative spectral distribution of illuminant

$\tau(\lambda)$ : spectral transmittance of layer stack

$V(\lambda)$ : spectral luminous efficiency

The values of the product  $S^*V$  of the relative spectral distribution of the light source (here: D65) and the spectral luminous efficiency are these:



They are taken from the following table (also from DIN 67504):

Wavelength [nm]     $S^*V$

380 0.0000

390 0.0005

400 0.0030

410 0.0103

420 0.0352

430 0.0948

440 0.2274

450 0.4192

460 0.6663

470 0.9850

480 1.5189

490	2.1336
500	3.3491
510	5.1393
520	7.0523
530	8.7990
540	9.4427
550	9.8077
560	9.4306
570	8.6891
580	7.8994
590	6.3306
600	5.3542
610	4.2491
620	3.1502
630	2.0812
640	1.3810
650	0.8070
660	0.4612
670	0.2485
680	0.1255
690	0.0536
700	0.0276
710	0.0146
720	0.0057
730	0.0035
740	0.0021
750	0.0008
760	0.0001
770	0.0000
780	0.0000

For other illuminants than D65 the product  $S^*V$  is divided by the intensity of the D65 light source and multiplied by the intensity of the chosen illuminant.

### 3.8 Light reflectance

The light reflectance is computed according to the following equation (DIN 67 504):

$$\rho = \frac{\sum_{\lambda=380nm}^{780nm} S_\lambda \rho(\lambda) V(\lambda) \Delta\lambda}{\sum_{\lambda=380nm}^{780nm} S_\lambda V(\lambda) \Delta\lambda}$$

$S_\lambda$  : relative spectral distribution of D65 illuminant

$\rho(\lambda)$ : spectral reflectance of layer stack

$V(\lambda)$ : spectral luminous efficiency

The values of the product  $S^*V$  of the relative spectral distribution of the light source and the spectral luminous efficiency are the same as used for the computation of the light transmittance (see above).

### 3.9 Solar direct transmittance

The solar direct transmittance  $\tau_e$  is computed (according to DIN 67507, EN 410 or ISO9050:2003) as

$$\tau_e = \frac{\sum_{\lambda=300nm}^{2500nm} S_\lambda \tau(\lambda) \Delta\lambda}{\sum_{\lambda=300nm}^{2500nm} S_\lambda \Delta\lambda}$$

$S_\lambda$  : relative spectral distribution of solar radiation  
 $\tau(\lambda)$ : spectral transmittance of layer stack

The set of numbers  $S_\lambda \cdot \Delta\lambda$  (Din 67507) used for the integration is given below.

### 3.10 Solar direct reflectance

The solar direct reflectance  $\rho_e$  is computed (according to DIN 67507, EN 410 or ISO9050:2003) as

$$\rho_e = \frac{\sum_{\lambda=300nm}^{2500nm} S_\lambda \rho(\lambda) \Delta\lambda}{\sum_{\lambda=300nm}^{2500nm} S_\lambda \Delta\lambda}$$

$S_\lambda$  : relative spectral distribution of solar radiation  
 $\rho(\lambda)$ : spectral reflectance of layer stack

The set of numbers  $S_\lambda \cdot \Delta\lambda$  used for the integration is given below.

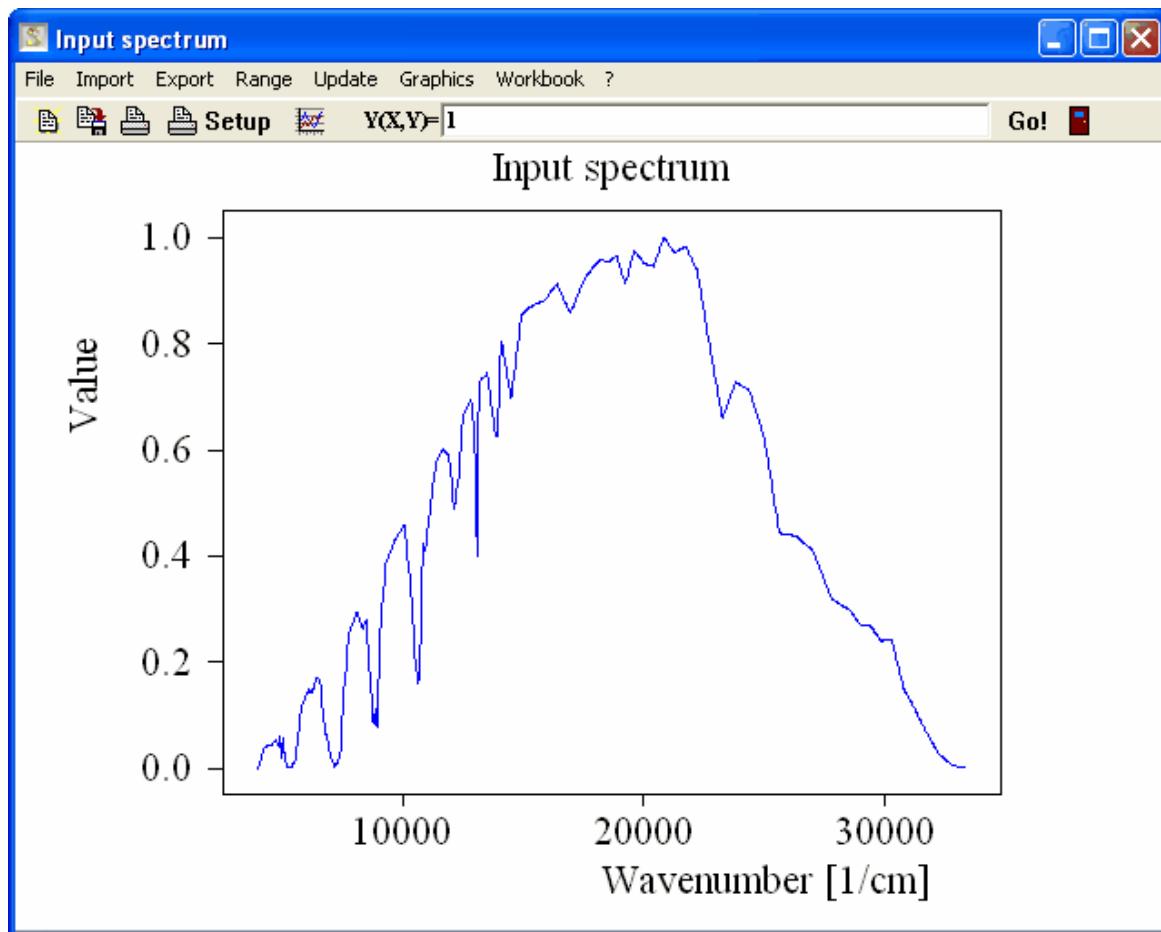
### 3.11 Spectrum product

This quantity is the integration over the product of the selected spectrum and a so-called 'Input spectrum'. The latter can be set in a window which is opened by the 'Edit' command of the 'Spectrum product' object. Here you can import the spectrum from a datafile.

The spectral range used for the integration is that of the input spectrum. The required values of the selected spectrum are computed by linear interpolation or extrapolation.

## 3.12 Photocurrent

Objects of type 'Photocurrent' compute the current density generated by absorption of light in a layer. You must assign a spectrum of type 'Charge carrier generation' to a photocurrent object to make it work correctly. Please see the 'SCOUT technical manual' for documentation of this spectrum type. When you edit a photocurrent object, you should set its unit to 'mA / cm<sup>2</sup>'. The **Edit** procedure will ask you to specify the total illumination power (in W/m<sup>2</sup>). Afterwards you have to set the spectral distribution of the illuminating radiation in the following window:



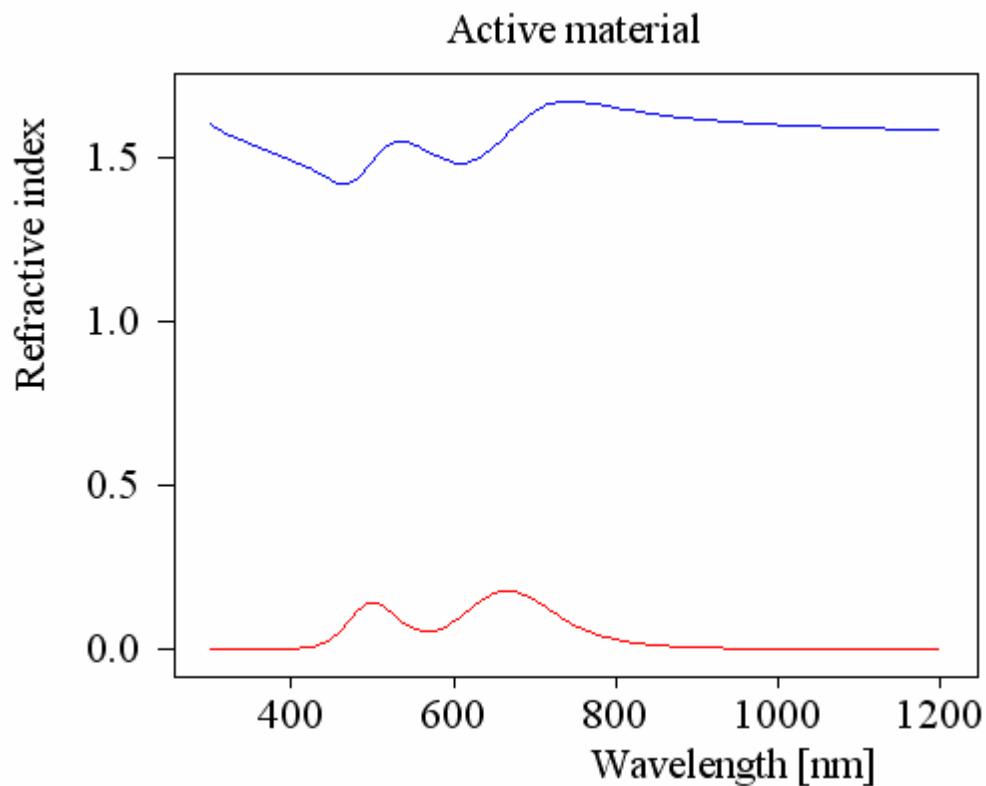
The example shows the normalized AM 1.5 spectrum. You do not have to care about normalization of the input spectrum - photocurrent objects normalize the distribution function before they use it for their computation.

Photocurrent objects compute for each spectral point the number of generated charge carriers and integrate over the whole input spectrum. The final number is the current density in mA / cm<sup>2</sup>.

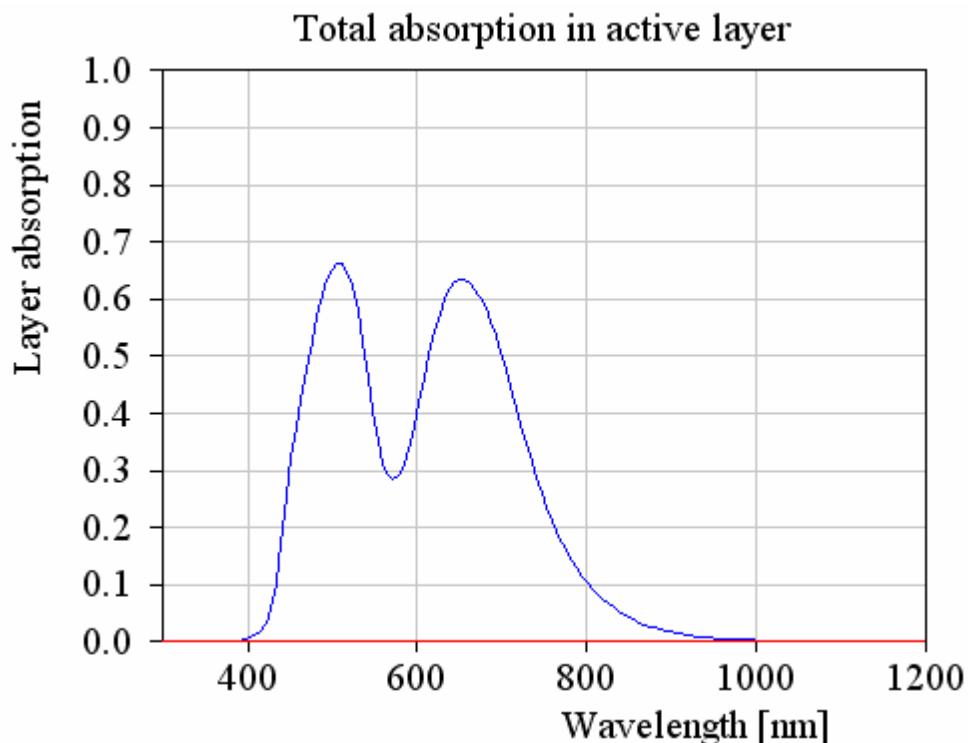
### Application example

The following example shows the optimization of a simple solar cell layer stack. The illuminating beam is incident from the top through a 4 mm glass pane. Underneath a transparent conductive layer (TCO) of several 100 nm thickness the active layer which is to convert sunlight to electrical current is placed. A aluminum layer at the bottom of the stack provides electrical contact on the backside of the active layer.

