DIPOG-2.0

User Guide
Direct Problems for Optical Gratings over Triangular Grids

A. Rathsfeld

Weierstraß-Institut für
Angewandte Analysis und Stochastik

Mohrenstr. 39
D-10117 Berlin
Germany

rathsfeld@wias-berlin.de

WIAS
Weierstrass Institute for Applied Analysis and Stochastics

Abstract[1]

This is the description of how to use the programs FEM, GFEM, FEM_CHECK, GFEM_CHECK, FEM_PLOT, GFEM_PLOT, FEM_FULLINFO, and GFEM_FULLINFO of the package DIPOG-2.0. The package is a collection of finite element (FEM) programs to determine the efficiencies of the diffraction of light by a periodic grating structure. It is based on the software package PDELIB and solves the classical case of TE and TM polarization and the case of conical diffraction. The code provides a conventional FEM and a generalized FEM (called GFEM). The latter is the variational approach of the conventional FEM combined with a new trial space. We note that the DIPOG-2.0 programs require the installation of the previous version DIPOG-1.3 or DIPOG-1.4, of the grid generator TRIANGLE-1.4, and of the equations solver PARDISO[2]. Additionally, some of them need the graphical package openGL (or the MESA emulation of openGL) together with GLTOOLS-2.4 or, alternatively, the package GNUPLOT. Examples of data and output files are enclosed.

[1] Don't read the complete user guide. Don't you have anything better to do? To use DIPOG-2.0, go to the directory DIPOG-2.0/CLASSICAL or DIPOG-2.0/CONICAL. Read the data file "example.dat". Change it according to your requirements. Run one of the executables with "example.dat" as argument. Read the results. If you still have a question, come back to this user guide and read the corresponding part, only. Good luck!

[2] PARDISO itself requires some routines from LAPACK and some BLAS routines.

Contents

1 Introductory Remarks and the Structure of the Package

1.1 What is DIPOG-2.0 and Dipog-1.4?

DIPOG-2.0 is a finite element (Fem) program to determine the efficiencies of the diffraction of light by a periodic grating structure. The unbounded domain is treated by coupling with boundary element methods. DIPOG-2.0 solves the classical Te and Tm cases, i.e. the cases of incident light in the plane perpendicular to the grooves of the periodic grating, and the case of conical diffraction, i.e. of oblique incidence of light. The code is based on the package PDELIB which is a collection of software components to create simulators based on partial differential equations:

http://www.wias-berlin.de/software/pdelib

DIPOG-2.0 provides a conventional Fem approach as well as a generalized Fem version called Gfem. The latter is nothing else than the variational approach of the conventional Fem combined with a new trial space for the approximation of the unknown solution. To compute follow the subsequent instructions.

The earlier version Dipog-1.4 does the same for the special case of binary (lamellar) gratings[3], i.e. if the different grating material pieces are of rectangular shape with sides parallel to the axes. Whenever the user is confronted with binary grating geometries, he can use Dipog-1.4 or DIPOG-2.0. However, he should prefer the more efficient Dipog-1.4. The fast generalized Fem used in Dipog-1.4 cannot be applied to general polygonal geometries.

Additionally to the computation of efficiencies, Dipog-1.4 computes optimal binary gratings for given efficiency sequences or for prescribed energy restrictions. The implementation of optimal design for DIPOG-2.0 is still in progress. For Dipog-1.4, we refer to the German Benutzer-Handbuch:

http://www.wias-berlin.de/software/DIPOG

1.2 Programming language and used packages

All programs are written in fortran or c language and based on the UNIX system. The programs require the previous version DIPOG-1.3 or DIPOG-1.4, the grid generator TRIANGLE-1.4, and the linear equations solver PARDISO together with some LAPACK and BLAS routines. For good visualization the package openGL (or at least the MESA emulation of openGL) is needed together with the auxiliary package GLTOOLS. A minor visualization is possible with the program package GNUPLOT. In emergency case, the computations run also without any visualization, i.e. without openGL and GNUPLOT. To get informations on the necessary packages, we refer to:

DIPOG:	schmidt@wias-berlin.de, rathsfeld@wias-berlin.de
TRIANGLE:	http://www.cs.cmu.edu/∼quake/triangle.html
GLTOOLS:	http://www.wias-berlin.de/software/gltools
GNUPLOT:	http://www.gnuplot.info

[3] Contrary to the original meaning of binary, several layers are admitted.

PARDISO: Olaf.Schenk@unibas.ch
O.Schenk, K.Gärtner, W.Fichtner: *Efficient Sparse LU Factorization with Left-Right Looking Strategy on Shared Memory Multiprocessors*, BIT, Vol.40, 158-176, 2000
O.Schenk, K.Gärtner: *Solving unsymmetric sparse systems of linear equations with* PARDISO, to appear in Journal of Future Generation Computer Systems.

1.3 Get executables, comment lines in input files

If the executable programs do not exist, then generate them by using the file "makefile" located in the DIPOG-2.0 home directory. To do so, produce a header file MHEAD in the subdirectory MAKES of the DIPOG-2.0 home directory. Just copy one of the example files MHEAD_SGI, MHEAD_LINUX, or MHEAD_DEC to MHEAD and change the operating system, the paths, and the flags according to your computer system.[4] Then go to the home directory

```
cd DIPOG-2.0
```

and add the commands:

```
make clean
make
```

If the package is installed, then the programs can be used by different users simultaneously. To this end, each user should have a private DIPOG-2.0 home directory containing the first four subdirectories. These subdirectories together with all data and example files and with the correct links to the executables can be created automatically in the chosen new private home directory by calling the executable MAKEHOME. Note that MAKEHOME has just been created by "make" in the subdirectory MAKES of the installed package. Before a user runs the executables, he has to set the environment variable LD_LIBRARY_PATH such that the directory containing PARDISO is included. Setting the environment variable OMP_NUM_THREADS to a non-negative integer, he limits the number of used CPUs. The solver routine PARDISO runs parallel.

Most of the subsequent executables can be called without argument. Then one gets information on the necessary arguments. Usually, all input files contain a lot of explanations and informations. Indeed, each line beginning with the sign "#" is a comment. Such lines can be added or deleted without any problem.

1.4 Structure of the package

The directory containing the README.txt file is the home directory of DIPOG-2.0. We suppose in this user guide that it is named DIPOG-2.0. There exist the following important subdirectories:

[4] The temporary directory is defined in MHEAD before the installation. Its name will be stored in the first line of the file MAKES/make_info and can be changed in this file at any time. Alternatively, the temporary working directory can be chosen by setting the environment variable TMPDIR.

`GEOMETRIES` → input files “name.inp”
(geometrical data of the gratings),
executable `SHOW`
(to visualize the input data “name.inp”, exists only with `openGL`, argument: “name.inp”)
executable `GEN_INPUT`
(to generate a general input file, no argument)
executable `GEN_ECHELLEA`
(to generate an input file for an echelle grating of type A,
first argument: name without tag “inp” of file to be created,
second argument: letter A,L,R,
third argument: depth/angle,
fourth argument: width of first part of layer,
fifth argument: width of second part of layer)
executable `GEN_ECHELLEB`
(to generate an input file for an echelle grating of type B,
first argument: name without tag “inp” of file to be created,
second argument: angle, third argument: width of layer)
executable `GEN_TRAPEZOID`
(to generate an input file for a trapezoidal grating, first argument: name without tag “inp” of file to be created,
second argument: angle, third argument: length of basis,
fourth argument: number of material layers in trapezoid,
next arguments: heights of material layers, last argument: height of coating layer)
executable `GEN_LAMELLAR`
(to generate an input file for a lamellar grating, first argument: name without tag “inp” of file to be created,
second argument: name of input file “lamellar.inp” containing location and widths of layers)
file lamellar.inp
(to define location and widths of layers in grating generated by `GEN_LAMELLAR`)
executable `GEN_POLYGON`
(to generate an input file for a grating with polygonal profile curve, first argument: name without tag “inp” of file to be created, second argument: name “file1” of file with nodes of polygon)
executable `GEN_POLYGON2`
(to generate an input file for a grating with polygonal profile curve and with coating, first argument: name without tag “inp” of file to be created, second argument: name “file1” of file with nodes of polygon, third argument: name “file2” of file with nodes of boundary of coated layer)

file file1
(to define profile line for a polygonal grating generated by
GEN_POLYGON or GEN_POLYGON2)
file file2
(to define polygonal boundary line for the coated layer
of polygonal grating generated by GEN_POLYGON2)
executable GEN_PROFILE
(to generate an input file for a profile grating given
by c-code, first argument: name without tag "inp" of file
to be created, second argument: stepsize of polygonal
approximation)
c-code file profile.c
(to define profile line for a profile grating of GEN_PROFILE)
executable GEN_PROFILES
(to generate an input file for a grating given by many
profile lines defined by c-code, first argument: name without
tag "inp" of file to be created, second argument: stepsize of
polygonal approximation)
c-code file profiles.c
(to define profile lines for a profile grating of GEN_PROFILES)
executable GEN_PIN
(to generate an input file for a pin grating given by a
profile line defined by c-code, first argument:
name of input file to be created, second argument: stepsize of
polygonal approximation)
c-code file pin.c
(to define the profile line for a pin grating of GEN_PIN)
executable GEN_CPIN
(to generate an input file for a coated pin grating given by
profile lines defined by c-code, first argument:
name of input file to be created, second argument:
stepsize of polygonal approximation)
c-code file cpin.c
(to define profile lines for a coated pin grating of GEN_CPIN)
executable GEN_CPIN2
(to generate an input file for a coated
pin grating of type 2 given by
profile lines defined by c-code, first argument:
name of input file to be created, second argument:
stepsize of polygonal approximation)
c-code file cpin2.c
(to define profile lines for a coated pin grating of GEN_CPIN2)

CLASSICAL → input files "name.dat"
(non-geometrical data of the gratings),
data file "generalized.Dat"

(data for the GFEM),
executables FEM and GFEM
(for simple calculation, case of classical diffraction,
argument “name.dat”),
executables FEM_CHECK, GFEM_CHECK
(for check of input, exists only with openGL,
argument “name.dat”),
executables FEM_PLOT and GFEM_PLOT
(for calculation with plots of resulting fields, case of classical
diffraction, exists only with openGL or GNUPLOT,
argument “name.dat”),
executables FEM_FULLINFO and GFEM_FULLINFO
(for calculation with additional information, case of classical
diffraction, argument “name.dat”)

CONICAL → input files “name.dat”
(non-geometrical data of the gratings),
data file “conical.Dat”
(data for the GFEM),
executables FEM and GFEM
(for simple calculation, case of conical diffraction,
argument “name.dat”),
executables FEM_CHECK, GFEM_CHECK
(for check of input, case of conical diffraction,
exists only with openGL, argument “name.dat”),
executables FEM_PLOT and GFEM_PLOT
(for calculation with plots of resulting fields, case of conical
diffraction, exists only with openGL or GNUPLOT,
argument “name.dat”),
executables FEM_FULLINFO and GFEM_FULLINFO
(for calculation with additional information, case of conical
diffraction, argument “name.dat”)

RESULTS → result files “name.res” and “name.erg”
(produced by executables in CLASSICAL and CONICAL)
executable PLOT_DISPLAY
(produces two-dimensional graph of data on the screen,
argument “name.res” and indices of modes the efficiencies
of which are to be plotted)
executable PLOT_PS
(produces ps file of two-dimensional graph of data,
argument “name.res” and indices of modes the efficiencies
of which are to be plotted)

MAKES → header MHEAD
(note that MHEAD is to be adapted to your computer
system before installation),
executable MAKEHOME

(for another user: produces new version of four subdirectories
`GEOMETRIES`, `CLASSICAL`, `CONICAL`, and `RESULTS` together
with all data and example files and links to executables)
body of makefile "makefile_all" and more

There exist subdirectories with technical files:

grid_tri → programs and input files for grating and grid data
dpogtr → programs and input file for the FEM computation in the classical case
gdpogtr → programs and input file for the GFEM computation in the classical case
conical → programs and input file for the FEM computation in the conical case
conical2 → programs and input files for the GFEM computation in the conical case
results → plot programs

Possibly, there exist subdirectories to install necessary packages[5]:

dipog-1.3 → necessary source files from previous version of `DIPOG`-1.3,
in dipog-1.3/gsl/src: files to install libhur.a
gltools_tar → necessary source files for Fuhrmann's package `GLTOOLS`-2.4,
this produces subdirectory gltools-2-4 during installation
triangle[6] → necessary source files for Shewchuk's package `TRIANGLE`-1.4

In case of a simultaneous use of the package, each user has its own home directory containing the four subdirectories `GEOMETRIES`, `CLASSICAL`, `CONICAL`, and `RESULTS`. The subdirectories contain the same example and data files as described above for the directories of the package. However, the executables are replaced by symbolic links to the executables of the package.

2 Diffraction Problems for Gratings

2.1 The classical TE problem

Consider an ideal optical grating (cf. the cross section in Figure 1). We choose the coordinate system such that the z-axis shows in the direction of the grooves and that the y-axis is orthogonal to the plane of the grooves. The width of the grooves in x direction is the periodicity d of the grating. The refractive index of the cover material is $\mathbf{n}^+$, that of the substrate under the grating surface structure $\mathbf{n}^-$. The grating part consists of several materials with indices $\mathbf{n}_i$. Above and below the grating structure there may exist some coated layers with different refractive index (cf. the indices $\mathbf{n}^{uc}$ and $\mathbf{n}^{lc}$ for one upper and one lower coating layer in Figure 1). We suppose that a plane wave is incident from above

[5] If needed, change to the subdirectories and follow the instructions of the corresponding files `README.txt`. If not needed, delete the subdirectories.

[6] This subdirectory exists only during internal installation.

with a direction located in the $x-y$ plane (i.e. in the plane perpendicular to the grooves) and under the incident angle θ. The wave length of the light in air is λ and we consider the case of TE polarization where the electric field vector is parallel to the grooves, i.e. it shows in the z direction. Hence, if μ_0 is the magnetic permeability of vacuum and c the speed of light, then the transverse z coordinate of the electric field is given as

$$\begin{aligned} \mathcal{E}_z^{incident}(x,y,z,t) &= E_z^{incident}(x,y,z)\,\exp(-\mathbf{i}\omega t)\,, \qquad \omega \;=\; \frac{2\pi c}{\lambda}\,, \qquad (2.1)\\ E_z^{incident}(x,y,z) &= \frac{1}{\sqrt{\mathbf{n}^+}}\exp\big(\mathbf{i}k^+\sin\theta\,x-\mathbf{i}k^+\cos\theta\,y\big)\,, \quad k^+ \;=\; \omega\sqrt{\mu_0\varepsilon_0}\,\mathbf{n}^+. \end{aligned}$$

The light is diffracted by the grating structure. Beside some evanescent part the diffracted light splits into a finite number of reflected and transmitted TE polarized plane wave modes, the propagation directions of which are independent of the grating geometry and the grating materials. The problem is to determine the amplitude and the phase of the reflected and transmitted modes. Note that the normalization factor $1/\sqrt{\mathbf{n}^+}$ in the second line of (2.1) has been introduced to obtain an incident light wave with a fixed intensity (length of Poynting vector) independent of the cover material. If the cover material is air, then $\mathbf{n}^+=1$ and the wave is normalized such that the amplitude of the electric field vector is of unit length. Of course the values of the efficiencies and phase shifts of the reflected and transmitted plane wave modes are independent of this normalization factor. The subsequent Rayleigh coefficients, however, depend on this scaling.

Using Maxwell's equations, it can be shown that the transverse component E_z satisfies the scalar Helmholtz equation $\{\triangle+k^2\}E_z=0$ in any domain of the cross section plane with constant material as well as some transmission conditions on the interfaces between materials of different refractive indices. The wave number k is equal to ω/c times the refractive index of the material. Thus we can determine E_z by the standard method for elliptic differential equations by the FEM. Using the periodicity of the problem and standard coupling techniques with the boundary element method, the domain of numerical computation can be reduced to a rectangle Ω (cf. Figure 1). This covers one period of the grating and is bounded by the horizontal lines $\Gamma^\pm$ located inside the last upper and first lower coating layer (counted from above to below) resp. in the cover material and substrate material for gratings without coatings.[7]

On Γ^+ and above Γ^+ resp. on Γ^- and below Γ^- the component E_z admits an expansion into the Rayleigh series of the form

$$E_z(x,y) \;=\; \sum_{n=-\infty}^{\infty} A_n^+\exp\big(+\mathbf{i}\beta_n^+y\big)\,\exp\left(\mathbf{i}\alpha_n x\right)+A_0^{inc}\exp\big(-\mathbf{i}\beta_0^+y\big)\,\exp\left(\mathbf{i}\alpha x\right)\,, \qquad (2.2)$$

$$E_z(x,y) \;=\; \sum_{n=-\infty}^{\infty} A_n^-\exp\big(-\mathbf{i}\beta_n^-y\big)\,\exp\left(\mathbf{i}\alpha_n x\right)\,, \qquad (2.3)$$

$$\begin{aligned} \beta_n^\pm &= \sqrt{[k^\pm]^2-[\alpha_n]^2}\,, \quad k^\pm \;=\; \frac{\omega\mathbf{n}^\pm}{c}\,, \qquad A_0^{inc} \;=\; \frac{1}{\sqrt{\mathbf{n}^+}}\\ \alpha &= k^+\sin\theta, \qquad \alpha_n \;=\; k^+\sin\theta+\frac{2\pi}{d}n\,. \end{aligned}$$

[7] For technical reasons in the FEM code, it is important to have the same material on both sides of the boundary lines $\Gamma^\pm$.

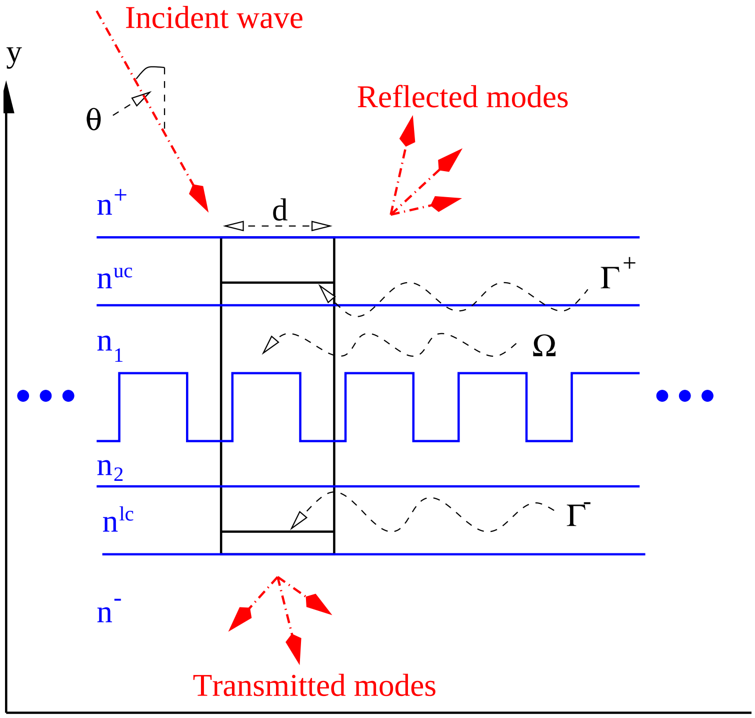

Figure 1: Cross section of grating.

Here d is the period of the grating and the complex constants $A_n^\pm$ are the so-called Rayleigh coefficients. The interesting Rayleigh coefficients are those with $n \in \mathcal{U}^\pm$,

$$\mathcal{U}^\pm = \begin{cases} \left\{ n \in \mathbb{Z} : |\alpha_n| < k^\pm \right\} & \text{if } \Im\text{m}\, k^\pm = 0 \\ \emptyset & \text{if } \Im\text{m}\, k^\pm > 0 \end{cases} .$$

Indeed, these coefficients $A_n^\pm$ describe magnitude and phase shift of the propagating plane waves. More precisely, the modulus $|A_n^\pm|$ is the amplitude of the nth reflected resp. transmitted wave mode and $\arg[A_n^\pm/|A_n^\pm|]$ the phase shift. The terms with $n \notin \mathcal{U}^\pm$ lead to evanescent waves, only. The optical efficiencies of the grating are defined by

$$e_n^\pm = \frac{\beta_n^\pm}{\beta_0^+} \frac{|A_n^\pm|^2}{|A_0^{inc}|^2}, \quad (n,\pm) \in \left\{ (n,+) : \; n \in \mathcal{U}^+ \right\} \cup \left\{ (n,-) : \; n \in \mathcal{U}^- \right\}, \tag{2.4}$$

which is the ratio of energy of the incident wave entailed to the nth propagating mode. Note that these efficiencies of propagating modes exist for non-absorbing materials, i.e. for $\Im\text{m}\, k^\pm = 0$. If the transverse component E_z has been computed approximately, then the Rayleigh coefficients can be obtained by a discretized Fourier series expansion applied to the FEM solution restricted to $\Gamma^\pm$ (cf. (2.2) and (2.3)). Formula (2.4) yields the efficiencies.

2.2 The classical TM problem

The case of TM polarization is quite similar to TE. Indeed, this time the vector of the magnetic field $\mathcal{H}$ shows in the direction of the grooves, i.e. in the direction of the z axis.

Analogously to formula (2.1) given in the last subsection for the incident electric field, we get

$$\begin{aligned}\mathcal{H}_z^{incident}(x,y,z,t) &= H_z^{incident}(x,y,z)\ \exp(-\mathbf{i}\omega t)\,, && (2.5)\\ H_z^{incident}(x,y,z) &= \frac{\sqrt{\varepsilon_0}\sqrt{\mathbf{n}^+}}{\sqrt{\mu_0}}\exp\big(\mathbf{i}k^+\sin\theta\, x - \mathbf{i}k^+\cos\theta\, y\big)\,, \quad k^+ \ = \ \omega\sqrt{\mu_0\varepsilon_0}\,\mathbf{n}^+, &&\end{aligned}$$

for the z component of the incident magnetic field $\mathcal{H}_z^{incident}$. Note that the additional factor $\sqrt{\varepsilon_0}\sqrt{\mathbf{n}^+}/\sqrt{\mu_0}$ in the definition of $H_z^{incident}$ guarantees that the incident light wave has a fixed intensity (length of Poynting vector) independent of the cover material. If the cover material is air, then the wave is normalized such that the amplitude of the electric field vector is of unit length. Like in the Te case, the values of the efficiencies and phase shifts of the reflected and transmitted plane wave modes are independent of this normalization factor. Only the subsequent Rayleigh coefficients depend on this scaling.

The z component of the complete field H_z satisfies the Helmholtz equation $\{\triangle+k^2\}H_z = 0$ in any domain of the cross section plane with constant materials. However, the transmission conditions on the interfaces are different. We can solve the transmission problem of the Helmholtz equation by Fem. Again we have a finite number of transmitted and reflected modes and the Rayleigh expansions hold for E_z replaced by H_z. More precisely, the Rayleigh coefficients are the $B_n^\pm$ of the expansions

$$\frac{\sqrt{\mu_0}}{\sqrt{\varepsilon_0}}H_z(x,y) = \sum_{n=-\infty}^{\infty} B_n^+\exp\big(+\mathbf{i}\beta_n^+ y\big)\ \exp\left(\mathbf{i}\alpha_n x\right) + B_0^{inc}\exp\big(-\mathbf{i}\beta_0^+ y\big)\ \exp\left(\mathbf{i}\alpha x\right)\ , \quad (2.6)$$

$$\frac{\sqrt{\mu_0}}{\sqrt{\varepsilon_0}}H_z(x,y) = \sum_{n=-\infty}^{\infty} B_n^-\exp\big(-\mathbf{i}\beta_n^- y\big)\ \exp\left(\mathbf{i}\alpha_n x\right)\ , \quad (2.7)$$

$$B_0^{inc} = \sqrt{\mathbf{n}^+}\ .$$

The objective is to compute the Rayleigh coefficients. They result from the Fem solution of the new transmission problems and from the discretization of the Fourier series expansion (2.6) and (2.7). The efficiencies (cf. (2.4)) are computed by

$$e_n^\pm \ = \ \frac{\beta_n^\pm}{\beta_0^+}\frac{[k^+]^2}{[k^\pm]^2}\frac{|B_n^\pm|^2}{|B_0^{inc}|^2}, \quad (n,\pm)\in\Big\{(n,+):\ n\in\mathcal{U}^+\Big\}\cup\Big\{(n,-):\ n\in\mathcal{U}^-\Big\}. (2.8)$$

Finally, we note that the case of an incident wave propagating in a direction of the $x-y$ plane together with an arbitrary polarization is the superposition of Te and Tm polarization.

2.3 Conical problems

The essential difference between the classical diffraction of the last two subsections and the conical one is that the direction of the incident light wave is oblique, i.e. it is not restricted to the $x-y$ plane. Whereas in the classical case the directions of the finitely many reflected and transmitted plane wave modes remain located in the $x-y$ plane, now they are located on a cone in the $x-y-z$ space. The Fem approach is analogous to the classical case. However, instead of a transmission problem for a scalar Helmholtz equation, Maxwell's system reduces to a coupled system of two scalar Helmholtz equations for the two transverse components E_z and H_z of the electric and magnetic field. Consequently, we have two Rayleigh expansions and two sequences of Rayleigh coefficients.

More precisely, skipping the time harmonic factor, we have an incident wave of the form

$$\begin{aligned} E^{incident}(x,y,z) &= E^{inc}\exp\Big(\mathbf{i}[\alpha x-\beta y+\gamma z]\Big)\,, \\ H^{incident}(x,y,z) &= H^{inc}\exp\Big(\mathbf{i}[\alpha x-\beta y+\gamma z]\Big)\,. \end{aligned}$$

with constant vectors E^{inc} and H^{inc} and a wave vector $\vec{k}=(\alpha,-\beta,\gamma)$ such that the wave number k^+ is the modulus of $\vec{k}$ and the direction of the incoming plane wave is $\vec{k}/k^+$. Here k^+ is the same wave number as in the classical TE and TM case. The direction $\vec{k}/k^+$ must be prescribed by the user of `DIPOG`-2.0. This can be characterized by two parameters, namely by the angles θ_i and ϕ_i which are the spherical coordinates of (α,β,γ). We emphasize that θ_i and ϕ_i are the spherical coordinates of (α,β,γ) and not those of the normalized wave vector $\vec{k}/k^+=(\alpha,-\beta,\gamma)$.[8] Contrary to this, the angles θ and ϕ of the reflected and transmitted plane wave modes are exactly the spherical coordinates of the normalized wave vectors. Unfortunately, this traditional notation is a little bit confusing.

Either we use the spherical coordinate system with the $x-y$ plane as basis plane (xy system) or the spherical coordinates based on the $x-z$ plane (xz system). In the xy system we define the direction as (cf. Figure 2):

$$D_{xy}=\Big(\sin\theta_{xy}\cos\phi_{xy}\;,\;\cos\theta_{xy}\cos\phi_{xy}\;,\;\sin\phi_{xy}\Big)$$

Here θ_{xy} is the angle of inclination of the plane, containing the direction D_{xy} and the z axis, from the $y-z$ plane. Angle ϕ_{xy} is the angle of direction D_{xy} inside this inclined plane, i.e. the angle between D_{xy} and the orthogonal projection of D_{xy} to the $x-y$ plane.

