Computational Photonics
Tutorial 3:
Finite difference time domain method

24 May 2023

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

Exercises for everyone:

- Task I(1,2,3): FDTD implementation in one spatial dimension

Extended exercises:

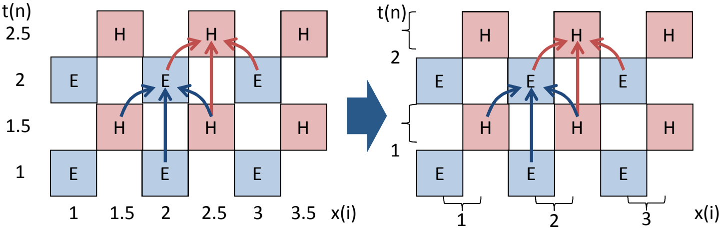
- Task II*(1,2,3): FDTD implementation in three spatial dimensions

Advice

- Do the arithmetic with numpy vectors. Avoid for-looping over the space index / indices, because this is much slower in non-accelerated Python.
- Indexing expressions like `E[1:]` and `E[0:-1]` and functions like `numpy.concatenate` may be in useful in manipulating the arrays.
- The 1D case can be computed in a fraction of a second. The 3D case may take a quite few seconds.

1D FDTD: Yee – Grid for E_z & H_y Components

Changing of index notation to integer indices



start at n=1:

$$E_z\Big|_i^{n+1} \approx E_z\Big|_i^{n} + \frac{1}{\varepsilon_0\varepsilon_i}\frac{\Delta t}{\Delta x}\left[H_y\Big|_{i+\frac{1}{2}}^{n+\frac{1}{2}} - H_y\Big|_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right] - \frac{\Delta t}{\varepsilon_0\varepsilon_i} j_z\Big|_i^{n+\frac{1}{2}}$$

$$H_y\Big|_{i+\frac{1}{2}}^{n+\frac{3}{2}} \approx H_y\Big|_{i+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{\mu_0}\frac{\Delta t}{\Delta x}\left[E_z\Big|_{i+1}^{n+1} - E_z\Big|_i^{n+1}\right]$$

n+½ → n

start at n=1 ($E^1=0$, $H^1=0$) :

$$E_z\Big|_i^{n+1} \approx E_z\Big|_i^{n} + \frac{1}{\varepsilon_0\varepsilon_i}\frac{\Delta t}{\Delta x}\left[H_y\Big|_{i}^{n} - H_y\Big|_{i-1}^{n}\right] - \frac{\Delta t}{\varepsilon_0\varepsilon_i} j_z\Big|_i^{n}$$

$$H_y\Big|_{i}^{n+1} \approx H_y\Big|_{i}^{n} + \frac{1}{\mu_0}\frac{\Delta t}{\Delta x}\left[E_z\Big|_{i+1}^{n+1} - E_z\Big|_i^{n+1}\right]$$

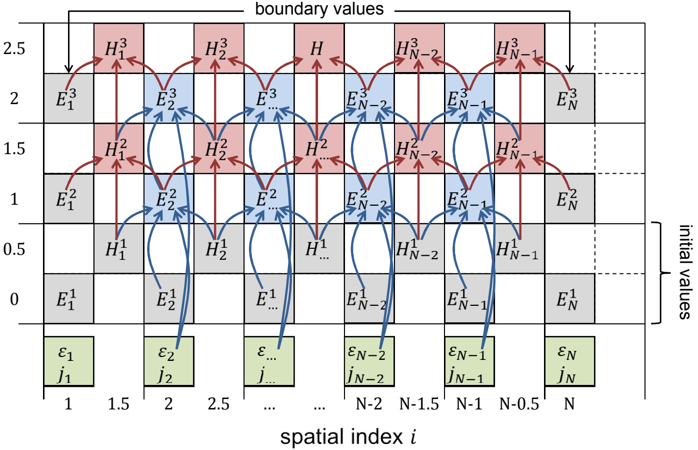
1D FDTD: Source

- Separable source:

$$j_z|_i^n = A\big(\Delta t(n+1/2)\big)e^{-2\pi i f \Delta t(n+1/2)} j_z(t=0)\Big|_i$$

