

# Finite-Difference Time-Domain Method – Solution to the Seminar Tasks

Matthias Zilk

The tasks of this exercise were to implement the finite-difference time-domain (FDTD) method in one and three dimensions with perfect electric conducting (PEC) boundary conditions. The implementation of the 1D-FDTD method should be tested by simulating the propagation of the electromagnetic fields in a homogeneous medium and at the interface between two dielectric media with different refractive indices. The excitation of the fields should be performed with a separable current source

$$j_z(x, t) = A(x) \cdot f(t) \quad (1)$$

with a delta-shaped spatial profile

$$A(x) = j_0 \delta(x - x_s). \quad (2)$$

The time dependence was specified as a harmonic carrier with a Gaussian envelope:

$$f(t) = \exp(-2i\pi f_0(t - t_0)) \exp\left(-\frac{(t - t_0)^2}{\tau^2}\right). \quad (3)$$

The 3D implementation should be tested by simulating the propagation of the electromagnetic fields in a homogeneous medium excited with a spatially extended current source

$$\mathbf{j}(x, y, z, t) = \mathbf{A}(x, y, z) \cdot f(t) \quad (4)$$

with the same time dependence as in the 1D case but with a Gaussian spatial profile

$$\mathbf{A}(x, y, z) = j_0 \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{w^2}\right) \mathbf{e}_z. \quad (5)$$

## 1 Task I: 1D-FDTD

The evolution of the electromagnetic field in the one dimensional case is governed by the equations

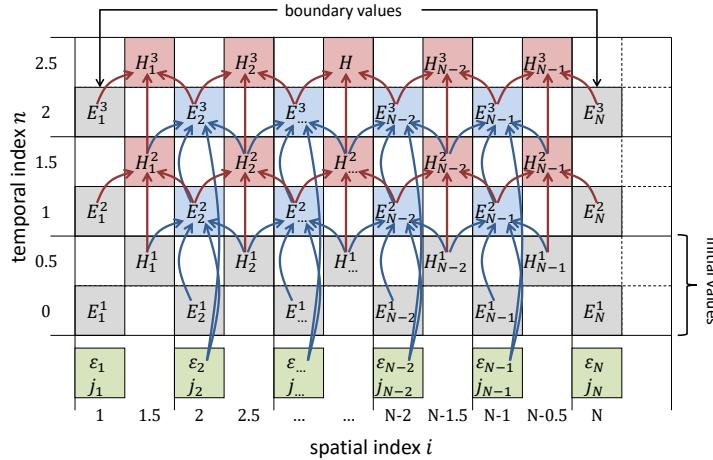
$$\frac{\partial H_y(x, t)}{\partial x} = \epsilon_0 \epsilon_r(x) \frac{\partial E_z(x, t)}{\partial t} + j_z(x, t), \quad (6)$$

$$\frac{\partial E_z(x, t)}{\partial x} = \mu_0 \frac{\partial H_y(x, t)}{\partial t}. \quad (7)$$

In the FDTD method space and time are discretized as  $x = i\Delta x + x_0$  and  $t = n\Delta t$ . This allows to approximate the space and time derivatives of equations (6) and (7) by finite differences and yields the following iteration equations (with subscript spatial index and superscript temporal index):

$$E_z|_i^{n+1} = E_z|_i^n + \frac{\Delta t}{\epsilon_0 \epsilon_r|_i} \left( \frac{H_y|_{i+0.5}^{n+0.5} - H_y|_{i-0.5}^{n+0.5}}{\Delta x} - j_z|_i^{n+0.5} \right), \quad (8)$$

$$H_y|_{i+0.5}^{n+1.5} = H_y|_{i+0.5}^{n+0.5} + \frac{\Delta t}{\mu_0} \frac{E_z|_{i+1}^{n+1} - E_z|_i^{n+1}}{\Delta x}. \quad (9)$$



**Figure 1:** Time stepping scheme of the one dimensional FDTD method. Superscript indices in the boxes correspond to the column of the Matlab matrix that stores the respective fields. Subscript indices correspond to the row of the respective Matlab matrix (or vector in case of the relative permittivity  $\epsilon$ ). Gray boxes indicate field values that are set to zero because of boundary or initial conditions.

To obtain second order approximations of the derivatives, centered differences are used. Hence the magnetic field is shifted by half a grid point with respect to the electric field both in space and time. The spatial staggering is called Yee grid, the temporal staggering is called leap-frog scheme because the fields “jump” over each other during the iteration.

In order to use equations (8) and (9) we need to specify boundary conditions at the edges of the computational domain of width  $W$  and initial conditions for the start of the iteration. The initial conditions are straight forward, we just assume that the electric field at  $t = 0$  and the magnetic field at  $t = 0.5\Delta t$  are zero.

The boundary conditions are more complicated and several choices are possible. We could set the tangential electric field at the boundaries to zero (PEC boundary conditions), we could set the tangential magnetic field to zero (PMC boundary conditions), we could wrap around the fields with some phase factor  $\exp(ik_x W)$  (Floquet-Bloch boundary conditions) or we could use a mixture (e.g. PEC conditions on the left side and PMC conditions on the right side or vice versa).

In this exercise we choose PEC boundary conditions on both sides. Assuming that the electric field is stored on  $N$  grid points with indices  $i = 1 \dots N$ , we set  $E_z|_1 = E_z|_N = 0$  and never update these values. For the magnetic field we do not need to store boundary values. We have to store the field only at fractional indices  $i = 1.5 \dots N - 0.5$  inside the computational domain giving in total  $N - 1$  grid points. The permittivity and the current density are stored on the same spatial grid as the electric field.

The spatial and temporal grids of the electric and magnetic field are illustrated in figure 1. Initial values and boundary values are grayed out. The fields that contribute to a specific field value are indicated by arrows. The indices in the boxes refer to the Matlab array indices that will be explained in the next subsection. As shown in the lecture and the seminar, the fractinal grid indices of the magnetic field have to be translated to integer array indices by renaming  $i + 0.5 \rightarrow i$  and  $n + 0.5 \rightarrow n$ . This yields the following iteration formulas for the time stepping of the fields:

$$E_z|_i^{n+1} = E_z|_i^n + \frac{\Delta r}{\epsilon_0 \epsilon_r|_i} \left[ \frac{H_y|_i^n - H_y|_{i-1}^n}{\Delta x} - j_z|_i^n \right] \quad \text{for } i = 2 \dots N - 1 \quad (10)$$

$$H_y|_i^{n+1} = E_y|_i^n + \frac{\Delta t}{\mu_0} \frac{E_z|_{i+1}^{n+1} - E_z|_i^{n+1}}{\Delta x} \quad \text{for } i = 1 \dots N - 1 \quad (11)$$

The current density follows from equations (1) to (3):

$$j_z|_i^n = j_0 \delta_{ii_s} \exp \left( -\frac{(\Delta t(n + 0.5) - 3\tau)^2}{\tau^2} - 2i\pi f \Delta t(n + 0.5) \right) \quad (12)$$

The temporal center of the pulse is set to  $t_0 = 3\tau$  to avoid a clipping of the envelope that would give rise to undesirable high frequency components. The index  $i_s$  is calculated from the location of the current source that is passed to the algorithm by the user.

