Computational Photonics
Tutorial 4:
Gratings / Fourier modal method

21 June 2023

Dr. Markus Nyman, markus.nyman@kit.edu
M.Sc. Nigar Asadova, nigar.asadova@kit.edu

Exercises for everyone:

- Task 1: Eigenmodes of a grating layer
- Task 2: Multi-mode transfer matrix method, diffraction efficiencies

Extended exercises:

- Task 3*: Numerically stable formulation (S-matrix method)

Computational Photonics

Seminar 7, July 10, 2020

Fourier Modal Method (FMM)

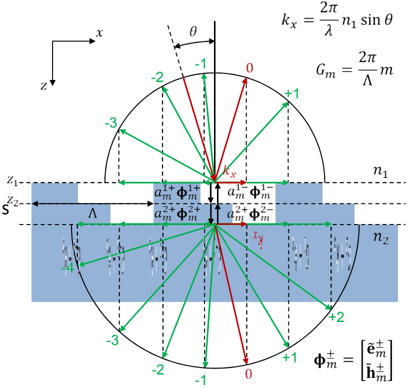
- Implementation of the 1D version of the Fourier mode solver in TE polarization

- Calculation of the diffraction efficiencies of a multi-layer grating in reflection and transmission

Abbe School of Photonics | JENA
Friedrich-Schiller-Universität

FMM fundamentals

- The grating is divided into z-invariant layers
- In each layer the Bloch modes are calculated in a Fourier basis
- The boundary value problem is solved by matching the continuous transverse electric and magnetic field components at the layer interfaces
- In the homogeneous regions the Bloch modes are plane waves and correspond to the reflected and transmitted diffraction orders



TE eigenmodes of a grating layer

- Each layer is z-invariant and periodic

$$\varepsilon(x,z) = \varepsilon(x) \text{ with } \varepsilon(x+\Lambda) = \varepsilon(x)$$

- Maxwell's equations for TE field components E_y, H_x and H_z

$$-\frac{\partial}{\partial_z} E_y = i\omega\mu_0 H_x, \qquad \frac{\partial}{\partial_x} E_y = i\omega\mu_0 H_z, \qquad \frac{\partial}{\partial_z} H_x - \frac{\partial}{\partial_x} H_z = -i\omega\varepsilon_0\varepsilon(x) E_y$$

- Combining yields Helmholtz equation

$$\frac{\partial^2}{\partial_x} E_y + \frac{\partial^2}{\partial_z} E_y + k_0^2 \varepsilon(x) E_y = 0$$

- Bloch modes

$$E_y(x,z) = e_y(x) e^{ik_x x + i\beta z} \text{ with } e_y(x+\Lambda) = e_y(x)$$

- Expansion of periodic quantities into Fourier series

$$e_y(x) = \sum_{m=-\infty}^{\infty} \tilde{e}_m e^{iG_m}, \qquad \varepsilon(x) = \sum_{m=-\infty}^{\infty} \tilde{\varepsilon}_m e^{iG_m}, \qquad G_m = m\frac{2\pi}{\Lambda}$$

TE eigenmodes of a grating layer

- Inserting into Helmholtz equation yields

$$\beta^2 \sum_m \tilde{e}_m e^{iG_m x} = k_0^2 \sum_m \sum_n \tilde{\varepsilon}_m \tilde{e}_n e^{i(G_m+G_n)x} - \sum_m (G_m + k_x)^2 \tilde{e}_m e^{iG_m x}$$

- For all x $\rightarrow$ has to hold for each grating vector G_m individually

$$\beta^2 \tilde{e}_m e^{iG_m x} = k_0^2 \sum_n \tilde{\varepsilon}_{m-n} \tilde{e}_n e^{i(G_{m-n}+G_n)x} - (G_m + k_x)^2 \tilde{e}_m e^{iG_m x}$$

- Truncating to $m \in [-N, N]$ yields an algebraic eigenvalue problem

$$\beta^2 \boldsymbol{\Phi}_\mathrm{e} = \left[k_0^2 \hat{\boldsymbol{\varepsilon}} - \widehat{\mathbf{K}}^2\right] \boldsymbol{\Phi}_\mathrm{e} = \widehat{\mathbf{M}} \boldsymbol{\Phi}_\mathrm{e}$$