For the xz system the direction is given by (cf. Figure 3):

$$D_{xz}=\Big(\sin\theta_{xz}\cos\phi_{xz}\;,\;\cos\theta_{xz}\;,\;\sin\theta_{xz}\sin\phi_{xz}\Big)$$

Here ϕ_{xz} is the angle of inclination of the plane, containing the direction D_{xz} and the y axis, from the $x-y$ plane. Angle θ_{xz} is the angle of direction D_{xz} inside this inclined plane, i.e. the angle between D_{xz} and the orthogonal projection of D_{xz} to the $x-z$ plane. To change between the xy system and the xz system the following formulae are useful

$$\begin{aligned} \phi_{xy} &= \arcsin\Big(\sin\theta_{xz}\sin\phi_{xz}\Big)\,, \quad \theta_{xy} = \arcsin\left(\frac{\sin\theta_{xz}\cos\phi_{xz}}{\sqrt{1-\sin^2\theta_{xz}\sin^2\phi_{xz}}}\right)\,, \\ \theta_{xz} &= \arccos\Big(\cos\theta_{xy}\cos\phi_{xy}\Big)\,, \\ \phi_{xz} &= \begin{cases} +\arcsin\left(\dfrac{\sin\phi_{xy}}{\sqrt{1-\cos^2\theta_{xy}\cos^2\phi_{xy}}}\right) & \text{if } \sin\theta_{xy}\cos\phi_{xy}>0 \\ -\arcsin\left(\dfrac{\sin\phi_{xy}}{\sqrt{1-\cos^2\theta_{xy}\cos^2\phi_{xy}}}\right)-\pi & \text{else}\,. \end{cases} \end{aligned}$$

[8] In other words $\pi-\theta_i$ and ϕ_i are the spherical coordinates of $\vec{k}/k^+$, and θ_i is not the angle enclosed by $\vec{k}/k^+$ and the positive y axis but the angle enclosed by $\vec{k}/k^+$ and the negative y axis.

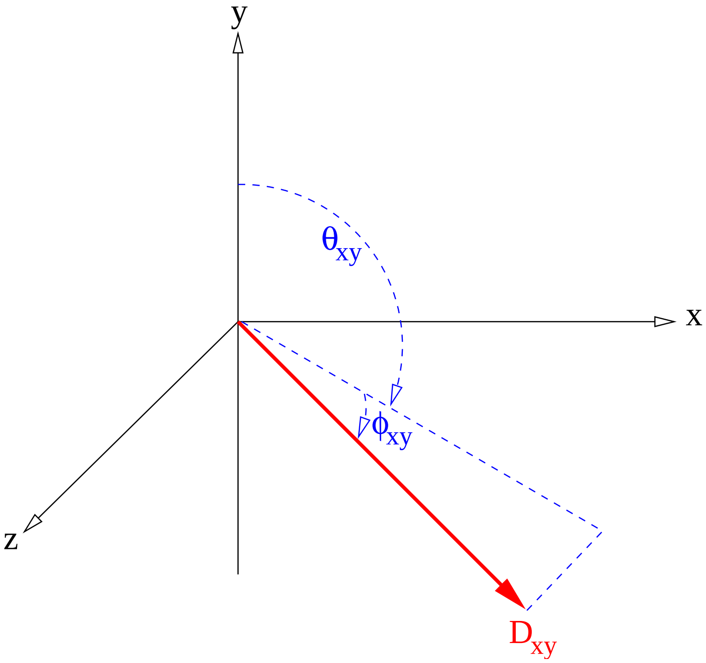

Figure 2: Coordinate system based on $x - y$ plane.

Though the user can choose his favourite spherical coordinate system for the input of the direction of incidence, the output of the directions for the reflected and transmitted modes are presented in the xz system.

Clearly, the fields E^{inc} and H^{inc} must be orthogonal. Moreover, the two vectors are uniquely determined by the normalization condition, by Maxwell's equation, and by the polarization type prescribing the polarization direction. Here the normalization condition

$$\mathbf{n}^+[E_z^{inc}]^2 + \frac{1}{\mathbf{n}^+}[B_z^{inc}]^2 \quad = \quad \frac{[k^+]^2 - \gamma^2}{[k^+]^2}, \quad B_z^{inc} \quad = \quad \frac{\sqrt{\mu_0}}{\sqrt{\varepsilon_0}} H_z^{inc}$$

means that the incident light wave has a fixed intensity (length of Poynting vector) independent of the cover material. If the cover material is air, then the wave is normalized such that the amplitude of the electric field vector is of unit length. The values of the efficiencies and phase shifts of the reflected and transmitted plane wave modes are not effected by the normalization factor. Only the subsequent Rayleigh coefficients depend on this scaling.

The polarization type must be prescribed by the user of `DIPOG`-2.0. There are three possibilities. The first is to choose TE polarization with the electric field vector E^{inc} pointing in the direction perpendicular to the wave vector (incidence direction) and to the y axis. The second is TM polarization with the magnetic field vector H^{inc} pointing in the direction perpendicular to the wave vector (incidence direction) and to the y axis. Note that the direction perpendicular to the wave vector and to the y axis is, by definition, the z axis if wave vector and y axis should be collinear. The third choice is to prescribe the angle ψ (cf. Figure 4) enclosed by the x axis and by the projection of the electric field vector E^{inc} to the $x - z$ plane.

Fixing the incident field, the resulting total field is determined and can be computed by FEM. It remains to describe the output data of `DIPOG`-2.0. The Rayleigh expansions above

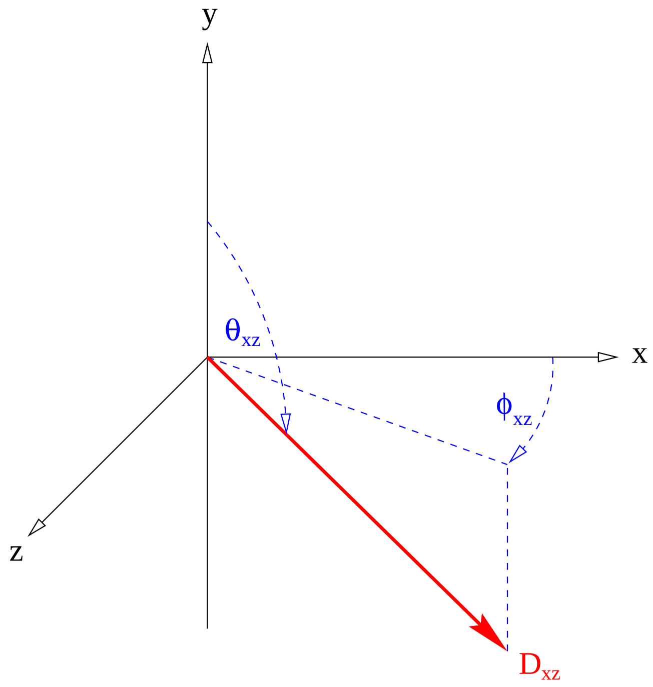

Figure 3: Coordinate system based on $x - z$ plane.

resp. below the grating take the form

$$\begin{aligned} E(x,y,z) &= E^{inc}\exp\Big(\mathbf{i}[\alpha x - \beta y + \gamma z]\Big) + \sum_{n\in\mathbb{Z}} \vec{A}_n^+ \exp\Big(\mathbf{i}[\alpha_n x + \beta_n^+ y + \gamma z]\Big)\ , \\ H(x,y,z) &= H^{inc}\exp\Big(\mathbf{i}[\alpha x - \beta y + \gamma z]\Big) + \sum_{n\in\mathbb{Z}} \vec{C}_n^+ \exp\Big(\mathbf{i}[\alpha_n x + \beta_n^+ y + \gamma z]\Big)\ , \\ & \alpha_n = \alpha + \frac{2\pi}{d}n,\ \beta_n^\pm = \sqrt{[k^\pm]^2 - [\alpha_n]^2 - \gamma^2}\ ,\ \Re\mathrm{e}\,\beta_n^\pm > 0\ ,\ \Im\mathrm{m}\,\beta_n^\pm \geq 0 \end{aligned}$$

resp.

$$\begin{aligned} E(x,y,z) &= \sum_{n\in\mathbb{Z}} \vec{A}_n^- \exp\Big(\mathbf{i}[\alpha_n x - \beta_n^- y + \gamma z]\Big)\ , \\ H(x,y,z) &= \sum_{n\in\mathbb{Z}} \vec{C}_n^- \exp\Big(\mathbf{i}[\alpha_n x - \beta_n^- y + \gamma z]\Big)\ . \end{aligned}$$

Now there are three variants of output data. The first computes the third components, i.e. the z components of the Rayleigh coefficients

$$p_n^\pm = [\vec{A}_n^\pm]_z\ ,\quad q_n^\pm = [\vec{B}_n^\pm]_z\ ,\quad \vec{B}_n^\pm = \frac{\sqrt{\mu_0}}{\sqrt{\varepsilon_0}}\vec{C}_n^\pm$$

and the efficiencies

$$\begin{aligned} e_n^+ &= \frac{\beta_n^+}{\beta}\frac{[k^+]^2}{[k^+]^2-\gamma^2}\left[\mathbf{n}^+|p_n^+|^2 + \frac{1}{\mathbf{n}^+}|q_n^+|^2\right]\ , \\ e_n^- &= \frac{\beta_n^-}{\beta}\frac{1}{[k^-]^2-\gamma^2}\left[\mathbf{n}^+[k^-]^2|p_n^-|^2 + \frac{1}{\mathbf{n}^+}[k^+]^2|q_n^-|^2\right] \end{aligned} \tag{2.9}$$

of the nth reflected resp. transmitted wave mode. The second output variant computes the TE and TM part of the total wave, i.e. if $\mathbf{s}_n^\pm$ stands for the direction perpendicular to the y

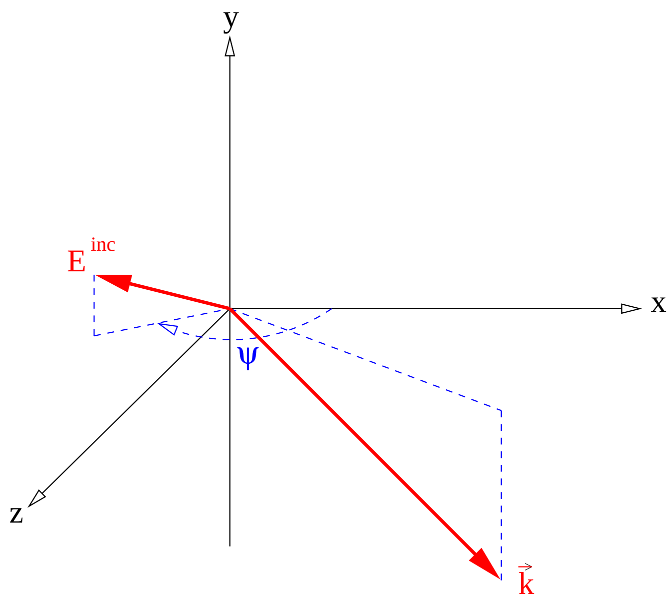

Figure 4: Coordinate system based on $x - z$ plane.

axis and to the direction of propagation of the nth reflected resp. transmitted plane wave mode ($\mathbf{s}_n^\pm = (\alpha_n, \pm\beta_n^\pm, \gamma) \times (0,1,0)/|(\alpha_n, \pm\beta_n^\pm, \gamma) \times (0,1,0)|$), then the output coefficients are the scalar products

$$\begin{aligned}\left\langle \vec{A}_n^\pm, \mathbf{s}_n^\pm \right\rangle &= \frac{ap_n^\pm + bc\, q_n^\pm/\mathbf{n}^\pm}{(1-c^2)\sqrt{a^2+c^2}}\,,\\ \left\langle \vec{B}_n^\pm, \mathbf{s}_n^\pm \right\rangle &= \frac{aq_n^\pm - bc\, p_n^\pm \mathbf{n}^\pm}{(1-c^2)\sqrt{a^2+c^2}}\,, \quad (a,b,c) \;=\; \frac{(\alpha_n, \pm\beta_n^\pm, \gamma)}{\sqrt{\alpha_n^2 + [\beta_n^\pm]^2 + \gamma^2}}\end{aligned}$$

The efficiencies of the second output are the total efficiencies $e_n^\pm$ of (2.9) and the efficiencies corresponding to the TE and TM parts

$$\frac{\beta_n^\pm}{\beta} \left| \left\langle \vec{A}_n^\pm, \mathbf{s}_n^\pm \right\rangle \right|^2 \mathbf{n}^+\,, \quad \frac{\beta_n^\pm}{\beta} \frac{\mathbf{n}^+}{[\mathbf{n}^\pm]^2} \left| \left\langle \vec{B}_n^\pm, \mathbf{s}_n^\pm \right\rangle \right|^2\,,$$

i.e. the efficiencies of the projection of the nth reflected resp. transmitted wave mode to the component with electric resp. magnetic field polarized in $\mathbf{s}_n^\pm$ direction. Finally, the third variant computes the S- and P-parts of the electric field, i.e. the components of the Jones vector. If $\mathbf{s}_n^\pm$ is defined as above and if $\mathbf{p}_n^\pm$ is the direction orthogonal to $\mathbf{s}_n^\pm$ and the direction of propagation of the nth reflected resp. transmitted plane wave mode ($\mathbf{p}_n^\pm = (\alpha_n, \pm\beta_n^\pm, \gamma) \times \mathbf{s}_n^\pm/|(\alpha_n, \pm\beta_n^\pm, \gamma) \times \mathbf{s}_n^\pm|$), then the S- and P-parts of the Rayleigh coefficients are

$$\begin{aligned}\left\langle \vec{A}_n^\pm, \mathbf{s}_n^\pm \right\rangle &= \frac{ap_n^\pm + bc\, q_n^\pm/\mathbf{n}^\pm}{(1-c^2)\sqrt{a^2+c^2}}\,,\\ \left\langle \vec{A}_n^\pm, \mathbf{p}_n^\pm \right\rangle &= -\frac{1}{\mathbf{n}^\pm} \left\langle \vec{B}_n^\pm, \mathbf{s}_n^\pm \right\rangle \;=\; \frac{bc\, p_n^\pm - a\, q_n^\pm/\mathbf{n}^\pm}{(1-c^2)\sqrt{a^2+c^2}}\,.\end{aligned}$$

The efficiencies of the third output are the total efficiencies $e_n^\pm$ of (2.9) and the efficiencies corresponding to the S- and P-parts, i.e. the efficiencies of the projection of the nth reflected resp. transmitted wave mode to the component with electric field polarized in $\mathbf{s}_n^\pm$ resp. $\mathbf{p}_n^\pm$ direction

$$\frac{\beta_n^\pm}{\beta} \left| \left\langle \vec{A}_n^\pm, \mathbf{s}_n^\pm \right\rangle \right|^2 \mathbf{n}^+ \,, \quad \frac{\beta_n^\pm}{\beta} \left| \left\langle \vec{A}_n^\pm, \mathbf{p}_n^\pm \right\rangle \right|^2 \mathbf{n}^+ \,.$$

2.4 References

For more details, see the following publications and the references therein:

- G. Bao, D.C. Dobson, and J.A. Cox: Mathematical studies in rigorous grating theory, *J. Opt. Soc. Amer. A* **12**, pp. 1029–1042 (1995).
- J. Elschner and G. Schmidt: Diffraction in periodic structures and optimal design of binary gratings I: Direct problems and gradient formulas, *Math. Meth. Appl. Sci.* **21**, pp. 1297–1342 (1998).
- J. Elschner and G. Schmidt: The numerical solution of optimal design problems for binary gratings, *J. Comput. Physics* **146**, pp. 603–626 (1998).
- J. Elschner, R. Hinder and G. Schmidt: Finite element solution of conical diffraction problems, *Adv. Comput. Math.* **16**, pp. 139–156 (2002).
- J. Elschner, R. Hinder and G. Schmidt: Direct and inverse problems for Diffractive Structures- Optimization of binary gratings, In: W. Jäger, H.J.Krebs (eds.), *Mathematics, key technology for the future: joint projects between universities and industry*, Springer Verlag Berlin Heidelberg, 2003, pp. 293–304.
- R. Petit (ed.): *Electromagnetic Theory of Gratings*, Springer, Berlin, 1980.
- H.P. Urbach: Convergence of the Galerkin method for two-dimensional electromagnetic problems, *SIAM J. Numer. Anal.* **28**, pp. 697–710 (1991).

For generalized Fem methods applied to the Helmholtz equation (modified and described in the subsequent Sections 6 and 7), we refer to:

- I. Babuška, F. Ihlenburg, E. Paik, and S. Sauter: A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution, *Comp. Methods Appl. Mech. Eng.* **128**, pp. 325–359 (1995).
- O. Cessenat and B. Depres: Application of an ultra weak variational formulation of elliptic PDEs to the two-dimensional Helmholtz problem, *SIAM J. Numer. Anal.* **35**, 255-299 (1998).
- F. Ihlenburg: *Finite element analysis of acoustic scattering*, Springer Verlag New-York Berlin Heidelberg, Applied Mathematical Sciences **132**, 1998.
- J.M. Melenk and I. Babuška: The partition of unity method: Basic theory and applications, *Comp. Methods Appl. Mech. Eng.* **139**, pp. 289–314 (1998).

3 Geometry Input

3.1 Geometrical data in input file “name.dat”

Computation starts with the change of the working directory to directory CLASSICAL (for classical diffraction) or to CONICAL (conical diffraction) and by calling an executable (e.g. FEM or GFEM) followed by the data file “name.dat” as argument of the executable. Here “name.dat” contains all information on the grating and the light.

```
cd DIPOG-2.0/CLASSICAL
FEM name.dat
```

or

```
cd DIPOG-2.0/CONICAL
GFEM name.dat
```

On the screen there will appear the output data of the computation and the name of an additional output file, where the output data is stored.

Mainly, the geometrical information of the input data in “name.dat” is fixed by the lines:

```
# Length factor of additional shift of grating geometry.
#        This is shift into the x-direction.
#        This is length of shift relative to period.
   0.

# Stretching factor for grating in y-direction:
   1.

# Length of additional shift of grating in micro meter.
#        This is shift in y-direction.
   0.

# Period of grating in micro meter:
   1.

# Grating data:
   name1
```

Here “name1” refers either to a file “name1.inp” with geometrical data located in the subdirectory GEOMETRIES or to some special code words to fix the geometry of the grating. We describe how to get the file “name1.inp” in point 3.2 and the alternative code words in the subsequent point 3.3 of this section.

As mentioned above, the computation starts in the directory “CLASSICAL” and is based upon a geometry input file “GEOMETRIES/name1.inp” indicated in “name.dat”. However, if the code is started from a directory different from “CLASSICAL” or if the geometry data file is located in a different directory with the path “path1”, then the file is to be specified by adding its path in “name.dat” as:

```
# Grating data.
   path1/name1
```

In particular, for a geometry input file in the current working directory use:

```
# Grating data.
  ./name1
```

Note that the geometry data in “name1.inp” should be given relative to the period which is specified in the data file “name.dat” of directory CLASSICAL resp. CONICAL. All data of “name1.inp” will later be multiplied by the given length of period (e.g. by 1 μm).

Additional geometrical information, fixed in file “name.dat”, concerns the coated layers. In principle, the grating part is a rectangular domain (cf. Ω in figure 1). Above and below this part we can add a few number of coated layers in form of strips parallel to the upper and lower side of the rectangle. The numbers of these layers together with the corresponding thickness is given in special lines of “name.dat”. Explanations of the lines in “name.dat” can be found directly in the neighbouring comment lines starting with symbol “#”.

3.2 How to get an input file “name1.inp”?

Change to subdirectory GEOMETRIES. Copy an existing file like e.g. “example.inp” (cf. the enclosed file in 10.1), and change its name into e.g. “name1.inp”.

```
cd DIPOG-2.0/GEOMETRIES
cp example.inp name1.inp
```

Change “name1.inp” in your editor (emacs,vi?) according to your requirements. You will find the necessary information as comments in the file “name1.inp”. Indeed, each line beginning with “#” is a comment.

To check the result, enter the command:

```
SHOW name1.inp
```

You will see a first picture (cf. left picture in Figure 5) with the chosen points of a polygonal structure. After pressing Escape or Bar/Space you see a second picture (cf. right picture in Figure 5) with a coarse triangulation and with the different regions (later distinguished by different optical indices) in different colours. Press Escape or Bar/Space to end the check.

Alternatively, to create “name1.inp”, one can call the executable GEN_INPUT from the subdirectory GEOMETRIES and work interactively. Just enter the command:

```
GEN_INPUT
```

This program prompts you for everything needed. Nevertheless, we recommend the first way of copying and modifying an existing file.

As mentioned above, special gratings like echelle gratings, lamellar, trapezoidal, and simple profile gratings need not to be generated by an input file “name1.inp”. Special code words will generate automatically hidden files of this type. However, in some situations the user might wish to change the automatically generated “name1.inp” files. He might wish to add small modifications to the geometry, or he wants to change the meshsize. To do this the user can create the otherwise hidden “name1.inp” files explicitly by the following executables.

If an input file for an echelle grating of type A is needed (right-angled triangle with

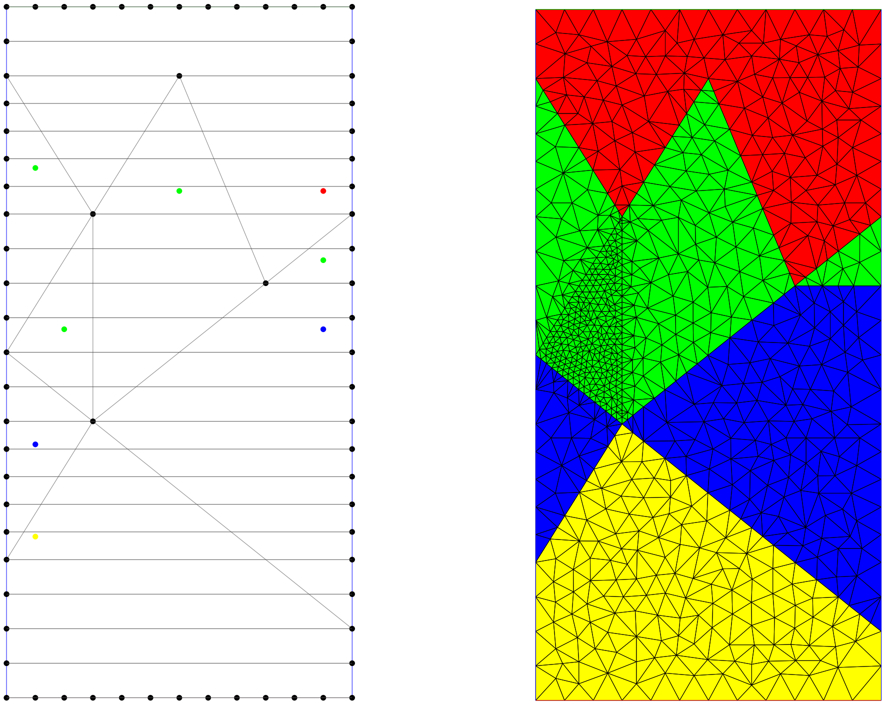

Figure 5: Pictures of grid produced by SHOW.

hypotenuse parallel to the direction of the periodicity, cf. Figure 6), then this can be accomplished by calling the executable GEN_ECHELLEA from the subdirectory GEOMETRIES. More precisely, the command

GEN_ECHELLEA name **R** 0.3 0.03 0.04

creates the file "name.inp" of the desired echelle profile grating, with right blaze angle greater than 45°, with a depth (triangle height) of 0.3 times period of the grating and with coated layers of height 0.03 resp. 0.04 times period over the first resp. second part of the grating (measured in direction perpendicular to the echelle profile, height greater or equal to zero). If the input letter **R** is replaced by an **L**, then the left blaze angle is greater than 45 degrees. Moreover, if the input letter **R** is replaced by an **A** and the following input number 0.3 by 60., then the left blaze angle is 60°.

If an input file for an echelle grating of type B is needed (right-angled triangle with one of the legs parallel to the direction of the periodicity, cf. Figure 7), then this can be accomplished by calling the executable GEN_ECHELLEB from the subdirectory GEOMETRIES. More precisely, the command

GEN_ECHELLEB name 60. 0.05

creates the file "name.inp" of the desired echelle profile grating, with angle 60° (angle enclosed by hypotenuse and by the leg parallel to the period) and with a coated layer of height 0.05 times period of the grating (measured in direction perpendicular to echelle profile, height greater or equal to zero).

If an input file for a trapezoidal grating is needed (isosceles trapezoid with the basis parallel to the direction of the periodicity, cf. Figure 8), then this can be accomplished by calling the executable GEN_TRAPEZOID from the subdirectory GEOMETRIES. More precisely, the command

```
GEN_TRAPEZOID name 60. 0.6 3 0.2 0.1 0.1 0.05
```

creates the file “name.inp” of the desired trapezoidal profile grating, with angle 60° (angle enclosed by basis and the sides) with a basis of length 0.6 times period of the grating consisting of 3 material layers of heights 0.2 times period, 0.1 times period, and 0.1 times period, respectively, and with a coated layer of height 0.05 times period (greater or equal to zero).

If an input file for a lamellar grating is needed (rectangular grating consisting of several materials placed in rectangular subdomains, cf. Figure 9), then this can be accomplished by calling the executable GEN_LAMELLAR from the subdirectory GEOMETRIES. More precisely, if the file “GEOMETRIES/lamellar.inp” contains the numbers (each number in a separate line) 3, 4, 0.2, 0.6, -0.2, 1.0, 0., 0.5, 0.70, .0, 0.50, 0.900, .00, 0.500, and 0.900, then the command

```
GEN_LAMELLAR name lamellar.inp
```

creates the file “name.inp” of the desired lamellar profile grating, with 3 columns each divided into 4 rectangular layers, first column with x coordinate in $0 < x < 0.2$, second column with $0.2 < x < 0.6$, third column with $0.6 < x < 1$ (all coordinates are normalized with respect to the period: period corresponds to x=1), whole grating with y coordinate s.t. $-0.2 < y < 1.0$, first column: first layer with $-0.2 < y < 0.$, second with $0. < y < 0.5$, third with $0.5 < y < 0.7$ and fourth with $0.7 < y < 1.$, second column: first layer with $-0.2 < y < 0.0$, second with $0.0 < y < 0.50$, third with $0.50 < y < 0.90$ and fourth with $0.90 < y < 1.$, third column: first layer with $-0.2 < y < 0.00$, second with $0.00 < y < 0.500$, third with $0.500 < y < 0.900$ and fourth with $0.900 < y < 1.$.

If an input file for a simple layer grating is needed, then this can be accomplished by calling the executable GEN_LAMELLAR from the subdirectory GEOMETRIES. More precisely, if the file “GEOMETRIES/lamellar.inp” contains the numbers (each number in a separate line) 1, 1, 0.2, and 0.8, then the command

```
GEN_LAMELLAR name lamellar.inp
```

creates the file “name.inp” of the desired layer grating, with layer material s.t. the y coordinate satisfies $0.2 < y < 0.8$ (all coordinates are normalized with respect to the period: period corresponds to x=1).

If an input file for a grating with a polygonal profile line is needed (cf. Figure 10), then this can be accomplished by calling the executable GEN_POLYGON from the subdirectory GEOMETRIES. More precisely, if the file “GEOMETRIES/file1” contains the corner points of a polygonal profile line (in “GEOMETRIES/file1”: in each line beginning without ‘#’ there should be the x- and y-coordinate of one of the consecutive corner points, first point with x-coordinate 0, last point with x-coordinate 1, same y-coordinate for first and last point, all x-coordinates between 0 and 1, at least two different y-coordinates, last line should be

"End"), then the command

```
GEN_POLYGON name file1
```

creates the file "name.inp" of the desired polygonal grating.

If an input file for a grating determined by two polygonal profile lines is needed (cf. Figure 11), then this can be accomplished by calling the executable `GEN_POLYGON2` from the subdirectory `GEOMETRIES`. More precisely, if the file "`GEOMETRIES`/file1" contains the corner points of a polygonal profile line (in "`GEOMETRIES`/file1": in each line beginning without '#' there should be the x- and y-coordinate of one of the consecutive corner points, first point with x-coordinate 0, last point with x-coordinate 1, same y-coordinate for first and last point, all x-coordinates between 0 and 1, at least two different y-coordinates, last line should be "End") and if the file "`GEOMETRIES`/file2" contains the corner points of a second polygonal profile line (in "`GEOMETRIES`/file2": in each line beginning without '#' there should be the x- and y-coordinate of one of the consecutive corner points, first and last point must be corner of first polygon, second polygon must be on left-hand side of first, one to one correspondence of the corners on the two polygons between first and last point of second polygon[9], quadrilateral domain between corresponding segments on the left of first polygon, these quadrilaterals must be disjoint, last line should be "End"), then the command

```
GEN_POLYGON2 name file1 file2
```

creates the file "name.inp" of the desired polygonal grating.