The active layer is equipped with a fictitious material which absorbs in the visible spectral range. Its complex refractive index shows two main absorption bands around 500 and 700 nm wavelength:



The light absorption of the active layer with 150 nm thickness (TCO thickness is 500 nm) is this:



Usually not all absorbed photons generate a charge carrier. The conversion efficiency may be much less than 1 and depend on details like the thickness of the active layer. In order to take into account a thickness dependent efficiency, the efficiency of the charge carrier generation spectrum is

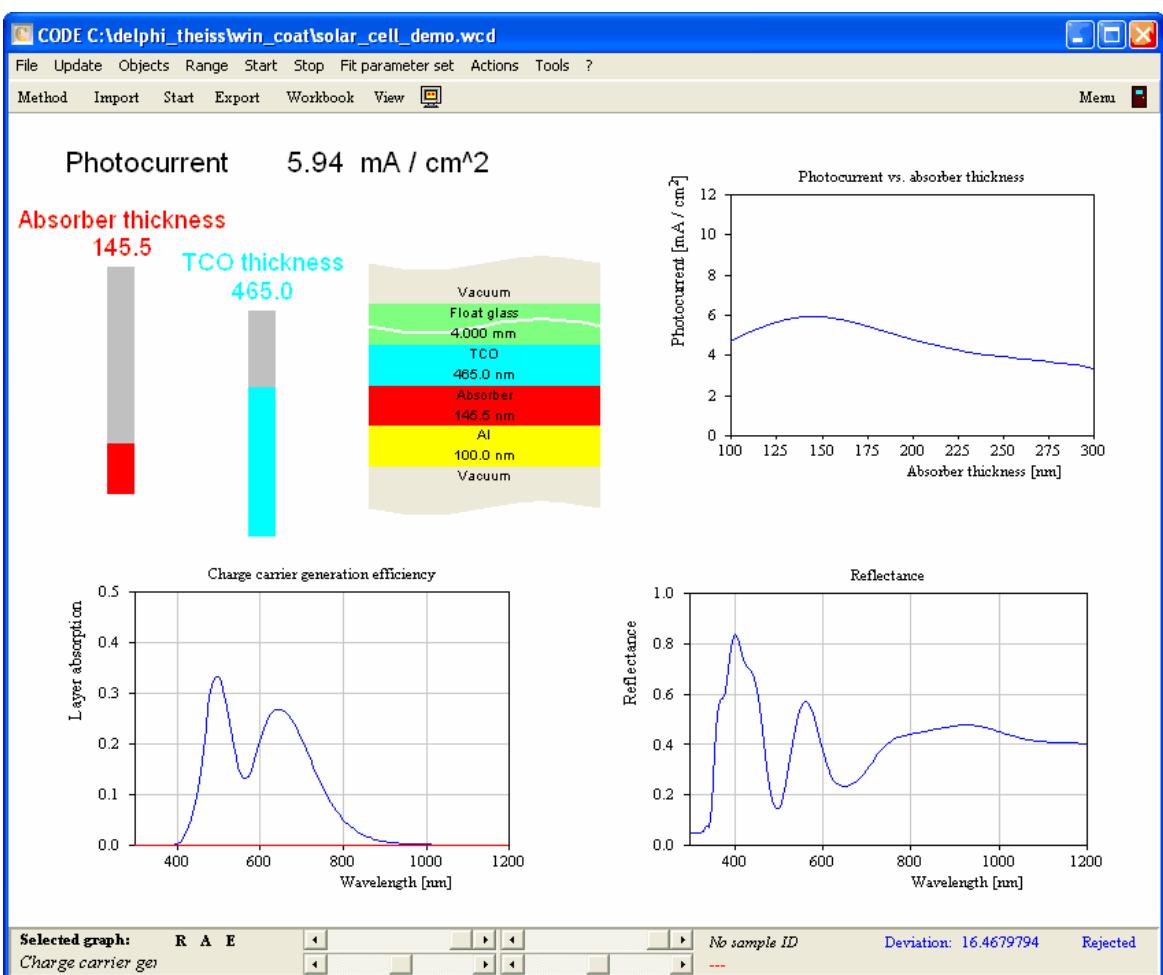
computed according to a user-defined formula. In this case the efficiency is simply a constant C1 where C1 is one of the possible fit parameters of the efficiency object.

With the help of a master parameter called 'Absorber\_thickness' the coupling of layer thickness and efficiency can be expressed in CODE. The fit parameters for the optimization of the photocurrent are the following:

Fit parameters						
<a href="#">File</a> <a href="#">New</a> <a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Delete all</a> <a href="#">Slider</a> <a href="#">Fit on grid</a> <a href="#">Limits from sliders</a> <a href="#">Update</a> ?						
	Value	Name	Variation	Low limit	High limit	Fac
1	145.5	Stack 1 Layer 4 Absorber: Layer thickness	Absorber_thickness	0.0	0.0	1
2	0.4575	Charge carrier generation efficiency:C1	0.7-Absorber_thickness/600	0.0000	0.0000	1
3	150.0	Absorber_thickness	Downhill simplex	50.0	500.0	1
4	500.0	Stack 1 Layer 3 TCO: Layer thickness	Downhill simplex	200.0	600.0	1

As shown above, the thickness of the layer is simply computed as Absorber\_thickness, whereas the efficiency linearly decreases with thickness following the relation  $0.7 \cdot \text{Absorber\_thickness}/600$ .

Setting a target value of  $10 \text{ mA/cm}^2$  for the current density in the list of integral quantities, the fit optimizes both the TCO and the active layer thickness with the following best result:



The upper right curve showing the relation of absorber thickness and photocurrent with fixed TCO thickness has been obtained using the 'Parameter variation' mechanism (see SCOUT manual) and a view object which displays workbook data (the results of the parameter variation).

### 3.13 g (total solar energy transmittance)

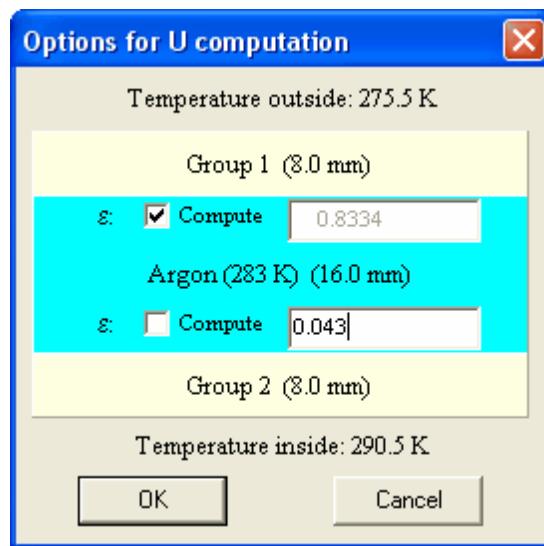
The total solar energy transmittance (or solar factor)  $g$  is computed according to DIN 67 507, EN 410, NFRC, ISO9050:2003 or ISO52022-3 (2017). CODE can handle an arbitrary number of glass panes in a glazing. A material object of type 'Gas' has to be used between two panes.

In order to compute  $g$  CODE has to use some physical properties of the filling gas. The values have been implemented in the material type 'Gas' which allows to define all kinds of gas mixtures. Please use this type of material for all gas fillings.

$g$  is the sum of the solar direct transmittance  $\tau_e$  and the secondary internal heat transfer factor  $q_i$ . The quantity  $q_i$  as well as the corresponding quantity  $q_e$  (the secondary external heat transfer factor) are also available as integral quantities.

$g$  can only be calculated for simulated spectra, not for measured spectra. CODE needs to compute the emissivities from parts of the layer stack. This is not possible in the case of experimental data. CODE needs to compute the emissivity of the coated glass panes in the far infrared (200 ... 1800 1/cm). Make sure that your optical model is valid in this spectra range. In most cases it is sufficient to compute the properties of the material silver that is used in the coatings in the far infrared. We recommend to use a model for the optical constants of silver and compute the model in the range 300 ... 50000 nm with at least 3000 data points. If there are other materials that might contribute to the emissivity (like ITO or thick metallic NiCr) then you should use the recommended wavelength range for these materials as well.

If you do not have optical constants in this spectral range you can still compute  $g$  if you know the emissivity of the involved surfaces. In this case, **Edit** the  $g$  quantity in the list of integral quantities. This dialog will open:



Note that the individual layers are not drawn in the dialog's sketch.

Using the checkboxes at the interfaces you can decide if CODE should compute the emissivity at this interface from the optical model. If the option is unchecked you can type in the emissivity to be used for this interface. If the option is checked CODE will indicate (in gray) the computed value of the emissivity.

The computation of  $g$ -values according to standards NFRC and ISO52022-3 involves an iterative routine that determines the temperature distribution of the glazing. As a consequence one can extract the temperature of the glass panes once the computation is finished. Other quantities like the

fraction of sunlight absorbed in each pane are available as well. These numbers can be extracted using optical functions or defining the wanted quantity as 'internal quantity' that is output as final result instead of the g-value.

#### **Internal quantities of g\_tot (ISO52022-3 (2017)):**

The following quantities are available as internal quantities. Make sure you correctly type their names:

- temperature pane 1
- temperature pane 2
- temperature pane 3
- temperature pane 4
- temperature pane 5
- absorbed power pane 1
- absorbed power pane 2
- absorbed power pane 3
- absorbed power pane 4
- absorbed power pane 5
- direct solar transmission
- u-value
- gain through thermal radiation
- gain through conduction and convection
- $q_i$

### **3.14 Emissivity**

The emissivity (both the normal and effective emissivity can be calculated in CODE) is computed following standard EN 673.

**Warning: Note that the spectrum must be known in the infrared region 5500 ... 50000 nm or 200 ... 1820 1/cm. Make sure that you have reliable optical constants in this range!**

### **3.15 Sheet resistance**

This function returns the sheet resistance of the layer stack that is assigned to the selected spectrum.

Note the following:

The sheet resistance is computed taking into account all layers which have a non-vanishing conductivity.

You can introduce electrical conductivity to a material in the following ways:

- Use an optical constant model and introduce a Drude model or an extended Drude model. This is the most flexible and recommended way.
- Assign a value for the electrical conductivity using the menu item **Property|Electrical conductivity**.

### **3.16 U (Thermal transmittance)**

This quantity is calculated according to standard EN 673. Please read the remarks about the computation rules of g above.

### 3.17 Sound insulation

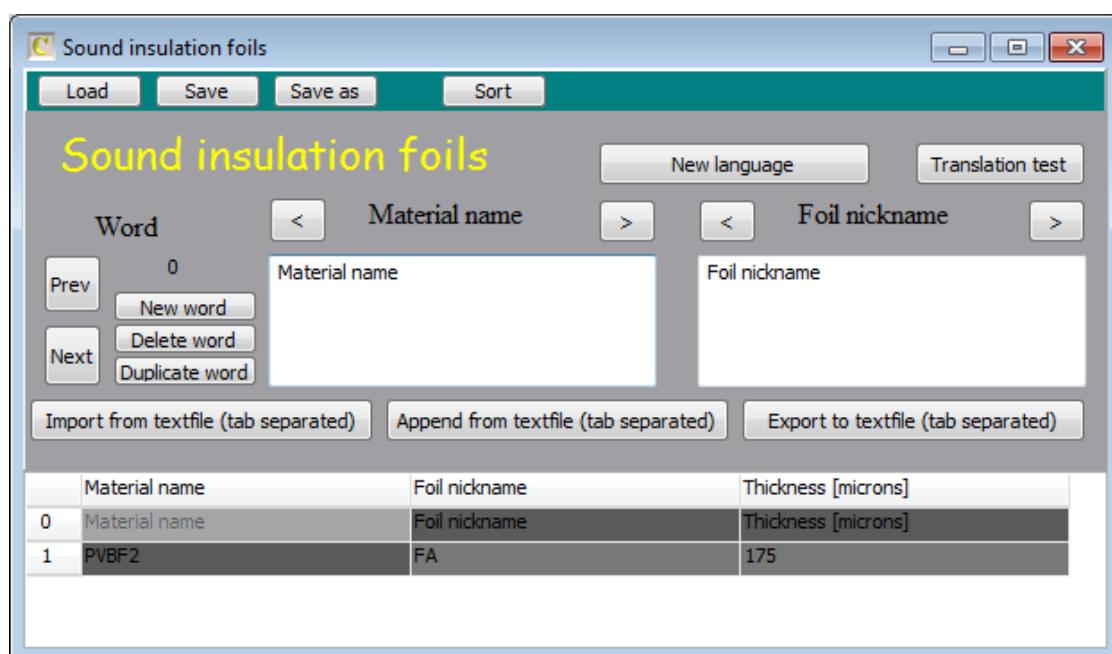
CODE can output sound insulation properties of the layer stack that belongs to the assigned spectrum. The given values are not computed based on a formula or a computational routine. Instead, they are looked up in a list of pre-defined layer stacks for which sound insulation data are available.