At the end of the simulation we would like to output both the electric and the magnetic field on the same grid points in time and space. That means that the magnetic field has to be interpolated while the electric field can be taken as it is:

$$E_z^{\text{out}}|_i^n = E_z|_i^n \quad (13)$$

$$H_y^{\text{out}}|_i^n = \frac{1}{4} \left( H_y|_i^n + H_y|_{i-1}^n + H_y|_i^{n-1} + H_y|_{i-1}^{n-1} \right) \quad (14)$$

The interpolation of  $H_y$  requires values at the boundaries that are neither calculated nor stored in the field array, namely  $H_y|_0$  and  $H_y|_N$ . However, these values can be inferred from the mirror symmetries of the electro-magnetic fields in front of a PEC. A PEC acts as an electric mirror. This means that the tangential electric field components and the normal magnetic field are antisymmetric while the normal electric field and the tangential magnetic field components are symmetric with respect to the mirror plane. In the specific case this yields

$$H_y|_0 = H_y|_1 \quad \text{and} \quad H_y|_N = H_y|_{N-1}. \quad (15)$$

## 1.1 Implementation

The mathematical operations that are required for the implementation of FDTD method are quite simple. In a naive implementation equations (10) and (11) could be translated almost literally into Matlab code using `for`-loops. However, this would be quite inefficient as explicit loops are rather slow in Matlab. Therefore, an efficient implementation has to use vectorized expressions.

In the implementation of the 1D-FDTD we want to store the fields at all time steps in matrices where one dimension corresponds to the spatial coordinate and the other dimension corresponds to the temporal coordinate. We have two choices for these matrices: we can associate either the row indices or the column indices with the spatial coordinate. In each time step, all spatial grid points are accessed. Hence, it is very important for the efficiency of the CPU caching and prefetching mechanisms, that the spatial dimension of the field arrays is contiguous. Matlab uses column major layout, i.e. the columns of a matrix are stored contiguously in memory. Thus, we should associate the row indices (i.e. the first dimension of the matrix) with the spacial coordinate. If you are using Python instead of Matlab you have to be careful: Python uses row major layout by default. Hence, in a Python implementation the spatial coordinate should be associated with the column indices (i.e. the second dimension of the matrix) instead.

The 1D-FDTD method is implemented in the file `fdtd_1d.m`. The file is documented extensively and will not be shown in full here. In the following we will focus only on some key aspects of the implementation.

The size of the computational grid is derived from the size of the relative permittivity vector `eps_rel` that is passed as an argument to the function:

`fdtd_1d.m`

41    `Nx = numel(eps_rel);`

The origin of the coordinate system is supposed to be in the center of the computational domain. Hence, the  $x$ -coordinates are given by

fdtd\_1d.m

```
42 x = (0:Nx-1)*dx - ceil((Nx - 1)/2)*dx;
```

The number of iterations follows from the requested simulation duration `time_span` and the temporal discretization `Delta_t` (which is fixed by the spatial resolution and the Courant condition):

fdtd\_1d.m

```
43 Niter = round(time_span/Delta_t);
```

The fields are stored at all time steps. The arrays to store the fields are allocated with

fdtd\_1d.m

```
47 Ez = zeros( Nx, Niter + 1);
48 Hy = zeros(Nx - 1, Niter + 1);
```

Both arrays have an extra entry along the time axis (the column index of the matrices) to store the initial values. The array of the electric field has the same number of points along spatial axis (the row index of the matrix) as the permittivity. The spatial axis of the magnetic field has one entry less than the electric field because it is shifted by half a grid point and no boundary values need to be stored. We also have to make sure that `eps_rel` is a column vector:

fdtd\_1d.m

```
49 eps_rel = reshape(eps_rel, [], 1);
```

The relation of the array indices to the true spatial and temporal coordinates of the fields is illustrated in figure 1. The implementation assumes a delta-shaped source current that has a non-zero amplitude only on a single grid point. Hence, it is not necessary to store an array for the current distribution but it is sufficient store the index where the current is non-zero:

fdtd\_1d.m

```
58 if (source_ind < 2) || (source_ind > (Nx - 1))
```

The actual simulation fits into a short `for`-loop:

fdtd\_1d.m

```
62 for n = 1:Niter
63 % calculate E at time n + 1, the values at the spatial indices
64 % 1 and Nx are given by the PEC boundary conditions
65 % and do not have to be updated
66 Ez(2:Nx-1, n+1) = Ez(2:Nx-1, n) ...
67 % + e_factor/dx*((Hy(2:Nx-1, n) - Hy(1:Nx-2, n))/eps_rel(2:Nx-1));
68 % add source term to Ez
69 % source current has to be taken at n + 1/2
70 t_source = t(n) + 0.5*Delta_t - t0;
71 j_source = exp(-1i*source_angular_frequency*t_source) ... % carrier
72 % *exp(-(t_source/source_pulse_length)^2); % envelope
73 Ez(source_ind, n+1) = Ez(source_ind, n+1) ...
74 % - e_factor/eps_rel(source_ind)*j_source;
75 % calculate H at time n + 3/2
76 Hy(:, n+1) = Hy(:, n) + h_factor/dx*(Ez(2:Nx, n+1) - Ez(1:Nx-1, n+1));
77 end
```

First, in line 66 the electric field is updated with the values of the magnetic field from the *previous* iteration. Please note that only the inner indices  $2:Nx-1$  of the electric field are updated. The derivative of the magnetic field is calculated by subtracting the field values at the indices  $1:Nx-2$  from the field values at the indices  $2:Nx-1$  (as the spatial dimension of the magnetic field has a length  $Nx - 1$ , both index expressions yield vectors with  $Nx-2$  elements). The grid of `eps_re1` is identical to the grid of the electric field and is indexed in the same fashion.

