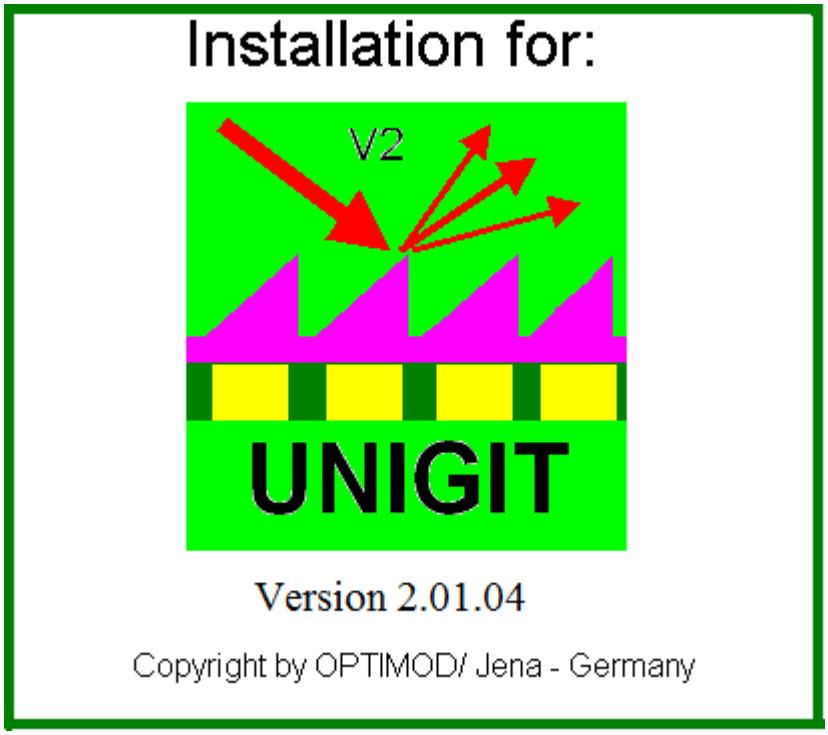
UNIGIT RIGOROUS GRATING SOLVER

VERSION 2.01.04
USER MANUAL



Developed by:

Optimod

Ricarda-Huch-Weg 12

D - 07745 JENA

GERMANY

phone: +49 03641 825944

cell phone: +49 162 9067015

email: support@unigit.com

Change Record

changes versus version 2.01.03

Reason for Change/ Extension	Section(s)	Page(s)
Parallel Editor expanded into General Settings	3	18
RCWA Solver for 1D extended to slanted (non-orthogonal) profiles	5.1.4 and 5.1.5	31 - 34
Cross section view of the whole stack for 1D gratings implemented	4.5	25
Color assignment for materials included via color table	3.2 and 6	19 and 51
Python link implemented	2.9 and 3.5	17 and 20
Layer Editor for C-method polygon layer changed	5.2.1	39
Layer Editor for C-method sinusoidal layer changed	5.2.2	40
C-method stack editor changed	4.2	22
Figure with definition of angles added	2.3	11

Table of Content

1. Introduction

Unigit is a rigorous diffraction solver for 2D (1D periodic) or 3D (2D periodic) multilayer stacks (see Fig. 1). It runs on PC with Windows NT, Windows 2000, Windows XP, Vista and Windows 7 both 32 and 64 bit machines. The schema in Fig. 1 gives an idea what types of complex patterns can be solved with Unigit.

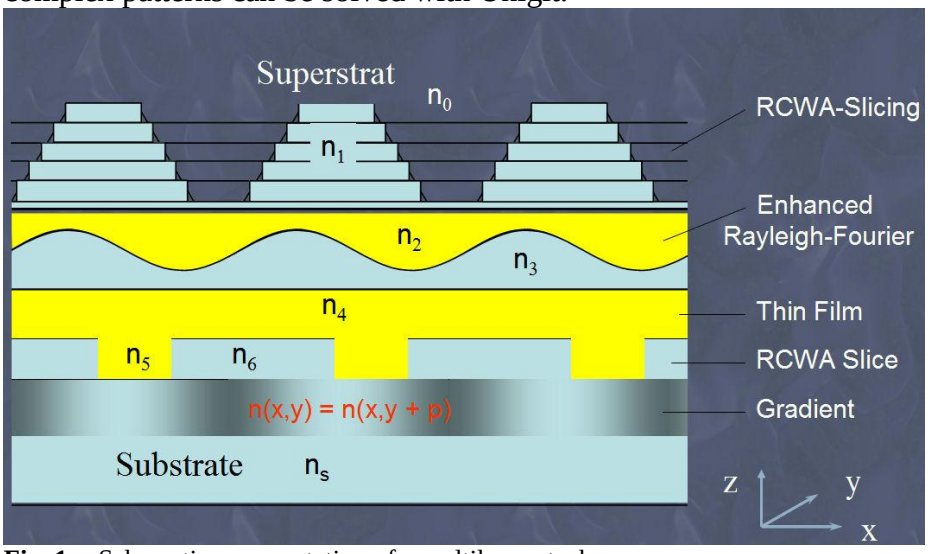


Fig. 1: Schematic representation of a multilayer stack

Unigit V2.XX.XX comprises three basic solution algorithms:

- the Rigorous Coupled Wave Approach (RCWA) a.k.a. Modal Method with Fourier Expansion (see e.g., /1/, /2/),
- the Chandezon method (C-method) a.k.a. Coordinate Transformation Method (see e.g., /8/ & /9/),
- and the Rayleigh Fourier Method (/3/) which is not rigorous.

Presently, algorithms one and three are embedded in the same S-matrix algorithm (see e.g., /4/) ensuring a high degree of stability as well as flexibility while the C-method is implemented separately. It is planned however to merge all three algorithms into one frame for upcoming versions. The algorithms can be applied layer wise, i.e., one layer can be treated with Rayleigh-Fourier and another layer with RCWA. The implementation of RCWA was

done in compliance with Lifeng Li's factorization rules (see e.g., /5/) to insure accurate results. The 2D algorithm follows closely Lifeng Li's paper /6/. In addition, it offers the choice of the Lalanne method /7/ instead of the Li Method. The implementation of the C-method followed mainly the description in /9/ as well as some sophisticated schemas to couple the fields between non-parallel interfaces. The C-method is an ideal supplement to the RCWA. Its preferred application cases are profiles with shallow slopes made from high contrast materials. A particular case are Echelle gratings. Some examples proving the superiority of the C-method in these cases are discussed in /10/.

Unigit consists of a Graphical User Interface (GUI) and four computation kernels. The computation kernels are:

- unigit_1D.exe: A solver for one-dimensional (1D) or line/ space gratings (with orthogonal and/or slanted layers) in classical mount based on RCWA and Rayleigh-Fourier method,
- unigit_1C.exe: A solver for one-dimensional gratings in classical mount based on the C-method,
- unigit_co.exe: A solver for one-dimensional (1D) or line/ space gratings (with orthogonal and/or slanted layers) in conical mount based on RCWA,
- unigit_2D.exe: A solver for non-orthogonal crossed (2D) gratings.

The four computation kernels can be utilized without the Unigit-GUI by embedding it in user applications. This can be done for example with Matlab, Visual Basic, C++ or C#.

In order to speed up the computation, Unigit enables the parallel threading of loops. This feature is particularly recommended to run slow 2D computations on multi-core machines. For instance, a speed up factor of about 4-6 will become possible for a dual quad-core machine. The parallel threading can be activated via the pull down menu of the central control board (for more details see section **Fehler! Verweisquelle konnte nicht gefunden werden.**). Moreover, the simulation of symmetric gratings illuminated in a symmetric setup may be accelerated by taking advantage of these symmetries (see section 2.2). The related acceleration is about 3x (theoretically 4x for both polarizations, 8x for one polarization).

2. Central Control Board

2.1. Introductory Remarks

The central control board is activated immediately after the UNIGIT program is started. It appears as shown in Fig. 2.

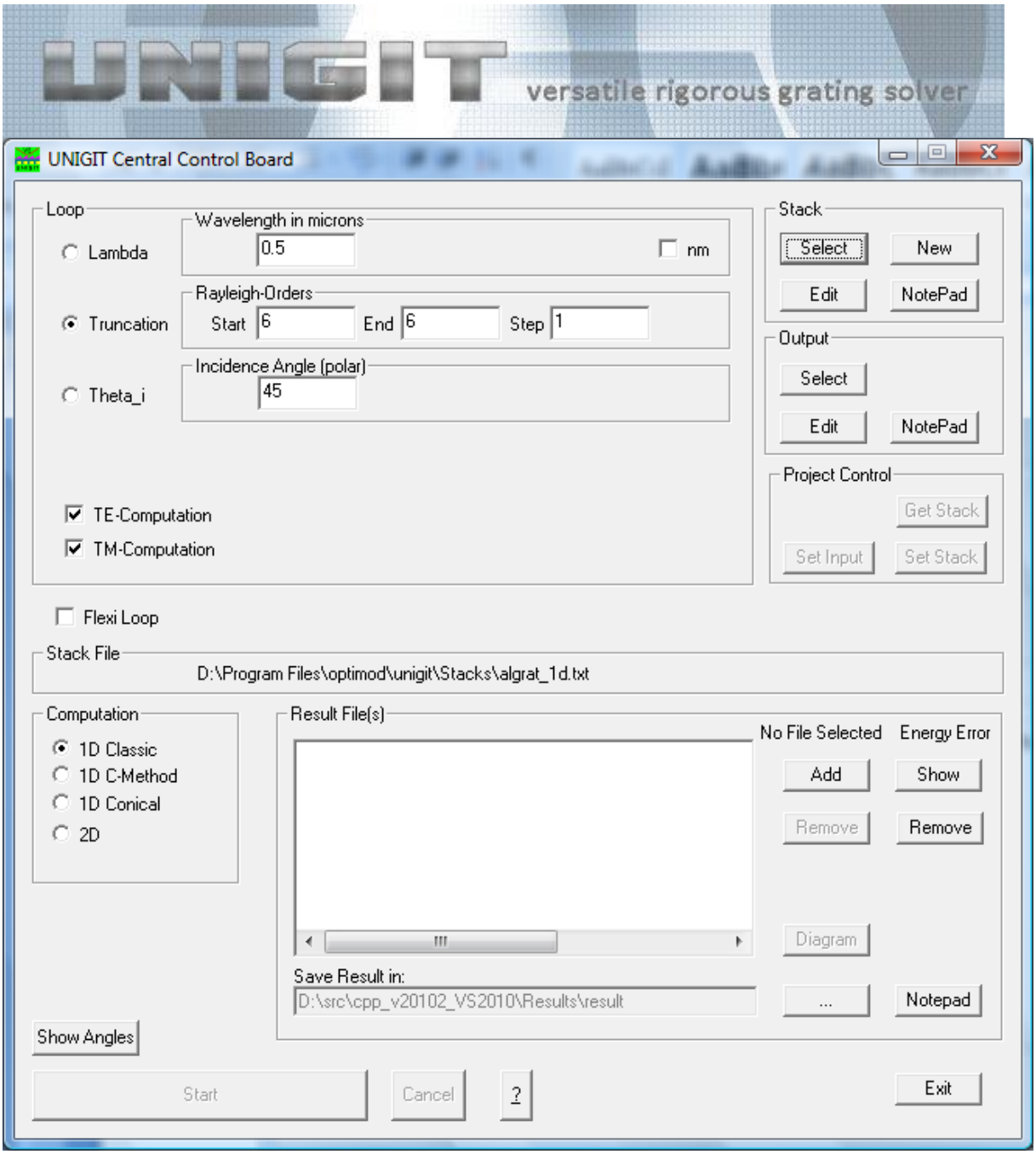


Fig. 2: Central Control Board of Unigit

Mainly, the central control board serves for:

· The preparation of the input files of the underlying Grating Solver Code,
· The launch of the Grating Solver and
· The graphical presentation of the calculation results.

Besides, the basic operation modes of UNIGIT can be selected just by selecting either a regular stack file (extensions ".ust" preferred though you can use other, e.g. ".txt" have been used with older versions of UNIGIT) or a UNIGIT project file (extension ".upr"). Accordingly, there are two basic modes how Unigit can be operated:

- The regular mode,
- The project retrieval mode.

The following description is devoted to the regular mode. The project retrieval mode is described in more detail in section 8.

2.2. Computation Preparation

An essential function of the central control board is to determine the excitation conditions for the diffraction calculation. Especially, the following parameters may be defined:

· The wavelength (to be entered in microns),
· The truncation number (i.e., the minimum and maximum Rayleigh order to be kept in the diffraction matrix algorithm),
· The (polar) incidence angle and
· The azimuthal incidence angle (This option is activated only in the conical and 2D case).

A loop option is linked to each of these four parameters. This means that the computation will be performed for several values of this parameter defined by a start, stop and step value.

Optionally, one geometry parameter of the stack may be also integrated in the loop selection. In order to do this, the particular parameter has to be indicated by means of a dollar sign at the end of its line in the stack file. You can do this by opening the file with the notepad button. The example in Fig. 3 shows the selection of the grating thickness as a parameter.

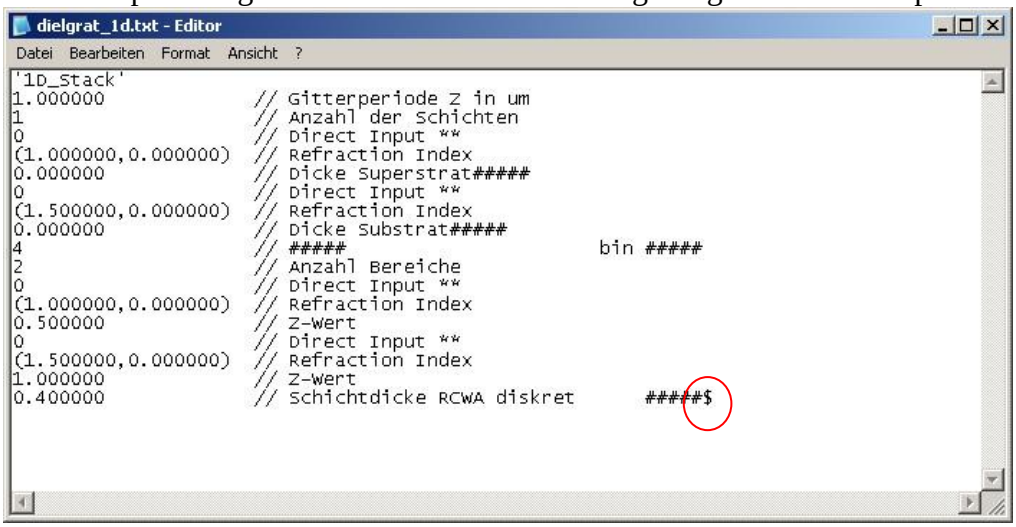


Fig. 3: Marking a geometry parameter for looping

The mark is recognized automatically and an additional radio button together with input field(s) pops up after saving the change and leaving the notepad editor. This is shown in Fig. 4. If the "/d" box is checked, than the loop parameters are related to the pitch, i.e., entering 0.2 at pitch 0.5 microns means 0.1 micron.

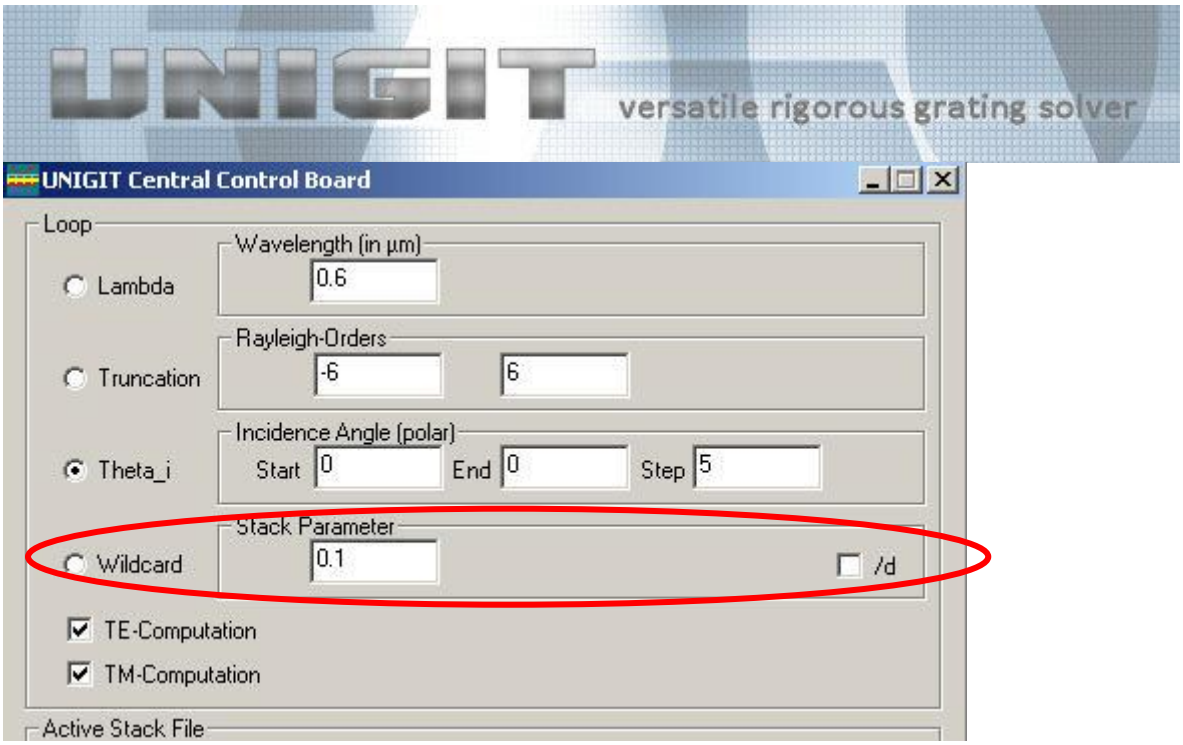


Fig. 4: Loop over a geometry parameter option occurring in the CCB

Furthermore, in case of classical diffraction the polarization of the incident light may be chosen (The selection fields are positioned at the same place where the radio button and the input fields for the azimuthal incidence angle occurs if conical diffraction computation is chosen). The light may have TE-polarization and/or TM-polarization.

In addition to these fixed (one) parameter loops, Unigit version 2.01.02 upwards permits a flexible loop a.k.a. batch processing. Here, the parameter variation is not limited to just one parameter. Instead, various combinations of certain parameters can be run. The flexible mode is activated just by clicking the associated check box as depicted in Fig. 5.

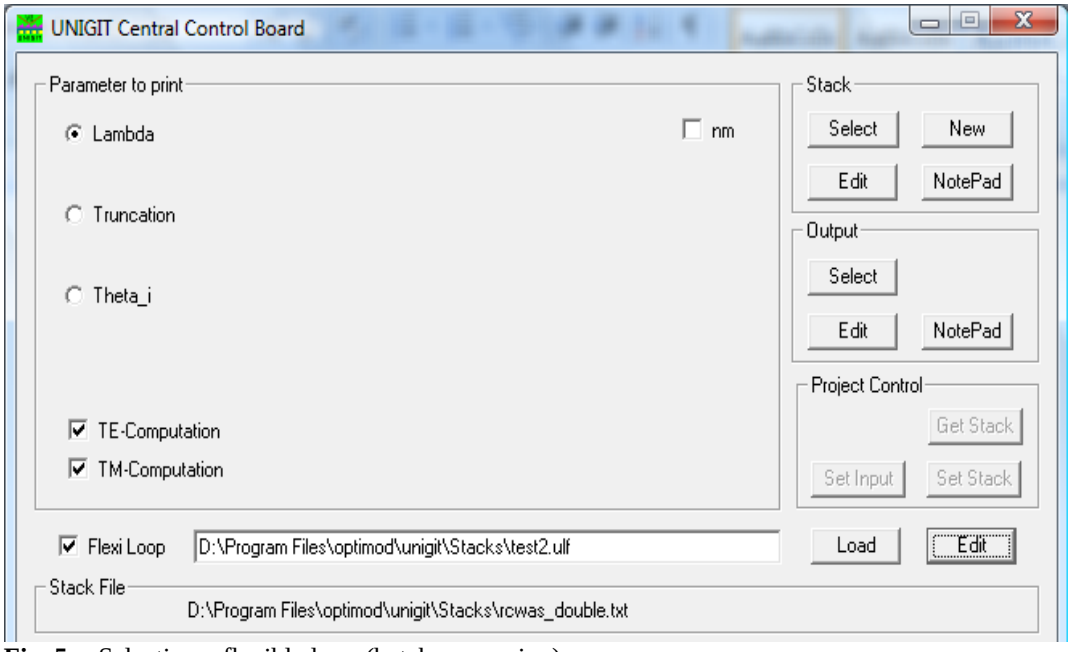


Fig. 5: Selecting a flexible loop (batch processing)

In order to tell Unigit how to do the loop, an Unigit Loop (Control) File – extension .ulf has to be selected. An example of an .ulf file is shown in Fig. 6. The parameters have to be listed in exactly the order to be seen in the figure. Furthermore, the total number of parameter sets has

to be entered in the second line (here 6). Beside of these constraints, the parameter values in each line (corresponding to a parameter set) are quite arbitrary within the range of physical values. Of course, single loops can also be specified but as opposed to the single loop selection, a variable step width can be invoked such as shown in the example of Fig. 6. Finally, the output parameter out of the set which shall be appear in the output file (parameter to be printed) can also be selected via the radio buttons (compare to Fig. 5).

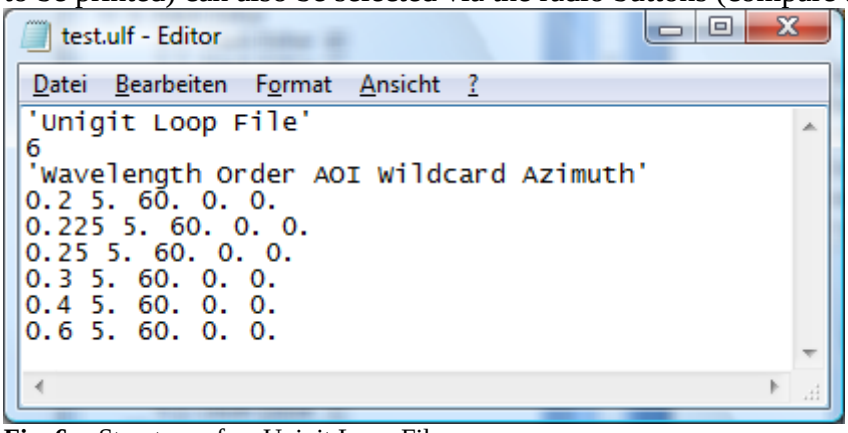


Fig. 6: Structure of an Unigit Loop File

Next, a stack file has to selected for the regular operation mode described here. This can simply be done either by clicking the SELECT button or by clicking the NEW button which are both located in the "Stack group". When a stack file is selected, the solver is automatically selected (see below). Moreover, the stack editor (see section 4) is adapted to the selected solver.

Eventually, the solver routine has to specified. This has to be done, of course, related to the selected stack file, i.e., you can run 1D RCWA based stack files only with a 1D routine (1D classical or 1D conical), similar applies for 2D. On the other hand, C-method based stack files can only be run by the "1C" routine[1].

There are four solver routines available:

- **1D classical, RCWA & Rayleigh-Fourier** for line space gratings in classical mount, i.e., azimuthal angle is equal to 0 and no polarization coupling occurs.
- **1C classical, C-method** for line space gratings in classical mount.
- **1D conical** for line space gratings in conical mount, where the azimuthal angle can be freely defined. Here, the polarization may couple between TE and TM and therefore the polarisations cannot be separated any more.
- **2D** for crossed gratings. These gratings are characterized by two spatial periods. When the two periodic directions include a 90 degree angle, it is called an orthogonal grating. Otherwise the grating is called non-orthogonal. Furthermore, the 2D grating can be run either according to Lifeng Li's algorithm /6/ or based on Lalanne's approach /7/. The selection is done by means of the checkbox "La". If checked, the Lalanne approach is asking for an weighing factor (factor α in equation 1b of /7/, it has to be between 0 and 1) as shown in Fig. 7. An additional way to accelerate 2D computations is to take advantage of inherent symmetries. This means that both the stack as well as the excitation have to be symmetric. Presently, this feature is only implemented for a symmetric orthogonal grating illuminated by a beam having a

[1] There are only one-dimensional stack files for the C-method

projection into the x-y plane parallel to the y-axis, i.e., for an incident azimuthal angle of 90 degrees. In order to activate the symmetry accelerator, the checkbox *Sym* has to be checked. The user has to make sure that the selected grating is symmetric. Otherwise, wrong computation output will result.

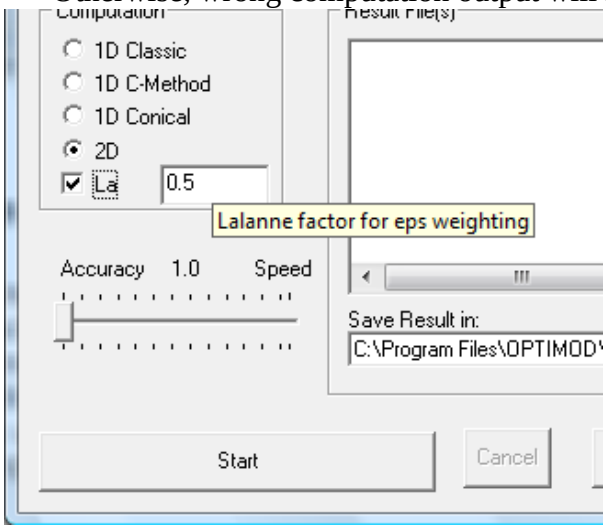
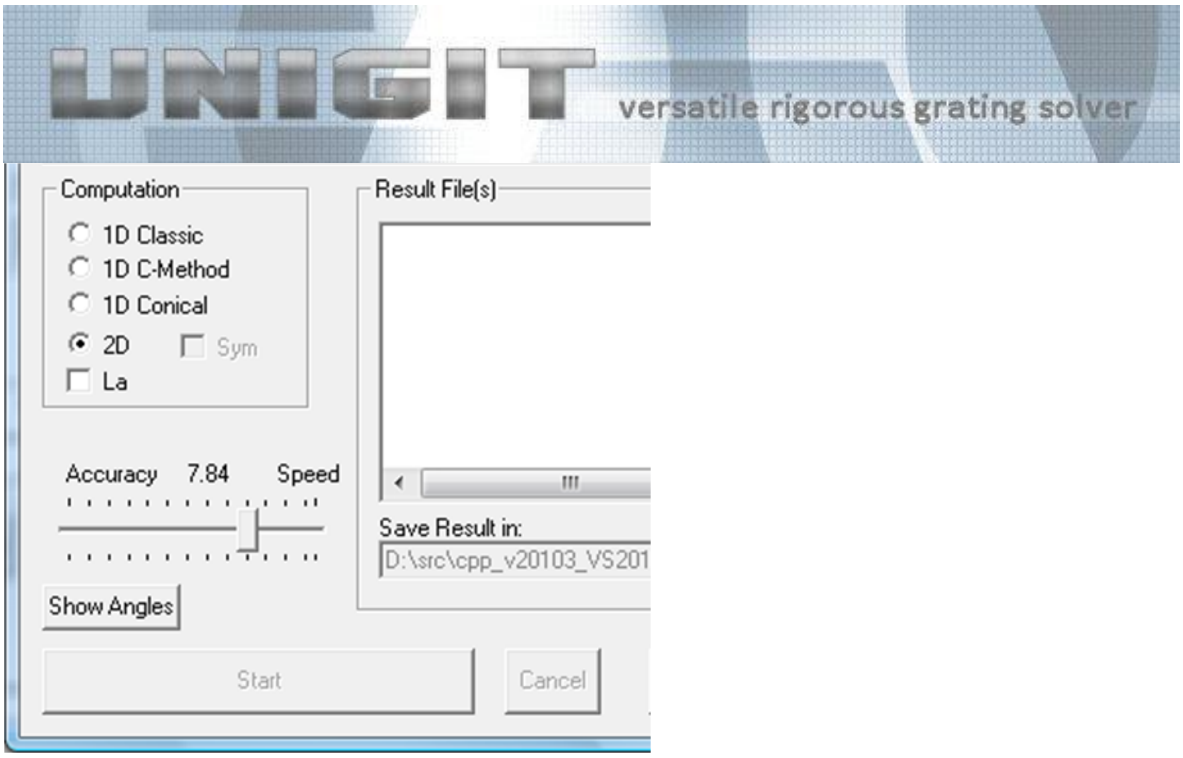


Fig. 7: Selecting the "Lalanne"-approach for 2D computations

The appropriate solution kernel (1D, 1C or 2D) is checked automatically when a stack file is selected. Furthermore, Unigit detects automatically whether the correct solver routine is chosen and issues an error message if this not the case.

