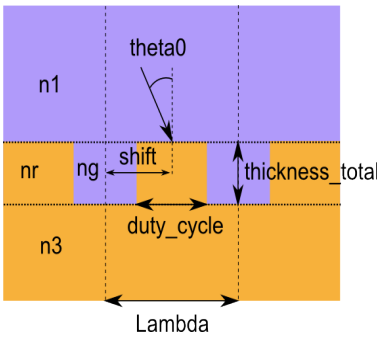
Binary grating



We want to analyze a binary metallic grating, e.g. from paper: *Li, Haggans, "Convergence of the coupled-wave method for metallic lamellar diffraction gratings," J. Opt. Soc. Am. A **10**, 1184-1189 (1993)*. It consist of a lamellar grating with a groove depth of 1 μm. The dielectric has an optical index of 0.22-i*6.71. The incident wavelength λ, the period Λ, and the height of the grating is 1 μm. The angle of incidence is equal to 30º. The incident polarization is TM.

Firstly, we have to set the following parameters in the file main.m as follows:

```
number_of_orders=41;

lambda=1;
use_dispersion=2; % 2= no dispersion
theta0=30;
polarization=1; % 1=TM polarization

grating=0; % 0=binary grating
Lambda=1;
thickness_total=1;

n1=1;
n3=0.22-6.71*1i;
ng=1;
nr=n3;

duty_cycle=.5;
shift=.5;

measurement=0; % 0=shows all diffraction efficiencies
```

We obtain this result:

```
reflected orders
   order                efficiency
  -1.000000000000000   0.101539667397081
                   0   0.844253933170622

transmitted orders
only evanescent orders

sum_eff =

   0.945793600567703   0.011854620163793   0.957648220731496
```

There are only two reflected diffraction orders, all transmitted orders are evanescent. Sum of reflected orders is equal to 0.94579, whereas sum of transmitted order is 0.01185. This grating is lossy, so the total sum 0.9576482 is lesser than 1. Secondly, we want to investigate, if the convergence of the zero-order diffraction efficiency is satisfied. We have to set:

```
measurement=1; % 1=dependence of the diff. efficiency on the number of orders
```

```
diffraction_efficiencies_c=1; % reflected orders
studying_order=0;

% in case 1

switch measurement

    case 1
        minimum_number_of_orders=3; % only odd number
        maximum_number_of_orders=101; % only odd number
        step=2;
```

We get a figure as shown here:

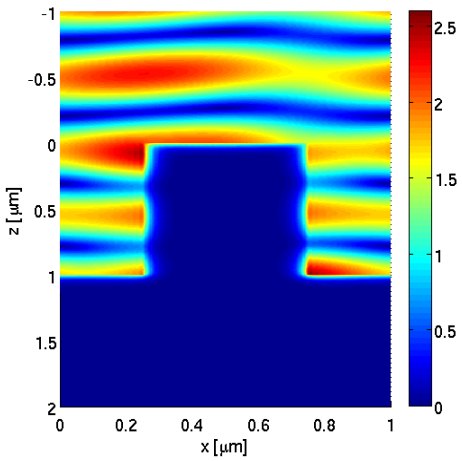
You can see, that `number_of_orders=41` is sufficient, but in order to obtain accurate results we can increase this parameter to 101 orders. Next, if we want to see field the component abs(Hy), we have to change properties as follows:

```
measurement=0;

plot_field_region_1=1; %1-yes, incident region
plot_field_region_2=1; %1-yes, grating region
plot_field_region_3=1; %1-yes, substrate region

field_component_operation=1; % 1-abs, 2-angle, 3-real, 4-imag
planar_field_component=2; % TM → H_y
set_view_grating_boundaries=2; % 1-yes, 2-no, you can see grating boundaries when
field is plotted
```

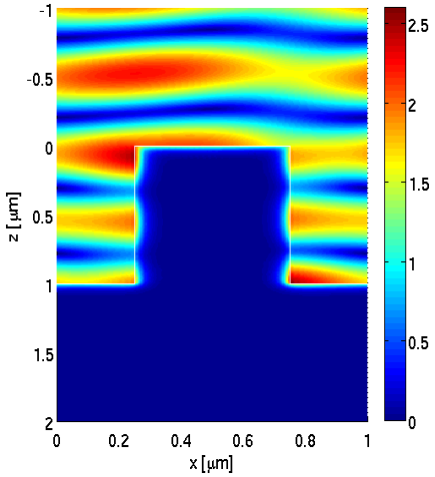
If we set `set_view_grating_boundaries=1;` then we can also see boundaries of the grating.
Program RCWA-1D offers all field components: E_x, H_y, E_z in case of TM polarization, H_x, E_y, H_z in case of TE polarization and H_x, H_y, H_z, E_x, E_y, E_z in case of conical diffraction.