To finish the update of the electric field the source current has to be subtracted. In line 70 the source time is calculated taking into account the temporal shift between current and electric field. In line 71 the source amplitude is calculated according to equation (3). A complex exponential function is used for the oscillating carrier. This yields a complex source amplitude and causes the whole FDTD implementation to use complex fields. Complex fields are actually quite convenient: they allow an easy calculation of the time averaged Poynting vector and they give access to the instantaneous amplitude and phase at every point in time and space. However, this convenience comes at a price: the complex fields require twice as much memory as real valued fields and complex addition and subtraction operations take twice as time as real operations. A real valued FDTD implementation can be obtained by replacing the complex exponential function in line 71 with a cosine function. The update of the electric field with the current in line 73 only affects a single point of the array at the index `source_ind`.

Finally, in line 76 the magnetic field is updated with the electric field from the *current* iteration. Please note that all spatial coordinates of the magnetic field are updated. There are no boundary values that have to be preserved.

At the end of the 1D-FDTD implementation, the electric and the magnetic fields are transformed to a common coordinate system by interpolating the magnetic field from fractional indices to integer indices:

`fDTD_1d.m`

```
85 Hy = cat(1, Hy(1, :), Hy, Hy(end, :));
86 Hy(:, 2:end) = 0.5*(Hy(:, 1:end-1) + Hy(:, 2:end));
87 Hy = 0.5*(Hy(1:end-1, :) + Hy(2:end, :));
```

First the missing boundary values are replicated according to equation (15). Then the magnetic field is interpolate along the time axis. And finally it is interpolated along the spatial dimension.

The interpolation of the magnetic field is important in order to preserve the second order temporal accuracy of the FDTD method in calculations that involve both the electric and the magnetic field (e.g. the calculation of the Poynting vector).

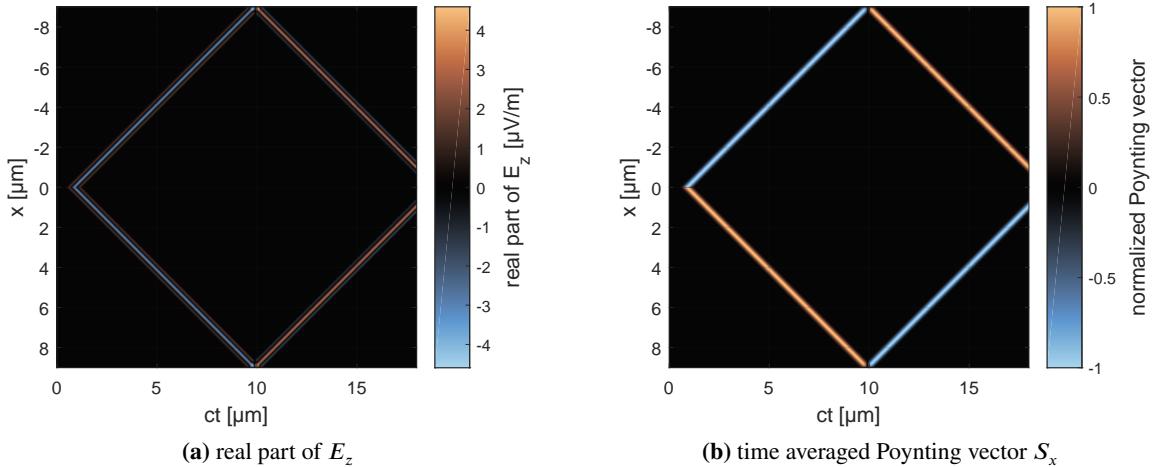
## 1.2 Results

The implementation of the 1D-FDTD method has been tested by simulating the propagation of the electromagnetic fields both in a homogeneous medium with refractive index  $n = 1$  and in presence of an interface between two dielectric media with refractive indices  $n_1 = 1$  and  $n_2 = 2$ . The width of the computational domain was  $18 \mu\text{m}$  and the interface was located at  $x_{\text{if}} = 4.5 \mu\text{m}$ . The current source was placed in the center of the computational domain at  $x_s = 0$ . The carrier frequency was  $f_0 = 500 \text{ THz}$  and the width of the Gaussian envelope was  $\tau = 1 \text{ fs}$ . The grid spacing was set to  $\Delta x = 30 \text{ nm}/n_2$  corresponding to a spatial resolution in the high index medium of about 20 grid points per wavelength at the carrier frequency. The test case is implemented in the file `test_fDTD_1d.m`. All parameters are summarized in table 1.

The time traces of the real part of the electric field and the  $x$ -component of the time averaged Poynting vector

$$S_x = -\frac{1}{2} \Re \{ E_z H_y^* \} \quad (16)$$

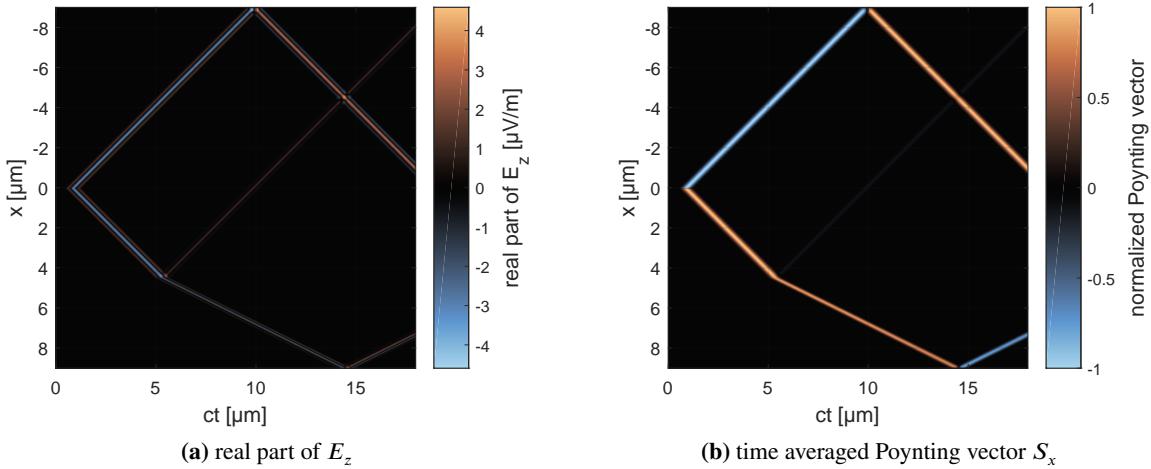
parameter	value	test script variable
width of computational domain	$W = 18 \mu\text{m}$	x_span
position of interface	$x_{\text{if}} = W/4$	x_interface
refractive index in front of interface	$n_1 = 1$	n1
refractive index behind interface	$n_2 = 2$	n2
grid spacing	$\Delta x = 30 \text{ nm} / \max(n_1, n_2)$	dx
duration of simulation	$T = 60 \text{ fs}$	time_span
source position	$x_s = 0$	source_position
source carrier frequency	$f_0 = 500 \text{ THz}$	source_frequency
source pulse width	$\tau = 1 \text{ fs}$	source_pulse_length

**Table 1:** Parameters of the one-dimensional test case.**Figure 2:** Time traces of the electric field and the Poynting vector excited by a pulsed current source at  $x = 0$  in a one dimensional homogeneous medium with refractive index  $n = 1$ . The computational domain is bounded by perfectly electric conducting walls.

of the homogeneous case are shown in figure 2. In the plots the time axis has been multiplied by the speed of light for an easier relation of the time to the the spatial position of the excited pulses.

