DORT2002 version 2.0
User Manual

Per Edström
Marcus Lehto
Mid Sweden University

FSCN Report – ISSN 1650-5387 2003:18
Internal FSCN Report Number: R-03-43
April 2003

Mid Sweden University
Fibre Science and Communication Network
SE-851 70 Sundsvall, Sweden

Internet: http://www.mh.se/fscn

Contents Page

DORT2002 version 2.0 User Manual

Per Edström
Marcus Lehto
Mid Sweden University, S-871 88 Härnösand

Abstract

The DORT2002 software is a fast and accurate tool for solving radiative transfer problems in vertically inhomogeneous turbid media, using a discrete ordinate model geometry. DORT2002 is implemented in MATLAB, and is adapted to light scattering simulations in paper and print.

The DORT2002 Graphical User Interface is a tool developed to provide users with a fast and easy way of performing DORT2002 simulations. It works as a shell that encapsulates the parameters and the function calls, and offers powerful simulations through a mouse click.

This report gives a thorough description of how to install and use DORT2002, version 2.0, and the Graphical User Interface.

1. Introduction

DORT2002 is a numerical implementation of a discrete ordinate solution method for the radiative transfer problem in vertically inhomogeneous turbid media. The model is based on papers by Edström [1] and Edström and Lehto [2], which are recommended for those who whish to get a thorough understanding of the model and its features. DORT2002 has been developed during the past year and is now a fast and accurate tool for light scattering simulations in paper and print applications.

The light scattering simulations can be performed using a Graphical User Interface, which makes the simulations convenient and accessible. It is also possible to call DORT2002 directly from the MATLAB Command Window, which may be preferred by the experienced user. Finally it is possible to embed DORT2002 as a component in a script or a larger application.

This report is a user manual for DORT2002 version 2.0 and focuses on how to use DORT2002 software rather than describing the theories and implementation in detail. Furthermore, the main focus is on the usage of the DORT2002 - Graphical User Interface, since the command line calls are mainly for advanced users with a basic knowledge of MATLAB function calls.

Section 1 gives references and some general information, section 2 describes how to install DORT2002, sections 3 and 4 describe how to use DORT2002 via the Graphical User Interface and from the MATLAB Command Window respectively, and section 5 briefly describes some more advanced features included in DORT2002. Tables of the parameter and result structs are given in appendices, together with a description of the plot options.

2. Installation

2.1. System Requirements

Version 2.0 of DORT2002 requires the following from your system:

- Microsoft Windows 98 (original or Second Edition), Windows Millennium Edition (ME), Windows NT 4.0, Windows 2000, or Windows XP.
- MATLAB 6.5.
- See also the MATLAB system requirements.

2.2. Installation

DORT2002 is packaged in a zip file containing the software, this manual, help files and reports on the theory behind the solution method. To install DORT2002, extract the zip file into a folder on the target computer. Note that all files must be extracted into the same folder.

2.3. Getting Started

The DORT2002 software can only be used in a MATLAB environment. Therefore, MATLAB has to be opened before using DORT2002.

To access DORT2002, the MATLAB current directory must be changed to the directory containing the DORT2002 files, or the directory of the DORT2002 files must be added to the MATLAB search path.

The DORT2002 simulations can be performed in two ways, either through the Graphical User Interface by executing the *dort2002interface.p* script (see section 3), or directly from the MATLAB Command Window by executing the *dort2002.p* script (see section 4).

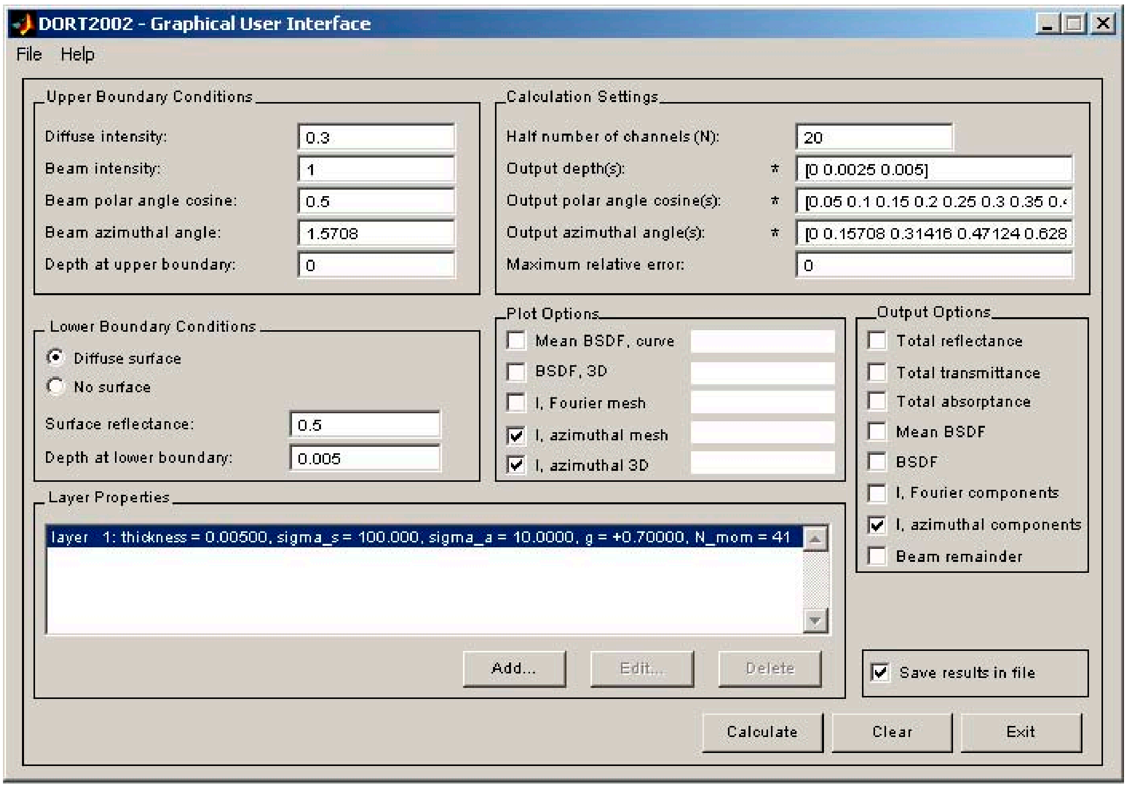
3. DORT2002 Graphical User Interface

The DORT2002 Graphical User Interface is a tool developed to provide users with a fast and easy way of performing DORT2002 simulations. It works as a shell that encapsulates the parameters and the function calls, and offers powerful simulations through a mouse click. The interface calls the command line version of DORT2002 in the background.