If an input file for a grating determined by profile line given as $\{(f_x(t), f_y(t)) :\ 0 \le t \le 1\}$ is needed, then this can be accomplished by calling the executable `GEN_PROFILE` from the subdirectory `GEOMETRIES`. More precisely, suppose the profile line $\{(f_x(t), f_y(t)) :\ 0 \le t \le 1\}$ is given by the functions $t \mapsto f_x(t)$ and $t \mapsto f_y(t)$ defined by the c-code in the file "`GEOMETRIES`/profile.c". Then

```
GEN_PROFILE name 0.06
```

creates the file "name.inp" of the desired profile grating, where the profile curve is approximated by a polygonal line with a step size equal to 0.06 times the length of period (cf. Figure 12 where $f_x(t) = t$ and $f_y(t) = \{1.5 + 0.2\exp(\sin(6\pi t)) + 0.3\exp(\sin(8\pi t))\}/\{2\pi\}$).

If an input file for a grating determined by more than one non-intersecting and periodic profile lines given as $\{(f_x(j,t), f_y(j,t)) :\ \ 0 \le t \le 1\},\ j, = 1, \dots, n$ is needed, then this can be accomplished by calling the executable `GEN_PROFILES` from the subdirectory `GEOMETRIES`. More precisely, suppose n and the profile lines $\{(f_x(j,t), f_y(j,t)) :\ 0 \le t \le 1\}$ are given by the c-code in the file "`GEOMETRIES`/profiles.c". Then

```
GEN_PROFILES name 0.06
```

creates the file "name.inp" of the desired profile grating, where the profile curves are approximated by a polygonal line with a step size equal to 0.06 times the length of period

[9] To make it precise, suppose P and Q are the common corner points of the two polygonal curves and that R^1_1, R^1_2, … , R^1_m are the consecutive corner points between P and Q on the first polygonal line and R^2_1, R^2_2, … , R^2_n those on the second polygonal. Then the code requires $m = n$ and that the coating area between the two polygonal lines is the disjoint union of the triangle $PR^1_1R^2_1$, the quadrilaterals $R^1_1R^1_2R^2_2R^2_1$, … , $R^1_{(m-1)}R^1_mR^2_mR^2_{(m-1)}$, and the triangle $R^1_mQR^2_m$.

(cf. Figure 15 where $n = 3$, $f_x(j,t) = t$, $f_y(1,t) = \sin(2\pi t)$, $f_y(2,t) = \sin(2\pi t) - 0.5$, and $f_y(3,t) = \sin(2\pi t) - 1$).

If an input file for a pin grating determined by a simple non-intersecting profile line given as $\{(f_x(t), f_y(t)) : \ 0 \le t \le 1\}$ is needed[10], then this can be accomplished by calling the executable `GEN_PIN` from the subdirectory `GEOMETRIES`. More precisely, suppose x_{min} and the profile line $\{(f_x(t), f_y(t)) : 0 \le t \le 1\}$ are given by the c-code in the file "`GEOMETRIES`/pin.c". Then

`GEN_PIN` name 0.06

creates the file "name.inp" of the desired profile grating, where the profile curves are approximated by a polygonal line with a step size equal to 0.06 times the length of period (cf. Figure 16 where $x_{min} = 0.2$, $f_x(t) = x_{min} + (1 - 2x_{min})\,t$, and $f_y(t) = 0.5\,\sin(\pi t)$).

If an input file for a coated pin grating determined by the simple non-intersecting profile lines given as $\{(f_x(j,t), f_y(j,t)) : 0 \le t \le 1\}$, $j = 1,2$ is needed[11], then this can be accomplished by calling the executable `GEN_CPIN` from the subdirectory `GEOMETRIES`. More precisely, suppose x_{min}, a_1, a_2 and the profile lines $\{(f_x(j,t), f_y(j,t)) : 0 \le t \le 1\}$ j=1,2 are given by the c-code in the file "`GEOMETRIES`/cpin.c". Then

`GEN_CPIN` name 0.06

creates the file "name.inp" of the desired profile grating, where the profile curves are approximated by a polygonal line with a step size equal to 0.06 times the length of period (cf. Figure 17 where $x_{min} = 0.2$, $a_1 = 0.2$, $a_1 = 0.8$, $f_x(1,t) = x_{min} + (1 - 2x_{min})\,t$, $f_y(1,t) = 0.5\,\sin(\pi t)$, and $\{(f_x(2,t), f_y(2,t)) : \ 0 \le t \le 1\}$ is the polygonal curve connecting the four points $(f_x(1,a_1), f_y(1,a_1))$, $(f_x(1,a_1), f_y(1,0.5){+}0.1)$, $(f_x(1,a_2), f_y(1,0.5){+}0.1)$, and $(f_x(1,a_2), f_y(1,a_2))$).

If an input file for a coated pin grating of type 2 determined by the simple non-intersecting profile lines given as $\{(f_x(j,t), f_y(j,t)) : 0 \le t \le 1\}$, $j = 1,2$ is needed[12],

[10] I.e., over a flat grating with surface $\{(x,0) : \ 0 \le x \le 1\}$ a material part is attached which is located between $\{(x,0) : 0 \le x \le 1\}$ and $\{(f_x(t), f_y(t)) : 0 \le t \le 1\}$. Here $\{(f_x(t), f_y(t)) : 0 \le t \le 1\}$ is a simple open arc connecting $(f_x(0), f_y(0)) = (x_{min}, 0)$ with $(f_x(1), f_y(1)) = (1 - x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(t) < 1$, $0 < t < 1$, and such that $0 < f_y(t)$, $0 < t < 1$.

[11] I.e., over a flat grating with surface $\{(x,0) : 0 \le x \le 1\}$ a material part is attached which is located between $\{(x,0) : 0 \le x \le 1\}$ and $\{(f_x(1,t), f_y(1,t)) : 0 \le t \le 1\}$. Here $\{(f_x(1,t), f_y(1,t)) : 0 \le t \le 1\}$ is a simple open arc connecting $(f_x(1,0), f_y(1,0)) = (x_{min}, 0)$ with $(f_x(1,1), f_y(1,1)) = (1 - x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(1,t) < 1$, $0 < t < 1$, and such that $0 < f_y(1,t)$, $0 < t < 1$. Additionally, a coating layer is attached located between the first curve $\{(f_x(1,t), f_y(1,t)) : 0 \le t \le 1\}$ and a second curve $\{(f_x(2,t), f_y(2,t)) : 0 \le t \le 1\}$. The last connects the point $(f_x(1,a_1), f_y(1,a_1)) = (f_x(2,0), f_y(2,0))$ with $(f_x(1,a_2), f_y(1,a_2)) = (f_x(2,1), f_y(2,1))$. Moreover, $\{(f_x(2,t), f_y(2,t)) : 0 \le t \le 1\}$ is a simple open arc above $\{(f_x(1,t), f_y(1,t)) : 0 \le t \le 1\}$ such that $0 < f_x(2,t) < 1$, $0 < t < 1$.

[12] I.e., over a flat grating with surface $\{(x,0) : \ 0 \le x \le 1\}$ a material part is attached which is located between the line $\{(x,0) : \ 0 \le x \le 1\}$ and $\{(f_x(1,t), f_y(1,t)) : \ 0 \le t \le 1\}$. Here $\{(f_x(1,t), f_y(1,t)) : 0 \le t \le 1\}$ is a simple open arc connecting $(f_x(1,0), f_y(1,0)) = (x_{min}, 0)$ with $(f_x(1,1), f_y(1,1)) = (1 - x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(1,t) < 1$, $0 < t < 1$, and such that $0 < f_y(1,t)$, $0 < t < 1$. Additionally, a coating layer is attached located between the first curve $\{(f_x(1,t), f_y(1,t)) : \ 0 \le t \le 1\}$ united with the the two straight line segments $\{(x,0) : \ x_1 \le x \le x_{min}\}$ and $\{(x,0) : \ 1 - x_{min} \le x \le x_2\}$ and a second curve $\{(f_x(2,t), f_y(2,t)) : \ 0 \le t \le 1\}$. The last connects the first point $(x_1, 0) = (f_x(2,0), f_y(2,0))$ with the last point $(x_2, 0) = (f_x(2,1), f_y(2,1))$. Moreover, $\{(f_x(2,t), f_y(2,t)) : \ 0 \le t \le 1\}$ is a simple open arc above $\{(f_x(1,t), f_y(1,t)) : \ 0 \le t \le 1\}$ such that

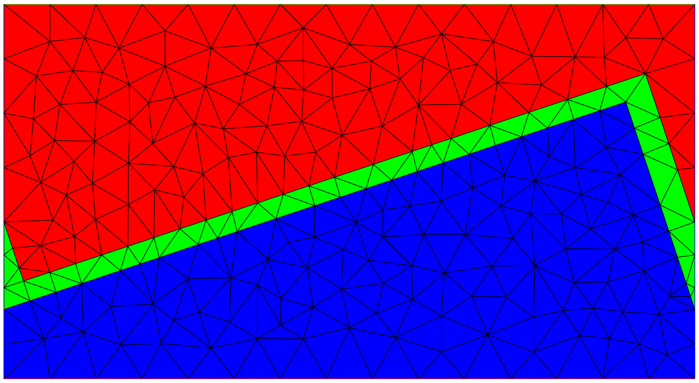

Figure 6: Echelle grating of type A.

then this can be accomplished by calling the executable `GEN_CPIN2` from the subdirectory `GEOMETRIES`. More precisely, suppose x_{min} and the profile lines $\{(f_x(j,t), f_y(j,t)) : 0 \leq t \leq 1\}$ j=1,2 are given by the c-code in the file "`GEOMETRIES`/cpin2.c". Then

```
GEN_CPIN2 name 0.05
```

creates the file "name.inp" of the desired profile grating, where the profile curves are approximated by a polygonal line with a step size equal to 0.05 times the length of period (cf. Figure 18 where $x_{min} = 0.2$, $f_x(1,t) = x_{min} + (1 - 2x_{min})\,t$, $f_y(1,t) = 0.5\,\sin(\pi t)$, and $f_x(2,t) = 0.5\,x_{min} + (1 - x_{min})\,t$, $f_y(2,t) = 0.8\,\sin(\pi t)$).

3.3 Code words to indicate special geometries

One can indicate special grating geometries in the input file "name.dat" by special code words. We explain these here.

```
# Grating data:
  echellea R 0.3 0.03 0.04
```

indicates an ECHELLE GRATING TYPE A (right-angled triangle with hypotenuse parallel to the direction of the periodicity, right interior angle greater than 45°, cf. Figure 6) with depth of 0.3 μm (i.e., triangle height = 0.3 μm) and with coated layers of height 0.03 μm resp. 0.04 μm over the first resp. second part of the grating (measured in direction perpendicular to the echelle profile, height greater or equal to zero).

```
# Grating data:
```

$0 < f_x(2,t) < 1$, $0 < t < 1$. The functions $f_x(1,.)$, $f_x(2,.)$, $f_y(1,.)$, and $f_y(2,.)$ and the parameter x_{min} are defined by the code of the file "../`GEOMETRIES`/cpin2.c".

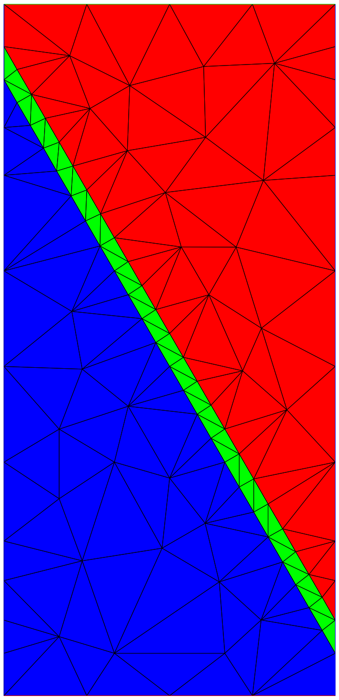

Figure 7: Echelle grating of type B.

```
echellea L 0.3 0.03 0.04
```

indicates an ECHELLE GRATING TYPE A (right-angled triangle with hypotenuse parallel to the direction of the periodicity, left interior angle greater than 45°) with parameters like above.

```
# Grating data:
   echellea A 60. 0.03 0.04
```

indicates an ECHELLE GRATING TYPE A (right-angled triangle with hypotenuse parallel to the direction of the periodicity) with left interior angle $\alpha = 60°$ (i.e. the depth is equal to the period multiplied by $\sin(\alpha)\cos(\alpha)$) and other parameters like above.

```
# Grating data:
   echelleb 60. 0.05
```

indicates an ECHELLE GRATING TYPE B (right-angled triangle with one of the legs parallel to the direction of the periodicity, cf. Figure 7) with angle 60° (angle enclosed by hypotenuse and by the leg parallel to the period) and with a coated layer of height 0.05 μm (measured in direction perpendicular to echelle profile, height greater or equal to zero).

```
# Grating data:
   trapezoid 60. 0.6 3 0.2 0.1 0.1 0.05
```

indicates a `TRAPEZOIDal Grating` (isosceles trapezoid with the basis parallel to the direction of the periodicity, cf. Figure 8) with angle of 60° (angle enclosed by basis and the sides) with a basis of length 0.6 μm consisting of 3 material layers of heights 0.2 μm, 0.1 μm, and 0.1 μm, respectively, and with a coated layer of height 0.05 μm (greater or equal to zero).

```
# Grating data:
```

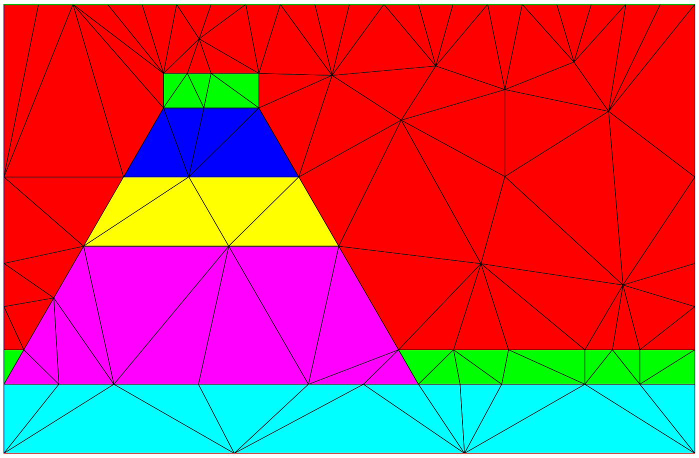

Figure 8: Trapezoidal grating.

```
lAmellar 3 4
0.2 0.6
-0.2 1.0
0.  0.5 0.7
0.0 0.50 0.90
0.00 0.500 0.900
```

indicates a Lamellar Grating (rectangular grating consisting of several materials placed in rectangular subdomains, cf. Figure 9) with 3 columns each divided into 4 rectangular layers, first column with x coordinate in 0 μm<x<0.2 μm, second column with 0.2 μm<x<0.6 μm, third column with 0.6 μm<x<period (period given above), whole grating with y coordinate s.t. -0.2 μm<y<1.0 μm, first column: first layer with -0.2 μm<y<0. μm, second with 0. μm<y<0.5 μm, third with 0.5 μm<y<0.7 μm and fourth with 0.7 μm<y<1. μm, second column: first layer with -0.2 μm<y<0.0 μm, second with 0.0 μm<y<0.50 μm, third with 0.50 μm<y<0.90 μm and fourth with 0.90 μm<y<1. μm, third column: first layer with -0.2 μm<y<0.00 μm, second with 0.00 μm<y<0.500 μm, third with 0.500 μm<y<0.900 μm and fourth with 0.900 μm<y<1. μm .

```
# Grating data:
   lAmellar 1 1
   0.2 0.8
```

indicates a Simple Layer (special case of lamellar grating) with layer material s.t. y-coordinate satisfies 0.2 μm<y<0.8 μm.

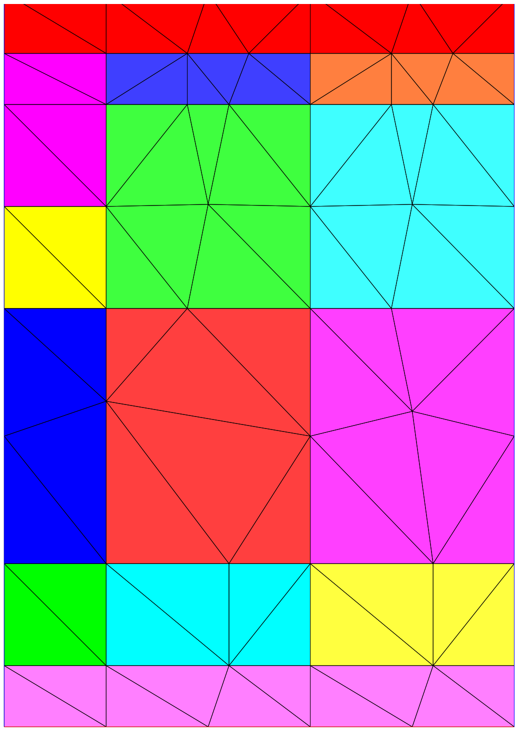

Figure 9: Lamellar grating.

```
# Grating data:
  polygon file1
```

indicates a Grating Determined by a Polygonal Line (cf. Figure 10) defined by the data in the file with name "`GEOMETRIES`/file1" (in "`GEOMETRIES`/file1": in each line beginning without '#' there should be the x- and y-coordinate of one of the consecutive corner points, first point with x-coordinate 0, last point with x-coordinate 1, same y-coordinate for first and last point, all x-coordinates between 0 and 1, at least two different y-coordinates, last line should be "End").

```
# Grating data:
  polygon2 file1 file2
```

indicates a Coated Grating Determined by Polygonal Lines (cf. Figure 11), i.e. the grating profile line is defined by the data in the file with name "`GEOMETRIES`/file1" (in "`GEOMETRIES`/file1": in each line beginning without '#' there should be the x- and y-coordinate of one of the consecutive corner points, first point with x-coordinate 0, last point with x-coordinate 1, same y-coordinate for first and last point, all x-coordinates between 0 and 1, at least two different y-coordinates, last line should be "End") and the coated layer is enclosed between the polygonal line of "`GEOMETRIES`/file1" and the polygonal line of the file with name "`GEOMETRIES`/file2" (in "`GEOMETRIES`/file2": in each line beginning without '#' there should be the x- and y-coordinate of one of the consecutive corner points, first and last point must be corner of first polygon, second polygon must be on left-hand side of first, one to one correspondence of the corners on the two polygons between first and

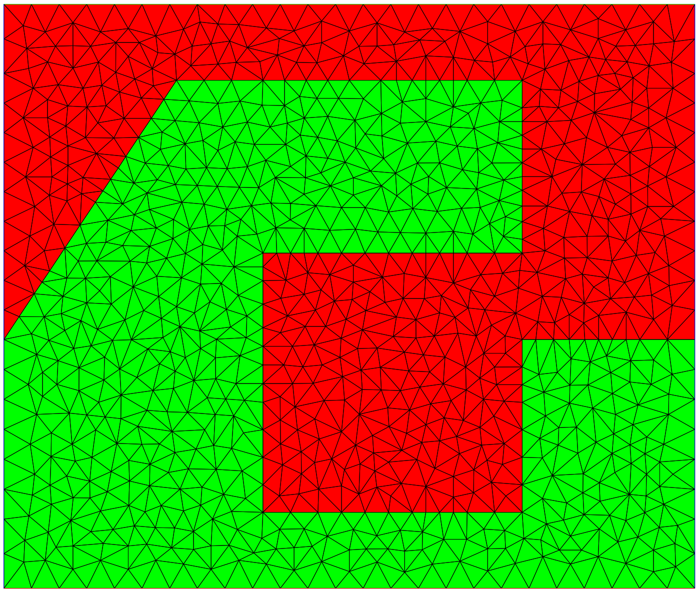

Figure 10: Grating determined by a polygonal line.

last point of second polygon[13], quadrilateral domain between corresponding segments on the left of first polygon, these quadrilaterals must be disjoint, last line should be "End").

```
# Grating data:
  profile
```

indicates a Grating Determined by a Smooth Parametric Curve, i.e. grating determined by profile line given as $\{(f_x(t), f_y(t)) : \ 0 \le t \le 1\}$, where the functions $t \mapsto f_x(t)$ and $t \mapsto f_y(t)$ are defined by the c-code of the file "GEOMETRIES/profile.c" (cf. Figure 12 where $f_x(t) = t$ and $f_y(t) = \{1.5 + 0.2 \exp(\sin(6\pi t)) + 0.3 \exp(\sin(8\pi t))\}/\{2\pi\}$).

```
# Grating data:
  profile_par 2 3
  1
  0
  1.5
  0.2
  0.3
```

indicates a Grating Determined by a Smooth Parametric Curve, With Parameters, i.e. grating determined by profile line given as $\{(f_x(t), f_y(t)) : \ 0 \le t \le$

[13] To make it precise, suppose P and Q are the common corner points of the two polygonal curves and that R^1_1, R^1_2, ... , R^1_m are the consecutive corner points between P and Q on the first polygonal line and R^2_1, R^2_2, ... , R^2_n those on the second polygonal. Then the code requires $m = n$ and that the coating area between the two polygonal lines is the disjoint union of the triangle $PR^1_1R^2_1$, the quadrilaterals $R^1_1R^1_2R^2_2R^2_1$, ... , $R^1_{(m-1)}R^1_mR^2_mR^2_{(m-1)}$, and the triangle $R^1_mQR^2_m$.

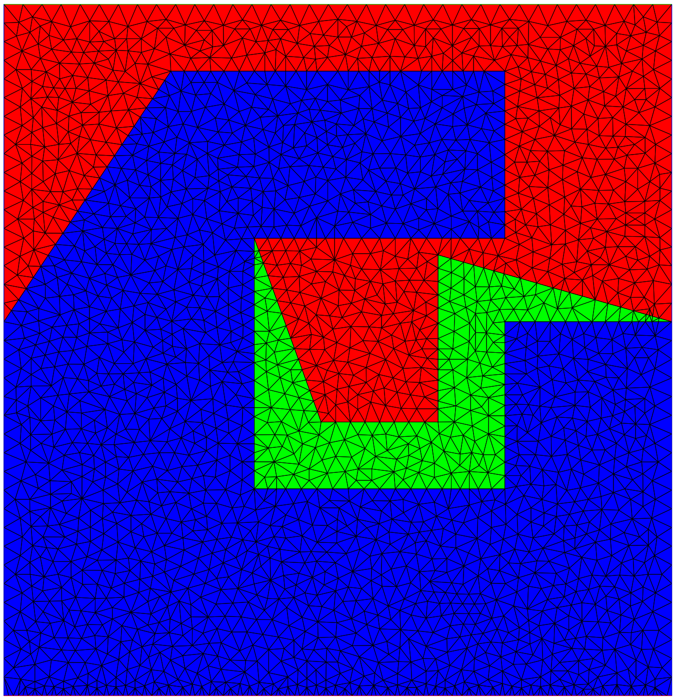

Figure 11: Grating determined by polygonal lines.

1}, where the functions $t \mapsto f_x(t)$ and $t \mapsto f_y(t)$ are defined by the c-code of the file "GEOMETRIES/profile_par.c". The last code uses 2 integer parameters and 3 real parameters named IPARaM1, IPARaM2, RPARaM1, RPARaM2, and RPARaM3. The integer parameters take the values 1 and 0 following the first line of the calling sequence and the real parameters take the values 0.15, 1., and 0. following the integer parameter values (cf. Figure 12 where $f_x(t) = t$ and $f_y(t) = \{1.5 + 0.2\exp(\sin(6\pi t)) + 0.3\exp(\sin(8\pi t))\}/\{2\pi\}$, parameter 1 is the index of the curve chosen from "GEOMETRIES/profile_par.c", parameter 0 is the number of corners of the curve, parameters 1.5, 0.2, and 0.3 are scaling parameters in the y-coordinate of the curve). Note that any number of parameters is possible for a corresponding file "GEOMETRIES/profile_par.c".

# Grating data:
    profile $0.125 * \sin(2. * \text{M\_PI} * t)$

indicates a Grating Determined by a Simple Smooth Function (cf. Figure 13), i.e. grating determined by a profile line given as $\{(t, f_y(t)) : \; 0 \le t \le 1\}$, where the function $t \mapsto f_y(t)$ is defined by the c-code $f_y(t) = 0.125\sin(2\pi t)$, (do not use any blank/space in the c-code).

# Grating data:
    profile $0.5 + 0.5 * \cos(\text{M\_PI} * (1. - t)) \;\; 0.25 * \sin(\text{M\_PI} * t)$

indicates a Grating Determined by a Simple Smooth Parametric Curve (cf. Figure 14), i.e., grating determined by ellipsoidal profile line given as $\{(f_x(t), f_y(t)) : \; 0 \le t \le 1\}$, where the functions $t \mapsto f_x(t)$ and $t \mapsto f_y(t)$ are defined by the c-codes $f_x(t) =$

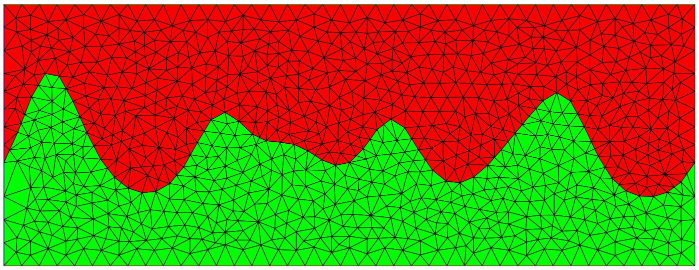

Figure 12: Grating determined by a smooth parametric curve.

$0.5 + 0.5\cos(\pi(1-t))$ and $f_y(t) = 0.25\sin(\pi t)$, respectively (no blank/space in c-code!).

```
# Grating data:
  profiles
```

indicates a Grating Determined by Smooth Parametric Curves, i.e. grating determined by n non-intersecting and periodic profile lines given from above to below as $\{(f_x(j,t), f_y(j,t)) : \ 0 \le t \le 1\}, \ j = 1, \dots, n$, where n and the functions $t \mapsto f_x(j,t)$ and $t \mapsto f_y(j,t)$ are defined by the c-code of the file "GEOMETRIES/profiles.c" (cf. Figure 15 where $n = 3$, $f_x(j,t) = t$, $f_y(1,t) = \sin(2\pi t)$, $f_y(2,t) = \sin(2\pi t) - 0.5$, and $f_y(3,t) = \sin(2\pi t) - 1$).

```
# Grating data:
  profiles_par 1 2
  3
  0.5
  0.50
```

indicates a Grating Determined by Smooth Parametric Curves, With Parameters, i.e. grating determined by n non-intersecting and periodic profile lines given from above to below as $\{(f_x(j,t), f_y(j,t)) : \ 0 \le t \le 1\}, \ j = 1, \dots, n$, where n and the functions $t \mapsto f_x(j,t)$ and $t \mapsto f_y(j,t)$ are defined by the c-code of file "GEOMETRIES/profiles_par.c". The last code uses 1 integer parameter and 2 real parameters named IPARaM1, RPARaM1, and RPARaM2. The integer parameter takes the value 3 following the first line of the calling sequence and the real parameters take the values 0.5 and 1. following the integer parameter values (cf. Figure 15 where $n = 3$, $f_x(j,t) = t$, $f_y(1,t) = \sin(2\pi t)$, $f_y(2,t) = \sin(2\pi t) - 0.5$,

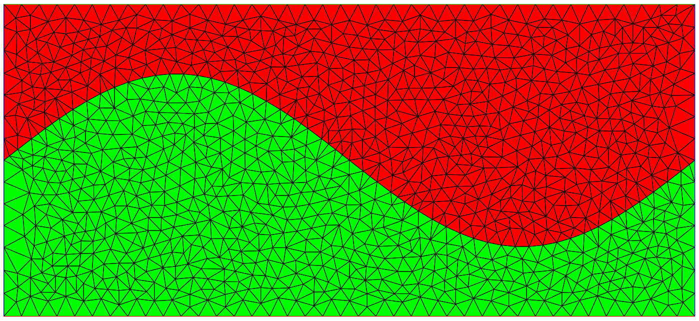

Figure 13: Grating determined by a simple smooth function.

and $f_y(3,t) = \sin(2\pi t) - (0.5 + 0.50)$, parameter 3 is the number n of boundary and interface curves, parameter 0.5 is the width of the first layer and parameter 0.50 that of the second). Note that any number of parameters is possible for a corresponding file "`GEOMETRIES`/profiles_par.c".