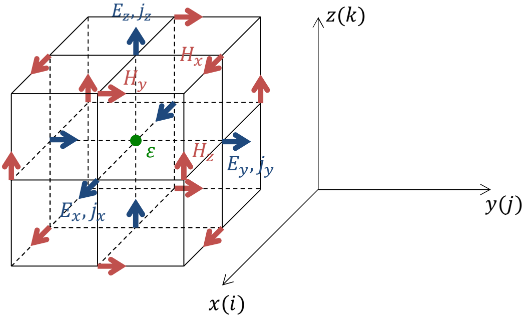
- Spatial distribution: $j_z\,(t=0)|_i$
- Carrier $e^{-2\pi i f \Delta t(n+1/2)}$
- Envelope: $A(\Delta t(n+1/2))$

1D FDTD: Layout of the field arrays



3D FDTD: Yee-Grid

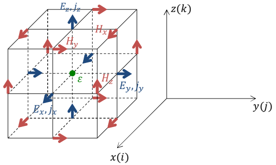
Center of the cube is in the center of the coordinate system (i, j, k)



Grid size is determined by the permittivity distribution:

$$\mathrm{size}(\varepsilon) = [N_x, N_y, N_z]$$

3D FDTD: Electric Field Components



Permittivity must be interpolated:

$$\frac{1}{\varepsilon_{i+0.5,j,k}} = \frac{1}{2}\left(\frac{1}{\varepsilon_{i,j,k}} + \frac{1}{\varepsilon_{i+1,j,k}}\right)$$

$$\frac{1}{\varepsilon_{i,j+0.5,k}} = \frac{1}{2}\left(\frac{1}{\varepsilon_{i,j,k}} + \frac{1}{\varepsilon_{i,j+1,k}}\right)$$

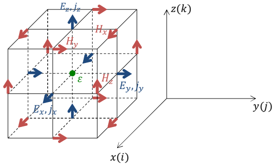
$$\frac{1}{\varepsilon_{i,j,k+0.5}} = \frac{1}{2}\left(\frac{1}{\varepsilon_{i,j,k}} + \frac{1}{\varepsilon_{i,j,k+1}}\right)$$

$$E_x\Big|_{i+0.5,j,k}^{n+1} = E_x\Big|_{i+0.5,j,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_{i+0.5,j,k}}\left(\frac{H_z\Big|_{i+0.5,j+0.5,k}^{n+0.5} - H_z\Big|_{i+0.5,j-0.5,k}^{n+0.5}}{\Delta y} - \frac{H_y\Big|_{i+0.5,j,k+0.5}^{n+0.5} - H_y\Big|_{i+0.5,j,k-0.5}^{n+0.5}}{\Delta z} - j_x\Big|_{i+0.5,j,k}^{n+0.5}\right)$$

$$E_y\Big|_{i,j+0.5,k}^{n+1} = E_y\Big|_{i,j+0.5,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_{i,j+0.5,k}}\left(\frac{H_x\Big|_{i,j+0.5,k+0.5}^{n+0.5} - H_x\Big|_{i,j+0.5,k-0.5}^{n+0.5}}{\Delta z} - \frac{H_z\Big|_{i+0.5,j+0.5,k}^{n+0.5} - H_z\Big|_{i-0.5,j+0.5,k}^{n+0.5}}{\Delta x} - j_y\Big|_{i,j+0.5,k}^{n+0.5}\right)$$

$$E_z\Big|_{i,j,k+0.5}^{n+1} = E_z\Big|_{i,j,k+0.5}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_{i,j,k+0.5}}\left(\frac{H_y\Big|_{i+0.5,j,k+0.5}^{n+0.5} - H_y\Big|_{i-0.5,j,k+0.5}^{n+0.5}}{\Delta x} - \frac{H_x\Big|_{i,j+0.5,k+0.5}^{n+0.5} - H_x\Big|_{i,j-0.5,k+0.5}^{n+0.5}}{\Delta y} - j_z\Big|_{i,j,k+0.5}^{n+0.5}\right)$$