Finally, this grating is made of gold, if we will change wavelength, we will use dispersion model/data. RCWA-1D program is able to easily incorporate dispersion model/data. Set `use_dispersion=1`, open file setup_dispersion.m, and write/uncomment

```
n3=rix_spline(lambda,'gold_palik.txt'); % application of a spline function to
Palik's gold data

nr=n3;
```



Field profile abs(Hy)



Field profile with grating boundaries (thin white line)

Sinusoidal grating

Now, we want to study a spectral dependence of the sinusoidal grating (Λ=2 μm, d=1 μm, n_1=n_g=1, n_3=n_r=1.5, θ=30º) in conical mount (φ=30°, ψ=30°). Set `polarization=0` to calculate conical diffraction and switch grating to sinusoidal grating → 3. In the beginning we will use only 20 approximate layers and set λ=1 μm.

```
number_of_orders=51;

lambda=1; % wavelength (um)
use_dispersion=2; % 1-yes, 2-no, see file setup_dispersion.m setup_dispersion.m
theta0=30; % incident angle [degree]
polarization=0; % 0-conical diffraction, 1-TM polarization, 2-TE polarization

phi0=30; % conical angle [degree], 0=planar diffraction
psi0=30; % polarization angle [degree], 0-TM polarization, 90-TE polarization

grating=3; % switch grating
Lambda=1; % grating period [um] (exception grating=10)
thickness_total=1; % total thickness [um] (exception grating=10)


% refractive indicies
%-----------------------------------------------------
n1=1; % homogenous incident region 1
n3=1.5; % homogenous region 3
ng=1; % Groove indicies
nr=n3; % Ridge indices

switch grating
    case 3 % sinusoidal grating
        number_of_layers=20;
```

```
measurement=0;
```

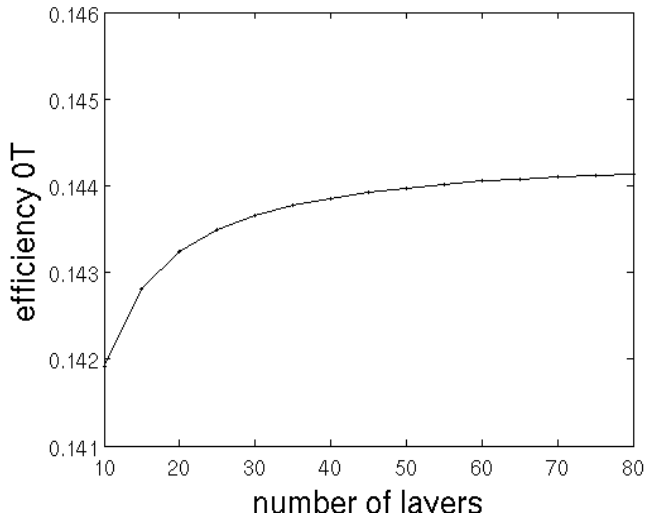
We get this result:

```
  reflected orders
   order                efficiency
  -2.000000000000000   0.004421523239889
  -1.000000000000000   0.004224700994904
                   0   0.000676667880030
   1.000000000000000   0.005561992177244

transmitted orders
   order                efficiency
  -3.000000000000000   0.013089017839922
  -2.000000000000000   0.024058203584373
  -1.000000000000000   0.278806883151152
                   0   0.143246725076277
   1.000000000000000   0.513569971871019
   2.000000000000000   0.012344314183415


sum_eff =

   0.014884884292066   0.985115115706159   0.999999999998225
```

Before calculating a spectral dependence of the sinusoidal grating, we have to check the convergence of the diffraction efficiencies on the number of aproximate layers → measurement 7. For example, we are interested in only the zero transmitted order.

```
measurement=7; % measurement 1 --- check studying_order

% !!! setup of measured diff. order measurement>0
diffraction_efficiencies_c=2; %1--D_R, 2--D_T
studying_order=0; %check minimum_number_of_orders

switch measurement

     case 7
        minimum_layer=10;
        maximum_layer=80;
        step=5;
```

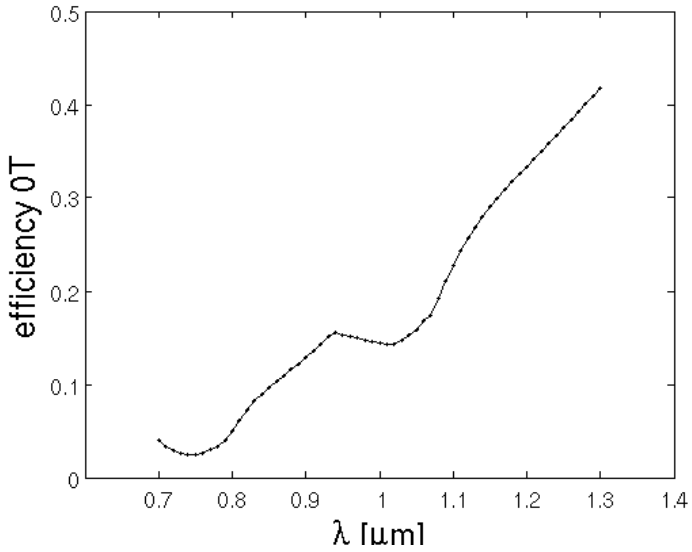
We get a figure as shown here.