Most users will prefer the Graphical User Interface, since it is so convenient. However, advanced users may prefer the direct control of all parameters from the command line, and the possibility to use DORT2002 as a component in scripts or larger applications.

3.1. Getting Started

The DORT2002 Graphical User Interface is started by executing the *dort2002interface.p* script in MATLAB. When executed, the main window will appear on the screen, see figure 1.



Upper Boundary Conditions

Diffuse intensity: 0.3
Beam intensity: 1
Beam polar angle cosine: 0.5
Beam azimuthal angle: 1.5708
Depth at upper boundary: 0

Calculation Settings

Half number of channels (N): 20
Output depth(s): * [0 0.0025 0.005]
Output polar angle cosine(s): * [0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.
Output azimuthal angle(s): * [0 0.15708 0.31416 0.47124 0.628
Maximum relative error: 0

Lower Boundary Conditions

● Diffuse surface
○ No surface
Surface reflectance: 0.5
Depth at lower boundary: 0.005

Plot Options

□ Mean BSDF, curve
□ BSDF, 3D
□ I, Fourier mesh
☑ I, azimuthal mesh
☑ I, azimuthal 3D

Output Options

□ Total reflectance
□ Total transmittance
□ Total absorptance
□ Mean BSDF
□ BSDF
□ I, Fourier components
☑ I, azimuthal components
□ Beam remainder

Layer Properties

layer 1: thickness = 0.00500, sigma_s = 100.000, sigma_a = 10.0000, g = +0.70000, N_mom = 41

☑ Save results in file

Figure 1. The DORT2002 Graphical User Interface

If it does not start, check that all components are properly installed, that the MATLAB current directory or search path is set to the folder containing all your DORT2002 files, and that the MATLAB version used is version 6.5 or higher.

To get further help, choose *Help…* from the Help menu to open this manual.

3.2. Performing Simulations

When a DORT2002 simulation is to be performed, all parameter fields in figure 1 must be given manually or by opening a predefined parameter file, see section 3.3.3. The ranges of the input parameters are given in Appendix A.

Note that the fields *Beam polar angle cosine*, *Beam azimuthal angle*, *Output depth(s)*, *Output polar angle cosine(s)* and *Output azimuthal angle(s)* can be filled in using MATLAB expressions. This makes it possible to use MATLAB variables and functions such as `pi` and `cos`. The fields requiring vector input must be defined using MATLAB vector expressions.

3.2.1. Boundary Conditions

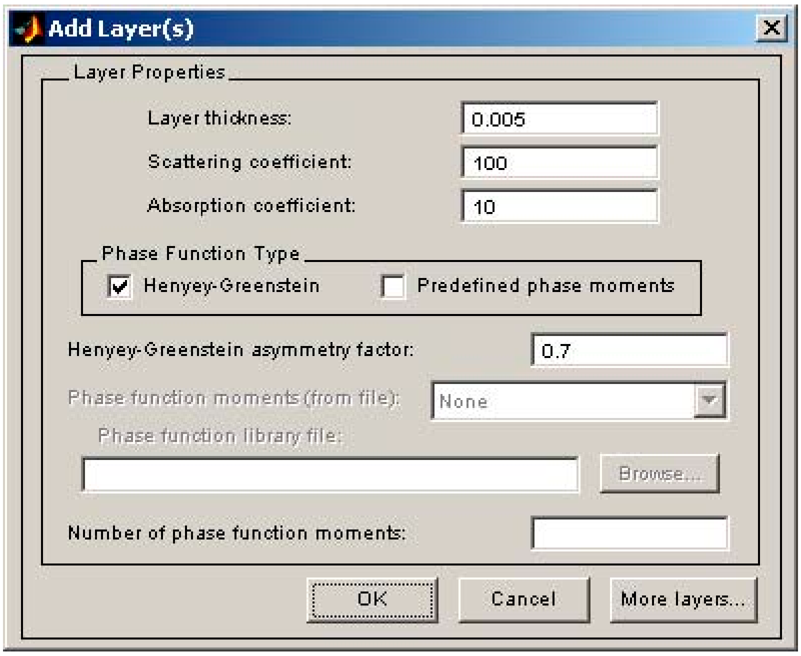
The upper boundary conditions to be defined are the incident illumination, e.g. any incident diffuse intensity and the direction and intensity of any incident beam.

At the lower boundary, the bottom surface type has to be defined. Note that the bottom surface is *not* the lower surface of the medium, but the surface of the underlying medium.

There are two optional types of the underlying surface, a perfectly diffuse surface or no surface. If a diffuse underlying surface is selected, the total surface reflectance of that surface must be defined.

3.2.2. Layer Properties

Adding new layers and setting their properties is carried out by pressing the *Add...* button in the main window. This activates the *Add Layer(s)* dialog window shown in figure 2.