Sound insulation data are usually given for glazing compositions which are denoted like this:  
G4/FA/FA/G4/Ar16/G4

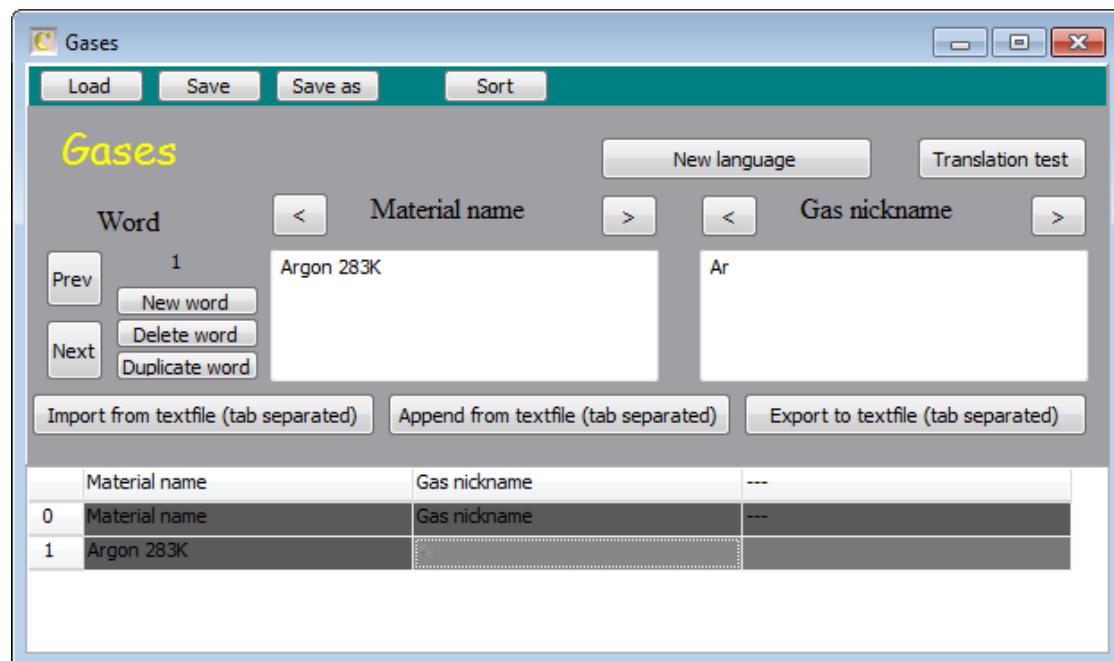
The notation means the following: We have 4 mm glass, then a foil named FA, another foil named FA, 4 mm glass, 16 mm Argon gas and another 4 mm glass pane.

In order to search for eventually existing sound data, CODE has to reduce the layer stack definition to the notation given above. This is achieved by the following sequence of actions:

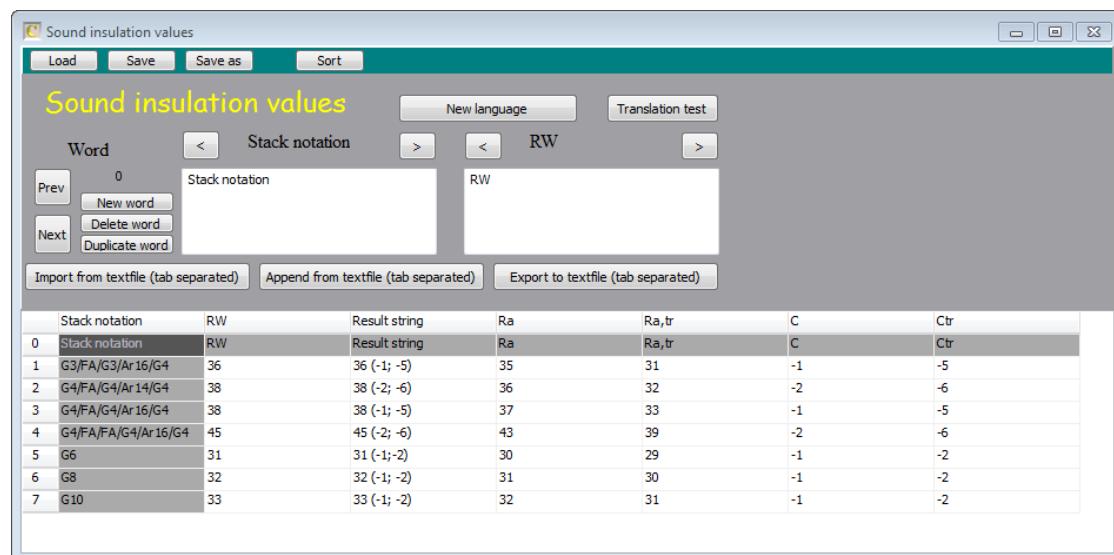
- Copy the layer stack and remove all thin films or simple layers which are treated coherently. It is assumed that the remaining layers are either glass, foil or gas.
- Foils are identified by the name of the material. All foils which may be used in glazing compositions must be entered in a list which is accessible in the list of integral quantities (menu command: 'Sound insulation/Edit foil list'). Such a list has 3 columns: The name of the material as it shows up in the optical constant database, the nickname which is used for the foil in the 'sound insulation notation', and the foil thickness in microns. If a layer material and the layer thickness agree with a foil definition, then the relevant foil nickname is added to the glazing notation. If the layer thickness is a multiple of the defined foil thickness then the nickname is added several times to the glazing notation.



- Now the layer stack is scanned for gas names. The assigned material names in the layer stack are compared to a list of gases. This list must contain the name of the gas used in the optical constant database (first column) and the nickname which is used in the notation of glazing compositions. It opens with the menu command 'Sound insulation/Edit gas list'.



- All layers which are not identified as foils or gas are assumed to be glass.
- Once the layer stack definition has been converted to the notation for the glazing component (like G4/FA/G4), CODE checks if for this composition there are sound insulation data available. This is done by going through the list which opens with the menu command 'Sound insulation/Edit stack list':



- If a composition match is found, the given RW value (or one of the other available values such as Ra, Ra,tr or C and Ctr) is taken as result. In addition, the text defined in the column 'Result string' is stored in the integral quantity. Both the value and the result text may be displayed in a view (see below).

Please be aware that the 3 lists (foils, gases, stacks) are stored as part of the configuration file. They are not available in other configurations.

Please do not change the order of the columns in the stack list.

The list of integral quantities will show the result value of a sound insulation object. If you display the sound insulation object in a view you can edit the view element and select the display of the result text.

## 3.18 NFRC support

Technical data computed according to procedures defined by the US organization NFRC (National Fenestration Rating Council) have been introduced with CODE version 3.79. CODE calculates and optimizes center-of-glass values only - no frame effects are taken into account.

### 3.18.1 NFRC VTc

Averaged transmittance in the visible spectral range (called 'Photopic' in NFRC documents), computed according to the procedure described in NFRC 300-2004. Transmittance data are required in the wavelength range 380 ... 780 nm. The averaging procedure can be used for reflectance data as well.

### 3.18.2 NFRC 300-2004 solar average

Averaged spectral values (used for reflectance as well as for transmittance) computed according to the procedure described in NFRC 300-2004. The calculation is based on the spectral distribution of solar radiation in ISO 9845-1. The required wavelength range for input data is 300 ... 2500 nm.

### 3.18.3 Tdw-ISO

This quantity is a weighted UV-Vis transmission (wavelength range: 300 ... 500 nm) which tries to quantify radiation damage to human skin. Its computation is described in 'Window 4.0 - Documentation of Calculation Procedures', LBL-33943 (1993), page 29.

### 3.18.4 Tuv

This quantity is a weighted UV transmission (wavelength range: 300 ... 390 nm) which tries to quantify radiation damage to human skin. Its computation is described in 'Window 4.0 - Documentation of Calculation Procedures', LBL-33943 (1993), page 29.

### 3.18.5 NFRC U-value (winter)

The NFRC center-of-glass U-value is computed following the procedure described in the document "WINDOW 4.0: Documentation of Calculation Procedures" (July 1993). The method is based on physical principles and experimental results concerning heat transfer through a glazing system. In an iterative process the temperature distribution in the glazing is obtained, taking into account temperature dependent gas properties in between the panes. Gas properties are not taken from the document mentioned above but from the standard ISO 15099.

The expressions for temperature dependent gas properties have been implemented in objects of type 'Gas' in the list of materials (starting with version 3.79). These objects allow the definition of gas mixtures and re-compute all required gas properties at any given temperature. Make sure that you use objects of this type as filling gas in your layer stacks.

NFRC U-values in CODE are computed using the following environmental conditions (NFRC 100-2010 Winter) :

Room temperature: 21°

Outside temperature: -18°  
 Glazing oriented windward, wind speed: 5.5 m/s  
 Sky temperature: -18°  
 Sky emissivity: 1.0  
 Incident solar power: 0 W/m<sup>2</sup>

#### Extracting additional information using optical functions

Besides the U-Value you can extract additional values that are obtained during the computation, in particular the obtained pane temperatures. You can do so using defining appropriate optical functions in the list of 'Optical functions'. Follow the examples below:

The function 'NFRC U-value (R, simulation, temperature pane 1)' returns the temperature of pane 1 where panes are counted 1, 2, 3, ... from the exterior to the interior.  
 The function 'NFRC U-value (R, simulation, temperature pane 2)' returns the temperature of pane 2 and so forth up to pane 5.

### 3.18.6 NFRC U-value (summer)

Computed like the quantity NFRC U-value (winter) but applying different environmental conditions (NFRC 100-2010 Summer) :

Room temperature: 24°  
 Outside temperature: 32°  
 Glazing oriented windward, wind speed: 2.75 m/s  
 Sky temperature: 32°  
 Sky emissivity: 1.0  
 Incident solar power: 0 W/m<sup>2</sup>

### 3.18.7 NFRC SHGC

The secondary heat gain coefficient SHGC is computed following the same procedures as the NFRC U-value. With solar power incident on the glazing, the temperature profile is computed and the total transmission of solar energy into the room is calculated. In addition to the directly transmitted radiation sun light is absorbed by the glazing and converted into heat. The fraction of heat entering the room is computed and added to the directly transmitted portion in order to obtain the final SHGC value.

Due to the temperature dependent gas properties, the computational scheme is complex enough. In order to avoid additional difficulties it is assumed that the absorption of solar radiation is evenly distributed across the depth of the panes - this is not a good approximation in the case of absorptive coatings.

Environmental conditions used in the computation of NFRC SHGC values:

Room temperature: 24°  
 Outside temperature: 32°  
 Glazing oriented windward, wind speed: 2.75 m/s  
 Sky temperature: 32°  
 Sky emissivity: 1.0  
 Incident solar power: 783 W/m<sup>2</sup>

#### Extracting additional information using optical functions

Besides the U-Value you can extract additional values that are obtained during the computation. You can do so using defining appropriate optical functions in the list of 'Optical functions'. Follow the examples below:

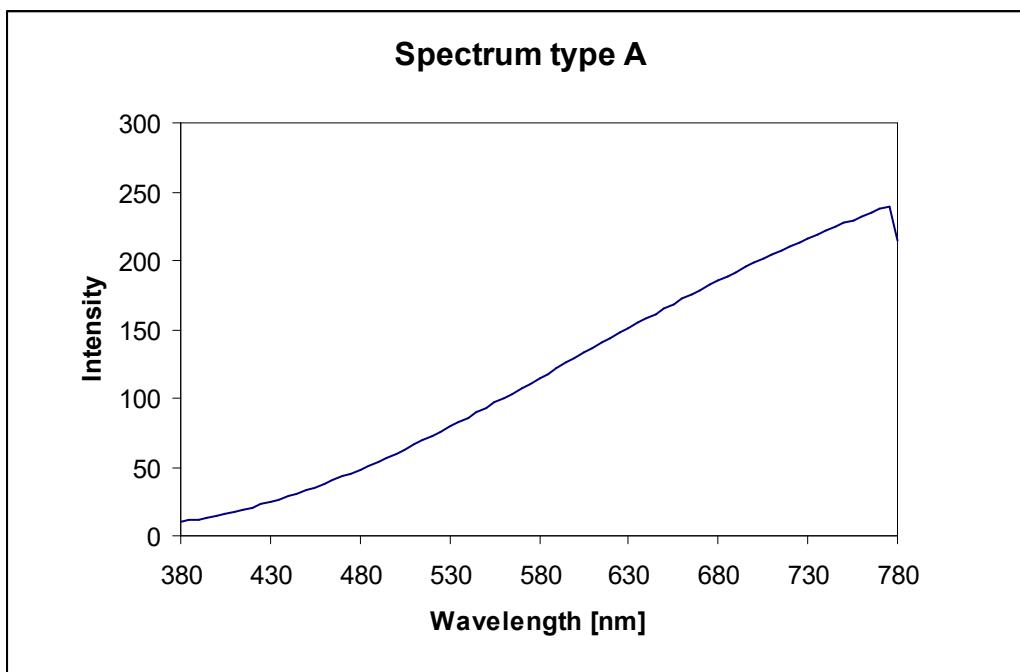
The function 'NFRC SHGC (R, simulation, temperature pane 1)' returns the temperature of pane 1 where panes are counted 1, 2, 3, ... from the exterior to the interior.

