Computational Photonics
Seminar 04, May 24, 2019

Finite-Difference Time-Domain Method (FDTD)

• Lecture review
• Implementation of 1D version of the FDTD method
• Implementation of 3D FDTD
Lecture review:
Yee grid in 3D
Yee grid in 3D

curl $\mathbf{H}(\mathbf{r}, t) = \varepsilon_0 \varepsilon(\mathbf{r}) \frac{\partial \mathbf{E}(\mathbf{r}, t)}{\partial t} + \mathbf{j}_{makr}$

\[
\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon_0 \varepsilon(\mathbf{r})} \left[ \frac{\partial H_y}{\partial y} - \frac{\partial H_x}{\partial z} - j_x \right]
\]

\[
E_z|_{i,j,k+\frac{1}{2}}^{n+1} = E_z|_{i,j,k+\frac{1}{2}}^n + \frac{\Delta t}{\varepsilon_0 \varepsilon_{i,j,k+\frac{1}{2}}} \left( \frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+0.5} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+0.5}}{\Delta x} - \frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+0.5} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+0.5}}{\Delta y} - j_z|_{i,j,k+\frac{1}{2}}^{n+0.5} \right)
\]
Grid size is determined by the permittivity distribution:

\[ \text{size}(\varepsilon) = [N_x, N_y, N_z] \]
3D FDTD: Electric Field Components

Permittivity must be interpolated:

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

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

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

\[
E_{x}^{n+1}_{i+0.5,j,k} = E_{x}^{n}_{i+0.5,j,k} + \frac{\Delta t}{\varepsilon_{0} \varepsilon_{i+0.5,j,k}} \left( \frac{H_{z}^{n+0.5}_{i+0.5,j+0.5,k} - H_{z}^{n+0.5}_{i+0.5,j-0.5,k}}{\Delta y} \right) - \frac{H_{y}^{n+0.5}_{i+0.5,j,k+0.5} - H_{y}^{n+0.5}_{i+0.5,j,k-0.5}}{\Delta z} - j_{x}^{n+0.5}_{i+0.5,j,k}
\]

\[
E_{y}^{n+1}_{i,j+0.5,k} = E_{y}^{n}_{i,j+0.5,k} + \frac{\Delta t}{\varepsilon_{0} \varepsilon_{i,j+0.5,k}} \left( \frac{H_{x}^{n+0.5}_{i,j+0.5,k+0.5} - H_{x}^{n+0.5}_{i,j+0.5,k-0.5}}{\Delta z} \right) - \frac{H_{z}^{n+0.5}_{i+0.5,j+0.5,k} - H_{z}^{n+0.5}_{i-0.5,j+0.5,k}}{\Delta x} - j_{y}^{n+0.5}_{i,j+0.5,k}
\]

\[
E_{z}^{n+1}_{i,j,k+0.5} = E_{z}^{n}_{i,j,k+0.5} + \frac{\Delta t}{\varepsilon_{0} \varepsilon_{i,j,k+0.5}} \left( \frac{H_{y}^{n+0.5}_{i+0.5,j,k+0.5} - H_{y}^{n+0.5}_{i-0.5,j,k+0.5}}{\Delta x} \right) - \frac{H_{x}^{n+0.5}_{i,j+0.5,k+0.5} - H_{x}^{n+0.5}_{i,j-0.5,k+0.5}}{\Delta y} - j_{z}^{n+0.5}_{i,j,k+0.5}
\]
3D FDTD: Magnetic Field Components

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

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

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

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

Changing of index notation to integer indices

$t(n)$

\begin{align*}
t(n) & \approx 2.5 \\
    2.5 & \\
    2 & \\
    1.5 & \\
    1 &
\end{align*}

$x(i)$

\begin{align*}
x(i) & \approx 1 \\
    1 & \\
    1.5 & \\
    2 & \\
    2.5 & \\
    3 & \\
    3.5 &
\end{align*}

Changing of index notation to integer indices:

\begin{align*}
E_z^{n+\frac{1}{2}} & \approx E_z^n + \frac{1}{\varepsilon_0\varepsilon_i} \frac{\Delta t}{\Delta x} \left[ H_y^{n+\frac{1}{2}} - H_y^{n-\frac{1}{2}} \right] - \frac{\Delta t}{\varepsilon_0\varepsilon_i} j_z^{n+\frac{1}{2}} \\
H_y^{n+\frac{3}{2}} & \approx H_y^{n+\frac{1}{2}} + \frac{1}{\mu_0} \frac{\Delta t}{\Delta x} \left[ E_z^{n+1} - E_z^n \right]
\end{align*}

start at $n=1$:

\begin{align*}
E_z^{n+1} & \approx E_z^n + \frac{1}{\varepsilon_0\varepsilon_i} \frac{\Delta t}{\Delta x} \left[ H_y^n - H_y^{n-1} \right] - \frac{\Delta t}{\varepsilon_0\varepsilon_i} j_z^n \\
H_y^{n+1} & \approx H_y^n + \frac{1}{\mu_0} \frac{\Delta t}{\Delta x} \left[ E_z^{n+1} - E_z^n \right]
\end{align*}