TE eigenmodes of a grating layer

- Algebraic eigenvalue problem

$$\beta^2 \boldsymbol{\Phi}_\mathrm{e} = \left[k_0^2 \hat{\boldsymbol{\varepsilon}} - \widehat{\mathbf{K}}^2\right] \boldsymbol{\Phi}_\mathrm{e} = \widehat{\mathbf{M}} \boldsymbol{\Phi}_\mathrm{e}$$

- The Fourier components are stored in a column vector

$$\boldsymbol{\Phi}_\mathrm{e} = \begin{bmatrix} \tilde{e}_{-N} \\ \vdots \\ \tilde{e}_N \end{bmatrix}$$

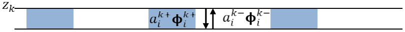
- $\hat{\boldsymbol{\varepsilon}}$ is a Toeplitz matrix where $\hat{\varepsilon}_{mn} = \tilde{\varepsilon}_{m-n}$

$$\hat{\varepsilon} = \begin{bmatrix} \tilde{\varepsilon}_0 & \tilde{\varepsilon}_{-1} & \cdots & \tilde{\varepsilon}_{-2N} \\ \tilde{\varepsilon}_1 & \tilde{\varepsilon}_0 & \cdots & \tilde{\varepsilon}_{-2N+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\varepsilon}_{2N} & \tilde{\varepsilon}_{2N-1} & \cdots & \tilde{\varepsilon}_0 \end{bmatrix}$$

- $\widehat{\mathbf{K}} = k_x \mathbb{I} + \mathrm{diag}[G_{-N}, \cdots, G_N]$ is a diagonal matrix

$$\widehat{\mathbf{K}} = \begin{bmatrix} G_{-N} + k_x & 0 & \cdots & 0 \\ 0 & G_{-N+1} + k_x & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & G_N + k_x \end{bmatrix}$$

TE eigenmodes of a grating layer

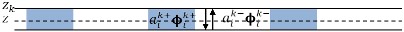


- $\widehat{\mathbf{M}}$ has $2N+1$ rows and columns → $2N+1$ eigenvalues and eigenvectors
- Each eigenvector $\boldsymbol{\Phi}_{\mathrm{e},i}^{k}$ contains the Fourier components of the periodic part of the electric field $e_{y,i}^{k}(x)$ of the i^{th} mode in layer k
- The total field is decomposed into a superposition of **forwards** (+) and **backwards** (-) propagating eigenmodes with amplitudes a_i^{k+} and a_i^{k-}
- The Fourier components $\boldsymbol{\Phi}_{\mathrm{h},i}^{k\pm}$ of the periodic part of the normalized transverse magnetic field $-i\omega\mu_0 h_{x,i}^{k}(x)$ of the forwards (+) and backwards (-) modes in layer k are given by

$$-i\omega\mu_0 H_x = \frac{\partial}{\partial_z} E_y \rightarrow \boldsymbol{\Phi}_{\mathrm{h},i}^{k\pm} = \pm\beta_i^k \boldsymbol{\Phi}_{\mathrm{e},i}^{k\pm}$$

- β_i^k is the propagation constant of the i^{th} **forwards** mode in layer k
- The electric Fourier components of the forwards and backwards modes are the same $\boldsymbol{\Phi}_{\mathrm{e},i}^{k-} = \boldsymbol{\Phi}_{\mathrm{e},i}^{k+} = \boldsymbol{\Phi}_{\mathrm{e},i}^{k}$

TE eigenmodes of a grating layer



- The Fourier components of the total electromagnetic field at position z in layer k (starting at $z = z_k$) are given by

$$\begin{bmatrix} \tilde{\mathbf{e}}^k(z) \\ \tilde{\mathbf{h}}^k(z) \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_\mathrm{e}^k & \widehat{\boldsymbol{\Phi}}_\mathrm{e}^k \\ \widehat{\boldsymbol{\Phi}}_\mathrm{e}^k \widehat{\boldsymbol{\beta}}^k & -\widehat{\boldsymbol{\Phi}}_\mathrm{e}^k \widehat{\boldsymbol{\beta}}^k \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{p}}^{k+}(z - z_k) & 0 \\ 0 & \widehat{\mathbf{p}}^{k-}(z - z_k) \end{bmatrix} \begin{bmatrix} \mathbf{a}^{k+} \\ \mathbf{a}^{k-} \end{bmatrix}$$