While the symmetry usage is an exact approach that requires a symmetric setup, the smart order truncation is applicable to all 2D gratings. It is automized and all the user has to do is to move the accuracy-speed slider at the bottom of the CCB. The default is the value 1.0 or left position of the slider. It corresponds to standard 2D-RCWA as known from publications. Moving the slider to the right increases the speed but may reduce the accuracy. Values between 4 and 8 (not beyond) are recommended but it may differ from case to case (the example below shows 7.84). Approximately, these values correspond to the speed up. An appropriate setting can be verified by means of an extented convergence analysis. This means to run a regular 2D RCWA with very high order first and save the result as a reference. Then, several order convergence loops have to be run with different slider position (e.g. 1,2,4,6,8 and 10). Finally, all this results have to be related to the reference (which is supposed to be the correct result) resulting in a trade-off order-speed pair for a given accuracy.



2.3. Calculation of the Diffraction Angles

Basically, Unigit offers two different options for the calculation of the diffraction angles. The first quick option can be done upfront and does not need to run the RCWA- or C-method computation kernels. It can be launched by clicking the button SHOW ANGLES. The results are shown immediately in a Notepad window. An example is inserted in Fig. 8 for a dielectric grating ($n_{substrate}$ = 1.5) with pitch = 1 micron at λ = 0.5 and AOI = 45°.

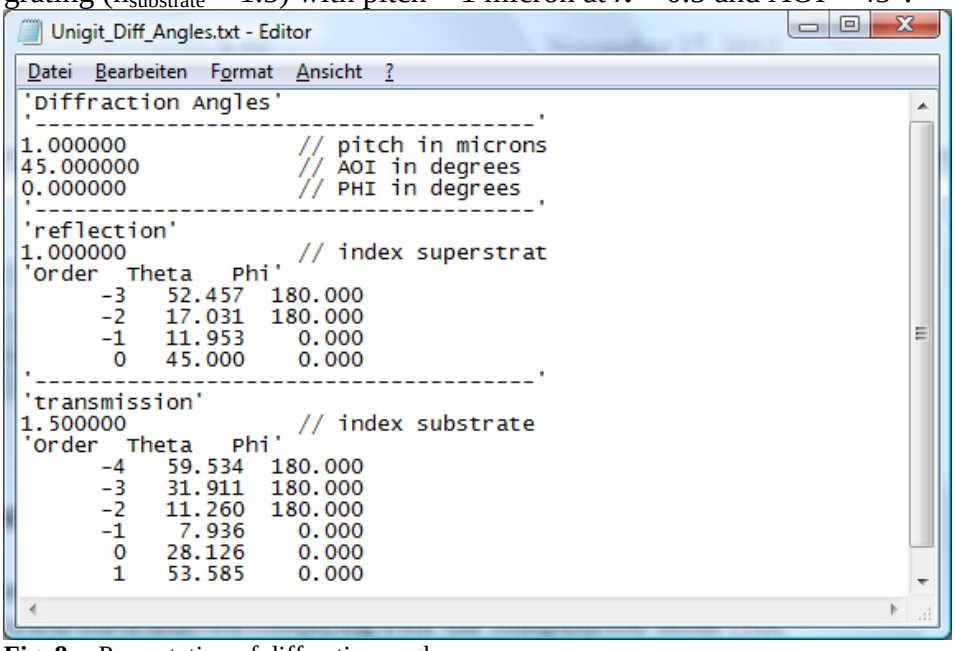


Fig. 8: Presentation of diffraction angles

In the second option, the angular information is a by-product of a full computation run. In order to access it, one has to generate an Unigit project file *.upr (see section 7.2). In order to retrieve the angular information from the project file, one has to reload the project file after the computation run has finished (see section 7.3) and pick the “diffraction angles” option in the output editor (see Fig. 48).

The angles are defined according to Fig. 9. The polar angles θ are measured between the normal axis z and the incident or diffracted ray. The azimuthal angles φ are measured between the projection of the ray into the x-y plane and the x-axis.

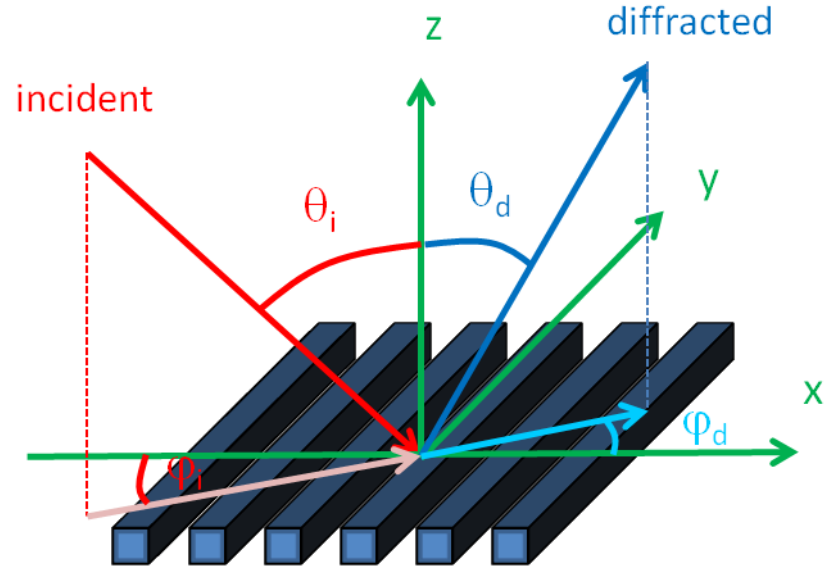


Fig. 9: Definition of the incidence and diffraction angles in Unigit

2.4. Computation Run

The computation is launched by clicking the START-button below. Before hitting the button, make sure that the wanted stack-file is selected complying with the computation mode (1D, 1C or 2D). Otherwise an error message will occur. During computation, a DOS window appears which may show additional information about the state of numerical calculations. The START-button will be replaced by a progress indicator. After having completed the task, the START-button appears again and the computation time is displayed above the button.

During computation, the progress is shown instead of the START button (see Fig. 10).

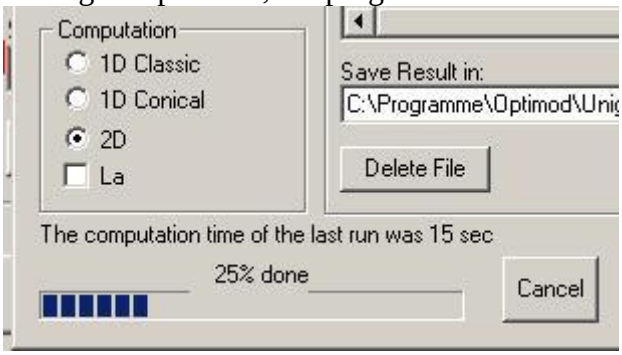
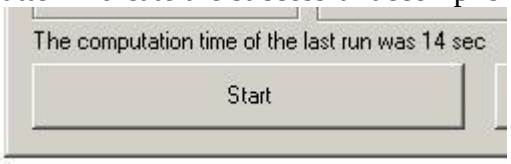


Fig. 10: The progress is shown during running a computation

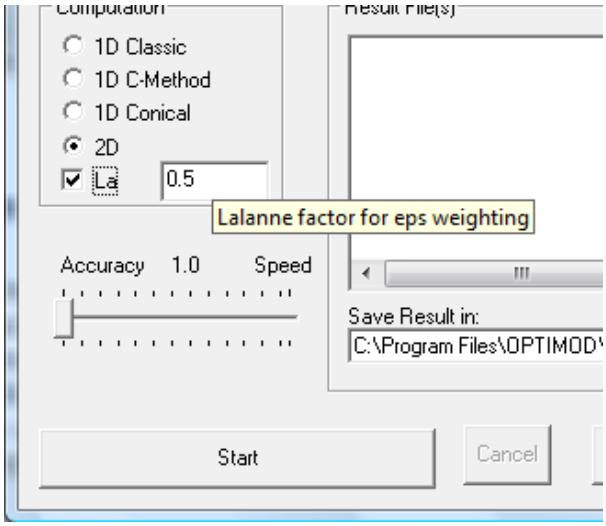
It is possible to stop the computation with the CANCEL button. If you have checked the "Show Info in DOS Window" check box or the "No output file" radio button in the output editor, a DOS window pops up during computation showing either intermediate auxiliary information or in the latter case the final results.

The display of the total computation time for the loop and the reappearance of the START button indicate the successful accomplishment of the computation run.



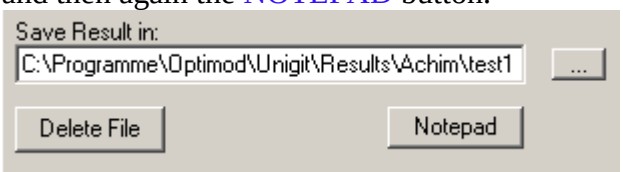
There are two particular features for the 2D solver:

- It can be chosen between the Li "staircase" or the Lalanne "mixing" approach (check the LA-box) for the Fourier-transformation of the 2D grating,
- an accelerator for 2D crossed gratings with speed up factor up to 4 ... 8 w/o accuracy loss (trade secret of Optimod) can be activated by moving the slicer (then, the speed factor appears).



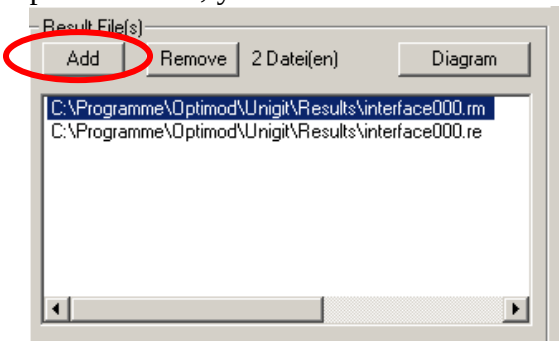
2.5. Result Presentation

Depending on your choice in the output editor, the results are either displayed in the DOS window (and can be copied and pasted from there directly) or they are written into file(s).
When written into one file (selection "All in one") it can be viewed immediately after the computation is finished in a Notepad window. To this end, you need just to call it by the NOTEPAD button. The corresponding part of the central control board is shown below. In order to retrieve other results, you can utilize the request button ("…") to search for the file and then again the NOTEPAD button.

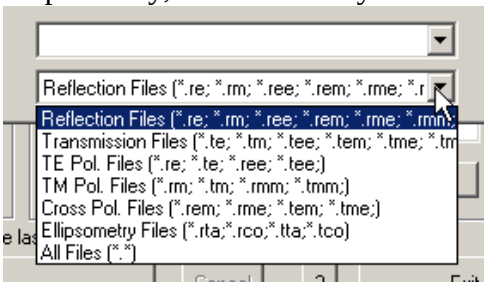


Another gadget of the Central Control Board is the graphical interface. It is designed to present the results if the "Single Files" option was chosen for output.

In order to view the results in the graphical window, you first have to select which data to present. Here, you need to click the ADD button (see below).



After clicking, a file selection dialog opens and you can select the result files of your choice. A preselect ion for the file type, i.e., reflection, transmission, TE, TM, Cross Pol. or ellipsometry, can be done by means by means of the file type drop down box.



After the file selection is made, the file name will appear in the related list (see above) and the graphic can be started by clicking the DIAGRAM button. The example for the dielectric interface is shown in Fig. 11.

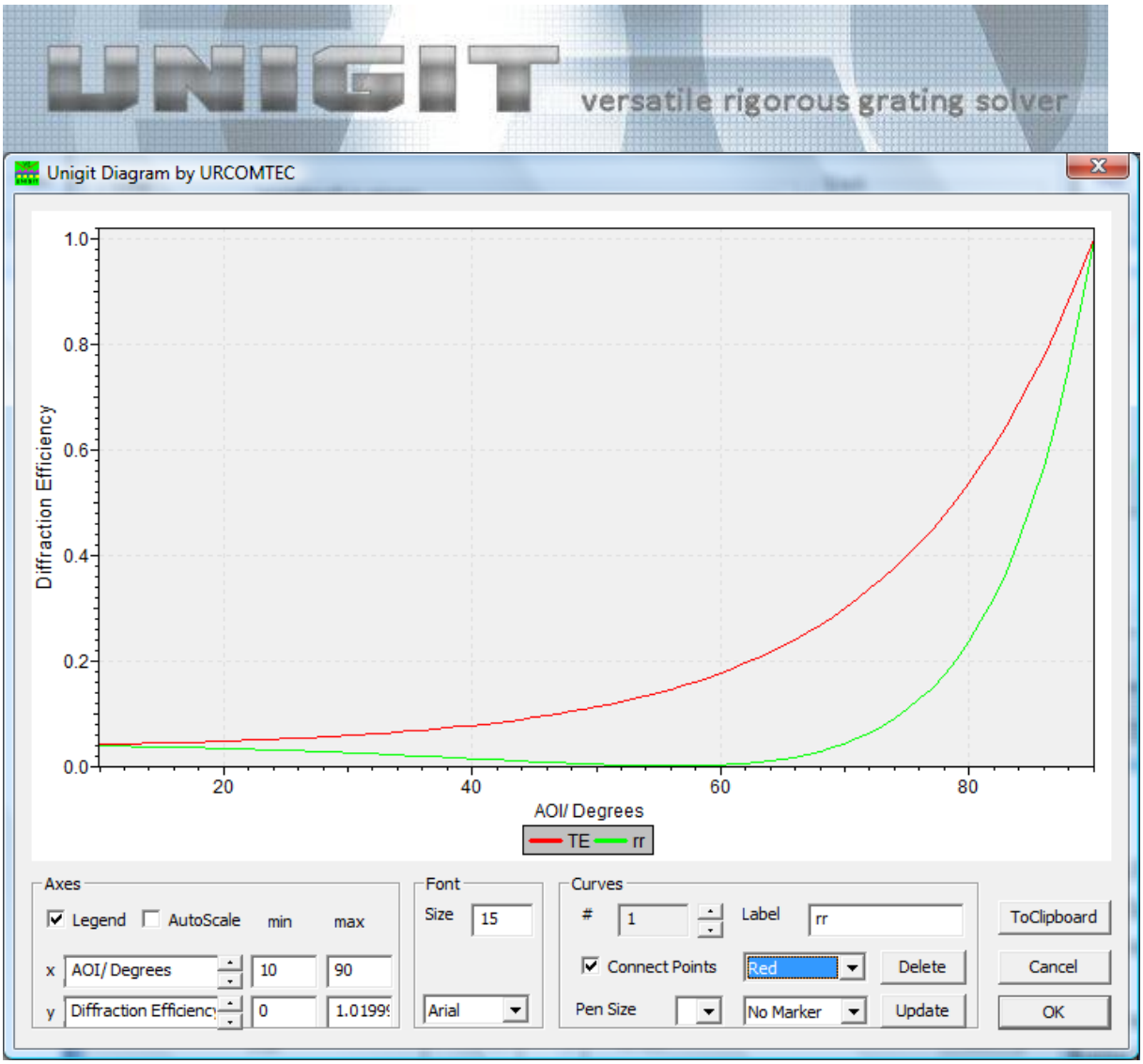


Fig. 11: Unigit diagram plot of computation results

Unigit version 2.01.02 upwards features a new diagram tool provided by UrcomTech Berlin/ Germany. It is based on an activeX library and has to be registered before usage.

The implemented diagram tool offers the following options to modify the plot:

- Switching the legend on/ off by checking/ unchecking the associated box,
- Autoscale the diagram to min/ max range by checking/ unchecking the associated box,
- Scale the diagram by entering min/ max values for x and y (axes group),
- Change the font type and size (font group),
- Select the active curve number # by means of the up/down arrows (curves group),
- Change the label name of the active curve by entering the new name into the label edit field and push the *Update* button,
- Change the color of the active curver by means of the color selector,
- Change the marker type of the active curver by means of the marker type selector,
- Change the pen size of the active curver by means of the pen size selector,
- Delete the active curve by means of the *Delete* button,
- Connect/ Disconnect curve points by checking/ unchecking the *Connect Points* box,
- Note: all changes in the curves group have to be activated by means of the *Update* button,
- Save the graph to the clippboard for further use.

2.6. Short Cut

An important extension of UNIGIT version 2.01.01 upwards is the introduction of UNIGIT projects (file extension ".upr"). When activated in the output editor (see 5.2), the complete input and result information of a UNIGIT run is stored in the project file. Later, all information can be restored from this file. What's more, all kind of results related to the application case can be generated from it with few exceptions saving time consuming reruns. For a detailed description it is referred to section 8 of this manual.

2.7. Hot Keys

For faster access, there are two hotkeys implemented in the CCB:

- SELECT STACK CTR-S,
- EDIT STACK - CTR-E,
- OUTPUT EDITOR CTR-O,
- NEW STACK CTR-N,
- DIAGRAM TOOL CTR-D,
- ADD FILE 2 Diagram CTR-A,
- SELECT RESULT FILE CTR-R,
- SETTINGS EDITOR CTR-P,
- CANCEL UNIGIT CTR-BREAK,
- EXIT UNIGIT CTR-X,
- RUN - ALT-R.

2.8. Pull Down Menue

In addition, many functions can be accessed through the pull down menue of the CCB. The following groups and functions are implemented:

- File
  - Load Stack: loads an existing stack file,
  - Load Output: loads an existing output control file,
  - Result File: opens an explorer for entering the result file,
  - New Stack File: opens the stack editor to create a new stack file,
  - Exit: exits the Unigit program.
- Edit
  - Edit Stack File: opens the stack editor to edit the selected stack file,
  - Edit Out Ctr: opens the output editor to edit the output control file.
- Project
  - Get Stack:
  - Set Stack:
  - Set Input: retrieve the input information from a Unigit project file.
- Misc
  - Parallel: opens the parallel loop editor,
  - Show Angles: computes and displays the diffraction angles,
  - Add Curve: adds a result data set to be displayed as diagram,
  - Remove Curve: removes a selected curve to be displayed as diagram,

- Diagram: starts the diagram tool.

2.9. Python Link

Python scripts can be directly run from Unigit by clicking the PYTHON button to the left of the SHOW ANGLES button. The Python script to be run can be selected in the general settings (see section 3.5). Examples of Python scripts that show how to call Unigit are provided on request.

3. General Settings

The general settings can be accessed via “Misc – General Settings” in the pull down menu of the CCB. An example of the settings window is shown in Fig. 12.

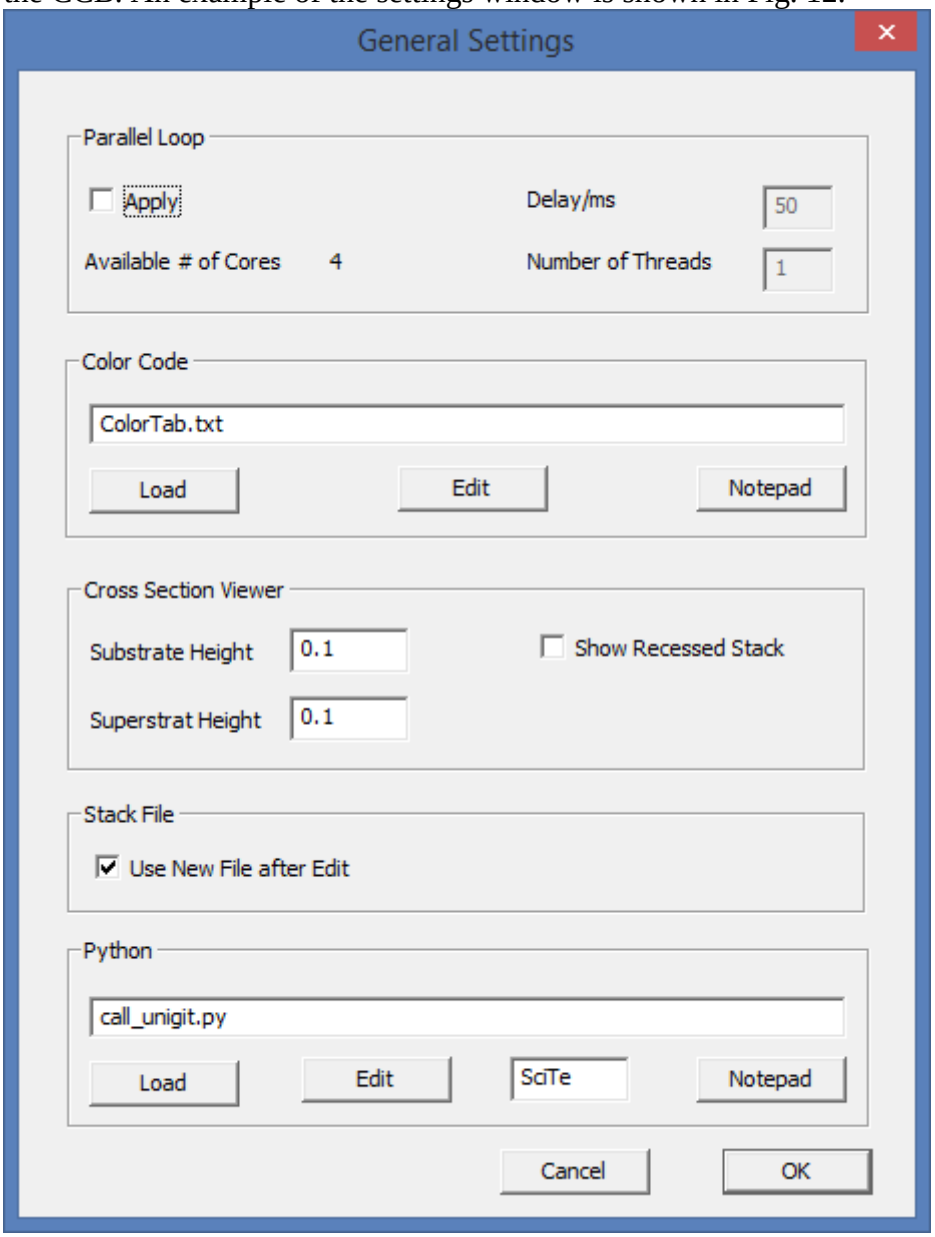


Fig. 12: General Settings Table

3.1. Parallel Editor

The former parallel editor has been replaced by the new general settings in order to accommodate more controls for Unigit runs and presentations. It can now be found as a group box named "Parallel Loop" on top of the settings window.

First of all, the choice whether or not to take advantage of the parallel loop threading has to be done by checking (or unchecking) the check box *Parallel*. Presently, the parallel looping does not allow to output the computation results in single files (ready to be used with the diagram

tool). Moreover, it may show synchronization issues (fail of single threads, e.g. loop variables) when Unigit project files are to be written. In addition, one should be aware that Unigit generates temporary files during parallel looping which are deleted afterwards. So, the simulation run may be vulnerable to conflicts with the file system (issues caused by missing user privileges).

After activiation of the parallel looping mode, two edit fields become active – the number (parallel) threads and the delay time between the various thread launches. An arbitrary integer number greater zero should be entered for the number of threads. The default is the number of available computation cores. However, the entered value can be different. It is recommended to find an optimum number (to achieve maximum computation speed) by means of numerical experiments. The default value for the delay time is 50 ms. It is recommended to increase this value if synchronization issues occur, i.e., single process threads fail.

3.2. Color Table

The color tab settings are closely linked to the color coding of materials and n&k files. In the second group box of the general settings called "Color Code", a color table file can be loaded and edited. The color table itself is a simple Ascii-file that contains a list with pairs of color-ID's (names) and assigned RGB color reference values. The table can either be edited by Notepad (or another Ascii editor) or by means of the built in color table editor which can be called via the EDIT button (see Fig. 13).

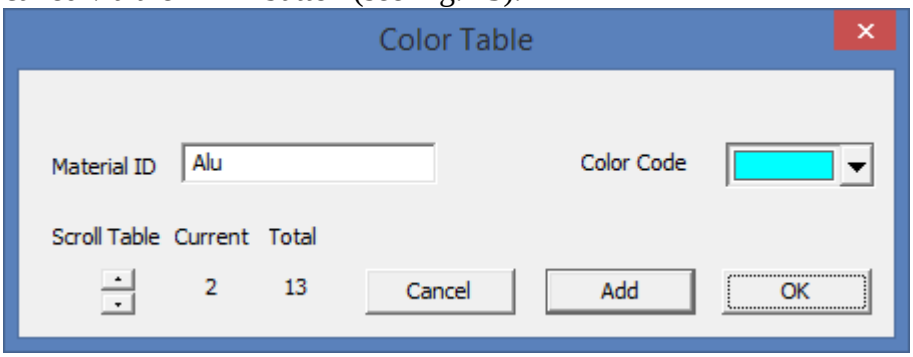


Fig. 13: Color Table Editor

The color table editor enables the viewing, changing and adding of color assignments. In order to add an entrance, fill the material ID edit window with a new name of your choice and select a color by clicking the down arrow next to the color code . Finally, append the new pair by clicking the ADD button or discard your choice by doing anything else. After adding a new pair the total score will be increased by 1. Currently, the elimination of an entrance is only possible via the Notepad editor.