3D FDTD: Magnetic Field Components

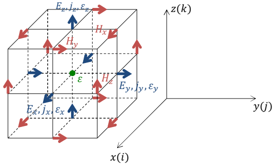


$$H_x\Big|_{i,j+0.5,k+0.5}^{n+1.5} = H_x\Big|_{i,j+0.5,k+0.5}^{n+0.5} + \frac{\Delta t}{\mu_0}\left(\frac{E_y\Big|_{i,j+0.5,k+1}^{n+1} - E_y\Big|_{i,j+0.5,k}^{n+1}}{\Delta z} - \frac{E_z\Big|_{i,j+1,k+0.5}^{n+1} - E_z\Big|_{i,j,k+0.5}^{n+1}}{\Delta y} \right)$$

$$H_y\Big|_{i+0.5,j,k+0.5}^{n+1.5} = H_y\Big|_{i+0.5,j,k+0.5}^{n+0.5} + \frac{\Delta t}{\mu_0}\left(\frac{E_z\Big|_{i+1,j,k+0.5}^{n+1} - E_z\Big|_{i,j,k+0.5}^{n+1}}{\Delta x} - \frac{E_x\Big|_{i+0.5,j,k+1}^{n+1} - E_x\Big|_{i+0.5,j,k}^{n+1}}{\Delta z} \right)$$

$$H_z\Big|_{i+0.5,j+0.5,k}^{n+1.5} = H_z\Big|_{i+0.5,j+0.5,k}^{n+0.5} + \frac{\Delta t}{\mu_0}\left(\frac{E_x\Big|_{i+0.5,j+1,k}^{n+1} - E_x\Big|_{i+0.5,j,k}^{n+1}}{\Delta y} - \frac{E_y\Big|_{i+1,j+0.5,k}^{n+1} - E_y\Big|_{i,j+0.5,k}^{n+1}}{\Delta x} \right)$$

3D FDTD: Electric Field Components
Change Index Notation to Integer Indices



Renaming of fractional indices:

$$i+0.5 \to i$$
$$j+0.5 \to j$$
$$k+0.5 \to k$$

Renaming of interpolated permittivity:

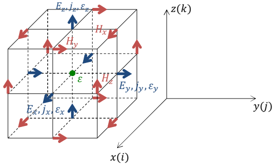
$$\varepsilon_{i+0.5,j,k} \to \varepsilon_x\big|_{i,j,k}$$
$$\varepsilon_{i,j+0.5,k} \to \varepsilon_y\big|_{i,j,k}$$
$$\varepsilon_{i,j,k+0.5} \to \varepsilon_z\big|_{i,j,k}$$

$$E_x\big|_{i,j,k}^{n+1} = E_x\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_x\big|_{i,j,k}} \left(\frac{H_z\big|_{i,j,k}^{n} - H_z\big|_{i,j-1,k}^{n}}{\Delta y} - \frac{H_y\big|_{i,j,k}^{n} - H_y\big|_{i,j,k-1}^{n}}{\Delta z} - j_x\big|_{i,j,k}^{n} \right)$$

$$E_y\big|_{i,j,k}^{n+1} = E_y\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_y\big|_{i,j,k}} \left(\frac{H_x\big|_{i,j,k}^{n} - H_x\big|_{i,j,k-1}^{n}}{\Delta z} - \frac{H_z\big|_{i,j,k}^{n} - H_z\big|_{i-1,j,k}^{n}}{\Delta x} - j_y\big|_{i,j,k}^{n} \right)$$

$$E_z\big|_{i,j,k}^{n+1} = E_z\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_z\big|_{i,j,k}} \left(\frac{H_y\big|_{i,j,k}^{n} - H_y\big|_{i-1,j,k}^{n}}{\Delta x} - \frac{H_x\big|_{i,j,k}^{n} - H_x\big|_{i,j-1,k}^{n}}{\Delta y} - j_z\big|_{i,j,k}^{n} \right)$$