- With

$$\mathbf{a}^{k\pm} = \left[\mathrm{a}_1^{k\pm}, \cdots, \mathrm{a}_{2N+1}^{k\pm}\right]^T$$

$$\widehat{\boldsymbol{\Phi}}_\mathrm{e}^k = \left[\boldsymbol{\Phi}_{\mathrm{e},1}^k, \cdots, \boldsymbol{\Phi}_{\mathrm{e},2N+1}^k\right] = \begin{bmatrix} \tilde{e}_{-N,1}^k & \cdots & \tilde{e}_{-N,2N+1}^k \\ \vdots & & \vdots \\ \tilde{e}_{N,1}^k & \cdots & \tilde{e}_{N,2N+1}^k \end{bmatrix},$$

$$\widehat{\boldsymbol{\beta}}^k = \mathrm{diag}\left[\beta_1^k, \cdots, \beta_{2N+1}^k\right]$$

$$\widehat{\mathbf{p}}^{k\pm}(z) = \mathrm{diag}\left[e^{\pm i\beta_1^k z}, \cdots, e^{\pm i\beta_{2N+1}^k z}\right]$$

TE eigenmodes of a grating layer

- The regions before and after the grating are homogeneous

$$\varepsilon(x,z)=\varepsilon=\mathrm{const}$$

- The eigenmodes of the homogeneous regions are plane waves and correspond to the reflected and transmitted diffraction orders:

$$\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}=\mathbb{I},\qquad \widehat{\boldsymbol{\beta}}=\sqrt{k_0^2\varepsilon\mathbb{I}-\widehat{\mathbf{K}}^2}$$

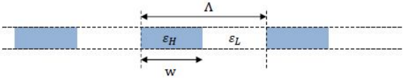
Tasks

1. Implement the function **fmm1d_TE_layer_modes** that calculates the TE eigenmodes of a single grating layer in the Fourier domain.

2. Solve the boundary value problem in order to calculate the amplitudes and diffraction efficiencies of the transmitted and reflected diffraction order (function **fmm1d_te**).

Task 1: Layer modes

- Implement a function that calculates the Bloch modes of a grating layer with a binary permittivity distribution
- Handle homogeneous layers explicitly to ensure the expected ordering of the modes (test if permittivity is a scalar or if all entries of the vector are equal)



- Test with the following parameters:
  - $\lambda = 1.064\ \mu\text{m}, k_x = 0$
  - $\Lambda = 3\ \mu\text{m}, w = 1.5\ \mu\text{m}$
  - $\varepsilon_L = 1, \varepsilon_H = 4$
  - Use $N = 25$ for the Fourier expansion of the electric Field
  - Sample the spatial permittivity distribution with $N_x = 1001$ points
- Calculate the spatial field distributions $e_{y,i}(x)$ of the propagating modes

Task 1: Layer modes

```
def fmm1d_te_layer_modes(perm, period, k0, kx, N):
    '''Calculates the TE eigenmodes of a one-dimensional grating layer.

    Arguments
    ---------
        perm: 1d-array
            permittivity distribution
        period: float
            grating period
        k0: float
            vacuum wavenumber
        kx: float
            transverse wave vector
        N: int
            number of positive Fourier orders

    Returns
    -------
        beta: 1d-array
            propagation constants of the eigenmodes
        phie: 2d-array
            Fourier coefficients of the eigenmodes (each column corresponds to one mode)
    '''
    pass
```