3.3. Cross Section Viewer Settings

While the colors of the cross section view are controlled by the color table in conjunction with the n&k settings, the overall appearance can be controlled by the settings in the cross section viewer group box. There are three settings. The substrate height defines the relative height of the substrate shown in the plot in relation to the total stack height. Similarly, the superstrat height defines its relative height, however, since the superstrat is not color coded - it is just the relation of the unfilled area above the grating. Eventually, the check box "Show recessed stack" enables to recess the stack relative to the substrate.

3.4. Stack File Settings

The only one setting of this group box gives the user the choice how to proceed after editing and rename the active stack file. Previously, Unigit has kept the old file as active file (the one that is run when hitting the start button). Because this has lead to some misunderstanding, the user has now the option to decide whether or not to have the new edited stack file as active file. The default setting is to use the new file (box is checked).

3.5. Python Settings

Unigit offers the possibility to run Python scripts directly from the CCB (see section 2.9). The active python script file can be loaded and edited via the buttons LOAD and EDIT. The Python editor that is called can be verified in the edit box next to the button.

4. Stack Editor

4.1. Stack Editor 1D

This editor is devised to facilitate the assembling of the patterned multilayer stack. It is launched by clicking either the EDIT or NEW button in the central control board (group field stack). The basic dialog of this editor is shown in Fig. 14.

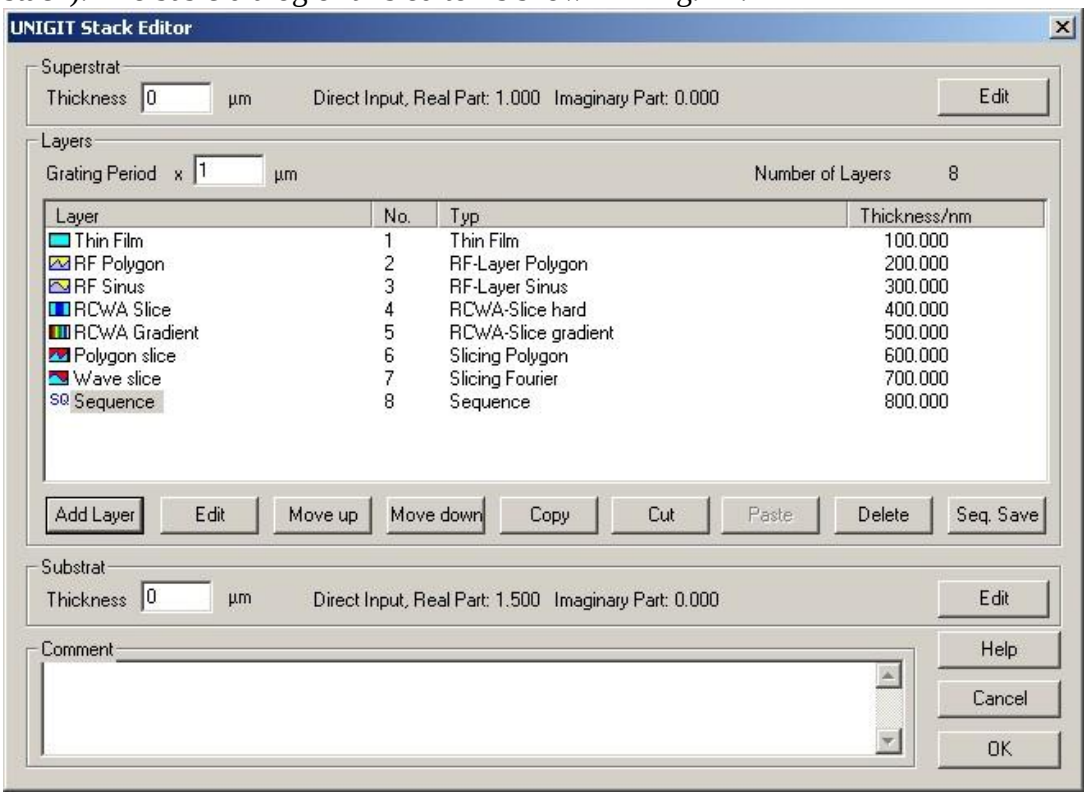


Fig. 14: Unigit Stack Editor for 1D-stacks

The stack editor facilitates the assembling of arbitrary multilayer grating structures. To this end, a number of buttons are arranged below the spreadsheet list.
There are several feasibilities to assemble the stack.

· First, new layers can be inserted by starting the layer editor clicking the INSERT button.
· Second, an existing layer can be edited by clicking the EDIT-button. This action also launches the layer editor with the selected layer. In order to select a layer, the cursor has to be positioned at the beginning of the associate description line.
· Third, the order can be changed by moving a marked layer up or down.
· Fourth, a layer can be deleted by marking it and press the DELETE button.

· Moreover, there are additional options coming with version 2.XX.XX.
· Layers can be marked by clicking with the left mouse button and concurrently holding the CTRL or SHIFT key and saved as a sequence file. The marked layers must not necessarily form a coherent string, i.e., unmarked lines are permitted between marked lines.

· Marked layers can be copied or cut and later pasted (this is also possible between different stacks for the information is stored to the clipboard). For instance, HL(high low index)-quarter wave stacks can be built in this way.

Basically, a multilayer comprises an arbitrary number of different layers embedded between substrate and superstrat.

For 1D (line gratings) there are 5 basic types of layers:

· homogeneous flat layer (type 1),
· sinusoidal Rayleigh-Fourier layer (type 2),
· polygonal Rayleigh Fourier layer (type 3),
· discrete RCWA layer (type 4) and
· continuous RCWA (i.e., gradient) layer (type 5).

In addition to these basic layer types, so-called composite layers can be input:

· polygonal RCWA layer (type 6),
· sinusoidal RCWA layer (type 7) and
· sequence of layers, comprising an arbitrary number of layers from type 1 through type 7 in an arbitrary order.

As opposed to the sinusoidal or polygonal Rayleigh-Fourier (/3/) layer, the corresponding composite layers are automatically decomposed (that means sliced) into discrete RCWA slices of type 4 immediately before being processed. A sequence of layers (also called sub-stack) has to be stored as a file in the same way as the total stack description is stored. The only difference between a complete stack and a sub-stack consists in that the complete stack comprises beside the stack data the grating period as well as the superstrat and substrate description. Moreover, a hierarchy of sequences is not permitted, i.e., a sub-stack must not contain another sub-stack.

In order to demonstrate the various options of the editor, an example of a rather artificial layer stack is shown in the picture above. In this example, the stack is embedded between air with a fixed index of 1 and aluminum specified by the nk-file (see refraction index editor).

Beside the refraction indices, the superstrat and the substrate are also defined by a "thickness", i.e., the position inside the material where the efficiencies are measured. These values make sense if the phase information shall be output or if the materials are absorbing.

The period of the stack has to entered into the grating period x field (upper left of the group field "stack"). Being in the 2D computation mode a second input mask for the period in y direction (orthogonal to x) is provided automatically.

4.2. Stack Editor 1C

This editor enables the assembling of stack files for the C-method. An unlimited number of non-parallel interfaces is permitted. The stack editors considers the interface as a “layer” just like in the same way as it does for Rayleigh-Fourier interfaces. This arises from the layer-wise

processing of RCWA-slices. After entering one layer, the ADD LAYER button is disabled. Furthermore, the stack editor enables only three choices:

· homogeneous flat layer (type 1) (which is not useful, i.e., the simulation of flat interfaces or thin film stacks should be done with the "1D" routine – see 2.1),
· polygonal C-method "layer" (meaning interface),
. sinusoidal C-method "layer" (meaning interface).

The basic dialog of this editor is shown in Fig. 15. Similar icon as for the Rayleigh-Fourier layers are used. The "layer" types are uniquely identified by its names either "CM-Layer Polygonal" or "CM-Layer Sinus". Beside the corrugation of the layers, the distances and if existent the material ID's are shown. Note, that there is now distance entrance for the uppermost interface (because it would have no meaning).

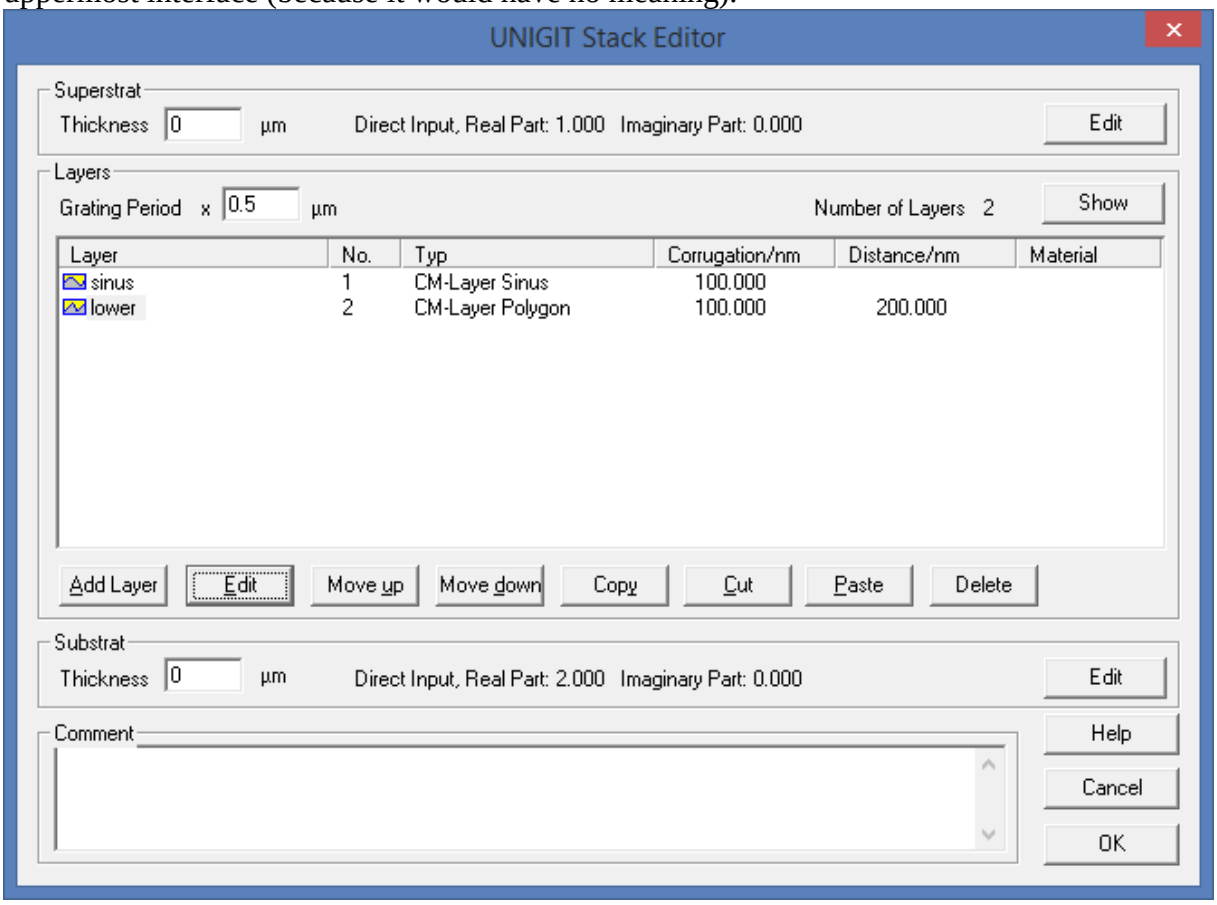


Fig. 15: Unigit Stack Editor for 1D-stacks for C-method

4.3. Stack Editor 2D

This editor is devised to facilitate the assembling of the patterned multilayer stack for a two-dimensional (or crossed) grating. It is launched by clicking either the EDIT or NEW box in the central control board (group field stack) if the radio button "2D" is specified in the central control board. The basic dialog of this editor is shown below. There are two essential differences compared to the 1D case:

First, the period in y and the non-orthogonality angle zeta have to be specified in addition to the grating period in x. Note, zeta has to be put to zero for the standard orthogonal cell.

Second, there are different layer types available in 2D. The stack editor for a 2D grating is shown in Fig. 16.

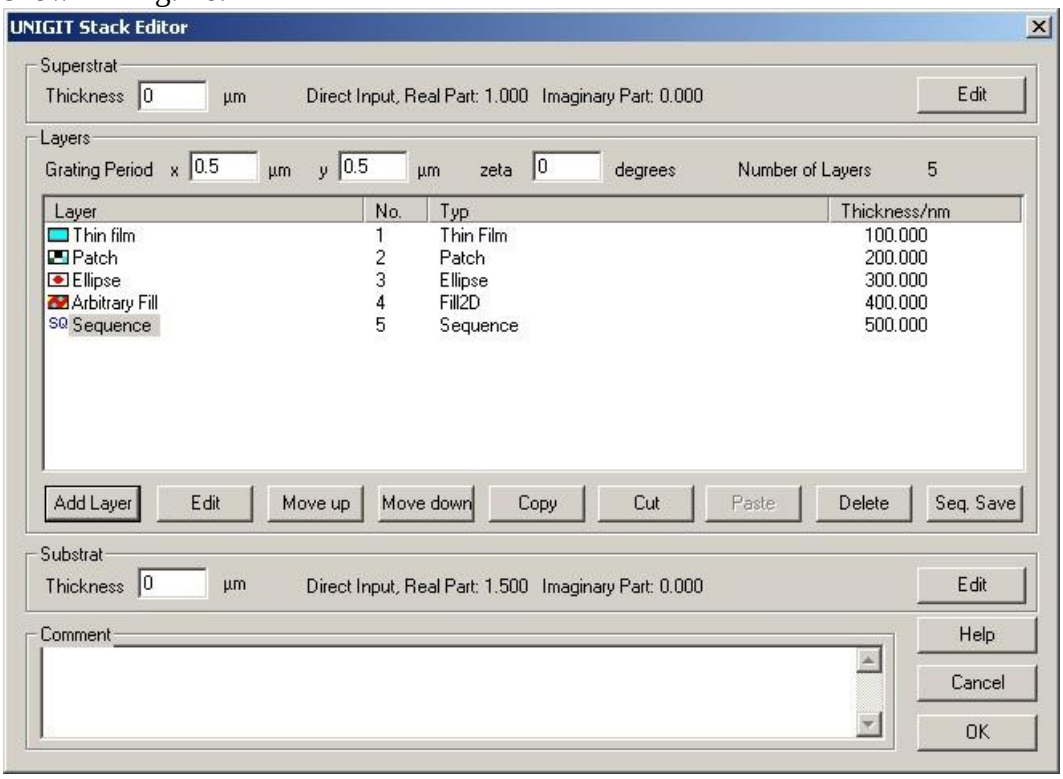


Fig. 16: Unigit Stack Editor for 2D-stacks

For 2D (crossed gratings) there are 4 basic types of layers:

- homogeneous flat layer a.k.a. thin film (type 1),
- patch (rectangle) layer (type 2),
- Super Ellipse layer (type 3),
- Arbitrary fill layer (type 4),

Like in 1D, sequence layers can be build. An example of the stack editor is shown in the figure above.

Rayleigh-Fourier layers are not available for the 2D case.

However, there is an option available to describe 2D composite layers, i.e., real 3D-structures. This can be achieved via the CONE_3D option in the layer editor (see section 5.3.5).

4.4. Hot keys

There are some hotkeys available in the stack editor to make life easier, namely:

- ADD LAYER - CTR-A,
- EDIT LAYER - CTR-E,
- MOVE UP - CTR-↑(cursor up),

- MOVE DOWN - CTR-↓(cursor down),
- COPY - CTR-C,
- CUT - CTR-X,
- PASTE - CTR-V,
- DELETE - Delete key,
- SAVE SEQUENCE - CTR-S.

4.5. Cross Section Viewer

In order to show the multi-layer grating as a whole, Unigit offers a new feature called cross section viewer. The cross section viewer can be started after having loaded or created a valid stack file from within the stack editor by clicking the SHOW button (see Fig. 17). Currently, it is only available for 1D gratings.

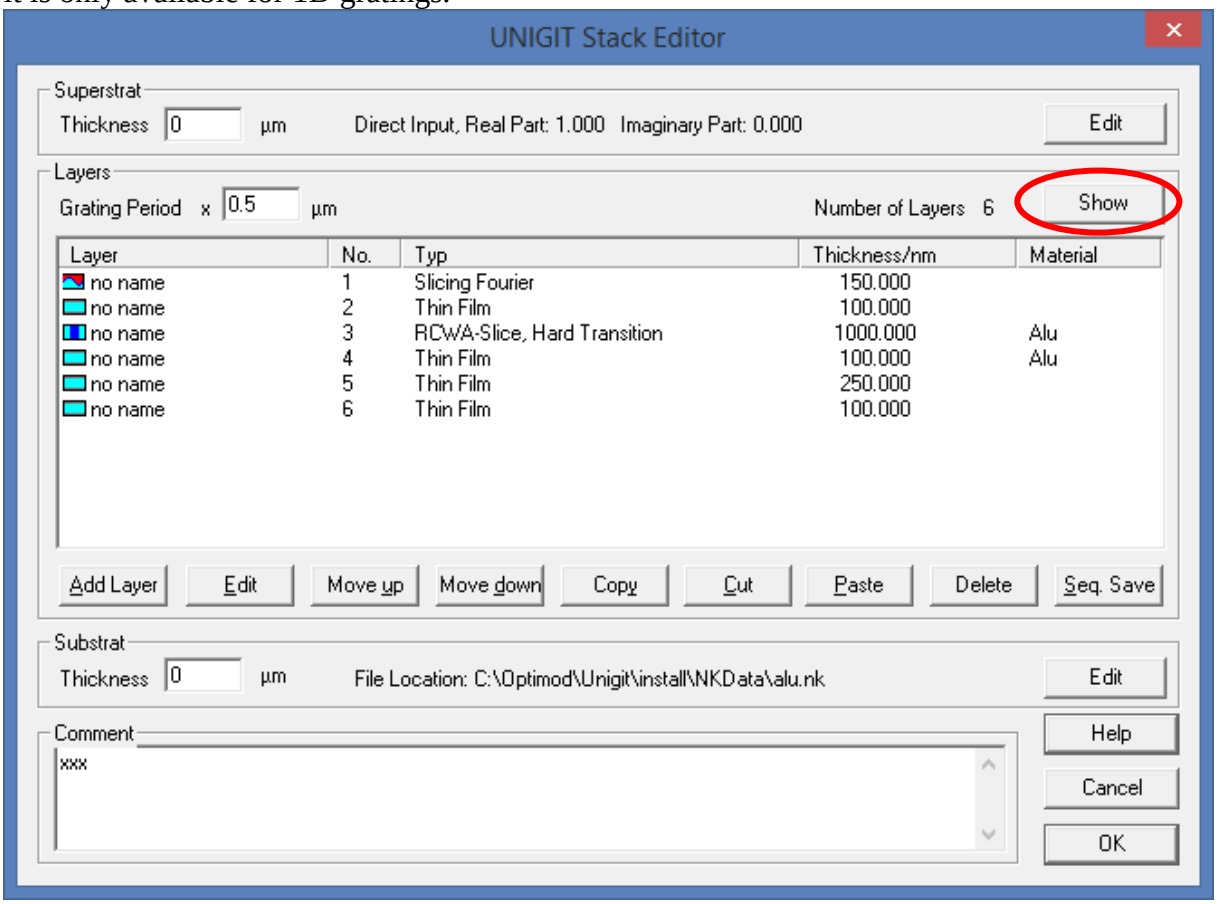


Fig. 17: Stack Editor with Show button (starts the cross section viewer)

It works both for RCWA as well as C-method gratings. Some examples are shown below.

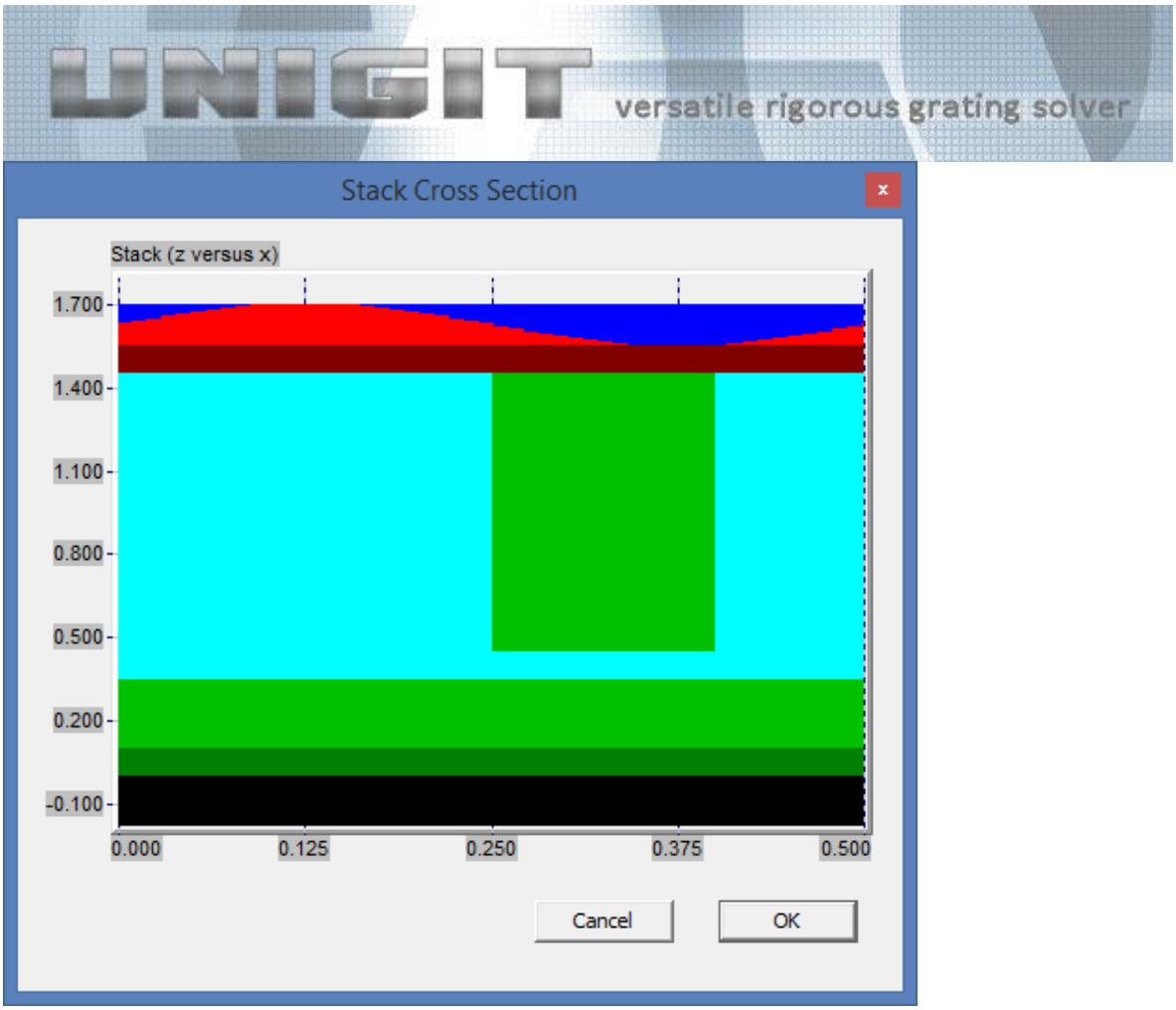


Fig. 18: Cross section of a RCWA-grating

Example 1 (shown in Fig. 18) visualizes the stack as given in the stack editor in Fig. 17. Obviously, it is an RCWA example. As can be realized from the sinusoidal layer on top (red), the slicing is fully included. In contrast, a C-method example is given below. Fig. 19 shows the cross section of the stack in Fig. 15.

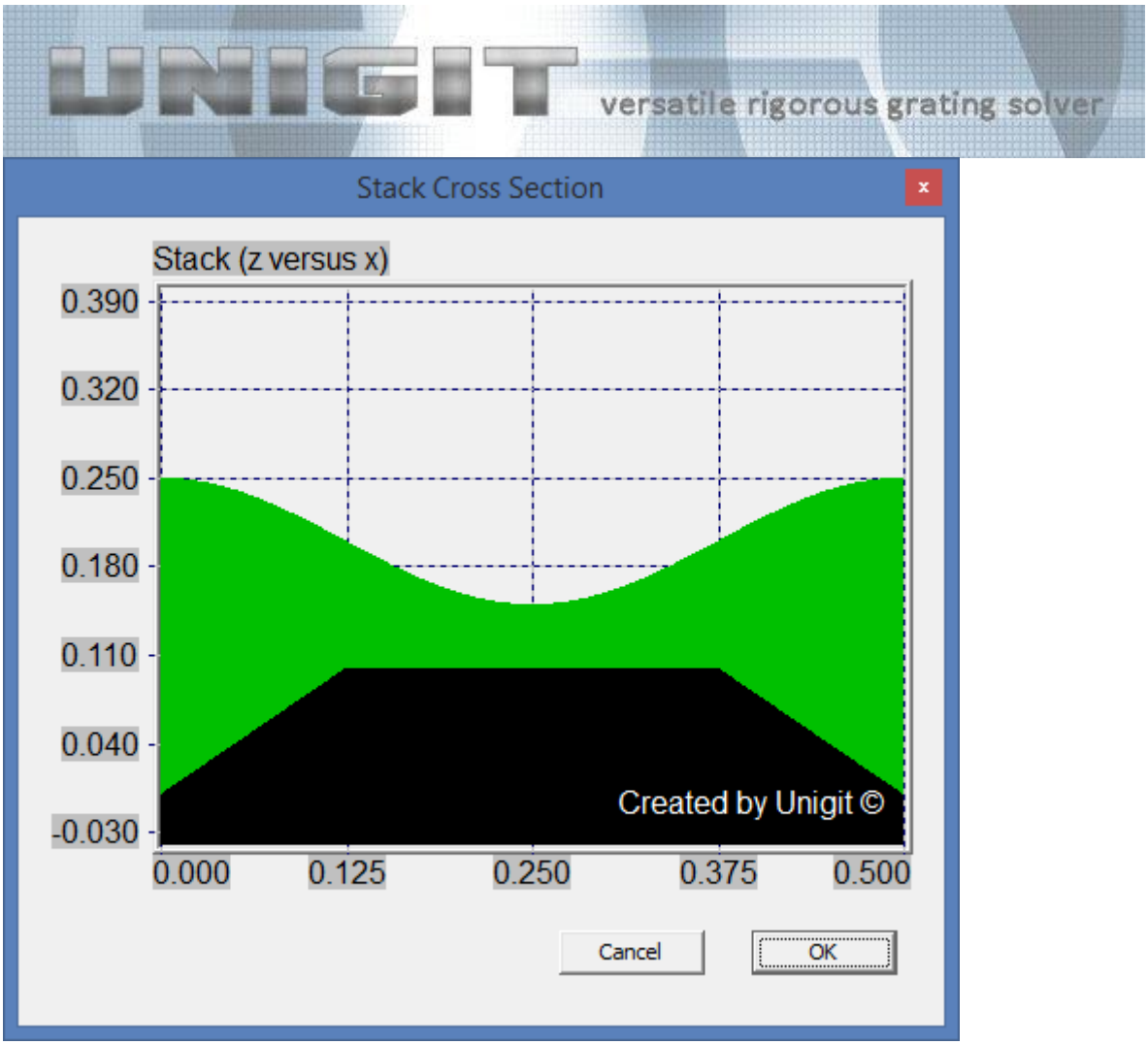


Fig. 19: Cross section of the stack as listed in fig. 15

5. Layer Editor

5.1. Layer Editor 1D (RCWA & Rayleigh-Fourier)

The layer editor can be activated from within the stack editor. Its appearance and functionality depends on whether the 1D (classical or conical) or 2D option is selected.