3D FDTD: Magnetic Field Components
Change Index Notation to Integer Indices



Renaming of fractional indices:

$$i+0.5 \rightarrow i$$
$$j+0.5 \rightarrow j$$
$$k+0.5 \rightarrow k$$

$$H_x\Big|_{i,j,k}^{n+1} = H_x\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_y\big|_{i,j,k+1}^{n+1} - E_y\big|_{i,j,k}^{n+1}}{\Delta z} - \frac{E_z\big|_{i,j+1,k}^{n+1} - E_z\big|_{i,j,k}^{n+1}}{\Delta y} \right)$$

$$H_y\Big|_{i,j,k}^{n+1} = H_y\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_z\big|_{i+1,j,k}^{n+1} - E_z\big|_{i,j,k}^{n+1}}{\Delta x} - \frac{E_x\big|_{i,j,k+1}^{n+1} - E_x\big|_{i,j,k}^{n+1}}{\Delta z} \right)$$

$$H_z\Big|_{i,j,k}^{n+1} = H_z\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_x\big|_{i,j+1,k}^{n+1} - E_x\big|_{i,j,k}^{n+1}}{\Delta y} - \frac{E_y\big|_{i+1,j,k}^{n+1} - E_y\big|_{i,j,k}^{n+1}}{\Delta x} \right)$$

3D FDTD: Array Sizes and Boundary Conditions

- Permittivity grid and output grid:

$$\text{size}(\varepsilon) = [N_x, N_y, N_z]$$

- Fields:
  - Tangential E-fields and normal H-fields are stored at integer indices $1:N$ $\rightarrow N$ grid points
  - Normal E-fields and tangential H-field are stored at fractional indices $1.5:N-0.5 \rightarrow N-1$ grid points
- Array sizes:
  - $E_x: (N_x - 1, N_y, N_z)$; $H_x: (N_x, N_y - 1, N_z - 1)$;
  - $E_y: (N_x, N_y - 1, N_z)$; $H_y: (N_x - 1, N_y, N_z - 1)$;
  - $E_z: (N_x, N_y, N_z - 1)$; $H_z: (N_x - 1, N_y - 1, N_z)$;
- PEC boundary conditions: At the boundaries the tangential E-fields and the normal H-fields are set to zero and are not updated

3D FDTD: Array Sizes and Boundary Conditions

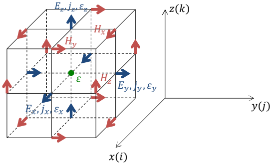
- PEC boundary conditions: At the boundaries the tangential E-fields and the normal H-fields are set to zero and are not updated

$$E_x(:,1,:) = 0 \qquad E_x(:,N_y,:) = 0$$
$$E_x(:,:,1) = 0 \qquad E_x(:,:,N_z) = 0$$
$$H_x(1,:,:) = 0 \qquad H_x(N_x,:,:) = 0$$

$$E_y(1,:,:) = 0 \qquad E_y(N_x,:,:) = 0$$
$$E_y(:,:,1) = 0 \qquad E_y(:,:,N_z) = 0$$
$$H_y(:,1,:) = 0 \qquad H_y(:,N_y,:) = 0$$

$$E_z(1,:,:) = 0 \qquad E_z(N_x,:,:) = 0$$
$$E_z(:,1,:) = 0 \qquad E_z(:,N_y,:) = 0$$
$$H_z(:,:,1) = 0 \qquad H_z(:,:,N_z) = 0$$

3D FDTD: Time Stepping
Update of the Electric Field



Separable source:

$$j_x\big|_{i,j,k}^{n} = A\big(\Delta t(n+0.5)\big)e^{-i\omega\Delta t(n+0.5)} j_x(t=0)\big|_{i,j,k}$$

$$j_y\big|_{i,j,k}^{n} = A\big(\Delta t(n+0.5)\big)e^{-i\omega\Delta t(n+0.5)} j_y(t=0)\big|_{i,j,k}$$

$$j_z\big|_{i,j,k}^{n} = A\big(\Delta t(n+0.5)\big)e^{-i\omega\Delta t(n+0.5)} j_z(t=0)\big|_{i,j,k}$$

$$E_x\big|_{i,j,k}^{n+1} = E_x\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0\varepsilon_x\big|_{i,j,k}}\left(\frac{H_z\big|_{i,j,k}^{n} - H_z\big|_{i,j-1,k}^{n}}{\Delta y} - \frac{H_y\big|_{i,j,k}^{n} - H_y\big|_{i,j,k-1}^{n}}{\Delta z} - j_x\big|_{i,j,k}^{n}\right)$$