# Grating data:
pin

indicates a Pin Grating Determined By Parametric Curve (cf. Figure 16 where $x_{min} = 0.2$, $f_x(t) = x_{min} + (1 - 2x_{min})\,t$, and $f_y(t) = 0.5\,\sin(\pi t)$), i.e. over a flat grating with surface $\{(x,0) : 0 \leq x \leq 1\}$ a material part is attached which is located between $\{(x,0) : 0 \leq x \leq 1\}$ and $\{(f_x(t), f_y(t)) : 0 \leq t \leq 1\}$. Here $\{(f_x(t), f_y(t)) : 0 \leq t \leq 1\}$ is a simple open arc connecting $(f_x(0), f_y(0)) = (x_{min}, 0)$ with $(f_x(1), f_y(1)) = (1 - x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(t) < 1$, $0 < t < 1$, and such that $0 < f_y(t)$, $0 < t < 1$. The functions f_x, f_y and the parameter x_{min} are defined by the code in "../`GEOMETRIES`/pin.c".

# Grating data:
cpin

indicates a Coated Pin Grating Determined By Two Parametric Curves (cf. Figure 17 where $x_{min} = 0.2$, $a_1 = 0.2$, $a_1 = 0.8$, $f_x(1,t) = x_{min} + (1 - 2x_{min})\,t$, $f_y(1,t) = 0.5\,\sin(\pi t)$, and $\{(f_x(2,t), f_y(2,t)) : 0 \leq t \leq 1\}$ is the polygonal curve connecting the four points $(f_x(1,a_1), f_y(1,a_1))$, $(f_x(1,a_1), f_y(1,0.5) + 0.1)$, $(f_x(1,a_2), f_y(1,0.5) + 0.1)$, and $(f_x(1,a_2), f_y(1,a_2)))$, i.e. over a flat grating with surface $\{(x,0) : 0 \leq x \leq 1\}$ a material

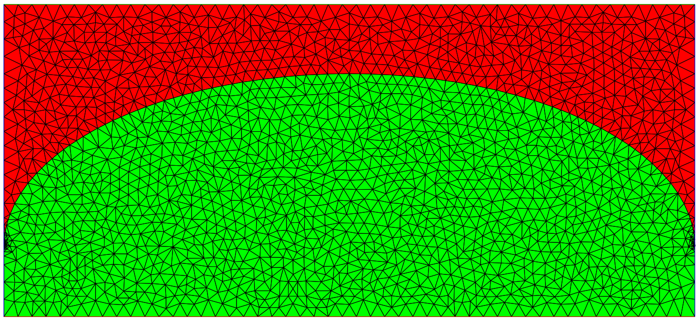

Figure 14: Grating determined by a simple smooth parametric curve.

part is attached which is located between $\{(x,0): \ 0 \leq x \leq 1\}$ and $\{(f_x(1,t), f_y(1,t)) : 0 \leq t \leq 1\}$. Here $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$ is a simple open arc connecting $(f_x(1,0), f_y(1,0)) = (x_{min}, 0)$ with $(f_x(1,1), f_y(1,1)) = (1-x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(1,t) < 1$, $0 < t < 1$, and such that $0 < f_y(1,t)$, $0 < t < 1$. Additionally, a coating layer is attached located between the first curve $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$ and a second curve $\{(f_x(2,t), f_y(2,t)) : \ 0 \leq t \leq 1\}$. The last connects the first point $(f_x(1,a_1), f_y(1,a_1)) = (f_x(2,0), f_y(2,0))$ with the last point $(f_x(1,a_2), f_y(1,a_2)) = (f_x(2,1), f_y(2,1))$. Moreover, $\{(f_x(2,t), f_y(2,t)) : \ 0 \leq t \leq 1\}$ is a simple open arc above $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$ such that $0 < f_x(2,t) < 1$, $0 < t < 1$. The functions $f_x(1,.)$, $f_x(2,.)$, $f_y(1,.)$, and $f_y(2,.)$ and the parameters a_1, a_2, and x_{min} are defined by the code of the file "../`GEOMETRIES`/cpin.c".

```
# Grating data:
  cpin2
```

indicates a Coated Pin Grating Determined By Two Parametric Curves, Type 2 (cf. Figure 18 where $x_{min} = 0.2$, $f_x(1,t) = x_{min} + (1-2x_{min})\,t$, $f_y(1,t) = 0.5\,\sin(\pi t)$, and $f_x(2,t) = 0.5\,x_{min} + (1-x_{min})\,t$, $f_y(2,t) = 0.8\,\sin(\pi t)$), i.e. over a flat grating with surface $\{(x,0): \ 0 \leq x \leq 1\}$ a material part is attached which is located between the line $\{(x,0) : 0 \leq x \leq 1\}$ and $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$. Here $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$ is a simple open arc connecting $(f_x(1,0), f_y(1,0)) = (x_{min}, 0)$ with $(f_x(1,1), f_y(1,1)) = (1-x_{min}, 0)$ such that $0 < x_{min} < 0.5$ is a fixed number, such that $0 < f_x(1,t) < 1$, $0 < t < 1$, and such that $0 < f_y(1,t)$, $0 < t < 1$. Additionally, a coating layer is attached located between the first curve $\{(f_x(1,t), f_y(1,t)) : \ 0 \leq t \leq 1\}$ united with the the

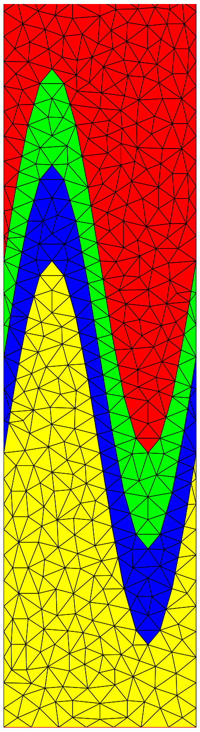

Figure 15: Grating determined by smooth parametric curves.

two straight line segments $\{(x,0):\ x_1 \le x \le x_{min}\}$ and $\{(x,0):\ 1-x_{min} \le x \le x_2\}$ and a second curve $\{(f_x(2,t),f_y(2,t)):\ 0 \le t \le 1\}$. The last connects the first point $(x_1,0) = (f_x(2,0),f_y(2,0))$ with the last point $(x_2,0) = (f_x(2,1),f_y(2,1))$. Moreover, $\{(f_x(2,t),f_y(2,t)):\ 0 \le t \le 1\}$ is a simple open arc above $\{(f_x(1,t),f_y(1,t)):\ 0 \le t \le 1\}$ such that $0 < f_x(2,t) < 1$, $0 < t < 1$. The functions $f_x(1,.)$, $f_x(2,.)$, $f_y(1,.)$, and $f_y(2,.)$ and the parameter x_{min} are defined by the code of the file "../`GEOMETRIES`/cpin2.c".

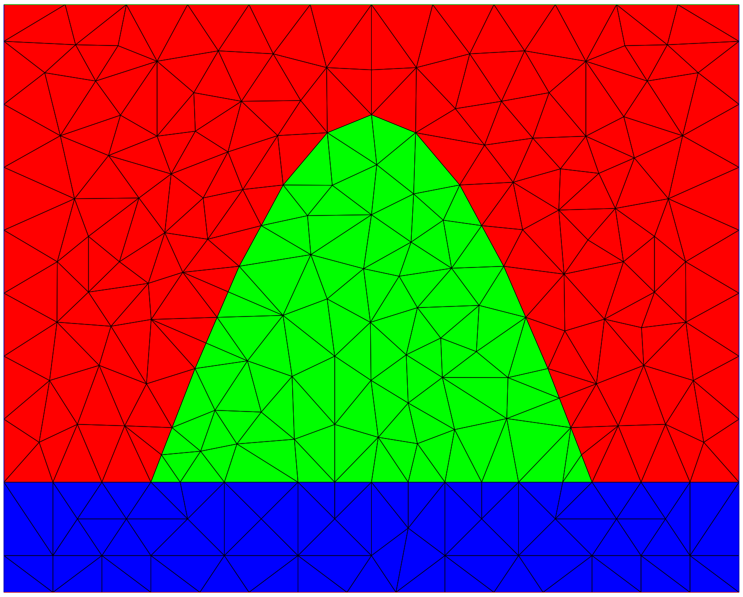

Figure 16: Pin grating determined by parametric curve.

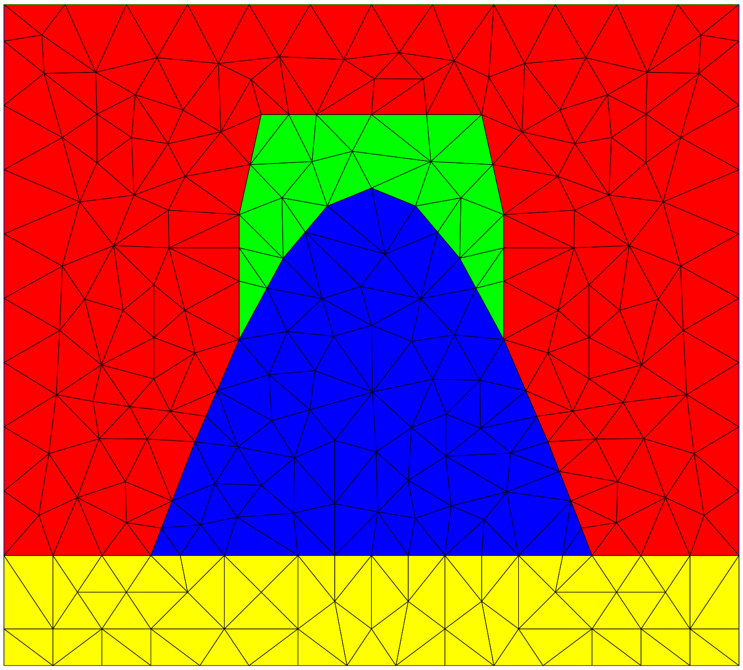

Figure 17: Coated pin grating determined by two parametric curves.

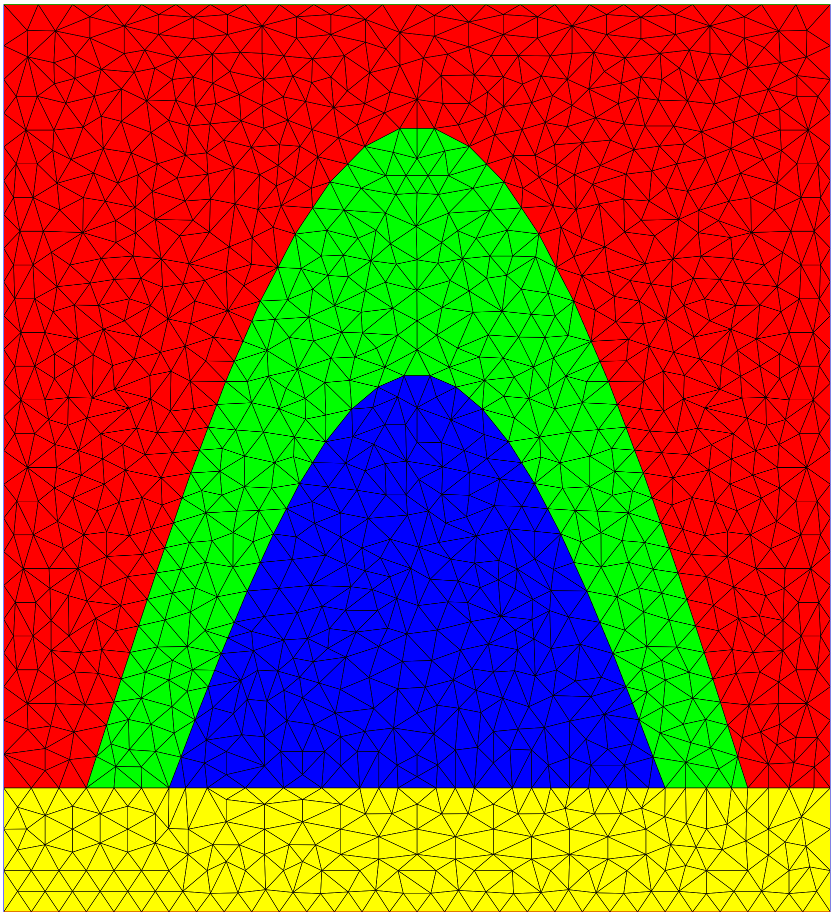

Figure 18: Coated pin grating determined by two parametric curves, Type 2.

3.4 Stack grating by code words

For the stack grating, there appear the following code words in the geometry input of the name.dat file:

```
# Grating data:
  stack k
  line_1
  line_2
     ...
  line_2k
```

Here k is the number of profile curves in the stack. These profile curves are defined by the following $2k$ code word lines. Each profile curve is represented by two of the lines. They are listed from above to below. No intersection points of these curves are allowed. With the exception of pin curves the j-th curve ($j = 1, \dots, k$) takes the form $\{(f_x(j,t), f_y(j,t)) : 0 \le t \le 1\}$ with the first end point such that $f_x(j,0) = 0$, with the second end point such that $f_x(j,1) = 1$, and with $0 < f_x(j,t) < 1$ for $0 < t < 1$.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{echellea } \mathbf{R} \; d_1 \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get an Echelle Type A grating profile with right blaze angle greater than 45°, with depth d_1 and without coated layer (cf. Section 3.3). The profile is vertically shifted by h_1 μm. Note that $0 < d_1$ and d_1 is less than half the period d. If the input letter **R** is replaced by **L**, then the left interior angle is greater than 45°. Moreover, if the input letter **R** is replaced by **A** and the following number d_1 by α, then the left interior angle is α degrees.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{echelleb } \alpha \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get an Echelle Type B grating profile with angle α and without coated layer (cf. Section 3.3). The profile is vertically shifted by h_1 μm. Note that $0 < \alpha < 90$.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{binary } i \; d_0 \; d_1 \; d_2 \dots d_{2i} \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get a Binary Grating Profile with i teeth, with height d_0, with transition points d_1, d_2, ..., d_{2i}, and without coating layer. In other words, before the shift, the grating function is zero between 0 and d_1, between d_2 and d_3, ..., between d_{2i-2} and d_{2i-1}, and between d_{2i} and the period d. The grating function is d_0 between d_1 and d_2, between d_3 and d_4, ..., between d_{2i-3} and d_{2i-2}, and between d_{2i-1} and d_{2i-1}. The profile is vertically shifted by h_1 μm. Note that $i \le 6$, $0 < d_0$, $0 < d_1 < d_2 < \dots < d_{2i} < d$. For i equal to 1 or 2, $d_1 = 0$ and $d_{2i} = d$ is allowed.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{trapezoid } d_1\ d_2\ d_3\ \alpha \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get a symmetric Trapezoidal Grating profile (cf. Section 3.3) with the trapezoid starting at $x = d_1$, ending at $x = d_2$, and with the angle α and the height d_3. The profile is vertically shifted by h_1 μm. Note that we require $d_1 > 0$, $0 \leq d_2 < d_3 \leq 1$, and $0 < \alpha < 90$. The height must be sufficiently small such that the trapezoid does not degenerate.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{profile } \mathit{ccode} \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get a profile curve Determined By A Smooth Simple Curve defined by $f_x(j,t) = t$ and $f_y(j,t) = \mathit{ccode}$. The profile is vertically shifted by h_1 μm. Note that *ccode* is an expression of the parameter argument t, $0 \leq t \leq 1$, written in the c programming language. This expression will appear in the code as "$fct = ccode$;". Even an if case is possible. E.g. substituting the code "$0; if(t < 0.3)fct = t; else\ if(t < 0.6)fct = 0.6 - t; else\ fct = 0.$" into "$fct = ccode$;" leads to the meaningful code:

```
                      fct=0;
if          (t<0.3)   fct=t;
else   if   (t<0.6)   fct=0.6-t;
else                  fct=0.;
```

The code must be simple since the program can read no more than 399 symbols per input line.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{profile } \mathit{ccode}_1\ /\#\#/\ \mathit{ccode}_2 \\ \text{line}_{2j} & = & h_1 \end{array}$$

then we get a profile curve Determined By A Smooth Simple Parametric Curve defined by $f_x(j,t) = \mathit{ccode}_1$ and $f_y(j,t) = \mathit{ccode}_2$. The profile is vertically shifted by h_1 μm. The same remarks as in the previous mode apply.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{profile}i\ \mathit{ccode} \\ \text{line}_{2j} & = & h_1\ d_1\ d_2\ \ldots\ d_i \end{array}$$

then we get a profile curve Determined By A Non-smooth Simple Curve defined by $f_x(j,t) = t$ and $f_y(j,t) = \mathit{ccode}$. The curve has i $(1 \leq i \leq 9)$ corners with the parameter arguments d_1, d_2, …, d_i such that $0 < d_1 < d_2 < \ldots < d_i$. The profile is vertically shifted by h_1 μm. The same remarks as in the previous mode apply.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{profile}i\ \mathit{ccode}_1\ /\#\#/\ \mathit{ccode}_2 \\ \text{line}_{2j} & = & h_1\ d_1\ d_2\ \ldots\ d_i \end{array}$$

then we get a profile curve Determined By A Non-smooth Simple Parametric Curve defined by $f_x(j,t) = \mathit{ccode}_1$ and $f_y(j,t) = \mathit{ccode}_2$. The curve has i $(1 \leq i \leq 9)$ corners with the parameter arguments d_1, d_2, … ,d_i such that $0 < d_1 < d_2 < \ldots < d_i$. The profile is vertically shifted by h_1 μm. The same remarks as in the previous mode apply.

Beside the above profile curves, pin curves are possible. Then the meaning of f_x and f_y is changed. The curve $t \mapsto (f_x(j,t), f_y(j,t))$ with $0 \leq t \leq 1$ connects the points $(f_x(j,0), f_y(j,0)) = (0,0)$ and $(f_x(j,1), f_y(j,1)) = (1,0)$. The corresponding pin curve is just the affine image of the last curve connecting the two points $(f_x(j_2,p_1), f_y(j_2,p_1))$ and $(f_x(j_2,p_2), f_y(j_2,p_2))$ of the profile curve with index j_2. I.e.:

$$\begin{aligned} t \quad \mapsto \quad \Big(& f_x(j_2,p_1) \\ & + f_x(j,t) * [f_x(j_2,p_2) - f_x(j_2,p_1)] \\ & - f_y(j,t) * [f_y(j_2,p_2) - f_y(j_2,p_1)], \\ & f_y(j_2,p_1) \\ & + f_x(j,t) * [f_y(j_2,p_2) - f_y(j_2,p_1)] \\ & + f_y(j,t) * [f_x(j_2,p_2) - f_x(j_2,p_1)] \Big) \end{aligned}$$

Thus, for each of the above profile curves, we can define several areas with new material attached to it and bounded by the just mentioned pin curves. These areas are listed immediately before the profile curve and in correspondence with their attachment from the right to the left. In other words, the index j_2 of the profile curve to which the j-th pin curve is attached to is the smallest integer l larger than j such that l is an ordinary profile curve, and, in case of two and more pins, the j-th pin curve is located to the right of the $(j+1)$-th pin curve. No intersections of pin and profile curves are allowed.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{pin } ccode \\ \text{line}_{2j} & = & p_1 \; p_2 \end{array}$$

then we get a Simple Smooth Pin Curve defined by $f_x(j,t) = t$ and $f_y(j,t) = ccode$. The parameter arguments p_1 and p_2 of the connection points to the ordinary profile curve are fixed by the second line. The remarks on the profile curves apply also here.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{pin } ccode_1 \; /\#\#/ \; ccode_2 \\ \text{line}_{2j} & = & p_1 \; p_2 \end{array}$$

then we get a Simple Smooth Parametric Pin Curve defined by $f_x(j,t) = ccode_1$ and $f_y(j,t) = ccode_2$. The parameter arguments p_1 and p_2 of the connection points to the ordinary profile curve are fixed by the second line. The remarks on the profile curves apply also here.

If the defining two lines are

$$\begin{array}{lcl} \text{line}_{2j-1} & & \text{pin}i \; ccode \\ \text{line}_{2j} & = & p_1 \; p_2 \; d_1 \; d_2 \; \ldots \; d_i \end{array}$$

then we get a Simple Non-smooth Pin Curve defined by $f_x(j,t) = t$ and $f_y(j,t) = ccode$. The curve has i $(1 \leq i \leq 9)$ corners with the parameter arguments d_1, d_2, ..., d_i such that $0 < d_1 < d_2 < \ldots < d_i$. The remarks on the profile curves apply also here.

If the defining two lines are

line$_{2j-1}$ pin$i$ $ccode_1$ /##/ $ccode_2$
line$_{2j}$ = p_1 p_2 d_1 d_2 $\ldots$ d_i

then we get a Simple Non-smooth Parametric Pin Curve defined by $f_x(j,t) = ccode_1$ and $f_y(j,t) = ccode_2$. The curve has i $(1 \leq i \leq 9)$ corners with the parameter arguments d_1, d_2, $\ldots$, d_i such that $0 < d_1 < d_2 < \ldots < d_i$. The remarks on the profile curves apply also here.

For example, Figure 19 presents the stack grating generated by

```
# Grating data:
  stack 5
  echellea R 0.3
  1.2
  profile 0.3 * sin(2. * M_PI * t)
  0.8
  pin 0.5 - 0.5 * cos(M_PI * t) /##/ sin(M_PI * t)
  0.6 0.9
  pin1 0; if (t < 0.5) fct = t; else fct = 1. - t
  0.1 0.4 0.5
  profile t /##/ 0.
  0.
```

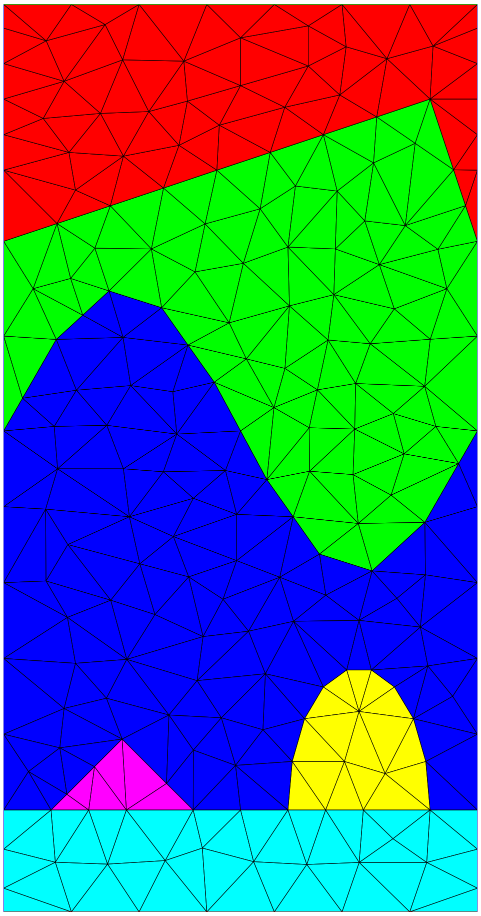

Figure 19: Stack grating.

4 Input of Refractive Indices

The optical properties of the materials involved in the grating are characterized by the refractive indices. Hence, for each material piece, the corresponding index must be added through the input file “name.dat”. This is done in lines like the following:

```
# Optical index (refractive index) of cover material.
  Air
# Optical indices of the materials of the upper coatings.
  1.1
  1.2
# Optical indices of the materials of the lower coatings.
  2.3 +i .0
  2.2 +i .5
  2.1 +i .0
# Optical index of substrate material.
  Al

  ...

# Wave length in micro m (lambda).
  .635

  ...

# Temperature:
  20.

  ...

# Optical indices of grating materials.
  1.2
  user
  1.7 +i .0
  2.3 +i .0
```

As seen in this example, the indices can be added as real or complex numbers (e.g. “1.1” resp. “2.2 +i .5”) or as code words of known materials (e.g. “Al” for aluminum). In the last case there is a program which interpolates the refractive index from a table in dependence on the temperature and on the wave length. The temperature enters only through such materials given by code words. If the indices are all numbers, then the temperature is not used. The code words for materials can be “Air”, “Ag”, “Al”, “Au”, “CsBr”, “Cu”, “InP”, “MgF2”, “NaCl”, “PMMA”, “PSKL”, “SF5”, “Si”, “TlBr”, “TlCl”, “Cr”, “ZnS”, “Ge”, “TiO2r”, “Quarz”, “AddOn”, and “Si1.0” - “Si2.0”. Here “Sia.b” with the real number “x=a.b” indicates a blending of “SiO” and “SiO_2” with the refractive index $n = (2-x) \cdot n_{SiO} + (x-1) \cdot n_{SiO_2}$. For example, “Air” corresponds to an index $n = 1$.

On the other hand, the value of the refractive index can be interpolated from a user defined table, indicated by the name of the file. This file is to be located in the current directory of the computation (`CLASSICAL/CONICAL`). Its name must begin with the letter “u” and may consist of no more than five letters like e.g. “user”. The file consists of at most 112 lines each with three real numbers, the first is the wave length in micro meter, the second the real part of the corresponding optical index, and the third the imaginary part of the index. At the end of each line a comment beginning with the sign # can be added. Optical indices with negative real or imaginary parts are not admitted.

As seen in the example presented above, first the index of the cover material is given. Then the indices of the materials of upper coated layers follow. These are rectangular layers over the whole period, and their number and widths are given in extra lines before the indices not presented in the example lines from above. If the number of coated layers is zero, then no lines with optical indices are needed. Next the indices of the materials of lower coated layers and that of the substrate follow. The indices of the materials in the area between upper and lower coatings (resp. between cover and substrate material if no rectangular coatings exist) are the last refractive indices of the input files. These indices are listed from above to below if possible. In some cases the ordering is indicated in the description of the geometrical part, or the indices have to be in accordance with the numbering of the material parts in the file "name.inp". In any case, the first index of the "grating materials" is to be the same as that of the adjacent last upper coating layer (resp. of the cover if there does not exist any rectangular upper coating) and the last index of the "grating materials" is to be the same as that of the adjacent first lower coating layer (resp. of the substrate if there does not exist any rectangular lower coating).

The input of refractive indices can be checked using the executables `FEM_CHECK` resp. `GFEM_CHECK` (cf. Sect. 5.3).

5 Computation of Efficiencies Using FEM in CLASSICAL

5.1 How to get an input file "name2.dat"?

First an input file "name2.dat" (cf. the enclosed data file in 10.2) in the subdirectory `CLASSICAL` is needed. To get this, change the directory to `CLASSICAL`, copy one of the existing files with tag ".dat", e.g. the file "example.dat" and call it "name2.dat".

```
cd DIPOG-2.0/CLASSICAL
cp example.dat name2.dat
```

Change "name2.dat" in the editor according to your requirements. You will find the necessary information as comments in the file "name2.dat". Indeed, each line beginning with "#" is a comment. Comment lines can be added and deleted without any trouble.

5.2 Simple calculation with minimal output

Now enter the command:

```
FEM name2
```

The program is running and produces an output on the screen. Additionally, a result file is produced (compare the similar file enclosed in point 10.6) the name of which is announced on the screen. You will find all Rayleigh coefficients, the efficiencies, and energies on both the screen and in this file. Note that the result file has the tag “.res” and is located in the subdirectory `RESULTS`. If a lot of data is produced, then computer programs should have an easy access to the data. To enhance readability by computer, a second output file can be produced setting a switch in “name2.dat” to yes. The name of this second file is the same as that of the first but with tag “.erg” instead of tag “.res”. The file is normally located in the subdirectory `RESULTS`. The name “name3” of the result files “`../RESULTS/name3.res`” resp. “`../RESULTS/name3.erg`” is indicated by the “name2.dat” lines:

```
# Name of output file.
  name3
```

However, if the code is started from a directory different from `CLASSICAL` or if the output file should be written into a different directory, then the file is to be specified by adding its path as:

```
# Name of output file.
  path/name3
```

Again the tag “.res” resp. “.erg” will be added. In particular, for an output file in the current working directory use:

```
# Name of output file.
  ./name3
```

The computation proceeds over several levels, where the mesh size is halved at each new level. The maximal number of levels is indicated at the end of “name2.dat” e.g. by:

```
# Number of levels.
  3
```

However, if efficiencies are computed for more than one angle of incidence or for several wave lengths (or if a single value of angle/wave length is given by incremental or vector mode, i.e. beginning with the letter I or V), then the computation is performed for the highest level, only.