If activated by the INSERT button within the stack editor the layer editor will appear as shown in the screen shot of Fig. 20.

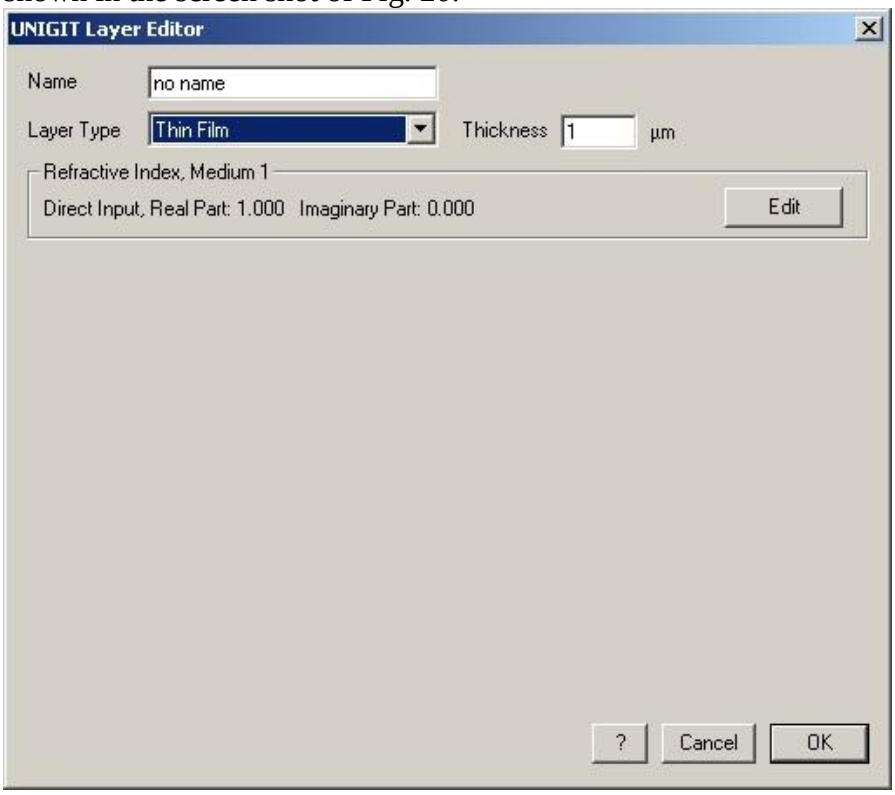
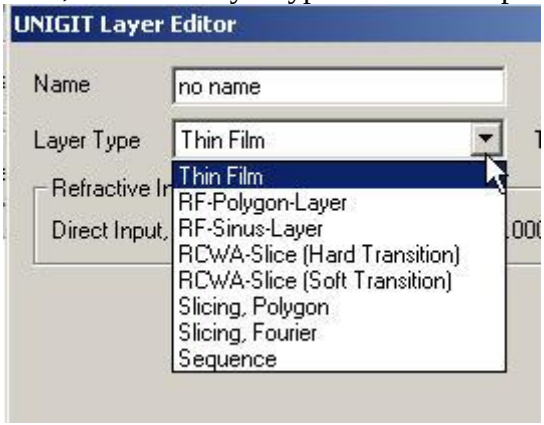


Fig. 20: Layer editor for 1D layers (general)

First, select the layer type from the drop down box as shown below.



Depending on the selected layer type, the layer editor shows up appropriately.
In the 1D case, there are 8 options available:

- Thin Film
- Rayleigh Fourier Polygonal Layer
- Rayleigh Fourier Sinus Layer
- RCWA Slice (Hard Transition)
- RCWA Slice (Soft Transition)
- Composite Layer (Slicing Polygon)
- Composite Layer (Slicing Fourier)
- Sequence

5.1.1. Flat Homogeneous Layer

If the intention is to insert a flat homogeneous layer (thin film), the only thing left to do is to specify the thickness of the layer (it has to be entered in microns) and to the refractive index by means of the refraction index editor. Moreover, a name can be attached to the layer. An example is shown in Fig. 21.

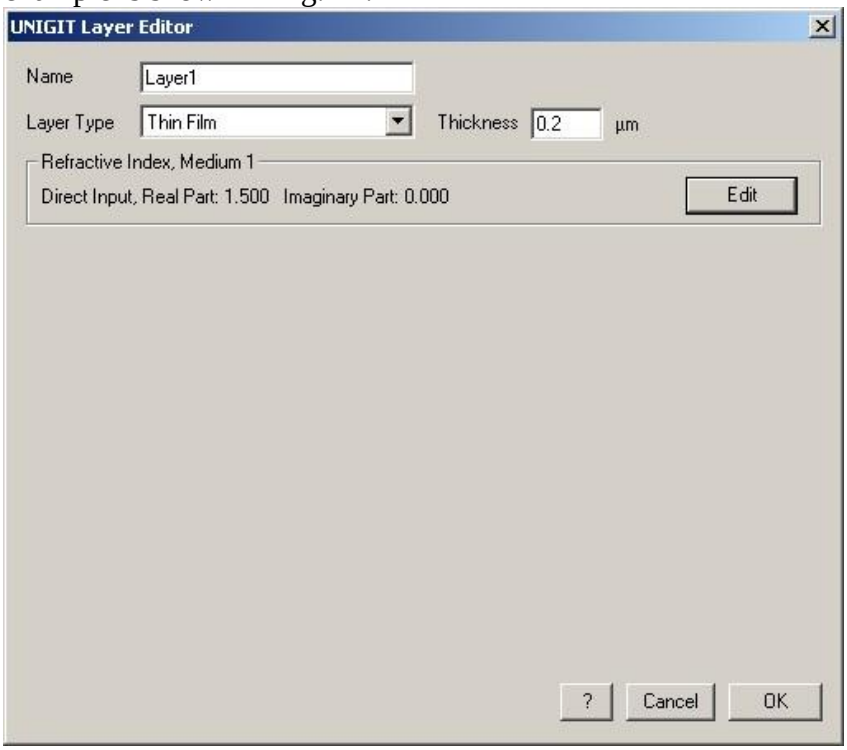


Fig. 21: Layer editor for 1D thin film layers

5.1.2. RF Polygon Layer

This type of layer as well as the RF Sinus layer is backed by the so-called Rayleigh-Fourier approach. The input scheme of the layer editor is depicted in Fig. 22.

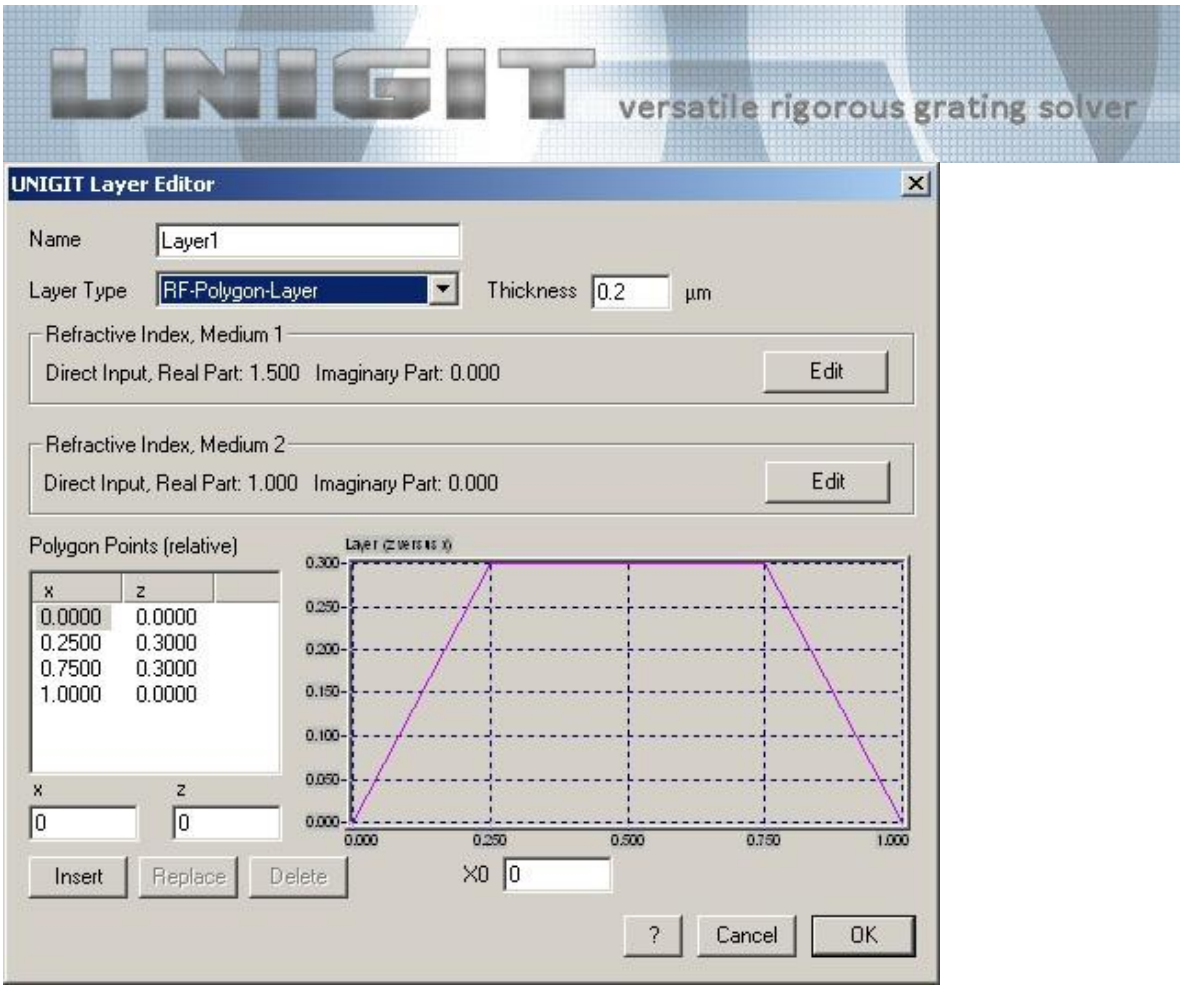


Fig. 22: Layer editor for 1D RF Polygon layers

A polygonal RF-layer is characterized by two materials that are separated by a polygonal interface. The specification of this layer type is as follows:

· First, one has to specify the refraction indices of the materials above (medium 1) and below (medium 2) the interface by means of the respective refraction index editor.
· Second, the thickness of the layer has to be fixed. In the context of a RF layer, the thickness corresponds to the absolute peak-to-valley value of the profile. Actually, the entered profile is renormalized with this value.
· Next, this interface profile has to be defined. To this end, the polygon points have to be entered in the "Polygon Points" field. This can be done by means of the two input fields labelled with x and z below (notice that x is the abscissa and z the ordinate) and pressing either INSERT or REPLACE. Of course, existing polygon points can also be deleted again. While inputting the profile points one must not care about the absolute geometry. Only the relative values have to be tuned to each other. In addition, the whole profile can be shifted laterally by specifying the phase x0. In the example above a trapezium interface was defined. Of course, all kinds of profiles including undercuts can be defined by the suitable choice of the points and their order. An example is shown below. It was realized by inserting a point (0.5,0.) after the third point of the previous trapezoid profile. An example is shown in Fig. 23.

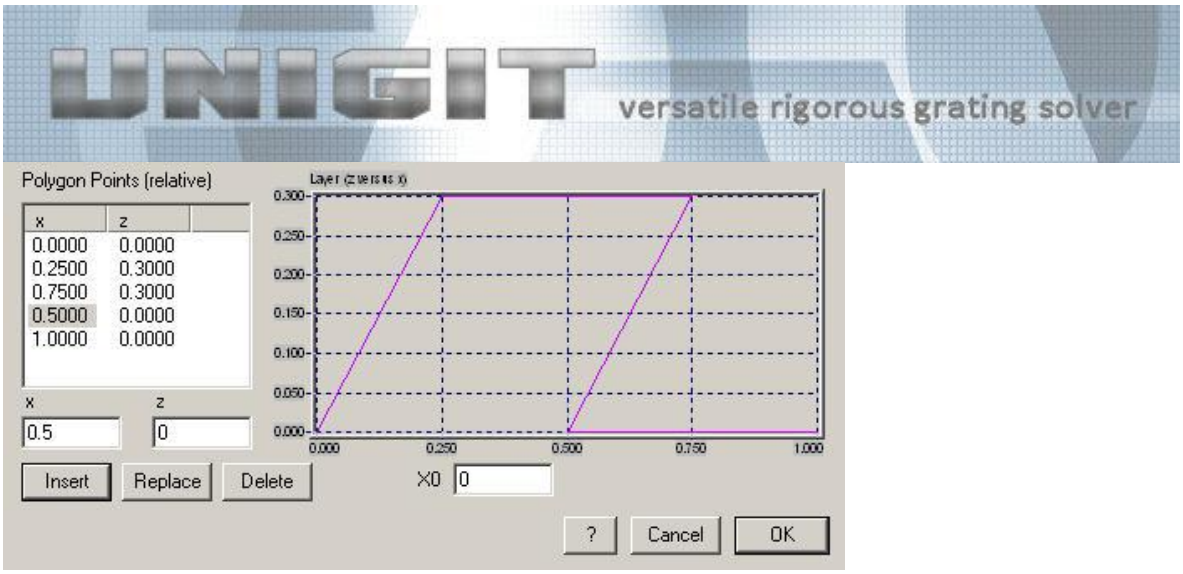


Fig. 23: Example for the input of a polygon layer

5.1.3. Rayleigh Fourier Sinus Layer

The RF sinus layer resembles widely the RF polygonal layer except for the interface between the two media. The layer editor window appearance is shown in Fig. 24.

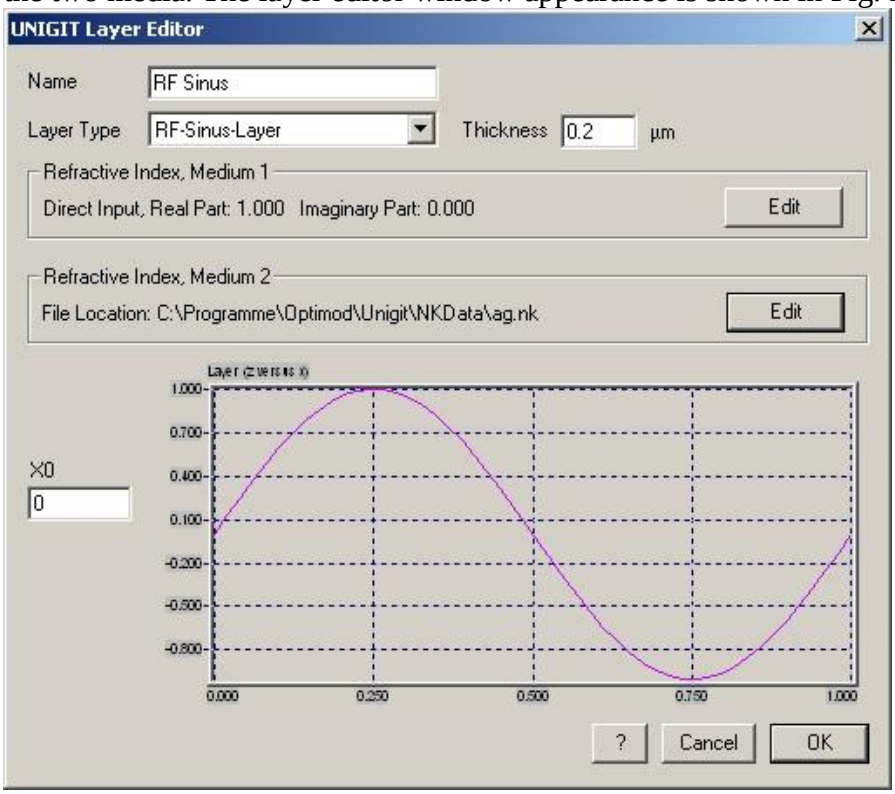


Fig. 24: Layer editor for 1D RF Sinus layers

Here, the input procedure is the same for step 1 and 2 like in the RF polygon layer. The profile is determined by the sinus function. The lateral shift can be adjusted by means of the phase x0 (a value of 0.25 defines for example a negative cosine function).

5.1.4. RCWA Slice (hard transition)

This slice type is a sort of universal work horse of the UNIGIT code. In principal, all kinds of patterned multilayer stacks including all kinds of layers (even inhomogeneous ones = volume scatterers) that are separated by all kinds of interfaces (including of course undercut profiles) can be constructed by means of it. The related layer editor is shown in Fig. 25.

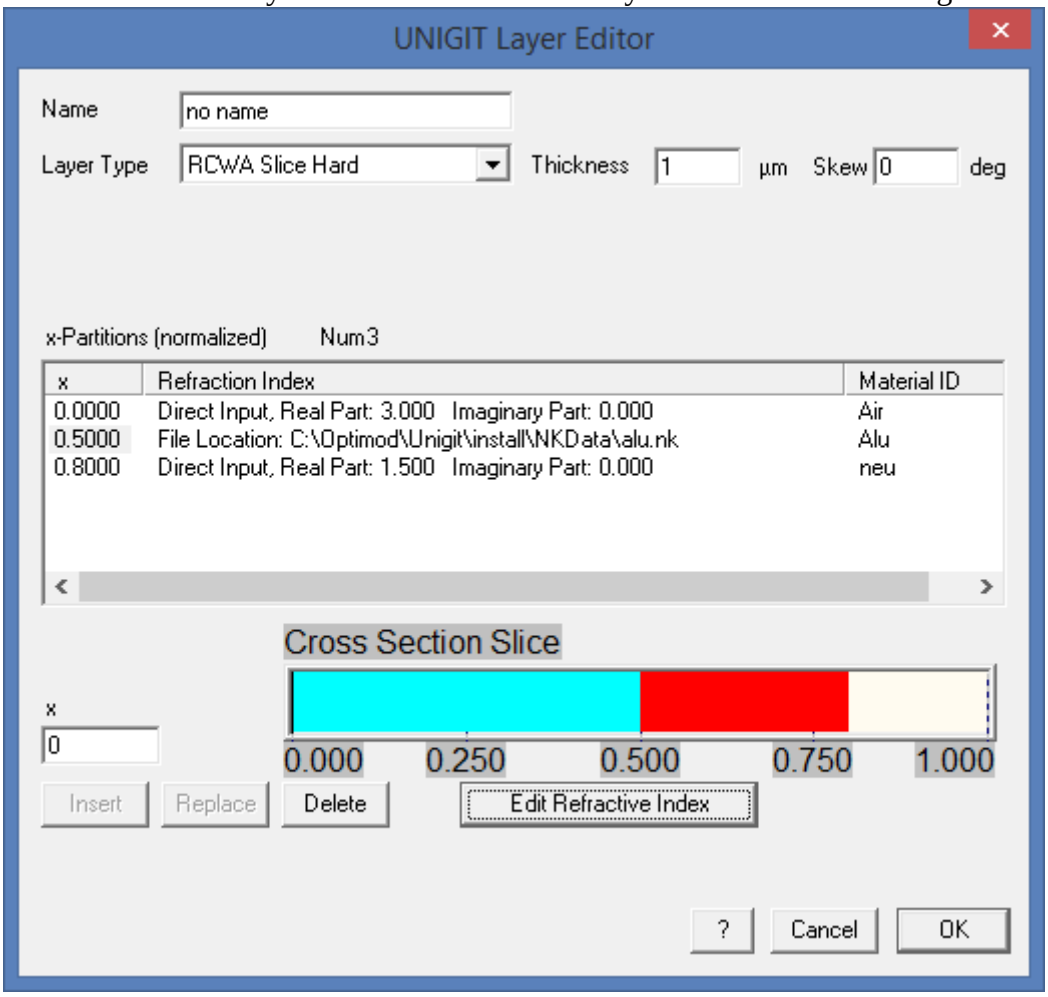


Fig. 25: Layer editor for 1D RCWA slices

As opposed to the homogeneous flat layer, a RCWA slice is characterized by different regions (or partitions) with different optical properties. The example shows three areas with different refraction index. The first area extends from 0 through 0 and is only a sort of placeholder here. It is followed by the second partition (apparently Aluminum with the n&k values stored at the shown file location) which fills the area between 0 and 0.5 pitch. The third material has the index 1.5 and fills the area between 0.5 and 0.pitch whereas the rest up to full pitch is filled by same material as the first partition (here n = 1). Note that it is possible in many cases to leave out the x = 0 entry (e.g. a slice with duty cycle = 1:1 could look like x1 = 0.25 & n1, x2 = 0.75 & n2 – here the regions from 0 through 0.25 and from 0.75 through 1.0 pitch would be filled with n1 and the remaining from 0.25 through 0.75 pitch with n2). All materials are checked for material ID's which are compared with the entrances in the color table (section 3.2). If positive, the according color is assigned and used in the cross section presentation of slice as well as in the stack cross section viewer (section 4.5).

The input procedure for the slice is as follows:

- First, enter the thickness of the slice (in microns).
- Second, enter the skew angle[2] in degrees (default is 0° which relates to standard RCWA).
- Third, specify the partitions (in the "x-partitions" group field). This is done by entering the lateral value of the right hand border of the partitions in the x input field and press either REPLACE (in order to replace the marked partition) or INSERT (in order to insert the value - it is sorted automatically into the existing list). Notice that the x-value input is relative, i.e., between 0 and 1. The renormalization to the grating period is done automatically. Furthermore, if the biggest x value xmax is not equal to 1, the region from xmax through 1 is filled with the same material like the left hand partition. In other words, the partition is completed across the period.

· Third, define the optical properties of the partitions by activating it, i.e., clicking at the x-value and the activated partition is highlighted in grey colour. Then press the button EDIT REFRACTIVE INDEX to call the refraction index editor.

Unigit features a new enhanced RCWA solver that is able to directly solve slanted gratings thus avoiding the tantalizing and time consuming procedure of slicing. What is more, the skewness can be defined layer wise. An example of a slanted grating is shown below in Fig. 26.

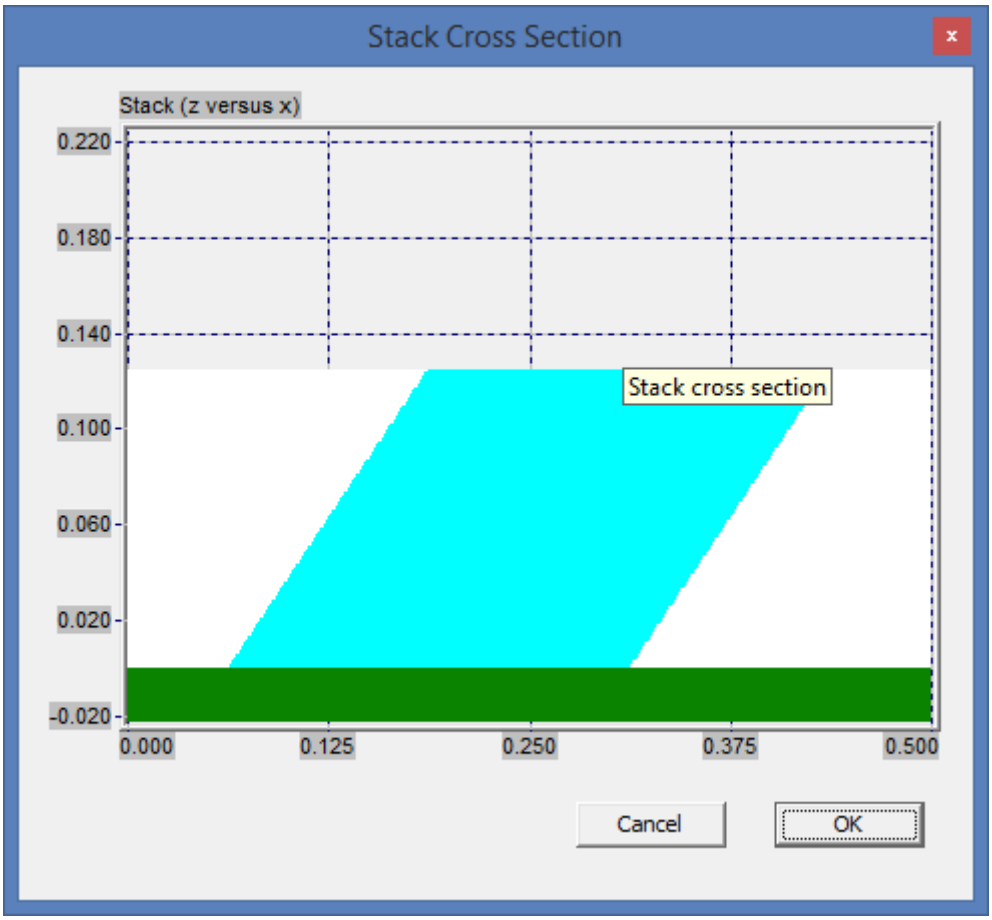


Fig. 26: Slanted grating

[2] the skew angle is measured between the vertical axis and the new non-orthogonal axis which follows the slanted profile.

5.1.5. RCWA Slice (soft transition)

The soft transition slice a.k.a. gradient layer resembles widely the hard transition case. However, there is a distinct difference in the layer description. While for the hard transition arbitrary values for the position and number of borders can be selected, the partitions in the soft case have all the same widths and the number of partitions is restricted to a power of 2, e.g., 16, 32, 64 or 128. An example of the layer editor is shown in Fig. 27.