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

\begin{align*}
E_z^{n+1} & \approx E_z^n + \frac{1}{\varepsilon_0\varepsilon_i} \frac{\Delta t}{\Delta x} \left[ H_y^n - H_y^{n-1} \right] - \frac{\Delta t}{\varepsilon_0\varepsilon_i} j_z^n \\
H_y^{n+1} & \approx H_y^n + \frac{1}{\mu_0} \frac{\Delta t}{\Delta x} \left[ E_z^{n+1} - E_z^n \right]
\end{align*}
1D FDTD: Source

- Excitation of the fields by a separable current source:

\[ j_z^n_i = A(\Delta t(n + 1/2))e^{-2\pi if\Delta t(n+1/2)}j_z(t = 0) \bigg|_i \]

- Spatial distribution: \( j_z(t = 0) \bigg|_i \)
- Carrier: \( e^{-2\pi if\Delta t(n+1/2)} \)
- Envelope: \( A(\Delta t(n + 1/2)) \)
3D FDTD: Electric Field Components
Change Index Notation to Integer Indices

Renaming of fractional indices:

\[
i + 0.5 \rightarrow i
\]
\[
j + 0.5 \rightarrow j
\]
\[
k + 0.5 \rightarrow k
\]

Renaming of interpolated permittivity:

\[
\epsilon_{i+0.5,j,k} \rightarrow \epsilon_x |_{i,j,k}
\]
\[
\epsilon_{i,j+0.5,k} \rightarrow \epsilon_y |_{i,j,k}
\]
\[
\epsilon_{i,j,k+0.5} \rightarrow \epsilon_z |_{i,j,k}
\]

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

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

\[
E_{z |i,j,k}^{n+1} = E_{z |i,j,k}^{n} + \frac{\Delta t}{\varepsilon_0 \varepsilon_z |_{i,j,k}} \left( \frac{H_y |_{i,j,k}^{n} - H_y |_{i-1,j,k}^{n}}{\Delta x} - \frac{H_x |_{i,j,k}^{n} - H_x |_{i,j-1,k}^{n}}{\Delta y} - j_z |_{i,j,k}^{n} \right)
\]
3D FDTD: Magnetic Field Components
Change Index Notation to Integer Indices

Renaming of fractional indices:

\[i + 0.5 \rightarrow i\]
\[j + 0.5 \rightarrow j\]
\[k + 0.5 \rightarrow k\]

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

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

\[
H_z^{n+1}_{i, j, k} = H_z^n_{i, j, k} + \frac{\Delta t}{\mu_0} \left( \frac{E_x^{n+1}_{i+1, j, k} - E_x^{n+1}_{i, j, k}}{\Delta y} - \frac{E_y^{n+1}_{i+1, j, k} - E_y^{n+1}_{i, j, k}}{\Delta x} \right)
\]
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 \textit{integer indices} \( 1: N \rightarrow N \) grid points
  – Normal E-fields and tangential H-field are stored at \textit{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 (Perfect electric conductor) boundary conditions: At the boundaries the tangential E-fields and the normal H-fields are set to zero and are not updated
1D FDTD: Layout of the Matlab Arrays
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

\[
\begin{align*}
E_x(:, 1, :) &= 0 \\
E_x(:, :, 1) &= 0 \\
H_x(1, :, :) &= 0 \\
E_y(1, :, :) &= 0 \\
E_y(:, :, 1) &= 0 \\
H_y(:, 1, :) &= 0 \\
E_z(1, :, :) &= 0 \\
E_z(:, 1, :) &= 0 \\
H_z(:, :, 1) &= 0
\end{align*}
\]
3D FDTD: Time Stepping
Update of the Electric Field

Separable source:

\[ j_x^{n+1}_{i,j,k} = A(\Delta t(n + 0.5))e^{-i\omega\Delta t(n+0.5)}j_x(t = 0)|_{i,j,k} \]
\[ j_y^{n+1}_{i,j,k} = A(\Delta t(n + 0.5))e^{-i\omega\Delta t(n+0.5)}j_y(t = 0)|_{i,j,k} \]
\[ j_z^{n+1}_{i,j,k} = A(\Delta t(n + 0.5))e^{-i\omega\Delta t(n+0.5)}j_z(t = 0)|_{i,j,k} \]

E_x^{n+1}_{i,j,k} = E_x^n_{i,j,k} + \Delta t e_{0}^{x} \left( \frac{H_y^n_{i,j,k-1} - H_y^n_{i,j,k}}{\Delta y} - j_y^n_{i,j,k} \right) \]