The function 'NFRC SHGC (R, simulation, temperature pane 2)' returns the temperature of pane 2 and so forth up to pane 5.

The function 'NFRC SHGC (R, simulation, absorbed power pane 1)' returns the absorbed solar power in W/m<sup>2</sup> of pane 1. Use similar expressions for panes 2 to 5.

### 3.19 A spectrum

The A spectrum used in some calculations is this:



Here is a table of the values:

Wavelength[nm]	Intensity
----------------	-----------

380	9.8
-----	-----

385	10.9
-----	------

390	12.1
-----	------

395	13.4
-----	------

400	14.7
-----	------

405	16.1
-----	------

410	17.7
-----	------

415	19.3
-----	------

420	21.0
-----	------

425	22.8
-----	------

430	24.7
-----	------

435	26.6
-----	------

440	28.7
-----	------

445	30.9
-----	------

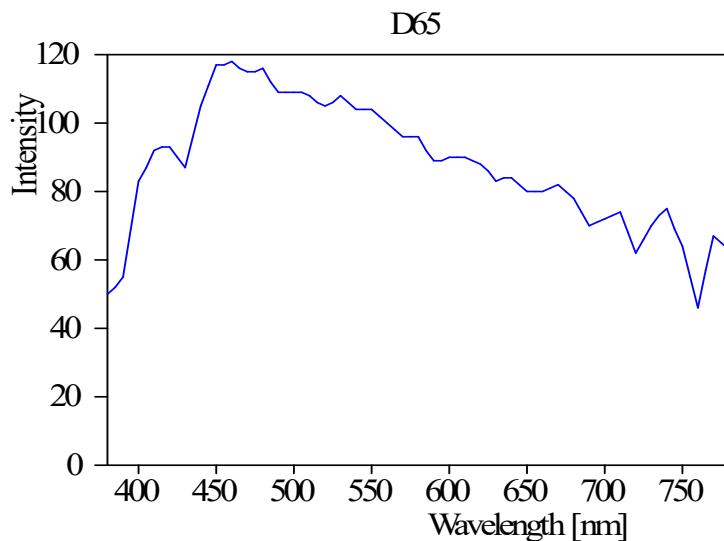
450	33.1
-----	------

455	35.4
460	37.8
465	40.3
470	42.9
475	45.5
480	48.2
485	51.0
490	53.9
495	56.8
500	59.9
505	62.9
510	66.1
515	69.3
520	72.5
525	75.8
530	79.1
535	82.5
540	85.9
545	89.4
550	92.9
555	96.4
560	100.0
565	103.6
570	107.2
575	110.8
580	114.4
585	118.1
590	121.7
595	125.4
600	129.0
605	132.7
610	136.4
615	140.0
620	143.6
625	147.2
630	150.8
635	154.4
640	158.0
645	161.5
650	165.0
655	168.5
660	172.0
665	175.4
670	178.8
675	182.1
680	185.4
685	188.7
690	191.9
695	195.1
700	198.3
705	201.4
710	204.4
715	207.4
720	210.4
725	213.3
730	216.1
735	218.9

740	221.7
745	224.4
750	227.0
755	229.6
760	232.1
765	234.6
770	237.0
775	239.4
780	214.7

### 3.20 D65 spectrum

The D65 spectrum used in some calculations is this:



Here is a table of the values:

Wavelength[nm]	Intensity
----------------	-----------

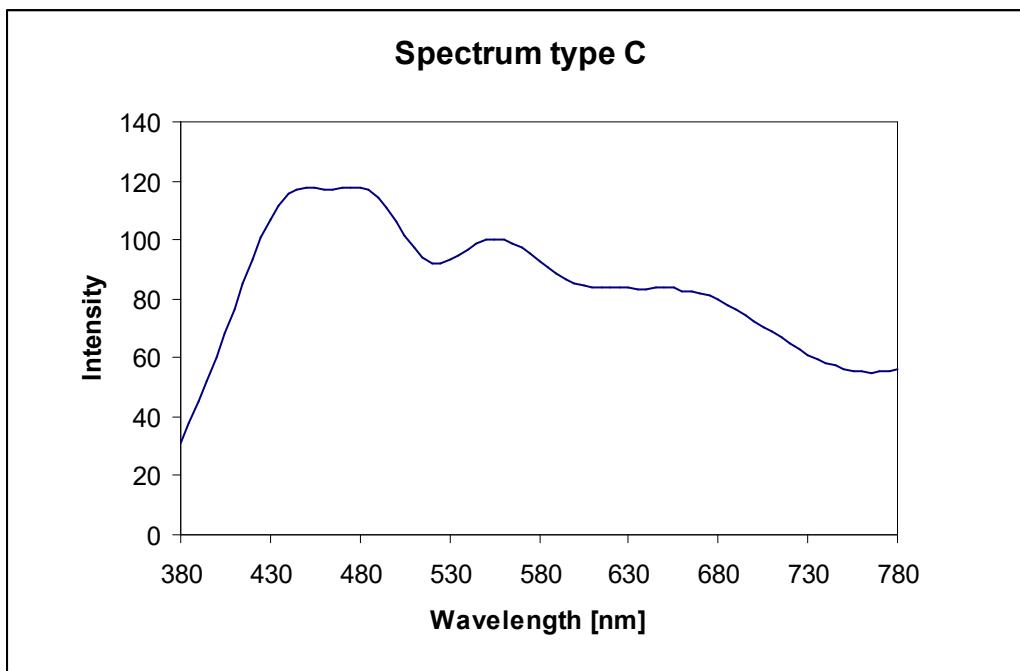
380	50.0
385	52.3
390	54.6
395	68.7
400	82.8
405	87.1
410	91.5
415	92.5
420	93.4
425	90.1
430	86.7
435	95.8
440	104.9
445	110.9
450	117.0
455	117.4
460	117.8
465	116.3

470	114.9
475	115.4
480	115.9
485	112.4
490	108.8
495	109.1
500	109.4
505	108.6
510	107.8
515	106.3
520	104.8
525	106.2
530	107.7
535	106.0
540	104.4
545	104.2
550	104.0
555	102.0
560	100.0
565	98.2
570	96.3
575	96.1
580	95.8
585	92.2
590	88.7
595	89.3
600	90.0
605	89.8
610	89.6
615	88.6
620	87.7
625	85.5
630	83.3
635	83.5
640	83.7
645	81.9
650	80.0
655	80.1
660	80.2
665	81.2
670	82.3
675	80.3
680	78.3
685	74.0
690	69.7
695	70.7
700	71.6
705	73.0
710	74.3
715	68.0
720	61.6
725	65.7
730	69.9
735	72.5
740	75.1
745	69.3
750	63.6

755	55.0
760	46.4
765	56.6
770	66.8
775	65.1
780	63.4

### 3.21 C spectrum

The C spectrum used in some calculations is this:



Here is a table of the values:

Wavelength[nm] Intensity

380	31.3
385	37.9
390	45.0
395	52.4
400	60.1
405	68.2
410	76.5
415	85.0
420	93.2
425	100.5
430	106.7
435	111.8
440	115.4
445	117.2
450	117.8
455	117.4

460	116.9
465	117.1
470	117.6
475	117.8
480	117.7
485	116.7
490	114.6
495	111.0
500	106.5
505	101.6
510	97.2
515	93.8
520	92.0
525	91.9
530	93.1
535	94.9
540	97.0
545	98.7
550	99.9
555	100.3
560	100.0
565	98.9
570	97.2
575	95.1
580	92.9
585	90.6
590	88.5
595	86.6
600	85.2
605	84.4
610	83.9
615	83.8
620	83.7
625	83.6
630	83.6
635	83.4
640	83.4
645	83.6
650	83.8
655	83.8
660	82.5
665	82.8
670	82.0
675	81.0
680	79.8
685	78.1
690	76.2
695	74.3
700	72.5
705	70.6
710	68.8
715	66.9
720	64.9
725	63.0
730	61.2
735	59.6
740	58.4

745	57.2
750	56.2
755	55.6
760	55.2
765	55.1
770	55.3
775	55.6
780	56.1

## 3.22 User-defined illuminants for color computation

The list of integral quantities offers the menu item **File/Options/Import additional illuminants for color computations**. Execute this command if you need to compute color values for illuminants other than the default ones A, D65 and C. You are asked to select a file which must be a text file with columns separated by semicolons. The first column contains 81 wavelengths from 380 to 780 nm in 5 nm steps, beginning in row 2. Starting with the second column, the file contains the name of the illuminant in the top row, following by 81 intensity values in the rows below. The table below shows the demo file illuminants.csv as generated by Excel.

```
;D50;D55;D75;F1;F2;F3;F4;F5;F6;F7;F8;F9;F10;F11;F12
380;24.488;32.584;66.703;1.87;1.18;0.82;0.57;1.87;1.05;2.56;1.21;0.9;1.11;0.91;0.96
385;27.179;35.335;68.333;2.36;1.48;1.02;0.7;2.35;1.31;3.18;1.5;1.12;0.63;0.63;0.64
390;29.871;38.087;69.963;2.94;1.84;1.26;0.87;2.92;1.63;3.84;1.81;1.36;0.62;0.46;0.4
5
395;39.589;49.518;85.946;3.47;2.15;1.44;0.98;3.45;1.9;4.53;2.13;1.6;0.57;0.37;0.33
400;49.308;60.949;101.929;5.17;3.44;2.57;2.01;5.1;3.11;6.15;3.17;2.59;1.48;1.29;1.1
9
405;52.91;64.751;106.911;19.49;15.69;14.36;13.75;18.91;14.8;19.37;13.08;12.8;12.16;
12.68;12.48
410;56.513;68.554;111.894;6.13;3.85;2.7;1.95;6;3.43;7.37;3.83;3.05;2.12;1.59;1.12
415;58.273;70.065;112.346;6.24;3.74;2.45;1.59;6.11;3.3;7.05;3.45;2.56;2.7;1.79;0.94
420;60.034;71.577;112.798;7.01;4.19;2.73;1.76;6.85;3.68;7.71;3.86;2.86;3.74;2.46;1.
08
425;58.926;69.746;107.945;7.79;4.62;3;1.93;7.58;4.07;8.41;4.42;3.3;5.14;3.33;1.37
430;57.818;67.914;103.092;8.56;5.06;3.28;2.1;8.31;4.45;9.15;5.09;3.82;6.75;4.49;1.7
8
435;66.321;76.76;112.145;43.67;34.98;31.85;30.28;40.76;32.61;44.14;34.1;32.62;34.39
;33.94;29.05
440;74.825;85.605;121.198;16.94;11.81;9.47;8.03;16.06;10.74;17.52;12.42;10.77;14.86
;12.13;7.9
445;81.036;91.799;127.104;10.72;6.27;4.02;2.55;10.32;5.48;11.35;7.68;5.84;10.4;6.95
;2.65
450;87.247;97.993;133.01;11.35;6.63;4.25;2.7;10.91;5.78;12;8.6;6.57;10.76;7.19;2.71
455;88.93;99.228;132.682;11.89;6.93;4.44;2.82;11.4;6.03;12.58;9.46;7.25;10.67;7.12;
2.65
460;90.612;100.463;132.355;12.37;7.19;4.59;2.91;11.83;6.25;13.08;10.24;7.86;10.11;6
.72;2.49
465;90.99;100.188;129.838;12.75;7.4;4.72;2.99;12.17;6.41;13.45;10.84;8.35;9.27;6.13
;2.33
470;91.368;99.913;127.322;13;7.54;4.8;3.04;12.4;6.52;13.71;11.33;8.75;8.29;5.46;2.1
475;93.238;101.326;127.061;13.15;7.62;4.86;3.08;12.54;6.58;13.88;11.71;9.06;7.29;4.
79;1.91
480;95.109;102.739;126.8;13.23;7.65;4.87;3.09;12.58;6.59;13.95;11.98;9.31;7.91;5.66
;3.01
485;93.536;100.409;122.291;13.17;7.62;4.85;3.09;12.52;6.56;13.93;12.17;9.48;16.64;1
4.29;10.83
490;91.963;98.078;117.783;13.13;7.62;4.88;3.14;12.47;6.56;13.82;12.28;9.61;16.73;14
.96;11.88
```