The ideal choice for the level would be the minimal positive integer such that the solution falls under a certain error bound. We have implemented the following choice. If “name2.dat” contains

```
# Number of levels.
  e ε
```

with the lower case letter “e” and with ε a number greater than zero, then the code computes the efficiency for the levels 1, 2, ... (but no more than 15) until the maximum of the differences of efficiencies corresponding to two consecutive levels is less than ε. Thus, assuming a monotonic convergence, the smallest level for the given error bound ε is the smaller one of the last two consecutive levels. The efficiencies will be presented on the screen and in the output files for this level. (In other words, a computation for a level higher by one than that of the output is necessary for this variant.) If a computation over several angles, wave lengths or polarizations is required, then the “optimal” level will be determined for the first angle, for the first wave length resp. for the TM polarization, only.

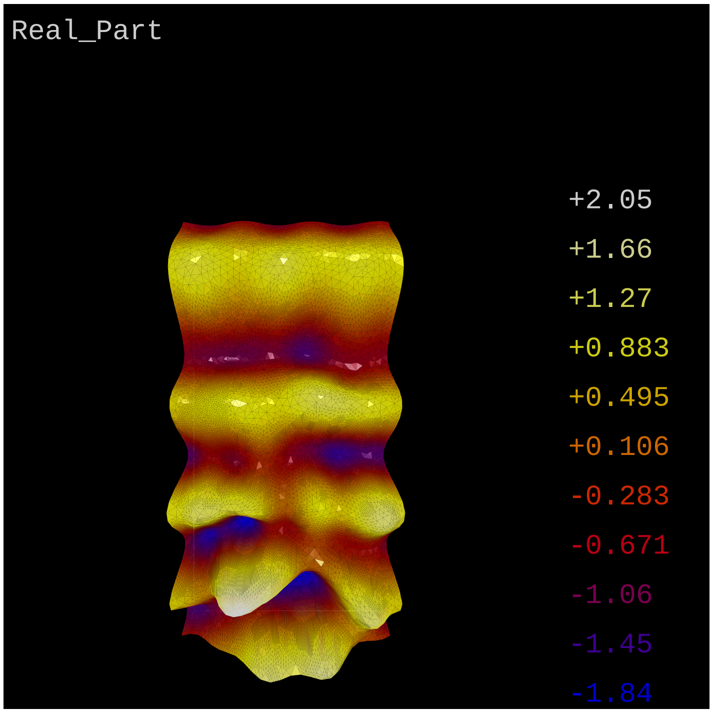

Figure 20: Real part of transverse component of magnetic field. Output of `FEM_PLOT` via `openGL` in `CLASSICAL`.

All other calculations are performed with this level. Clearly, there is no warranty that the efficiencies really deviate by a number less than ε from the true values.

5.3 Check before computation, more infos, and plots?

Instead, if `openGL` is available and if you wish to check your input data, then use the command:

```
FEM_CHECK name2
```

All the input information without output data will appear on the screen and in the result file. Moreover, there will appear a picture of the grating geometry with indicated refractive indices on the screen. The picture looks like that on the right-hand side in Figure 5.

Instead, if you use the command

```
FEM_FULLINFO name2
```

then the same is done as in point 5.2. Additionally, there appears more information (including the full input data and the convergence history, cf. the enclosed file in point 10.5) on the screen and in the result file.

Instead, if `openGL` or `GNUPLOT` is available and if you use the command

```
FEM_PLOT name2
```

then you have the same results as in point 5.2. Additionally, you will see pictures of the real part, the imaginary part (cf. Figure 20 for an `openGL` picture and Figure 21 for a `GNUPLOT` picture), and the square modulus of the solution (z-component of electric field for TE polarization, z-component of magnetic field for TM polarization). Note that the square

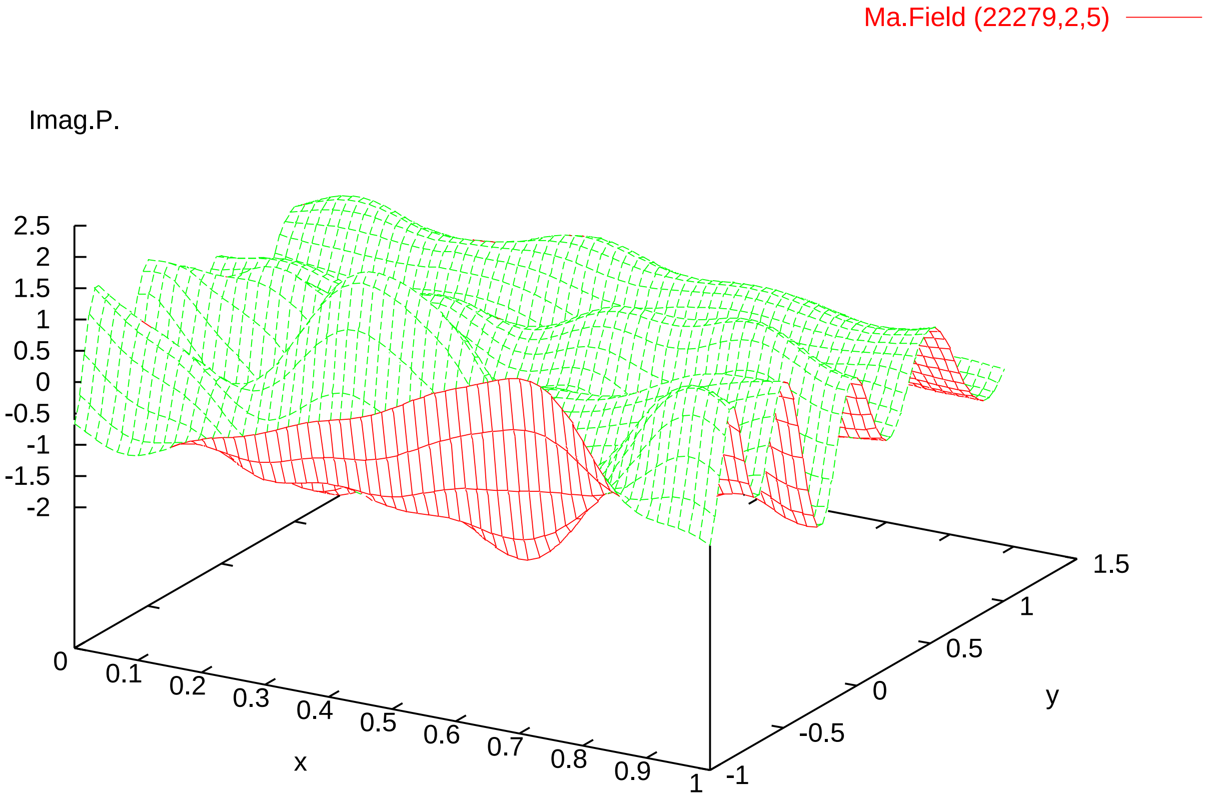

Figure 21: Imaginary part of transverse component of magnetic field. Output of executable `GFEM_PLOT` in `CLASSICAL` via `GNUPLOT`.

modulus is proportional to the energy intensity distribution of the wave. Moreover, similar pictures for the fields above and below the coated grating area will be plot. Program stops at each picture.

To control the graphical facilities of `GLTOOLS` in `FEM_PLOT`, use:

Backspace: Enter user control mode.
tab: toggle state change mode.
Return: Quit user control mode.
Space: Mode control.
+: Increase mouse sensitivity.
,: decrease control parameter.
-: Decrease mouse sensitivity.
.: increase control parameter.
¡: Zoom out.
¿: Zoom in.
?: This help.
B: Toggle background color (black/white).
d: Dump actual picture to ppm file (look for *-*.ppm).
F: Toggle rendering volume frame (bounding box) drawing.
I: Change number of isolines.

.: increase control parameter by a factor.
O: Toggle Ortho.
D: Print actual picture using ppm dump.
R: Reset to internal default.
S: Save actual state (look for .*-rndstate).
V: Start/Stop video recording.
a: Switch to GUI.
c: Toggle remembered lists.
g: Toggle Gouraud/flat shading.
h: This help.
i: Toggle isoline mode.
,: decrease control parameter by a factor.
l: Toggle level surface mode.
m: Toggle model display when moving.
p: Dump actual picture to eps file (look for *-*.eps).
q: Mode control (Quit).
r: Restore last saved state.
v: Toggle vscale for plane sections.
w: toggle wireframe mode.
x: Show x orthogonal plane section.
y: Show y orthogonal plane section.
z: Show z orthogonal plane section.
prev: toggle state change mode.
next: toggle state change mode.
left: move left.
up: move up.
right: move right.
down: move down.
Backspace: Enter user control mode.

To continue the computation of the main program press Bar/Space.

After the installation with the package GNUPLOT, the pictures are created interactively with the program on the main terminal window. To continue the computation of the main program, click the main terminal window and press Enter/Return.

6 Computation of Efficiencies Using GFEM in CLASSICAL

The same computation from the last section can be performed by generalized Fem (cf. the result file enclosed in point 10.6). The latter is nothing else than the variational approach of the conventional Fem combined with a new trial space for the approximation of the unknown solution. The trial space is defined over the triangular Fem partition and the trial functions are piecewise approximate solutions of the Helmholtz equation. More precisely, for integers n_{DOF} and n_{LFEM} with $0 < n_{\text{DOF}}$, with $1 < n_{\text{LFEM}}$, and with $[n_{\text{LFEM}} + 1]$ a multiple

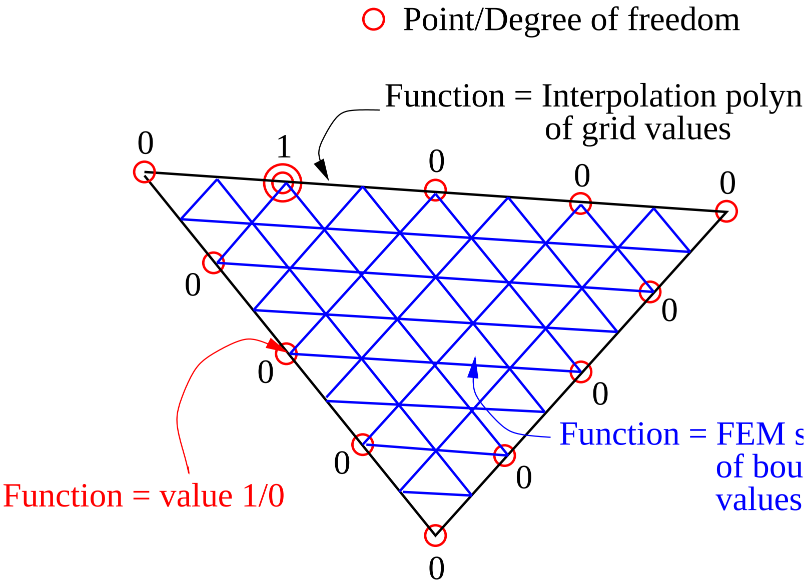

Figure 22: Trial basis function over a single grid triangle.

of $[n_{\mathrm{DOF}}+1]$, the degrees of freedom of the trial space are the function values at the corner points of the triangulations and at the n_{DOF} points of a uniform partition of each triangle side. The restrictions of the trial functions to the triangle sides are polynomial interpolants of the degrees of freedom. The restrictions of the trial functions to the triangles are the finite element solutions of the Dirichlet problem for the Helmholtz equation over a uniform triangulation of the partition triangle into $[n_{\mathrm{LFEM}}+1]\times[n_{\mathrm{LFEM}}+1]$ equal subtriangles (cf. the trial functions over the grid triangle indicated in Figure 22 with $n_{\mathrm{DOF}}=3$ and $n_{\mathrm{LFEM}}=7$). Hence, `GFEM` treats the same problems as `FEM` but is more convenient to treat higher wave numbers and faster for simple geometries. In order to accelerate the computation, one can use a Fem grid which is a uniform refinement of a coarse grid with each coarse triangle split into $[n_{\mathrm{UPA}}\times n_{\mathrm{UPA}}]$ congruent subtriangles. In this case, the trial space over the congruent grid triangles (approximate Dirichlet solutions) have to be computed only ones and can be reused several times.

To use generalized Fem for Fem, the executables `FEM`, `FEM_CHECK`, `FEM_FULLINFO`, and `FEM_PLOT` are to be replaced by the codes `GFEM`, `GFEM_CHECK`, `GFEM_FULLINFO`, and `GFEM_PLOT`, respectively. These executable use the same input file “name2.dat” but require the additional input file “generalized.Dat” (cf. the enclosed file in 10.3). Normally the latter is to be located in the current working directory. If there is no such file in the current working directory and if the output is written into a directory indicated by a certain path, then the code looks for the “generalized.Dat” file also in the directory determined by this path. The file “generalized.Dat” fixes the parameters:

n_{DOF}

Additional degrees of freedom on each triangle side. Indeed, the trial functions on each subdivision triangle are approximate solutions of the Dirichlet problem for the Helmholtz equation s.t. their restriction to the triangle sides coincides with the Lagrange interpolation polynomials on the triangle side. Here interpolation is taken over a uniform grid with $[n_{\mathrm{DOF}} + 2]$ interpolation knots including the two end-points.

n_{LFEM}

Approximate solution determined by FEM over subdivision triangle, where an additional uniform FEM partition on each grid triangle is chosen such that the step size is side length divided by $[n_{\mathrm{LFEM}} + 1]$.

n_{UPA}

This is for additional uniform partition of all primary grid triangles into $n_{\mathrm{UPA}} \times n_{\mathrm{UPA}}$ equal subdomains, i.e. the original side of the grid triangle is split into n_{UPA} sides of uniform partition subtriangles. After this uniform refinement the degrees of freedom and the trial space of approximate Helmholtz solutions are defined using n_{DOF} and n_{LFEM}.

Change these parameters according to your diffraction problem, computer memory capacity, and computing time requirements. How should they be chosen?

Suppose that n_{GP} is the number of grid points. Roughly speaking: For the FEM, the computation time as well as the necessary storage capacity is proportional to $[n_{\mathrm{GP}}]^2$. The time for `GFEM` is proportional to

$$\frac{[n_{\mathrm{GP}}]^2 \times [n_{\mathrm{LFEM}} + 1]^2}{[n_{\mathrm{UPA}}]^2} + [n_{\mathrm{GP}}]^2 \times [n_{\mathrm{DOF}} + 1]^2 \tag{6.1}$$

and the storage to $[n_{\mathrm{GP}}]^2 \times [n_{\mathrm{DOF}} + 1]^2$. Hence, a doubling of $[n_{\mathrm{LFEM}} + 1]$ leads to about the same computing time as halving the mesh size of the grid. Taking into account (6.1), we recommend to choose n_{UPA} as high as possible since the accuracy is almost independent of n_{UPA} but the computing time reduces significantly. The only exceptional case, when a larger n_{UPA} is not efficient, occurs if the geometry forces the triangulation to have a few large triangles and a huge number of small triangles (e.g. geometries with thin layers). In this case, the next level uniform partition increases the number of grid points and the computation time significantly, whereas a standard triangulation with halved mesh size leads to a small increase of grid points and to about the same numerical error.

If $n_{\mathrm{DOF}} = 0$ and $n_{\mathrm{LFEM}} = 1$, then the conventional FEM is computed. With $n_{\mathrm{DOF}} = 0$, a higher n_{LFEM} is not recommended. For $n_{\mathrm{DOF}} > 1$ and not so restrictive accuracy requirements, $n_{\mathrm{LFEM}} = 2n_{\mathrm{DOF}} + 1$ is a good choice. For $n_{\mathrm{DOF}} > 1$ and challenging accuracy requirements, a larger n_{LFEM} is useful. E.g. if $n_{\mathrm{DOF}} = 3$, then $n_{\mathrm{LFEM}} = 31$ is a good choice. For $n_{\mathrm{DOF}} = 7$, one should take, e.g., $n_{\mathrm{LFEM}} = 127$ and, for $n_{\mathrm{DOF}} = 15$, the value $n_{\mathrm{LFEM}} = 511$. However, large n_{LFEM} will lead to long computation times at least if n_{UPA} is not large. In case of large wave numbers, long computation times cannot be avoided. More hints on how to chose the right n_{DOF} and n_{LFEM} will be given in the numerical tests in Section 9.

7 Computation of Efficiencies Using FEM/GFEM in CONICAL

The computation in the case of conical diffraction is completely the same as for the classical computation (cf. the result file enclosed in point 10.7). The same names of executables can

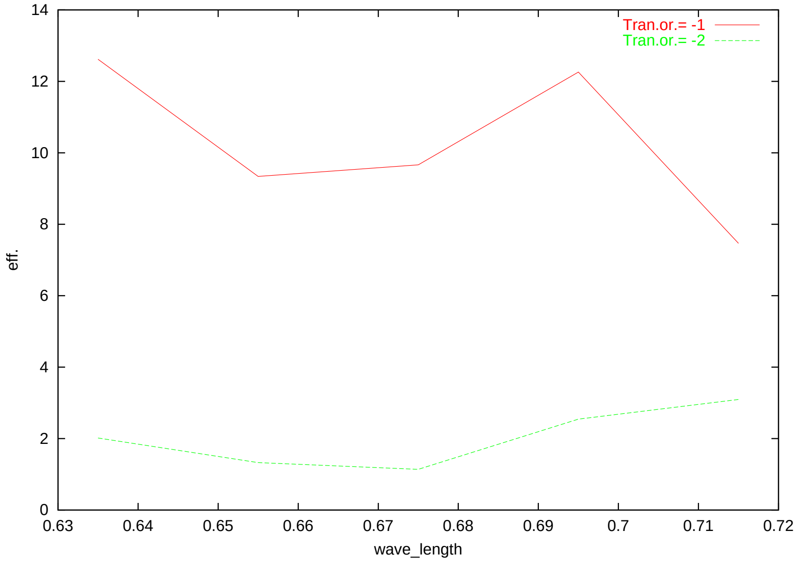

Figure 23: Efficiencies depending on wavelength. Output of PLOT_PS.

be used as in Section 2. The only differences are:

- All computations are to be done in CONICAL instead of CLASSICAL.
- Of course, the now used input file "name2.dat" (cf. the enclosed data file in 10.4) is longer than that of the classical case. To create such a file copy the example file "example.dat" in CONICAL not that in CLASSICAL.
- The input file for the generalized method "generalized.Dat" is the same but has a new name. Now it is called "conical.Dat".

8 Produce a Graph of the Efficiencies

If a result file "name3.res" is produced containing the values for several wave lengths or incident angles, then one can have a look at the two-dimensional graph of the efficiencies depending on the wave length or incident angle. Make sure to be in the subdirectory RESULTS, where the result file, e.g. "name3.res" exists. Then enter the command:

```
PLOT_DISPLAY name3 *1,*2,*3,*4
```

Here *1, *2, *3, and *3 stand for the efficiency/energy to be plot. E.g. setting *1 equal to R-1 means efficiency of reflected mode of order -1, setting *2 equal to T+0 means efficiency of transmitted mode of order 0, setting *3 equal to RE means total reflected energy, and setting *4 equal to TE means total transmitted energy. The number of efficiency/energy can vary between one and nine. Now a graph with the efficiencies/energies pops up on the screen (cf. Figure 23). To interrupt the presentation of the picture, press Enter/Return.

Alternatively, one can enter the command:

```
PLOT_PS name3 *1,*2,*3,*4
```

Everything is like in the previous case. However, instead of showing the graph on the screen, a ps file (cf. Figure 23) is produced. The name of the ps file will be printed on the screen.

9 Parameter Test for GFEM

Here we present the results of test computations to give an orientation for the choice of the parameters n_{DOF} and n_{LFEM}. Recall that n_{GP} is the number of grid points. For simplicity, we present results with $n_{\text{UPA}} = 1$, only. We consider a trapezoidal grating with basis angle of 60° and with one material which covers 60% of the period. The height of the trapezoid is 0.3 times length of period and the refractive index is 2.0. Moreover, we assume an additional layer which covers the whole period and which has a refractive index of 1.3 and a height of 0.05 times period. The substrate has a refractive index of 1.5 and the superstrate is air. The grating is illuminated in a classical TE scenario by light of wave length 635nm under an incidence angle of 65°. The length of one period is 1μm, 2μm, 4μm, 8μm, and 16μm. In other words, we have chosen the geometry generated by the code words

```
# Grating data:
  trapezoid 60 a 1 b c
```

where a is 0.6 times the length of the period, b is 0.3 times period, and c is 0.05 times period.

Now the accuracy and the best choice of parameters depend on the maximal relative wave number which is period times refractive index over wave length. We have checked the accuracy in percent. For instance, one percent accuracy means that: The absolute error of the percentage of energy reflected by the grating is less than 1% (the value itself is less than 100%). The absolute error of the percentage of light transmitted into the minus first order is less than 1% (the efficiency itself is less than 100%). The absolute error of real or imaginary part of the Rayleigh coefficient of the minus first reflected mode is less than 0.01 (the modulus of the coefficient itself is less than one).

The corresponding relative mesh size[14] h_r and the corresponding number of refinement levels (starting from 1 for the coarsest) necessary to achieve an accuracy up to 1%, 0.1%, and 0.01%, respectively, are presented in the Tables 1–3. Here we define the relative mesh size h_r of the grid by[15]

$$h_r = \frac{2\pi}{d} \frac{h}{n_{\text{DOF}} + 1}$$

with h the absolute mesh size of the triangulation and with d the length of the period. By the symbol k_r in the tables we denote the maximal relative wave number (length of period d times refractive index divided by wave length). The numerical methods are either FEM or GFEM($n_{\text{DOF}}, n_{\text{LFEM}}$), i.e. the GFEM with the parameters n_{DOF}, n_{LFEM}, and $n_{\text{UPA}} = 1$.

[14] Of course, the values presented in the Tables 1–3 are taken from a discrete set of test values, only. Indeed, we have computed only those relative mesh sizes which result from the halving the mesh size strategy realized in the code by switching to higher refinement levels.

[15] Note that the mesh size shown on the screen or in the result files "name.res" after calling the program `FEM_FULLINFO` and `GFEM_FULLINFO` are just the h_r.

Stars indicate that the accuracy is not reached due to the restricted main memory of the computer. The number of grid points n_{GP} is 67 for the first level, 75 for the second, 169 for the third, 600 for the fourth, 2 430 for the fifth, 8 858 for the sixth, 39 698 for the seventh, 159 140 for the eighth, and 637 914 for the ninth (cf. the computation time in (6.1)).

It is impossible to derive a general recommendation from the numbers in the Tables 1–3. We have indicated the necessary relative mesh size for the fastest[16] method with parameter $n_{\mathrm{UPA}} = 1$ by bold letters. However, the methods with doubled $[n_{\mathrm{DOF}} + 1]$ and $[n_{\mathrm{LFEM}} + 1]$ and doubled mesh size (one lower refinement level) require almost the same computation time and lead to the same accuracy. If $[n_{\mathrm{LFEM}} + 1]$ is large and the grid is of a higher refinement level, then the computing time can be reduced by first generating a preliminary grid with the doubled maximal mesh size and second applying an additional uniform refinement step of each triangle into four equal subtriangles. Recall that the trial functions for congruent triangles need to be computed only once. In other words, reducing the level by one and changing n_{UPA} from 1 to 2, turns GFEM into a competitive method even if $[n_{\mathrm{LFEM}} + 1]$ is large. Similarly, the level can be reduced by 2 or 3, and n_{UPA} can be set to 4 or 8. So GFEM with larger n_{DOF} and n_{LFEM} outperforms the GFEM indicated by bold letters.

For TM polarization and the same grating and light scenario, we get similar results. E.g., in the case of $k_r = 12.60$ (i.e. $d = 4$) and GFEM(3,31), we get an error of 1%, 0.1%, and 0.01% choosing the refinement levels 4, 6, and 6, respectively. For $k_r = 25.20$ (i.e. $d = 8$) and GFEM(7,127), we get an error of 1%, 0.1%, and 0.01% choosing the refinement levels 4, 4, and 5, respectively.

[16] Fastest method means the one with the smallest complexity estimate (6.1).

Method	k_r=3.15	k_r=6.30	k_r=12.60	k_r=25.20	k_r=50.39
	(d=1)	(d=2)	(d=4)	(d=8)	(d=16)
FEM	**0.401** (4)	0.089 (6)	0.024 (8)	$\star\star\star$	$\star\star\star$
GFEM(1,3)	0.401 (3)	0.201 (4)	0.044 (6)	**0.012** (8)	$\star\star\star$
GFEM(1,7)	0.401 (3)	0.201 (4)	0.044 (6)	0.023 (7)	$\star\star\star$
GFEM(1,15)	0.401 (3)	0.201 (4)	0.044 (6)	0.023 (7)	0.012 (8)
GFEM(3,7)	0.524 (1)	0.201 (3)	0.044 (5)	0.012 (7)	$\star\star\star$
GFEM(3,15)	0.524 (1)	0.324 (2)	0.100 (4)	0.022 (6)	**0.012** (7)
GFEM(3,31)	0.524 (1)	0.324 (2)	0.100 (4)	0.044 (5)	0.012 (7)
GFEM(3,63)	0.524 (1)	0.324 (2)	0.100 (4)	0.044 (5)	0.044 (5)
GFEM(7,15)	0.262 (1)	**0.262** (1)	0.050 (4)	0.011 (6)	$\star\star\star$
GFEM(7,63)	0.262 (1)	0.262 (1)	0.100 (3)	0.050 (4)	0.022 (5)
GFEM(7,127)	0.262 (1)	0.262 (1)	0.100 (3)	0.050 (4)	0.022 (5)
GFEM(7,255)		0.262 (1)	0.100 (3)	0.100 (3)	0.022 (5)
GFEM(15,31)		0.131 (1)	**0.131** (1)	0.011 (5)	$\star\star\star$
GFEM(15,127)		0.131 (1)	0.131 (1)	0.050 (3)	0.011 (5)
GFEM(15,255)			0.131 (1)	0.050 (3)	0.025 (4)
GFEM(15,511)				0.081 (2)	0.025 (4)

Table 1: Relative mesh size h_r (refinement levels) necessary to reach 1% accuracy.

Method	k_r=3.15	k_r=6.30	k_r=12.60	k_r=25.20	k_r=50.39
	(d=1)	(d=2)	(d=4)	(d=8)	(d=16)
FEM	0.047 (7)	0.024 (8)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(1,3)	0.088 (5)	0.044 (6)	**0.012** (8)	$\star\star\star$	$\star\star\star$
GFEM(1,7)	0.201 (4)	0.088 (5)	0.023 (7)	$\star\star\star$	$\star\star\star$
GFEM(1,15)	0.201 (4)	0.088 (5)	0.023 (7)	0.012 (8)	$\star\star\star$
GFEM(3,7)	0.100 (4)	0.044 (5)	0.012 (7)	$\star\star\star$	$\star\star\star$
GFEM(3,15)	0.201 (3)	0.100 (4)	0.022 (6)	**0.012** (7)	$\star\star\star$
GFEM(3,31)	**0.524** (1)	0.201 (3)	0.044 (5)	0.022 (6)	$\star\star\star$
GFEM(3,63)	0.524 (1)	0.201 (3)	0.044 (5)	0.022 (6)	**0.012** (7)
GFEM(7,15)	0.100 (3)	**0.100** (3)	0.011 (6)	$\star\star\star$	$\star\star\star$
GFEM(7,63)	0.100 (3)	0.262 (1)	0.050 (4)	0.011 (6)	$\star\star\star$
GFEM(7,127)	0.262 (1)	0.262 (1)	0.100 (3)	0.050 (4)	0.011 (6)
GFEM(7,255)		0.262 (1)	0.100 (3)	0.050 (4)	0.022 (5)
GFEM(15,31)		0.050 (3)	0.011 (5)	$\star\star\star$	$\star\star\star$
GFEM(15,127)		0.131 (1)	0.050 (3)	0.025 (4)	$\star\star\star$
GFEM(15,255)			0.081 (2)	0.050 (3)	0.011 (5)
GFEM(15,511)				0.050 (3)	0.025 (4)

Table 2: Relative mesh size h_r (refinement levels) necessary to reach 0.1% accuracy.