You can see in the figure above that we should increase the number of layers, i.e. to 50 layers. To study spectral dependence (0.7—1.3 μm) set `measurement` to 2. To save all results of all diffraction orders set `save_m` to 1 and choose a proper name of this file.

```
switch grating
     case 3 % sinusoidal grating
        number_of_layers=50;

measurement=2;

diffraction_efficiencies_c=2; %1--D_R, 2--D_T
studying_order=0; %check minimum_number_of_orders
save_m=1; % 1 --- yes, 2 ---no
save_as='conical_sinusoidal_grating';


switch measurement
    case 2 %
        minimum_wavelength=0.7; %um
        maximum_wavelength=1.3; %um
        step=0.01;
```

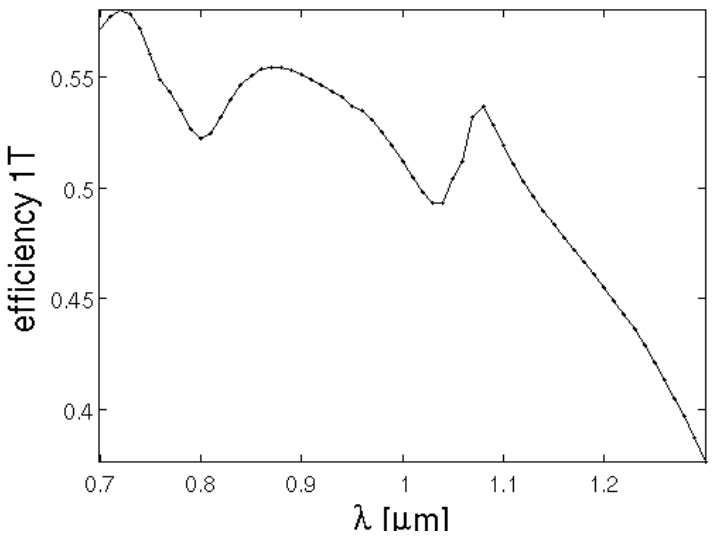
Following figure will apear:



This is only the graph of the zero transmitted order. The file (`conical_sinusoidal_grating.mat`) containing all results is saved in directory /measurements (actual data are stored in the file `conical_sinusoidal_grating.dat`). These data are avaiable through file load_measurements_results.m. If you want to see/save the spectral dependence of the first transmitted order change respective variables.

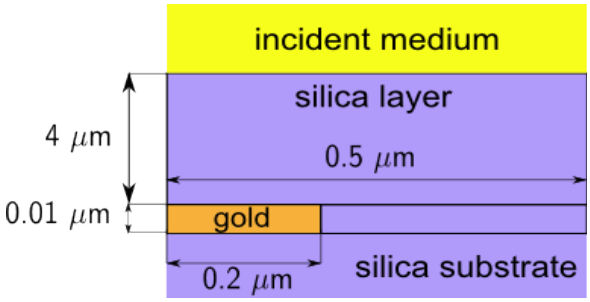
`load_measurements_results.m`

```
load conical_sinusoidal_grating.mat;
diffraction_efficiencies_c=2; %1--D_R, 2--D_T
studying_order=1;
export_studying_order=1; % 1-yes, 2-no;
save_export_as='conical_sinusoidal_grating_p1T.txt';
```



Grating from file

We would like to calculate a grating from paper: Z. Wu, J. W. Haus, Q. Zhan and R. L. Nelson, "Plasmonic notch filter design based on long-range surface plasmon excitation along metal grating," *Plasmonics*, **3**, 103-108 (2008). The period of the gold grating is fixed at Λ=500 nm, the thickness of silica layer is d=4 μm, the grating height is h=10 nm, and the grating duty cycle is 0.3. The refractive index of the incident medium is equal to 1.9 and the refractive index of the substrate medium is equal to refractive index of silica.



