

Electromagnetic scattering 1: Finite-difference time-domain method (FDTD)

Johannes Markkanen

University of Helsinki

September 9, 2014

Finite-difference time-domain (FDTD)

- Numerical technique for solving Maxwell's equations in time domain
- Maxwell's equations are directly discretized by finite-differences
- Central difference approximations to the space and time derivatives
- The space is approximated by cubical cells

Finite-difference time-domain (Yee-algorithm)

Strengths:

- Universal applicability
- Broadband response obtained with one simulation
- Inhomogeneous, anisotropic, non-linear materials are easily modelled
- Matrix inversion is not needed (recursive update scheme)
- Simple implementation
- Evolution of fields can be studied in time

Weaknesses:

- Accuracy -Error is dispersive and accumulates as waves propagate through the grid
- Staircase approximation
- Time-step is a function of grid size
- PML is needed for open region problems

FDTD literature

Introduction to FDTD, FEM, IEM

- Sheng Xin-Qing, Song Wei, Essentials of computational electromagnetics, IEEE, Wiley, 2012.

Some FDTD books

- Allen Taflove and Susan C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, Artech House Publishers, 2005.
- Wenhua Yu, Raj Mittra, Tao Su, Yongjun Liu, and Xiaoling Yang, Parallel Finite-Difference Time-Domain Method, Artech House Publishers, 2006.

Maxwell's equation

Faraday's law

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1)$$

Ampères law

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \sigma \mathbf{E} + \mathbf{J} \quad (2)$$

Gauss's law for electric field

$$\nabla \cdot \mathbf{D} = \rho \quad (3)$$

Gauss's law for magnetic field

$$\nabla \cdot \mathbf{B} = 0 \quad (4)$$

Constitutive relations:

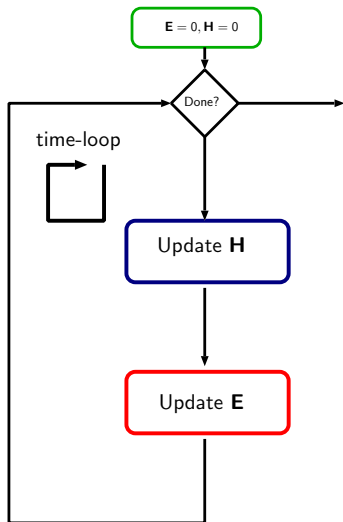
$$\mathbf{D} = \epsilon * \mathbf{E} \quad (5)$$

$$\mathbf{B} = \mu * \mathbf{H} \quad (6)$$

ϵ electric permittivity

μ magnetic permeability

FDTD engine (empty space)



Set $\mathbf{E} = 0$ and $\mathbf{H} = 0$

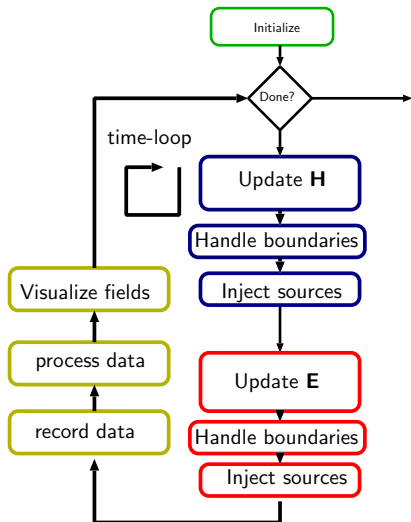
Marching-in-time procedure:

- Update \mathbf{H} :

$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t}$$
- Update \mathbf{E} :

$$\nabla \times \mathbf{H} = \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

FDTD engine (more realistic case)



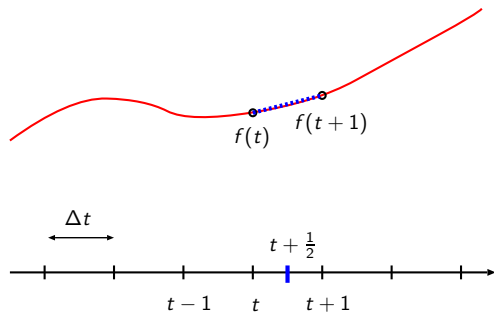
Initialize everything
 Marching-in-time procedure:

- Update **H**:
- PML
- Gaussin pulse
- Update **E**:
- PML
- Gaussin pulse
- Fourier transform, etc.

Update equations

Finite-difference formula (central difference)

$$\frac{\partial f(t + \frac{\Delta t}{2})}{\partial t} \approx \frac{f(t + \Delta t) - f(t)}{\Delta t} \quad (7)$$



Difference is defined at “half integer point” in discrete time-domain

Update equations

Faraday's law

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \Rightarrow \frac{\partial \mathbf{H}}{\partial t} = -\mu^{-1} \nabla \times \mathbf{E} \quad (8)$$

Apply finite-difference formula for $\frac{\partial \mathbf{H}}{\partial t}$

$$\frac{\mathbf{H}|^{t+\frac{1}{2}} - \mathbf{H}|^{t-\frac{1}{2}}}{\Delta t} = -\mu^{-1} \nabla \times \mathbf{E}|^t \quad (9)$$