Method	k_r=3.15	k_r=6.30	k_r=12.60	k_r=25.20	k_r=50.39
	(d=1)	(d=2)	(d=4)	(d=8)	(d=16)
FEM	0.013 (9)	**0.013** (9)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(1,3)	0.023 (7)	0.023 (7)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(1,7)	0.044 (6)	0.044 (6)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(1,15)	0.088 (5)	0.044 (6)	**0.012** (8)	$\star\star\star$	$\star\star\star$
GFEM(3,7)	0.022 (6)	0.022 (6)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(3,15)	0.044 (5)	0.044 (5)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(3,31)	0.100 (4)	0.100 (4)	0.012 (7)	$\star\star\star$	$\star\star\star$
GFEM(3,63)	0.201 (3)	0.100 (4)	0.022 (6)	**0.012** (7)	$\star\star\star$
GFEM(7,15)	0.022 (5)	0.022 (5)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(7,63)	0.100 (3)	0.100 (3)	0.011 (6)	$\star\star\star$	$\star\star\star$
GFEM(7,127)	**0.262** (1)	0.162 (2)	0.022 (5)	0.011 (6)	$\star\star\star$
GFEM(7,255)			0.050 (4)	0.022 (5)	**0.011** (6)
GFEM(15,31)		0.025 (4)	$\star\star\star$	$\star\star\star$	$\star\star\star$
GFEM(15,127)		0.081 (2)	0.011 (5)	$\star\star\star$	$\star\star\star$
GFEM(15,255)			0.025 (4)	0.011 (5)	$\star\star\star$
GFEM(15,511)				0.025 (4)	0.011 (5)

Table 3: Relative mesh size h_r (refinement levels) necessary to reach 0.01% accuracy.

10 Enclosed Files

10.1 Geometry input file “example.inp”

```
#################################################################
#                                                               #
#                    #################                          #
#                    #  example.inp  #                          #
#                    #################                          #
#                                                               #
#         all lines beginning with # are comments               #
#                                                               #
#################################################################
#
#      - geometry input file for periodic grating
#      - located in directory ‘‘GEOMETRIES’’
#      - input file for ‘‘gen_polyx’’
#
#################################################################
# Name of the files without extensions ‘‘.inp’’.
#    Output files will have the same name but with
#    tags ‘‘.polyx’’ and ‘‘.sg’’.
# Name:
  example
#################################################################
# Comments.
#    Input must be ended by a ‘‘0’’ in an extra line.
#    These comments will appear in several output and
#    result files.
# Comments:
  This is a fantasy grid
  for the test of gen_polyx!
  0
#################################################################
# Number of materials:
  4
#################################################################
# Minimal angle of subdivision triangles:
  20.000000
#################################################################
# Upper bound for mesh size:
  0.500000
#################################################################
# Width of additional strip above and below.
#    Automatic choice of small width
#    if this value is 0.
# Width:
  0.200000
#################################################################
```

```
# Grid points.
#       #########################################################
#       points of triangulation which is part of the domain
#       for the FEM:
#
#       -> x-components between 0 and 1
#       -> triangles should be disjoint
#       -> union of triangles should be a simply connected domain
#       -> union of triangles should connect the lines x=0,x=1
#       -> union of triangles should be bounded by two vertical
#          lines and by two piecewise linear functions in x
#
#       first add the nodes of all the triangles
#       later give the triangles by the indices of their nodes
#       ########################################################
#    Each point in a separate line.
#    Scaled to period 1.
#    Input ended by ‘‘-1. -1.’’.
# Grid points:
  0.000000 0.800000
  0.500000 0.800000
  0.000000 0.400000
  0.250000 0.400000
  1.000000 0.400000
  0.750000 0.200000
  1.000000 0.200000
  0.000000 0.000000
  0.250000 -0.200000
  1.000000 -0.200000
  0.000000 -0.600000
  1.000000 -0.800000
  -1. -1.
#################################################################
# Triangles.
#    Each given in a separate line by 5 parameters,
#    namely by index of first point, by index of
#    second point, by index of third point, by index
#    of material, and by additional factor for maximal
#    mesh size of partition inside the triangle.
#    Input ended by ‘‘-1 -1 -1 -1 -1.’’.
# Triangles:
  1 3 4 2 1.000000
  4 6 2 2 1.000000
  6 7 5 2 1.000000
  3 8 4 2 1.000000
  8 9 4 2 0.300000
  4 9 6 2 1.000000
  6 10 7 3 1.000000
  8 11 9 3 1.000000
  9 12 6 3 1.000000
  6 12 10 3 1.000000
  -1 -1 -1 -1 -1.

#################################################################
```

```
# End
#################################################################
```

10.2 Data file “example.dat” for CLASSICAL

```
###################################################################
#                                                                 #
#                     #################                           #
#                     #  example.dat  #                           #
#                     #################                           #
#                                                                 #
#           all lines beginning with # are comments!              #
#                                                                 #
###################################################################
#
#         - input file for ‘‘FEM/GFEM’’
#         - located in directory ‘‘CLASSICAL’’
#
###################################################################
# Name of the output file.
#    The tag ‘‘.res’’ will be added.
#    File will be written into directory ‘‘RESULTS’’.
#    (Alternatively, a path for the location of the file
#     can be added before the name. This must contain at
#     least one slash ’/’. E.g. for a file ‘‘name.res’’
#     in the current working directory write ‘‘  ./name’’)
# Name:
  example
###################################################################
# Should there be an additional output file in the old style of
# DIPOG-1.3.
#    Add ‘‘  no’’ if not needed.
#    Add ‘‘  yes’’ if needed. The name will be the same as
#    the standard output file given above but with the tag
#    ‘‘.erg’’ instead of ‘‘.res’’.
#    Add ‘‘  phaseshifts’’ if no additional output is needed
#    but if phase shifts are preferred instead of Rayleigh
#    coefficients.
# yes or no or phaseshifts:
  yes
###################################################################
# Number of coatings over the grating (N_co_ov).
#    The grating cross section consists of
#    a rectangular area parallel to the axes.
#    This inhomogeneous part is determined
#    by a triangular grid and can have already a few
#    layers of coatings involved. Beneath and above
#    this rectangular structure, there might be additional
#    coated layers of rectangular shape. These kind of
#    layers are called coatings over the grating and
#    coatings beneath the grating, respectively.
# Number of coatings:
  2
####################################################################
# Widths of coatings in micro m.
#    Needed only if N_co_ov >0.
#    Else no number no line.
# Widths:
  0.5
  0.2
####################################################################
# Number of coatings beneath the grating (N_co_be):
  3
####################################################################
# Widths of coatings in micro m.
#    Needed only if N_co_be >0.
#    Else no number no line.
# Widths:
  0.2
  0.3
  0.2
####################################################################
# Wave length in micro m (lambda).
#    Either add a single value e.g. ‘‘.63’’.
#    Either add more values by e.g.
#      ‘‘  V
#          5
#          .63
#          .64
#          .65
#          .69
#          .70 ’’.
#      The last means that computation is to be done for
#      the wave lengths from the Vector of length 5:
#      ‘‘.63’’, ‘‘.64’’,‘‘.65’’,‘‘.69’’, and ‘‘.70’’.
#    Or add e.g. ‘‘  I .63 .73 .02’’.
#      The last means that computation is to be done for
#      the wave lengths ‘‘.63+i*.02’’ with i=0,1,2,... and with
#      wave length ‘‘.63+i*.02’’ less than ‘‘.73’’.
#
# Wave length:
  .635
####################################################################
# Temperature in degrees Celsius.
#    From 0 to 400.
#    For room temperature set to 20.
#    Will be ignored for explicitly
#    given refractive indices.
# Temperature:
  20.
####################################################################
# Optical index (refractive index) of cover material.
#    This is c times square root of mu times epsilon.
#    This could be complex like ‘‘4.298 +i 0.073’’ for
#    Si with wave length 500nm.
#    This could be also given by the name of a material
#    like: Air Ag Al Au CsBr Cu InP MgF2 NaCl PMMA PSKL
#          SF5 Si TlBr TlCl Cr ZnS Ge Si1.0 - Si2.0
#          TiO2r Quarz AddOn
#    This could be a value interpolated from a user
#    defined table, determined by the name of the file
#    (file is to be located in the current directory,
#    name of file must begin with letter ‘‘u’’ and may
#    consist of no more than five letters like e.g. user,
#    the file consists of lines each with three real
#    numbers, first: wave length in micro meter, second:
#    the real part of the corresponding optical index,
#    third: the imaginary part of the corresponding index).
# Optical index:
  1.0 +i .0
###################################################################
# Optical indices of the materials of the upper coatings.
#    Needed only if N_co_ov >0.
#    Else no number no line.
# Optical indices:
  1.1 +i .0
  1.2 +i .0
###################################################################
# Optical indices of the materials of the lower coatings.
#    Needed only if N_co_be >0.
#    Else no number no line.
# Optical indices:
  2.3 +i .0
  2.2 +i .0
  2.1 +i .0
###################################################################
# Optical index of substrate material.
  2.0 +i .0
###################################################################
# Angle of incident wave in degrees (theta).
#    Either add a single value e.g. ‘‘45.’’.
#    Either add more values by e.g.
#      ‘‘  V
#          5
#          63.
#          64.
#          65.
#          69.
#          70. ’’.
#      The last means that computation is to be done for
#      the angles from the Vector of length 5:
#      ‘‘63.’’, ‘‘64.’’,‘‘65.’’,‘‘69.’’, and ‘‘70.’’.
#    Or add e.g. ‘‘  I 45 56 2’’.
#      The last means that computation is to be done for
#      the angles ‘‘45+i*2’’ with i=0,1,2,... and with
#      angle ‘‘45+i*2’’ less than ‘‘56’’.
#    Note that either the wave length or the angle of
#    incident wave must be single valued.
# Angle of incident wave:
  65
####################################################################
# Type of polarization.
#    Either TE, TM or TE/TM.
# Type:
  TM
####################################################################
# Length factor of additional shift of grating geometry.
#    This is shift into the x-direction, i.e. the
#    direction of the period to the right.
#    This is length of shift relative to period, i.e.
#    the grating structure given by subsequent input
#    will be shifted by factor times the period given
#    in subsequent input.
#    However, only the Rayleigh numbers and efficiencies
#    will be computed according to the shift. The field
#    vectors in the plots are drawn without shift, and
#    the graphics of the executable with tag ‘‘_CHECK’’
#    is drawn without shift!
#    Must be a real number between 0 and 1.
# Length:
  0.
####################################################################
# Stretching factor for grating in y-direction.
#    Must be a positive real number.
# Length:
  1.
####################################################################
# Length of additional shift of grating geometry in micro m.
#    This is shift into the y-direction, i.e. the
#    direction perpendicular to the grating surface
#    pointing into the cover material.
#    Must be a real number.
# Length:
  0.
####################################################################
# Period of grating in micro m:
  1.
####################################################################
# Grating data.
#    Either this should be e.g. ‘‘name1’’ if ‘‘name1.inp’’
#       is the input file with the geometry data in sub-
#       directory ‘‘GEOMETRIES’’.
#       (Alternatively, a path for the location of the file
#        can be added before the name. This must contain at
#        least one slash ’/’. E.g. for a file in the current
#        working directory write ‘‘  ./name1’’)
#    Or this could be a stack of profiles given by the
#       code word stack and many more lines (cf. Userguide)
#    Or this could be
#       e.g. ‘‘  echellea R 0.3 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity, right interior angle > 45 degrees)
#          with depth of 0.3 micro meter and with coated
#          layers of height 0.03 micro meter resp. 0.04 micro
#          meter over the first resp. second part of the
#          grating (measured in direction perpendicular to
#          echelle profile, height greater or equal to zero)
#       e.g. ‘‘  echellea L 0.3 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity, left interior angle > 45 degrees)
#          with parameters like above
#       e.g. ‘‘  echellea A 60 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity) with left interior angle Alpha=60
#          degrees (i.e. depth = period times sin(Alpha)
#          times cos(Alpha)) and other parameters like above
#       e.g. ‘‘  echelleb 60. 0.05’’
#       -> ECHELLE GRATING TYPE B (right-angled triangle
#          with one of the legs parallel to the direction
#          of the periodicity) with angle 60 (angle enclosed
#          by hypotenuse and by the leg parallel to the
#          period) and with a coated layer of height 0.05
#          micro meter (measured in direction perpendicular
#          to echelle profile, height greater or equal to
#          zero)
#       e.g. ‘‘  trapezoid 60. 0.6 3 0.2 0.1 0.1 0.05’’
#       -> TRAPEZOIDAL GRATING (trapezoid with the basis
#          parallel to the direction of the periodicity)
#          with angle of 60 degrees (angle enclosed by
#          basis and the sides) with a base of length 0.6
#          micro meter consisting of 3 material layers of
#          heights 0.2, 0.1, and 0.1 micro meter,
#          respectively, and with a coated layer of height
#          0.05 micro meter (greater or equal to zero)
#       e.g. ‘‘  lAmellar 3 4
#                0.2 0.6
#               -0.2 1.0
#                0.   0.5   0.7
#                0.0  0.50  0.90
#                0.00 0.500 0.900  ’’
#       -> LAMELLAR GRATING (rectangular grating consisting
#          of several materials placed in rectangular sub-
#          domains) with 3 columns each divided into 4
#          rectangular layers, first column with x coordinate
#          in 0<x<0.2 (given in micro meter), second column with 0.2<x<0.6,
#          third columnn with 0.6<x<period (period given above),
#          whole grating with y coordinate s.t. -0.2<y<1.0,
#          first column: first layer with -0.2<y<0., second with
#          0.<y<0.5, third with 0.5<y<0.7 and fourth with 0.7<y<1,
#          second column: first layer with -0.2<y<0.0, second with
#          0.0<y<0.50, third with 0.50<y<0.90 and fourth with 0.90<y<1.,
#          third column: first layer with -0.2<y<0.00, second with
#          0.00<y<0.500, third with 0.500<y<0.900 and fourth with
#          0.900<y<1.
#       e.g. ‘‘  lAmellar 1 1
#                0.2 0.8     ’’
#       -> SIMPLE LAYER (special case of lamellar grating) with
#          layer material s.t. y-coordinate satisfies 0.2<y<0.8
#          (given in micro meter).
#       e.g. ‘‘  polygon file1’’
#       -> GRATING DETERMINED BY A POLYGONAL LINE defined
#          by the data in the file with name ‘‘../GEOMETRIES/file1’’
#          (in ‘‘../GEOMETRIES/file1’’: in each line beginning without ‘#’
#          there should be the x- and y-coordinate of one
#          of the consecutive corner points, first point
#          with x-coordinate 0, last point with x-coordinate
#          1, same y-coordinate for first and last point,
#          all x-coordinates between 0 and 1, at least two
#          different y-coordinates, last line should be
#          ‘‘End’’)
#       e.g. ‘‘  polygon2 file1 file2’’
#       -> COATED GRATING DETERMINED BY POLYGONAL LINES,
#          i.e. grating profile line is defined by the data
#          in the file with name ‘‘../GEOMETRIES/file1’’
#          (in ‘‘../GEOMETRIES/file1’’:
#          in each line beginning without ‘#’ there should
#          be the x- and y-coordinate of one of the
#          consecutive corner points, first point with
#          x-coordinate 0, last point with x-coordinate
#          1, same y-coordinate for first and last point,
#          all x-coordinates between 0 and 1, at least two
#          different y-coordinates, last line should be
#          ‘‘End’’) and the coated layer is enclosed between
#          the polygonal line of ‘‘../GEOMETRIES/file1’’
#          and the polygonal line of the file with name
#          ‘‘../GEOMETRIES/file2’’
#          (in ‘‘../GEOMETRIES/file2’’:
#          in each line beginning without ‘#’ there should be
#          the x- and y-coordinate of one of the consecutive
#          corner points, first and last point must be corner
#          of first polygon, second polygon must be on left-
#          hand side of first, one to one correspondence of
#          the corners on the two polygons between first and
#          last point of second polygon, quadrilateral between
#          corresponding segments on the left of first
#          polygon, these quadrilaterals must be disjoint, last
#          line should be ‘‘End’’)
#       e.g. ‘‘  profile’’
#       -> GRATING DETERMINED BY A SMOOTH PARAMETRIC CURVE,
#          i.e. grating determined by profile line given as
#          {(fctx(t),fcty(t)):0<=t<=1}, where the functions
#          ‘‘t|->fctx(t)’’ and ‘‘t|->fcty(t)’’ are defined
#          by the ‘‘c’’-code of the file ‘‘../GEOMETRIES/profile.c’’
#       e.g. ‘‘  profile_par 2 3
#                1
#                0
#                1.5
#                0.2
#                0.3             ’’
#       -> GRATING DETERMINED BY A SMOOTH PARAMETRIC CURVE,
#          WITH PARAMETERS,
#          i.e. grating determined by profile line given as
#          {(fctx(t),fcty(t)):0<=t<=1}, where the functions
#          ‘‘t|->fctx(t)’’ and ‘‘t|->fcty(t)’’ are defined
#          by the ‘‘c’’-code of ‘‘../GEOMETRIES/profile_par.c’’;
#          the last code uses 2 integer parameters and 3
#          real parameters named IPARaM1, IPARaM2, RPARaM1,
#          RPARaM2, RPARaM3;
#          the integer parameters take the values 1 and 0
#          following the first line of the calling sequence
#          and the real parameters take the values 1.5, 0.2,
#          and 0.3 following the integer parameter values
#          (Any number of parameters is possible for a
#          corresponding file ‘‘../GEOMETRIES/profile_par.c’’.)
#       e.g. ‘‘  profile 0.125*sin(2.*M_PI*t)’’
#       -> GRATING DETERMINED BY A SIMPLE SMOOTH FUNCTION,
#          i.e. grating determined by sine profile line given
#          as {(t,fcty(t)):0<=t<=1}, where the function
#          ‘‘t|->fcty(t)’’ is defined by the ‘‘c’’-code
#          fcty(t)=0.125*sin(2.*M_PI*t).
#          (do not use any ‘‘blank’’/‘‘space’’ in the c-code)
#       e.g. ‘‘  profile 0.5+0.5*cos(M_PI*(1.-t)) 0.25*sin(M_PI*t)’’
#       -> GRATING DETERMINED BY A SIMPLE SMOOTH PARAMETRIC CURVE,
#          i.e., grating determined by ellipsoidal profile line
#          given as {(fctx(t),fcty(t)):0<=t<=1}, where the
#          functions ‘‘t|->fcty(t)’’ and ‘‘t|->fcty(t)’’ are
#          defined by the ‘‘c’’-codes
#          fctx(t)=0.5+0.5*cos(M_PI*(1.-t)) and
#          fcty(t)=0.25*sin(M_PI*t), respectively
#          (do not use any ‘‘blank’’/‘‘space’’ in the c-codes)
#       e.g. ‘‘  profiles’’
#       -> GRATING DETERMINED BY SMOOTH PARAMETRIC CURVES,
#          i.e. grating determined by profile lines given as
#          {(fctx(j,t),fcty(j,t)):0<=t<=1}, j=1,...,n=nmb_curves,
#          where the functions ‘‘t|->fctx(j,t)’’ and
#          ‘‘t|->fcty(j,t)’’ are defined by the ‘‘c’’-code
#          of the file ‘‘../GEOMETRIES/profiles.c’’
#       e.g. ‘‘  profiles_par 1 2
#                3
#                0.5
#                0.50            ’’
#       -> GRATING DETERMINED BY SMOOTH PARAMETRIC CURVES,
#          WITH PARAMETERS,
#          i.e. grating determined by profile lines given as
#          {(fctx(j,t),fcty(j,t)):0<=t<=1}, j=1,...,n=nmb_curves,
#          where the functions ‘‘t|->fctx(j,t)’’ and
#          ‘‘t|->fcty(j,t)’’ are defined by the ‘‘c’’-code
#          of the file ‘‘../GEOMETRIES/profiles_par.c’’;
#          the last code uses 1 integer parameter and 2
#          real parameters named IPARaM1, RPARaM1, RPARaM2;
#          the integer parameter takes the value 3
#          following the first line of the calling sequence
#          and the real parameters take the values 0.5 and 0.50
#          following the integer parameter values
#          (Any number of parameters is possible for a
#          corresponding file ‘‘../GEOMETRIES/profiles_par.c’’.)
#       e.g. ‘‘  pin’’
#       -> PIN GRATING DETERMINED BY PARAMETRIC CURVE,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(t),fcty(t)):0<=t<=1}.
#          Here {(fctx(t),fcty(t)):0<=t<=1} is a simple open
#          arc connecting (fctx(0),fcty(0))=(xmin,0) with
#          (fctx(1),fcty(1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(t)<1, 0<t<1, and
#          such that 0<fcty(t), 0<t<1. The functions fctx, fcty
#          and the parameter xmin are defined by the code in
#          ‘‘../GEOMETRIES/pin.c’’.
#       e.g. ‘‘  cpin’’
#       -> COATED PIN GRATING DETERMINED BY TWO PARAMETRIC CURVES,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(1,t),fcty(1,t)):0<=t<=1}.
#          Here {(fctx(1,t),fcty(1,t)):0<=t<=1} is a simple open
#          arc connecting (fctx(1,0),fcty(1,0))=(xmin,0) with
#          (fctx(1,1),fcty(1,1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(1,t)<1, 0<t<1, and
#          such that 0<fcty(1,t), 0<t<1. Additionaly, a coating
#          layer is attached located between the first curve
#          {(fctx(1,t),fcty(1,t)):0<=t<=1} and a second curve
#          {(fctx(2,t),fcty(2,t)):0<=t<=1}. The last connects the
#          point (fctx(1,arg1),fcty(1,arg1))=(fctx(2,0),fcty(2,0))
#          with (fctx(1,arg2),fcty(1,arg2))=(fctx(2,1),fcty(2,1)).
#          Moreover, {(fctx(2,t),fcty(2,t)):0<=t<=1} is a simple open
#          arc above {(fctx(1,t),fcty(1,t)):0<=t<=1} such that
#          0<fctx(2,t)<1, 0<t<1. The functions fctx(1,.), fctx(2,.),
#          fcty(1,.), and fcty(2,.) and the parameters arg1, arg2,
#          and xmin are defined by the code of the file
#          ‘‘../GEOMETRIES/cpin.c’’.
#       e.g. ‘‘  cpin2’’
#       -> COATED PIN GRATING DETERMINED BY TWO PARAMETRIC CURVES
#          TYPE 2,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(1,t),fcty(1,t)):0<=t<=1}.
#          Here {(fctx(1,t),fcty(1,t)):0<=t<=1} is a simple open
#          arc connecting (fctx(1,0),fcty(1,0))=(xmin,0) with
#          (fctx(1,1),fcty(1,1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(1,t)<1, 0<t<1, and
#          such that 0<fcty(1,t), 0<t<1. Additionaly, a coating
#          layer is attached located between the first curve
#          {(fctx(1,t),fcty(1,t)):0<=t<=1} and a second curve
#          {(fctx(2,t),fcty(2,t)):0<=t<=1}. The last connects the
#          point (x1,0)=(fctx(2,0),fcty(2,0)) with (x2,0)=
#          (fctx(2,1),fcty(2,1)) with 0<x1<xmin<1-xmin<x2. Moreover,
#          {(fctx(2,t),fcty(2,t)):0<=t<=1} is a simple open
#          arc above {(fctx(1,t),fcty(1,t)):0<=t<=1} such that
#          0<fctx(2,t)<1, 0<t<1. The functions fctx(1,.), fctx(2,.),
#          fcty(1,.), and fcty(2,.) and the parameter xmin are
#          defined by the code of the file ‘‘../GEOMETRIES/cpin2.c’’.
#       e.g. ‘‘  stack 3
#                profile t /##/ 0.2*sin(2.*M_PI*t)
#                2.
#                profile 0.2*sin(2.*M_PI*t)
#                1.
#                profile t /##/ 0.
#                0.                   ’’
#       -> STACK GRATING,
#          i.e. a stack of 3 profile curves shifted by 2, 1, 0
#          micro meter in vertical direction. For more details, see
#          the description in the USERGUIDE.
# Grating data:
  example
####################################################################
# Number of different grating materials (N_mat).
#    This includes the material of substrate and cover material.
#    For example,
#       if Grating data is  ‘‘  name1’’
#       -> Number of materials given in file ‘‘name1.inp’’
#       if Grating data is  ‘‘  echellea ...’’
#       -> N_mat = 3 with coating height >0
#          N_mat = 2 with coating height =0
#       if Grating data is  ‘‘  echelleb ...’’
#       -> N_mat = 3 with coating height >0
#          N_mat = 2 with coating height =0
#       if Grating data is  ‘‘  trapezoid ...’’
#       -> N_mat = number of material layers +3
#                  for coating height >0
#          N_mat = number of material layers +2
#                  for coating height =0
#       if Grating data is  ‘‘  lAmellar k m
#                            ...  ’’
#       -> N_mat = k times m plus 2
#       if Grating data is  ‘‘  polygon file1’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  polygon2 file1 file2’’
#       -> N_mat = 3
#       if Grating data is  ‘‘  profile’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  profile 2 3
#                            ...        ’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  profile 0.125*sin(2.*M_PI*t)’’
#       -> N_mat = 2
#       if Grating data is
#       ‘‘  profile 0.5+0.5*cos(M_PI*(1.-t)) 0.25*sin(M_PI*t)’’
#       -> N_mat = 2
#       if Grating data is ‘‘  profiles’’
#       -> N_mat = n+1 with n=nmb_curves from
#                  the file ‘‘../GEOMETRIES/profiles.c’’
#       if Grating data is ‘‘  profiles ... ’’
#       -> N_mat = n+1 with n=nmb_curves from
#                  the file ‘‘../GEOMETRIES/profiles_par.c’’
#       if Grating data is ‘‘  pin ’’
#       -> N_mat = 3
#       if Grating data is ‘‘  cpin ’’
#       -> N_mat = 4
#       if Grating data is ‘‘  cpin2 ’’
#       -> N_mat = 4
#       if Grating data is ‘‘  stack k’’
#       -> N_mat = k+1
# Number of materials:
  4
####################################################################
# Optical indices of grating materials.
#    This is c times square root of mu times epsilon.
#    If meaningful, then the refractive indices should be ordered
#    according to the location from above to below.
#    If an input file ‘‘name1.inp’’ is used, then the optical index
#    of a subdomain with the material index j is just the j-th
#    optical index following below.
#    If grating is ‘‘lAmellar ...’’, then first material is cover
#    material, last material is substrate, and all other materials
#    are ordered from left to right and inside the columns from
#    below to above.
#    For technical reasons, the index of the material adjacent to
#    the upper line of the grating structure must coincide with
#    that of the material in the adjacent upper coated layer resp.
#    in the adjacent superstrate. Similarly the index of the
#    materials adjacent to the lower line of the grating structure
#    must coincide with that of the material in the adjacent lower
#    coated layer resp. in the adjacent substrate.
#    N_mat numbers are needed.
# Optical indices:
  1.2 +i .0
  1.5 +i .0
  1.7 +i .0
  2.3 +i .0
####################################################################
# Number of levels (Lev).
#    In each refinement step the step size of
#    the mesh is halved.
#    Number of refinement steps is Lev.
#    (Alternatively, one can prescribe an bound for the maximal
#     error of the efficiencies. E.g. the input ‘‘e 1.’’ means
#     that the level for the computation is the smallest positive
#     integer such that all efficiencies are computed with an
#     estimated error less than 1 per cent.)
# Number:
  3
####################################################################
```

```
# End
###################################################################
```