Firstly, we have to set material dispersion of gold and silica (file setup_dispersion.m and `use_dispersion=1`):

```
silica=cauchy(lambda,1.491,0.00686,-0.0007648);
gold=rix_spline(lambda,'gold_palik.txt');
```

after that, we create the file notch_filter.m

```
notch_thickness_g=0.01;
notch_thickness_s=4;
notch_duty=0.3;
notch_L=0.5;

data=[
```

```
notch_thickness_s, 0, silica, notch_L, NaN, NaN;
notch_thickness_g, 0, gold, notch_L*notch_duty, silica, notch_L;
];

n1=1.9;
n3=silica;
```

Grating properties are stored in the matrix `data`. The grating is composed from two layers: uniform silica layer (4 μm) and gold-silica layer (10 nm). The first column of the matrix `data` denotes the thickness of the particular layer, all dimensions are in μm. Then (second layer), there is an array of coordinate (0 in this case), refractive index (gold), coordinate (`notch_L*notch_duty=0.2`), refractive index (silica), coordinate (`notch_L=Lambda=0.5`). Since the first layer is uniform (coordinate, r. index, coordinate) and gratings parameters are stored in the matrix we have to fill in this matrix row by the special symbol NaN. After that, we have to load this grating in the file main.m (the period and the thickness are load from the file)

```
grating=10;


switch grating
->
    case 10 % arbitrary grating from file
        grating_relative_dimensions=2; % 1-relative size, 2-absolute size
        input_grating_file='notch_filter.m';
```

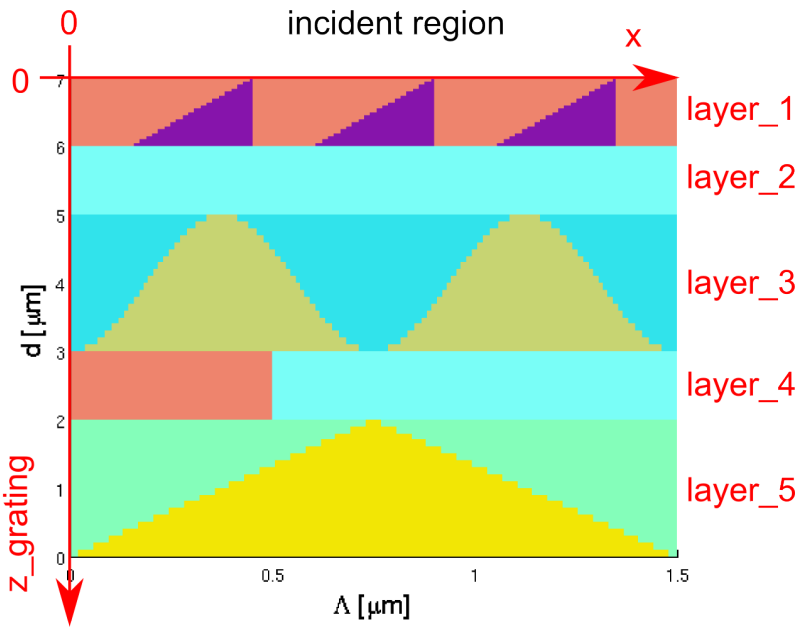
Finaly, we have to set parameters of calculation (`number_of_orders, lambda, theta, measurement,…`) for specific simulation (see “Binary grating” and “Sinusoidal grating”).

Advanced “Grating from file”

Example:

This example can be found in the /gratings/test.m file.

```
clear all
plot_grating=1;

Lambda=1.5;

x=Lambda.*[0.1 0.3, 0.4, 0.6, 0.7, 0.9, 0.9, 0.7, 0.6, 0.6, 0.4, 0.3, 0.3];
z_local=[0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1];

layer_1=add_pol_layer(Lambda,x,z_local,3.3,1,20); % polygon layer
layer_2=add_hom_layer(Lambda,1,3); % homogeneous layer
layer_3=add_sin_layer(Lambda,Lambda/2,2,2.5,2.3,20); % sinus layer
layer_4=[1,0,1,0.5,3,Lambda]; % user's layer
layer_5=add_tri_layer(Lambda,Lambda,2,3.5,2.4,20); % triangle layer

create_grating; % this scripts create input data
```

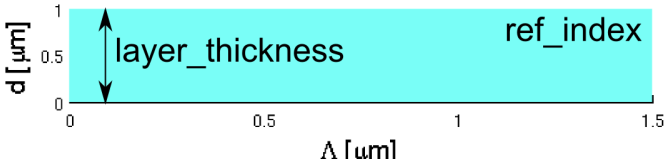


!!! For RCWA calculation: comment "`clear all`" and set `plot_grating=2;` then (See section "Grating from file") `case 10` (in `main.m` file) and `input_grating_file='test.m';`

Homogeneous layer

Function add_hom_layer creates a homogeneous layer.

```
add_hom_layer(Lambda,layer_thickness,ref_index)
```