$i = 1:N_x - 1$, $j = 2:N_y - 1$, $k = 2:N_z - 1$

$$E_y\big|_{i,j,k}^{n+1} = E_y\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0\varepsilon_y\big|_{i,j,k}}\left(\frac{H_x\big|_{i,j,k}^{n} - H_x\big|_{i,j,k-1}^{n}}{\Delta z} - \frac{H_z\big|_{i,j,k}^{n} - H_z\big|_{i-1,j,k}^{n}}{\Delta x} - j_y\big|_{i,j,k}^{n}\right)$$

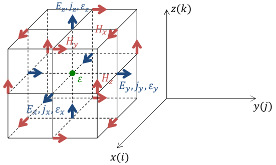
$i = 2:N_x - 1$, $j = 1:N_y - 1$, $k = 2:N_z - 1$

$$E_z\big|_{i,j,k}^{n+1} = E_z\big|_{i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0\varepsilon_z\big|_{i,j,k}}\left(\frac{H_y\big|_{i,j,k}^{n} - H_y\big|_{i-1,j,k}^{n}}{\Delta x} - \frac{H_x\big|_{i,j,k}^{n} - H_x\big|_{i,j-1,k}^{n}}{\Delta y} - j_z\big|_{i,j,k}^{n}\right)$$

$i = 2:N_x - 1$, $j = 2:N_y - 1$, $k = 1:N_z - 1$

Tangential E-fields at boundary are not updated!

3D FDTD: Time Stepping
Update of the Magnetic Field



$$H_x\Big|_{i,j,k}^{n+1} = H_x\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_y\Big|_{i,j,k+1}^{n+1} - E_y\Big|_{i,j,k}^{n+1}}{\Delta z} - \frac{E_z\Big|_{i,j+1,k}^{n+1} - E_z\Big|_{i,j,k}^{n+1}}{\Delta y} \right)$$

$i = 2 : N_x - 1$
$j = 1 : N_y - 1$
$k = 1 : N_z - 1$

$$H_y\Big|_{i,j,k}^{n+1} = H_y\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_z\Big|_{i+1,j,k}^{n+1} - E_z\Big|_{i,j,k}^{n+1}}{\Delta x} - \frac{E_x\Big|_{i,j,k+1}^{n+1} - E_x\Big|_{i,j,k}^{n+1}}{\Delta z} \right)$$

$i = 1 : N_x - 1$
$j = 2 : N_y - 1$
$k = 1 : N_z - 1$

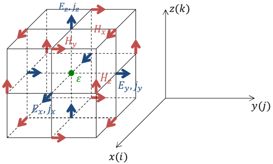
$$H_z\Big|_{i,j,k}^{n+1} = H_z\Big|_{i,j,k}^{n} + \frac{\Delta t}{\mu_0}\left(\frac{E_x\Big|_{i,j+1,k}^{n+1} - E_x\Big|_{i,j,k}^{n+1}}{\Delta y} - \frac{E_y\Big|_{i+1,j,k}^{n+1} - E_y\Big|_{i,j,k}^{n+1}}{\Delta x} \right)$$

$i = 1 : N_x - 1$
$j = 1 : N_y - 1$
$k = 2 : N_z - 1$

Normal H-fields at boundary are not updated!