10.3 Data file “generalized.Dat” for CLASSICAL, same as “conical.Dat” in CONICAL

```
############################################################
#                                                          #
#                #####################                     #
#                #  generalized.Dat  #                     #
#                #####################                     #
#                                                          #
#        all lines beginning with # are comments           #
#                                                          #
############################################################
#
#         - input file for ‘‘GFEM’’
#         - located in directory ‘‘CLASSICAL’’
#         - contains constants for numerical
#           method in program ‘‘GFEM’’
#
############################################################
#
#  Recommendation for n_DOF and n_LFEM:
#  ------------------------------------
#
#       a) mild accuracy requirements
#          and wave numbers not too large:
#
#          n_DOF=1,3,7  n_LFEM=2*n_DOF+1
#
#       b) challenging accuracy requirements
#          or large wave numbers:
#
#          n_DOF=3  with  n_LFEM=31   or
#          n_DOF=7  with  n_LFEM=127  or
#          n_DOF=15 with  n_LFEM=512
#
#  Recommendation for n_UPA:
#  -------------------------
#
#       Take the first level l_0 (cf. the last input in
#       ‘‘name.dat’’ which is the upper bound for all
#       levels to be computed, and cf. the levels
#       indicated in the result files ‘‘name2.res’’)
#       such that the next level results in about four
#       times the number of grid points. Then, if you
#       wish to compute on level l_0+l_1, set the
#       maximum level of computation (last input in
#       ‘‘name.dat’’) to l_0 and choose n_UPA as 2 to
#       the power l_1.
#
############################################################
# n_DOF.
#    Additional degrees of freedom on each triangle side.
#    Indeed, trial functions on each subdivision triangle
#    are approximate solutions of pde s.t. restriction to
#    triangle sides coincides with Lagrange interpolation
#    polynomials on triangle side (Dirichlet’s problem).
#    Here interpolation is taken over uniform grid with
#    [n_DOF+2] interpolation knots.
#    Value should satisfy 0<=n_DOF<100.
# value:
  3
############################################################
# n_LFEM.
#    Approximate solution determined by FEM over subdivision
#    triangle, where additional uniform FEM partition on
#    each small triangle is chosen such that the step size
#    is side length divided by [n_LFEM+1].
#    If n_LFEM=1, n_DOF=0: conventional FEM method.
#    If n_DOF=n_LFEM: conventional FEM method with
#                     elimination of interior nodes of grid
#                     triangle, i.e. real mesh size is mesh
#                     size shown in result file divided by
#                     [n_DOF+1].
#    If n_DOF<n_LFEM: method resembles p=method or PUM.
#    Value should satisfy 1<=n_LFEM<2048 and
#    [n_LFEM+1] must be a multiple of [n_DOF+1].
# value:
  63
############################################################
# n_UPA.
#    This is for additional uniform partition of all primary
#    grid triangles into n_UPA*n_UPA equal subdomains,
#    i.e. original side of grid triangle is split
#    into n_UPA sides of uniform partition subtriangles.
#    Value should satisfy 1<=n_UPA<=128.
# value:
  1
############################################################
#
# that’s it
#
############################################################
```

10.4 Data file “example.dat” for CONICAL

```
######################################################################
#                                                                    #
#                     #################                              #
#                     #  example.dat  #                              #
#                     #################                              #
#                                                                    #
#         all lines beginning with ‘‘#’’ are comments!               #
#                                                                    #
######################################################################
#
#         - input file for ‘‘FEM/GFEM’’
#         - located in directory ‘‘CONICAL’’
#
######################################################################
# Name of the output file.
#    The tag ‘‘.res’’ will be added.
#    The file will be written in the ‘‘RESULTS’’ directory.
#    (Alternatively, a path for the location of the file
#     can be added before the name. This must contain at
#     least one slash ’/’. E.g. for a file ‘‘name.res’’
#     in the current working directory write ‘‘  ./name’’)
# Name:
  example
###################################################################
# Should there be an additional output file in the old style of
# DIPOG-1.3.
#    Add ‘‘  no’’ if not needed.
#    Add ‘‘  yes’’ if needed. The name will be the same as
#    the standard output file given above but with the tag
#    ‘‘.erg’’ instead of ‘‘.res’’.
# yes or no:
  yes
######################################################################
# Number of coating layers over the grating (N_co_ov).
#    The grating cross section consists of a rectangular area
#    parallel to the axes. This inhomogeneous part is determined
#    by a triangular grid and can have already a few
#    layers of coatings involved. Beneath and above
#    this rectangular structure, there might be additional
#    coated layers of rectangular shape. These kind of
#    layers are called coating layers over the grating and
#    coating layers beneath the grating, respectively.
# Number:
  2
######################################################################
# Widths of coating layers in micro meter.
#    N_co_ov entries. Needed only if N_co_ov >0.
#    Else no entry and no line.
# Widths:
  0.05
  0.03
######################################################################
# Number of coating layers beneath the grating (N_co_be):
  1
######################################################################
# Widths of coating layers in micro meter.
#    N_co_be entries. Needed only if N_co_be >0.
#    Else no entry and no line.
# Widths:
  0.05
######################################################################
# Wave length in micro meter (lambda).
#    Either add a single value e.g. ‘‘.63’’.
#    Either add more values by e.g.
#      ‘‘  V
#          5
#          .63
#          .64
#          .65
#          .69
#          .70  ’’.
#      The last means that computation is to be done for
#      the wave lengths from the Vector of length 5:
#      ‘‘.63’’, ‘‘.64’’,‘‘.65’’,‘‘.69’’, and ‘‘.70’’.
#    Or add e.g.
#      ‘‘  I .63 .73 .02’’.
#      The last means that computation is to be done for
#      the wave lengths ‘‘.63+i*.02’’ with i=0,1,2,... and with
#      wave length ‘‘.63+i*.02’’ less than ‘‘.73’’.
# Wave length:
  .635
######################################################################
# Temperature in degrees Celsius from 0 to 400.
#    20. for room temperature!
#    Must be set to any fixed number.
#    Will be ignored if optical indices are given explicitly.
# Temperature:
  20.
######################################################################
# Optical index (refractive index) of cover material.
#    This is c times square root of mu times epsilon.
#    This could be complex like e.g. ‘‘4.298 +i 0.073’’ for
#    Si with wave length 500nm.
#    This could be also given by the name of a material
#    like: Air Ag Al Au CsBr Cu InP MgF2 NaCl PMMA PSKL
#          SF5 Si TlBr TlCl Cr ZnS Ge Si1.0 - Si2.0
#          TiO2r Quarz AddOn
#    This could be a value interpolated from a user
#    defined table, determined by the name of the file
#    (file is to be located in the current directory,
#    name of file must begin with letter ‘‘u’’ and may
#    consist of no more than five letters like e.g. user,
#    the file consists of lines each with three real
#    numbers, first: wave length in micro meter, second:
```

```
#    the real part of the corresponding optical index,
#    third: the imaginary part of the corresponding index).
# Optical index:
  Air
######################################################################
# Optical indices of the materials of the upper coating layers.
#    This is c times square root of mu times epsilon.
#    N_co_ov entries. Needed only if N_co_ov >0.
#    Else no entry and no line.
# Optical indices:
  1.2
  1.3
######################################################################
# Optical indices of the materials of the lower coating layers.
#    This is c times square root of mu times epsilon.
#    N_co_be entries. Needed only if N_co_be >0.
#    Else no entry and no line.
# Optical indices:
  1.6
######################################################################
# Optical index of substrate material.
#    This is c times square root of mu times epsilon.
# Optical index:
  1.5 +i 0.
######################################################################
# Type of output results.
#    Either ‘‘TE/TM’’: results in terms of TE and TM part of Wave.
#    Either ‘‘Jones’’: results in terms of Jones vector
#                      representation.
#    Or ‘‘3.Com’’:     results in terms of the component in the z
#                      axis, that is in the direction of the grooves.
#    For more details cf. Section 2.3 in USERGUIDE.ps.
# Type:
  3.Com
######################################################################
# Type of polarization and coordinate system for incoming wave vector.
#    Either ‘‘TE’’: means that incident electric field is perpendicular
#                   to wave vector and to normal of grating plane
#                   (plane of grating grooves) and
#                   incoming wave vector is presented in xz system as
#                   (sin theta cos phi, -cos theta, sin theta sin phi).
#    Either ‘‘TM’’: means that incident magnetic field is perpendicular
#                   to wave vector and to normal of grating plane and
#                   incoming wave vector is presented in xz system as
#                   (sin theta cos phi, -cos theta, sin theta sin phi).
#    Either ‘‘TE/TM’’:
#                   means TE and TM, two calculations.
#    Either ‘‘TP’’: means polarized electro-magnetic field and
#                   incoming wave vector is presented in xz system as
#                   (sin theta cos phi, -cos theta, sin theta sin phi).
#    Or ‘‘pol’’:    means polarized electro-magnetic field and
#                   incoming wave vector is presented in xy system as
#                   (sin theta cos phi, -cos theta cos phi, sin phi).
# Type:
  TM
######################################################################
# Parameter of polarization.
#    If type of polarization is ‘‘pol’’ or ‘‘TP’’, then this is
#    the angle (in degrees) between x axis (axis in plane of
#    grating grooves which is perpendicular to grooves)
#    and projection of electric field vector onto x-z plane of
#    grating grooves.
#    Needed only if polarization is of type ‘‘pol’’ or ‘‘TP’’.
#    Else no entry and no line.
# Parameter:
######################################################################
# Angle of incident wave in degrees (theta).
#    If type of polarization is ‘‘pol’’,
#    then the incident light beam takes the direction
#    (sin theta cos phi, -cos theta cos phi, sin phi)
#    with the restriction -90 < phi,theta < 90.
#    If type of polarization is ‘‘TE’’/‘‘TM’’/‘‘TP’’,
#    then the incident light beam takes the direction
#    (sin theta cos phi, -cos theta, sin theta sin phi)
#    with the restriction 0 < theta < 90.
#    Either add a single value e.g. ‘‘45.’’.
#    Either add more values by e.g.
#      ‘‘  V
#          5
#          63.
#          64.
#          65.
#          69.
#          70. ’’.
#      The last means that computation is to be done for
#      the angles from the Vector of length 5:
#      ‘‘63.’’, ‘‘64.’’,‘‘65.’’,‘‘69.’’, and ‘‘70.’’.
#    Or add e.g.
#      ‘‘  I 45 56 2’’.
#      The last means that computation is to be done for
#      the angles ‘‘45+i*2’’ with i=0,1,2,... and with
#      angle ‘‘45+i*2’’ less than ‘‘56’’.
#    Note that either the wave length or the angle of
#    incident wave theta must be single valued.
# Angle:
  30.
######################################################################
# Angle of incident wave in degrees (phi).
#    If type of polarization is ‘‘pol’’,
#    then the incident light beam takes the direction
#    (sin theta cos phi, -cos theta cos phi, sin phi)
#    with the restriction -90 < phi,theta < 90.
#    If type of polarization is ‘‘TE’’/‘‘TM’’/‘‘TP’’,
#    then the incident light beam takes the direction
#    (sin theta cos phi, -cos theta, sin theta sin phi)
#    with the restriction 0 < theta < 90.
#    Either add a single value e.g. ‘‘45.’’.
#    Either add more values by e.g.
#      ``  V
#          5
#          63.
#          64.
#          65.
#          69.
#          70. ''.
#      The last means that computation is to be done for
#      the angles from the Vector of length 5:
#      ``63.'', ``64.'',``65.'',``69.'', and ``70.''.
#    Or add e.g.
#      ``  I 45 56 2''.
#      The last means that computation is to be done for
#      the angles ``45+i*2'' with i=0,1,2,... and with
#      angle ``45+i*2'' less than ``56''.
#    Note that two of the three, the wave length, the
#    angle of incident wave theta, and the angle of
#    incident wave phi, must be single valued.
# Angle:
  47.
###################################################################
# Length factor of additional shift of grating geometry.
#    This is shift into the x-direction, i.e. the
#    direction of the period to the right.
#    This is length of shift relative to period, i.e.
#    the grating structure given by subsequent input
#    will be shifted by factor times the period given
#    in subsequent input.
#    However, only the Rayleigh numbers and efficiencies
#    will be computed according to the shift. The field
#    vectors in the plots are drawn without shift, and
#    the graphics of the executable with tag ``_CHECK''
#    is drawn without shift!
#    Must be a real number between 0 and 1.
# Length:
  0.
###################################################################
# Stretching factor for grating in y-direction.
#    Must be a positive real number.
# Length:
  1.
###################################################################
# Length of additional shift of grating geometry in micro m.
#    This is shift into the y-direction, i.e. the
#    direction perpendicular to the grating surface
#    pointing into the cover material.
#    Must be a real number.
# Length:
  0.
######################################################################
# Period of grating in micro meter:
  1.
###################################################################
# Grating data.
#    Either this should be e.g. ‘‘name1’’ if ‘‘name1.inp’’
#       is the input file with the geometry data in sub-
#       directory ‘‘GEOMETRIES’’.
#       (Alternatively, a path for the location of the file
#        can be added before the name. This must contain at
#        least one slash ’/’. E.g. for a file in the current
#        working directory write ‘‘  ./name1’’)
#    Or this could be a stack of profiles given by the
#       code word stack and many more lines (cf. Userguide)
#    Or this could be
#       e.g. ‘‘  echellea R 0.3 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity, right interior angle > 45 degrees)
#          with depth of 0.3 micro meter and with coated
#          layers of height 0.03 micro meter resp. 0.04 micro
#          meter over the first resp. second part of the
#          grating (measured in direction perpendicular to
#          echelle profile, height greater or equal to zero)
#       e.g. ‘‘  echellea L 0.3 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity, left interior angle > 45 degrees)
#          with parameters like above
#       e.g. ‘‘  echellea A 60 0.03 0.04’’
#       -> ECHELLE GRATING TYPE A (right-angled triangle
#          with hypotenuse parallel to the direction of the
#          periodicity) with left interior angle Alpha=60
#          degrees (i.e. depth = period times sin(Alpha)
#          times cos(Alpha)) and other parameters like above
#       e.g. ‘‘  echelleb 60. 0.05’’
#       -> ECHELLE GRATING TYPE B (right-angled triangle
#          with one of the legs parallel to the direction
#          of the periodicity) with angle 60 (angle enclosed
#          by hypotenuse and by the leg parallel to the
#          period) and with a coated layer of height 0.05
#          micro meter (measured in direction perpendicular
#          to echelle profile, height greater or equal to
#          zero)
#       e.g. ‘‘  trapezoid 60. 0.6 3 0.2 0.1 0.1 0.05’’
#       -> TRAPEZOIDAL GRATING (trapezoid with the basis
#          parallel to the direction of the periodicity)
#          with angle of 60 degrees (angle enclosed by
#          basis and the sides) with a base of length 0.6
#          micro meter consisting of 3 material layers of
#          heights 0.2, 0.1, and 0.1 micro meter,
#          respectively, and with a coated layer of height
#          0.05 micro meter (greater or equal to zero)
#       e.g. ‘‘  lAmellar 3 4
#                0.2 0.6
#               -0.2 1.0
#                0.   0.5   0.7
#                0.0  0.50  0.90
#                0.00 0.500 0.900  ’’
#       -> LAMELLAR GRATING (rectangular grating consisting
#          of several materials placed in rectangular sub-
#          domains) with 3 columns each divided into 4
#          rectangular layers, first column with x coordinate
#          in 0<x<0.2 (given in micro meter), second column with 0.2<x<0.6,
#          third columnn with 0.6<x<period (period given above),
#          whole grating with y coordinate s.t. -0.2<y<1.0,
#          first column: first layer with -0.2<y<0., second with
#          0.<y<0.5, third with 0.5<y<0.7 and fourth with 0.7<y<1,
#          second column: first layer with -0.2<y<0.0, second with
#          0.0<y<0.50, third with 0.50<y<0.90 and fourth with 0.90<y<1.,
#          third column: first layer with -0.2<y<0.00, second with
#          0.00<y<0.500, third with 0.500<y<0.900 and fourth with
#          0.900<y<1.
#       e.g. ‘‘  lAmellar 1 1
#                0.2 0.8     ’’
#       -> SIMPLE LAYER (special case of lamellar grating) with
#          layer material s.t. y-coordinate satisfies 0.2<y<0.8
#          (given in micro meter).
#       e.g. ‘‘  polygon file1’’
#       -> GRATING DETERMINED BY A POLYGONAL LINE defined
#          by the data in the file with name ‘‘../GEOMETRIES/file1’’
#          (in ‘‘../GEOMETRIES/file1’’: in each line beginning without ‘#’
#          there should be the x- and y-coordinate of one
#          of the consecutive corner points, first point
#          with x-coordinate 0, last point with x-coordinate
#          1, same y-coordinate for first and last point,
#          all x-coordinates between 0 and 1, at least two
#          different y-coordinates, last line should be
#          ‘‘End’’)
#       e.g. ‘‘  polygon2 file1 file2’’
#       -> COATED GRATING DETERMINED BY POLYGONAL LINES,
#          i.e. grating profile line is defined by the data
#          in the file with name ‘‘../GEOMETRIES/file1’’
#          (in ‘‘../GEOMETRIES/file1’’:
#          in each line beginning without ‘#’ there should
#          be the x- and y-coordinate of one of the
#          consecutive corner points, first point with
#          x-coordinate 0, last point with x-coordinate
#          1, same y-coordinate for first and last point,
#          all x-coordinates between 0 and 1, at least two
#          different y-coordinates, last line should be
#          ‘‘End’’) and the coated layer is enclosed between
#          the polygonal line of ‘‘../GEOMETRIES/file1’’
#          and the polygonal line of the file with name
#          ‘‘../GEOMETRIES/file2’’
#          (in ‘‘../GEOMETRIES/file2’’:
#          in each line beginning without ‘#’ there should be
#          the x- and y-coordinate of one of the consecutive
#          corner points, first and last point must be corner
#          of first polygon, second polygon must be on left-
#          hand side of first, one to one correspondence of
#          the corners on the two polygons between first and
#          last point of second polygon, quadrilateral between
#          corresponding segments on the left of first
#          polygon, these quadrilaterals must be disjoint, last
#          line should be ``End'')
#       e.g. ``  profile''
#       -> GRATING DETERMINED BY A SMOOTH PARAMETRIC CURVE,
#          i.e. grating determined by profile line given as
#          {(fctx(t),fcty(t)):0<=t<=1}, where the functions
#          ``t|->fctx(t)'' and ``t|->fcty(t)'' are defined
#          by the ``c''-code of the file ``../GEOMETRIES/profile.c''
#       e.g. ``  profile_par 2 3
#                1
#                0
#                1.5
#                0.2
#                0.3              ''
#       -> GRATING DETERMINED BY A SMOOTH PARAMETRIC CURVE,
#          WITH PARAMETERS,
#          i.e. grating determined by profile line given as
#          {(fctx(t),fcty(t)):0<=t<=1}, where the functions
#          ``t|->fctx(t)'' and ``t|->fcty(t)'' are defined
#          by the ``c''-code of ``../GEOMETRIES/profile_par.c'';
#          the last code uses 2 integer parameters and 3
#          real parameters named IPARaM1, IPARaM2, RPARaM1,
#          RPARaM2, RPARaM3;
#          the integer parameters take the values 1 and 0
#          following the first line of the calling sequence
#          and the real parameters take the values 1.5, 0.2,
#          and 0.3 following the integer parameter values
#          (Any number of parameters is possible for a
#          corresponding file ``../GEOMETRIES/profile_par.c''.)
#       e.g. ``  profile 0.125*sin(2.*M_PI*t)''
#       -> GRATING DETERMINED BY A SIMPLE SMOOTH FUNCTION,
#          i.e. grating determined by sine profile line given
#          as {(t,fcty(t)):0<=t<=1}, where the function
#          ``t|->fcty(t)'' is defined by the ``c''-code
#          fcty(t)=0.125*sin(2.*M_PI*t).
#          (do not use any ``blank''/``space'' in the c-code)
#       e.g. ``  profile 0.5+0.5*cos(M_PI*(1.-t)) 0.25*sin(M_PI*t)''
#       -> GRATING DETERMINED BY A SIMPLE SMOOTH PARAMETRIC CURVE,
#          i.e., grating determined by ellipsoidal profile line
#          given as {(fctx(t),fcty(t)):0<=t<=1}, where the
#          functions ``t|->fcty(t)'' and ``t|->fcty(t)'' are
#          defined by the ``c''-codes
#          fctx(t)=0.5+0.5*cos(M_PI*(1.-t)) and
#          fcty(t)=0.25*sin(M_PI*t), respectively
#          (do not use any ``blank''/``space'' in the c-codes)
#       e.g. ``  profiles''
#       -> GRATING DETERMINED BY SMOOTH PARAMETRIC CURVES,
#          i.e. grating determined by profile lines given as
#          {(fctx(j,t),fcty(j,t)):0<=t<=1}, j=1,...,n=nmb_curves,
#          where the functions ``t|->fctx(j,t)'' and
#          ``t|->fcty(j,t)'' are defined by the ``c''-code
#          of the file ``../GEOMETRIES/profiles.c''
#       e.g. ``  profiles_par 1 2
#                3
#                0.5
#                0.50             ’’
#       -> GRATING DETERMINED BY SMOOTH PARAMETRIC CURVES,
#          WITH PARAMETERS,
#          i.e. grating determined by profile lines given as
#          {(fctx(j,t),fcty(j,t)):0<=t<=1}, j=1,...,n=nmb_curves,
#          where the functions ‘‘t|->fctx(j,t)’’ and
#          ‘‘t|->fcty(j,t)’’ are defined by the ‘‘c’’-code
#          of the file ‘‘../GEOMETRIES/profiles_par.c’’;
#          the last code uses 1 integer parameter and 2
#          real parameters named IPARaM1, RPARaM1, RPARaM2;
#          the integer parameter takes the value 3
#          following the first line of the calling sequence
#          and the real parameters take the values 0.5 and 0.50
#          following the integer parameter values
#          (Any number of parameters is possible for a
#          corresponding file ‘‘../GEOMETRIES/profiles_par.c’’.)
#       e.g. ‘‘  pin’’
#       -> PIN GRATING DETERMINED BY PARAMETRIC CURVE,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(t),fcty(t)):0<=t<=1}.
#          Here {(fctx(t),fcty(t)):0<=t<=1} is a simple open
#          arc connecting (fctx(0),fcty(0))=(xmin,0) with
#          (fctx(1),fcty(1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(t)<1, 0<t<1, and
#          such that 0<fcty(t), 0<t<1. The functions fctx, fcty
#          and the parameter xmin are defined by the code in
#          ‘‘../GEOMETRIES/pin.c’’.
#       e.g. ‘‘  cpin’’
#       -> COATED PIN GRATING DETERMINED BY TWO PARAMETRIC CURVES,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(1,t),fcty(1,t)):0<=t<=1}.
#          Here {(fctx(1,t),fcty(1,t)):0<=t<=1} is a simple open
#          arc connecting (fctx(1,0),fcty(1,0))=(xmin,0) with
#          (fctx(1,1),fcty(1,1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(1,t)<1, 0<t<1, and
#          such that 0<fcty(1,t), 0<t<1. Additionaly, a coating
#          layer is attached located between the first curve
#          {(fctx(1,t),fcty(1,t)):0<=t<=1} and a second curve
#          {(fctx(2,t),fcty(2,t)):0<=t<=1}. The last connects the
#          point (fctx(1,arg1),fcty(1,arg1))=(fctx(2,0),fcty(2,0))
#          with (fctx(1,arg2),fcty(1,arg2))=(fctx(2,1),fcty(2,1)).
#          Moreover, {(fctx(2,t),fcty(2,t)):0<=t<=1} is a simple open
#          arc above {(fctx(1,t),fcty(1,t)):0<=t<=1} such that
#          0<fctx(2,t)<1, 0<t<1. The functions fctx(1,.), fctx(2,.),
#          fcty(1,.), and fcty(2,.) and the parameters arg1, arg2,
#          and xmin are defined by the code of the file
#          ‘‘../GEOMETRIES/cpin.c’’.
#       e.g. ‘‘  cpin2’’
#       -> COATED PIN GRATING DETERMINED BY TWO PARAMETRIC CURVES
#          TYPE 2,
#          i.e. over a flat grating with surface {(x,0):0<=x<=1}
#          a material part is attached which is located between
#          {(x,0):0<=x<=1} and {(fctx(1,t),fcty(1,t)):0<=t<=1}.
#          Here {(fctx(1,t),fcty(1,t)):0<=t<=1} is a simple open
#          arc connecting (fctx(1,0),fcty(1,0))=(xmin,0) with
#          (fctx(1,1),fcty(1,1))=(1-xmin,0) such that 0<xmin<0.5
#          is a fixed number, such that 0<fctx(1,t)<1, 0<t<1, and
#          such that 0<fcty(1,t), 0<t<1. Additionaly, a coating
#          layer is attached located between the first curve
#          {(fctx(1,t),fcty(1,t)):0<=t<=1} and a second curve
#          {(fctx(2,t),fcty(2,t)):0<=t<=1}. The last connects the
#          point (x1,0)=(fctx(2,0),fcty(2,0)) with (x2,0)=
#          (fctx(2,1),fcty(2,1)) with 0<x1<xmin<1-xmin<x2. Moreover,
#          {(fctx(2,t),fcty(2,t)):0<=t<=1} is a simple open
#          arc above {(fctx(1,t),fcty(1,t)):0<=t<=1} such that
#          0<fctx(2,t)<1, 0<t<1. The functions fctx(1,.), fctx(2,.),
#          fcty(1,.), and fcty(2,.) and the parameter xmin are
#          defined by the code of the file ‘‘../GEOMETRIES/cpin2.c’’.
#       e.g. ‘‘  stack 3
#                profile t /##/ 0.2*sin(2.*M_PI*t)
#                2.
#                profile 0.2*sin(2.*M_PI*t)
#                1.
#                profile t /##/ 0.
#                0.                  ’’
#       -> STACK GRATING,
#          i.e. a stack of 3 profile curves shifted by 2, 1, 0
#          micro meter in vertical direction. For more details, see
#          the description in the USERGUIDE.
# Grating data:
  lamellar
######################################################################
# Number of different grating materials (N_mat).
#    This includes the material of substrate and cover material.
#    For example,
#       if Grating data is  ‘‘  name1’’
#       -> Number of materials given in file ‘‘name1.inp’’
#       if Grating data is ‘‘  echellea ...’’
#       -> N_mat = 3 with coating height >0
#          N_mat = 2 with coating height =0
#       if Grating data is  ‘‘  echelleb ...’’
#       -> N_mat = 3 with coating height >0
#          N_mat = 2 with coating height =0
#       if Grating data is  ‘‘  trapezoid ...’’
#       -> N_mat = number of material layers +3
#                  for coating height >0
#          N_mat = number of material layers +2
#                  for coating height =0
#       if Grating data is  ‘‘  lAmellar k m
#                            ...  ’’
#       -> N_mat = k times m plus 2
#       if Grating data is  ‘‘  polygon file1’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  polygon2 file1 file2’’
#       -> N_mat = 3
#       if Grating data is  ‘‘  profile’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  profile 2 3
#                            ...        ’’
#       -> N_mat = 2
#       if Grating data is  ‘‘  profile 0.125*sin(2.*M_PI*t)’’
#       -> N_mat = 2
#       if Grating data is
#       ‘‘  profile 0.5+0.5*cos(M_PI*(1.-t)) 0.25*sin(M_PI*t)’’
#       -> N_mat = 2
#       if Grating data is ‘‘  profiles’’
#       -> N_mat = n+1 with n=nmb_curves from
#                  the file ‘‘../GEOMETRIES/profiles.c’’
#       if Grating data is ‘‘  profiles ... ’’
#       -> N_mat = n+1 with n=nmb_curves from
#                  the file ‘‘../GEOMETRIES/profiles_par.c’’
#       if Grating data is ‘‘  pin ’’
#       -> N_mat = 3
#       if Grating data is ‘‘  cpin ’’
#       -> N_mat = 4
#       if Grating data is ‘‘  cpin2 ’’
#       -> N_mat = 4
#       if Grating data is ‘‘  stack k’’
#       -> N_mat = k+1
# Number of materials:
  2
######################################################################
# Optical indices of grating materials.
#    This is c times square root of mu times epsilon.
#    If meaningful, then the refractive indices should be ordered
#    according to the location from above to below.
#    If an input file ‘‘name1.inp’’ is used, then the optical index
#    of a subdomain with the material index j is just the j-th
#    optical index following below.
#    If grating is ‘‘lAmellar ...’’, then first material is cover
#    material, last material is substrate, and all other materials
#    are ordered from left to right and inside the columns from
#    below to above.
#    For technical reasons, the index of the material adjacent to
#    the upper line of the grating structure must coincide with
#    that of the material in the adjacent upper coated layer resp.
#    in the adjacent superstrate. Similarly the index of the
#    materials adjacent to the lower line of the grating structure
#    must coincide with that of the material in the adjacent lower
#    coated layer resp. in the adjacent substrate.
#    N_mat numbers are needed.
# Optical indices:
  1.3 +i 0.
  1.6 +i 0.
######################################################################
# Number of levels (Lev).
#    In each refinement step the step size of the mesh is halved.
#    Lev refinement steps are performed.
#    (Alternatively, one can prescribe an bound for the maximal
#     error of the efficiencies. E.g. the input ‘‘e 1.’’ means
#     that the level for the computation is the smallest positive
#     integer such that all efficiencies are computed with an
#     estimated error less than 1 per cent.)
# Number:
  3
######################################################################
# End
######################################################################
```