The delta-shaped current source has no preferred direction of emission. Hence, both a forward traveling and a backward traveling pulse are excited. Both pulses have a symmetric field profile because the current pulse given by equation (3) has a carrier-envelope phase of zero that results in a symmetric real part of the current's temporal profile. The pulses are so short that their electric field has some unipolar character (i.e. their temporal average does not vanish). The refractive index of the homogeneous medium exhibits no dispersion. Thus the emitted pulses should not change their shape upon propagation. This is almost true.

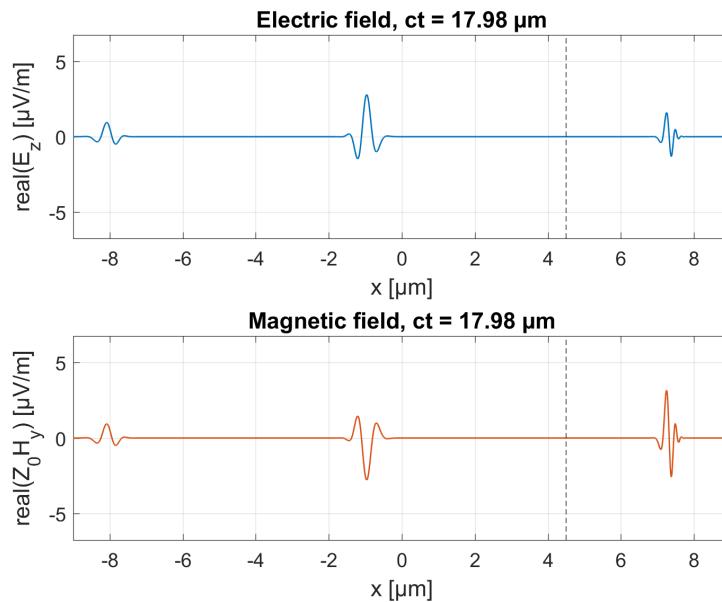
Some small asymmetry develops during the propagation due to the limited spatial resolution which is insufficient for the high frequency components of the pulse. The pulses are so short that they have some spectral amplitude even at the double carrier frequency. The sign of the power flow is constant across the pulses and is consistent with the propagation direction. When the pulses reach the boundaries of the computational domain they are perfectly reflected. Thereby the tangential electric field changes it's sign while the magnetic field does not change the sign (as can be deduced because the Poynting vector also changes the sign).



**Figure 3:** Time traces of the electric field and the Poynting vector excited by a pulsed current source at  $x = 0$  in a one dimensional domain containing an interface at  $x_{\text{if}} = 4.5 \mu\text{m}$  between two dielectric media with  $n_1 = n(x < x_{\text{if}}) = 1$  and  $n_2 = n(x \geq x_{\text{if}}) = 2$ . The computational domain is bounded by perfectly electric conducting walls.

The time traces of the real part of the electric field and the  $x$ -component of the time averaged Poynting vector of the case with an interface between two dielectric media with  $n(x < x_{\text{if}}) = n_1 = 1$  and  $n(x \geq x_{\text{if}}) = n_2 = 1$  are shown in figure 3. At the interface the forward traveling pulse is partially reflected. The fraction of the reflected power is consistent with the expectation from the Fresnel equation

$$R = \frac{|n_1 - n_2|^2}{|n_1 + n_2|^2} \approx 11.1 \%. \quad (17)$$



**Figure 4:** Video of the 1D FDTD simulation. The current source is located at  $x = 0 \mu\text{m}$ . The interface between the two dielectric media is indicated by the black vertical line at  $x = 4.5 \mu\text{m}$ . Depending on your pdf viewer you have to click, right-click or double-click on the image to access the attached video file. Most pdf viewers also have a separate menu item or side bar button (e.g. a paper clip) to view attached files.

In the high index medium the slope of the time traces is flatter than in the low index medium due to the reduced group velocity.

A snapshot of the video of the temporal evolution of electric and magnetic field is show in figure 4.

In the plots the magnetic field has been multiplied with the vacuum impedance to equalize the scales. The video illustrates that the pulses get slightly distorted during the propagation. As mentioned above, this is a numerical artifact due to the limited spatial resolution which is not sufficient for the high frequency components of the pulse. In the high index medium the ratio between the electric and the magnetic field is lower than in the low index medium because of the lower impedance  $Z_2 = \sqrt{\mu_0/\epsilon_0}/n_2$ .

## 2 Task II: 3D-FDTD

In three dimensions we have to integrate six coupled differential equations. Ampere's law yields

$$\frac{\partial H_z(\mathbf{r}, t)}{\partial y} - \frac{\partial H_y(\mathbf{r}, t)}{\partial z} = \epsilon_0 \epsilon_r(\mathbf{r}) \frac{\partial E_x(\mathbf{r}, t)}{\partial t} + j_x(\mathbf{r}, t), \quad (18)$$

$$\frac{\partial H_x(\mathbf{r}, t)}{\partial z} - \frac{\partial H_z(\mathbf{r}, t)}{\partial x} = \epsilon_0 \epsilon_r(\mathbf{r}) \frac{\partial E_y(\mathbf{r}, t)}{\partial t} + j_y(\mathbf{r}, t), \quad (19)$$

$$\frac{\partial H_y(\mathbf{r}, t)}{\partial x} - \frac{\partial H_x(\mathbf{r}, t)}{\partial y} = \epsilon_0 \epsilon_r(\mathbf{r}) \frac{\partial E_z(\mathbf{r}, t)}{\partial t} + j_z(\mathbf{r}, t). \quad (20)$$

And Faraday's law yields

$$\frac{\partial E_z(\mathbf{r}, t)}{\partial y} - \frac{\partial E_y(\mathbf{r}, t)}{\partial z} = -\mu_0 \frac{\partial H_x(\mathbf{r}, t)}{\partial t}, \quad (21)$$

$$\frac{\partial E_x(\mathbf{r}, t)}{\partial z} - \frac{\partial E_z(\mathbf{r}, t)}{\partial x} = -\mu_0 \frac{\partial H_y(\mathbf{r}, t)}{\partial t}, \quad (22)$$

$$\frac{\partial E_y(\mathbf{r}, t)}{\partial x} - \frac{\partial E_x(\mathbf{r}, t)}{\partial y} = -\mu_0 \frac{\partial H_z(\mathbf{r}, t)}{\partial t}. \quad (23)$$

Again these equations are discretized on a Yee grid and the derivatives are approximated by finite differences. In three dimension the shifting of the individual field components along the spatial and temporal coordinate axes is a bit more involved. The offsets are summarized in table 2.