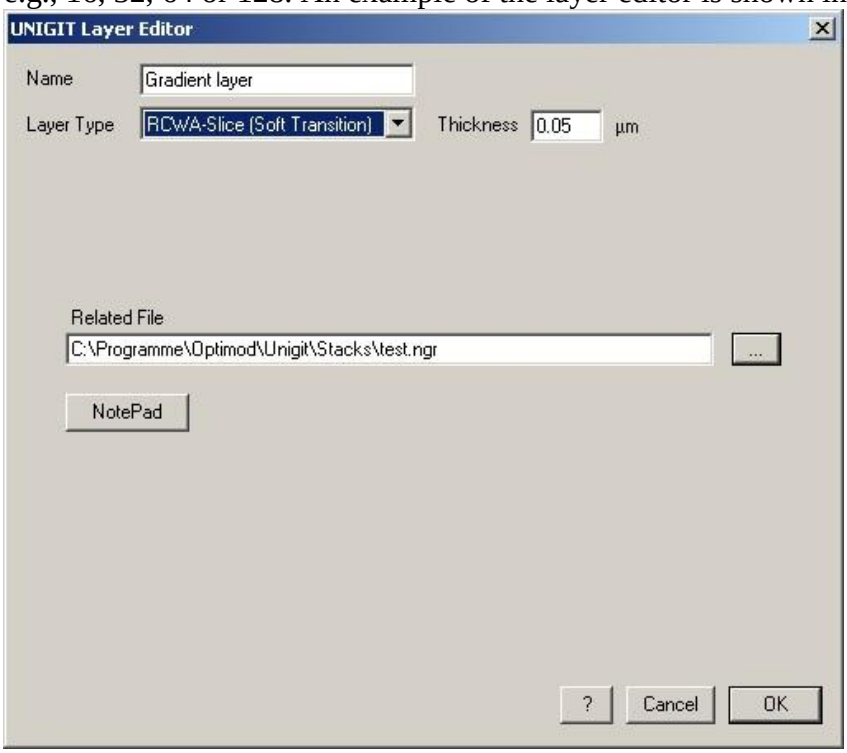


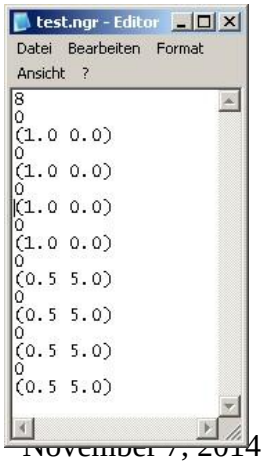
Fig. 27: Layer editor for 1D gradient layer (RCWA slice with soft transition)

Accordingly, the input procedure is different.

- First, the thickness of the slice has to be entered.
- Second, the skewness angle has to be specified (see section 5.1.4).
- Third, the layer has to be specified. The description of the layer is stored in a file that must be selected (see cursor file). In addition, the specified file can be edited by calling the notepad editor.

The composition of a file describing a RCWA soft slice is as follows:

- First, the number of partitions np has to be specified.
- Second, a refraction index specification has to be specified (i.e., 0 for direct n&k input, 1 for reading from file, ≥2 dispersion formula, see also section 6 refraction index editor)
- Third, the information according to the n&k specifier has to be provided, e.g., for 0 a n&k value, for 1 a file name etc.)
- Rows 2 and 3 are repeated np times.



An example for the data describing the gradient is shown below.

It shows 8 partitions, the first 4 with n = 1 and the second half with n = 0.5 & k = 5. In general, it is recommended to pick a power of 2 for the number of partitions. The specification of the material follows closely the Unigit standard (see index refraction index editor).

Basically, the RCWA hard and soft slice are very similar. The only differences are that whereas the size of the partitions of the hard slice can be chosen arbitrarily, the partitions of the soft slice are equidistant and their size is determined by the period divided by their total number. Behind the scene, the Fourier transform for the hard slice is done analytically whereas a FFT is applied for the soft slice.

5.1.6. Composite Polygon Layer (slicing)

In order to alleviate the assembling of complicated interfaces, the composite (or "slicing") layers were implemented. Basically, there are two types available that differ by the interface description. The polygon slicing layer is characterized by the specification of the interface by means of a set of points that are connected by straight lines. Thus, the input scheme is widely similar to those of the RF polygon layer. After selecting this layer type, the layer editor shows up as depicted in Fig. 28.

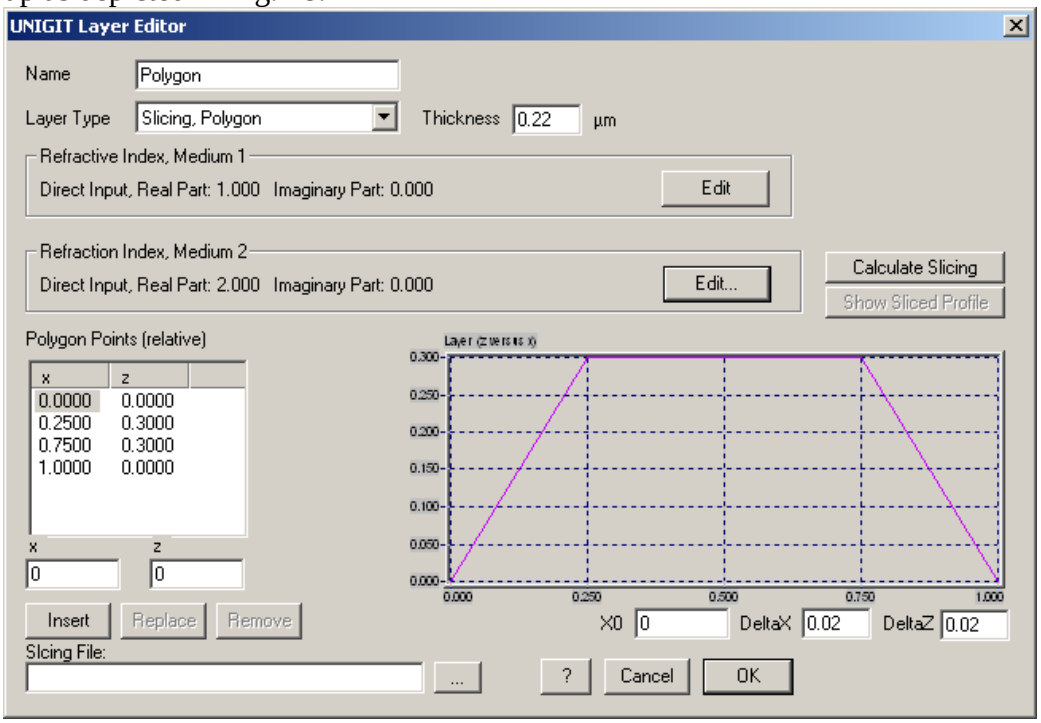


Fig. 28: Layer editor for 1D composite polygon layer (a.k.a. slicing polygon)

First, the same specifications like for the RF polygon layer (step 1 thru 3) have to be done. The big difference in comparison to RF layers, however, consists in the further processing of the layer, i.e., the solution of the diffraction problem. While for the RF a solution is obtained in one single step by the application of the Rayleigh-Fourier method, a composite layer is first sliced into elementary RCWA slices that are solved separately.

In addition to polygonal profile specification (a trapezoid profile can be seen in the image example), there are two input parameter that control the slicing process, namely DeltaX and

DeltaZ. While the former defines the relative maximum step from one slice to the next one in lateral direction, the latter regulates the slicing process near maxima with very smooth slopes. Clearly, these slicing parameters may have a strong impact on the diffraction computation results. Keeping the lateral step width constant results in slice thicknesses of the individual RCWA slices indirectly proportional to the local slope of the profile to be sliced.

Generally, the necessary quantization is determined by various factors such as the aspect ratio, the wavelength-to-corrugation ratio, the profile shape, the refractive indices etc. As a rule of the thumb, a slicing criteria postulates that the absolute step width (i.e., DeltaX multiplied by the grating period) should be less than about a tenth to a twentieth of a wavelength. On the one hand, a finer slicing means increased accuracy of the computations. On the other hand, the computation time is increased likewise. An optimum value can be ascertained by means of numerical investigations.

When you are finished entering the specification you need to click the button ”Calculate Slicing” or the button ”OK” to run the slicer module. The result (sliced profile) is stored to the location contained in the editing field ”File Location” (make sure you have filled this field in order to avoid an error message). By choosing the first option, the window stays open and the button ”Show Sliced Profile” becomes active. Then, the sliced profile can be visualised just by clicking this button with a result similar to what is shown in Fig. 29.

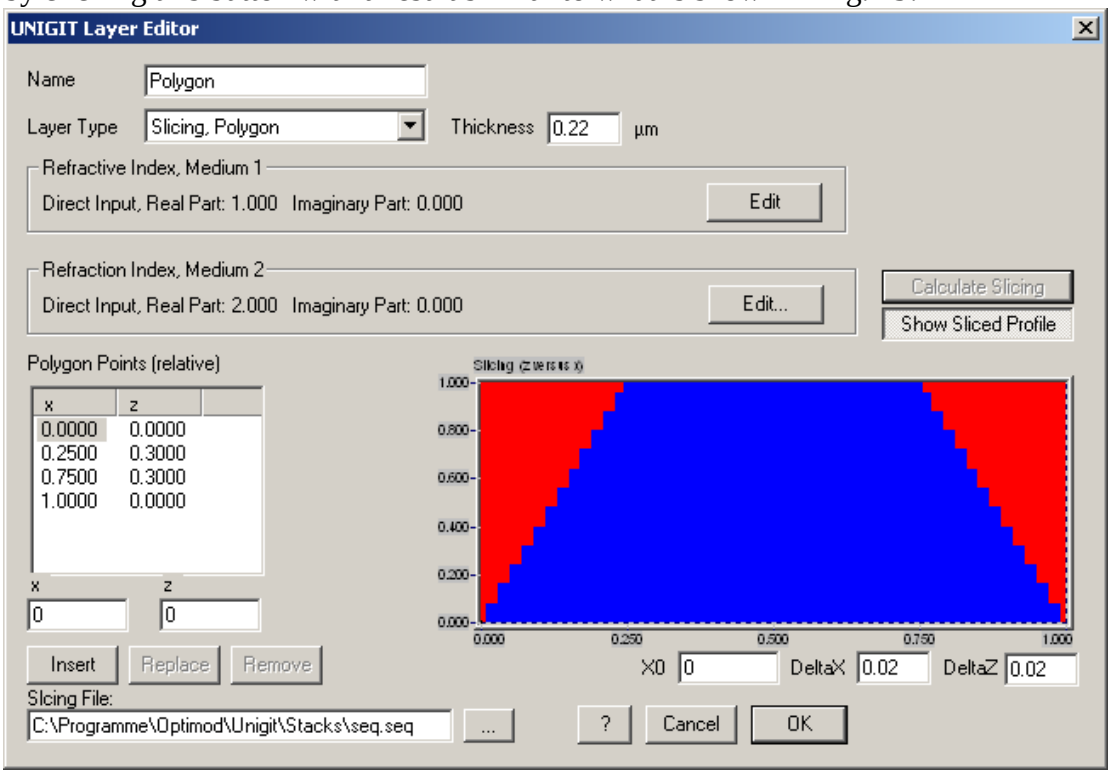


Fig. 29: Presentation of the sliced profile in 1D composite polygon layer

5.1.7. Composite Fourier Layer (slicing)

In principal, the composite polygon layer is sufficient to specify all kinds of interfaces by applying an appropriate dense series of points. With concern to curved surfaces this may be sometimes rather cumbersome. To this end, a definition of the profile in terms of parametric

Fourier series was implemented. Here, both the ordinate and the abscissa are given as a Fourier series of a parameter t as follows:

$$x(t) = t + \sum_k XA_k \cdot \sin(2\pi k t + XP_k)$$
$$z(t) = \sum_k ZA_k \cdot \sin(2\pi k t + ZP_k)$$

Setting all coefficients XA_k to zero leads to the known Cartesian Fourier representation. Instead of the direct input of the interface by means of point series, the coefficients ZA_k and XA_k as well as the phases ZP_k and XP_k have to be entered. The related layer editor is shown in Fig. 30.

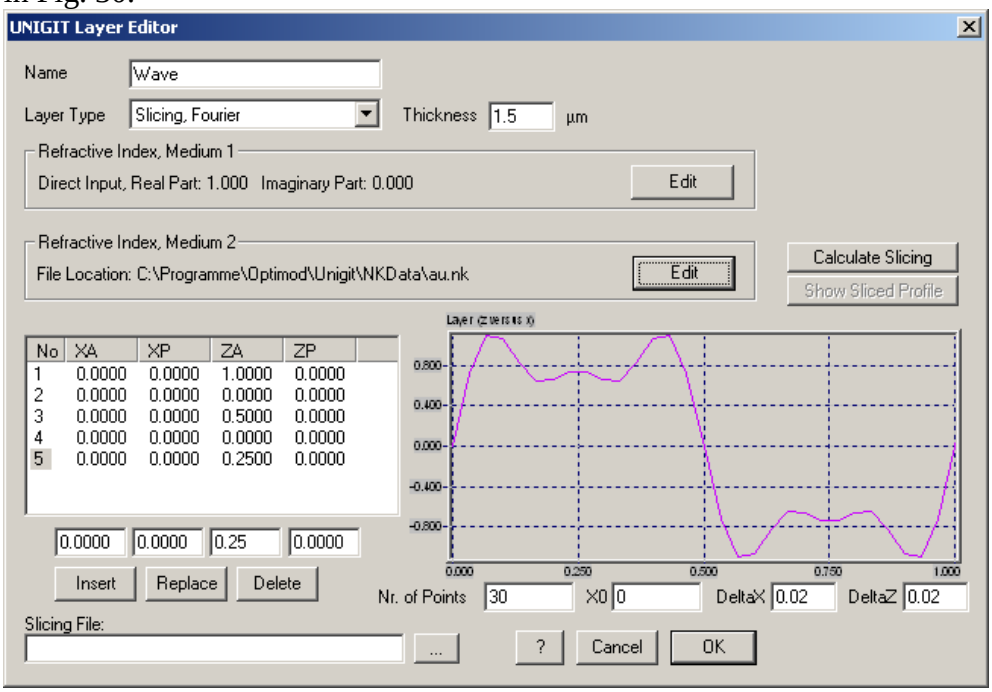


Fig. 30: Layer editor for 1D composite Fourier layer (a.k.a. slicing Fourier)

In the above example, a sinus along with the first two odd harmonics (k = 3 and k = 5) was input. The waves have the amplitudes $ZA_1 = 1$, $ZA_3 = 0.5$ and $ZA_5 = 0.25$. The waves are not offset, i.e., $ZP_k = 0$. Moreover, there is no distortion, i.e., $XA_k = 0$. Basically, the input table has to be filled from k = 0 thru the highest relevant order, that means the highest order for which XA is different from zero. This has to be obeyed even in the case that some orders (here 2 and 4) disappear (in this case ZA = 0 has to be set).

In order to delete an entry, click on the last number of the first column which activates the DELETE button. Then, just click the DELETE button. Note, only the last entry can be deleted. Similar applies for the replacement of entry lines. However, here arbitrary lines can be replaced. The information from an entry line is copied into the edit boxes by double clicking the number of this line.

Distortion can be implemented by means of adjusting the XA and XP properly. An example of a distorted sine wave is shown below. Depending on the XA values, profiles with strong undercuts can be modeled (see Fig. 31).

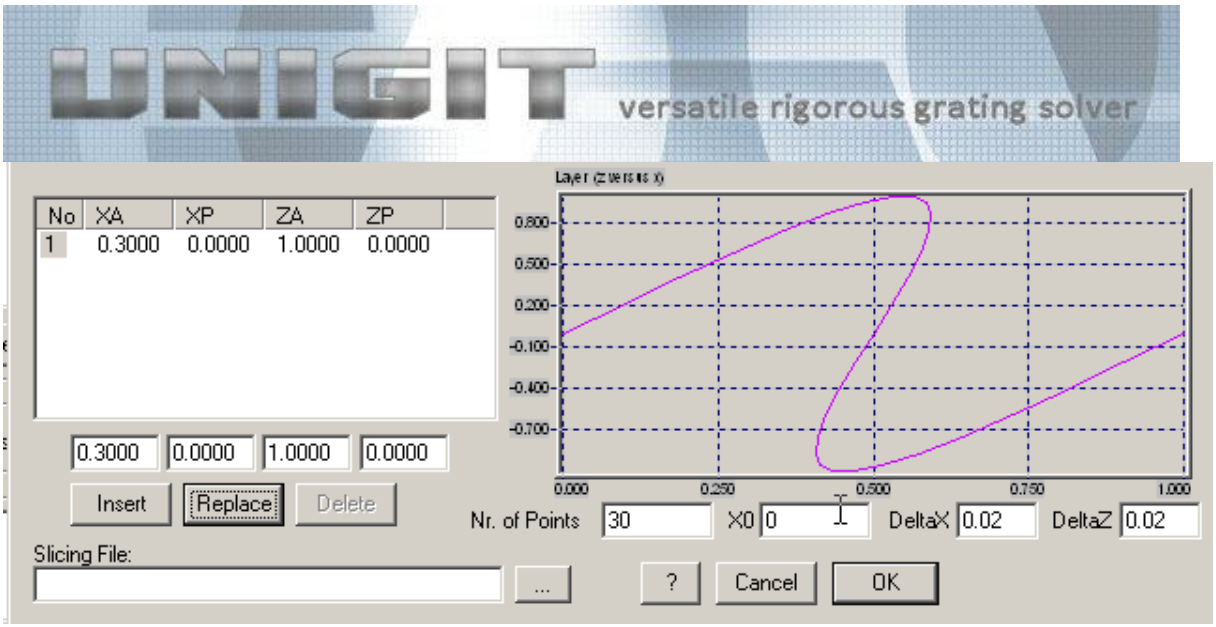


Fig. 31: Input of a distorted re-entrant profile by means of the 1D slicing Fourier layer

The associated sliced profile is shown in Fig. 32.

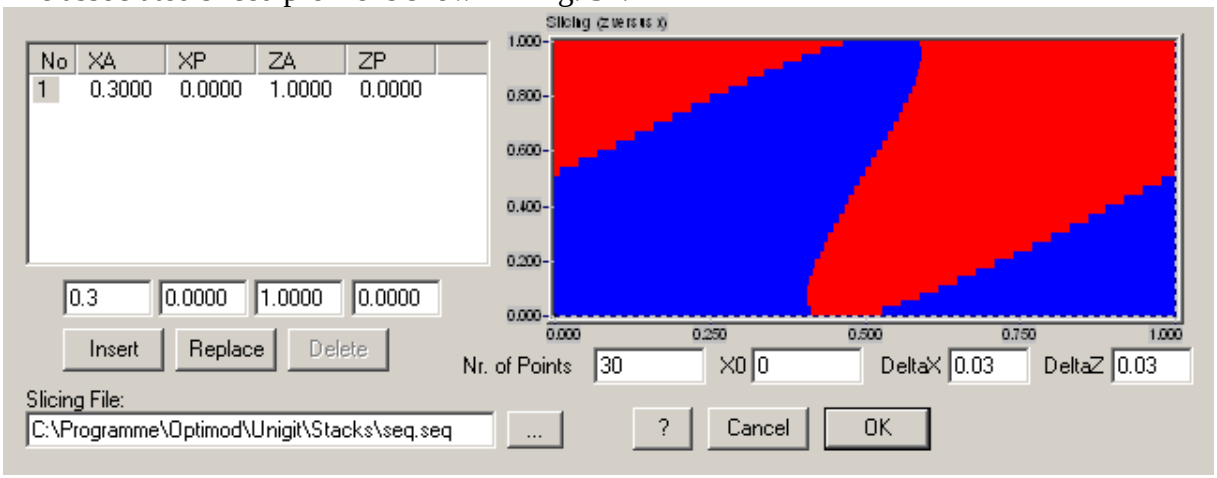


Fig. 32: Sliced profile of Fig. 31

Eventually, discrete interface points (that are equidistant along the profile) are generated from the Fourier series. The total number of these points has to be specified in the input field "Nr. of points".

5.1.8. Sequence

The layer type sequence like a thin film layer is common for both 1D as well as 2D. Therefore, the editor appearance will also be the same. It is depicted below. Basically, a sequence can only be used if defined before. This can be done in the stack editor or it is automatically done for the 1D slicing layers. If a sequence layer is to be built into a stack, the location of the stored file has to be specified in the corresponding edit field. In addition, the total thickness of the sequence has to be entered. The structure of the sequence (or sub-stack) is renormalized to this thickness. Likewise, the lateral dimensions are renormalized to the new pitch (grating period). The button EDIT permits to edit the sequence in the stack editor whereas the button Notepad opens a notepad with the sequence in ASCII-format. The layer editor window for a sequence is shown in Fig. 33.

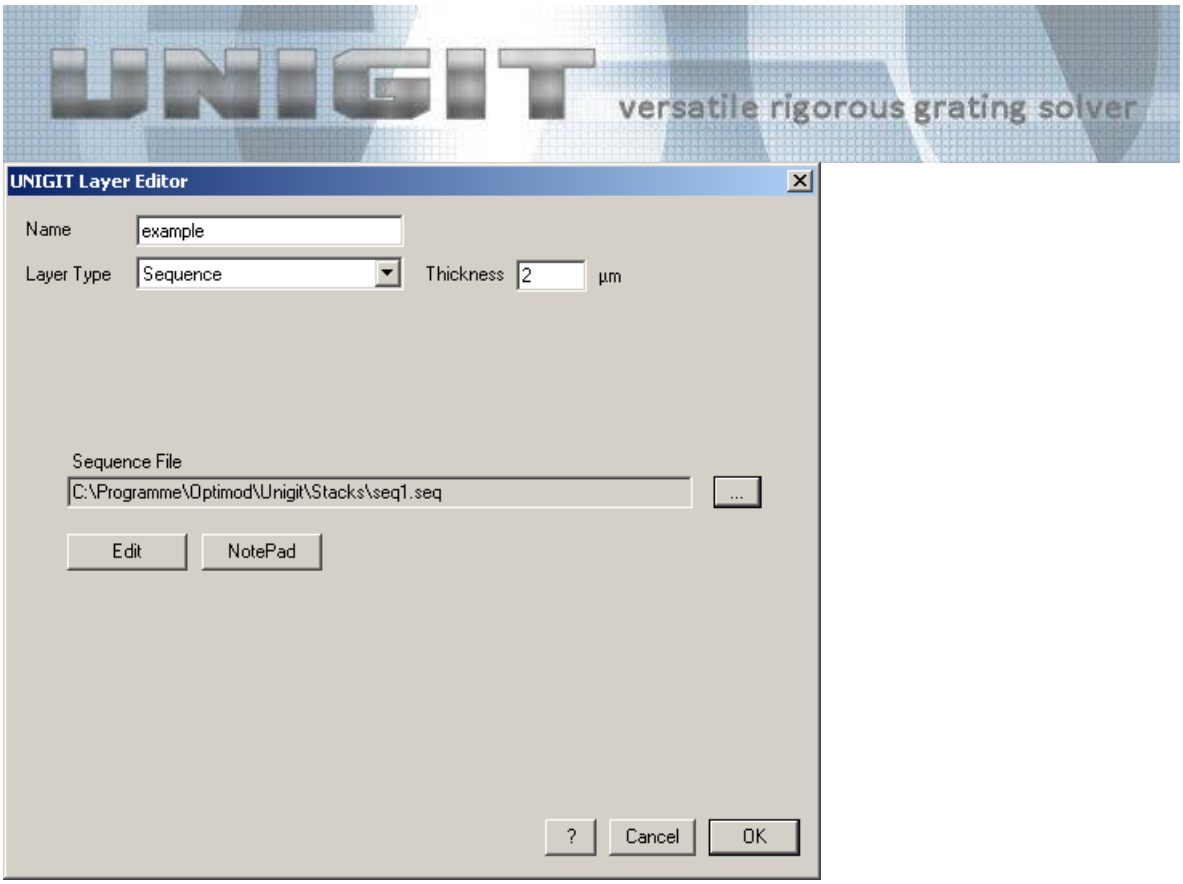


Fig. 33: Layer editor for 1D sequence layers

5.2. Layer Editor 1C (C-method)

Likewise, the layer editor to run the C-method solver is activated from the stack editor by clicking either the button ADD_LAYER or highlighting one layer and clicking EDIT (hotkey CTR-E). As opposed to the RCWA, where the stack is assembled by layers such as slices or thin films, the stack in the C-method is rather described by interfaces, its corrugations and distances to each other. Therefore, the term interface is mainly used in the following (while "layer" if used is understood as a synonym for interface rather than a "real" layer).

5.2.1. CM-Polygon

The layer editor for the CM polygonal layer (or better interface) is shown in Fig. 34. It is very similar to the RF polygonal layer, i.e., there entrances for the corrugation of the interface (corresponding to the height of the profile) and for the materials at either side of the interface (see refraction index editor in section 6). In addition, the distance to the next interface above has to be specified. Due to the specifics of the C-method, there are no overhanging or undercut features allowed (see /8/, /9/). The material between the shown profile and the profile above is verified in the refractive index, medium 1 entrance.