3D FDTD: Interpolation of Output

- For postprocessing purposes it is desirable to have all fields on a common grid in space and time → fields must be interpolated (e.g. to the integer grid where ε is given)



Field	Interpolated Axes	Field	Interpolated Axes
E_x	x	H_x	y, z, t
E_y	y	H_y	x, z, t
E_z	z	H_z	x, y, t

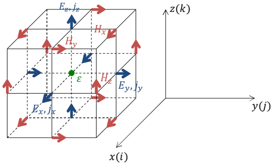
3D FDTD: Interpolation of Output

- For postprocessing purposes it is desirable to have all fields on a common grid in space and time → fields must be interpolated (e.g. to ε-grid)

$$E_x^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{2}\left(E_x\Big|_{i-1,j,k}^{n+1} + E_x\Big|_{i,j,k}^{n+1} \right)$$

$$E_y^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{2}\left(E_y\Big|_{i,j-1,k}^{n+1} + E_y\Big|_{i,j,k}^{n+1} \right)$$

$$E_z^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{2}\left(E_z\Big|_{i,j,k-1}^{n+1} + E_z\Big|_{i,j,k}^{n+1} \right)$$



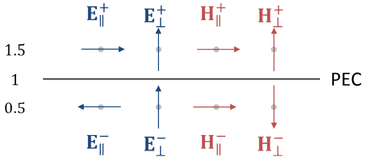
$$H_x^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{8}\left(H_x\Big|_{i,j-1,k-1}^{n} + H_x\Big|_{i,j-1,k}^{n} + H_x\Big|_{i,j,k-1}^{n} + H_x\Big|_{i,j,k}^{n} + H_x\Big|_{i,j-1,k-1}^{n+1} + H_x\Big|_{i,j-1,k}^{n+1} + H_x\Big|_{i,j,k-1}^{n+1} + H_x\Big|_{i,j,k}^{n+1} \right)$$

$$H_y^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{8}\left(H_y\Big|_{i-1,j,k-1}^{n} + H_y\Big|_{i-1,j,k}^{n} + H_y\Big|_{i,j,k-1}^{n} + H_y\Big|_{i,j,k}^{n} + H_y\Big|_{i-1,j,k-1}^{n+1} + H_y\Big|_{i-1,j,k}^{n+1} + H_y\Big|_{i,j,k-1}^{n+1} + H_y\Big|_{i,j,k}^{n+1} \right)$$

$$H_z^{\text{out}}\Big|_{i,j,k}^{n+1} = \frac{1}{8}\left(H_z\Big|_{i-1,j-1,k}^{n} + H_z\Big|_{i-1,j,k}^{n} + H_z\Big|_{i,j-1,k}^{n} + H_z\Big|_{i,j,k}^{n} + H_z\Big|_{i-1,j-1,k}^{n+1} + H_z\Big|_{i-1,j,k}^{n+1} + H_z\Big|_{i,j-1,k}^{n+1} + H_z\Big|_{i,j,k}^{n+1} \right)$$

3D FDTD: Interpolation of Output

- What about missing values at the boundaries? E.g.:
  - Interpolation of $E_x^{\mathrm{out}}(1,:,:)$ requires $E_x^{\mathrm{out}}(0,:,:)$
  - Interpolation of $H_x^{\mathrm{out}}(:,1,:)$ requires $H_x^{\mathrm{out}}(:,0,:)$
  - Interpolation of $H_x^{\mathrm{out}}(:,:,1)$ requires $H_x^{\mathrm{out}}(:,:,0)$
- At the PEC boundary the following mirror symmetries hold:
  - $\mathbf{E}_{\parallel}^{-} = -\mathbf{E}_{\parallel}^{+}, \mathbf{E}_{\perp}^{-} = +\mathbf{E}_{\perp}^{-}$
  - $\mathbf{H}_{\parallel}^{-} = +\mathbf{H}_{\parallel}^{+}, \mathbf{H}_{\perp}^{-} = -\mathbf{H}_{\perp}^{-}$
- Missing values behind the boundary can be obtained by duplicating the values in front of the boundary



Tasks

1. Implement the FDTD method in 1D and 3D versions (functions **fdtd_1d** and **fdtd_3d)**
2. Simulate the test problems:

   Propagation of a subcycle pulse through a homogeneous and inhomogeneous medium

3. Test the convergence and accuracy of obtained results vs. parameters **dx** and **dt**

Task I: Implementation of the 1D FDTD method

Physical problem:

- Simulate the propagation of an ultrashort pulse in a dispersion-free dielectric medium $\varepsilon(x) = 1$
- See what happens when the pulse hits the interface between two different dielectric media with permittivities $\varepsilon_2 = 1$ and $\varepsilon_2 = 4$, the interface should be located at a distance of $4.5\ \mu\mathrm{m}$ in positive direction from the center of the computational domain

Excitation:

- Pulsed source with frequency $f = 500\ \mathrm{THz}$ (red light)
  - delta-shaped spatial profile $j_z(t = 0, x) = j_0\delta(x - x_0)$ with $j_0 = 1\ \mathrm{A/m}^2$ located at the center of the computational domain at $x_0 = 0$
  - Gaussian temporal envelope $A(t) = \exp(-(t - t_0)^2/\tau^2)$ with $\tau = 1$ fs and $t_0 = 3\tau$

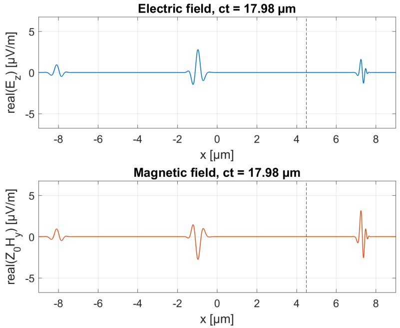
Simulation grid:

- Spatial window size of $\mathrm{W} = 18\ \mu\mathrm{m}$ with discretization $\Delta x = 15\ \mathrm{nm}$ and metallic walls ($E_z = 0$ at the boundaries)
- Simulation time span $T = 60$ fs with discretization $\Delta t = \Delta x/(2c)$

Output:

- $E_z(x, t)$ and $H_y(x, t)$ at every time step interpolated to the integer grid both in space and time

Task I: Implementation of the 1D FDTD method



Please include relevant plots of the fields (e.g. snapshots at certain time steps, time traces) in your report but do not include or submit video files!

Task I convergence and accuracy

Make Δx larger until you see numerical dispersion: due to bad discretization the pulse will broaden and deform as it propagates through the domain. How does the $\Delta x \approx \lambda/20/n_{\text{max}}$ rule of thumb fare? Does adjusting the Courant factor $c\Delta t/\Delta x$ help?
Also study what happens if the Courant factor $c\Delta t/\Delta x$ exceeds 1.

Task I: Implementation of the 1D FDTD method

```
def fdtd_1d(eps_rel, dx, time_span, source_frequency, source_position, source_pulse_length):
    '''Computes the temporal evolution of a pulsed excitation using the 1D FDTD method. The temporal center of
    the pulse is placed at a simulation time of 3*source_pulse_length. The origin x=0 is in the center of the
    computational domain. All quantities have to be specified in SI units.

    Arguments
    ---------
        eps_rel : 1d-array
            Rel. permittivity distribution within the computational domain.
        dx : float
            Spacing of the simulation grid (please ensure dx <= lambda/20).
        time_span : float
            Time span of simulation.
        source_frequency : float
            Frequency of current source.
        source_position : float
            Spatial position of current source.
        source_pulse_length :
            Temporal width of Gaussian envelope of the source.

    Returns
    -------
        Ez : 2d-array
            Z-component of E(x,t) (each row corresponds to one time step)
        Hy : 2d-array
            Y-component of H(x,t) (each row corresponds to one time step)
        x  : 1d-array
            Spatial coordinates of the field output
        t  : 1d-array
            Time of the field output
    '''
    pass
```

Task I: Implementation of the 1D FDTD method

- You can use the provided animation function to watch a movie of the fields

```
class Fdtd1DAnimation(animation.TimedAnimation):

    '''Animation of the 1D FDTD fields.

    Based on https://matplotlib.org/examples/animation/subplots.html

    Arguments
    ---------
    x : 1d-array
        Spatial coordinates
    t : 1d-array
        Time
    x_interface : float
        Position of the interface (default: None)
    step : float
        Time step between frames (default: 2e-15/25)
    fps : int
        Frames per second (default: 25)
    Ez: 2d-array
        Ez field to animate (each row corresponds to one time step)
    Hy: 2d-array
        Hy field to animate (each row corresponds to one time step)
    '''
```

Task II: Implementation of the 3D FDTD method

Physical problem:

- Investigate the radiation characteristics of a pulsed line current with a Gaussian spatial envelope

$$\mathbf{j}(x, y, z, t) = j_0 \exp(-2\pi i f t) \exp\left(-\frac{(t-t_0)^2}{\tau^2}\right) \exp\left(-\frac{x^2+y^2}{w^2}\right) \mathbf{e}_z$$

Simulation grid:

- Spatial domain size of $199x201x5$ grid points with a step size of $\Delta x = \Delta y = \Delta z = 30\ \mathrm{nm}$
- PEC boundary conditions
- Simulation time span $T = 10$ fs with discretization $\Delta t = \Delta x/(2c)$
- Specify all input quantities ($\varepsilon(\mathbf{r})$, $j_x(\mathbf{r})$, $j_y(\mathbf{r})$ and $j_z(\mathbf{r})$) on the same centered integer grid and interpolate the quantities to the required shifted grids within the implementation

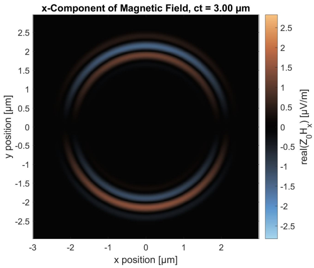
Excitation:

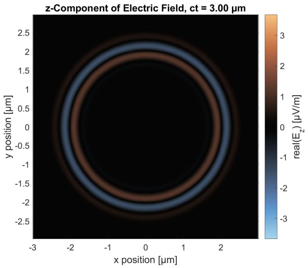
- Pulsed current source with amplitude $j_0 = 1\ \mathrm{A/m^2}$, frequency f=500 THz (red light), temporal width $\tau = 1$ fs and offset $t_0 = 3\tau$ and spatial width $w = 2\Delta x$

Output:

- H_x and E_z in the xy-plane centered in the middle along the z-direction at every 4th time step interpolated to the integer grid in space and time

Task II: Implementation of the 3D FDTD method





Please include relevant plots of the fields (e.g. snapshots at $t = T$) in your report but do not include or submit video files!

Task II: Implementation of the 3D FDTD method

```
def fdtd_3d(eps_rel, dr, time_span, freq, tau, jx, jy, jz, field_component, z_ind, output_step):
    '''Computes the temporal evolution of a pulsed spatially extended current source using the 3D FDTD method.
     Returns z-slices of the selected field at the given z-position every output_step time steps. The pulse is
     centered at a simulation time of 3*tau. All quantities have to be specified in SI units.

    Arguments
    ---------
        eps_rel: 3d-array
            Rel. permittivity distribution within the computational domain.
        dr: float
            Grid spacing (please ensure dr<=lambda/20).
        time_span: float
            Time span of simulation.
        freq: float
            Center frequency of the current source.
        tau: float
            Temporal width of Gaussian envelope of the source.
        jx, jy, jz: 3d-array
            Spatial density profile of the current source.
        field_component : str
            Field component which is stored (one of ‘ex’,’ey’,’ez’, ’hx’,’hy’,’hz’).
        z_index: int
            Z-position of the field output.
        output_step: int
            Number of time steps between field outputs.

    Returns
    -------
        F: 3d-array
            Z-slices of the selected field component at the z-position specified by z_ind stored every output_step
            time steps (time varies along the first axis).
        t: 1d-array
            Time of the field output.
    '''
    pass
```

Task II: Implementation of the 3D FDTD method

- You can use the provided animation function to watch a movie of the fields

```
class Fdtd3DAnimation(animation.TimedAnimation):

    '''Animation of a 3D FDTD field.

    Based on https://matplotlib.org/examples/animation/subplots.html

    Arguments
    ---------
    x, y : 1d-array
        Coordinate axes.
    t : 1d-array
        Time
    field: 3d-array
        Slices of the field to animate (the time axis is assumed to be the first axis of the array)
    titlestr : str
        Plot title.
    cb_label : str
        Colrbar label.
    rel_color_range: float
        Range of the colormap relative to the full scale of the field magnitude.
    fps : int
        Frames per second (default: 25)
    '''
```