E_y^{n+1}_{i,j,k} = E_y^n_{i,j,k} + \Delta t e_{0}^{y} \left( \frac{H_z^n_{i,j,k-1} - H_z^n_{i,j,k}}{\Delta z} - j_z^n_{i,j,k} \right) \]

E_z^{n+1}_{i,j,k} = E_z^n_{i,j,k} + \Delta t e_{0}^{z} \left( \frac{H_x^n_{i,j,k-1} - H_x^n_{i,j,k}}{\Delta x} - j_x^n_{i,j,k} \right) \]

\[ i = 1: N_x - 1 \]
\[ j = 2: N_y - 1 \]
\[ k = 2: N_z - 1 \]

Tangential E-fields at boundary are not updated!
3D FDTD: Time Stepping

Update of the Magnetic Field

\[
H_x^{i+1} = H_x^n + \frac{\Delta t}{\mu_0} \left( E_y^{i+1} - E_y^n \right) \frac{\Delta z}{\Delta y} - E_z^{i+1} - E_z^n \frac{\Delta z}{\Delta y}
\]

\[
H_y^{i+1} = H_y^n + \frac{\Delta t}{\mu_0} \left( E_z^{i+1} - E_z^n \right) \frac{\Delta x}{\Delta z} - E_x^{i+1} - E_x^n \frac{\Delta x}{\Delta z}
\]

\[
H_z^{i+1} = H_z^n + \frac{\Delta t}{\mu_0} \left( E_x^{i+1} - E_x^n \right) \frac{\Delta y}{\Delta x} - E_y^{i+1} - E_y^n \frac{\Delta y}{\Delta x}
\]

Normal H-fields at boundary are not updated!
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)

<table>
<thead>
<tr>
<th>Field</th>
<th>Interpolated Axes</th>
<th>Field</th>
<th>Interpolated Axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_x$</td>
<td>$x$</td>
<td>$H_x$</td>
<td>$y, z, t$</td>
</tr>
<tr>
<td>$E_y$</td>
<td>$y$</td>
<td>$H_y$</td>
<td>$x, z, t$</td>
</tr>
<tr>
<td>$E_z$</td>
<td>$z$</td>
<td>$H_z$</td>
<td>$x, y, t$</td>
</tr>
</tbody>
</table>
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)

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

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

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

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

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

\[
H_z^{\text{out}}|_{i,j,k}^{n+1} = \frac{1}{8} \left( H_z|_{i-1,j-1,k}^{n} + H_z|_{i-1,j,k}^{n} + H_z|_{i,j-1,k}^{n} + H_z|_{i,j,k}^{n} + H_z|_{i-1,j-1,k}^{n+1} + H_z|_{i-1,j,k}^{n+1} + H_z|_{i,j-1,k}^{n+1} + H_z|_{i,j,k}^{n+1} \right)
\]
3D FDTD: Interpolation of Output

• What about missing values at the boundaries? E.g.:
  – Interpolation of $E_x^{\text{out}} (1, :, :)$ requires $E_x^{\text{out}} (0, :, :)$
  – Interpolation of $H_x^{\text{out}} (:, 1, :)$ requires $H_x^{\text{out}} (:, 0, :)$
  – Interpolation of $H_x^{\text{out}} (:, :, 1)$ requires $H_x^{\text{out}} (:, :, 0)$

• At the PEC boundary the following mirror symmetries hold:
  ➢ $E_\parallel^- = -E_\parallel^+, E_\perp^- = +E_\perp$
  ➢ $H_\parallel^- = +H_\parallel^+, H_\perp^- = -H_\perp$

• Missing values behind the boundary can be obtained by duplicating the values in front of the boundary
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 \( 4.5 \) \( \mu \)m in positive direction from the center of the computational domain

Excitation:
• Pulsed source with frequency \( f = 500 \) THz (red light)
  • delta-shaped spatial profile \( j_z(t = 0, x) = j_0 \delta(x - x_0) \) with \( j_0 = 1 \) 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 \( W = 18 \) \( \mu \)m with discretization \( \Delta x = 15 \) 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

Useful Matlab functions:
• round, cat, meshgrid, drawnow, subplot, imagesc
• Useful constants: \( c = 2.99792458 \cdot 10^8 \) m/s, \( \mu_0 = 4\pi \cdot 10^{-7} \) Vs/Am, \( \varepsilon_0 = 1/(c^2\mu_0) \) As/Vm
Task I: Implementation of the 1D FDTD method