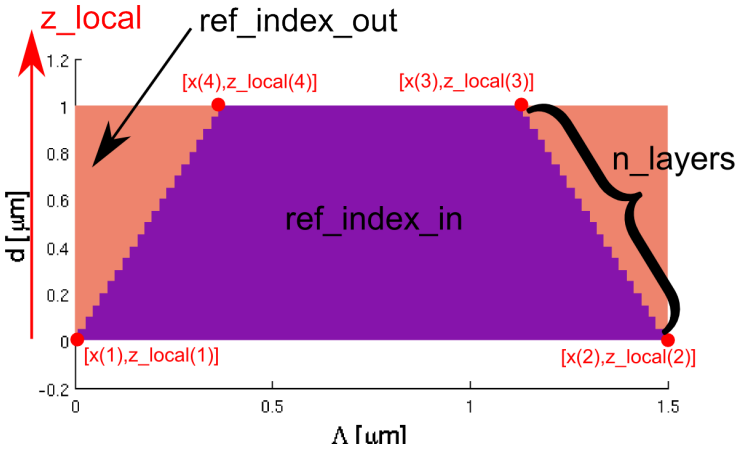


Polygon layer

Function add_pol_layer creates a polygon using coordinates *x* and *y*.

```
add_pol_layer(Lambda,x,y,ref_index_in,ref_index_out,n_layers)
```

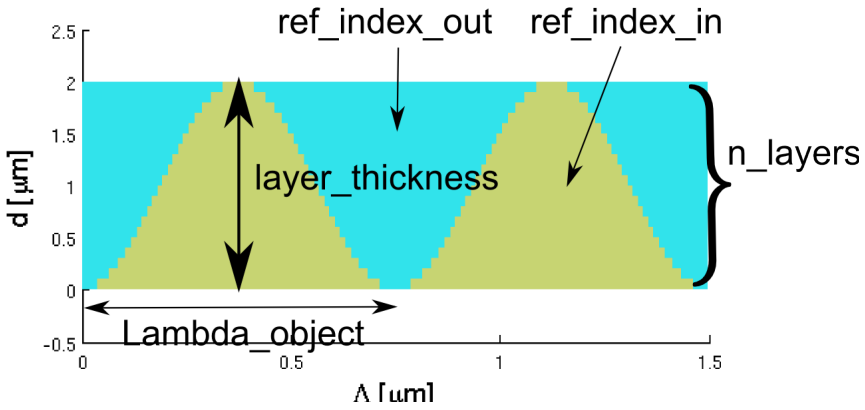
```
x=1.5.*[0 1 0.75 0.25];
z_local=[0 0 1 1];
```



Sinus layer

Function add_sin_layer creates a sinus layer.

```
add_sin_layer(Lambda,Lambda_object,layer_thickness,ref_index_in,ref_index_out,n_la
yers)
```



note: *Lambda_object=Lambda/n*, where *n* is integer.

User's layer

```
[layer_thickness, boundary_1=0, ref_index_1, boundary_2, ref_index_2,
boundary_3, ..., Lambda]
```

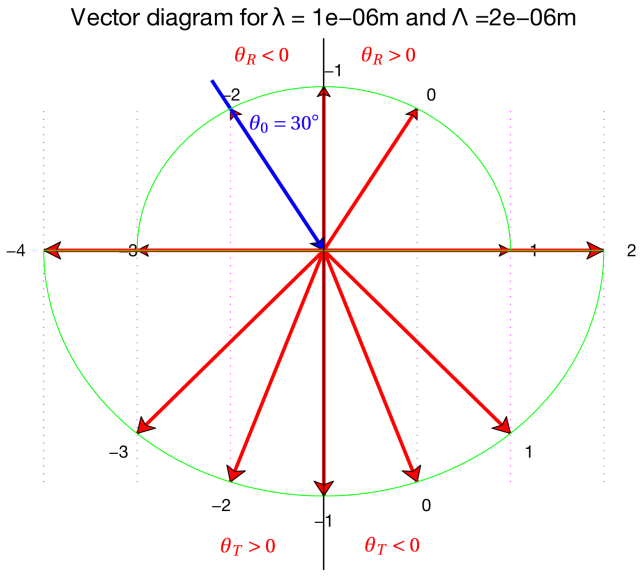
See section “Grating from file”

Some notes

Diffraction order angles

Planar diffraction: The range of angle θ is (-90º,90º)

Setup: see Sinusoidal grating (planar diffraction)




```
diff_R_angles =

  -2.000000000000000 -30.000000000000011
  -1.000000000000000  -0.000000000000004
   0                  29.999999999999993
   1.000000000000000  90.000000000000000


diff_T_angles =

  -4.000000000000000  90.000000000000000
```

```
-3.000000000000000  41.810314895778603
-2.000000000000000  19.471220634490692
-1.000000000000000  0.000000000000003
 0                 -19.471220634490685
 1.000000000000000 -41.810314895778589
 2.000000000000000 -89.999998683799234
```

Conical diffraction: The range of angle θ is (0º,90º). There is an inclination φ_i – variable diff_phi_i. The range of angle φ_i is (-90º,90º).