**Table 2:** Offsets of the various field components along the different coordinate axes in the 3D FDTD algorithm. All offsets are given in multiples of the respective step sizes.

field	axis				field	axis			
	x	y	z	t		x	y	z	t
$\epsilon_r$	0	0	0	-	$j_x, j_y, j_z$	0	0	0	0.5
$E_x$	0.5	0	0	0	$H_x$	0	0.5	0.5	0.5
$E_y$	0	0.5	0	0	$H_y$	0.5	0	0.5	0.5
$E_z$	0	0	0.5	0	$H_z$	0.5	0.5	0	0.5

Applying these offsets to equations (18) to (20) and rearranging the resulting expressions yields the following update equations for the electric field components:

$$\begin{aligned} E_x|_{i+0.5,j,k}^{n+1} &= E_x|_{i+0.5,j,k}^n - \frac{\Delta t}{\epsilon_0} \frac{j_x|_{i,j,k}^{n+0.5} + j_x|_{i+1,j,k}^{n+0.5}}{\epsilon_r|_{i,j,k} + \epsilon_r|_{i+1,j,k}} \\ &+ \frac{\Delta t}{2\epsilon_0(\epsilon_r|_{i,j,k} + \epsilon_r|_{i+1,j,k})} \left[ \frac{H_z|_{i+0.5,j+0.5,k}^{n+0.5} - H_z|_{i+0.5,j-0.5,k}^{n+0.5}}{\Delta y} - \frac{H_y|_{i+0.5,j,k+0.5}^{n+0.5} - H_y|_{i+0.5,j,k-0.5}^{n+0.5}}{\Delta z} \right], \end{aligned} \quad (24)$$

$$\begin{aligned} E_y|_{i,j+0.5,k}^{n+1} = & E_y|_{i,j+0.5,k}^n - \frac{\Delta t}{\epsilon_0} \frac{j_y|_{i,j,k}^{n+0.5} + j_y|_{i,j+1,k}^{n+0.5}}{\epsilon_r|_{i,j,k} + \epsilon_r|_{i,j+1,k}} \\ & + \frac{\Delta t}{2\epsilon_0(\epsilon_r|_{i,j,k} + \epsilon_r|_{i,j+1,k})} \left[ \frac{H_x|_{i,j+0.5,k+0.5}^{n+0.5} - H_x|_{i,j+0.5,k-0.5}^{n+0.5}}{\Delta z} - \frac{H_z|_{i+0.5,j+0.5,k}^{n+0.5} - H_z|_{i-0.5,j+0.5,k}^{n+0.5}}{\Delta x} \right], \end{aligned} \quad (25)$$

$$\begin{aligned} E_z|_{i,j,k+0.5}^{n+1} = & E_z|_{i,j,k+0.5}^n - \frac{\Delta t}{\epsilon_0} \frac{j_z|_{i,j,k}^{n+0.5} + j_z|_{i,j,k+1}^{n+0.5}}{\epsilon_r|_{i,j,k} + \epsilon_r|_{i,j,k+1}} \\ & + \frac{\Delta t}{2\epsilon_0(\epsilon_r|_{i,j,k} + \epsilon_r|_{i,j,k+1})} \left[ \frac{H_y|_{i+0.5,j,k+0.5}^{n+0.5} - H_y|_{i-0.5,j,k+0.5}^{n+0.5}}{\Delta x} - \frac{H_x|_{i,j+0.5,k+0.5}^{n+0.5} - H_x|_{i,j-0.5,k+0.5}^{n+0.5}}{\Delta y} \right]. \end{aligned} \quad (26)$$

Please note that now also the electric field components are shifted along one spatial coordinate. Hence, the relative permittivity and the current densities have to be interpolated. The update equations for the magnetic field components that result from equations (21) to (23) are

$$H_x|_{i,j+0.5,k+0.5}^{n+1.5} = H_x|_{i,j+0.5,k+0.5}^{n+0.5} - \frac{\Delta t}{\mu_0} \left[ \frac{E_z|_{i,j+1,k+0.5}^{n+1} - E_z|_{i,j,k+0.5}^{n+1}}{\Delta y} - \frac{E_y|_{i,j+0.5,k+1}^{n+1} - E_y|_{i,j+0.5,k}^{n+1}}{\Delta z} \right], \quad (27)$$

$$H_y|_{i+0.5,j,k+0.5}^{n+1.5} = H_y|_{i+0.5,j,k+0.5}^{n+0.5} - \frac{\Delta t}{\mu_0} \left[ \frac{E_x|_{i+0.5,j,k+1}^{n+1} - E_x|_{i+0.5,j,k}^{n+1}}{\Delta z} - \frac{E_z|_{i+1,j,k+0.5}^{n+1} - E_z|_{i,j,k+0.5}^{n+1}}{\Delta x} \right], \quad (28)$$

$$H_z|_{i+0.5,j+0.5,k}^{n+1.5} = H_z|_{i+0.5,j+0.5,k}^{n+0.5} - \frac{\Delta t}{\mu_0} \left[ \frac{E_y|_{i+1,j+0.5,k}^{n+1} - E_y|_{i,j+0.5,k}^{n+1}}{\Delta x} - \frac{E_x|_{i+0.5,j+1,k}^{n+1} - E_x|_{i+0.5,j,k}^{n+1}}{\Delta y} \right]. \quad (29)$$

In the following we will assume the same grid spacing along all spatial dimension, i.e.  $\Delta x = \Delta y = \Delta z = \Delta r$ . The computational domain is defined by the distribution of the realtive permittivity that is specified on the integer grid with a size of  $(N_x, N_y, N_z)$ . Along the axes with fractional integer coordinates only the inner points with coordinates  $1.5 \dots N - 0.5$  are stored. Along the axes with integer coordinates all points including the boundary points are stored. This yields the following array sizes for storing field components at each iteration step:

$$E_x : (N_x - 1, N_y, N_z), \quad E_y : (N_x, N_y - 1, N_z), \quad E_z : (N_x - 1, N_y, N_z - 1), \quad (30)$$

$$H_x : (N_x, N_y - 1, N_z - 1), \quad H_y : (N_x - 1, N_y, N_z - 1), \quad H_z : (N_x - 1, N_y - 1, N_z). \quad (31)$$

As in the 1D implementation we choose PEC boundary conditions to terminate the computational domain. This means that along each axis the tangential electric field components and the normal magnetic field component are set to zero at the boundaries and are kept fixed, i.e. only the inner grid points of the respective fields are update.