Task 1: Implementation hints

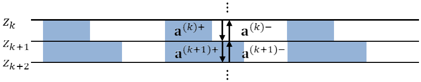
- The spatial coordinates are `x = (0:Nx-1)/Nx*period`
- The Fourier expansion of the permittivity is given by
  `perm_ft = fft(perm)/Nx`
  assuming `Nx = length(perm)`, dividing by Nx ensures that `perm_ft` is scaled correctly , i.e. that its first element is equal to the average of `perm`
  `perm_ft(1) == mean(perm)`
- Keep in mind that in Numpy/Scipy the vector returned by `fft()` contains first the positive, then negative frequencies:
$$\hat{\varepsilon}_m = [\hat{\varepsilon}_0, \hat{\varepsilon}_1, \cdots, \hat{\varepsilon}_{n^+}, \hat{\varepsilon}_{-n^-}, \cdots, \hat{\varepsilon}_{-1}]$$
$$n^+ = \lfloor (N_x - 1)/2 \rfloor$$
$$n^- = \lceil (N_x - 1)/2 \rceil$$
- Make sure that $n^+ = \lfloor (N_x - 1)/2 \rfloor \geq 2N$, otherwise you will not have enough Fourier components to fill the Toeplitz matrix
- Don't use `fftshift()` unless you know exactly what you are doing
- For the construction of the matrix, you can use the function `toeplitz()` in `scipy.linalg`

Task 1: Implementation hints

- In Python block matrices can be assembled using either the function `numpy.block()` or you have to combine `numpy.hstack()` and `numpy.vstack()`

- When you take the square root of the eigenvalues β_i^2 to calculate the propagation constants β_i of the modes, make sure that the obtained propagation constants belong to forward modes, i.e. $\Re\{\beta_i\} + \Im\{\beta_i\} > 0$, otherwise reverse β_i

Boundary value problem

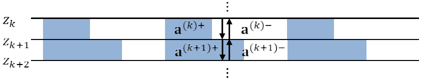


- Problem:
  - N layers, incident wave $\mathbf{a}^{(0)+}$ is known
  - Wanted: amplitudes of the reflected and transmitted modes $\mathbf{R} = \mathbf{a}^{(0)-}$ and $\mathbf{T} = \mathbf{a}^{(N+1)+}$
- At the interface between two layers the transverse electric and magnetic fields are continuous ($d_k = z_{k+1} - z_k$)

$$\begin{bmatrix} \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k+1)} & \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k+1)} \\ \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k+1)}\widehat{\boldsymbol{\beta}}^{(k+1)} & -\widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k+1)}\widehat{\boldsymbol{\beta}}^{(k+1)} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{p}}^{(k+1)+}(0) & 0 \\ 0 & \widehat{\mathbf{p}}^{(k+1)-}(0) \end{bmatrix} \begin{bmatrix} \mathbf{a}^{(k+1)+} \\ \mathbf{a}^{(k+1)-} \end{bmatrix}$$
$$= \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k)} & \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k)} \\ \widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k)}\widehat{\boldsymbol{\beta}}^{(k)} & -\widehat{\boldsymbol{\Phi}}_\mathrm{e}^{(k)}\widehat{\boldsymbol{\beta}}^{(k)} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{p}}^{(k)+}(d_k) & 0 \\ 0 & \widehat{\mathbf{p}}^{(k)-}(d_k) \end{bmatrix} \begin{bmatrix} \mathbf{a}^{(k)+} \\ \mathbf{a}^{(k)-} \end{bmatrix}$$

- $\widehat{\mathbf{p}}^{(k+1)\pm}(0) = \mathbb{I}$

Boundary value problem



- Interface transfer matrix

$$\hat{\mathbf{t}}(k,k+1)=\begin{bmatrix}\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k+1)} & \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k+1)} \\ \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k+1)}\widehat{\boldsymbol{\beta}}^{(k+1)} & -\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k+1)}\widehat{\boldsymbol{\beta}}^{(k+1)}\end{bmatrix}^{-1}\begin{bmatrix}\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k)} & \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k)} \\ \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k)}\widehat{\boldsymbol{\beta}}^{(k)} & -\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{(k)}\widehat{\boldsymbol{\beta}}^{(k)}\end{bmatrix}$$

