Report 6: The Yee algorithm in one dimension

Don Winter[1] **and Madhulan Suresh**[2]

[1]don.winter@rwth-aachen.de, Mat. No. 431380
[2]madhulan.suresh@rwth-aachen.de, Mat. No. 429481

Introduction

In this exercise we simulate electric and magnetic fields according to Maxwell's equations by the Yee algorithm. This algorithm makes use of a special grid to intertwine discrete electric and magnetic field components in space and time to efficiently integrate them in a leap-frog scheme. We first derive the update equations for the algorithm from the Maxwell equations in one dimension, then introduce our simulation setup and finally present our results.

Simulation model and method

For the simulation we consider $\hat{z}$-polarized plane waves propagating in the $\hat{x}$-direction through linear, isotropic and nondispersive matter and vacuum. Furthermore, we neglect charge density ρ_e inside materials. In this case the Maxwell equations reduce to

$$\begin{aligned}\frac{\partial E_z(x,t)}{\partial t} &= \frac{1}{\varepsilon(x)}\left(\frac{\partial H_y(x,t)}{\partial x} - J_z(x,t)\right)\\ \frac{\partial H_y(x,t)}{\partial t} &= \frac{1}{\mu(x)}\left(\frac{\partial E_z(x,t)}{\partial x} - M_z(x,t)\right),\end{aligned} \tag{1}$$

where $\varepsilon(x) = \varepsilon_0\varepsilon_r(x)$ and $\mu(x) = \mu_0\mu_r(x)$ are the electrical permittivity and the magnetic permeability of respective materials at location x. Furthermore, we introduce an external electric source producing current density $J_z^{(source)}(x,t)$ in the simulation space of electrical conductivity $\sigma(x)$ and magnetic loss $\sigma^*(x)$

$$\begin{aligned}J_z(x,t) &= J_z^{(source)}(x,t) + \sigma(x)E_z(x,t)\\ M_z(x,t) &= M_z^{(source)}(x,t) + \sigma^*(x)H(x,t) = \sigma^*(x)H(x,t),\end{aligned} \tag{2}$$

where $M_z^{(source)}(x,t) = 0$ as the source excites the electric field only. In order to run the simulation we need to discretize Eqs. 1 with respect to both dependent variables time and space. For this end, we approximate the partial derivatives by central differences:

$$\begin{aligned}\frac{E_z(x,t+\Delta t) - E_z(x,t-\Delta t)}{2\Delta t} &= \frac{1}{\varepsilon(x)}\left(\frac{H_y(x+\Delta x,t) - H_y(x-\Delta x,t)}{2\Delta x} - J_z^{(source)}(x,t) - \sigma(x)E_z(x,t+\Delta t)\right)\\ \frac{H_y(x,t+\Delta t) - H_y(x,t-\Delta t)}{2\Delta t} &= \frac{1}{\mu(x)}\left(\frac{E_z(x+\Delta x,t) - E_z(x-\Delta x,t)}{2\Delta x} - \sigma^*(x)H_y(x,t+\Delta t)\right).\end{aligned} \tag{3}$$

A smart choice now is to distribute the discrete electric and magnetic field components in space and time according to the Yee grid. We define $\Delta = 2\Delta x$, $\tau = 2\Delta t$, $t = n\tau$, $x = l\Delta$ and move $E_z(x,t) = E_z(l\Delta, n\tau) \to E_z(l\Delta,(n+1/2)\tau)$ and $H_y(x,t) = H_y(l\Delta, n\tau) \to H_y((l+1/2)\Delta, n\tau)$, for which we introduce the grid values $E_z|_l^{n+1/2}$ and $H_y|_{l+1/2}^n$, respectively. With this we get from Eqs. 3 for the electric field after approximating the field value $E_z|_l^n$ which we are not tracking by the mean of its neighbors $E_z|_l^{n-1/2}$ and $E_z|_l^{n+1/2}$ which we do track:

$$\begin{aligned}E_z|_l^{n+1/2} &= E_z|_l^{n-1/2} + \frac{\tau}{\varepsilon_l}\left(\frac{H_y|_{l+1/2}^n - H_y|_{l-1/2}^n}{\Delta} - J_z^{(source)}|_l^n - \sigma_l\frac{E_z|_l^{n-1/2} + E_z|_l^{n+1/2}}{2}\right)\\ &= \left(\frac{1-\frac{\sigma_l\tau}{2\varepsilon_l}}{1+\frac{\sigma_l\tau}{2\varepsilon_l}}\right)E_z|_l^{n-1/2} + \left(\frac{\frac{\tau}{\varepsilon_l}}{1+\frac{\sigma_l\tau}{2\varepsilon_l}}\right)\left(\frac{H_y|_{l+1/2}^n - H_y|_{l-1/2}^n}{\Delta} - J_z^{(source)}|_l^n\right),\end{aligned} \tag{4}$$