10.5 Output file “example.res” of FEM-FULLINFO in CLASSICAL

```
          **************************************
          **************************************
          **                                  **
          **          D P O G T R             **
          **                                  **
          **************************************
          **************************************


             date =’10. Feb 2003, 09:55:17’


Program solves Helmholtz equation for
optical grating and TE/TM polarization:

boundary conditions - periodic with respect to x,
                      non-local condition on {(x,y),y=y_max}
                      and {(x,y),y=y_min}
method              - FEM
partitioning        - based on Shewchuk’s ‘‘Triangle’’
matrix assembly     - pdelib
solver              - pardiso


code generated with: -DCONV


=================================================================
 INPUT FILE FOR GRATING:
=================================================================


 Name of input file without extension ‘‘.inp’’:
   example
 Comments:
   This is a fantasy grid
   for the test of gen_polyx!
 Number of materials:
   4
 Minimal angle of subdivision triangles:
   20.000000
 Upper bound for mesh size:
   0.500000
 Width of additional strip above and below:
   0.200000
 Grid points:
    1:   0.000000  0.800000
    2:   0.500000  0.800000
    3:   0.000000  0.400000
    4:   0.250000  0.400000
    5:   1.000000  0.400000
    6:   0.750000  0.200000
    7:   1.000000  0.200000
    8:   0.000000  0.000000
    9:   0.250000 -0.200000
   10:   1.000000 -0.200000
   11:   0.000000 -0.600000
   12:   1.000000 -0.800000
 Triangles:
   ( 1,  3,  4), mat= 2, fac= 1.000000
   ( 4,  6,  2), mat= 2, fac= 1.000000
   ( 6,  7,  5), mat= 2, fac= 1.000000
   ( 3,  8,  4), mat= 2, fac= 1.000000
   ( 8,  9,  4), mat= 2, fac= 0.300000
   ( 4,  9,  6), mat= 2, fac= 1.000000
   ( 6, 10,  7), mat= 3, fac= 1.000000
   ( 8, 11,  9), mat= 3, fac= 1.000000
   ( 9, 12,  6), mat= 3, fac= 1.000000
   ( 6, 12, 10), mat= 3, fac= 1.000000


=================================================================
 INPUT DATA:
=================================================================

COATED LAYERS

nmb of upper layers:    2
corresponding widths:
                        (last width = last width of *.dat
                        file + width in grating geometry)
                        0.50 micro m
                        0.40 micro m

nmb of lower layers:    3
corresponding widths:
                        (first width = first width of *.dat
                        file + width in grating geometry)
                        0.40 micro m
                        0.30 micro m
                        0.20 micro m

REFRACT.INDICES

cover material:         1.00 + i 0.00
layers above grating:
                        1.10 + i 0.00
                        1.20 + i 0.00
layers below grating:
                        2.30 + i 0.00
                        2.20 + i 0.00
                        2.10 + i 0.00
substrate material:     2.00 + i 0.00

FURTHER DATA

temperature:            20.00 degrees Celsius
wave length:            0.635 micro m
angle of inc.theta:     65.00 degrees
polarization type:      TM

GRATING

grating period:         1.00 micro m
grating height:         1.60 micro m
fem grid height:        2.00 micro m
nmb of materials:       4
corr.refract.indices:
                        1.20 + i 0.00
                        1.50 + i 0.00
                        1.70 + i 0.00
                        2.30 + i 0.00
nmb of levels f.comp.:  5


=================================================================
 INFO OF SOLUTION (LEVEL=1):
=================================================================

degrees of freedom   =  813
stepsize of discr.   =  0.81789
numb.of nonzero entr.=  5859
rate of nonzero entr.=  0.886426 per cent
memory for pardiso   =  536 kB

       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.039594,  0.181811)   e_    0 =  3.462309
 n=   -1  theta =  -15.74   ( -0.008538,  0.021079)   e_   -1 =  0.117800
 n=   -2  theta =   21.33   (  0.031693,  0.008219)   e_   -2 =  0.236279
 n=   -3  theta =   87.07   (  0.102928, -0.466177)   e_   -3 =  2.757185
 Reflected energy:            6.573574

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.132885,  0.438747)   e_    0 = 22.164372
 n=    1  theta =   50.41   ( -0.034493, -0.866995)   e_    1 = 56.761614
 n=   -1  theta =    7.80   ( -0.228543, -0.163526)   e_   -1 =  9.256892
 n=   -2  theta =  -10.48   ( -0.171167,  0.062429)   e_   -2 =  3.861878
 n=   -3  theta =  -29.96   ( -0.096782, -0.007596)   e_   -3 =  0.966041
 n=   -4  theta =  -54.77   ( -0.001540,  0.078023)   e_   -4 =  0.415630

 Transmitted energy:         93.426426


=================================================================
 INFO OF SOLUTION (LEVEL=2):
=================================================================

degrees of freedom   =  3197
stepsize of discr.   =  0.34705
numb.of nonzero entr.=  23379
rate of nonzero entr.=  0.228739 per cent
memory for pardiso   =  2606 kB

       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   ( -0.004059,  0.131487)   e_    0 =  1.730540
 n=   -1  theta =  -15.74   (  0.003041, -0.001482)   e_   -1 =  0.002606
 n=   -2  theta =   21.33   ( -0.009015,  0.008718)   e_   -2 =  0.034668
 n=   -3  theta =   87.07   (  0.007285,  0.148711)   e_   -3 =  0.268177

 Reflected energy:            2.035991

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.439727,  0.251543)   e_    0 = 27.066021
 n=    1  theta =   50.41   (  0.521995, -0.681800)   e_    1 = 55.589973
 n=   -1  theta =    7.80   (  0.061566, -0.302391)   e_   -1 = 11.162601
 n=   -2  theta =  -10.48   ( -0.021529, -0.115618)   e_   -2 =  1.609071
 n=   -3  theta =  -29.96   ( -0.071286, -0.050452)   e_   -3 =  0.781807
 n=   -4  theta =  -54.77   ( -0.002095,  0.160324)   e_   -4 =  1.754536

 Transmitted energy:         97.964009


=================================================================
 INFO OF SOLUTION (LEVEL=3):
=================================================================

degrees of freedom   =  12628
stepsize of discr.   =  0.19522
```

```
numb.of nonzero entr.=  92896
rate of nonzero entr.=  0.058254 per cent
memory for pardiso   =  12659 kB

       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.012070,  0.139848)   e_    0 =  1.970304
 n=   -1  theta =  -15.74   (  0.003512,  0.014030)   e_   -1 =  0.047639
 n=   -2  theta =   21.33   ( -0.002098,  0.002527)   e_   -2 =  0.002377
 n=   -3  theta =   87.07   ( -0.079551,  0.082779)   e_   -3 =  0.159453

 Reflected energy:           2.179773

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.498922,  0.127814)   e_    0 = 27.975753
 n=    1  theta =   50.41   (  0.613317, -0.568916)   e_    1 = 52.762242
 n=   -1  theta =    7.80   (  0.134396, -0.299329)   e_   -1 = 12.619565
 n=   -2  theta =  -10.48   (  0.040241, -0.125325)   e_   -2 =  2.015636
 n=   -3  theta =  -29.96   (  0.038098, -0.111135)   e_   -3 =  1.414798
 n=   -4  theta =  -54.77   ( -0.101705, -0.069144)   e_   -4 =  1.032234

 Transmitted energy:         97.820227



=================================================================
 INFO OF SOLUTION (LEVEL=4):
=================================================================

degrees of freedom   =  50133
stepsize of discr.   =  0.09156
numb.of nonzero entr.=  369931
rate of nonzero entr.=  0.014719 per cent
memory for pardiso   =  59346 kB

       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.012847,  0.143535)   e_    0 =  2.076732
 n=   -1  theta =  -15.74   (  0.000916,  0.015911)   e_   -1 =  0.057844
 n=   -3  theta =   87.07   ( -0.088879,  0.060381)   e_   -3 =  0.139668

 Reflected energy:           2.274312

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.502017,  0.095119)   e_    0 = 27.533702
 n=    1  theta =   50.41   (  0.636562, -0.542658)   e_    1 = 52.752053
 n=   -1  theta =    7.80   (  0.165801, -0.295567)   e_   -1 = 13.462258
 n=   -2  theta =  -10.48   (  0.064145, -0.115134)   e_   -2 =  2.020839
 n=   -3  theta =  -29.96   (  0.059586, -0.088603)   e_   -3 =  1.168650
 n=   -4  theta =  -54.77   ( -0.049270, -0.095506)   e_   -4 =  0.788186

 Transmitted energy:         97.725688
```

```
================================================================
 INFO OF SOLUTION (LEVEL=5):
================================================================

degrees of freedom   =  200431
stepsize of discr.   =  0.04493
numb.of nonzero entr.=  1481809
rate of nonzero entr.=  0.003689 per cent
memory for pardiso   =  278626 kB

       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.012980,  0.144468)   e_    0 =  2.103941
 n=   -1  theta =  -15.74   (  0.000471,  0.016141)   e_   -1 =  0.059386
 n=   -2  theta =   21.33   ( -0.000145, -0.000968)   e_   -2 =  0.000211
 n=   -3  theta =   87.07   ( -0.090729,  0.055189)   e_   -3 =  0.136430

 Reflected energy:            2.299969

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.502603,  0.087406)   e_    0 = 27.447303
 n=    1  theta =   50.41   (  0.642427, -0.535243)   e_    1 = 52.715032
 n=   -1  theta =    7.80   (  0.174148, -0.293832)   e_   -1 = 13.675032
 n=   -2  theta =  -10.48   (  0.069855, -0.111978)   e_   -2 =  2.026452
 n=   -3  theta =  -29.96   (  0.062905, -0.082480)   e_   -3 =  1.102941
 n=   -4  theta =  -54.77   ( -0.036440, -0.097038)   e_   -4 =  0.733271

 Transmitted energy:         97.700031


================================================================
 CONVERGENCE PROPERTIES:
================================================================

1.VALUE (ORDER 0)

+--------+--------------+--------------+--------------+-----------+
|  h     |    value     |    extrap.   |     error    | conv.ord. |
+--------+--------------+--------------+--------------+-----------+
| 0.818 |     3.4623   |              |     1.3312   |           |
| 0.347 |     1.7305   |              |     0.4006   |           |
| 0.195 |     1.9703   |    1.9411    |     0.1608   |    2.85   |
| 0.092 |     2.0767   |    2.1617    |     0.0544   |    1.17   |
| 0.045 |     2.1039   |    2.1133    |     0.0272   |    1.97   |
+--------+--------------+--------------+--------------+-----------+

2.VALUE (ORDER -1)

+--------+--------------+--------------+--------------+-----------+
|  h    |    value     |    extrap.   |     error    | conv.ord.  |
+-------+--------------+--------------+--------------+------------+
| 0.818 |     0.1178   |              |     0.0569   |            |
| 0.347 |     0.0026   |              |     0.0583   |            |
| 0.195 |     0.0476   |     0.0350   |     0.0133   |    1.36    |
| 0.092 |     0.0578   |     0.0608   |     0.0031   |    2.14    |
| 0.045 |     0.0594   |     0.0597   |     0.0015   |    2.73    |
+-------+--------------+--------------+--------------+------------+

NO ERROR ANALYSIS FOR ORDER 3

4.VALUE (ORDER -3)

+-------+--------------+--------------+--------------+------------+
|  h    |    value     |    extrap.   |     error    | conv.ord.  |
+-------+--------------+--------------+--------------+------------+
| 0.818 |     2.7572   |              |     2.6240   |            |
| 0.347 |     0.2682   |              |     0.1350   |            |
| 0.195 |     0.1595   |     0.1545   |     0.0263   |    4.52    |
| 0.092 |     0.1397   |     0.1353   |     0.0065   |    2.46    |
| 0.045 |     0.1364   |     0.1358   |     0.0032   |    2.61    |
+-------+--------------+--------------+--------------+------------+

5.VALUE (REFLECTED ENERGY)

+-------+--------------+--------------+--------------+------------+
|  h    |    value     |    extrap.   |     error    | conv.ord.  |
+-------+--------------+--------------+--------------+------------+
| 0.818 |     6.5736   |              |     4.2479   |            |
| 0.347 |     2.0360   |              |     0.2896   |            |
| 0.195 |     2.1798   |     2.1754   |     0.1459   |    4.98    |
| 0.092 |     2.2743   |     2.4558   |     0.0513   |    0.60    |
| 0.045 |     2.3000   |     2.3095   |     0.0257   |    1.88    |
+-------+--------------+--------------+--------------+------------+

6.VALUE (ORDER 0)

+-------+--------------+--------------+--------------+------------+
|  h    |    value     |    extrap.   |     error    | conv.ord.  |
+-------+--------------+--------------+--------------+------------+
| 0.818 |    22.1644   |              |     5.1965   |            |
| 0.347 |    27.0660   |              |     0.2949   |            |
| 0.195 |    27.9758   |    28.1831   |     0.6148   |    2.43    |
| 0.092 |    27.5337   |    27.6783   |     0.1728   |    1.04    |
| 0.045 |    27.4473   |    27.4263   |     0.0864   |    2.36    |
+-------+--------------+--------------+--------------+------------+

7.VALUE (ORDER 1)

+-------+--------------+--------------+--------------+------------+
|  h    |    value     |    extrap.   |     error    | conv.ord.  |
+-------+--------------+--------------+--------------+------------+
| 0.818 |    56.7616   |              |     4.0836   |            |
| 0.347 |    55.5900   |              |     2.9120   |            |
| 0.195 |    52.7622   |    57.5905   |     0.0842   |   -1.27   |
| 0.092 |    52.7521   |    52.7520   |     0.0740   |    8.12   |
| 0.045 |    52.7150   |    52.7661   |     0.0370   |   -1.86   |
+-------+--------------+--------------+--------------+-----------+

8.VALUE (ORDER -1)

+-------+--------------+--------------+--------------+-----------+
|  h    |    value     |    extrap.   |     error    | conv.ord. |
+-------+--------------+--------------+--------------+-----------+
| 0.818 |     9.2569   |              |     4.6309   |           |
| 0.347 |    11.1626   |              |     2.7252   |           |
| 0.195 |    12.6196   |    17.3500   |     1.2682   |    0.39   |
| 0.092 |    13.4623   |    14.6183   |     0.4255   |    0.79   |
| 0.045 |    13.6750   |    13.7469   |     0.2128   |    1.99   |
+-------+--------------+--------------+--------------+-----------+

9.VALUE (ORDER -2)

+-------+--------------+--------------+--------------+-----------+
|  h    |    value     |    extrap.   |     error    | conv.ord. |
+-------+--------------+--------------+--------------+-----------+
| 0.818 |     3.8619   |              |     1.8298   |           |
| 0.347 |     1.6091   |              |     0.4230   |           |
| 0.195 |     2.0156   |     1.9535   |     0.0164   |    2.47   |
| 0.092 |     2.0208   |     2.0209   |     0.0112   |    6.29   |
| 0.045 |     2.0265   |     1.9495   |     0.0056   |   -0.11   |
+-------+--------------+--------------+--------------+-----------+

10.VALUE (ORDER -3)

+-------+--------------+--------------+--------------+-----------+
|  h    |    value     |    extrap.   |     error    | conv.ord. |
+-------+--------------+--------------+--------------+-----------+
| 0.818 |     0.9660   |              |     0.0712   |           |
| 0.347 |     0.7818   |              |     0.2554   |           |
| 0.195 |     1.4148   |     0.9245   |     0.3776   |   -1.78   |
| 0.092 |     1.1687   |     1.2376   |     0.1314   |    1.36   |
| 0.045 |     1.1029   |     1.0790   |     0.0657   |    1.91   |
+-------+--------------+--------------+--------------+-----------+

11.VALUE (ORDER -4)

+-------+--------------+--------------+--------------+-----------+
|  h    |    value     |    extrap.   |     error    | conv.ord. |
+-------+--------------+--------------+--------------+-----------+
| 0.818 |     0.4156   |              |     0.2627   |           |
| 0.347 |     1.7545   |              |     1.0762   |           |
| 0.195 |     1.0322   |     1.2853   |     0.3539   |    0.89   |
| 0.092 |     0.7882   |     0.6637   |     0.1098   |    1.57   |
| 0.045 |     0.7333   |     0.7173   |     0.0549   |    2.15   |
+-------+--------------+--------------+--------------+-----------+

12.VALUE (TRANSMITTED ENERGY)
```

```
+-------+-------------+-------------+-------------+----------+
|  h    |    value    |    extrap.  |     error   | conv.ord. |
+-------+-------------+-------------+-------------+----------+
| 0.818 |    93.4264  |             |     4.2479  |           |
| 0.347 |    97.9640  |             |     0.2896  |           |
| 0.195 |    97.8202  |   97.8246   |     0.1459  |    4.98   |
| 0.092 |    97.7257  |   97.5442   |     0.0513  |    0.60   |
| 0.045 |    97.7000  |   97.6905   |     0.0257  |    1.88   |
+-------+-------------+-------------+-------------+----------+


=================================================================
 END:
=================================================================

      date ='10. Feb 2003, 09:56:18'

Thank you for choosing ``dpogtr''!
Bye, bye!
```

10.6 Output file "example.res" of GFEM in CLASSICAL

```
             date ='10. Feb 2003, 10:13:19'


          ***************************************
          ***************************************
          **                                   **
          **        G D P O G T R              **
          **                                   **
          ***************************************
          ***************************************



=================================================================
 INFO OF SOLUTION (LEVEL=1):
=================================================================


       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.013025,  0.144735)   e_    0 =  2.111777
 n=   -1  theta =  -15.74   (  0.000363,  0.016237)   e_   -1 =  0.060075
 n=   -2  theta =   21.33   (  0.000035, -0.001347)   e_   -2 =  0.000400
 n=   -3  theta =   87.07   ( -0.091483,  0.053904)   e_   -3 =  0.136395

 Reflected energy:           2.308647

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.502786,  0.085087)   e_    0 = 27.424602
 n=    1  theta =   50.41   (  0.644183, -0.532792)   e_    1 = 52.687989
 n=   -1  theta =    7.80   (  0.176698, -0.293270)   e_   -1 = 13.741238
 n=   -2  theta =  -10.48   (  0.071550, -0.111089)   e_   -2 =  2.031272
 n=   -3  theta =  -29.96   (  0.063904, -0.080675)   e_   -3 =  1.085740
 n=   -4  theta =  -54.77   ( -0.032720, -0.097400)   e_   -4 =  0.720512

 Transmitted energy:        97.691353


=================================================================
 INFO OF SOLUTION (LEVEL=2):
=================================================================


       Reflection efficiencies and coefficients

 n=    0  theta =  -65.00   (  0.013043,  0.144752)   e_    0 =  2.112318
 n=   -1  theta =  -15.74   (  0.000365,  0.016240)   e_   -1 =  0.060098
 n=   -2  theta =   21.33   (  0.000044, -0.001384)   e_   -2 =  0.000423
 n=   -3  theta =   87.07   ( -0.091547,  0.053854)   e_   -3 =  0.136472

 Reflected energy:           2.309310

       Transmission efficiencies and coefficients

 n=    0  theta =   26.95   ( -0.502838,  0.084941)   e_    0 = 27.427416
 n=    1  theta =   50.41   (  0.644282, -0.532607)   e_    1 = 52.682703
 n=   -1  theta =    7.80   (  0.176861, -0.293228)   e_   -1 = 13.745111
 n=   -2  theta =  -10.48   (  0.071650, -0.111036)   e_   -2 =  2.031559
 n=   -3  theta =  -29.96   (  0.063954, -0.080553)   e_   -3 =  1.084372
 n=   -4  theta =  -54.77   ( -0.032481, -0.097406)   e_   -4 =  0.719528

 Transmitted energy:        97.690690



=================================================================
 END:
=================================================================

      date ='10. Feb 2003, 10:30:53'

Thank you for choosing ``gdpogtr''!
Bye, bye!
```

10.7 Output file “example.res” of FEM in CONICAL

```
             date ='10. Feb 2003, 10:06:22'


          ***************************************
          ***************************************
          **                                   **
          **         C O N I C A L             **
          **                                   **
          ***************************************
          ***************************************


=====================================================================
 INFO OF SOLUTION (LEVEL=1):
=====================================================================



       Reflection efficiencies and coefficients
       ----------------------------------------

 Order   Phi    Theta          E_z
                               H_z           Efficiency
     0   47.00  30.00  (  0.19657, -0.16355)
                       ( -0.05050, -0.15697)    9.25813
    -1  128.80  27.98  (  0.01482, -0.03567)
                       (  0.06213, -0.13130)    2.30360
    -2  158.51  86.74  (  0.07634,  0.09663)
                       (  0.12963,  0.05033)    0.22670

 Reflected energy:     11.78843

       Transmission efficiencies and coefficients
       ------------------------------------------

 Order   Phi    Theta          E_z
                               H_z           Efficiency
     0   47.00 160.53  ( -0.18113,  0.45695)
                       ( -0.14912,  0.68850)   80.24261
     1   20.54 135.99  (  0.07497,  0.15146)
                       ( -0.05129,  0.15348)    5.32346
    -1  128.80 161.77  ( -0.05523, -0.08440)
                       (  0.02083, -0.07834)    2.29025
    -2  158.51 138.27  (  0.03830,  0.01953)
                       (  0.01062,  0.03930)    0.35525

 Transmitted energy:   88.21157


=====================================================================
 INFO OF SOLUTION (LEVEL=2):
=====================================================================




       Reflection efficiencies and coefficients
       ----------------------------------------

 Order   Phi    Theta           E_z
                                H_z           Efficiency
     0   47.00  30.00  (  0.00901, -0.00844)
                       ( -0.13596, -0.02289)    1.91613
    -1  128.80  27.98  ( -0.00200,  0.08454)
                       (  0.04245,  0.01373)    0.93218
    -2  158.51  86.74  ( -0.06824,  0.05857)
                       ( -0.06300,  0.12368)    0.17972

 Reflected energy:      3.02802

       Transmission efficiencies and coefficients
       ------------------------------------------

 Order   Phi    Theta           E_z
                                H_z           Efficiency
     0   47.00 160.53  ( -0.31129,  0.27182)
                       ( -0.49973,  0.42367)   62.77211
     1   20.54 135.99  ( -0.26115,  0.05446)
                       ( -0.32171,  0.20006)   17.87244
    -1  128.80 161.77  ( -0.12492, -0.19255)
                       ( -0.13873, -0.20804)   14.07546
    -2  158.51 138.27  (  0.08036, -0.05007)
                       (  0.10289, -0.07817)    2.25198

 Transmitted energy:   96.97198


=====================================================================
 INFO OF SOLUTION (LEVEL=3):
=====================================================================




       Reflection efficiencies and coefficients
       ----------------------------------------

 Order   Phi    Theta          E_z
                               H_z          Efficiency
     0   47.00  30.00  (  0.06129, -0.00846)
                       ( -0.09079, -0.04955)    1.45263
    -1  128.80  27.98  ( -0.02946,  0.01247)
                       (  0.02120, -0.02318)    0.20498
    -2  158.51  86.74  ( -0.05922, -0.00385)
                       ( -0.01123,  0.00823)    0.02441

 Reflected energy:      1.68202

       Transmission efficiencies and coefficients
       ------------------------------------------

 Order   Phi    Theta          E_z
                               H_z          Efficiency
     0   47.00 160.53  ( -0.38108,  0.19799)
                       ( -0.58998,  0.31418)   66.49400
     1   20.54 135.99  ( -0.22472, -0.00453)
                       ( -0.37512,  0.09299)   15.48188
    -1  128.80 161.77  ( -0.07243, -0.23512)
                       ( -0.06585, -0.25190)   15.85666
    -2  158.51 138.27  (  0.05169,  0.00215)
                       (  0.03000, -0.03203)    0.48544

 Transmitted energy:   98.31798


=====================================================================
 INFO OF SOLUTION (LEVEL=4):
=====================================================================




       Reflection efficiencies and coefficients
       ----------------------------------------

 Order   Phi    Theta          E_z
                               H_z          Efficiency
     0   47.00  30.00  (  0.06068,  0.01649)
                       ( -0.07617, -0.03097)    1.07145
    -1  128.80  27.98  ( -0.02951,  0.00963)
                       (  0.01652, -0.00861)    0.13367
    -2  158.51  86.74  ( -0.04706, -0.03015)
                       (  0.00604,  0.01967)    0.02330

 Reflected energy:      1.22842

       Transmission efficiencies and coefficients
       ------------------------------------------
 Order   Phi    Theta          E_z
                                H_z          Efficiency
     0   47.00 160.53  ( -0.38965,  0.12856)
                       ( -0.61617,  0.20649)   61.81559
     1   20.54 135.99  ( -0.25045, -0.03050)
                       ( -0.41925,  0.03884)   18.86494
    -1  128.80 161.77  ( -0.02063, -0.25490)
                       ( -0.01257, -0.27967)   17.53158
    -2  158.51 138.27  (  0.05192,  0.01214)
                       (  0.04725, -0.02301)    0.55947

 Transmitted energy:   98.77158


=====================================================================
 INFO OF SOLUTION (LEVEL=5):
=====================================================================




       Reflection efficiencies and coefficients
       ----------------------------------------

 Order   Phi    Theta          E_z
                                H_z          Efficiency
     0   47.00  30.00  (  0.06175,  0.01688)
                       ( -0.07173, -0.03108)    1.02091
    -1  128.80  27.98  ( -0.02812,  0.00775)
                       (  0.01579, -0.00606)    0.11595
    -2  158.51  86.74  ( -0.04494, -0.03733)
                       (  0.00386,  0.01787)    0.02462

 Reflected energy:      1.16148

       Transmission efficiencies and coefficients
       ------------------------------------------

 Order   Phi    Theta          E_z
                                H_z          Efficiency
     0   47.00 160.53  ( -0.39161,  0.10975)
                       ( -0.61958,  0.17733)   60.76400
     1   20.54 135.99  ( -0.25177, -0.03911)
                       ( -0.42754,  0.02212)   19.38467
    -1  128.80 161.77  ( -0.00565, -0.25896)
                       (  0.00395, -0.28489)   18.04542
    -2  158.51 138.27  (  0.05587,  0.01457)
                       (  0.05139, -0.02021)    0.64442

 Transmitted energy:   98.83852


=====================================================================
 END:
=====================================================================

      date =’10. Feb 2003, 10:06:25’

Thank you for choosing ‘‘CONICAL’’!
Bye, bye!
```

11 Copyright

Responsible programmer:

A. Rathsfeld

The programs are part of the package:

`DIPOG`
(Direct and Inverse Problems for Optical Gratings)

The programs require codes written by:

J.R. Shewchuk : triangulation code `TRIANGLE`
O. Schenk, K. Gärtner : direct solver `PARDISO`
R.W. Freund, N.M. Nachtigal : qmr solver

The programs are based on codes written by:

K. Gärtner : direct solver, cgs solver
R. Schlundt : gmres solver
J. Fuhrmann, T. Koprucki, H. Langmach : `PDELIB`, adaption of `GLTOOLS`
B. Kleemann, G. Schmidt, A. Rathsfeld : adaption to the grating,
diffraction problem,
generalized Fem

Owner of program:

Weierstraß-Institut für Angewandte Analysis and Stochastik
D-10117 Berlin, Mohrenstr. 39, Germany
im Forschungsverbund Berlin e.V.
Wissenschaftsgemeinschaft Gottfried Wilhelm Leibniz e.V.

References: see Section 2.4.

Acknowledgements:

The author gratefully acknowledges the support of the German Ministry of Education, Research and Technology under Grant No. 03-ELM3B5.