- Layer transfer matrix

$$\widehat{\mathbf{T}}(k,k+1)=\hat{\mathbf{t}}(k,k+1)\begin{bmatrix}\widehat{\mathbf{p}}^{(k)+}(d_k) & 0 \\ 0 & \widehat{\mathbf{p}}^{(k)-}(d_k)\end{bmatrix}$$

- With

$$\begin{bmatrix}\mathbf{a}^{(k+1)+} \\ \mathbf{a}^{(k+1)-}\end{bmatrix}=\widehat{\mathbf{T}}(k,k+1)\begin{bmatrix}\mathbf{a}^{(k)+} \\ \mathbf{a}^{(k)-}\end{bmatrix}$$

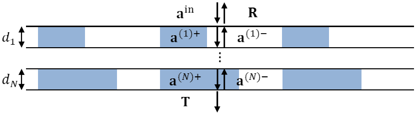
- **Transfer matrices can be stacked** (Pay attention to the correct multiplication order! Matrix multiplication is not commutative.)

$$\widehat{\mathbf{T}}(0,k+1)=\widehat{\mathbf{T}}(k,k+1)\widehat{\mathbf{T}}(0,k)$$

Transfer matrix algorithm (T-matrix algorithm)

- Start in the homogeneous layer before the grating
  - $\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{0} = \mathbb{I}_{2N+1},\ \widehat{\boldsymbol{\beta}}^{0} = \sqrt{k_0^2 \varepsilon_{\mathrm{in}} \mathbb{I}_{2N+1} - \widehat{\mathbf{K}}^2},\ d_0 = 0$
  - $\widehat{\mathbf{T}}(0,0) = \mathbb{I}_{2(2N+1)}$
- For each layer $i = 1 \dots N$
  - Calculate layer modes and propagation constants $\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{i}$ and $\widehat{\boldsymbol{\beta}}^{i}$
  - Calculate layer T-matrix $\widehat{\mathbf{T}}(i-1,i) \leftarrow (\widehat{\boldsymbol{\beta}}^{i-1}, \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{i-1}, d_{i-1}, \widehat{\boldsymbol{\beta}}^{i}, \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{i})$
  - Update stack T-matrix $\widehat{\mathbf{T}}(0,i) = \widehat{\mathbf{T}}(i-1,i)\widehat{\mathbf{T}}(0,i-1)$
- In the homogeneous layer after the grating
  - $\widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{N+1} = \mathbb{I}_{2N+1},\ \widehat{\boldsymbol{\beta}}^{N+1} = \sqrt{k_0^2 \varepsilon_{\mathrm{out}} \mathbb{I}_{2N+1} - \widehat{\mathbf{K}}^2},\ d_{N+1} = 0$
  - Calculate layer T-matrix $\widehat{\mathbf{T}}(N,N+1) \leftarrow (\widehat{\boldsymbol{\beta}}^{N}, \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{N}, d_N, \widehat{\boldsymbol{\beta}}^{N+1}, \widehat{\boldsymbol{\Phi}}_{\mathrm{e}}^{N+1})$
  - Update stack T-matrix $\widehat{\mathbf{T}}(0,N+1) = \widehat{\mathbf{T}}(N,N+1)\widehat{\mathbf{T}}(0,N)$

Reflection and transmission coefficients



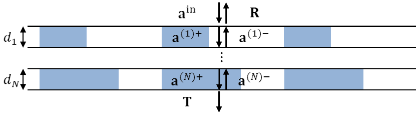
- Incident waves:
  - $\mathbf{a}^{(0)+} = \mathbf{a}^{\mathrm{in}}$, $a_m^{\mathrm{in}} = \delta_{mN+1}$
  - $\mathbf{a}^{(N+1)-} = 0$
- Reflected and transmitted diffraction orders

$$\mathbf{R} = \mathbf{a}^{(0)-}, \qquad \mathbf{T} = \mathbf{a}^{(N+1)+}$$

$$\begin{bmatrix} \mathbf{T} \\ 0 \end{bmatrix} = \hat{\mathbf{T}}(0, N+1) \begin{bmatrix} \mathbf{a}^{\mathrm{in}} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{t}}_{11} & \hat{\mathbf{t}}_{12} \\ \hat{\mathbf{t}}_{21} & \hat{\mathbf{t}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{a}^{\mathrm{in}} \\ \mathbf{R} \end{bmatrix}$$