The update equation for \mathbf{H} read as

$$\mathbf{H}|^{t+\frac{1}{2}} = \mathbf{H}|^{t-\frac{1}{2}} - \Delta t \mu^{-1} \nabla \times \mathbf{E}|^t \quad (10)$$

Update equations

Ampère's law

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} \Rightarrow \frac{\partial \mathbf{E}}{\partial t} = \epsilon^{-1} \nabla \times \mathbf{H} \quad (11)$$

Apply finite-difference formula for $\frac{\partial \mathbf{E}}{\partial t}$

$$\frac{\mathbf{E}|^{t+1} - \mathbf{E}|^t}{\Delta t} = \epsilon^{-1} \nabla \times \mathbf{H}|^{t+\frac{1}{2}} \quad (12)$$

The update equation for \mathbf{E} read as

$$\mathbf{E}|^{t+1} = \mathbf{E}|^t + \Delta t \epsilon^{-1} \nabla \times \mathbf{H}|^{t+\frac{1}{2}} \quad (13)$$

Update equations for components of \mathbf{H}

Assume μ is diagonal

$$\mathbf{H}|^{t+\frac{1}{2}} = \mathbf{H}|^{t-\frac{1}{2}} - \Delta t \mu^{-1} \nabla \times \mathbf{E}|^t \quad (14)$$

$$\nabla \times \mathbf{E} = \hat{\mathbf{x}} \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) + \hat{\mathbf{y}} \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) + \hat{\mathbf{z}} \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \quad (15)$$

x-component:

$$H_x|^{t+\frac{1}{2}} = H_x|^{t-\frac{1}{2}} - \Delta t \mu_{xx}^{-1} \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \Big|_x^t \quad (16)$$

y-component:

$$H_y|^{t+\frac{1}{2}} = H_y|^{t-\frac{1}{2}} - \Delta t \mu_{yy}^{-1} \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) \Big|_y^t \quad (17)$$

z-component:

$$H_z|^{t+\frac{1}{2}} = H_z|^{t-\frac{1}{2}} - \Delta t \mu_{zz}^{-1} \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \Big|_z^t \quad (18)$$

Update equations for components of \mathbf{E}

Assume ϵ is diagonal

$$\mathbf{E}|^{t+1} = \mathbf{E}|^t + \Delta t \epsilon^{-1} \nabla \times \mathbf{H}|^{t+\frac{1}{2}} \quad (19)$$

x-component:

$$E_x|^{t+1} = E_x|^t + \Delta t \epsilon_{xx}^{-1} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \Big|^{t+\frac{1}{2}} \quad (20)$$

y-component:

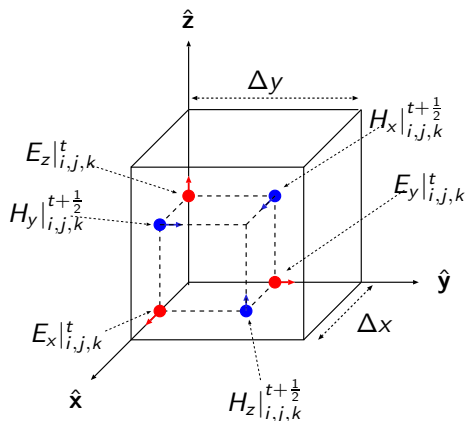
$$E_y|^{t+1} = E_y|^t + \Delta t \epsilon_{yy}^{-1} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \Big|^{t+\frac{1}{2}} \quad (21)$$

z-component:

$$E_z|^{t+1} = E_z|^t + \Delta t \epsilon_{zz}^{-1} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \Big|^{t+\frac{1}{2}} \quad (22)$$

Yee's unit cell

Components of vectors in each cell (i, j, k) are located at different points in space!



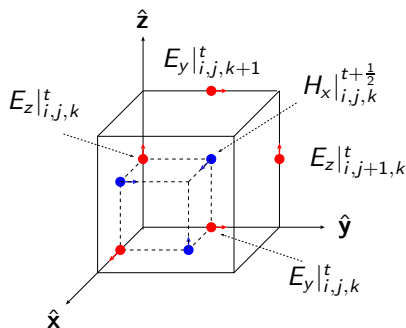
- Automatically satisfies divergence conditions
- Automatically satisfies interface conditions
- Simplifies the FDTD-algorithm

(K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Trans. Antennas and Propagation*, vol.14, no.3, pp.302–307, May 1966)

Update equation for x-component of H-field

$$H_x|_{i,j,k}^{t+\frac{1}{2}} = H_x|_{i,j,k}^{t-\frac{1}{2}} - \Delta t \mu_{xx}^{-1} \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \Big|_t \quad (23)$$

$$\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) = \frac{E_z|_{i,j+1,k}^t - E_z|_{i,j,k}^t}{\Delta y} - \frac{E_y|_{i,j,k+1}^t - E_y|_{i,j,k}^t}{\Delta z} \quad (24)$$