TODO: polarization angles

Local absorption

Description of the calculation of the local absorption with the rigorous coupled-wave method is given in the paper:

K.-H. Brenner, "Aspects for calculating local absorption with the rigorous coupled-wave method," *Optics Express*, **18**, 10369 (2010).

Integrated absorption of difractive structures is defined by

$$A_I = \frac{\Delta z}{NX}\frac{k_0^2}{k_{i,z}}\sum_{j=0}^{NX-1}\sum_{k=0}^{NZ-1}\Im\left(\epsilon(x_j,u_k)\right)\left|\left(E(x_j,z_k)\right)\right|^2,$$

where Δz is the discretization step in the *z* direction and *NX* is number of discretization elements in the *x* direction.

In case of the TM polarization, the integrated absorption must be multiply by a factor $\frac{\epsilon_1}{\frac{\mu_0}{\epsilon_0}}$, where ϵ_1 is a permittivity of the incident medium.

Global absoption of difractive structures is defined by

$A_g = 1-\sum\left(D_{R,i}+D_{T,i}\right)$, where $D_{R,i}$ and $D_{T,i}$ are diffraction efficiencies for each diffraction mode.

The integrated absorption should be equal to the global absorption.

Example from the paper: the incident region has an index of refraction of n_1=3.65, the transmitted region is air. The layer region contains a symmetric grating with a period Λ = 0.6 μm, a thickness d = 0.3 μm and two regions of equal diameter with $\epsilon_1 = 13.3+0.025\mathrm{i}$ and air. The incident wave has a wavelength of 1 μm and enters at perpendicular incidence.

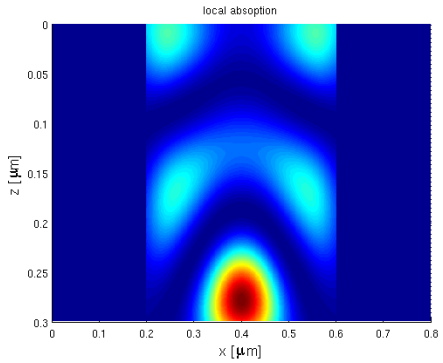
```
grating=0; % switch grating
Lambda=0.8; % grating period [um] (exception grating=10)
thickness_total=0.3; % total thickness [um] (exception grating=10)


n1=3.65; % homogenous incident region 1
n3=1; % homogenous region 3
ng=1; % Groove indicies, disabled when grating=10
nr=conj(sqrt(13.3+0.025*1i)); % Ridge indices, disabled when grating=10
```

```
duty_cycle=.5;
shift=.5;
```

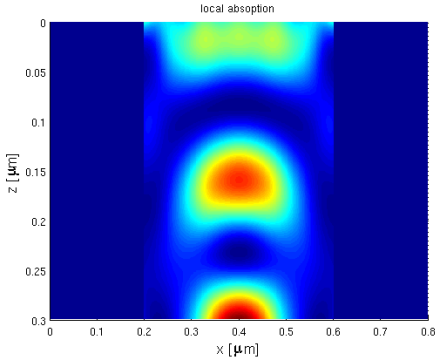
After setting

```
get_local_absoption=1;
```

following figures will apear:



TE polarization



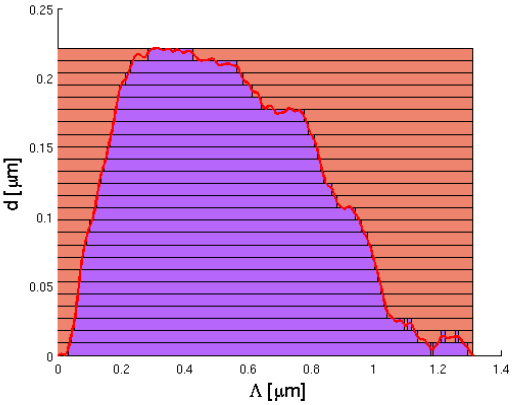
TM polarization

In case of TE polarization the integrated_absoption = 0.017370704494757 and the global_absorption = 0.017325005985810.

Real profiles

Script: create_real_profile.m

Since the RCWA-1D is able to solve gratings with arbitrary profiles, RCWA-1D can easily solve a grating as is shown here. For this use script *create_real_profile.m* to create an input file (see Grating from file).