and for the magnetic field we have by approximating $H_y|_l^n$ in a similar way:

$$
\begin{aligned}
H_y|_{l+1/2}^{n+1} &= H_y|_{l+1/2}^{n} + \frac{\tau}{\mu_{l+1/2}}\left(\frac{E_z|_{l+1}^{n+1/2} - E_z|_{l}^{n+1/2}}{\Delta} - \sigma^*_{l+1/2}\frac{H_y|_{l+1/2}^{n+1} + H_y|_{l+1/2}^{n}}{2}\right) \\
&= \left(\frac{1 - \frac{\sigma^*_{l+1/2}\tau}{2\mu_{l+1/2}}}{1 + \frac{\sigma^*_{l+1/2}\tau}{2\mu_{l+1/2}}}\right) H_y|_{l+1/2}^{n} + \left(\frac{\frac{\tau}{\mu_{l+1/2}}}{1 + \frac{\sigma^*_{l+1/2}\tau}{2\mu_{l+1/2}}}\right)\left(\frac{E_z|_{l+1}^{n+1/2} - E_z|_{l}^{n+1/2}}{\Delta}\right),
\end{aligned}
\tag{5}
$$

which define the update rules for our algorithm. Furthermore, we choose a one-dimensional grid of $L = 5000$ points in which $N = 50$ points correspond to one wavelength $\lambda = 1$. Our whole grid thus covers $X = 100\ \lambda$s and hence has spatial resolution $\Delta = \lambda/N = 0.02$. For conductivity and magnetic loss we define $\sigma(x) = \sigma^*(x) = 0$ for $6\lambda < x < X - 6\lambda$ and $\sigma(x) = \sigma^*(x) = 1$ for $x \leq 6\lambda$ and $x \geq X - 6\lambda$, which translates to $\sigma_l = \sigma_l^* = 0$ for $6\lambda/\Delta = 300 < l < L - 6\lambda/\Delta = 4700$ and $\sigma_l = \sigma_l^* = 1$ for $l \leq 6\lambda/\Delta = 300$ and $l \geq L - \lambda/\Delta = 4700$ in our grid. These conditions create reflectionless, completely absorbing boundaries of our simulation box while having lossless wave propagation on the inside. Moreover, we place a glass plate of refractive index $n = 1.46$ of thickness $\Delta x = 2\lambda \rightarrow \Delta l = 2\lambda/\Delta = 100$ in the middle of our box from $l = 2500$ to $l = 2500 + \Delta l = 2600$. In the rest of the box we assume vacuum $\varepsilon(x) = \mu(x) = 1$ where we used $\varepsilon_0 = \mu_0 = 1$ due to our choice of units $c = 1/\sqrt{\mu_0\varepsilon_0} = 1$. Our source is located at $x_s = 20\lambda \rightarrow l_s = 20\lambda/\Delta = 1000$ and produces a Gaussian-modulated sinusoidal pulse of frequency $f = 1$, which peaks in time at $t_0 = 30$ with spread $s = 10$

$$
J_z^{(source)}(l_s, t) = \sin(2\pi t f) e^{-((t-30)/10)^2}.
\tag{6}
$$

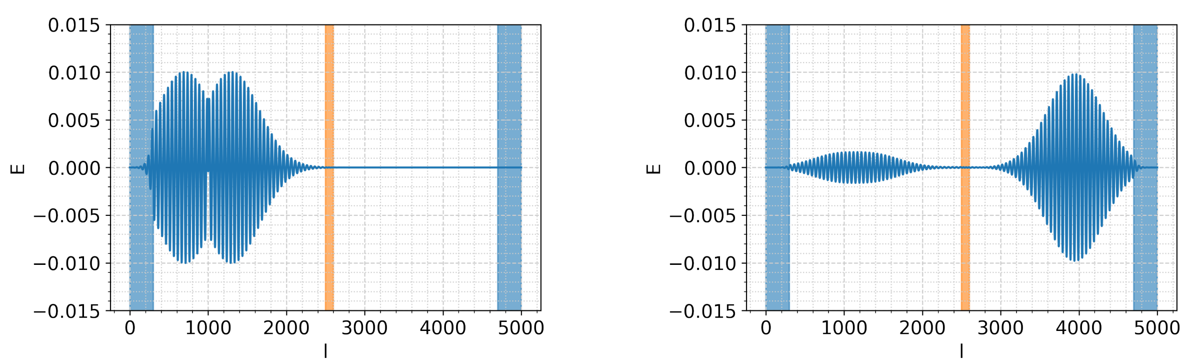
We simulate for $m = 10000$ time steps of time resolution $\tau = 0.9\Delta$. By this choice we assure the Yee algorithm yields stable results due to fullfillment of the Courant condition:

$$
S = \frac{c\tau}{|\Delta|} \leq 1,
\tag{7}
$$

for in our case $c = 1$. If we would violate the Courant condition, for example set $\tau = 1.05\Delta$, our propagating wave amplitude would accumulate with each time step and deviate more and more from the true amplitude. How fast this deviation appears will be illustrated, amongst other results, in the following section.