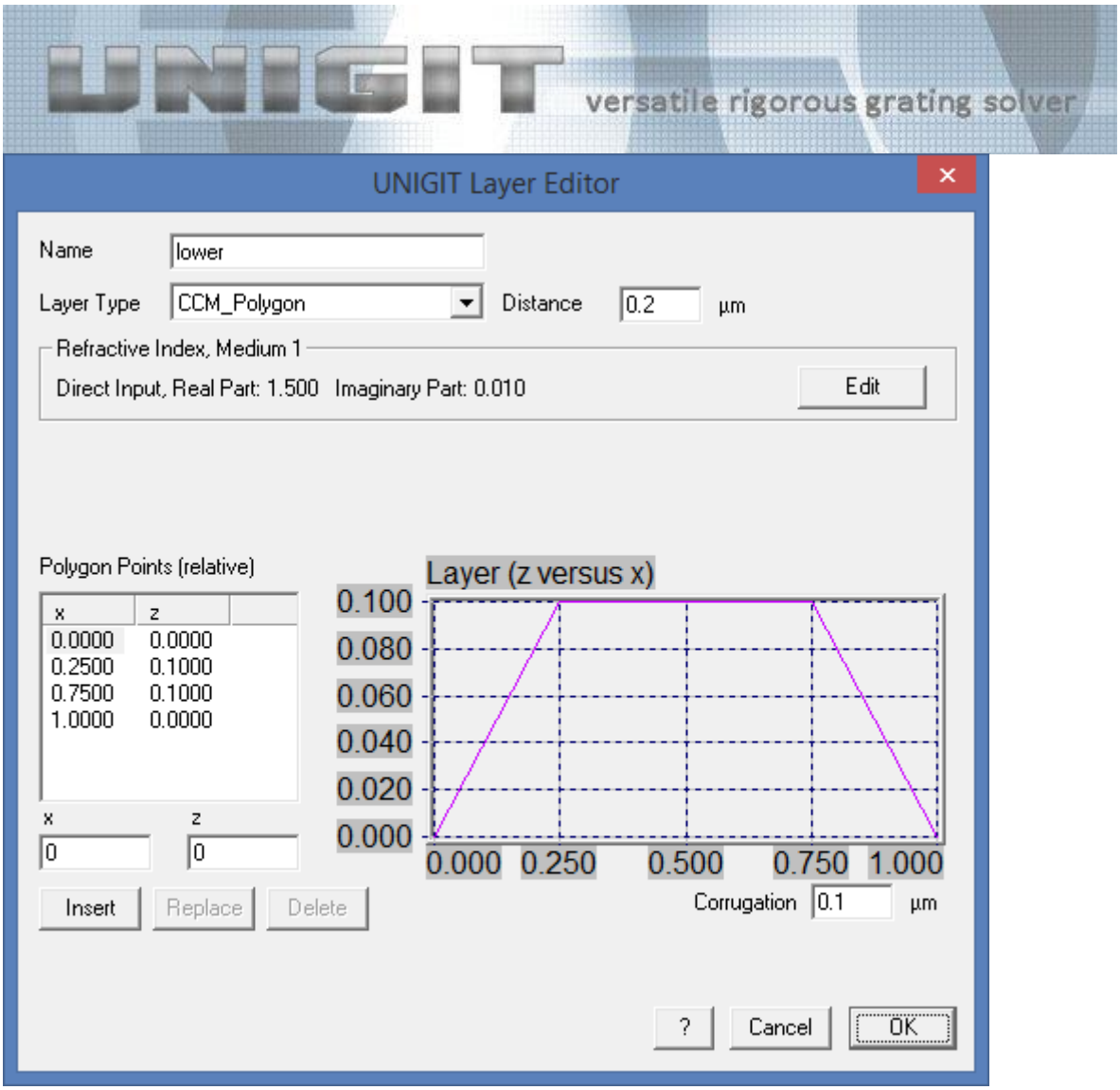


Fig. 34: Layer editor for a 1D CM polygon layer (to be run with C-method)

5.2.2. CM Sinusoidal Layer

The layer editor for the CM polygonal layer is shown in Fig. 35. It is very similar to the RF sinusoidal layer, i.e., there entrances for the amplitude (thickness of the layer), the lateral phase X0 and the relative strength of the second harmonic HR. In addition, the lateral phase x1 of the second harmonic can also be specified. In general, there would also be an entrance for the distance to the layer above. This is not visible here since the interface shown in Fig. 35 is the uppermost of the stack shown in Fig. 15. For the same reason, the index is set automatically to the superstrat index that is specified in the stack editor (see section 4).

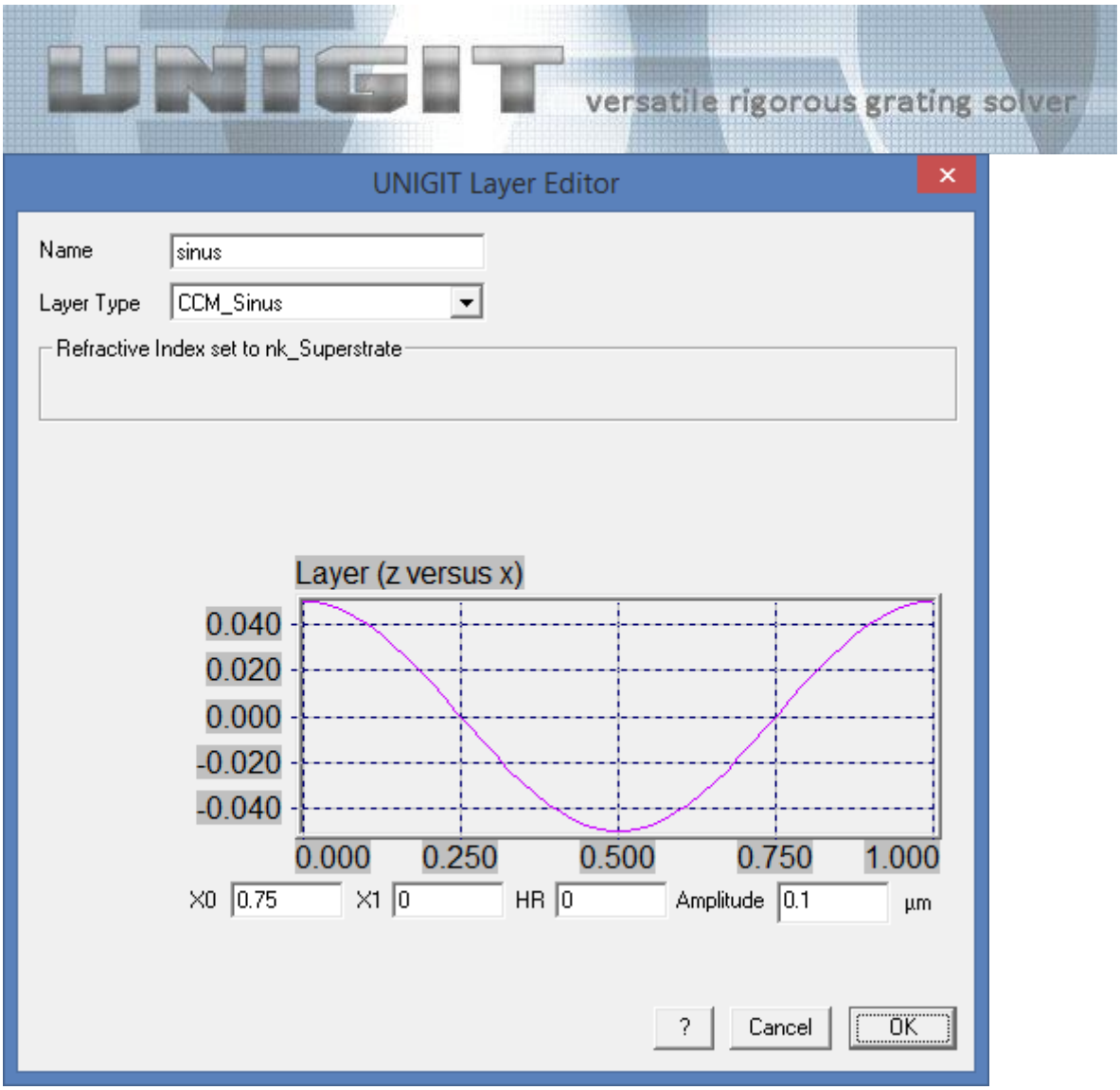


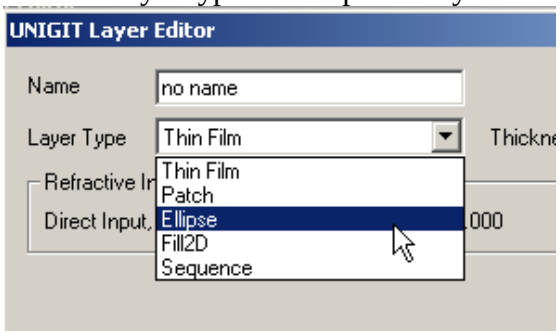
Fig. 35: Layer editor for a 1D CM sinusoidal layer (to be run with C-method)

5.3. Layer Editor 2D (RCWA)

The layer editor can be activated from within the stack editor. Its appearance and functionality depends on whether the 1D (classical or conical) or 2D option is selected.

If activated by the ADD LAYER button within the stack editor the layer editor will appear and offer the input of a thin film layer.

Another layer type can be picked by means of the drop down box as shown below:



Depending on the selected layer type, the 2D layer editor is opened. In the 2D case, there are 5 options available:

- Flat Homogeneous Layer (Thin film),
- Patch (Rectangles) Layer,
- Super-Ellipse Layer,
- Arbitrary Filling,
- Sequence.

5.3.1. Flat Homogeneous Layer

The specification for homogeneous layer (a.k.a. thin film) in 2D is the same as in 1D. You need only to enter the thickness of the layer (in microns) and to specify the material by means of the refraction index editor. Moreover, the layer a name can be attached to the layer.

5.3.2. Patch (Rectangle) Layer

This layer type will be of advantage if your 2D area can be approximated by means of rectangular patches. With the help of the editor you can specify the position of the patch as well as its material within the elementary cell. In addition, you need to enter the thickness of the layer. An example is shown in Fig. 36.

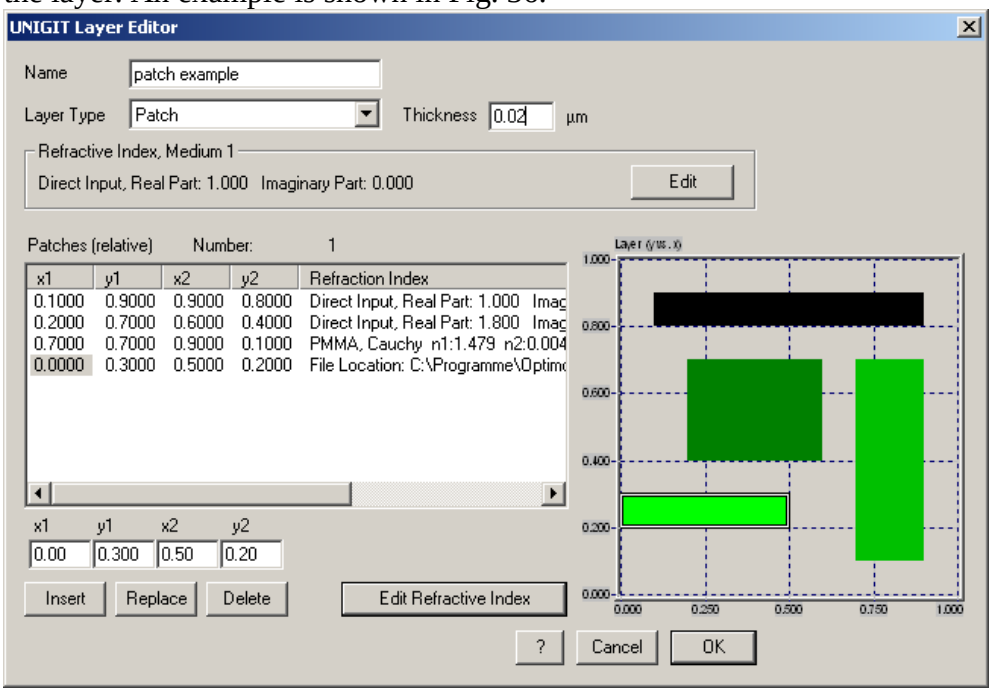


Fig. 36: Layer editor for a 2D patch filling layer

The rectangles are specified by 2 diagonal corner points (x1,y1 and x2,y2) and its material entry (by means of the refraction index editor). The embedding material is medium 1.

Note that unlike the super-ellipse layer, there is no “orthogonal” option, i.e., the patch is always defined in the given coordinate system. This means in other words that patches which have been defined in a non-orthogonal system (pitch angle different from 0) are parallelograms in the real world.

5.3.3. Ellipse Layer

Contrary to the patch layer type where a number of different areas can be specified, the super ellipse layer type can include only one feature. However, this feature is a less restricted one namely a super ellipse. The mathematical expression for a super ellipse is given below:

$$\left(\frac{x}{dx}\right)^p + \left(\frac{y}{dy}\right)^p = 1$$

The appearance of the layer editor for an ellipse is shown in Fig. 37.

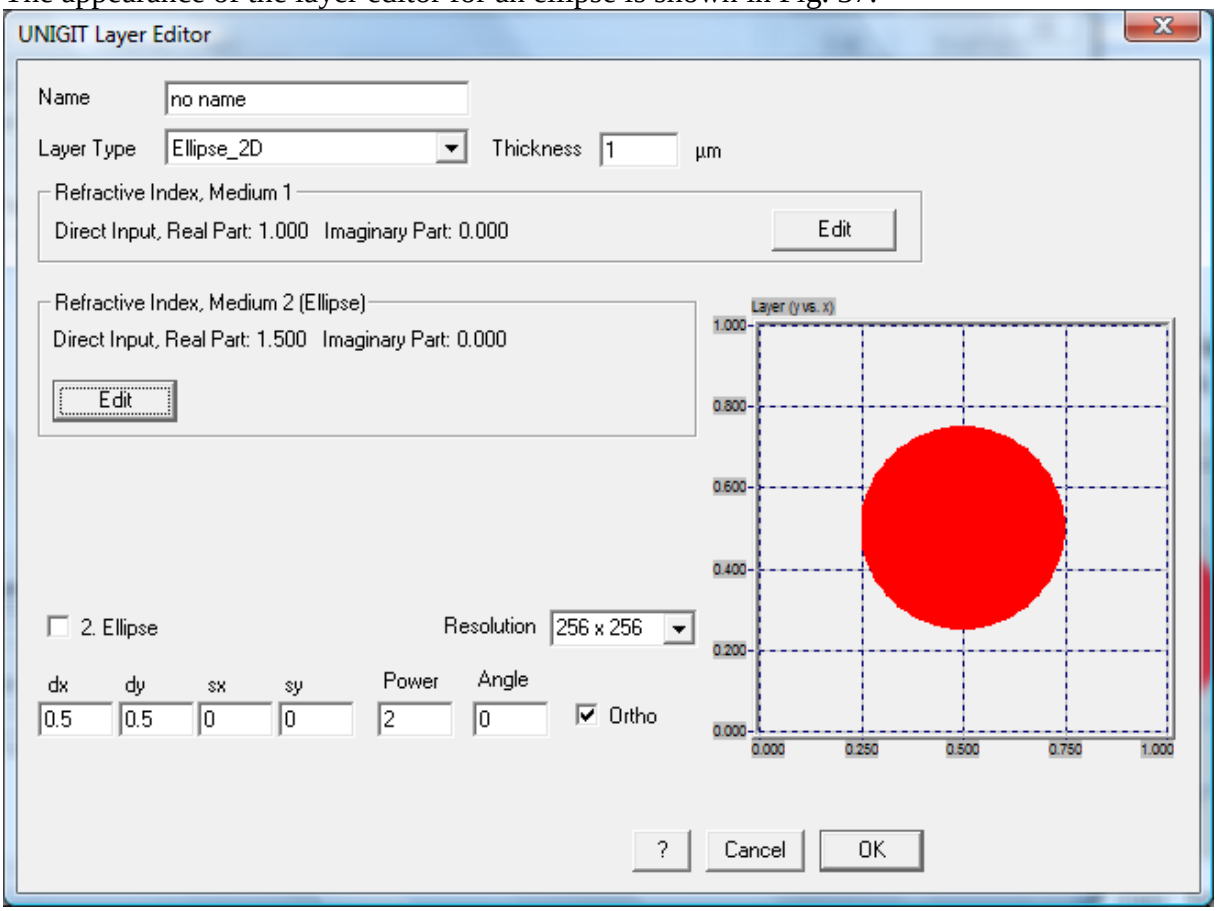
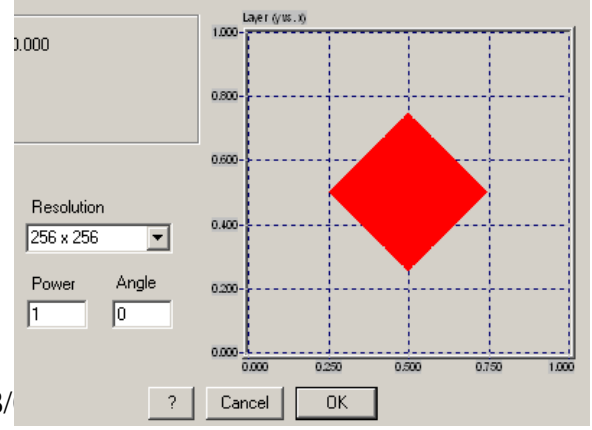
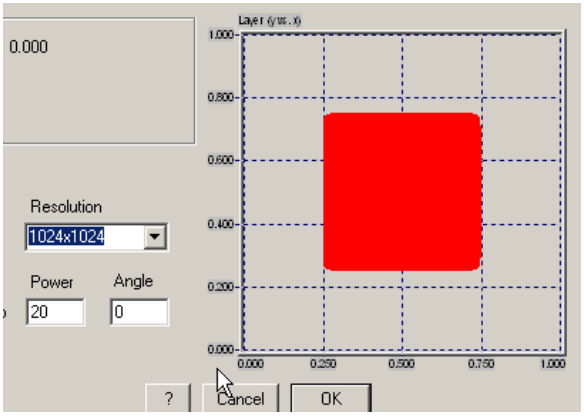


Fig. 37: Layer editor for a 2D ellipse filling layer

The embedding medium is medium 1 and the ellipse is medium 2. The dx and dy are the diameter in x and y direction whereas sx and sy are the relative lateral shifts. Angle is the rotation angle of the ellipse.

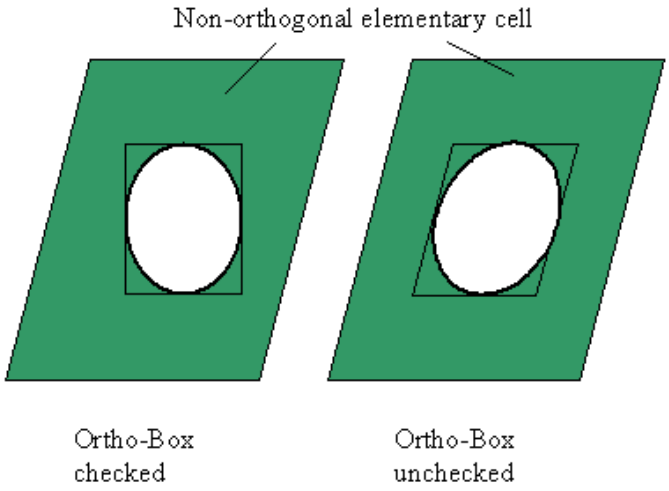
Unlike a regular ellipse with power p = 2, the power p of a super ellipse can be an arbitrary positive rational number. In this way, a range of features between an ellipse and a rectangle (p = infinity) can be covered. Moreover, a diamond can be described with p = 1 and concave shapes with p < 1. Two cases are shown below.



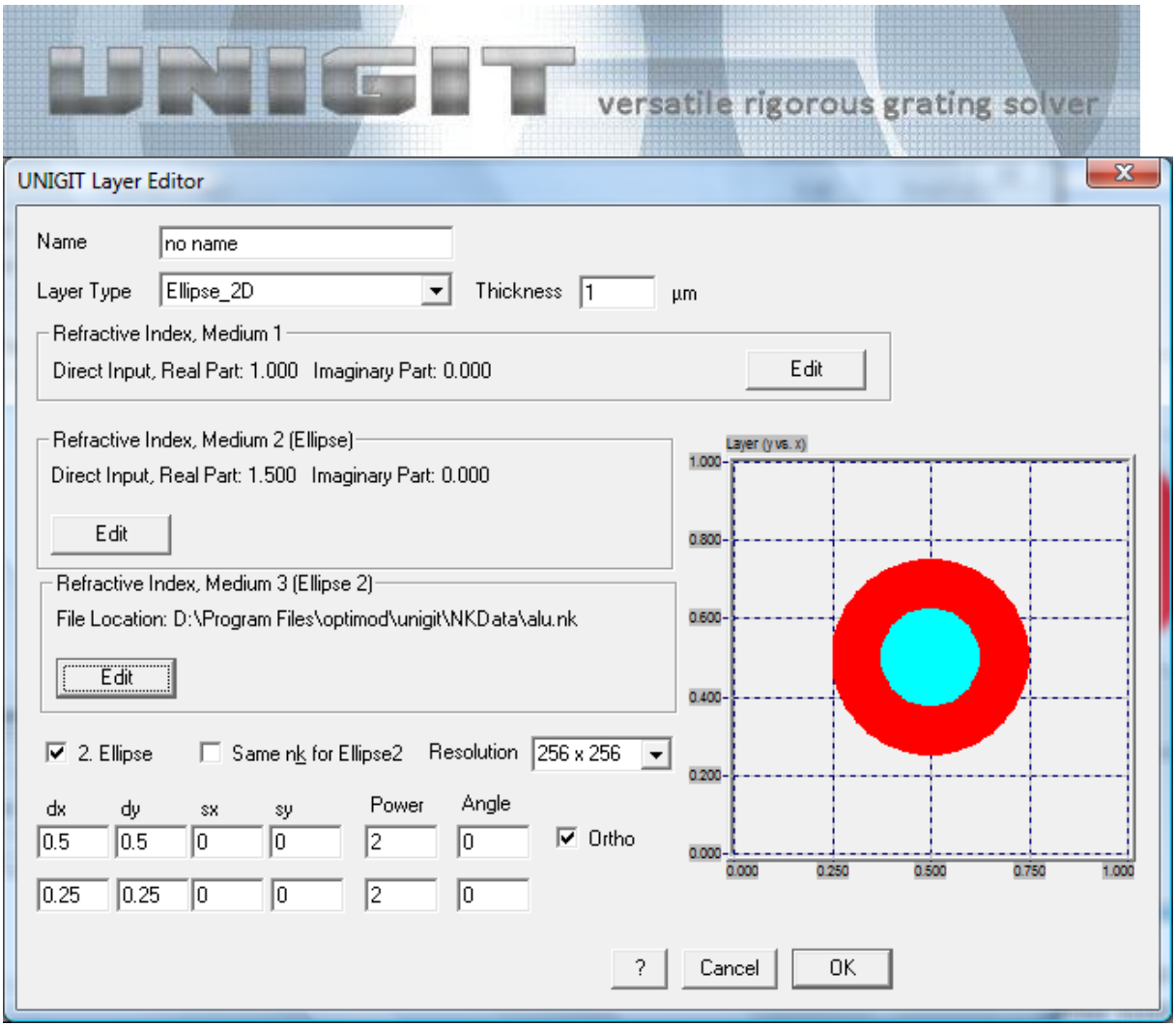


Moreover, the ellipse can be rotated by specifying the wanted rotation angle and it can be shifted laterally by entering the correspond relative values into the edit fields sx and sy. By checking or unchecking ”Ortho” you can define whether or not the ellipse is ”orthogonal” in real world in spite of a non-orthogonal cell (defined by zeta – see stack editor). For further explanation see also the drawing below.

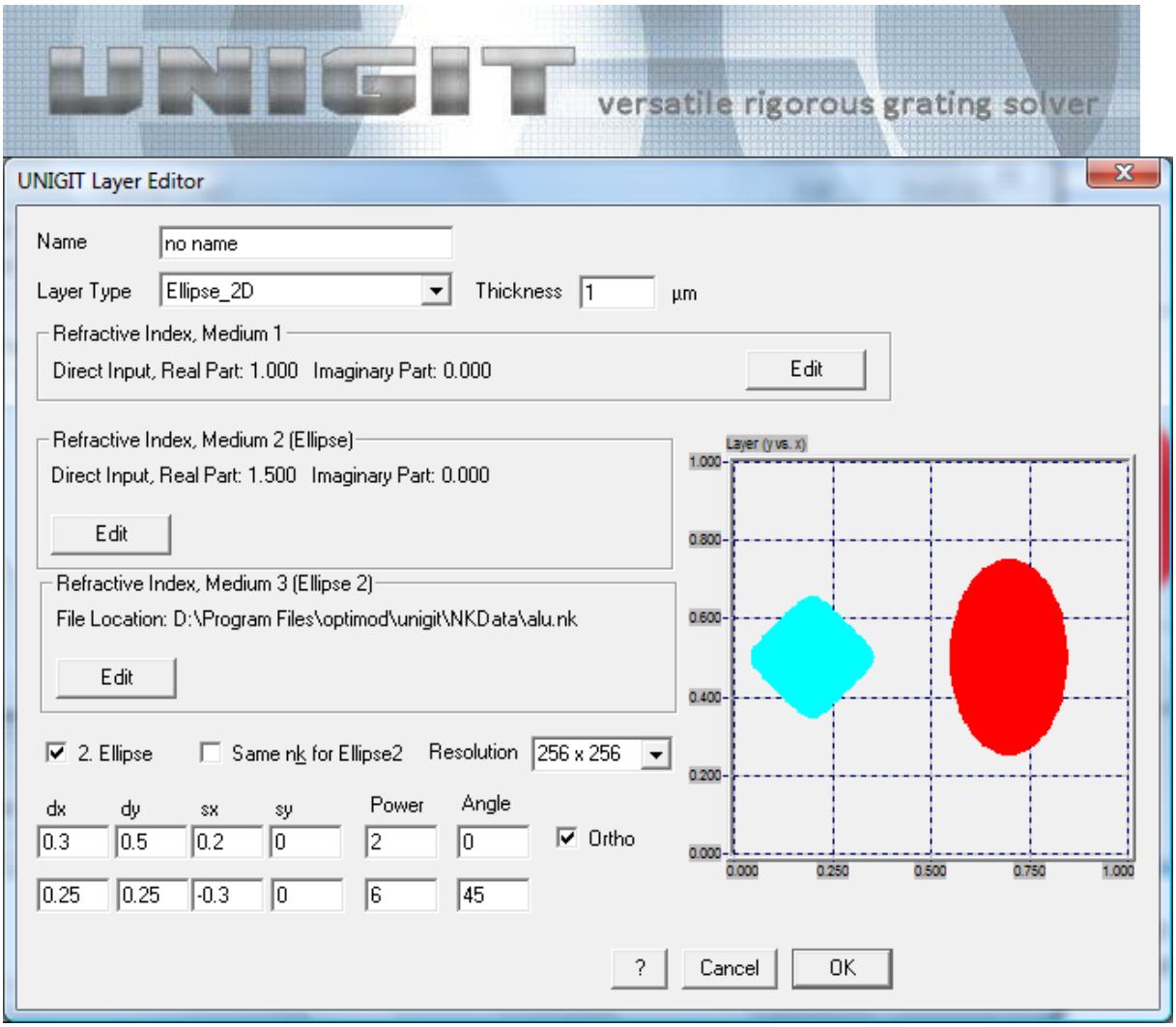
Check the box Ortho if you want to have an ellipse in real world, otherwise the ellipse is defined for the non-orthogonal coordinate system, i.e., it will be distorted in real world. Schematically, this is shown in the drawing below. (Notice, the drawing does not show this, however, your calculation result is supposed to be different for the cases of a checked and an unchecked Ortho box.)



The checkbox *2. Ellipse* enables the definition of “an ellipse in an ellipse” as can be seen in the figure below.



However, the second ellipse has not necessarily to be located inside the first ellipse as the setup below shows. Principally, the second ellipse always overrules the first one.



5.3.4. Arbitrary Filling

Unigit version 1.02.02 offers a new feature for arbitrary pixel filling of a 2D layer. It is comparable to the gradient RCWA (or soft slice) in 1D.

If you want to include a arbitrary fill layer then pick the Fill2D option in the drop down box of the 2D layer editor. Then, the layer editor opens with the window shown in Fig. 38.