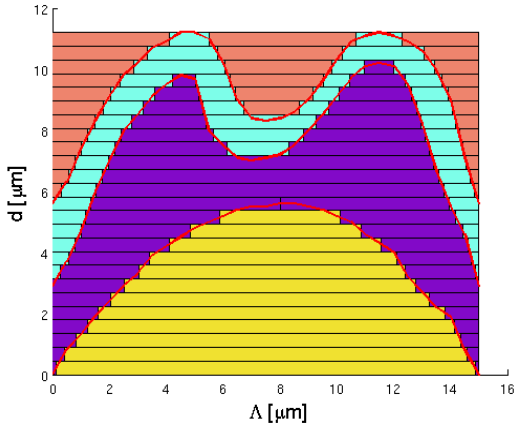



```
number_of_layers=25;
number_of_input_file=1;

save_result_file_as='Nm20f50_output';
save_result_file=1; % 1-yes, 2-no
plot_grating=1;

refractive_indices=[1 0.22-6.71*1i]; % from top of the grating
load_data_1='Nm20f50_um.txt';


filename_1=[load_data_1];
data(:,:,1)=dlmread(filename_1);
```

After running this script, you obtain a file *Nm20f50_output.m.* This file can be used as an input file for RCWA simulation (see Grating from file).

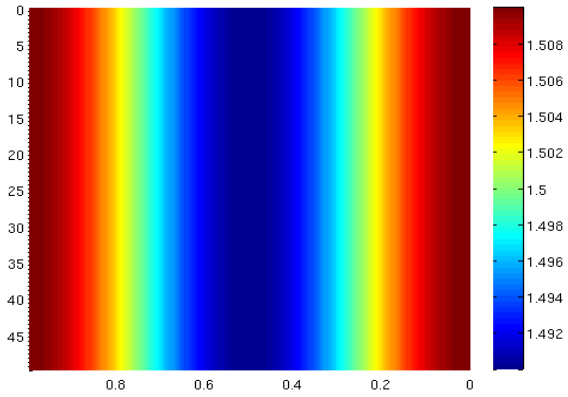
You can create more complicated structures ...

```
number_of_layers=25;
number_of_input_file=3;

save_result_file_as='output';
save_result_file=1; % 1-yes, 2-no
plot_grating=1;

refractive_indices=[1 1.2 1.5, 1.7];

load_data_1='profile_1.txt';
load_data_2='profile_2.txt';
load_data_3='profile_3.txt';

filename_1=[load_data_1];
data(:,:,1)=dlmread(filename_1);

filename_2=[load_data_2];
data(:,:,2)=dlmread(filename_2);

filename_3=[load_data_3];
data(:,:,3)=dlmread(filename_3);
```

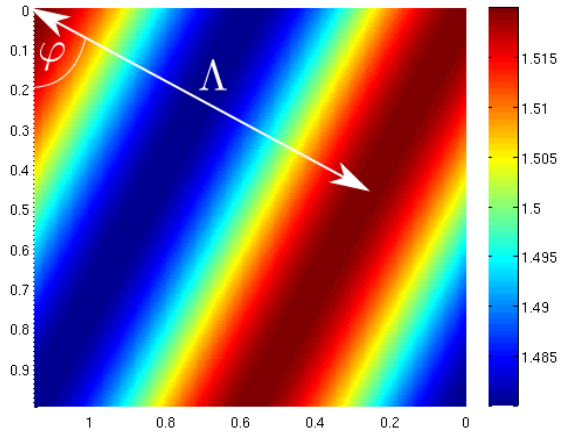


Volume gratings

Volume gratings can be solved using the discretization of the refractive index distribution. If lines of the refractive index are parallel to the z-axis, then use the script *volume_grating.m* , because the discretization is performed only in the x-axis direction, so the file generates the grating composed of one layer → quick simulation. You can easily modify a distribution of the refractive index.



If the angle phi is not equal to 90 degree, then use the script *general_volume_grating.m*. This file generates a grid of refractive index distribution, because of discretization in the x-axis direction and in the z-axis direction. Due to discretization in the z-axis direction → many layers, the calculation time can be long.



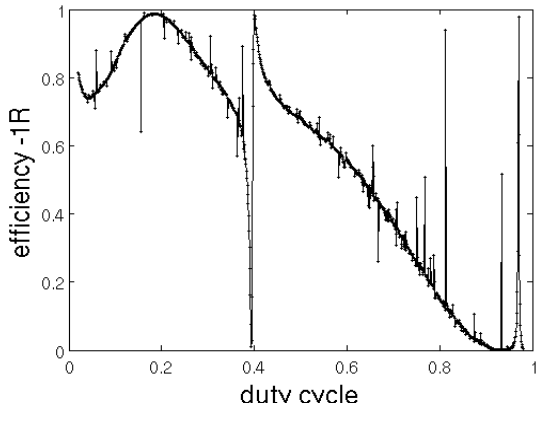
After generating the output file, the procedure is the same as in the case of "Grating from file".