Layer Properties

Layer thickness: 0.005
Scattering coefficient: 100
Absorption coefficient: 10

Phase Function Type

☑ Henyey-Greenstein ☐ Predefined phase moments

Henyey-Greenstein asymmetry factor: 0.7

Phase function moments (from file): None

Phase function library file:

Number of phase function moments:

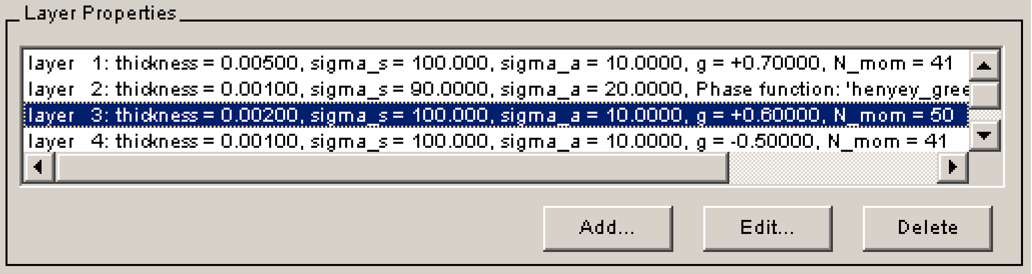
Figure 2. The window for defining layer properties

The first three fields in figure 2 define thickness, scattering coefficient and absorption coefficient of the given layer. The DORT2002 software handles all types of phase functions, but only Henyey-Greenstein phase function moments are generated internally. Therefore, the phase function moments have to be supplied for any other phase functions to be used, while only the asymmetry factor needs to be supplied for the Henyey-Greenstein case.

If other phase functions than Henyey-Greenstein are to be used, choose the *Predefined phase moments* alternative and open the phase function library file containing the desired phase function moments. The fieldnames defining the phase function moments included in the opened file will be shown in the *Phase function moments* pop-up menu and the wanted field can be selected. See section 3.3.4 for more information about phase function library files.

The parameter field *Number of phase moments* specifies the number of phase moments to be used in the intensity correction procedures. If left empty, the number of phase moments will be set automatically for a high accuracy. See section 5.1 for more information on the intensity correction procedures.

When all parameter fields are filled in, press *OK* to return to the main window or press *More layers…* to define properties for the next layer. The layer properties are displayed in the *Layer Properties* list box, see figure 3, and are updated for every new layer defined.



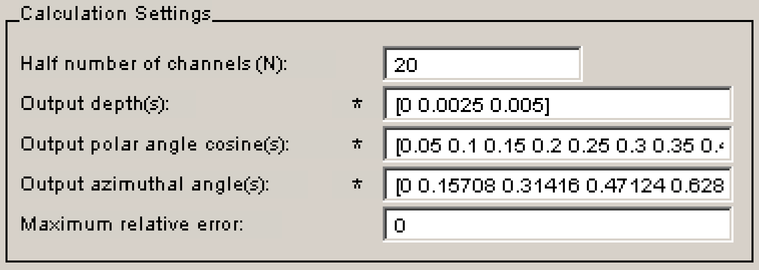
```
layer  1: thickness = 0.00500, sigma_s = 100.000, sigma_a = 10.0000, g = +0.70000, N_mom = 41
layer  2: thickness = 0.00100, sigma_s = 90.0000, sigma_a = 20.0000, Phase function: 'henyey_gree
layer  3: thickness = 0.00200, sigma_s = 100.000, sigma_a = 10.0000, g = +0.60000, N_mom = 50
layer  4: thickness = 0.00100, sigma_s = 100.000, sigma_a = 10.0000, g = -0.50000, N_mom = 41
```

Figure 3. The Layer Properties list box

The defined layer properties can be edited by selecting the layer to edit and pressing the *Edit…* button. The *Add Layer(s)* dialog window then appears with the chosen layer properties filled in. After editing, press *OK* and the updated layer properties will be displayed in the *Layer Properties* list box. Any layer can be deleted by selecting the specific layer and pressing *Delete*. More layers can always be added by pressing the *Add…* button.

3.2.3. Calculation Settings

The *Calculation Settings* fields found in figure 4 specify the accuracy of the calculation, for which depths and angles the results should be calculated and any convergence criterion of the simulation.



Calculation Settings

Half number of channels (N): 20

Output depth(s): * [0 0.0025 0.005]

Output polar angle cosine(s): * [0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.

Output azimuthal angle(s): * [0 0.15708 0.31416 0.47124 0.628

Maximum relative error: 0

Figure 4. The Calculation Settings frame

Since understanding of the calculation settings is crucial for the user, these parameters are described more in detail below.

The parameter *Half number of channels (N)* specifies the number of channels, a feature of the discrete ordinate model geometry, and a parameter of the core calculations. A low number of channels decreases the accuracy of the results, but gives faster calculations. A larger number of channels increase both accuracy and calculation time. In order to make efficient simulations, the user should be aware of approximately how many channels that are needed in a specific calculation. Performance tests show that DORT2002 converges for $N = 15\text{-}20$ in most cases. This means that a larger number of channels would not increase the accuracy of the results (since the results already are fully accurate within the DORT model geometry), but the calculation time would still increase. The choice of N thus affects the accuracy and calculation time, but it has nothing to do with the angular resolution of the results.

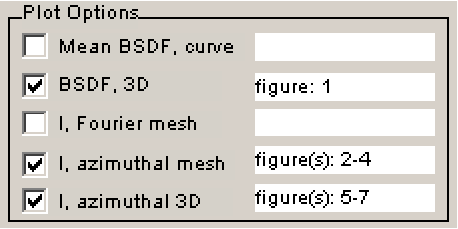
The parameters *Output depth(s)*, *Output polar angle cosine(s)* and *Output azimuthal angle(s)* specify for which depths and for which polar and azimuthal angles the results are to be calculated. Depending on the chosen plot and output options, some or all of these properties do not need to be defined. *Output depth(s)* only affects intensity and beam, *Output azimuthal angle(s)* only affects BSDF and intensity, and *Output polar angle cosine(s)* affects mean BSDF, BSDF and intensity. The parameter fields needed for the chosen plot and output options are therefore marked with an asterisk, see figure 4. This only applies to these three parameter fields and no other parameter fields in the Graphical User Interface.