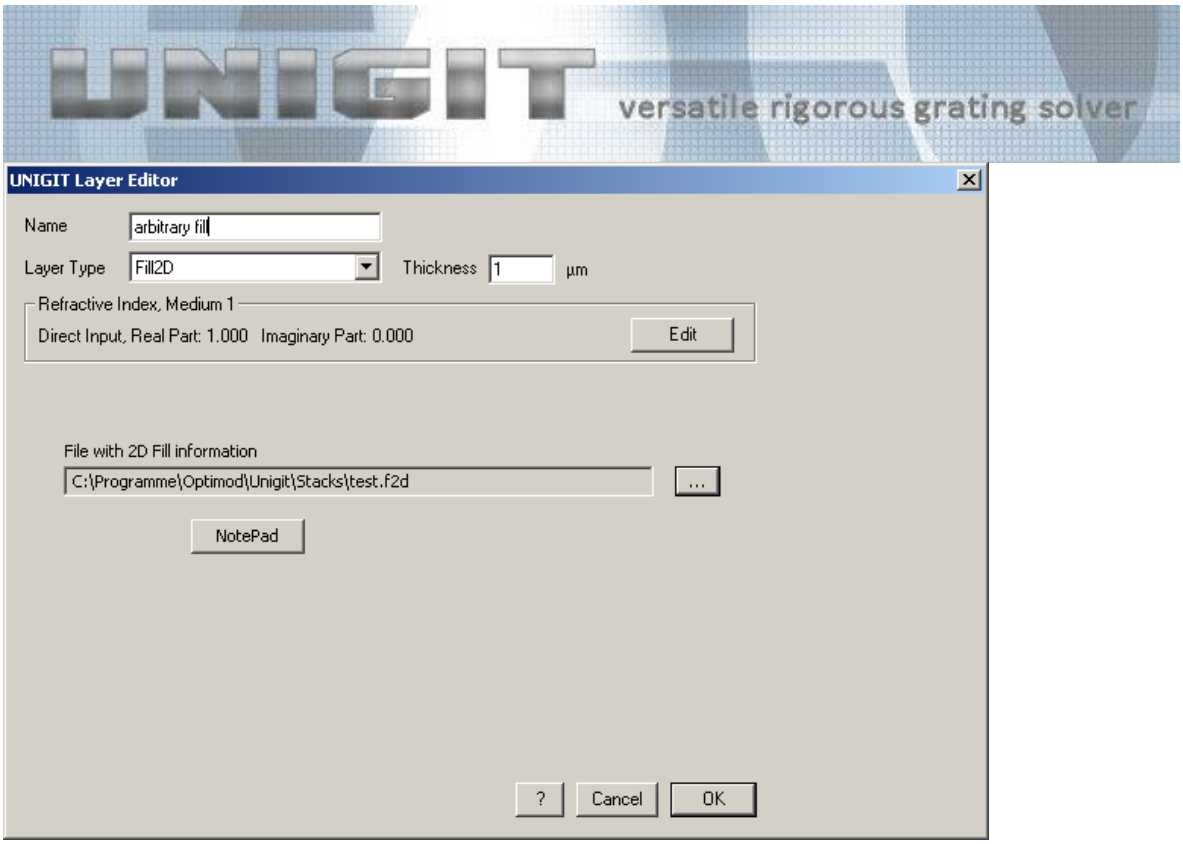


Fig. 38: Layer editor for a 2D arbitrary filling layer

Here, you need to specify the layer thickness and you can enter a name for the layer similar to other layer types. Moreover, you need to specify a material for the embedding medium and a file (with the extension .f2d) which contains the bitmap description of the layer.

The file content is explained as follows:

2 /number of materials beside the embedding material
0 /specification ID for material 1 (0 = direct input)
1.5 0.0 /n&k for material 1
1 /specification ID for material 2 (1 = n&k file)
'C:\Program Files\OPTIMOD\Unigit\NKData\SILICON.NK' /pathname of the n&k file for material 2
6 97 /Fourier discretization power (e.g., discretization is 2^6) & number of lines to follow
528 0 /pixel number & material ID, e.g., 528 pixels with material 0 (embedding material)
32 1 /pixel number & material ID, e.g., 32 pixels with material 1
32 2 /pixel number & material ID, e.g., 32 pixels with material 2
……..
32 1 /pixel number & material ID, e.g., 32 pixels with material 1
528 0 /pixel number & material ID, e.g., 528 pixels with material 0 (embedding material)

Note, the total number of pixel description lines in this example must be 97 and the total number of added pixel numbers (i.e., 528 + 32 + 32 + … + 32 + 528) must be equal to (2^6)^2 = 4096

A graphical example for a non-orthogonal cell is shown below. First, the elementary cell has to be defined as shown (red color in Fig. 39).

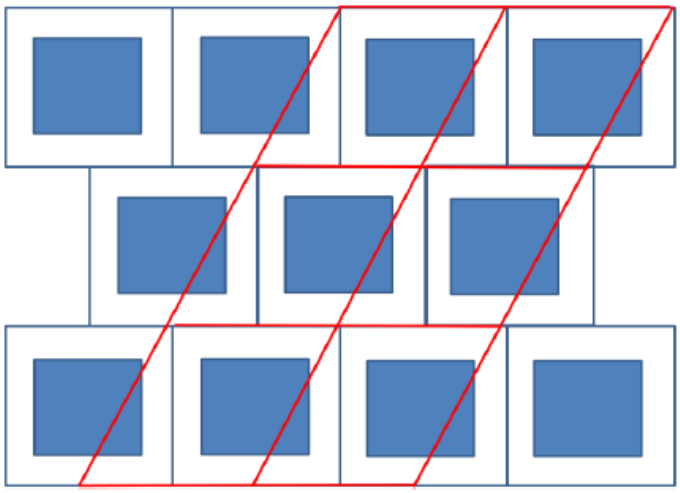


Fig. 39: Determination of the elementary cell

As a next step, one has to do the discretization in unit cell coordinates. The linear pixel number has to be a power of 2 (64, 128, 256 etc.). The pixel number in both axes has to be the same (no matter what the pitch is).

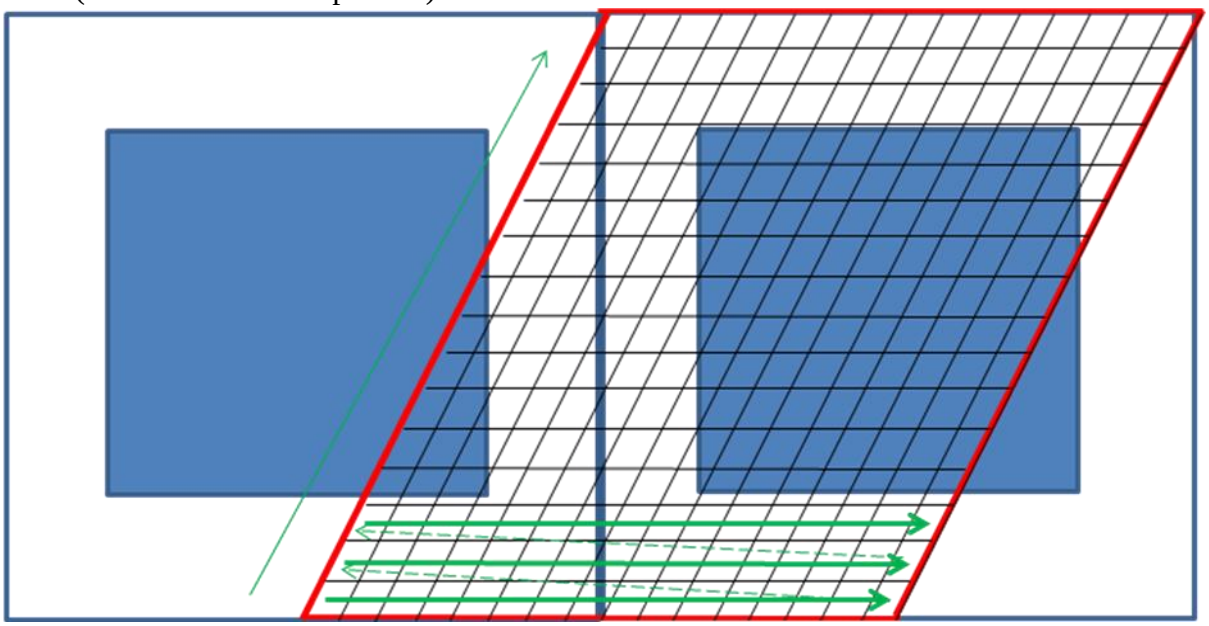


Fig. 40: Filling of an arbitrary layer

The counting direction is indicated by the green arrows in Fig. 40, i.e., scanning row by row with blanking like behaviour (jump from the end of one row to the begin of the next row) as shown. As long as pixels have the same material (defined by largest area percentage) just add, when material is changing → save the value and restart counting (across scan blanking).

This would result in the following .f2d information for the graphical example assuming material 0 is white and material 1 is blue:

48 0 (3 lines with 16 pixel each)
3 1 (3 pixels blue)
6 0 (6 pixels white)
10 1 (7 pixels in row #4 right hand and 3 pixels in row #5 left hand, counting across the blanking)
5 0 (5 pixels white)
etc.

The installation of version 2.XX.XX comes with 3 stack files that describe all the same geometry (a rectangular area with n = 1.5 embedded in vacuum with n = 1) but with different means:

- stack2d_elli.ust using the ellipse descriptor (ellipse filling),
- stack2d_arbi.ust using the new bitmap descriptor (arbitrary filling),
- stack2d_rect.ust using the patch descriptor (patch or rectangle filling).

Running these files should give very similar results (for details see Unigit tutorial).

5.3.5. Composite 2D Layers

This layer type permits to implement real 3D structures by means of a very compact description. It is quite similar to the 1D composite layer types in the sense that an automatic slicing works behind the scenes to transform the input description in a number of 2D slices of the 2D ellipse layer type. It is activated by selecting the CONE-3D option in the 2D layer editor. In doing so, the layer editor window as depicted in Fig. 41 shows up:

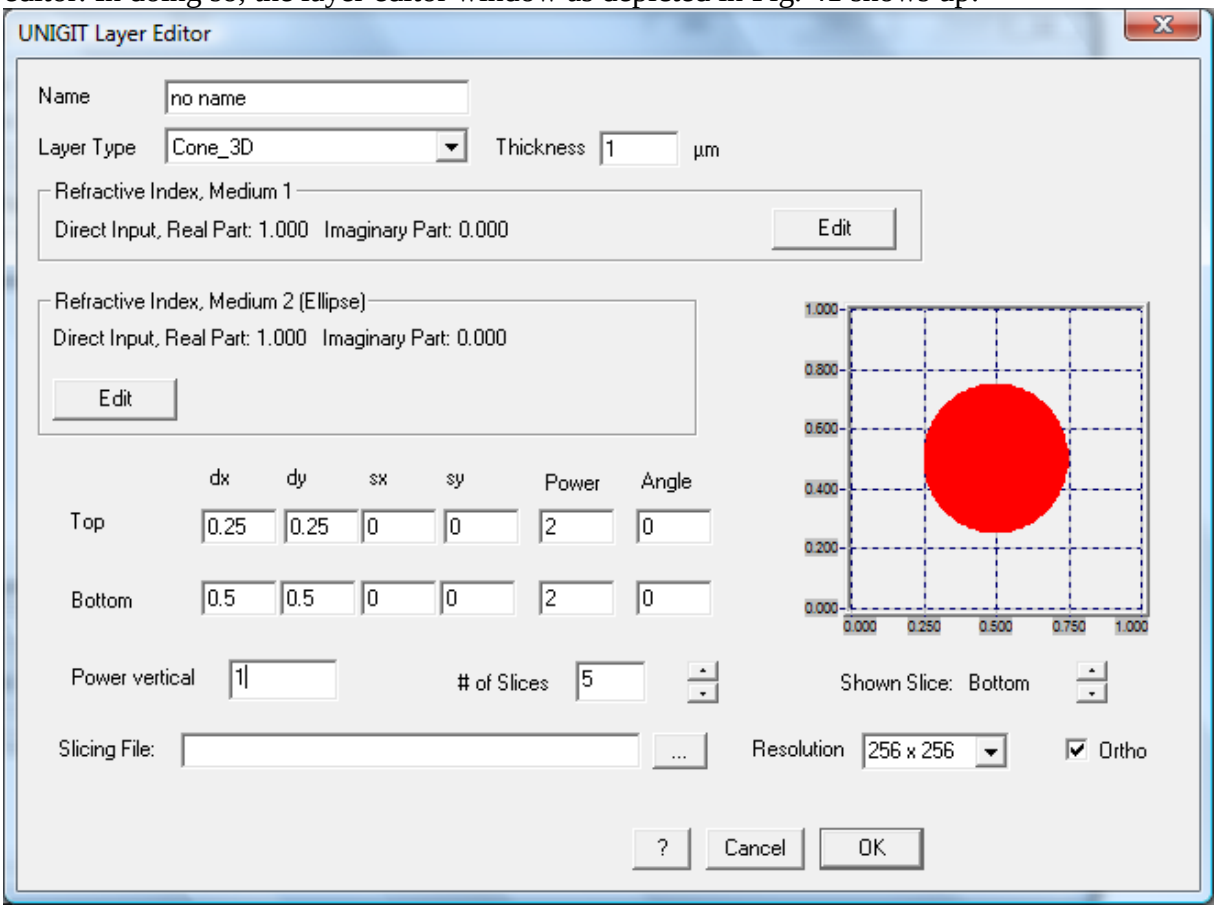


Fig. 41: Unigit Layer Editor for 2D-composite layers

Here, the user can define the top and the bottom cross section shape of his grating pattern just like the same way as he is used to with the ellipse filling feature. But as opposed to it, now he defines a real 3D body. The vertical shape of this body is defined by the vertical power. Here, the 1 results in a straight line while a value >1 gives a convex and a value <1 results in a concave shape, respectively. Furthermore, the user can define the number of slices. Actually, a equidistant slicing routine takes care to automatically translate the compact 3D description in something digestible for the RCWA solver. Finally, the user can sweep up and down through the 3D body by means of the "Show Slice" option. Here, the cross section shape of the activated slice number is plotted. Fig. 42 shows such a sequence of cross sections for the body defined above.

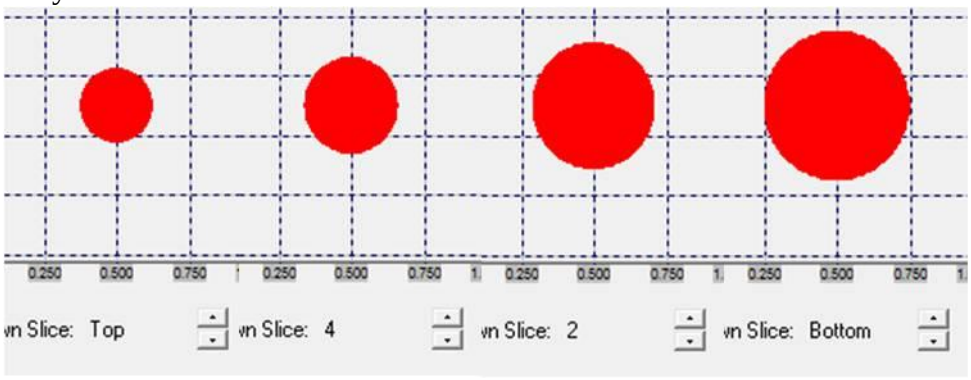


Fig. 42: Sequence of cross sections when vertically sweeping through a 2D composite layer

5.3.6. Sequence 2D

The layer type sequence like a thin film layer is common for 1D and 2D. The difference is that a 1D sequence can only contain 1D layer types whereas a 2D sequence has to contain 2D layer types only. However, the appearance of the layer editor will be the same. Basically, a sequence can only be used if defined before. This can be done in the stack editor or it is automatically done for the 1D slicing layers. If a sequence layer is to be built into a stack, the location of the stored file has to be specified in the corresponding edit field. In addition, the total thickness of the sequence has to be entered. The structure of the sequence (or sub-stack) is renormalized to this thickness. Likewise, the lateral dimensions are renormalized to the new pitch (grating period). The button EDIT permits to edit the sequence in the stack editor whereas the button Notepad opens a notepad with the sequence in ASCII-format.

6. Refraction Index Editor

This editor serves to define the refraction indices of all materials needed to build the layer stack, i.e., the substrate, the superstrat and all kinds of layers between. It is available wherever required within the stack editor. To call the editor one has simply to click the related EDIT-button.

Related to the three basic possibilities of describing the materials, the appearance of the editor changes correspondingly. After being called, it comes always up in the direct refraction index input mode as shown in Fig. 43.

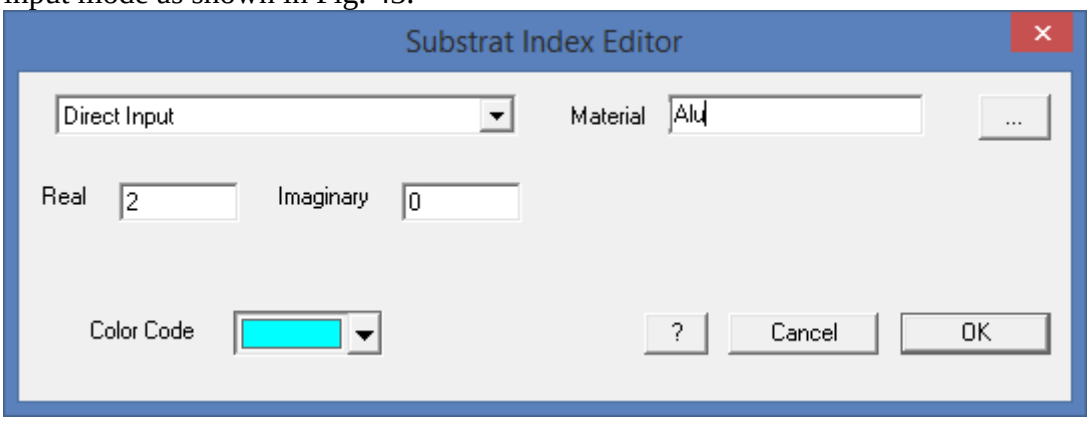


Fig. 43: Refraction index editor

Here, the real and the imaginary part of the refraction can be inserted directly. Then, this values are taken for all related calculations no matter what wavelength was specified. Apparently, in this way no dispersion is taken into account.

Moreover, a material ID can be assigned. While entering an ID name, Unigit checks the color table (see section 3.2) for this name and if being found presents the color assigned to it.

Other selections can be straightforwardly made by choosing the wanted option in the drop down box. This is shown in Fig. 44 along with all the possible options.

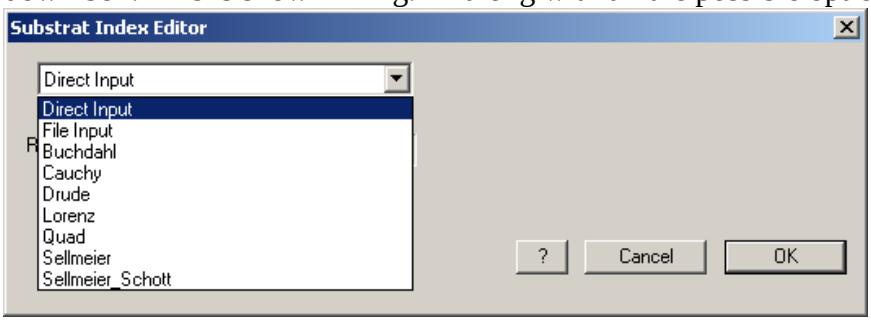


Fig. 44: Selection of the refraction index input mode

The available selections fall into three basic options:

- Beside the direct input, one can
- get the index information from a file,

- or have it computed from a certain dispersion formula (Buchdahl, Cauchy etc.).

If you pick the file input, the index editor will have the look depicted in Fig. 45.

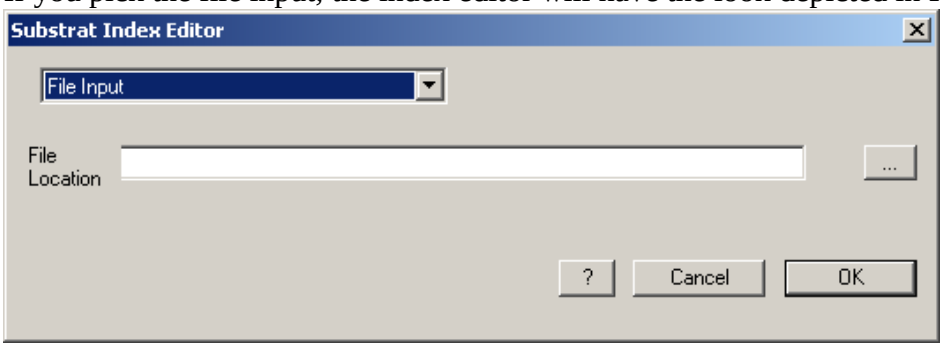


Fig. 45: Refraction input via nk-file selection

Here, the complete path name of the file with the required index data has to specified. This can either be done by typing the pathname into the corresponding field (file location) or by clicking the file request button (marked by "…"). In the letter case a file selection box pops up and one has to make ones choice. Of course, the file location is not fixed. However, it is recommended to use the folder "...\Unigit\NKData" that is generated during the installation of the UNIGIT code. Basically, the file extension ".nk" is preferred to recognize refractive index data. The Unigit installation comes with a couple of .nk example files. Feel free to complete this library according to your needs.

Furthermore, it is possible to connect a n&k file directly with a material ID. This can be done by replacing the comment line (first line of the file) with the code word "color" followed by a colon and the material ID name (see example below in Fig. 46). Then, when the n&k file will be used somewhere in the stack, Unigit checks the comment line, when identifying a material ID compares it with the color table and when successful assigns a color to it.

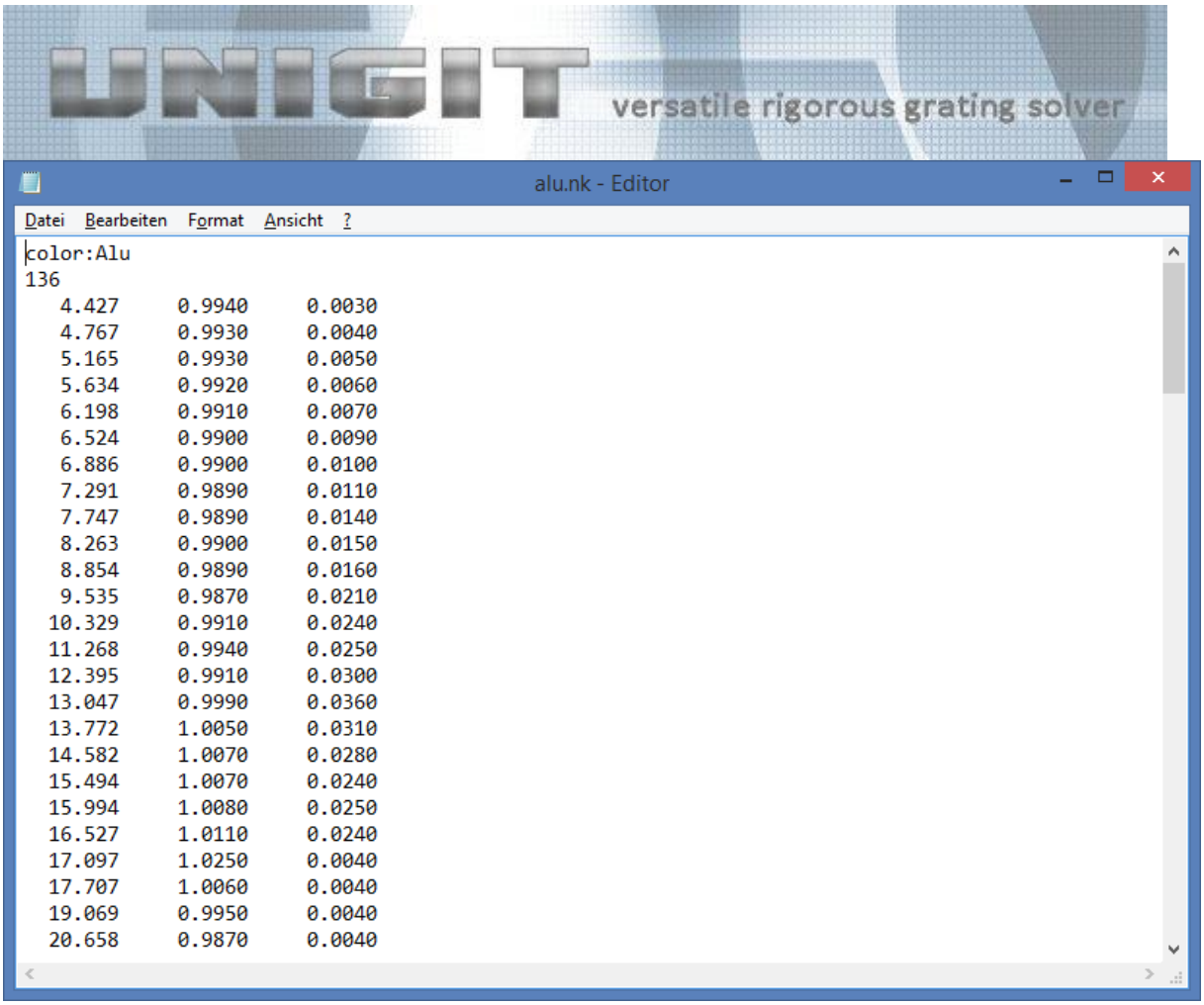


Fig. 46: n&k file listing with material ID for color coding (indicated by key word "color:")

Moreover, the refractive index can be defined by means of a certain dispersion formula. Then, the input mask looks like shown in Fig. 47.

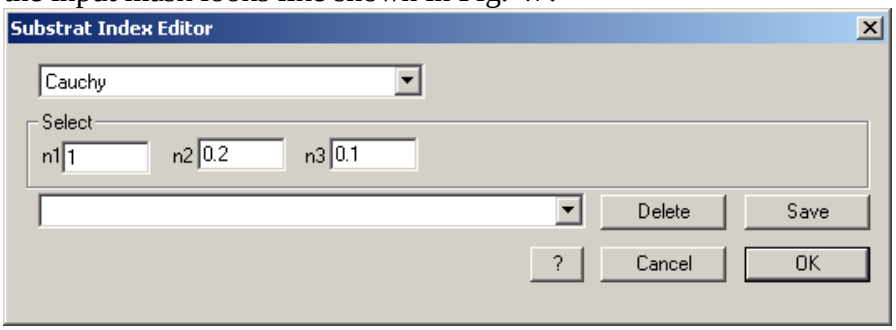


Fig. 47: Refraction input via dispersion formula

Depending on the selected dispersion formula, the coefficients n1 thru n3,4 or 5 as required by the related dispersion formula have to be entered. Make sure that the coefficients are related to the wavelength as given in microns rather than nanometer, Angstrom or other length measures. Another possibility is to select a certain coefficient set from the combination field. In addition, once the coefficients are entered, the present coefficient set can be saved. In order to be able to recall the set properly, a name has to be entered in the select box. Then, it is available under this name for future operations. If it is not needed anymore, it should be deleted again. The coefficient information is stored in one file per dispersion formula (with the related file name). These files are also located in the folder "...\Unigit\NKData" and can be identified by the file extension ".nkk".