495;93.843;99.379;117.186;12.85;7.45;4.77;3.06;12.2;6.42;13.64;12.32;9.68;10.44;8.9  
7;6.88  
500;95.724;100.68;116.589;12.52;7.28;4.67;3;11.89;6.28;13.43;12.35;9.74;5.94;4.72;3  
.43  
505;96.169;100.688;115.146;12.2;7.15;4.62;2.98;11.61;6.2;13.25;12.44;9.88;3.34;2.33  
.1.49  
510;96.613;100.695;113.702;11.83;7.05;4.62;3.01;11.33;6.19;13.08;12.55;10.04;2.35;1  
.47;0.92  
515;96.871;100.341;111.181;11.5;7.04;4.73;3.14;11.1;6.3;12.93;12.68;10.26;1.88;1.1;  
0.71  
520;97.129;99.987;108.659;11.22;7.16;4.99;3.41;10.96;6.6;12.78;12.77;10.48;1.59;0.8  
9;0.6  
525;99.614;102.098;109.552;11.05;7.47;5.48;3.9;10.97;7.12;12.6;12.72;10.63;1.47;0.8  
3;0.63  
530;102.099;104.21;110.445;11.03;8.04;6.25;4.69;11.16;7.94;12.44;12.6;10.76;1.8;1.1  
8;1.1  
535;101.427;103.156;108.367;11.18;8.88;7.34;5.81;11.54;9.07;12.33;12.43;10.96;5.71;  
4.9;4.56  
540;100.755;102.102;106.289;11.53;10.01;8.78;7.32;12.12;10.49;12.26;12.22;11.18;40.  
98;39.59;34.4  
545;101.536;102.535;105.596;27.74;24.88;23.82;22.59;27.78;25.22;29.52;28.96;27.71;7  
3.69;72.84;65.4  
550;102.317;102.968;104.904;17.05;16.64;16.14;15.11;17.73;17.46;17.05;16.51;16.29;3  
3.61;32.61;29.48  
555;101.159;101.484;102.452;13.55;14.59;14.59;13.88;14.47;15.63;12.44;11.79;12.28;8  
.24;7.52;7.16  
560;100;100;100;14.33;16.16;16.63;16.33;15.2;17.22;12.58;11.76;12.74;3.38;2.83;3.08  
565;98.868;98.608;97.808;15.01;17.56;18.49;18.68;15.77;18.53;12.72;11.77;13.21;2.47  
;1.96;2.47  
570;97.735;97.216;95.616;15.52;18.62;19.95;20.64;16.1;19.43;12.83;11.84;13.65;2.14;  
1.67;2.27  
575;98.327;97.482;94.914;18.29;21.47;23.11;24.28;18.54;21.97;15.46;14.61;16.57;4.86  
;4.43;5.09  
580;98.918;97.749;94.213;19.55;22.79;24.69;26.26;19.5;23.01;16.75;16.11;18.14;11.45  
;11.28;11.96  
585;96.208;94.59;90.605;15.48;19.29;21.41;23.28;15.39;19.41;12.83;12.34;14.55;14.79  
;14.76;15.32  
590;93.499;91.432;86.997;14.91;18.66;20.85;22.94;14.64;18.56;12.67;12.53;14.65;12.1  
6;12.73;14.27  
595;95.593;92.926;87.112;14.15;17.73;19.93;22.14;13.72;17.42;12.45;12.72;14.66;8.97  
;9.74;11.86  
600;97.688;94.419;87.227;13.22;16.54;18.67;20.91;12.69;16.09;12.19;12.92;14.61;6.52  
;7.33;9.28  
605;98.478;94.78;86.684;12.19;15.21;17.22;19.43;11.57;14.64;11.89;13.12;14.5;8.81;9  
.72;12.31  
610;99.269;95.14;86.14;11.12;13.8;15.65;17.74;10.45;13.15;11.6;13.34;14.39;44.12;55  
.27;68.53  
615;99.155;94.68;84.861;10.03;12.36;14.04;16;9.35;11.68;11.35;13.61;14.4;34.55;42.5  
8;53.02  
620;99.042;94.22;83.581;8.95;10.95;12.45;14.42;8.29;10.25;11.12;13.87;14.47;12.09;1  
3.18;14.67  
625;97.382;92.334;81.164;7.96;9.65;10.95;12.56;7.32;8.96;10.95;14.07;14.62;12.15;13  
.16;14.38  
630;95.722;90.448;78.747;7.02;8.4;9.51;10.93;6.41;7.74;10.76;14.2;14.72;10.52;12.26  
;14.71  
635;97.29;91.389;78.587;6.2;7.32;8.27;9.52;5.63;6.69;10.42;14.16;14.55;4.43;5.11;6  
.46  
640;98.857;92.33;78.428;5.42;6.31;7.11;8.18;4.9;5.71;10.11;14.13;14.4;1.95;2.07;2.5  
7  
645;97.262;90.592;76.614;4.73;5.43;6.09;7.01;4.26;4.87;10.04;14.34;14.58;2.19;2.34;  
2.75  
650;95.667;88.854;74.801;4.15;4.68;5.22;6;3.72;4.16;10.02;14.5;14.88;3.19;3.58;4.18  
655;96.929;89.586;74.562;3.64;4.02;4.45;5.11;3.25;3.55;10.11;14.46;15.51;2.77;3.01;  
3.44  
660;98.19;90.317;74.324;3.2;3.45;3.8;4.36;2.83;3.02;9.87;14;15.47;2.29;2.48;2.81

665;100.597;92.133;74.873;2.81;2.96;3.23;3.69;2.49;2.57;8.65;12.58;13.2;2;2.14;2.42  
 670;103.003;93.95;75.422;2.47;2.55;2.75;3.13;2.19;2.2;7.27;10.99;10.57;1.52;1.54;1.  
 64  
 675;101.068;91.953;73.499;2.18;2.19;2.33;2.64;1.93;1.87;6.44;9.98;9.18;1.35;1.33;1.  
 36  
 680;99.133;89.956;71.576;1.93;1.89;1.99;2.24;1.71;1.6;5.83;9.22;8.25;1.47;1.46;1.49  
 685;93.257;84.817;67.714;1.72;1.64;1.7;1.91;1.52;1.37;5.41;8.62;7.57;1.79;1.94;2.14  
 690;87.381;79.677;63.852;1.67;1.53;1.55;1.7;1.43;1.29;5.04;8.07;7.03;1.74;2;2.34  
 695;89.492;81.258;64.464;1.43;1.27;1.27;1.39;1.26;1.05;4.57;7.39;6.35;1.02;1.2;1.42  
 700;91.604;82.84;65.076;1.29;1.1;1.09;1.18;1.13;0.91;4.12;6.71;5.72;1.14;1.35;1.61  
 705;92.246;83.842;66.573;1.19;0.99;0.96;1.03;1.05;0.81;3.77;6.16;5.25;3.32;4.1;5.04  
 710;92.889;84.844;68.07;1.08;0.88;0.83;0.88;0.96;0.71;3.46;5.63;4.8;4.49;5.58;6.98  
 715;84.872;77.539;62.256;0.96;0.76;0.71;0.74;0.85;0.61;3.08;5.03;4.29;2.05;2.51;3.1  
 9  
 720;76.854;70.235;56.443;0.88;0.68;0.62;0.64;0.78;0.54;2.73;4.46;3.8;0.49;0.57;0.71  
 725;81.683;74.768;60.343;0.81;0.61;0.54;0.54;0.72;0.48;2.47;4.02;3.43;0.24;0.27;0.3  
 730;86.511;79.301;64.242;0.77;0.56;0.49;0.49;0.68;0.44;2.25;3.66;3.12;0.21;0.23;0.2  
 6  
 735;89.546;82.147;66.697;0.75;0.54;0.46;0.46;0.67;0.43;2.06;3.36;2.86;0.21;0.21;0.2  
 3  
 740;92.58;84.993;69.151;0.73;0.51;0.43;0.42;0.65;0.4;1.9;3.09;2.64;0.24;0.24;0.28  
 745;85.405;78.437;63.89;0.68;0.47;0.39;0.37;0.61;0.37;1.75;2.85;2.43;0.24;0.24;0.28  
 750;78.23;71.88;58.629;0.69;0.47;0.39;0.37;0.62;0.38;1.62;2.65;2.26;0.21;0.2;0.21  
 755;67.961;62.337;50.623;0.64;0.43;0.35;0.33;0.59;0.35;1.54;2.51;2.14;0.17;0.24;0.1  
 7  
 760;57.692;52.793;42.617;0.68;0.46;0.38;0.35;0.62;0.39;1.45;2.37;2.02;0.21;0.32;0.2  
 1  
 765;70.307;64.36;51.985;0.69;0.47;0.39;0.36;0.64;0.41;1.32;2.15;1.83;0.22;0.26;0.19  
 770;82.923;75.927;61.352;0.61;0.4;0.33;0.31;0.55;0.33;1.17;1.89;1.61;0.17;0.16;0.15  
 775;80.599;73.872;59.838;0.52;0.33;0.28;0.26;0.47;0.26;0.99;1.61;1.38;0.12;0.12;0.1  
 780;78.274;71.818;58.324;0.43;0.27;0.21;0.19;0.4;0.21;0.81;1.32;1.12;0.09;0.09;0.05

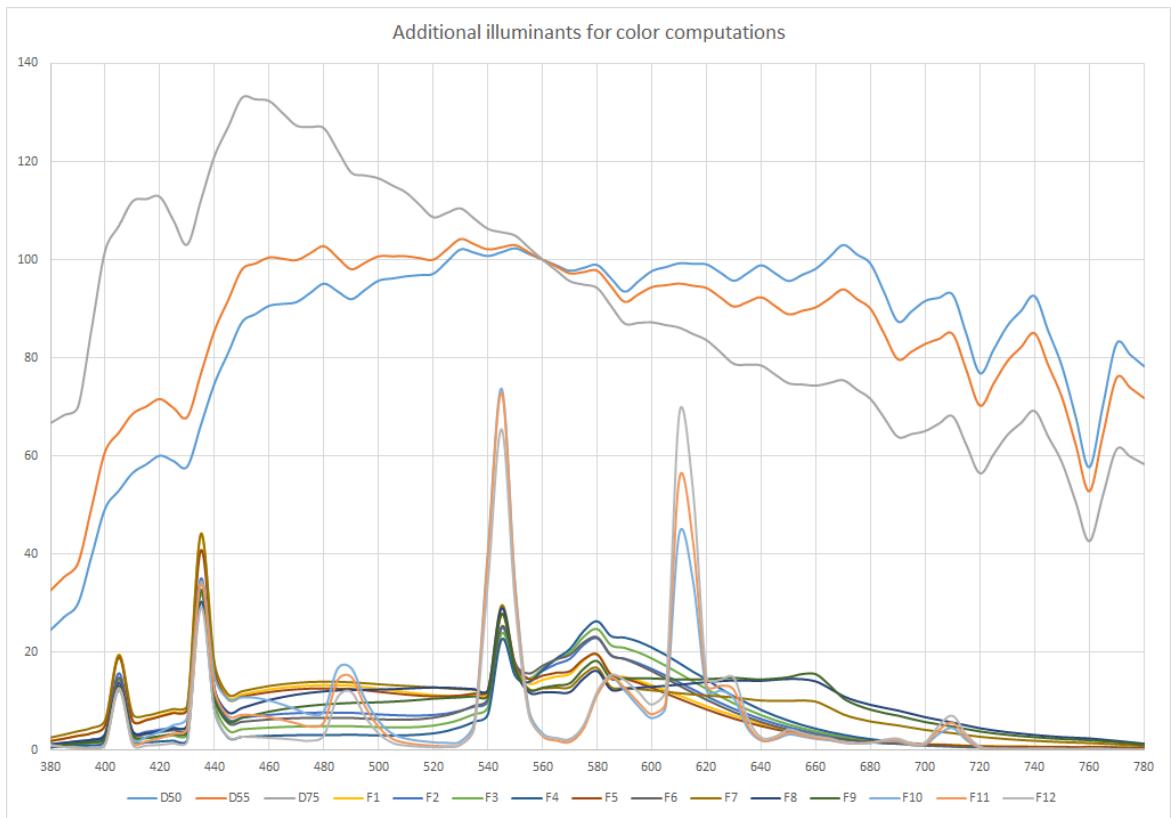
Opened in Excel this looks like this:

D50	D55	D75	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	
380	24.488	32.584	66.703	1.87	1.18	0.82	0.57	1.87	1.05	2.56	1.21	0.9	1.11	0.91	0.96
385	27.179	35.335	68.333	2.36	1.48	1.02	0.7	2.35	1.31	3.18	1.5	1.12	0.63	0.63	0.64
390	29.871	38.087	69.963	2.94	1.84	1.26	0.87	2.92	1.63	3.84	1.81	1.36	0.62	0.46	0.45
395	39.589	49.518	85.946	3.47	2.15	1.44	0.98	3.45	1.9	4.53	2.13	1.6	0.57	0.37	0.33
400	49.308	60.949	101.929	5.17	3.44	2.57	2.01	5.1	3.11	6.15	3.17	2.59	1.48	1.29	1.19
405	52.91	64.751	106.911	19.49	15.69	14.36	13.75	18.91	14.8	19.37	13.08	12.8	12.16	12.68	12.48
410	56.513	68.554	111.894	6.13	3.85	2.7	1.95	6	3.43	7.37	3.83	3.05	2.12	1.59	1.12
415	58.273	70.065	112.346	6.24	3.74	2.45	1.59	6.11	3.3	7.05	3.45	2.56	2.7	1.79	0.94
420	60.034	71.577	112.798	7.01	4.19	2.73	1.76	6.85	3.68	7.71	3.86	2.86	3.74	2.46	1.08
425	58.926	69.746	107.945	7.79	4.62	3	1.93	7.58	4.07	8.41	4.42	3.3	5.14	3.33	1.37
430	57.818	67.914	103.092	8.56	5.06	3.28	2.1	8.31	4.45	9.15	5.09	3.82	6.75	4.49	1.78
435	66.321	76.76	112.145	43.67	34.98	31.85	30.28	40.76	32.61	44.14	34.1	32.62	34.39	33.94	29.05
440	74.825	85.605	121.198	16.94	11.81	9.47	8.03	16.06	10.74	17.52	12.42	10.77	14.86	12.13	7.9
445	81.036	91.799	127.104	10.72	6.27	4.02	2.55	10.32	5.48	11.35	7.68	5.84	10.4	6.95	2.65
450	87.247	97.993	133.01	11.35	6.63	4.25	2.7	10.91	5.78	12	8.6	6.57	10.76	7.19	2.71
455	88.93	99.228	132.682	11.89	6.93	4.44	2.82	11.4	6.03	12.58	9.46	7.25	10.67	7.12	2.65
460	90.612	100.46	132.355	12.37	7.19	4.59	2.91	11.83	6.25	13.08	10.24	7.86	10.11	6.72	2.49
				3											
465	90.99	100.18	129.838	12.75	7.4	4.72	2.99	12.17	6.41	13.45	10.84	8.35	9.27	6.13	2.33
				8											
470	91.368	99.913	127.322	13	7.54	4.8	3.04	12.4	6.52	13.71	11.33	8.75	8.29	5.46	2.1

475	93.238	101.32	127.061	13.15	7.62	4.86	3.08	12.54	6.58	13.88	11.71	9.06	7.29	4.79	1.91
480	95.109	102.73	126.8	13.23	7.65	4.87	3.09	12.58	6.59	13.95	11.98	9.31	7.91	5.66	3.01
485	93.536	100.40	122.291	13.17	7.62	4.85	3.09	12.52	6.56	13.93	12.17	9.48	16.64	14.29	10.83
490	91.963	98.078	117.783	13.13	7.62	4.88	3.14	12.47	6.56	13.82	12.28	9.61	16.73	14.96	11.88
495	93.843	99.379	117.186	12.85	7.45	4.77	3.06	12.2	6.42	13.64	12.32	9.68	10.44	8.97	6.88
500	95.724	100.68	116.589	12.52	7.28	4.67	3	11.89	6.28	13.43	12.35	9.74	5.94	4.72	3.43
505	96.169	100.68	115.146	12.2	7.15	4.62	2.98	11.61	6.2	13.25	12.44	9.88	3.34	2.33	1.49
510	96.613	100.69	113.702	11.83	7.05	4.62	3.01	11.33	6.19	13.08	12.55	10.04	2.35	1.47	0.92
515	96.871	100.34	111.181	11.5	7.04	4.73	3.14	11.1	6.3	12.93	12.68	10.26	1.88	1.1	0.71
520	97.129	99.987	108.659	11.22	7.16	4.99	3.41	10.96	6.6	12.78	12.77	10.48	1.59	0.89	0.6
525	99.614	102.09	109.552	11.05	7.47	5.48	3.9	10.97	7.12	12.6	12.72	10.63	1.47	0.83	0.63
530	102.09	104.21	110.445	11.03	8.04	6.25	4.69	11.16	7.94	12.44	12.6	10.76	1.8	1.18	1.1
535	101.42	103.15	108.367	11.18	8.88	7.34	5.81	11.54	9.07	12.33	12.43	10.96	5.71	4.9	4.56
540	100.75	102.10	106.289	11.53	10.01	8.78	7.32	12.12	10.49	12.26	12.22	11.18	40.98	39.59	34.4
545	101.53	102.53	105.596	27.74	24.88	23.82	22.59	27.78	25.22	29.52	28.96	27.71	73.69	72.84	65.4
550	102.31	102.96	104.904	17.05	16.64	16.14	15.11	17.73	17.46	17.05	16.51	16.29	33.61	32.61	29.48
555	101.15	101.48	102.452	13.55	14.59	14.59	13.88	14.47	15.63	12.44	11.79	12.28	8.24	7.52	7.16
560	100	100	100	14.33	16.16	16.63	16.33	15.2	17.22	12.58	11.76	12.74	3.38	2.83	3.08
565	98.868	98.608	97.808	15.01	17.56	18.49	18.68	15.77	18.53	12.72	11.77	13.21	2.47	1.96	2.47
570	97.735	97.216	95.616	15.52	18.62	19.95	20.64	16.1	19.43	12.83	11.84	13.65	2.14	1.67	2.27
575	98.327	97.482	94.914	18.29	21.47	23.11	24.28	18.54	21.97	15.46	14.61	16.57	4.86	4.43	5.09
580	98.918	97.749	94.213	19.55	22.79	24.69	26.26	19.5	23.01	16.75	16.11	18.14	11.45	11.28	11.96
585	96.208	94.59	90.605	15.48	19.29	21.41	23.28	15.39	19.41	12.83	12.34	14.55	14.79	14.76	15.32
590	93.499	91.432	86.997	14.91	18.66	20.85	22.94	14.64	18.56	12.67	12.53	14.65	12.16	12.73	14.27
595	95.593	92.926	87.112	14.15	17.73	19.93	22.14	13.72	17.42	12.45	12.72	14.66	8.97	9.74	11.86
600	97.688	94.419	87.227	13.22	16.54	18.67	20.91	12.69	16.09	12.19	12.92	14.61	6.52	7.33	9.28
605	98.478	94.78	86.684	12.19	15.21	17.22	19.43	11.57	14.64	11.89	13.12	14.5	8.81	9.72	12.31
610	99.269	95.14	86.14	11.12	13.8	15.65	17.74	10.45	13.15	11.6	13.34	14.39	44.12	55.27	68.53
615	99.155	94.68	84.861	10.03	12.36	14.04	16	9.35	11.68	11.35	13.61	14.4	34.55	42.58	53.02
620	99.042	94.22	83.581	8.95	10.95	12.45	14.42	8.29	10.25	11.12	13.87	14.47	12.09	13.18	14.67
625	97.382	92.334	81.164	7.96	9.65	10.95	12.56	7.32	8.96	10.95	14.07	14.62	12.15	13.16	14.38
630	95.722	90.448	78.747	7.02	8.4	9.51	10.93	6.41	7.74	10.76	14.2	14.72	10.52	12.26	14.71
635	97.29	91.389	78.587	6.2	7.32	8.27	9.52	5.63	6.69	10.42	14.16	14.55	4.43	5.11	6.46
640	98.857	92.33	78.428	5.42	6.31	7.11	8.18	4.9	5.71	10.11	14.13	14.4	1.95	2.07	2.57
645	97.262	90.592	76.614	4.73	5.43	6.09	7.01	4.26	4.87	10.04	14.34	14.58	2.19	2.34	2.75
650	95.667	88.854	74.801	4.15	4.68	5.22	6	3.72	4.16	10.02	14.5	14.88	3.19	3.58	4.18
655	96.929	89.586	74.562	3.64	4.02	4.45	5.11	3.25	3.55	10.11	14.46	15.51	2.77	3.01	3.44
660	98.19	90.317	74.324	3.2	3.45	3.8	4.36	2.83	3.02	9.87	14	15.47	2.29	2.48	2.81
665	100.59	92.133	74.873	2.81	2.96	3.23	3.69	2.49	2.57	8.65	12.58	13.2	2	2.14	2.42
670	103.00	93.95	75.422	2.47	2.55	2.75	3.13	2.19	2.2	7.27	10.99	10.57	1.52	1.54	1.64
675	101.06	91.953	73.499	2.18	2.19	2.33	2.64	1.93	1.87	6.44	9.98	9.18	1.35	1.33	1.36
680	99.133	89.956	71.576	1.93	1.89	1.99	2.24	1.71	1.6	5.83	9.22	8.25	1.47	1.46	1.49

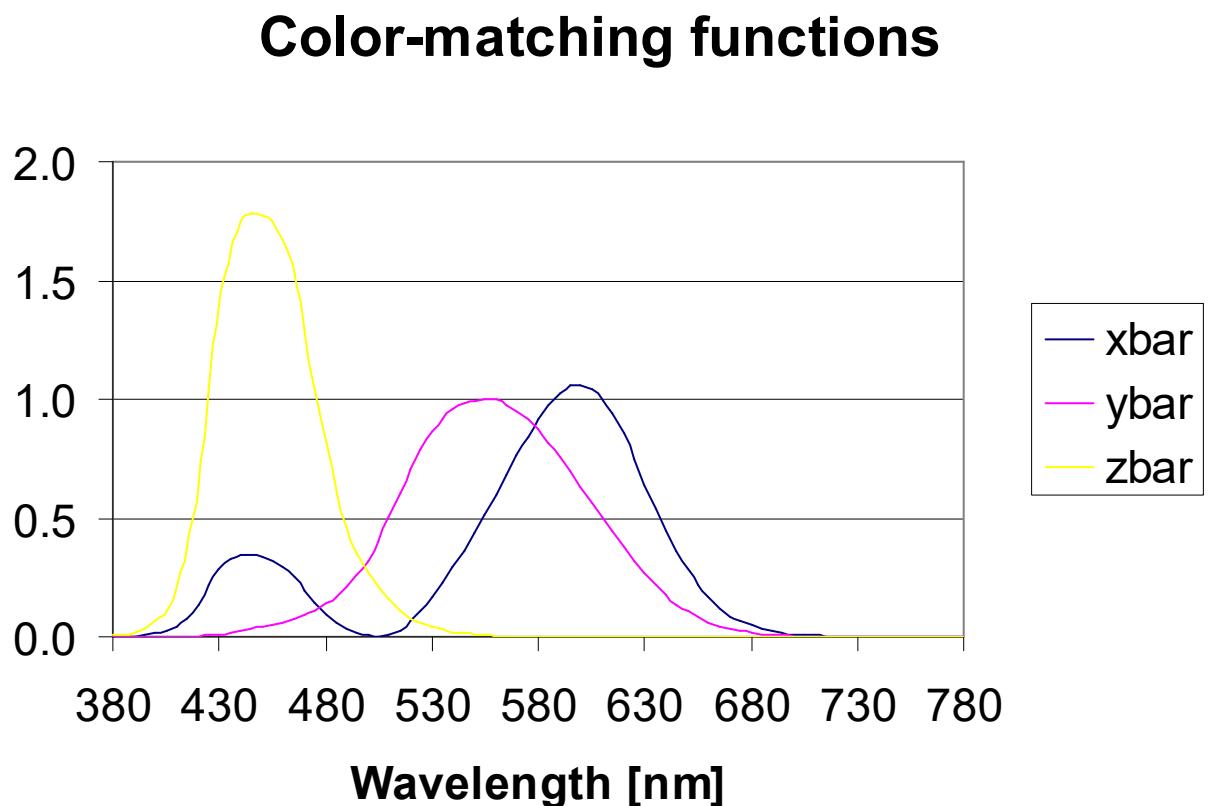
685	93.257	84.817	67.714	1.72	1.64	1.7	1.91	1.52	1.37	5.41	8.62	7.57	1.79	1.94	2.14
690	87.381	79.677	63.852	1.67	1.53	1.55	1.7	1.43	1.29	5.04	8.07	7.03	1.74	2	2.34
695	89.492	81.258	64.464	1.43	1.27	1.27	1.39	1.26	1.05	4.57	7.39	6.35	1.02	1.2	1.42
700	91.604	82.84	65.076	1.29	1.1	1.09	1.18	1.13	0.91	4.12	6.71	5.72	1.14	1.35	1.61
705	92.246	83.842	66.573	1.19	0.99	0.96	1.03	1.05	0.81	3.77	6.16	5.25	3.32	4.1	5.04
710	92.889	84.844	68.07	1.08	0.88	0.83	0.88	0.96	0.71	3.46	5.63	4.8	4.49	5.58	6.98
715	84.872	77.539	62.256	0.96	0.76	0.71	0.74	0.85	0.61	3.08	5.03	4.29	2.05	2.51	3.19
720	76.854	70.235	56.443	0.88	0.68	0.62	0.64	0.78	0.54	2.73	4.46	3.8	0.49	0.57	0.71
725	81.683	74.768	60.343	0.81	0.61	0.54	0.54	0.72	0.48	2.47	4.02	3.43	0.24	0.27	0.3
730	86.511	79.301	64.242	0.77	0.56	0.49	0.49	0.68	0.44	2.25	3.66	3.12	0.21	0.23	0.26
735	89.546	82.147	66.697	0.75	0.54	0.46	0.46	0.67	0.43	2.06	3.36	2.86	0.21	0.21	0.23
740	92.58	84.993	69.151	0.73	0.51	0.43	0.42	0.65	0.4	1.9	3.09	2.64	0.24	0.24	0.28
745	85.405	78.437	63.89	0.68	0.47	0.39	0.37	0.61	0.37	1.75	2.85	2.43	0.24	0.24	0.28
750	78.23	71.88	58.629	0.69	0.47	0.39	0.37	0.62	0.38	1.62	2.65	2.26	0.21	0.2	0.21
755	67.961	62.337	50.623	0.64	0.43	0.35	0.33	0.59	0.35	1.54	2.51	2.14	0.17	0.24	0.17
760	57.692	52.793	42.617	0.68	0.46	0.38	0.35	0.62	0.39	1.45	2.37	2.02	0.21	0.32	0.21
765	70.307	64.36	51.985	0.69	0.47	0.39	0.36	0.64	0.41	1.32	2.15	1.83	0.22	0.26	0.19
770	82.923	75.927	61.352	0.61	0.4	0.33	0.31	0.55	0.33	1.17	1.89	1.61	0.17	0.16	0.15
775	80.599	73.872	59.838	0.52	0.33	0.28	0.26	0.47	0.26	0.99	1.61	1.38	0.12	0.12	0.1
780	78.274	71.818	58.324	0.43	0.27	0.21	0.19	0.4	0.21	0.81	1.32	1.12	0.09	0.09	0.05