Note that the user supplied output depths, polar and azimuthal angles are entirely decoupled from the predefined channels in the core calculations, and a high angular resolution does *not* require a large number of channels. In fact, it is one of the main features of DORT2002 to offer high accuracy and resolution at a small number of channels, thus giving a large decrease in computation time.

The parameter field *Maximum relative error* specifies the convergence criterion of a given calculation. If left blank or set to zero, the calculations will be performed using maximum possible accuracy. If the full accuracy is not needed, a choice of lower maximum relative error will give a significant decrease in computation time in most cases. This parameter should however be used with caution, see section 5.2.

3.2.4. Plot Options

DORT2002 offers to present the results in various plots, which can be selected in the *Plot Options* frame, see figure 5.



Plot Options

☐ Mean BSDF, curve

☑ BSDF, 3D figure: 1

☐ I, Fourier mesh

☑ I, azimuthal mesh figure(s): 2-4

☑ I, azimuthal 3D figure(s): 5-7

Figure 5. The Plot Options frame showing figure numbers for desired plots

More detailed information on the different plot options is given in Appendix B.

When the simulation is carried out, the figures numbers of any generated plots will be displayed in the *Plot Options* frame. The figure numbers for each plot are given in ascending order of calculation depths, i.e. the lowest figure number corresponds to the smallest calculation depth etc.

3.2.5. Output Options

The output options define which results to be calculated and returned as output from the simulation. More detailed information on the fields in the result struct is given in Appendix C.

To save the requested output in a file, the *Save results in file* option must be selected, see figure 1. If it is not selected, the output can still be retrieved inside the current MATLAB workspace, see section 3.2.10.

3.2.6. Calculate

When all input parameter fields are filled in, press the *Calculate* button or choose *Calculate* from the File menu. If the parameters are valid, the simulation will be performed and the desired output will be generated. If any of the parameters defined are invalid, an error message will be shown, and the parameter must be changed accordingly.

A note for Windows platforms

When a parameter field has been edited, the new value visible on the screen will not be read into the application until the user clicks a button, clicks the background of the Graphical User Interface or presses Return. It is *not* read into the application when clicking on the menu bar.

This means that editing a parameter field and directly choosing *Calculate* from the File menu will cause the old parameter value to be used, although it has been edited. To be sure to always use the new parameter value, the user should either use the *Calculate* button, or always click outside the parameter field or press Return before choosing *Calculate* from the File menu.

This unfortunate behavior is consistent with the Microsoft Windows platform conventions, and is *not* an error in DORT2002. On UNIX systems, clicking on the menu bar will cause the parameter to be read, so this problem does not exist there.

3.2.7. Save Results in File

If the *Save results in file* option is selected, the *Save output file(s)* dialog window is opened, see figure 6.

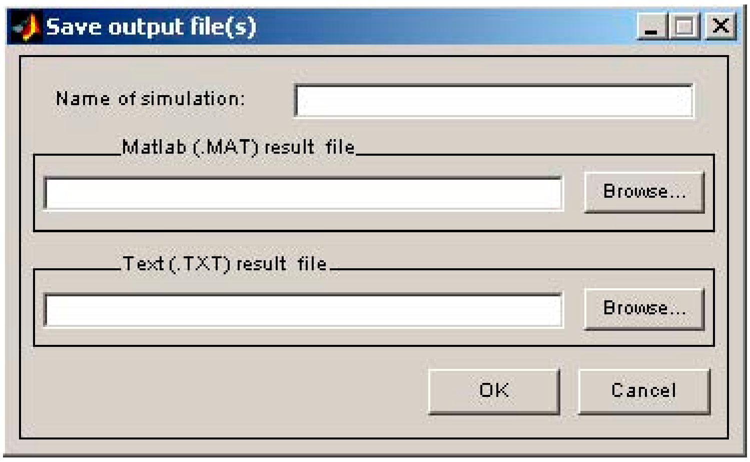


Figure 6. The window for saving output files

It is there possible to choose whether to save the results in a formatted text file, in a binary MATLAB file or both. The target directory and filename of the result file(s) are defined using the *Browse…* button(s). Leaving any of the fields *Matlab (.MAT) result file* and *Text (.TXT) result file* empty results in no file saving of that file type.

3.2.8. Clear

Pressing the *Clear* button or choosing *Clear* from the File menu erases all fields in the main window. This makes it easy to start from scratch. If a parameter file is opened, the fields are cleared, but the parameter file is unchanged.

3.2.9. Exit

Pressing the *Exit* button or choosing *Exit* from the File menu exits the DORT2002 Graphical User Interface.

3.2.10. Interaction with the MATLAB Command Window

There are two ways to retrieve the output from a calculation done in the Graphical User Interface into the current MATLAB workspace. One way is to load a MATLAB result file (.MAT) from a previous calculation. However, it is also possible to retrieve the DORT2002 output in the MATLAB Command Window without at all saving the results to a file.

The results of each DORT2002 simulation carried out using the Graphical User Interface are stored in the `global` variable `dort2002_output`. To get access to that variable, the variable must be declared as `global` in the current workspace. This is done using the following expression: `global dort2002_output.`

The struct `dort2002_output`, containing the chosen output, can then be used inside the current MATLAB workspace. Note that the variable `dort2002_output` is renewed for every calculation, e.g. only the results of the latest simulation are found in the variable.

If DORT2002 is used without the graphical user interface, the output is returned in a variable as from any MATLAB function: `result = dort2002(parameters)`.