Update equations for H-field

y-component:

$$H_y|_{i,j,k}^{t+\frac{1}{2}} = H_y|_{i,j,k}^{t-\frac{1}{2}} - \Delta t \mu_{yy}^{-1} \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) \Big|_t \quad (25)$$

$$\left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) = \frac{E_x|_{i,j,k+1}^t - E_x|_{i,j,k}^t}{\Delta z} - \frac{E_z|_{i+1,j,k}^t - E_z|_{i,j,k}^t}{\Delta x} \quad (26)$$

z-component:

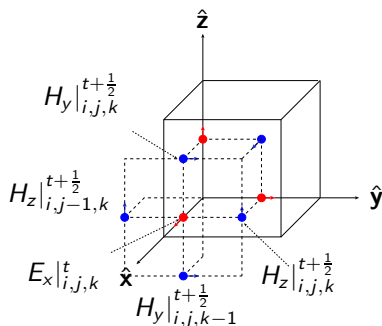
$$H_z|_{i,j,k}^{t+\frac{1}{2}} = H_z|_{i,j,k}^{t-\frac{1}{2}} - \Delta t \mu_{zz}^{-1} \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \Big|_t \quad (27)$$

$$\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) = \frac{E_y|_{i+1,j,k}^t - E_y|_{i,j,k}^t}{\Delta x} - \frac{E_x|_{i,j+1,k}^t - E_x|_{i,j,k}^t}{\Delta y} \quad (28)$$

Update equation for the x-component of E-field

$$E_x|_{i,j,k}^{t+1} = E_x|_{i,j,k}^t + \Delta t \epsilon_{xx}^{-1} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \Big|^{t+\frac{1}{2}} \quad (29)$$

$$\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) = \frac{H_z|_{i,j,k}^{t+\frac{1}{2}} - H_z|_{i,j-1,k}^{t+\frac{1}{2}}}{\Delta y} - \frac{H_y|_{i,j,k}^{t+\frac{1}{2}} - H_y|_{i,j,k-1}^{t+\frac{1}{2}}}{\Delta z} \quad (30)$$



Update equations for E-field

y-component:

$$E_y|_{i,j,k}^{t+1} = E_y|_{i,j,k}^t + \Delta t \epsilon_{yy}^{-1} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \Big|^{t+\frac{1}{2}} \quad (31)$$

$$\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) = \frac{H_x|_{i,j,k}^{t+\frac{1}{2}} - H_x|_{i,j,k-1}^{t+\frac{1}{2}}}{\Delta z} - \frac{H_z|_{i,j,k}^{t+\frac{1}{2}} - H_z|_{i-1,j,k}^{t+\frac{1}{2}}}{\Delta x} \quad (32)$$

z-component:

$$E_z|_{i,j,k}^{t+1} = E_z|_{i,j,k}^t + \Delta t \epsilon_{zz}^{-1} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \Big|^{t+\frac{1}{2}} \quad (33)$$

$$\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) = \frac{H_y|_{i,j,k}^{t+\frac{1}{2}} - H_y|_{i-1,j,k}^{t+\frac{1}{2}}}{\Delta x} - \frac{H_x|_{i,j,k}^{t+\frac{1}{2}} - H_x|_{i,j-1,k}^{t+\frac{1}{2}}}{\Delta y} \quad (34)$$

Boundary conditions

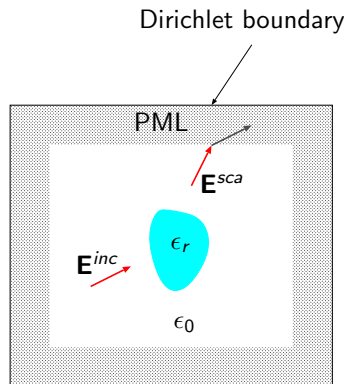
- In differential equation methods the unknowns are generally global i.e. they $\mathbf{E}, \mathbf{H} \in \mathbb{R}^3$
- We have to terminate the region of interest somehow
- We can use boundary conditions such as Dirichlet, Neumann, periodic, boundary, absorbing conditions
- Perfectly match layer (PML) for mimicking free space

Boundary conditions

- Consider FDTD-grid of size $1 : N_x, 1 : N_y, 1 : N_z$
- To calculate $\nabla \times \mathbf{H}$, we need to know values of terms $H_{i-1,j,k}$, $H_{i,j-1,k}$, and $H_{i,j,k-1}$. These, however, are not defined when $i=1$, $j=1$, $k=1$, respectively, since they are out of the computational domain.
- Similarly, $\nabla \times \mathbf{E}$ contain terms $E_{i+1,j,k}$, $E_{i,j+1,k}$, and $E_{i,j,k+1}$ which not defined when $i = N_x$, $j = N_y$, $k = N_z$.
- Force these values to be something (physically reasonable) by a boundary condition

Perfectly match layer (PML)

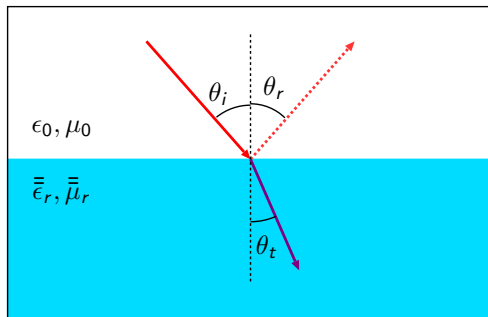
- In unbounded problems, e.g. scattering problems, we want to eliminate reflections from computational boundaries
- We introduce loss to absorb outgoing waves in the PML, and at the same time match the impedance to prevent reflections
- This has to be done for all incident angles and polarizations