Both to simplify the notation and to ease the implementation, it is convenient to introduce interpolated current densities and inverse permittivities (see also table 3):

$$\tilde{j}_x|_{i+0.5,j,k}^{n+0.5} = \frac{1}{2} \left[ j_x|_{i+1,j,k}^{n+0.5} + j_x|_{i,j,k}^{n+0.5} \right], \quad \frac{1}{\epsilon_x|_{i+0.5,j,k}} = \frac{1}{2} \left[ \frac{1}{\epsilon_r|_{i+1,j,k}} + \frac{1}{\epsilon_x|_{i,j,k}} \right], \quad (32)$$

$$\tilde{j}_y|_{i,j+0.5,k}^{n+0.5} = \frac{1}{2} \left[ j_y|_{i,j+1,k}^{n+0.5} + j_y|_{i,j,k}^{n+0.5} \right], \quad \frac{1}{\epsilon_y|_{i,j+0.5,k}} = \frac{1}{2} \left[ \frac{1}{\epsilon_r|_{i,j+1,k}} + \frac{1}{\epsilon_x|_{i,j,k}} \right], \quad (33)$$

$$\tilde{j}_z|_{i,j,k+0.5}^{n+0.5} = \frac{1}{2} \left[ j_z|_{i,j,k+1}^{n+0.5} + j_z|_{i,j,k}^{n+0.5} \right], \quad \frac{1}{\epsilon_z|_{i,j,k+0.5}} = \frac{1}{2} \left[ \frac{1}{\epsilon_r|_{i,j,k+1}} + \frac{1}{\epsilon_x|_{i,j,k}} \right]. \quad (34)$$

**Table 3:** Offsets of the interpolated permittivities and current densities along the different coordinate axes in the 3D FDTD algorithm. All offsets are given in multiples of the respective step sizes.

field	axis				field	axis			
	x	y	z	t		x	y	z	t
$\epsilon_x$	0.5	0	0	-	$\tilde{j}_y$	0.5	0	0	0.5
$\epsilon_y$	0	0.5	0	-	$\tilde{j}_y$	0	0.5	0	0.5
$\epsilon_z$	0	0	0.5	-	$\tilde{j}_z$	0	0	0.5	0.5

With these and after renaming the fractional indices  $i + 0.5 \rightarrow i$ ,  $j + 0.5 \rightarrow j$ ,  $k + 0.5 \rightarrow k$ ,  $n + 0.5 \rightarrow k$ , the update equations can be written as

$$E_x|_{i,j,k}^{n+1} = E_x|_{i,j,k}^n + \frac{\Delta t}{\epsilon_0 \epsilon_x|_{i,j,k}} \left[ \frac{H_z|_{i,j,k}^n - H_z|_{i,j-1,k}^n}{\Delta y} - \frac{H_y|_{i,j,k}^n - H_y|_{i,j,k-1}^n}{\Delta z} - \tilde{j}_x|_{i,j,k} \right] \quad \begin{array}{l} i = 1 \dots N_x - 1 \\ \text{for } j = 2 \dots N_y - 1, \\ k = 2 \dots N_z - 1 \end{array} \quad (35)$$

$$E_y|_{i,j,k}^{n+1} = E_y|_{i,j,k}^n + \frac{\Delta t}{\epsilon_0 \epsilon_y|_{i,j,k}} \left[ \frac{H_x|_{i,j,k}^n - H_x|_{i,j,k-1}^n}{\Delta z} - \frac{H_z|_{i,j,k}^n - H_z|_{i-1,j,k}^n}{\Delta x} - \tilde{j}_y|_{i,j,k} \right] \quad \begin{array}{l} i = 2 \dots N_x - 1 \\ \text{for } j = 1 \dots N_y - 1, \\ k = 2 \dots N_z - 1 \end{array} \quad (36)$$

$$E_z|_{i,j,k}^{n+1} = E_z|_{i,j,k}^n + \frac{\Delta t}{\epsilon_0 \epsilon_z|_{i,j,k}} \left[ \frac{H_y|_{i,j,k}^n - H_y|_{i-1,j,k}^n}{\Delta x} - \frac{H_x|_{i,j,k}^n - H_x|_{i,j-1,k}^n}{\Delta y} - \tilde{j}_z|_{i,j,k} \right] \quad \begin{array}{l} i = 2 \dots N_x - 1 \\ \text{for } j = 2 \dots N_y - 1, \\ k = 1 \dots N_z - 1 \end{array} \quad (37)$$

for the electric field components and

$$H_x|_{i,j,k}^{n+1} = H_x|_{i,j,k}^n - \frac{\Delta t}{\mu_0} \left[ \frac{E_z|_{i,j+1,k}^{n+1} - E_z|_{i,j,k}^{n+1}}{\Delta y} - \frac{E_y|_{i,j,k+1}^{n+1} - E_y|_{i,j,k}^{n+1}}{\Delta z} \right] \quad \begin{array}{l} i = 2 \dots N_x - 1 \\ \text{for } j = 1 \dots N_y - 1, \\ k = 1 \dots N_z - 1 \end{array} \quad (38)$$

$$H_y|_{i,j,k}^{n+1} = H_y|_{i,j,k}^n - \frac{\Delta t}{\mu_0} \left[ \frac{E_x|_{i,j,k+1}^{n+1} - E_x|_{i,j,k}^{n+1}}{\Delta z} - \frac{E_z|_{i+1,j,k}^{n+1} - E_z|_{i,j,k}^{n+1}}{\Delta x} \right] \quad \begin{array}{l} i = 1 \dots N_x - 1 \\ \text{for } j = 2 \dots N_y - 1, \\ k = 1 \dots N_z - 1 \end{array} \quad (39)$$

$$H_z|_{i,j,k}^{n+1} = H_z|_{i,j,k}^n - \frac{\Delta t}{\mu_0} \left[ \frac{E_x|_{i,j+1,k}^{n+1} - E_x|_{i,j,k}^{n+1}}{\Delta y} - \frac{E_y|_{i,j,k+1}^{n+1} - E_y|_{i,j,k}^{n+1}}{\Delta z} \right] \quad \begin{array}{l} i = 1 \dots N_x - 1 \\ \text{for } j = 1 \dots N_y - 1, \\ k = 2 \dots N_z - 1 \end{array} \quad (40)$$

for the magnetic field components.