3.3. Managing Parameter Files

Filling in all parameter fields manually for every simulation to be performed using the Graphical User Interface is quite time consuming. Therefore, there is a possibility for the user to save parameter files for later use. The parameters saved in a parameter (.MAT) file can later be opened in the interface and can, for example, be used as a template for new simulations or when redoing a specific simulation. It is always possible to save any changed parameters to a new parameter file.

3.3.1. Saving Parameter Files

When saving a parameter file all parameters fields must be filled in. It is only possible to save a full and valid set of parameters. The parameters and their valid ranges are given in Appendix A. The parameter file is saved by choosing *Save parameter file as…* from the File menu and defining the target directory and filename. The input parameters are checked to avoid input errors, and error messages are displayed if any invalid parameters are found.

The result (.MAT) files stored in previous simulations can also be used as parameter files, since the result files also contain the input parameters.

A note for Windows platforms

When a parameter field has been edited, the new value visible on the screen will not be read into the application until the user clicks a button, clicks the background of the Graphical User Interface or presses Return. It is *not* read into the application when clicking on the menu bar.

This means that editing a parameter field and directly choosing *Save parameter file as…* from the File menu will cause the old parameter value to be used, although it has been edited. To be sure to always use the new parameter value, the user should always click outside the parameter field or press Return before choosing *Save parameter file as…* from the File menu.

This unfortunate behavior is consistent with the Microsoft Windows platform conventions, and is *not* an error in DORT2002. On UNIX systems, clicking on the menu bar will cause the parameter to be read, so this problem does not exist there.

3.3.2. Creating Parameter Files in the MATLAB Command Window

A parameter file for the DORT2002 Graphical User Interface is simply a .MAT file containing a struct named `parameters`, which contains specified fields. The field names and their ranges are given in Appendix A.

Parameter files can be created outside the Graphical User Interface. This can be done using the *default_parameters.m* script file as a template. It is always possible to save any changed parameter set, specified in an m-file, into a .MAT file. The following example shows how to save the parameters defined in *my_parameters.m* into the file *filename.mat*:

```
parameters = my_parameters;
save('filename.mat','parameters');
```

3.3.3. Opening Parameter Files

A predefined parameter file is opened by choosing *Open parameter file...* from the File menu. If the parameters included in the chosen file are valid (as they must be if they were saved from the Graphical User Interface) the parameters will be extracted, otherwise an error message will appear on the screen.

3.3.4. Creating Phase Function Library Files

A phase function library file is a file containing phase function moments for any phase function to be used in DORT2002 calculations. The phase function moments are coefficients of the Legendre polynomial expansion of the phase function, and have to be supplied by the user. This can be done by separate calculation, or by looking up in a table.

The phase function file itself is simply a .MAT file containing a struct, `phase_function_moments`, which contains one or more fields with different phase function moments. Each field must be a row vector containing the phase function moments ordered by index in ascending order.

The following example shows how to create a phase function library file containing phase function moments for the Henyey-Greenstein phase function with six different asymmetry factors, and a Haze-L phase function with a specific asymmetry factor. The row vectors containing the phase function moments, denoted `r1-r7`, are assumed to be generated previously.

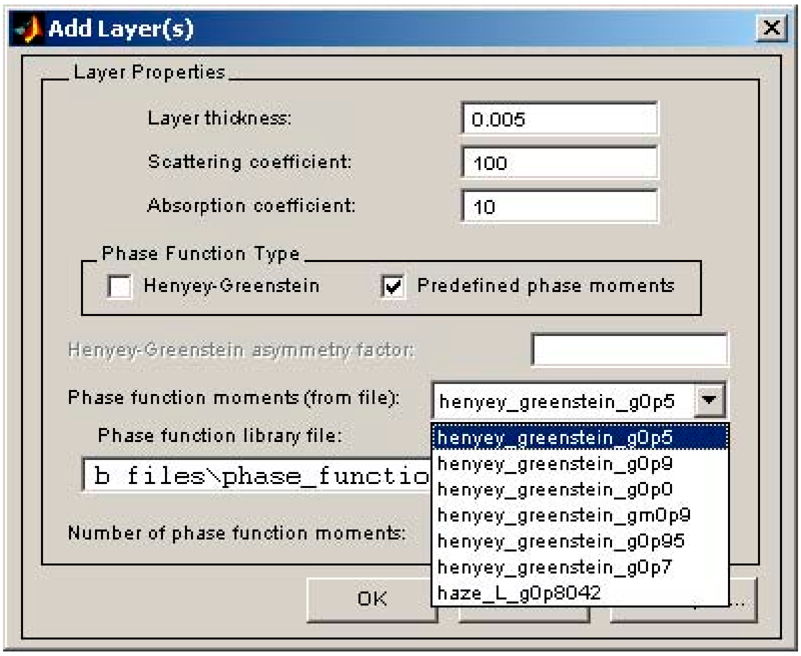
The struct is generated field by field:

```
phase_function_moments.henyey_greenstein_g0p5 = [r1];
phase_function_moments.henyey_greenstein_g0p9 = [r2];
phase_function_moments.henyey_greenstein_g0p0 = [r3];
phase_function_moments.henyey_greenstein_gm0p9 = [r4];
phase_function_moments.henyey_greenstein_g0p95 = [r5];
phase_function_moments.henyey_greenstein_g0p7 = [r6];
phase_function_moments.haze_L_g0p8042 =  [r7];
```

This gives a MATLAB struct with fieldnames *henyey_greenstein_g0p5*, *henyey_greenstein_g0p9*, *henyey_greenstein_g0p0*, *henyey_greenstein_gm0p9*, *henyey_greenstein_g0p95*, *henyey_greenstein_g0p7* and *haze_L_g0p8042*. For simplicity, the fieldnames should be chosen such that they give information of which phase function is used and the value of the asymmetry factor. A MATLAB struct of this kind can also be generated using the MATLAB struct command, see MATLAB help for more information.