Uniaxial perfectly match layer (UPML)

Consider a planewave reflection from a diagonally anisotropic interface with

$$\bar{\bar{\epsilon}}_r = \bar{\bar{\mu}}_r = \begin{pmatrix} a_2 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & c_2 \end{pmatrix} \quad (35)$$



Uniaxial perfectly match layer (UPML)

Refraction into a diagonally anisotropic material

$$\sin \theta_i = \sqrt{b_2 c_2} \sin \theta_t \quad (36)$$

Reflection coefficients (z-axis normal to the interface)

$$R^{TE} = \frac{\cos \theta_i - \sqrt{b_2/a_2} \cos \theta_t}{\cos \theta_i + \sqrt{b_2/a_2} \cos \theta_t} \quad (37)$$

$$R^{TM} = \frac{\sqrt{b_2/a_2} \cos \theta_t - \cos \theta_i}{\cos \theta_i + \sqrt{b_2/a_2} \cos \theta_t} \quad (38)$$

By choosing $\sqrt{b_2 c_2} = 1 \rightarrow \theta_i = \theta_t$ and $\sqrt{b_2/a_2} = 1$ we can see that $R^{TE} = 0$ and $R^{TM} = 0$, i.e., no reflection

We choose $b_2 = a_2$ and $c_2 = 1/b_2$

Uniaxial perfectly match layer (UPML)

By introducing losses a uniaxial material does the trick (for propagating waves, evanescence waves are more complicated but we can place the PML far enough from the scatterer)

$$\bar{\bar{\epsilon}}_r = \bar{\bar{\mu}}_r = \begin{bmatrix} S_y S_z / S_x & 0 & 0 \\ 0 & S_x S_z / S_y & 0 \\ 0 & 0 & S_x S_y / S_z \end{bmatrix} \quad (39)$$

Parameter can be chosen e.g.

$$S_x = 1 - \frac{\sigma_x(x)}{i\omega\epsilon_0}, \quad \sigma_x(x) = \frac{\epsilon_0}{2\Delta} \left(\frac{x}{L_x}\right)^3 \quad (40)$$

$$S_y = 1 - \frac{\sigma_y(y)}{i\omega\epsilon_0}, \quad \sigma_y(y) = \frac{\epsilon_0}{2\Delta} \left(\frac{y}{L_y}\right)^3 \quad (41)$$

$$S_z = 1 - \frac{\sigma_z(z)}{i\omega\epsilon_0}, \quad \sigma_z(z) = \frac{\epsilon_0}{2\Delta} \left(\frac{z}{L_z}\right)^3 \quad (42)$$

Uniaxial perfectly match layer (UPML)

UPML is a frequency domain concept hence we need convert it back to the time domain by Fourier transform

Some Fourier transform properties:

$$F(\omega) \leftrightarrow \partial f(t) \quad (43)$$

$$i\omega F(\omega) \leftrightarrow \frac{\partial f(t)}{\partial t} \quad (44)$$

$$\frac{1}{i\omega} F(\omega) \leftrightarrow \int_{-\infty}^t f(\tau) d\tau \quad (45)$$

Leads to more complicated update equations containing e.g. convolutions!

Stability of the FDTD algorithm

Maximum cell size:

$$\Delta_{max} < \frac{1}{10} \frac{\lambda}{\sqrt{\epsilon_r \mu_r}}, \quad \Delta_x, \Delta_y, \Delta_z < \Delta_{max} \quad (46)$$

and cells should be able to model geometrical details

Numerical velocity should not exceed c_0

– > Courant stability condition:

$$\Delta t < \frac{\sqrt{\epsilon_r \mu_r}}{c_0 \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}} \quad (47)$$

Dispersion relation

Consider a plane electromagnetic wave propagating along z-axis

$$\mathbf{E} = \mathbf{E}_0 e^{i(\omega t - kz)} \quad (48)$$

How fast a constant phase “point” propagates?

$$\frac{d}{dt}(\omega t - kz) = \frac{d}{dt}(\text{constant}) \quad (49)$$

Phase velocity

$$c_p = \frac{dz}{dt} = \frac{\omega}{k} \quad (50)$$

Unfortunately, the dispersion is not the same in the FDTD-grid

Numerical dispersion 1D

Planewave propagating z-direction

$$E_x|_q^t = E_0 e^{-i(\omega n \Delta t - \tilde{k}_z q \Delta z)} \quad (51)$$

$$H_y|_q^t = H_0 e^{-i(\omega n \Delta t - \tilde{k}_z q \Delta z)} \quad (52)$$

Faraday's law in free-space

$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t} \quad (53)$$

Faraday's law for the above planewave (y-component)

$$\frac{\partial E_x}{\partial z} = -\mu \frac{\partial H_y}{\partial t} \quad (54)$$