## 2.1 Implementation

The 3D-FDTD method is implemented in the file `fDTD_3d.m`. Some of the key implementation details will be discussed in the following.

In three dimension more fields have to be stored and the arrays of the fields are much larger than in one dimension. Hence, the whole algorithm is implemented using single precision floating point numbers instead of the usual double precision numbers. This should not limit the accuracy of the algorithm. In most cases the error due to the discretization will be larger than the round-off error due to the limited precision. Using single precision floating point numbers saves half of the memory and most of the numerical operations are twice as fast. Due to the large size of the arrays the fields can no longer be stored at every iteration step. Instead only a  $z$ -slice of one field component at a given  $z$ -index `z_ind` is stored every `output_step` time steps.

As in the one dimensional implementation, the size of the computational domain is derived from the relative permittivity `eps_rel`:

## fdtd\_3d.m

```
50 [Nx, Ny, Nz] = size(eps_rel);
```

The number of time steps is given by the duration of the simulation. But it is rounded to an integer number of outputs. An appropriately sized output array is allocated to store the slices of the requested field component:

## fdtd\_3d.m

```
54 Niter = round(time_span/Delta_t/output_step)*output_step;
55 t = (0:output_step:Niter)*Delta_t;
56 F = zeros(Nx, Ny, length(t), 'single');
```

The inverse relative permittivity and the current densities are interpolated to the shifted grids of the respective electric field component,

## fdtd\_3d.m

```
59 eps_rel = single(1.0)./eps_rel;
60 iepsx = fdtd_3d_interpolate_to_E_yee_grid(eps_rel, 'x');
61 iepsy = fdtd_3d_interpolate_to_E_yee_grid(eps_rel, 'y');
62 iepsz = fdtd_3d_interpolate_to_E_yee_grid(eps_rel, 'z');
63 clear eps_rel;
64
65 jx = fdtd_3d_interpolate_to_E_yee_grid(jx, 'x');
66 jy = fdtd_3d_interpolate_to_E_yee_grid(jy, 'y');
67 jz = fdtd_3d_interpolate_to_E_yee_grid(jz, 'z');
```

and field arrays that hold the fields at the current time step are allocated with sizes as described above. The keyword '`'single'`' has to be specified to obtain the desired precision:

## fdtd\_3d.m

```
70 Ex = zeros(Nx-1, Ny, Nz, 'single');
71 Ey = zeros( Nx, Ny-1, Nz, 'single');
72 Ez = zeros( Nx, Ny, Nz-1, 'single');
73 Hx = zeros( Nx, Ny-1, Nz-1, 'single');
74 Hy = zeros(Nx-1, Ny, Nz-1, 'single');
75 Hz = zeros(Nx-1, Ny-1, Nz, 'single');
```

For convenience index arrays are created that simplify the notation of the field updates:

## fdtd\_3d.m

```
77 % valid indices for unshifted axes (without boundary values)
78 iux = 2:Nx-1;
79 iuy = 2:Ny-1;
80 iuz = 2:Nz-1;
81
82 % indices for derivatives in E step
83 iuxm1 = 1:Nx-2;
84 iuym1 = 1:Ny-2;
85 iuzm1 = 1:Nz-2;
86
87 % valid indices for shifted axes
88 isx = 1:Nx-1;
89 isy = 1:Ny-1;
90 issz = 1:Nz-1;
91
```

```

92 % indices for derivatives in H step
93 isxp1 = 2:Nx;
94 isyp1 = 2:Ny;
95 iszp1 = 2:Nz;

```

The time stepping is performed by a `for`-loop just like in the 1D implementation. However, this time six field components have to be updated at each iteration and the expressions for the updates are a more complicated. This increases the length of the loop body a bit. Nonetheless, the basic logic is the same.

At the beginning of the loop the amplitude of the source is calculated:

```

fDTD_3d.m
105 t_source = (n + 0.5)*Delta_t - 3*tau;
106 source_factor = single(e_factor*exp(-1i*2*pi*freq*t_source)...
107 *exp(-(t_source/tau)^2));

```

After this the inner grid points of the electric field components are updated. For example the update of  $E_x$  looks as follows:

```

fDTD_3d.m
110 % Delta_t/eps0*dHz/dy
111 U = e_factor/dr*(Hz(isx,iuy,iuz) - Hz(isx,iuym1,iuz));
112 % - Delta_t/eps0*dHy/dz
113 U = U - e_factor/dr*(Hy(isx,iuy,iuz) - Hy(isx,iuym1,iuz));
114 % - Delta_t/eps0*jx (interpolated to Ex grid)
115 U = U - source_factor*jx(isx,iuy,iuz);
116 % divide by eps_rel (interpolated to Ex grid)
117 U = U.*iepsx(isx,iuy,iuz);
118 % + Ex(t=n*dt)
119 U = U + Ex(isx,iuy,iuz);
120 if mod(n + 1, output_step) == 0 && strcmpi(field_component, 'ex')
121     F(:, :, next_out) = fDTD_3d_interpolate_field(...
122                                         U, field_component, z_ind);
123     next_out = next_out + 1;
124 end
125 Ex(isx,iuy,iuz) = U; %store update without changing boundary values

```

For efficiency reasons the operations are performed in a different order than in equation (35). In lines 111 and 113 the derivatives of the magnetic field components are calculated on the inner grid points without the boundary values and are stored in a temporary array  $U$ . In line 115 the source term is subtracted from this array before it is multiplied with the interpolated inverse permittivity in line 117. Finally, in line 119 the values of the old field at the inner grid points are added to the new field. The calculation of the updated field is followed by a test if the current field component at the current time step has to be stored in the output array. If this is the case, the field is interpolated to the grid of  $\text{eps\_rel}$  before a  $z$ -slice at the  $z$ -index specified by  $z\_ind$  is extracted. The interpolation and slice extraction is performed by the auxiliary function `fDTD_3d_interpolate_field`. This function also takes care of handling the boundary in dependence of the field component by repeating boundary values or by adding zeros as only the inner grid points are passed to the function. In the end, the updated field is written back to the original field array. The remaining electric field components  $E_y$  and  $E_z$  are handled analogously. After the electric