The phase function library file is then created using the MATLAB save command:
`save('filename.mat','phase_function_moments').`

The saved phase function library file can then be opened in the Graphical User Interface and the field containing the wanted phase function moments can easily be selected, see figure 7.



Add Layer(s)

Layer Properties

Layer thickness: 0.005

Scattering coefficient: 100

Absorption coefficient: 10

Phase Function Type

□ Henyey-Greenstein ☑ Predefined phase moments

Henyey-Greenstein asymmetry factor:

Phase function moments (from file): henyey_greenstein_g0p5

Phase function library file:

b files\phase_functio

Number of phase function moments:

Figure 7. The window for defining layer properties when using predefined phase function moments

Note that the user is responsible for the validity of the phase function moments saved in this manner. Therefore, only users familiar with how phase function moments for a specific phase function are generated should use this option.

4. Running DORT2002 from the MATLAB Command Window

4.1. Input

When calling DORT2002 from the MATLAB Command Window a complete input parameter struct must be provided as input. If not, the missing input parameter fields will be set to default during the calculation. When creating an input parameter struct, the file *default_parameters.m* can be used as a template. The input parameters and their ranges are given in Appendix A.

4.2. Output

The output is returned in a variable as from any MATLAB function:

```
result = dort2002(parameters);
```

The output is a struct containing the desired output fields, as defined in the input parameters. The fields in the result struct are described in Appendix C.

The output can also be saved directly, as defined in the input parameters, in a formatted text file, a binary MATLAB file or both.

5. Notes on Some Advanced Features

5.1. The δ - N Method and Intensity Correction Procedures

To maintain accuracy without a tremendously increased computational burden for cases with strongly forward-peaked phase functions, the so-called δ-N method is used in DORT2002, together with single- and second-order scattering correction procedures, the TMS and IMS methods.

The intensity correction procedures uses, in a special way, a larger number of phase function moments than in the core calculations. The field `N_phase_moments` in the parameter struct sets the number of phase moments to be used in the intensity correction procedures. In the Graphical User Interface it is set through the *Number of phase moments* field in the *Add Layer(s)* dialog window.

If the *Number of phase moments* field is left empty in the Graphical User Interface, or if the field `phase_moments_check` is set to 1 in the input struct, the number of phase moments will be set automatically for a high accuracy. It is possible to set it to a lower number to get faster computations when a lower accuracy is acceptable. An unnecessary large number will give slower computations without higher accuracy.

The user who chooses to set this parameter should be aware of how many phase function moments are needed for different phase functions for desired accuracy, since this number is strongly varying for different types of phase functions and asymmetry factors.

The δ-N method combined with the TMS/IMS methods gives maintained accuracy for cases with strongly forward-peaked phase functions for considerably lower number of channels than would have been the case without these features. This of course gives rise to a significant performance improvement. See [2] for a detailed description of the δ-N method and the intensity correction procedures.

5.2. Breaking the Azimuthal Loop

The DORT2002 calculations are always made using highest possible accuracy for any given input parameters if the user states nothing else. This means that the maximum number of Fourier components of the intensity will be calculated at all times, limited only by the number of channels specified by the user. These calculations can be very time consuming, and often unnecessary, since the intensity often converges far earlier. Therefore, the user has the opportunity to specify a maximum relative error allowed for the intensity as a convergence criterion for the azimuthal loop over Fourier components, i.e. the calculations will continue until the convergence criterion is met or until the maximum possible precision is reached (dependent of the specified number of channels). Since the azimuthal loop is the outermost loop in the calculations, much is gained if it can be terminated earlier.

The method is based on an ad hoc assumption that the Fourier components of the intensity are approximately exponentially decreasing. It is an engineering method, based on experimental numerical studies of the Fourier components, that makes DORT2002 far more efficient. The method does not stand on a solid scientific ground. The algorithm has been carefully tested under a wide range of conditions and has proven to be valid for all tested single-layer cases. However, there is no guarantee that it will be valid for all possible single-layer simulations. There is no way to assure that the assumption of exponentially decreasing terms is valid or even reasonable in the infinity of combinations of optical properties that are possible to create. Some multi-layer cases give bad results. The user should be aware of this, and it is also possible to turn off this feature if there is any doubt that it yields bad results in a particular case.

6. Acknowledgements

This report was supported by T2F (“TryckTeknisk Forskning, a Swedish printing research program), which is gratefully acknowledged.

Appendix A. The DORT2002 Parameter Struct

The following tables show the fields in the DORT2002 parameter struct by category.

	Fieldname	Description	Range
Upper Boundary Conditions	I_in	Diffuse intensity	Vector of positive reals, size N×1
	I_0b	Beam intensity	Positive real
	mu_0	Beam polar angle cosine	Real in (0,1]
	phi_0	Beam azimuthal angle	Real in [0,2π]
	top_depth	Depth at upper boundary	Positive real
Lower Boundary Conditions	TSR_type	Bottom surface type	1 = Diffuse surface *, 2 = No surface
	rho_n	Bottom surface reflectance	Real in [0,1] *
	bottom_depth	Depth at lower boundary	Positive real ≥ top_depth
Layer Properties	N_layers	Number of layers	Positive integer
	sigma_s	Scattering coefficient(s)	Vector of non-negative reals, size 1×N_layers
	sigma_a	Absorption coefficient(s)	Vector of positive reals, size 1×N_layers
	phase_type	Phase function type	Vector of 1 or 2, size 1× N_layers 1 = Henyey-Greenstein 2 = User defined
	phase_moments	User defined phase moments	N_layers × 1 cell array of vectors, size 1×N, and/or reals **
	N_phase_moments	Number of phase moments used in intensity correction procedures	Vector of positive integers in [2N+1,600], size 1×N_layers
	phase_moments_check	Automatic check for how many phase moments to be used, for maximum accuracy.	0, No phase moments check 1, Phase moments check
	layer_thickness	Sub-layer thickness	Vector of positive reals, size 1×N_layers

Table 1. Parameters for characterizing the simulated medium

* If field TSR_type is set to 1 user must define the surface reflectance rho_n.