- Solving for $\mathbf{R}$ and $\mathbf{T}$ yields
  - $\mathbf{R} = -\hat{\mathbf{t}}_{22}^{-1} \hat{\mathbf{t}}_{21} \mathbf{a}^{\mathrm{in}}$
  - $\mathbf{T} = (\hat{\mathbf{t}}_{11} - \hat{\mathbf{t}}_{12} \hat{\mathbf{t}}_{22}^{-1} \hat{\mathbf{t}}_{21}) \mathbf{a}^{\mathrm{in}}$

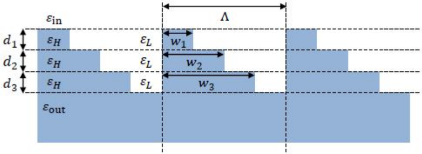
Diffraction efficiencies



- Reflection and transcoefficients
  - $\mathbf{R} = -\hat{\mathbf{t}}_{22}^{-1}\hat{\mathbf{t}}_{21}\mathbf{a}^{\mathrm{in}}$
  - $\mathbf{T} = (\hat{\mathbf{t}}_{11} - \hat{\mathbf{t}}_{12}\hat{\mathbf{t}}_{22}^{-1}\hat{\mathbf{t}}_{21})\mathbf{a}^{\mathrm{in}}$
- Diffraction efficiencies of the m^{th} diffraction order
  - z-component of Poynting vector in Fourier space: $\tilde{S}_{z,m} \propto \Re\{\beta_m\}|\tilde{e}_m|^2$
  - $\eta_{R/T,m} = \tilde{S}_{z,m}^{R/T} / \tilde{S}_z^{\mathrm{in}}$
  - $\boldsymbol{\eta}_R = \frac{1}{\Re\{\beta^{\mathrm{in}}\}}\Re\{\widehat{\boldsymbol{\beta}}^{(0)}\}(\mathbf{R} \circ \mathbf{R}^*)$
  - $\boldsymbol{\eta}_T = \frac{1}{\Re\{\beta^{\mathrm{in}}\}}\Re\{\widehat{\boldsymbol{\beta}}^{(N+1)}\}(\mathbf{T} \circ \mathbf{T}^*)$
  - $(\mathbf{A} \circ \mathbf{A})$ means elementwise product of $\mathbf{A}$ and $\mathbf{B}$ (Hadamard product)

Task 2: Diffraction efficiencies of a multilayer grating

- Implement a function that calculates the TE diffraction efficiencies and reflection and transmission coefficients of a multilayer grating with the T-matrix algorithm



- Calculate the diffraction efficiencies for the following parameters
  - $\lambda = 1.064\ \mu\text{m}, \theta \in \{-30°, 0°, 30°\}$
  - $\Lambda = 3\ \mu\text{m}, w_i = i \cdot \Lambda/4, d_i = 0.25\ \mu\text{m}$
  - $\varepsilon_{\text{in}} = \varepsilon_L = 1, \varepsilon_{\text{out}} = \varepsilon_H = 4$
  - Use $N = 20$ for the Fourier expansion of the electric Field
  - Sample the spatial permittivity distribution with $N_x = 1001$ points

Task 2: Diffraction efficiencies of a multilayer grating

Side questions to examine:

- What happens when
  - the number of positive Fourier components is increased to $N = 40$
  - the layer thicknesses are increased to $d_i = 0.5\ \mu\mathrm{m}$
- Why does this happen?