**Table 4:** Parameters of the three-dimensional test case.

parameter	value	test script variable
grid size	$N_x = 199, N_y = 201, N_z = 5$ ,	Nx, Ny, Nz
grid spacing	$\Delta r = 30 \text{ nm}$	dr
duration of simulation	$T = 10 \text{ fs}$	time_span
source carrier frequency	$f_0 = 500 \text{ THz}$	freq
source pulse width	$\tau = 1 \text{ fs}$	tau
spatial source width	$w = 2\Delta r$	source_width
z-index of output plane	$k_{\text{out}} = \text{ceil}((N_z - 1)/2)$	z_ind
output interval	$\Delta n_{\text{out}} = 4$	output_step

field components have been updated, the new magnetic field components are calculated. The update of  $H_x$  looks as follows:

```

fDTD_3d.m
164 U = Hx(iux,isy,isz); % old value at t=(n+0.5)*dt
165 % - Delta_t/mu_0*dEz/dy, only valid values of Ez along x
166 U = U - h_factor/dr*(Ez(iux,isyp1,isz) - Ez(iux,isy,isz));
167 % + Delta_t/mu_0*dEy/dz, only valid values of Ey along x
168 U = U + h_factor/dr*(Ey(iux,isy,iszp1) - Ey(iux,isy,isz));
169 if mod(n + 1, output_step) == 0 && strcmpi(field_component, 'hx')
170     % interpolation to t=(n+1)*dt: average of old and new field
171     F(:, :, next_out) = fDTD_3d_interpolate_field(...
172                             f*(Hx(iux,isy,isz) + U),...
173                             field_component, z_ind);
174 next_out = next_out + 1;
175 end

```

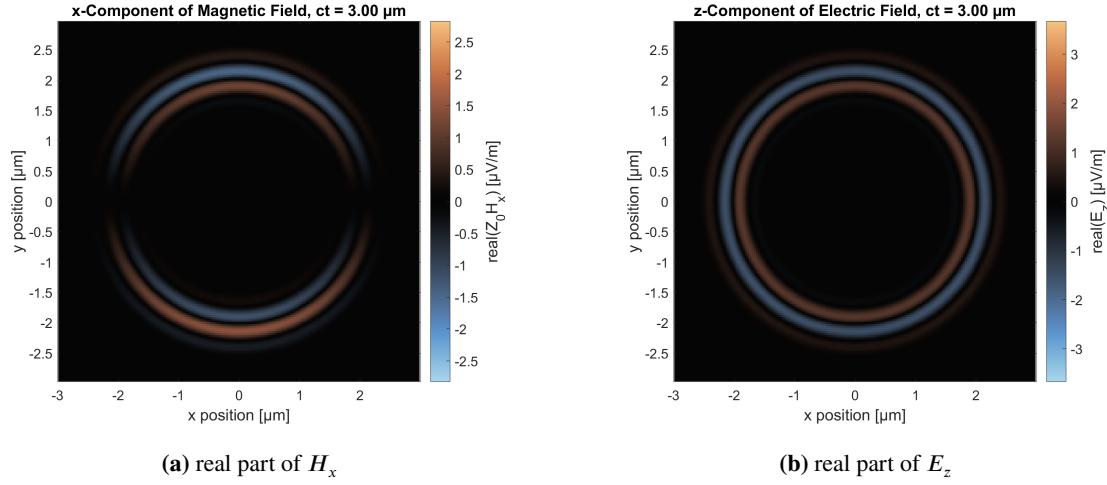
It is a literal translation of equation (38). In the case that the magnetic field component has to be stored in the output array, `fDTD_3d_interpolate_field` is called with the average value of the old field and the new field. This is done to obtain the magnetic field at the same integer time steps where the electric field components are stored. The reason why this is necessary has been explained at the end of section 1.1. The remaining magnetic field components  $H_y$  and  $H_z$  are handled analogously.

A 3D-FDTD calculation usually takes much longer than a one dimensional calculation. For this reason some code to create a progress report in regular intervals is added at the end of the loop.

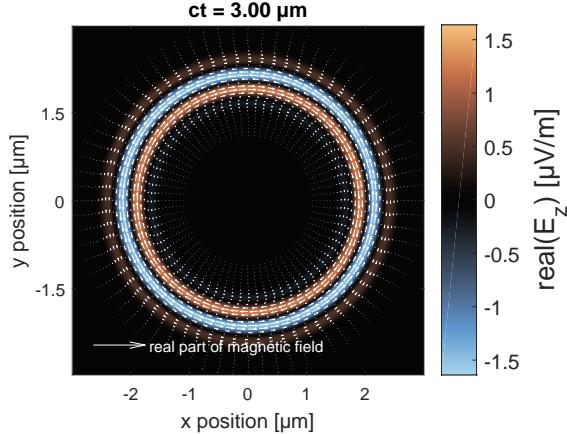
## 2.2 Results

The implementation of the 3D-FDTD method has been tested by simulating the propagation of the electromagnetic fields in a homogeneous medium with refractive index  $n = 1$ . To save memory, a very small extent of the computational grid of only 5 grid points along the  $z$ -direction was chosen. The grid resolution was set to  $\Delta r = 30 \text{ nm}$ . A  $z$ -polarized line current in the center of the computational domain with a Gaussian spatial profile along  $x$  and  $y$  as specified by equation (5) was used for excitation. The spatial width of the source was two grid points. The temporal pulse parameters were the same as in the 1D test case. The simulation duration was  $T = 10 \text{ fs}$ . The three dimensional test case is implemented in the file `test_fDTD_3d.m`. All parameters are summarized in table 4.

The  $z$ -polarized current source excites only the  $z$ -component of the electric field. As the current density is constant along the  $z$ -axis, and as the  $z$ -component of the electric field is not affected by the PEC boundaries in  $z$ -direction, no  $x$  and  $y$  components of the electric field and no  $z$  component of the magnetic



**Figure 5:** Videos of the 3D FDTD simulation. Depending on your pdf viewer you have to click, right-click or double-click on the image to access the attached video file. Most pdf viewers also have a separate menu item or side bar button (e.g. a paper clip) to view attached files.



**Figure 6:** Electric field (heatmap) and magnetic field (vectors) of the 3D test case at the end of the simulation after  $t = 15 \text{ fs}$ .

field are created. The whole setup corresponds to a quasi-2D simulation of an infinitely extended line current.

Snapshots of the videos showing the temporal evolution of  $H_x$  and  $E_z$  are shown in figure 5.

The electromagnetic fields excited by the current pulse in the center of the computational domain spread radially outwards. The simulation stops shortly before the emitted pulse reaches the boundaries.  $H_x$  exhibits an odd mirror symmetry with respect to the  $xz$ -plane through the origin. It is zero on the  $x$ -axis and has the largest magnitude on the  $y$ -axis. The electric field is rotationally symmetric with respect to the  $z$ -axis.  $H_y$  (not shown) looks the same as  $H_x$  only rotated by  $90^\circ$ . The vector plot of the magnetic field shown in figure 6 reveals that the magnetic field is actually  $\varphi$ -polarized, i.e. it is tangential to concentric circles centered at the source position. This is indeed expected from an  $z$ -polarized infinitely along the  $z$ -axis extended line current.