Spatial finite-difference for the planewave

$$\frac{\partial E_x}{\partial z} = \frac{(e^{i\tilde{k}_z \frac{\Delta z}{2}} - e^{-i\tilde{k}_z \frac{\Delta z}{2}})}{\Delta z} E_x|_q^t \quad (55)$$

Numerical dispersion 1D

Temporal finite-difference for the planewave

$$\frac{\partial H_y}{\partial t} = \frac{(e^{-i\omega \frac{\Delta t}{2}} - e^{i\omega \frac{\Delta t}{2}})}{\Delta t} H_y|_q^t \quad (56)$$

FD-Faraday law read as

$$\frac{(e^{i\tilde{k}_z \frac{\Delta z}{2}} - e^{-i\tilde{k}_z \frac{\Delta z}{2}})}{\Delta z} E_x|_q^t = -\mu_0 \frac{(e^{-i\omega \frac{\Delta t}{2}} - e^{i\omega \frac{\Delta t}{2}})}{\Delta t} H_y|_q^t \quad (57)$$

Since

$$\frac{E_x}{H_y} = \sqrt{\frac{\mu_0}{\epsilon_0}} \quad (58)$$

we can write the numerical dispersion relation as

$$\frac{1}{\Delta z} \sin\left(\frac{\tilde{k}_z \Delta z}{2}\right) = \frac{1}{c_p \Delta t} \sin\left(\frac{\omega \Delta t}{2}\right) \quad (59)$$

Numerical dispersion 1D

Numerical dispersion relation

$$\frac{1}{\Delta z} \sin \left(\frac{\tilde{k}_z \Delta z}{2} \right) = \frac{1}{c \Delta t} \sin \left(\frac{\omega \Delta t}{2} \right) \quad (60)$$

Case 1: $\Delta t, \Delta z \rightarrow 0$, for small x

$$\sin x = x - \frac{x^3}{3!} + \frac{5^3}{5!} + \mathcal{O}(x^7) \quad (61)$$

$$k_z = \frac{\omega}{c_p} \quad (62)$$

Case 2: set $\Delta t = \Delta z/c$

$$\frac{1}{\Delta z} \sin \left(\frac{\tilde{k}_z \Delta z}{2} \right) = \frac{1}{\Delta z} \sin \left(\frac{\omega \Delta z}{2c} \right) \rightarrow \tilde{k}_z = \frac{\omega}{c} \quad (63)$$

Magic time-step!

Numerical dispersion 3D

Three-dimensional case

$$\begin{aligned} \left[\frac{1}{c_p \Delta t} \sin \left(\frac{\omega \Delta t}{2} \right) \right]^2 &= \left[\frac{1}{\Delta x} \sin \left(\frac{\tilde{k}_x \Delta x}{2} \right) \right]^2 \\ &+ \left[\frac{1}{\Delta y} \sin \left(\frac{\tilde{k}_y \Delta y}{2} \right) \right]^2 + \left[\frac{1}{\Delta z} \sin \left(\frac{\tilde{k}_z \Delta z}{2} \right) \right]^2 \end{aligned} \quad (64)$$

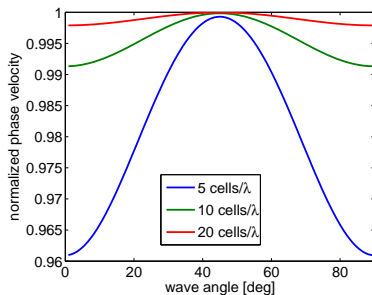
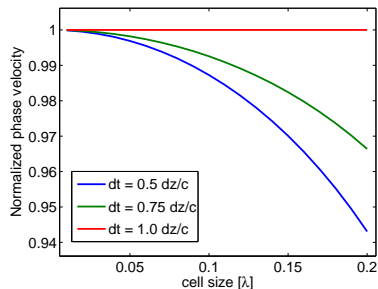
Depends on the time and spatial resolution, and propagation direction!
Numerical dispersion is anisotropic.

Approaches the continuous dispersion relation

$$\left(\frac{\omega}{c_p} \right)^2 = \tilde{k}_x^2 + \tilde{k}_y^2 + \tilde{k}_z^2 \quad (65)$$

when $\Delta t, \Delta x, \Delta y, \Delta z \rightarrow 0$, however a magic time-step does not exist!

Numerical dispersion



Numerical dispersion can be reduced by decreasing cell size

Recall: Courant stability condition – \rightarrow smaller time-step

Implementation of 1-D FDTD

1D problem

Consider uniform problem in the x and y-directions

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial y} = 0 \quad (66)$$

System decouples into two independent modes

E_x/H_y mode:

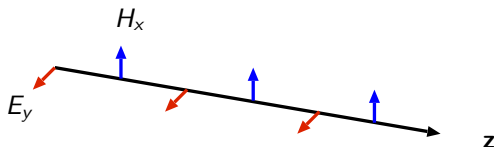
$$\frac{\partial E_x}{\partial z} = -\mu_{yy} \frac{\partial H_y}{\partial t}, \quad -\frac{\partial H_y}{\partial z} = -\epsilon_{xx} \frac{\partial E_x}{\partial t} \quad (67)$$