Simulation results

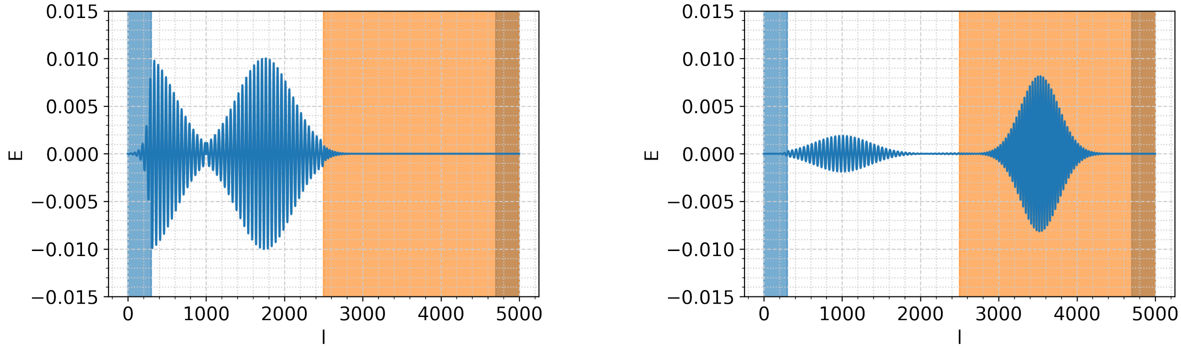
For the glass plate in the central position we reproduce the results from the exercise description which can be seen in Fig. 1 Now,



(a) Time step 2000 **(b)** Time step 5000

Figure 1. Electric field as a function of grid points l at two fixed time steps with thin glass plate.

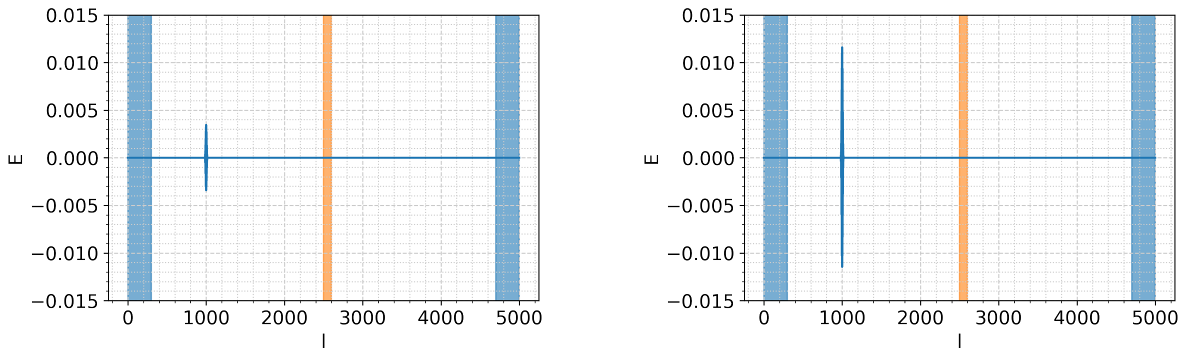
we increase the thickness of the glass plate to the right as much as possible. The result for two time steps can be seen in Fig. 2. We can visually estimate the maximum magnitude of the electric fields of the incident and reflected waves as $E_{incident}^{(max)} = 0.01$ and $E_{reflected}^{(max)} = 0.002$, respectively, from which we can calculate the reflection coeficient $R = |E_{reflected}^{(max)}|^2/|E_{incident}^{(max)}|^2 \approx 0.04$. Thus 4% of the light is reflected and the remaining 96% is transmitted. In comparison with Fig. 1 we can further observe the