** Setting parameter field phase_type to user defined for a certain layer allows the user to define other phase functions than Henyey-Greenstein, more suitable for the case at hand. If so, the phase function moments have to be specified. If Henyey-Greenstein phase function is to be used, phase_type should be set to Henyey-Greenstein for the specific layer and the asymmetry factor specified in phase_moments.

	Fieldname	**Description**	**Range**
Calculation Settings	`N`	Half number of channels	Positive integer
	`N_depths`	Number of calculation depths	Positive integer
	`depth`	Output depth(s)	Vector of reals in [`top_depth`, `bottom_depth`], size `N_depths` x 1
	`N_interpol`	Number of output polar angle cosines	Positive integer
	`mu_interpol`	Output polar angle cosine(s)	Vector of reals in (0,1], size `N_interpol`×1.
	`phi_interpol`	Output azimuthal angle(s)	Vector of reals in [0,2π].
	`azimuthal_break ***`	Use azimuthal break algorithm	0: Do not use 1: Use azimuthal break algorithm
	`azerr ***`	Convergence criterion: Max relative error in intensity calculations	Positive real in [0,0.2].
	`parameter_check`	Internal parameter check in DORT2002	0 = do not check parameters 1 = check parameters

Table 2. Parameters for controlling the calculation settings

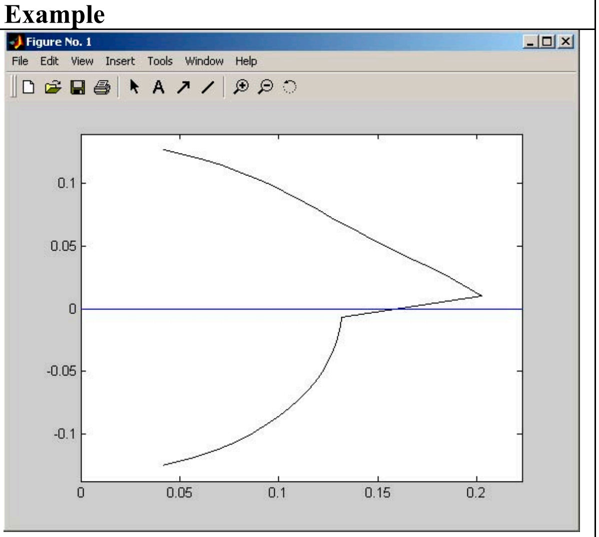
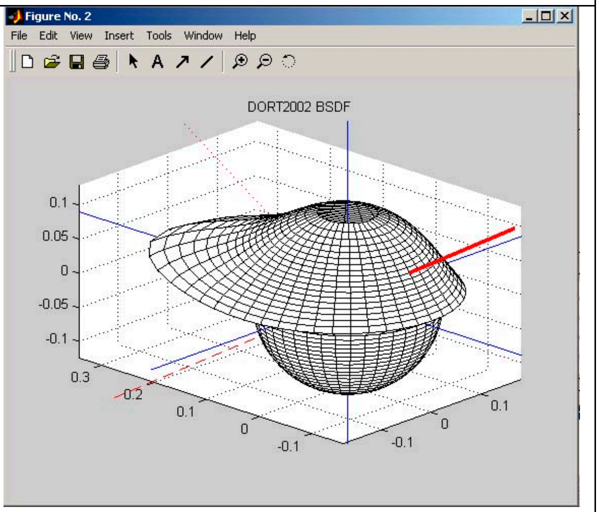
*** Should be used with caution. See section 5.2.

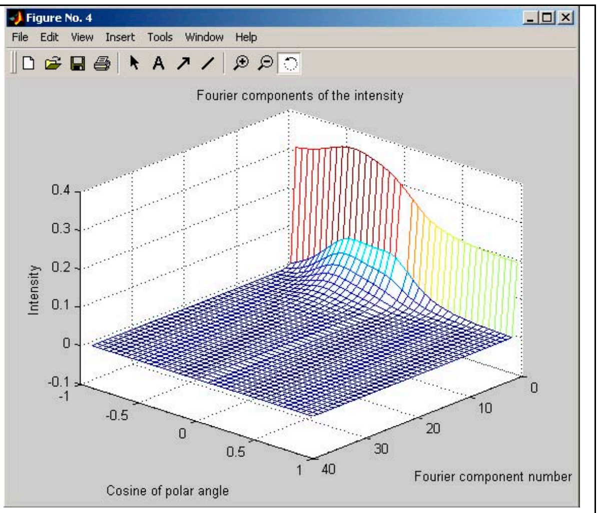
	Fieldname	**Description**
Output options 0 = no calculation 1 = calculation	`results_total_reflectance`	Total reflectance
	`results_total_transmittance`	Total transmittance
	`results_total_absorptance`	Total absorptance
	`results_ABSDF`	Azimuthally averaged BSDF
	`results_BSDF`	BSDF
	`results_I_fc`	Fourier component distribution of the intensity at desired depth(s)
	`results_I_ac`	Azimuthal distribution of the intensity at desired depth(s)
	`results_beam`	Remainder of the incident beam at desired depth(s)
Plot options 0 = no plot 1 = plot **Figure numbers** Range: Vector of positive integers, size N_depths×1.	`graphs_ABSDF`	Plot of azimuthally averaged BSDF (2D curve)
	`graphs_ABTDF_fig`	Figure number(s)
	`graphs_BSDF`	BSDF plot (3D plot)
	`graphs_BSDF_fig`	Figure number(s)
	`graphs_I_fc_mesh`	Fourier components of the intensity visualized in mesh plot.
	`graphs_I_fc_mesh_fig`	Figure number(s)
	`graphs_I_ac_mesh`	Azimuthal components of the intensity visualized in mesh plot.
	`graphs_I_ac_mesh_fig`	Figure number(s)
	`graphs_I_ac_3d`	Azimuthal components of the intensity visualized in 3D plot.
	`graphs_I_ac_3d_fig`	Figure number(s)
Save options	`simulation_name`	Name of simulation
	`save_results`	0 = do not save 1 = save output in .MAT file
	`result_file`	Name of .MAT file
	`save_results_text`	0 = do not save 1 = save output in .TXT file
	`result_file_text`	Name of .TXT file