E_y/H_x mode:

$$\frac{\partial E_y}{\partial z} = \mu_{xx} \frac{\partial H_x}{\partial t}, \quad \frac{\partial H_x}{\partial z} = \epsilon_{yy} \frac{\partial E_y}{\partial t} \quad (68)$$

1D update equation

We consider E_y/H_x -mode



$$H_x|_k^{t+\frac{1}{2}} = H_x|_k^{t-\frac{1}{2}} + m_{H_x}^k \left(\frac{E_y|_{k+1}^t - E_y|_k^t}{\Delta z} \right) \quad (69)$$

$$E_y|_k^{t+1} = E_y|_k^t + m_{E_y}^k \left(\frac{H_x|_k^{t+\frac{1}{2}} - H_x|_{k-1}^{t+\frac{1}{2}}}{\Delta z} \right) \quad (70)$$

where the update coefficients are defined as $m_{H_x}^k = \frac{\Delta t}{\mu_{xx}^k}$ and $m_{E_y}^k = \frac{\Delta t}{\epsilon_{yy}^k}$

PEC and PMC boundary conditions

Consider 1D-FDTD grid where $k = 1 : N_k$

When $k = N_k$, the update equation for H_x read as

$$H_x|_{N_k}^{t+\frac{1}{2}} = H_x|_{N_k}^{t-\frac{1}{2}} + m_{H_x}^{N_k} \left(\frac{E_y|_{N_{k+1}}^t - E_y|_{N_k}^t}{\Delta z} \right) \quad (71)$$

and $E_y|_{N_{k+1}}^t$ exists outside the computational domain. Setting $E_y|_{N_{k+1}}^t = 0$ enforces the PEC boundary condition $\mathbf{n} \times \mathbf{E} = 0$

When $k = 1$, the update equation for E_y read as

$$E_y|_1^{t+1} = E_y|_1^t + m_{E_y}^k \left(\frac{H_x|_1^{t+\frac{1}{2}} - H_x|_0^{t+\frac{1}{2}}}{\Delta z} \right) \quad (72)$$

and $H_x|_0^{t+\frac{1}{2}}$ exists outside the computational domain. Setting $H_x|_0^{t+\frac{1}{2}} = 0$ enforces the PMC boundary condition $\mathbf{n} \times \mathbf{H} = 0$

Basic 1D FDTD engine

Algorithm 1 1D-fdtd engine with PEC and PMC

- 1: Initialize grid
 - 2: **for** $t = 1$ to N_t **do**
 - 3: **for** $k = 1$ to $N_k - 1$ **do**
 - 4: Update H_x
 - 5: **end for**
 - 6: Compute $H_x(N_k)$ with BC
 - 7: **for** $k = 2$ to N_k **do**
 - 8: Update E_y
 - 9: **end for**
 - 10: Compute $E_y(1)$ with BC
 - 11: **end for**
-

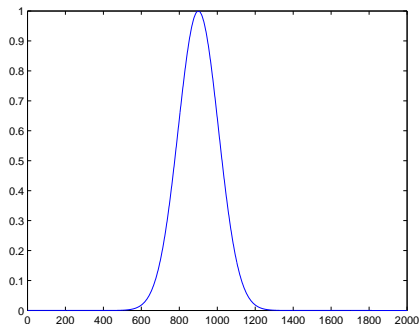
When we run this algorithm everything should stay zeros (no source included)

The Gaussian source for 1D-FDTD

The Gaussian pulse is a typical source in FDTD-simulations since it excites a broad range of frequencies.

$$g(t) = \exp \left[- \left(\frac{t - t_0}{\tau} \right)^2 \right] \quad (73)$$

t_0 is the delay and τ is the width of the pulse. Note, from the Fourier transform we can see that the bandwidth is about $B = \frac{1}{\pi\tau}$



Simple soft-source

Simple transparent Gaussian source. Add the source function to some field component at one point on the grid e.g.

$$E_y|_k^{t+1} = E_y|_k^{t+1} + g|_k \quad (74)$$

Algorithm 2 1D-fdtd engine with PEC/PMC and soft source

```

1: Compute source
2: Initialize grid
3: for t = 1 to N_t do
4:   for k = 1 to N_k - 1 do
5:     Update H_x
6:   end for
7:   Compute H_x(N_k) with BC
8:   for k = 2 to N_k do
9:     Update E_y
10:  end for
11:  Compute E_y(1) with BC
12:  Inject Source E_y|_s = E_y|_s + g|_s
13: end for

```

“two-way source” useful for testing boundary conditions

Ideal absorbing boundary condition

In 1D , it is possible find a perfectly absorbing boundary condition if

- Waves travelling only outward at the boundary
- Materials at both boundaries are linear, isotropic homogeneous and non-dispersive (same material for both boundaries)
- $\Delta t = \sqrt{\epsilon_r \mu_r} \Delta z / (2c)$

Waves propagate exactly one cell in two time-steps

$$E_y|_{N_{k+1}}^t = E_y|_{N_k}^{t-2} \quad (75)$$

$$H_x|_0^{t+\frac{1}{2}} = H_x|_1^{t-\frac{3}{2}} \quad (76)$$

Record field values at the boundaries and use recorded values as boundary conditions