Task 2: Diffraction efficiencies of a multilayer grating

```python
def fmm1d_te(lam, theta, period, perm_in, perm_out, layer_perm, layer_ticknesses, N):
    '''Calculates the TE diffraction efficiencies for a 1D layered grating using the T-matrix method.

    Arguments
    ---------
        lam: float
            vacuum wavelength
        theta: float
            angle of incidence in rad
        period: float
            grating period
        perm_in: float
            permittivity on the incidence side
        perm_out: float
            permittivity on the exit side
        layer_perm: 2d-array
            permittivity distribution within the grating layers (matrix, each row corresponds to one layer)
        layer_thicknesses: 1d-array
            thicknesses of the grating layers
        N: int
            number of positive Fourier orders

    Returns
    -------
        eta_r: 1d-array
            diffraction efficiencies of the reflected diffraction orders
        eta_t: 1d-array
            diffraction efficiencies of the transmitted diffraction orders
        r: 1d-array
            amplitude reflection coefficients of the reflected diffraction orders
        t: 1d-array
            amplitude transmission coefficients of the transmitted diffraction orders
    '''
    pass
```

Task 2: Implementation hints

- Use plain numpy arrays, don‘t use numpy‘s matrix class
- $A^{-1}B$ corresponds to `numpy.linalg.solve(A, B)`
- BA^{-1} corresponds to `numpy.linalg.solve(A.T, B.T).T`
- AB corresponds to A@B
- Changing an 1d-array to a 2d column: `b_as_column = b[:, numpy.newaxis]`
- Changing an 1d-array to a 2d row: `b_as_row = b[numpy.newaxis, :]`
- Matrix-vector multiplication of 2d-array A and 1d-array b corresponds to `A@b[:, numpy.newaxis]`
- The block matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ can be built with `numpy.block([[A, B], [C, D]])` or

```
numpy.vstack([numpy.hstack([A, B]),
              numpy.hstack([C, D])])
```

Task 3*: Numerically stable formulation (S-matrix method)

The transfer matrix method implemented in the previous tasks has $\exp(\pm i\beta d)$ factors that blow up / diminish to nothing for large d, many layers and highly evanescent waves. This leads to loss of precision and large numerical errors for demanding grating structures.
The S-matrix method is more stable. Read the next couple of slides and implement and test the method using the same test case as in Task 2.

If you're interested in a more detailed discussion, this method is explained in Section 3 of the provided paper [1]. [1] Lifeng Li, "Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings", *J. Opt. Soc. Am. A* **13**, 1024, 1996.

S-matrix method

Definition of S-matrix: (outgoing fields) = $\mathbf{S}$ (incoming fields):

$$\begin{bmatrix} a^{2+} \\ a^{1-} \end{bmatrix} = \mathbf{S} \begin{bmatrix} a^{1+} \\ a^{2-} \end{bmatrix} \tag{1}$$

Putting two S-matrices A and B together (B comes after A); unlike transfer matrices, S-matrices cannot be stacked just by matrix multiplication):

$$\mathbf{S} = \begin{bmatrix} B_{11}(I - A_{12}B_{21})^{-1}A_{11} & B_{12} + B_{11}A_{12}(I - B_{21}A_{12})^{-1}B_{22} \\ A_{21} + A_{22}B_{21}(I - A_{12}B_{21})^{-1}A_{11} & A_{22}(I - B_{21}A_{12})^{-1}B_{22} \end{bmatrix} \tag{2}$$

where A_{11}, A_{12} etc. are the upper left, upper right etc. blocks (see next slide). To calculate the transmission and reflection, set $a^{1+} = a_{\text{inc}}$ and $a^{2-} = 0$, and we immediately get

$$t = S_{11} a_{\text{inc}} \tag{3}$$

$$r = S_{21} a_{\text{inc}} \tag{4}$$

S-matrices

S-matrix of an interface between media 1 and 2:

$$\mathbf{S} = \begin{bmatrix} \mathbf{\Phi}_2 & -\mathbf{\Phi}_1 \\ \mathbf{\Phi}_2\beta_2 & \mathbf{\Phi}_1\beta_1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Phi}_1 & -\mathbf{\Phi}_2 \\ \mathbf{\Phi}_1\beta_1 & \mathbf{\Phi}_2\beta_2 \end{bmatrix} \tag{5}$$

where $\mathbf{\Phi}$ and β are blocks defined as in the T-matrix method.
S-matrix of propagation in a medium of thickness d

$$\begin{bmatrix} e^{i\beta d} & 0 \\ 0 & e^{i\beta d} \end{bmatrix} \tag{6}$$

Note that the $\exp(-i\beta d)$ factor is not present, which helps explain why this method is not as vulnerable to loss of precision.