Here is a graph of these demo spectra:



### 3.23 Color-matching functions

The color-matching functions for 2° used for the computation of color coordinates are these:



The values are taken from DIN 5033:

Wavelength [nm]	xbar	ybar	zbar
380	0.0014	0.0000	0.0065
385	0.0022	0.0001	0.0105
390	0.0042	0.0001	0.0202
395	0.0076	0.0002	0.0362
400	0.0143	0.0004	0.0679
405	0.0232	0.0006	0.1102
410	0.0435	0.0012	0.2074
415	0.0776	0.0022	0.3713
420	0.1344	0.0040	0.6456
425	0.2148	0.0073	1.0391
430	0.2839	0.0116	1.3856
435	0.3285	0.0168	1.6230
440	0.3483	0.0230	1.7471
445	0.3481	0.0298	1.7826
450	0.3362	0.0380	1.7721
455	0.3187	0.0480	1.7441
460	0.2908	0.0600	1.6692
465	0.2511	0.0739	1.5281
470	0.1954	0.0910	1.2876
475	0.1421	0.1126	1.0419

480 0.0956 0.1390 0.8130  
485 0.0580 0.1693 0.6162  
490 0.0320 0.2080 0.4652  
495 0.0147 0.2586 0.3533  
500 0.0049 0.3230 0.2720  
505 0.0024 0.4073 0.2123  
510 0.0093 0.5030 0.1582  
515 0.0291 0.6082 0.1117  
520 0.0633 0.7100 0.0782  
525 0.1096 0.7932 0.0573  
530 0.1655 0.8620 0.0422  
535 0.2257 0.9149 0.0298  
540 0.2904 0.9540 0.0203  
545 0.3597 0.9803 0.0134  
550 0.4334 0.9950 0.0087  
555 0.5121 1.0000 0.0057  
560 0.5945 0.9959 0.0039  
565 0.6784 0.9786 0.0027  
570 0.7621 0.9520 0.0021  
575 0.8425 0.9154 0.0018  
580 0.9163 0.8700 0.0017  
585 0.9786 0.8163 0.0014  
590 1.0263 0.7570 0.0011  
595 1.0567 0.6949 0.0010  
600 1.0622 0.6310 0.0008  
605 1.0456 0.5668 0.0006  
610 1.0026 0.5030 0.0003  
615 0.9384 0.4412 0.0002  
620 0.8544 0.3810 0.0002  
625 0.7514 0.3210 0.0001  
630 0.6424 0.2650 0.0000  
635 0.5419 0.2170 0.0000  
640 0.4479 0.1750 0.0000  
645 0.3608 0.1382 0.0000  
650 0.2835 0.1070 0.0000  
655 0.2187 0.0816 0.0000  
660 0.1649 0.0610 0.0000  
665 0.1212 0.0446 0.0000  
670 0.0874 0.0320 0.0000  
675 0.0636 0.0232 0.0000  
680 0.0468 0.0170 0.0000  
685 0.0329 0.0119 0.0000  
690 0.0227 0.0082 0.0000  
695 0.0158 0.0057 0.0000  
700 0.0114 0.0041 0.0000  
705 0.0081 0.0029 0.0000  
710 0.0058 0.0021 0.0000  
715 0.0041 0.0015 0.0000  
720 0.0029 0.0010 0.0000  
725 0.0020 0.0007 0.0000  
730 0.0014 0.0005 0.0000  
735 0.0010 0.0004 0.0000  
740 0.0007 0.0002 0.0000  
745 0.0005 0.0002 0.0000  
750 0.0003 0.0001 0.0000  
755 0.0002 0.0001 0.0000  
760 0.0002 0.0001 0.0000

```
765 0.0001 0.0000 0.0000
770 0.0001 0.0000 0.0000
775 0.0001 0.0000 0.0000
780 0.0000 0.0000 0.0000
```

For 10° the values are the following:

```
Wavelength [nm] xbar10 ybar10 zbar10
380 0.0002 0.0000 0.0007
385 0.0007 0.0001 0.0029
390 0.0024 0.0003 0.0105
395 0.0072 0.0008 0.0323
400 0.0191 0.0020 0.0860
405 0.0434 0.0045 0.1971
410 0.0847 0.0088 0.3894
415 0.1406 0.0145 0.6568
420 0.2045 0.0214 0.9725
425 0.2647 0.0295 1.2825
430 0.3147 0.0387 1.5535
435 0.3577 0.0496 1.7985
440 0.3837 0.0621 1.9673
445 0.3867 0.0747 2.0273
450 0.3707 0.0895 1.9948
455 0.3430 0.1063 1.9007
460 0.3023 0.1282 1.7454
465 0.2541 0.1528 1.5549
470 0.1956 0.1852 1.3176
475 0.1323 0.2199 1.0302
480 0.0805 0.2536 0.7721
485 0.0411 0.2977 0.5701
490 0.0162 0.3391 0.4153
495 0.0051 0.3954 0.3024
500 0.0038 0.4608 0.2185
505 0.0154 0.5314 0.1592
510 0.0375 0.6067 0.1120
515 0.0714 0.6857 0.0822
520 0.1177 0.7618 0.0607
525 0.1730 0.8233 0.0431
530 0.2365 0.8752 0.0305
535 0.3042 0.9238 0.0206
540 0.3768 0.9620 0.0137
545 0.4516 0.9822 0.0079
550 0.5298 0.9918 0.0040
555 0.6161 0.9991 0.0011
560 0.7052 0.9973 0.0000
565 0.7938 0.9824 0.0000
570 0.8787 0.9556 0.0000
575 0.9512 0.9152 0.0000
580 1.0142 0.8689 0.0000
585 1.0743 0.8256 0.0000
590 1.1185 0.7774 0.0000
595 1.1343 0.7204 0.0000
600 1.1240 0.6583 0.0000
605 1.0891 0.5939 0.0000
610 1.0305 0.5280 0.0000
615 0.9507 0.4618 0.0000
620 0.8563 0.3981 0.0000
625 0.7549 0.3396 0.0000
```

```

630 0.6475 0.2835 0.0000
635 0.5351 0.2283 0.0000
640 0.4316 0.1798 0.0000
645 0.3437 0.1402 0.0000
650 0.2683 0.1076 0.0000
655 0.2043 0.0812 0.0000
660 0.1526 0.0603 0.0000
665 0.1122 0.0441 0.0000
670 0.0813 0.0318 0.0000
675 0.0579 0.0226 0.0000
680 0.0409 0.0159 0.0000
685 0.0286 0.0111 0.0000
690 0.0199 0.0077 0.0000
695 0.0138 0.0054 0.0000
700 0.0096 0.0037 0.0000
705 0.0066 0.0026 0.0000
710 0.0046 0.0018 0.0000
715 0.0031 0.0012 0.0000
720 0.0022 0.0008 0.0000
725 0.0015 0.0006 0.0000
730 0.0010 0.0004 0.0000
735 0.0007 0.0003 0.0000
740 0.0005 0.0002 0.0000
745 0.0004 0.0001 0.0000
750 0.0003 0.0001 0.0000
755 0.0002 0.0001 0.0000
760 0.0001 0.0000 0.0000
765 0.0001 0.0000 0.0000
770 0.0001 0.0000 0.0000
775 0.0000 0.0000 0.0000
780 0.0000 0.0000 0.0000

```

### 3.24 Solar spectral distribution

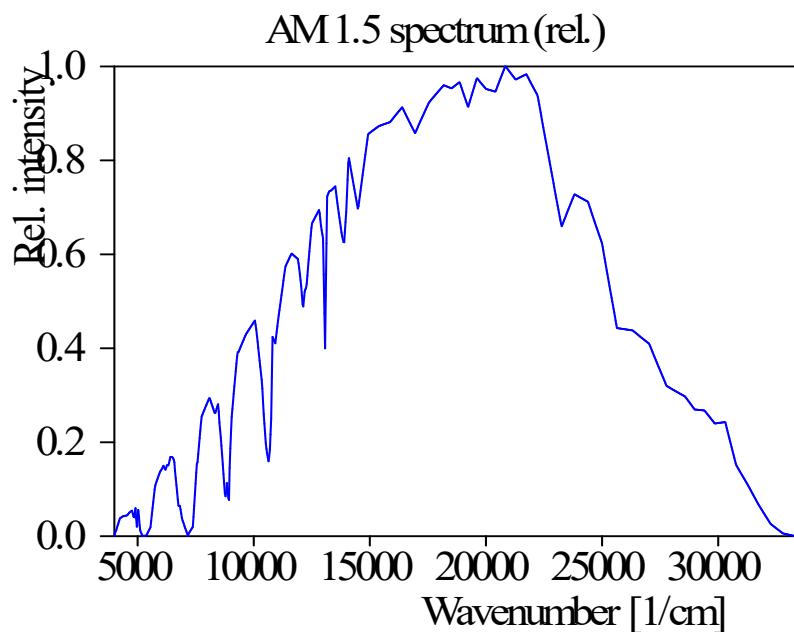
The product  $S_{\lambda} \cdot \Delta\lambda$  used for the computation of the solar direct transmittance and reflectance is given for the required wavelengths in the following (DIN 67 507):

Wavelength [nm]	$S_{\lambda} \cdot \Delta\lambda$
300	0.0050
340	0.0240
380	0.0320
420	0.0500
460	0.0650
500	0.0630
540	0.0580
580	0.0540
620	0.0550
660	0.0490
700	0.0460
740	0.0410
780	0.0370
900	0.1390
1100	0.0970
1300	0.0580
1500	0.0390

1700	0.0260
1900	0.0180
2500	0.0440

### 3.25 AM 1.5 spectrum

The relative spectral intensity of solar radiation in Europe is approximately given by the AM 1.5 spectrum:



# Index

## - % -

% 5

## - 1 -

10° 8

## - 2 -

2° 8

## - 3 -

3.79 78

3-point-averaging 39

## - A -

a 8, 67

A spectrum 80

a\* 8, 17, 21, 44, 67

absorption 79

absorption bands 71

absorption coefficient 40

absorption of light 71

acquire 26

ActiveX palette 60

add\_layer\_definition\_on\_top 26

add\_layer\_on\_top 28

add\_to\_report 28

additional illuminants 8

air 75

AM 1.5 71

angle distribution incidence spectrum 40, 41

angle distribution incidence weight spectrum 41

angle of incidence 8, 13, 17

angle range 8

angle variation 8

architectural applications 13

Argon 75

author 3

automatic fit 34

Automatic saving of page modifications 23

automation 60  
Automation open 60  
Automation refnum 60  
average time 42  
averaging 78

## - B -

b 8, 67  
b\* 8, 17, 21, 44, 67  
batch control window 21  
Begin superlattice 26  
black 8  
Block diagram 60  
bottom 38  
building 13  
button panel 24

## - C -

C 8, 67  
C spectrum 84  
cell 56  
center-of-glass 78  
center-of-glass value 78  
Charge carrier generation 71  
checkbox 17, 75  
checkboxes 8  
CIELAB 67  
clear\_fit\_parameters 29  
clear\_layer\_stack 29  
clear\_material\_list 30  
clear\_report 30  
clipboard 36  
coating 8, 56  
coatings 79  
coatings on glass 2  
CODE 21  
CODE library 60  
coherent 26, 76  
coherent superposition 28  
color 8, 44  
Color angle variation 17  
Color box 21  
color coordinate 44  
color coordinates 8, 17, 65, 67, 91  
color dialog 17  
color distance 8  
Color fluctuation 8

color impression 8  
 color matching functions 8  
 color rendering index 67  
 Color view 8, 17  
 color-matching functions 65, 91  
 Colors 21  
 computation 16  
 conductivity 75  
 configuration 21, 37, 45, 56  
 configuration\_file 45  
 conversion 8  
 conversion efficiency 71  
 count 55  
 csv file 8  
 current density 71  
 current rating 45

## - D -

D65 8, 65, 67, 82  
 D65 spectrum 82  
 damage 78  
 dashed line 17  
 data acquisition 60  
 data points 38  
 database 28, 35, 75  
 decimals 5  
 default name 5  
 default setting 29, 33  
 delete 32  
 delta 49  
 deviation 5, 8  
 df\_imag\_part\_nm 46  
 df\_load\_from\_file 32  
 df\_real\_part\_nm 45  
 dialog 5  
 dielectric function 32, 35, 37, 40, 45, 46, 52  
 differences 16, 21  
 DIN 5033 65, 67, 91  
 DIN 67 504 68, 69  
 DIN 67 507 74  
 Display 39  
 Dominant wavelength 8  
 Downhill simplex 48  
 download 25  
 drag 17  
 drawing 40  
 dropdown box 24  
 Drude 75

## - E -

Edit 5, 8, 17  
 effective emissivity 75  
 electrical conductivity 75  
 electrical contact 71  
 ellipsometry 49  
 emissivity 75, 79  
 EN 673 75  
 End superlattice 26  
 energy transmittance 74  
 example 56  
 Excel 56  
 experimental data 5  
 experimental spectra 34  
 experimental spectrum 46  
 experimental\_spectrum\_value 46  
 export 29, 33  
 Export data 5  
 extrapolation 5

## - F -

F4 function key 5  
 F5 function key 5  
 F7 23  
 factor 29, 33  
 fft 47  
 FFT thickness analysis 47  
 fft\_strength 47  
 fft\_thickness 47  
 file 45  
 file extension 23  
 filling gas 78  
 film thickness 8  
 fit activity 48  
 fit deviation 5, 47  
 Fit on a grid 48  
 fit parameter 29, 30, 32, 33, 47, 48  
 fit parameter set 48  
 fit parameters 29, 52, 56, 71  
 fit quality 37  
 fit\_parameter\_mode 48  
 fit\_parameter\_name 47  
 fit\_parameter\_set 48  
 fit\_parameter\_value 48  
 fit\_parameter\_value\_max 48  
 fit\_parameter\_value\_min 48

fitting 48  
 fluctuation 8  
 fluctuations 13  
 foil 76  
 format 17  
 formula 8, 48  
 found\_filename 49  
 foundfiles 49  
 frame effects 78  
 Frozen 48  
 function 16  
 Functions of integrals quantities 16

## - G -

g 74  
 gas 76  
 gas filling 75  
 gas properties 78  
 Generate PDF document 25  
 get\_sim\_val1 49  
 get\_sim\_val2 49  
 glass 76  
 glass pane 71, 75  
 glazing 79  
 glazing composition 76  
 glazing system 78  
 graphics 8  
 grid 34  
 grid\_test 34  
 grid\_test\_max 49  
 grid\_test\_min 49  
 grid\_test\_no\_points 49  
 grid\_test\_para 49

## - H -

halfspace 29, 38, 39  
 hardware 60  
 heat 79  
 heat transfer 74, 78  
 Height 51  
 hide 34  
 human 78

## - I -

illuminant 8, 44, 80, 82, 84  
 illumination power 71

image 17  
 imaginary part 46  
 import 29, 33, 34  
 import data 32  
 import illuminants 8  
 imported dielectric function 32  
 incidence\_angle 50  
 incoherent 26  
 incoherent superposition 28  
 index 53, 55  
 infrared 75  
 integral quantities 2, 5, 17, 21, 53, 76  
 integral quantity 17, 55  
 interpolation 5  
 ISO 15099 78  
 ISO 9845-1 78  
 iteration 42

## - K -

k 65  
 kappa 51  
 Krypton 75

## - L -

L 8, 67  
 L\* 8, 17, 44, 67  
 L\*a\*b\* space 8  
 LabView 60  
 layer 26, 28, 51  
 layer definitions 26  
 layer stack 13, 26, 28, 29, 38, 39, 51, 56, 76  
 layer stack definition 76  
 layer stacks 30, 52  
 layer thickness 8, 76  
 layer\_thickness 51  
 layers 29, 52  
 left 51  
 light reflectance 15, 69  
 light transmittance 15, 68  
 list of configurations 23  
 list of fit parameters 30  
 list of integral quantities 5, 76  
 list of layer stacks 26, 28  
 list of materials 28, 30, 35  
 list of optical functions 21  
 'List view' 24  
 load dielectric function 35

load\_material\_from\_database 35  
lower boundary 48

## - M -

macro language 56  
macros 56  
main window 17, 51  
master parameter 30, 48, 71  
Master parameter list 30  
material 35, 36, 38, 39, 51  
Materials 30  
maximum 48  
message 38  
methods 26  
micron 28  
mm 28  
mode parameter 17  
model 56, 75  
model parameters 56

## - N -

National Fenestration Rating Council 78  
Navigation 24  
NFRC 78, 79  
NFRC 300-2004 78  
nickname 5, 16, 17  
nm 28  
normal emissivity 75  
notation 76  
number of optical functions 52  
number\_of\_fit\_parameters 52  
number\_of\_spectra 53  
number\_of\_tec\_values 53  
numerical value 17

## - O -

observation angle 8, 44  
observer 13  
office tower 13  
OLE 26  
OLE automation 56  
OLE commands 26  
OLE example 56  
operator display 60  
optical constants 35, 37, 56, 75  
optical design 2

optical function 53  
optical functions 16, 21, 52, 78  
Optimize 5  
options 8  
order 16  
outside 74  
outside temperature 78, 79

## - P -

page 23, 25  
Page down 24  
Page up 24  
parameter fit 40  
Parameter fluctuation 21  
Parameter variation 21, 71  
partial wave 26  
PDF file 25  
Penalty shape 5  
Photocurrent 71  
photons 71  
Photopic 78  
physical properties 75  
polarization 8, 13  
position 39  
prepare\_shutdown 60  
presentation 22, 23, 24  
'Presentation contro 24  
presentation folder 25  
print 37  
properties 26  
Psi 49  
Purity 8

## - Q -

quantity 5

## - R -

Ra 67  
radiation 78  
range 38  
rating 45  
real part 45  
re-computation 40  
rectangle 17  
reflectance 13, 34  
refractive index 47, 51, 54

relative spectral distribution 68  
 rename\_material 36  
 report window 30, 36  
 report\_to\_clipboard 36  
 report\_to\_file 36  
 RGB 8  
 room temperature 78, 79  
 Rough interface 26

**- S -**

S 65  
 save configuration 37  
 save dielectric function 37  
 scaling factor 5  
 scanning spectrometer system 60  
 SCOU file format 2  
 SCOUT 2, 21  
 SCOUT manual 5  
 SCOUT technical manual 13  
 scoutole 60  
 Screen 34, 39, 51  
 searchfiles 38  
 secondary heat gain coefficient 79  
 secondary internal heat transfer factor 74  
 select a filename 40  
 Select ActiveX class 60  
 sequence of pages 23  
 sequences of CODE configurations 22  
 server 56  
 server name 26  
 set\_bottom\_halfspace\_material 38  
 set\_spectrum\_simulation\_range 38  
 set\_top\_halfspace\_material 39  
 SF6 75  
 Sheet resistance 75  
 SHGC 79  
 show 39  
 Show Creatable Objects Only 60  
 show\_maximized 39  
 show\_minimized 39  
 show\_normal 39  
 simple layer 26, 76  
 simulated quantities 49  
 simulated spectrum 5, 54  
 simulated\_spectrum\_value 54  
 simulated\_spectrum\_value\_nm 54  
 size 39  
 skin 78

sky 78, 79  
 slide show 22  
 slider 8  
 sliders 21  
 Smoothing 39  
 smoothing iterations 39  
 solar 78  
 solar cell 71  
 solar direct transmittance 74  
 solar direct absorptance 74  
 Solar direct reflectance 15, 70, 94  
 Solar direct transmittance 15, 70, 94  
 solar factor 74  
 solar power 78, 79  
 Solar relative spectral distribution 94  
 solid line 17  
 sound insulation 76  
 spectra 21, 56  
 Spectra list 30  
 spectral distribution 71  
 spectral luminous efficiency 68  
 spectral range 5  
 spectral unit 38  
 spectrometers 26  
 spectrum 5, 34, 37, 38, 44, 49, 55  
 Spectrum product 15  
 spectrum simulation software 2  
 Spectrum\_product 70  
 spectrum\_weight 55  
 squared difference 5  
 stack 51  
 stacks 52  
 standard file 32  
 start 40  
 start configuraton 21  
 Start.wcd 21  
 starting values 34  
 stop 40  
 sums 16  
 sun light 71  
 superlattice 26  
 superposition 26  
 surface 75

**- T -**

tab character 26  
 table 56, 82  
 table of content 24

Tables 5  
 target value 5, 8  
 TCO 71  
 Tdw-ISO 78  
 tec\_value 55  
 tec\_value\_name 55  
 technical data 56  
 technical values 53, 56  
 temperature distribution 78  
 temperature profile 79  
 text 53  
 text file 36  
 Thermal transmittance 75  
 Thick layer 26, 28  
 thickness 28, 51, 56  
 thickness fluctuations 13  
 thickness profile 60  
 thickness unit 28  
 thickness variation 26  
 Thin film 26, 28  
 thin film design 2  
 top 39  
 total solar energy transmittance 74  
 transfer 29, 33  
 transmittance 34, 65, 78  
 transparent conductive layer 71  
 treeview 23  
 tristimulus values 65  
 Tuv 78  
 Type 5  
 type library 60

**- U -**

U 75  
 unit 5, 28  
 update\_data 40  
 update\_plot 40  
 upper boundary 48  
 US standard 78  
 user-defined formula 71  
 UV 78  
 U-value 78, 79

**- V -**

Vacuum 29, 30  
 value 5, 48  
 Variable thickness 26

variant 34  
 variant arrays 34  
 variation mode 48  
 variations 8  
 view element 17  
 view object 71  
 views 17  
 Virtual Office Tower 13, 17  
 visual navigation help 24  
 VisualBasic 26  
 visualization 8  
 VTc 78

**- W -**

wavelength 45, 46, 54  
 wavenumber 40, 46, 51, 54  
 WCD 2  
 weight 5, 55  
 weight function 8, 17  
 weighted 78  
 weighted average 78  
 white 8  
 Width 51  
 wind speed 78  
 window 39  
 WINDOW 4.0 78  
 window panes 13  
 Windows Coating Designer 2  
 Wolfgang Theiss 3  
 workbook 5, 21  
 worksheet 56

**- X -**

X 8, 44, 65, 66, 67  
 Xenon 75  
 x-y-plane 8

**- Y -**

Y 8, 44, 65, 66, 67

**- Z -**

Z 8, 44, 65, 66, 67  
 zip file 25