A list of the available dispersion formulas is shown below:

- **Buchdahl:** 4 coefficients required: n1 ... n4

$$n = n_1 + B \cdot (n_2 + B \cdot n_3) \quad \text{and} \quad k = 0$$

with:

$$A = 0.001 \cdot (\lambda - n_4) \quad \text{and} \quad B = \frac{A}{(1 + 2.5 \cdot A)}$$

- **Cauchy:** 3 coefficients required: n1 ... n3

$$n = n_1 + \frac{n_2}{\lambda^2} + \frac{n_3}{\lambda^4} \quad \text{and} \quad k = 0$$

- **Drude:** 3 coefficients required: n1 ... n3

$$n = \sqrt{0.5 \cdot (A + B)} \quad \text{and} \quad k = \sqrt{0.5 \cdot (A - B)}$$

with:

$$A = \sqrt{B^2 + C^2} \quad \text{and} \quad B = n_1 - \lambda^2 \cdot \frac{n_3^2}{D}$$

$$C = \lambda^3 \cdot \frac{n_3}{D} \quad \text{and} \quad D = n_2^2 \cdot (\lambda^2 + n_3^2)$$

- **Lorentz:** 4 coefficients required: n1 ... n4

$$n = \sqrt{n_1 + k^2 + n_2 \cdot \lambda^2 \cdot \frac{A}{B}} \quad \text{and} \quad k = \frac{0.5}{n} \cdot n_2 \cdot n_4 \cdot \frac{\lambda^3}{B}$$

with:

$$A = \lambda^2 - n_3^2 \quad \text{and} \quad B = A^2 + n_4^2 \cdot \lambda^2$$

- **Quad:** 6 coefficients required: n1 ... n6

$$n = n_1 + n_2 \cdot \lambda + n_3 \cdot \lambda^2 \quad \text{and} \quad k = n_4 + n_5 \cdot \lambda + n_6 \cdot \lambda^2$$

- **Sellmeier:** 5 coefficients required: n1 ... n5

$$n = \sqrt{1 + \frac{n_1}{1 + \frac{n_2}{\lambda^2}}} \quad \text{and} \quad k = \frac{n_3}{n \cdot n_4 \cdot \lambda + \frac{n_5}{\lambda} + \frac{1}{\lambda^3}}$$

7. Output Editor

7.1. Basic features

The output editor is started either by the EDIT button in the "Output" group of the Central Control Board. It brings up the output editor window as shown in Fig. 48.

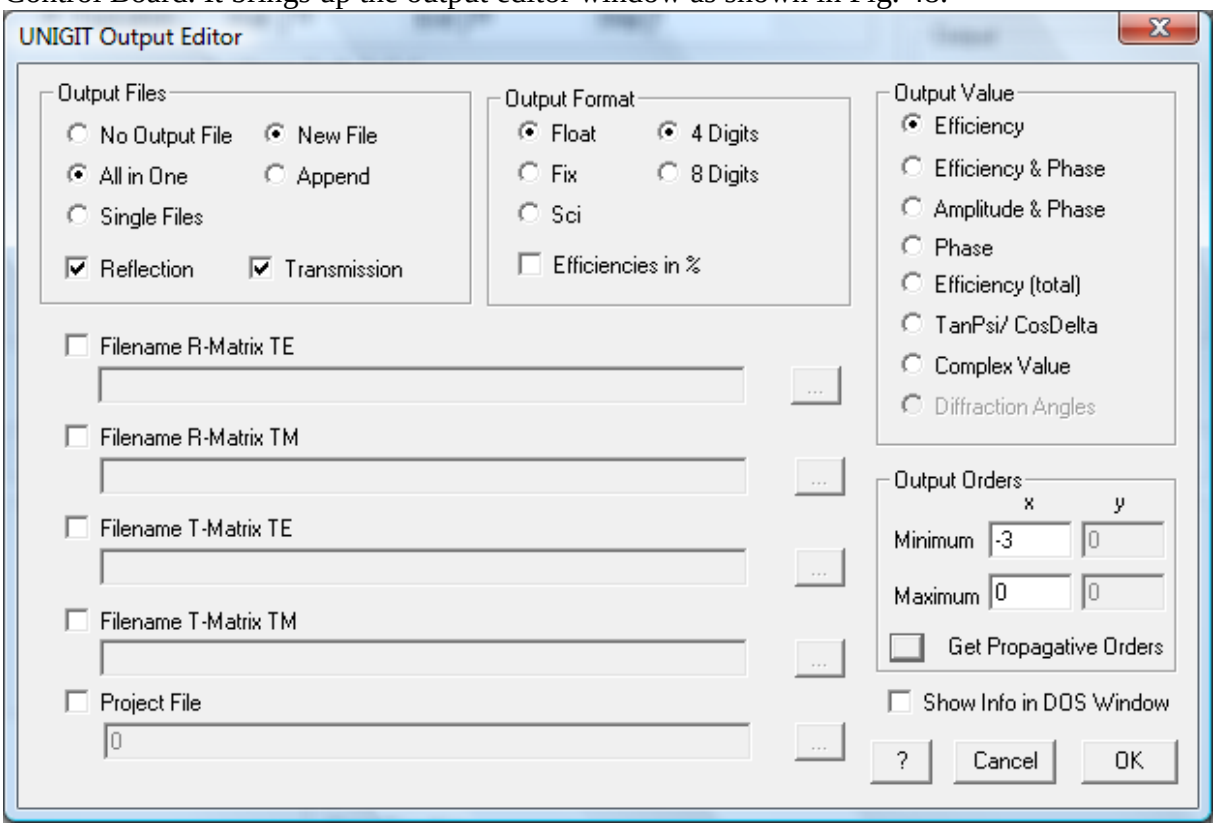


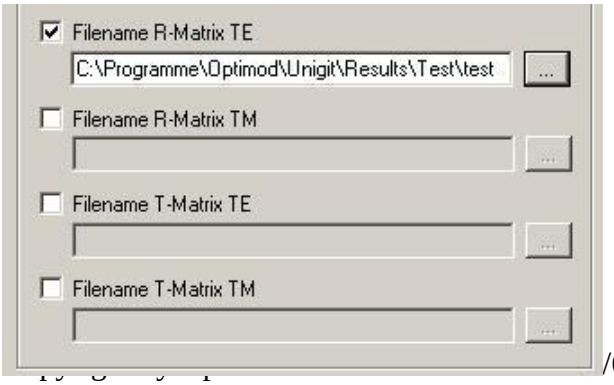
Fig. 48: Output control editor

Here, the following choices can be made:

- selection where to write the output to (radio buttons top left)
  - "No output file" means the result appears in the DOS window
  - "All in one" means the result, i.e., all selected diffraction orders are written into one file. This file can be specified in the Central Control Board in the text box "Save result in".
  - "Single Files" means that each diffraction order is saved in a separate file. The file location is specified as before, however, the name is used as a template and completed by different attachments depending on the order, whether it is reflection or transmission, the polarisation and the type of output information. It also depends on which solver was specified (1D or 2D).
    Basically, the following coding is used:
    Assume the basic name as specified in the text box is "test". Then, the file name is testOXOY.MPP where
    - OX is the diffraction order in x-direction,

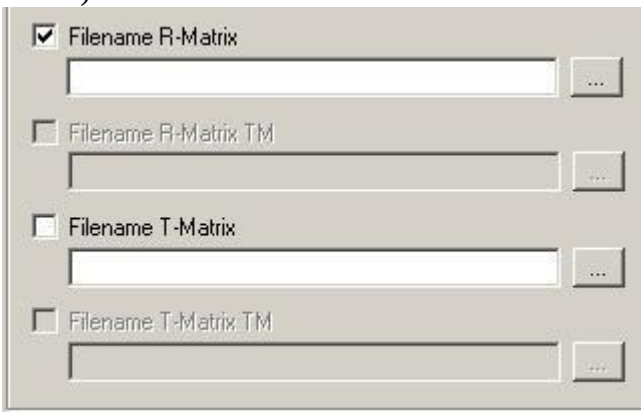
  - OY is the diffraction order in y-direction (is obsolete for 1D case and is skipped there),
  - M is the mode, i.e., R for reflection or T for transmission,
  - PP is the polarisation, first P for input and second P for output. P can either be E for TE or M for TM polarisation. To give an example, cross polarisation from TE to TM is characterized by PP = EM whereas TE polarisation is assigned by PP = EE. The second P is obsolete for 1D classic case (there is no cross polarisation) and is thus skipped.
  - As an example, test00m1.tmm means transmission of TM in 0./ -1. order (x/y).
- Selection of the Mode (checkbox transmission/reflexion),
- Selection whether depend the results to existing file(s) or create new one(s).
- Specification of the output format:
  - Select whether to output efficiencies in per cent or related to 1 (1 means 100%) with the "Efficiencies in %" checkbox.
  - Select the way of number output (Float, Fix or Scientific),
  - Select the number of digits (4 or 8).
- Selection of the output values (radio button group top right):
  - "Efficiency" outputs diffraction efficiencies in % or related to 1 (see above),
  - "Efficiency & Phase" same as before plus the absolute phases,
  - "Amplitude & Phase" outputs the absolute value of the complex amplitude plus the absolute phase (no "efficiency correction"),
  - "Phase" outputs only the absolute phase,
  - "Efficiency total" summarizes the efficiencies of both polarisations in the output channel, e.g., TMM & TEM.
  - "TanPsi/ CosDelta" outputs the ellipsometry values only for zeroth order (Note: TanPsi is output as log value).
  - "Complex Value" outputs real and imaginary value of the complex amplitude.
- Selection of the output orders: here you can determine the orders to be output. The input has to be done in a from (minimum) order to (maximum) order manner. In the example shown in the output editor, the –1. and the 0. order are output for a 1D stack. Another example for 2D is shown below. Here, the following order combinations are output: -1/-1; -1/0; -1/+1; -1/+2; 0/-1; 0/0; 0/+1 & 0/+2. Moreover, the propagative orders (in reflection) can be determined and automatically filled-in by clicking the *GET PROPAGATIVE ORDERS* button.

- Eventually, the complete diffraction matrices for reflection and/or transmission can be output by checking the corresponding check boxes and determine the location to where to save the data.



The figure shows the example where the reflection matrix for TE polarisation is saved the folder ..\Unigit\Results\Test\test.RTE. This matrix can be used for further computations such as the convolution with the angular plane wave spectrum of an incident beam resulting in the diffracted angular spectrum after passing the grating.

Please, note that in the case of conical diffraction, the polarisation does not decouple and therefore the diffraction matrices comprise all four polarisation combinations. In this case, only a reflection and/or transmission matrix can be selected for output (see below).



Finally, you need to check the "Show Info in DOS window" in order to display some auxiliary information during computation. If you check this box, the DOS window will get the focus otherwise it will remain in the background.

A context sensitive help for the output editor can be accessed by means of the "?" button (Note, the same applies to all other locations of the software, i.e., it will always directly jump to the relevant content of the help by hitting the help button.). After confirming with the OK button, the output control data are saved to the output.ctr file. Basically, it is possible to load output control files with different name. In this way, one can create a library of output control files for different standard situations.

7.2. Project file generation

In order to store all input information as well as the complete result information generated during a UNIGIT run (except the memory consuming diffraction matrices), the checkbox "Project File" has to be activated and a destination has to be entered where to save to the file (see Fig. 49). It receives automatically the extension ".upr" to be clearly identified as UNIGIT Project File. The file format is ASCII. The information is stored in three consecutive blocks. The first block contains the input comprising wavelengths, AOI, order truncation and polarisation. It is followed by the grating or stack information. Finally, the results are stored as complex amplitudes for all propagative orders and all cases according to the loop specification. Read more about the UNIGIT project handling and its features in section 8.

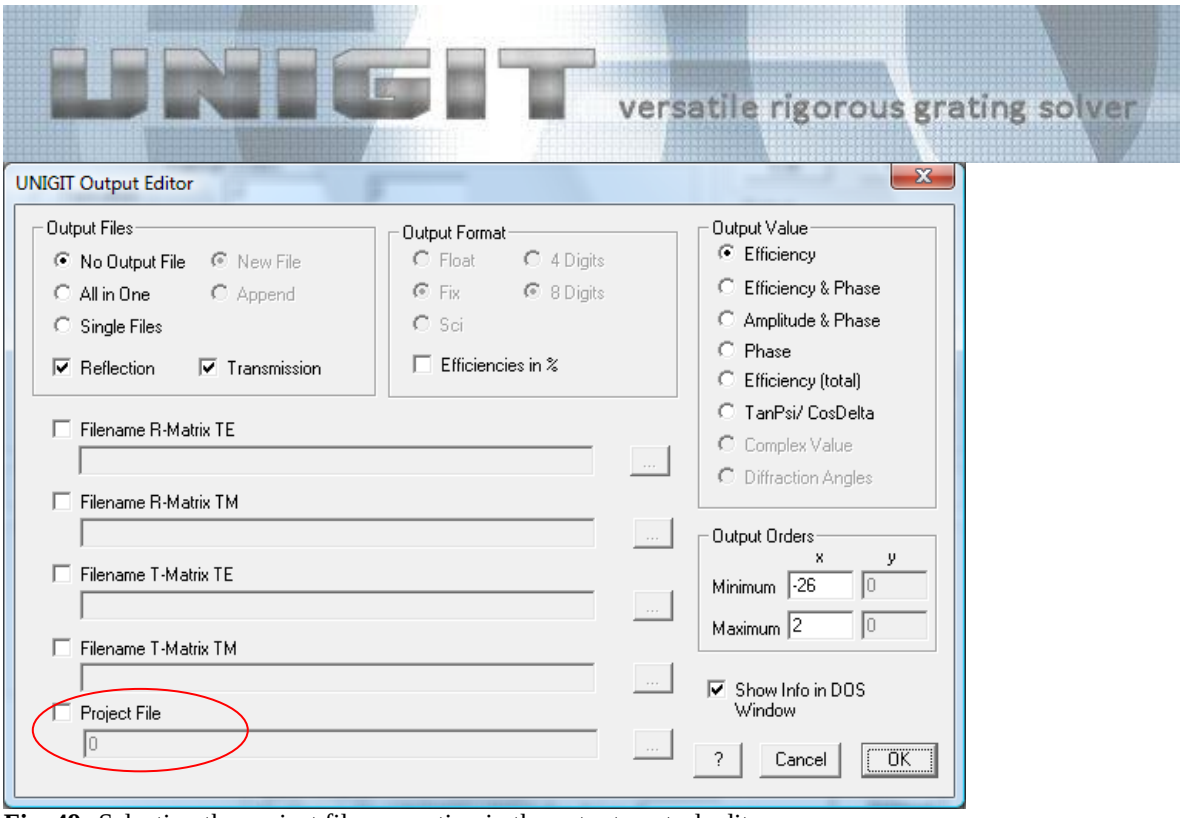
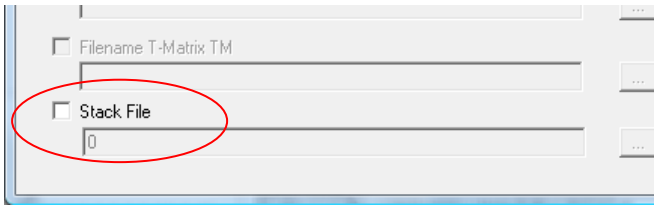


Fig. 49: Selecting the project file generation in the output control editor

7.3. Project retrieval mode

The project retrieval mode is activated when loading a project file instead of a stack file. In the output editor, this becomes visible by the name change from “Project File” (compare section 7.1) to “Stack File” (see figure below).



In case you want to retrieve the stack file from a Unigit project file (see section 8.3), you have to check this box and enter a file name where to save the stack information to. The extension is forced to the standard “.ust” if not specified.

8. UNIGIT Projects

8.1. General Remarks

Unigit project files have been introduced to facilitate the organization of complex modeling projects as well as to give some support in retrieving and reconstructing simulations dated back some time. Concurrently, it provides an appropriate mean to avoid time consuming reruns of the UNIGIT solver in order to obtain a different output format for the same application example. Suppose for instance you have calculated the diffraction efficiencies for a complex 2D example that took several hours. Later, it turns out that you also need the phase information. Then, it would have been a good idea to activate the project file generation in the first run (see section 7.2).

Unigit project files can load instead of stack files (see Fig. 50). It possesses the extension ".upr".

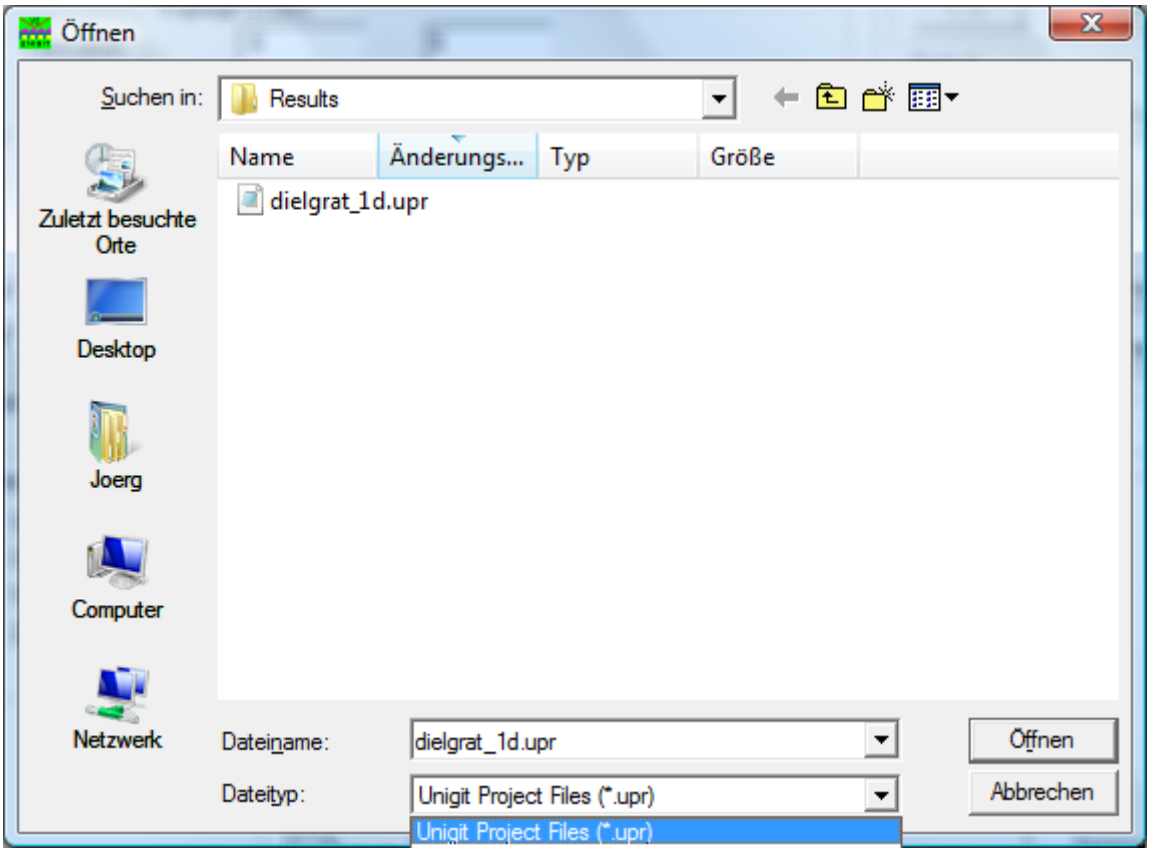


Fig. 50: Selection of an Unigit project file

After selecting an UPR file, UNIGIT switches from its regular mode into the project retrieval mode. This becomes obvious by two features:

First, the group in the upper right corner changes its name from “Stack” to “Project” and secondly, the START button changes its name into “Extract Results”. In addition, the buttons in the project control group (middle right) are enabled (see Fig. 51).

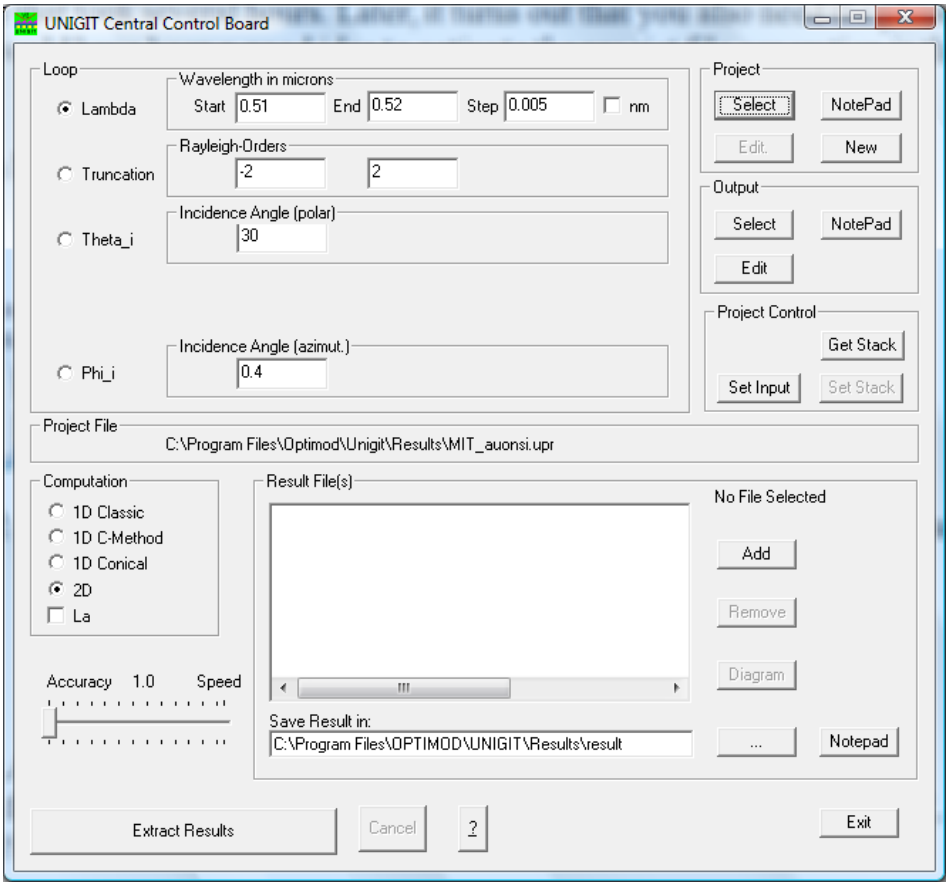


Fig. 51: Appearance of the central control board in project retrieval mode

8.2. Retrieving input information

The input information is simply retrieved by clicking the button SET INPUT in the group “Project Control”. Then, all input masks such as “Lambda”, “Truncation”, “Theta_i” and “Phi_i” are updated with the original state used to generate the active project file. Likewise, the loop setting is updated along with the associated start, stop and step parameters.

8.3. Retrieving stack (grating) information

The retrieval of the original stack information is a two-step procedure. In the first step, the stack information is retrieved and saved to a file by means of the GET STACK button. To this end, a file name has to specified by means of the output editor (see section 7.3). After successful operation, the SET STACK button is enabled.

In the second step, the stack file can be set as active stack file and the basic operation mode of Unigit can be switched back to regular by clicking the SET STACK button. Concurrently,

Unigit identifies whether the stack file is a valid Unigit format and issues an error message if this is not the case.

8.4. Retrieving computation results

Another very important functionality of UNIGIT's project retrieval mode is the generation of arbitrary result information from a given project file. This is simply done by clicking the EXTRACT RESULTS button. The way how the output is organised has to be set in exactly the same way as for the regular mode. Again, this is done by means of the output editor (see section 7.1) in exactly the same way as for the regular mode. Actually, you will see only subtle change in appearance when starting the output editor (described elsewhere). In addition, some options will be disabled (output only in one file). Analogously, either single files (one for each diffraction order), one big file (all in one) or no file meaning the output is displayed in a DOS window can be generated. The essential difference is that the information is retrieved from a sort of archive (UPR file) instead of being generated by a solver. Thus it is much faster. Of course, all onboard features for data exploration, i.e., starting a notepad to display the ASCII formatted result files or the diagram feature can be applied the usual way (see section 2.5).

References

/1/ M.G. Moharam, D.A. Pommet and E.B. Grann: "Stable implementation of the rigorous coupled wave analysis for surface-relief gratings: enhanced transmittance matrix approach", J. Opt. Soc. Am. A 12 (1995), pp. 1077-1086.

/2/ Lifeng Li, "Multilayer modal method for diffraction gratings of arbitrary profile, depth, and permittivity," J. Opt. Soc. Am. **A 10** (1993), pp. 2581-2591.

/3/ D. Agassi and T.F. George, "Convergent schema for light scattering from an arbitrary deep metallic grating," Phys. Rev. B **33** (1986), pp. 2393-2400.

/4/ Lifeng Li, "Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings," J. Opt. Soc. Am. **A 13** (1996), pp. 1024-1035.

/5/ Lifeng Li, "Use of Fourier series in the analysis of discontinuous periodic structures," J. Opt. Soc. Am. **A 13** (1996), pp. 1870-1876.

/6/ Lifeng Li, "New formulation of the Fourier Modal Method for crossed surface-relief gratings," J. Opt. Soc. Am. **A 14** (1997), pp. 2758-2767.

/7/ P. Lalanne, "Improved formulation of the coupled wave method for two-dimensional gratings," J. Opt. Soc. Am. **A 14** (1997), pp. 1592-1598.

/8/ J. Chandezon, D. Maystre and G. Raoult, "A new theoretical method for diffraction ratings and its numerical application,'' J. Opt. (Paris) **11**, 235–241 (1980).

/9/ L. Li, J. Chandezon, G. Granet, and J.-P. Plumey, "Rigorous and efficient grating analysis method made easy for optical engineers," Appl. Opt. **38**, 304-313 (1999).

/10/ J. Bischoff, "Improved diffraction computation with a hybrid C-RCWA method," Proc. SPIE 7272, part 2 (2009).
See e.g. http://www.osires.biz/page6.php - reference #5

Acronyms

AOI	-	Angle of Incidence
AR	-	Anti Reflective
BARC	-	Bottom Anti Reflective Coating
CCB	-	Central Control Board
CD	-	Critical Dimension
CM	-	Coordinate Transformation Method (a.k.a. C-method)
GUI	-	Graphical User Interface
PR	-	Photo resist
RCWA	-	Rigorous Coupled Wave Approach
RF	-	Rayleigh-Fourier (Method)
TE	-	Transverse Electrical
TM	-	Transverse Magnetical
1D	-	one-dimensional
2D	-	two-dimensional
3D	-	three-dimensional