Ideal absorbing boundary condition

At $k = 1$ record H-field values and modify the E-field update equation

$$\begin{aligned}
 h_3 &= h_2 \\
 h_2 &= h_1 \\
 h_1 &= H_x|_1^{t+\frac{1}{2}} \\
 E_y|_1^{t+1} &= E_y|_1^t + m_{E_y}^1 \frac{H_x|_1^{t+\frac{1}{2}} - h_3}{\Delta z}
 \end{aligned} \tag{77}$$

At $k = N_k$ record E-field values and modify the H-field update equation

$$\begin{aligned}
 e_3 &= e_2 \\
 e_2 &= e_1 \\
 e_1 &= E_y|_{N_k}^t \\
 H_x|_{N_k}^{t+\frac{1}{2}} &= H_x|_{N_k}^{t-\frac{1}{2}} + m_{H_x}^{N_k} \frac{E_y|_{N_k}^t - e_3}{\Delta z}
 \end{aligned} \tag{78}$$

Simple but works only in a special case

Ideal absorbing boundary condition

Algorithm 3 1D-fdtd engine with the ABC and soft source

```

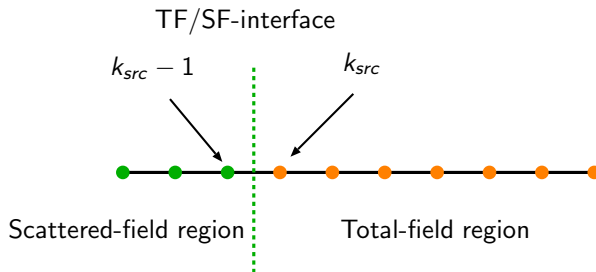
1: Compute time-step
2: Compute source
3: Initialize grid
4:
5: for  $t = 1$  to  $N_t$  do
6:
7:   for  $k = 1$  to  $N_k - 1$  do
8:     Update  $H_x$ 
9:   end for
10:  Compute  $H_x(N_k)$  with  $e_3$  BC
11:  Record H-field  $h_3 = h_2, h_2 = h_1, h_1 = H_x|_1$ 
12:
13:  for  $k = 2$  to  $N_k$  do
14:    Update  $E_y$ 
15:  end for
16:  Compute  $E_y(1)$  with  $h_3$  BC
17:  Record E-field  $e_3 = e_2, e_2 = e_1, e_1 = E_y|_{N_k}$ 
18:
19:  Inject Source  $E_y|_s = E_y|_s + g|_s$ 
20: end for

```

Total-field/scattered-field (TF/SF) source

Eliminates backward propagating waves “One way source”

Divide the grid into total-field region and scattered-field region



Total-field/scattered-field (TF/SF) source

Consider update equation for H_x at $k_{src} - 1$ (Scattered-field side)

$$H_x|_{k_{src}-1}^{t+\frac{1}{2}} = H_x|_{k_{src}-1}^{t-\frac{1}{2}} + m_{H_x}^{k_{src}-1} \left(\frac{E_y|_{k_{src}}^t - E_y|_{k_{src}-1}^t}{\Delta z} \right) \quad (79)$$

$E_y|_{k_{src}}^t$ exists at the total-field side.

Since $\mathbf{E}^{tot} = \mathbf{E}^{src} + \mathbf{E}^{sca}$, we need to subtract the source-field from the total-field to obtain the scattered-field

$$H_x|_{k_{src}-1}^{t+\frac{1}{2}} = H_x|_{k_{src}-1}^{t-\frac{1}{2}} + m_{H_x}^{k_{src}-1} \left(\frac{(E_y|_{k_{src}}^t - E_y^{src}|_{k_{src}}^t) - E_y|_{k_{src}-1}^t}{\Delta z} \right) \quad (80)$$

This can be written as

$$H_x|_{k_{src}-1}^{t+\frac{1}{2}} = H_x|_{k_{src}-1}^{t-\frac{1}{2}} + m_{H_x}^{k_{src}-1} \left(\frac{E_y|_{k_{src}}^t - E_y|_{k_{src}-1}^t}{\Delta z} \right) - \frac{m_{H_x}^{k_{src}-1}}{\Delta z} E_y^{src}|_{k_{src}}^t \quad (81)$$

i.e. the standard update equation + a correction term

Total-field/scattered-field (TF/SF) source

Consider update equation for E_y at k_{src} (Total-field side)

$$E_y|_{k_{src}}^{t+1} = E_y|_{k_{src}}^t + m_{E_y}^{k_{src}} \left(\frac{H_x|_{k_{src}}^{t+\frac{1}{2}} - H_x|_{k_{src}-1}^{t+\frac{1}{2}}}{\Delta z} \right) \quad (82)$$