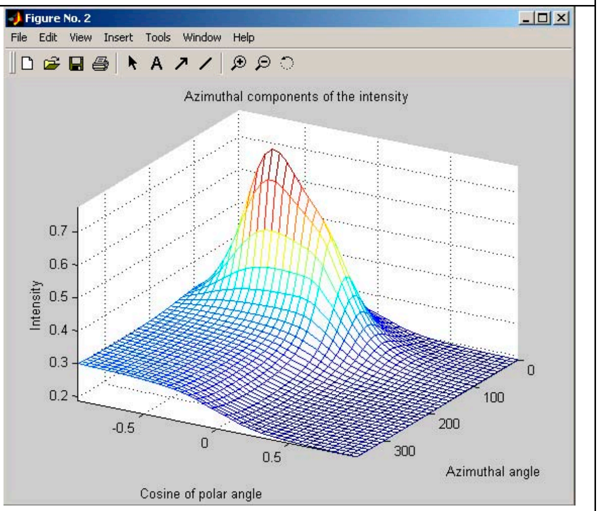
Table 3. Parameters for controlling the output

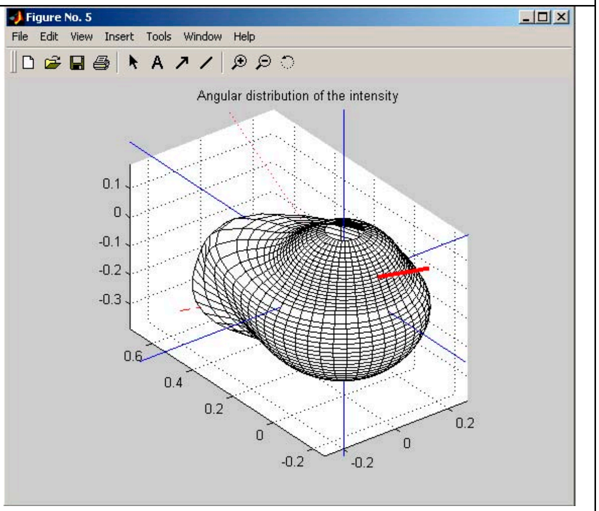
Appendix B. The DORT2002 Plot Options

The following table describes the plot options.

Plot Option	Description	Example
Mean BSDF, curve	2D plot of the azimuthally averaged Bidirectional Scattering Distribution Function (BSDF). The upper curve is the azimuthally averaged BRDF and the lower curve is the azimuthally averaged BTDF.	 Figure No. 1 File Edit View Insert Tools Window Help 0.1 0.05 0 -0.05 -0.1 0 0.05 0.1 0.15 0.2
BTDF, 3D	3D Plot of the Bidirectional Scattering Distribution Function (BSDF). The distribution plotted in the positive hemisphere is the BRDF and the distribution plotted in the negative hemisphere is the BTDF.	 Figure No. 2 File Edit View Insert Tools Window Help DORT2002 BSDF 0.1 0.05 0 -0.05 -0.1 0.3 0.2 0.1 0 -0.1 0.1 0 -0.1

<table>
<tr>
<td><em>I, Fourier mesh</em></td>
<td>The Fourier component distribution of the intensity as a surface plot.</td>
<td>


</td>
</tr>
<tr>
<td><em>I, azimuthal mesh</em></td>
<td>Surface plot of the azimuthal components of the intensity, i.e. the intensity at the user defined polar angle cosines and azimuthal angles.</td>
<td>


</td>
</tr>
<tr>
<td><em>I, azimuthal 3D</em></td>
<td>3D plot of the angular distribution of the intensity in Cartesian coordinates.</td>
<td>


</td>
</tr>
</table>

Table 4. Description of the plot options

Appendix C. The DORT2002 Result Struct

The following table describes the fields of the result struct.

	Fieldname	**Description**
Output Variables	`total_reflectance`	The fraction of the incident radiation reflected from the medium.
	`total_transmittance`	The fraction of the incident radiation transmitted through the medium.
	`total_absorptance`	The fraction of the incident radiation absorbed by the medium.
	`ABSDF`	Azimuthally averaged BSDF.
	`BSDF`	Bidirectional Scattering Distribution Function.
	`I_fc_{depth_index}`	The Fourier components of the intensity at the user defined polar angle cosines at depths defined by `depth(depth_index)`.
	`I_ac_{depth_index}`	The azimuthal components of the intensity at the user defined polar angle cosines and azimuthal angles at depths defined by `depth(depth_index)`.
	`beam_{depth_index}`	Remainder of the incident beam at depths defined by `depth(depth_index)`.

Table 5. The fields in the result struct

References

[1] P. Edström, *Fast and Stable Solution Method for Angle-Resolved Light Scattering Simulation*, FSCN Report – ISSN 1650-5387 2002:12. (Internal FSCN Report Number: R-02-35), Mid Sweden University, 2002

[2] P. Edström and M. Lehto, *Fast and Stable Solution Method for Angle-Resolved Light Scattering Simulation II – Model Enhancements*, FSCN Report – ISSN 1650-5387 2003:17. (Internal FSCN Report Number: R-03-42), Mid Sweden University, 2003