Please include relevant plots of the fields (e.g. snapshots a certain time steps, time traces) in your report but do not include or submit video files!
Task I: Implementation of the 1D FDTD method

```matlab
function [Ez,Hy,x,t] = fdtd_1d(eps_rel, dx, time_span, ... 
    source_frequency, source_position,... 
    source_pulse_length)
% function [Ez,Hy,x,t] = fdtd_1d(eps_rel, grid_size, 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 : rel. permittivity distribution within the
%            computational domain (vector)
%  dx      : spacing of the simulation grid
%            (scalar, please ensure dx <= lambda/20)
%  time_span: time span of simulation (scalar)
%  source_frequency : frequency of current source (scalar)
%  source_position : spatial position of current source (scalar)
%  source_pulse_length : temporal width of Gaussian envelope of
%                       the source (scalar)
% Returns:
%  Ez : z component of E(x,t) (matrix, each column corresponds to one
%       time step)
%  Hy : y component of H(x,t) (matrix, each column corresponds to one
%       time step)
%  x  : spatial coordinates of the field output (vector)
%  t  : time of the field output (vector)
```
Task I: Implementation of the 1D FDTD method

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

function fig = fdtd_1d_animation(x, t, Ez, Hy, x_interface, step, ...
    fps, filename)

fig = fdtd_1d_animation(x, t, Ez, Hy, x_interface, step, fps, filename)
% Creates an animation of the 1D FDTD fields.
%
% Arguments:
%   x    : Spatial coordinates (vector)
%   t    : Time (vector)
%   Ez   : Ez field to animate (matrix, each column corresponds
%          to one time step)
%   Hy   : Hy field to animate (matrix, each column corresponds
%          to one time step)
%   x_interface : Position of the interface (scalar)
%   step  : Time step between frames (scalar)
%   fps   : Frames per second (integer, default: 25)
%   filename : Filename of the video. A png image of the
%              last frame is also save to the filename with the
%              extension replaced by '.png'. If empty, no files
%              are saved to disk. (string, default: '')
% Returns:
%   fig : Figure of the animation.
Task II: Implementation of the 3D FDTD method

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

\[ j(x, y, z, t) = j_0 \exp(-2\pi if t)\exp\left(-\frac{(t - t_0)^2}{\tau^2}\right)\exp\left(-\frac{x^2 + y^2}{w^2}\right)e_z \]

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

Excitation:
- Pulsed current source with amplitude \( j_0 = 1 \) 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

Useful Matlab functions:
- mod, cat
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

```matlab
function [F, t] = fdtd_3d(eps_rel,dr,time_span,freq,tau,jx,jy,jz,...
    field_component,z_ind,output_step)
% function [F, t] = 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 : rel. permittivity distribution within the
%             computational domain (3D array)
%   dr       : grid spacing
%               (scalar, please ensure dr<=lambda/20)
%   time_span: time span of simulation (scalar)
%   freq     : center frequency of the current source (scalar)
%   tau      : temporal width of Gaussian envelope of
%               the source (scalar)
%   jx, jy, jz : spatial density profile of the current source
%               (3D arrays)
%   field_component : field component which is stored
%                      (one of ‘ex’,’ey’,’ez’,’hx’,’hy’,’hz’)
%   z_index   : z-position of the field output (integer)
%   output_step : number of time steps between field outputs
%                 (integer)
% Returns:
%   F : z-slices of the selected field component at the
%        z-position specified by z_ind stored every output_step
%        time steps (3D array, time varies along the last dimension)
%   t : time of the field output (vector)
```
Task II: Implementation of the 3D FDTD method

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

```matlab
function fig = fdtd_3d_animation(x, y, t, F, titlestr, cb_label, ...
    rel_color_range, fps, filename)
    fig = fdtd_3d_animation(x, y, t, F, titlestr, cb_label, ...
    rel_color_range, fps, filename)
    % Creates an animation of a 3D FDTD field.
    %
    % Arguments:
    % x, y         : Coordinate axes (vectors)
    % t           : Time (vector)
    % F           : Slices of the field to animate (3d-array,
    %               the time axis is assumed to be the last axis
    %               of the array)
    % titlestr     : Plot title (string)
    % cb_label     : Colorbar label (string)
    % rel_color_range : Range of the colormap relative to the full scale
    %               of the field magnitude (scalar)
    % fps          : Frames per second (integer, default: 25)
    % filename     : Filename of the video. A png image of the
    %               last frame is also save to the filename with the
    %               extension replaced py '.png'. If empty, no files
    %               are saved to disk. (string, default: '')
    % Returns:
    % fig : Figure of the animation.
```
Voluntary Homework (due June 6, 2019)

• Solve tasks I & II.
• For each task we require that each student implements a program that solves the problem and documents the code and its result (e.g. with an iPython Notebook)
• Submission via email to teaching-nanooptics@uni-jena.de by Thursday June 6, 2019. The subject line of the email should have the following format:
  – [family name]; [given names]; [student id]: CPho19 - solution to the assignment of seminar [seminar no.]
• If sending more than one file, gather them in a single zip archive (no rar, tar, 7z, gz or any other compression format)