$H_x|_{k_{src}-1}^{t+\frac{1}{2}}$ exists at the scattered-field side. We must add the source to it

$$E_y|_{k_{src}}^{t+1} = E_y|_{k_{src}}^t + m_{E_y}^{k_{src}} \left(\frac{H_x|_{k_{src}}^{t+\frac{1}{2}} - (H_x|_{k_{src}-1}^{t+\frac{1}{2}} + H_x^{src}|_{k_{src}-1}^{t+\frac{1}{2}})}{\Delta z} \right) \quad (83)$$

and this can be expressed as

$$E_y|_{k_{src}}^{t+1} = E_y|_{k_{src}}^t + m_{E_y}^{k_{src}} \left(\frac{H_x|_{k_{src}}^{t+\frac{1}{2}} - H_x|_{k_{src}-1}^{t+\frac{1}{2}}}{\Delta z} \right) - \frac{m_{E_y}^{k_{src}}}{\Delta z} H_x^{src}|_{k_{src}-1}^{t+\frac{1}{2}} \quad (84)$$

Total-field/scattered-field (TF/SF) source

Two source functions are needed:

$$E_y^{src}|_{k_{src}}^t \text{ and } H_x^{src}|_{k_{src}-1}^{t+\frac{1}{2}} \quad (85)$$

Note, these function exist at different locations in space and time
 → time-delay

$$E_y^{src}|_{k_{src}}^t = g(t) \quad (86)$$

$$H_x^{src}|_{k_{src}-1}^{t+\frac{1}{2}} = -\sqrt{\frac{\epsilon_{k_{src}}}{\mu_{k_{src}}}} g\left(t + \frac{\sqrt{\epsilon_r \mu_r} \Delta z}{2c} + \frac{\Delta t}{2}\right) \quad (87)$$

half grid delay

half time-step delay

Ideal absorbing boundary condition

Algorithm 4 1D-fdtd engine with the ABC and TF/SF source

```

1: Compute time-step
2: Compute sources
3: Initialize grid
4:
5: for  $t = 1$  to  $N_t$  do
6:
7:   for  $k = 1$  to  $N_k - 1$  do
8:     Update  $H_x$ 
9:   end for
10:  Compute  $H_x(N_k)$  with  $e_3$  BC
11:  Record H-field  $h_3 = h_2, h_2 = h_1, h_1 = H_x|_1$ 
12:  Inject H-field source  $H_x|_{s-1}^{t+1/2} = H_x|_{s-1}^{t+1/2} + \dots$ 
13:
14:  for  $k = 2$  to  $N_k$  do
15:    Update  $E_y$ 
16:  end for
17:  Compute  $E_y(1)$  with  $h_3$  BC
18:  Record E-field  $e_3 = e_2, e_2 = e_1, e_1 = E_y|_{N_k}$ 
19:  Inject E-field source  $E_y|_s^{t+1} = E_y|_s^{t+1} + \dots$ 
20:
21: end for

```

Fourier transforms

In many cases, we want to study scattering properties in frequency domain.

Fourier transform: conversion between time and frequency domains

$$G(f) = \int_{-\infty}^{\infty} g(t) \exp(-i2\pi ft) dt \quad (88)$$

Inverse transform

$$g(t) = \int_{-\infty}^{\infty} G(f) \exp(i2\pi tf) dx \quad (89)$$

Discrete transform:

$$G(f) = \sum_{n=1}^{N_t} g(n\Delta t) \exp(-i2\pi fn\Delta t)\Delta t \quad (90)$$

Calculation of reflectance and transmittance

Normalize the spectra by dividing the reflection and transmission spectrum by the source spectrum Reflectance:

$$R(f) = \left(\frac{DFT[E_{ref}(t)]}{DFT[E_{src}(t)]} \right)^2 \quad (91)$$

Transmittance:

$$R(f) = \left(\frac{DFT[E_{trn}(t)]}{DFT[E_{src}(t)]} \right)^2 \quad (92)$$

Note: Nyquist theorem \rightarrow maximum frequency

$$f_{max} = \frac{1}{2\Delta t} \quad (93)$$

$$\Delta f \cong \frac{1}{N_{steps}\Delta t} \quad (94)$$

Last FDTD example

Algorithm 5 1D-fdtd engine with the ABC, TF/SF source and DFT

```

1: for  $t = 1$  to  $N_t$  do
2:
3:   for  $k = 1$  to  $N_k - 1$  do
4:     Update  $H_x$ 
5:   end for
6:   Compute  $H_x(N_k)$  with  $e_3$  BC
7:   Record H-field  $h_3 = h_2, h_2 = h_1, h_1 = H_x|_1$ 
8:   Inject H-field source  $H_x|_{s-1}^{t+1/2} = H_x|_{s-1}^{t+1/2} + \dots$ 
9:
10:  for  $k = 2$  to  $N_k$  do
11:    Update  $E_y$ 
12:  end for
13:  Compute  $E_y(1)$  with  $h_3$  BC
14:  Record E-field  $e_3 = e_2, e_2 = e_1, e_1 = E_y|_{N_k}$ 
15:  Inject E-field source  $E_y|_s^{t+1} = E_y|_s^{t+1} + \dots$ 
16:
17:  for  $q = 1$  to  $N_q$  do
18:    Compute DFT  $E_{ref}(q) = E_{ref}(q) + \Delta t \exp(-i2\pi f_q t \Delta t) E_y|_2$ 
19:    Compute DFT  $E_{trn}(q) = E_{trn}(q) + \Delta t \exp(-i2\pi f_q t \Delta t) E_y|_{N_k-2}$ 
20:  end for
21: end for

```