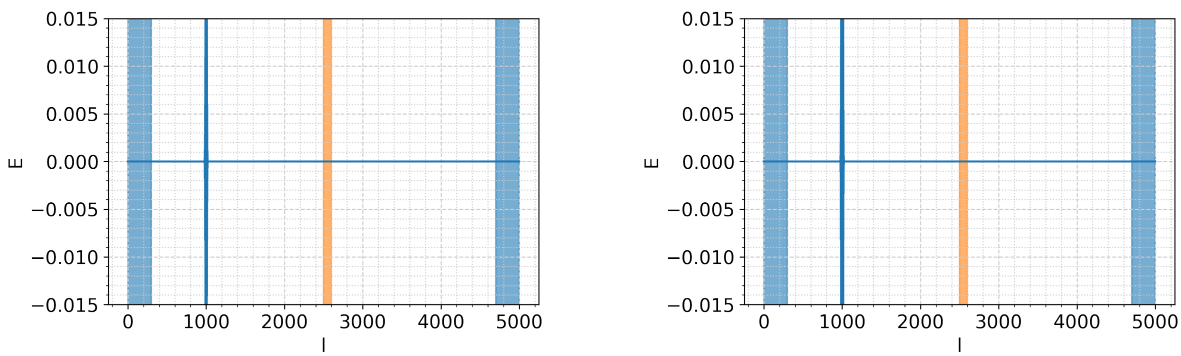
(a) Time step 2500 **(b)** Time step 5000

Figure 2. Electric field as a function of grid points l at two fixed time steps with thick glass plate.

loss which the material introduce on the field by noting the decrease in amplitude in Fig. 2. Next, let's take a look at what happens to Fig. 1 if we violate the Courant condition and set $\tau = 1.05\Delta$. For this we plot a few more steps of time evolution. The results are shown in Fig. 3. We can clearly see that within a few time steps after the source is activated the wave amplitude steeply increased, i.e. starts to "explode".



(a) Time step 20 **(b)** Time step 22



(c) Time step 24 **(d)** Time step 26

Figure 3. Electric field as a function of grid points l at four fixed time steps for violated Courant condition.

Discussion

In summary, we simulated the propagation of a Gaussian wave packet in one dimension through vacuum and through a glass plate of two thicknesses. As expected, we observed a large fraction of light being transmitted through the glass and only a small fraction being reflected. Furthermore, we observed that the Yee algorithm yields an unstable simulation if the Courant condition is not fullfilled, which manifests itself in "exploding" field amplitudes within only a few time steps.

Appendix

```python
import matplotlib.pyplot as plt
import numpy as np
import matplotlib
font = {'size': 14}
matplotlib.rc('font', **font)

L = 5_000 # no. grid points
m = 10_000 # no. time steps

l_src = 1_000
delta = 0.02
tau = 0.9*delta # case 1) fullfill Courant condition => stable
# tau = 1.05*delta # case 2) violate Courant cond. => instable

Ez = np.zeros(L+1) # Electric field components
Hy = np.zeros(L) # Magnetic field components

N = 50 # grid points per space step
n = 1.46 # index of refraction
eps = np.ones(L+1) # permittivity of simulation space
eps[L//2:L//2+2*N] = n**2 # case 1) thin glass plate in middle
# eps[L//2:] = n**2 # case 2) very thick glass plate

sigma = np.ones(L+1) # boundary conditions
sigma[6*N:-6*N] = 0 # everywhere 0 except close to bounds

A = (1-sigma[1:]*tau/2) / (1 + sigma[1:]*tau/2)
B = tau / (1 + sigma[1:]*tau/2)
C = (1-sigma*tau/(2*eps)) / (1+sigma*tau/(2*eps))
D = (tau/eps) / (1+sigma*tau/(2*eps))

def source(t):
    t0 = 30
    spread = 10
    return np.sin(2*np.pi*t)*np.exp(-((t-t0)/spread)**2)

for i in range(m):

    Ez[1:-1] = D[1:-1]*(Hy[1:]-Hy[:-1])/delta + C[1:-1]*Ez[1:-1] # Update E field + bound. cond.
    Ez[l_src] -= D[l_src]*source(i*tau) # Update field due to source
    Hy = B*(Ez[1:]-Ez[:-1])/delta + A*Hy # Update H field incl. bound. cond.

    if i in [2000,5000]:
    # if i in [20,22,24,26]:
        plt.figure(figsize=(6,4))
        plt.ylim([-0.015,0.015])
        plt.plot(Ez)

        plt.fill_between(range(L+1), sigma, alpha=0.6, color='tab:blue')
        plt.fill_between(range(L+1), -sigma, alpha=0.6, color='tab:blue')
        plt.fill_between(range(L+1), eps-1, alpha=0.6, color='tab:orange')
        plt.fill_between(range(L+1), -(eps-1), alpha=0.6, color='tab:orange')

        plt.xlabel('l')
        plt.ylabel('E')
        plt.minorticks_on()
        plt.grid(which='major', color='#CCCCCC', linestyle='--')
        plt.grid(which='minor', color='#CCCCCC', linestyle=':')
        plt.show()
```