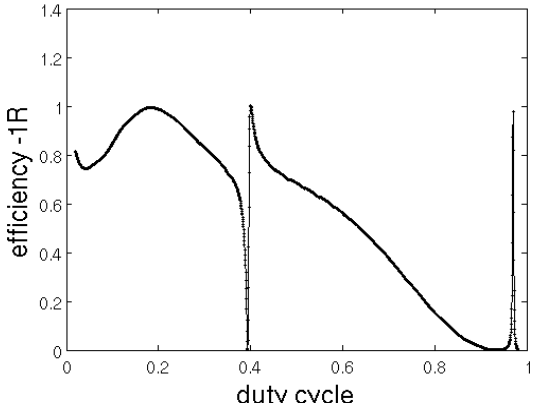
High-conductive gratings

When the RCWA method is applied to low-loss metallic gratings under TM incidence then numerical instabilities occur (left figure). Grating is made of a lossless metal with n=0-i*10, the grating period and depth are 500 nm, the wavelength is 632.8 nm, the number of orders is 31, and the incidence angle is 30º, the minus-first-order diffraction order diffraction efficiency is computed as a function of the duty cycle. Paper *Nikolay M. Lyndin, Olivier Parriaux, and Alexander V. Tishchenko, "Modal analysis and suppression of the Fourier modal method instabilities in highly conductive gratings," J. Opt. Soc. Am. A **24**, 3781-3788 (2007)* shows that these instabilities are caused by spurious modes. Mentioned paper have suggested to filter these spurious modes. Filtering is done in the file rcwa_procedure.m.

```
% spurious modes
   for k=1:1:number_of_orders
      if imag(Q(k,k,l))<-20
          Q(k,k,l)=Q(k,k,l)+1e40;
      end
   end
```

If `if imag(Q(k,k,l))<-20` then there is no filtering. Numerical instabilities will be erased when we set `if imag(Q(k,k,l))<-5` → right figure.





New measurement

This program is written in the Matlab/Octave environment, so there is a possibility to create new measurement. For example, you want to change the variable `notch_thickness_g` in the example **Grating from file**, It is very easy to do it, you have to insert new measurement in the file main.m

```
case 12
        minimum_notch_thickness_g=0.005; % [degree]
        maximum_notch_thickness_g=0.015; % [degree]
        step=0.001;
```

and insert following code (copy of the measurement 11) in the file help_control_file.m

```
case 12
        data_record_2=[measurement -(number_of_orders-1)/2:1:(number_of_orders-
1)/2 -(number_of_orders-1)/2:1:(number_of_orders-1)/2];
        fid = fopen('data12.dat', 'wt');
        for cyklus=minimum_notch_thickness_g :step: maximum_notch_thickness_g
            if use_dispersion==1
                setup_dispersion;
            end
            notch_thickness_g=cyklus;
            grating_setup;


            rcwa_procedure;

            if diffraction_efficiencies_c==1
                diffraction_efficiencies=D_R;
                symbol_R_or_T='R';
            else
                diffraction_efficiencies=D_T;
                symbol_R_or_T='T';
            end
            data_record_c=[cyklus D_R D_T];
            data_record_2=[data_record_2;data_record_c];
            data_record=[cyklus diffraction_efficiencies(p+studying_order)];
            fprintf(fid, '%0f %0f\n', data_record);
        end
        fclose(fid);
        save_all_measurements_results;
        figure;
        load data12.dat;
        a=data12;
        axes('FontSize',14);
        plot(a(:,1),a(:,2),'k','Marker','+','MarkerSize',3);
        ylabel(['efficiency ' num2str(studying_order)
symbol_R_or_T ],'FontSize',18);
        %title(['dependence of the diffraction efficiency '
num2str(studying_order) symbol_R_or_T  ' on the angle \phi'],'FontSize',20);
        delete('data12.dat')
```

Comparison of matrix algorithms

Program RCWA-1D provides two basic matrix algorithm: S-matrix (scattering matrix) algorithm and

enhanced T-matrix (transfer matrix) algorithm. Both algorithms are implemented in two versions. To change matrix algorithm set

```
matrix_algorithm=2; % 1-S-matrix, 2-T-matrix
change_matrix_base=1; % 1-s,p base, 2-x,y,z base
```

For speed comparison, I choose sinusoidal grating. Simulations were performed with 50 layers and 101 number of orders (Lenovo x200s, Core 2 Duo 1.86 GHz - 4 GB Ram, OpenSUSE 11.3, Matlab2011b)

1) conical diffraction

algorithm	base	total time [s]	procedure	field calculation
S-matrix	sp	61	S_Li_conical_final	yes, direct
S-matrix	xyz	40	S_matrix_W_S_grating_conical	no, use of S_Li_conical_final
T-matrix	sp	52	T_matrix_algorithm	yes, direct
T-matrix	xyz	43	T_matrix_algorithm	yes, direct

2) planar diffraction

algorithm	base	total time [s]	procedure	field calculation
S-matrix	sp	17	S_matrix_W_S_grating	no, use of S_Li_final
S-matrix	xyz	17	S_matrix_W_S_grating	no, use of S_Li_final
T-matrix	sp	20	T_matrix_algorithm	yes, direct
T-matrix	xyz	19	T_matrix_algorithm	yes, direct