

ECE 742 Final Project

Michelle King Suraj Suri

May 5, 2018

1 Theory

1.1 PML

Perfectly Matched Layer (PML) boundary conditions are absorbing boundary conditions. PML BCs decay the wave within a boundary layer at the edge of the simulation. The edge of the simulation BC can be implemented as PEC. Well-implemented PML BCs completely decay the wave from the time it enters the boundary layer to the time after it reflects and attempts to leave.

1.2 Finite-Difference Derivation

Let's start with equation:

$$\nabla \times \vec{H} = j\omega\epsilon\vec{E}$$

Evaluate the cross product and write in matrix form:

$$\begin{bmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = j\omega\epsilon \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

Where the values in the second rank tensor can be described by:

$$s_x = \kappa_x + \frac{\sigma_x}{j\omega\epsilon_0}$$

$$s_y = \kappa_y + \frac{\sigma_y}{j\omega\epsilon_0}$$

$$s_z = \kappa_z + \frac{\sigma_z}{j\omega\epsilon_0}$$

To make the calculation computationally more manageable, we can define the following relations:

$$D_x = \epsilon \frac{s_z}{s_x} E_x$$

$$D_y = \epsilon \frac{s_x}{s_y} E_y$$

$$D_z = \epsilon \frac{s_y}{s_z} E_z$$

such that now:

$$\begin{bmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = j\omega \begin{bmatrix} s_y & 0 & 0 \\ 0 & s_z & 0 \\ 0 & 0 & s_x \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

Using the defined values of s and that $\frac{\partial}{\partial t} = j\omega$:

$$\begin{bmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = \frac{\partial}{\partial t} \begin{bmatrix} \kappa_y & 0 & 0 \\ 0 & \kappa_z & 0 \\ 0 & 0 & \kappa_x \end{bmatrix} \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} + \frac{1}{\epsilon_0} \begin{bmatrix} \sigma_y & 0 & 0 \\ 0 & \sigma_z & 0 \\ 0 & 0 & \sigma_x \end{bmatrix} \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}$$

This has too much info. Since we are doing 2D, $\kappa_z = 1$, $\sigma_z = 0$ in this simulation. As well as $D_x = D_y = 0$ and $H_z = 0$. Equation reduces to:

$$\begin{bmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = \frac{\partial}{\partial t} \begin{bmatrix} \kappa_y & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \kappa_x \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ D_z \end{bmatrix} + \frac{1}{\epsilon_0} \begin{bmatrix} \sigma_y & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_x \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ D_z \end{bmatrix}$$

This leaves us with one equation to find an update equation for D_z :

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \kappa_x \frac{\partial}{\partial t} D_z + \frac{\sigma_x}{\epsilon_0} D_z$$

Using semi-implicit (???)

$$D^{n+1/2} = \frac{D^{n+1} + D^n}{2}$$

If we discretize around point i,j,k at timestep n

This equation discretizes to:

$$\begin{aligned} & \frac{1}{\Delta x} (H_y^{n+1/2}(i+1/2, j) - H_y^{n+1/2}(i-1/2, j)) \\ & - \frac{1}{\Delta y} (H_x^{n+1/2}(i, j+1/2) - H_x^{n+1/2}(i, j-1/2)) = \\ & \frac{\kappa_x}{\Delta t} (D_z^{n+1}(i, j) - D_z^n(i, j)) + \frac{\sigma_x}{2\epsilon_0} (D_z^{n+1}(i, j) + D_z^n(i, j)) \end{aligned}$$

We can solve this to find update equation for D:

$$\begin{aligned} & \frac{1}{\Delta_x} (H_y^{n+1/2}(i+1/2, j) - H_y^{n+1/2}(i-1/2, j)) \\ & - \frac{1}{\Delta_y} (H_x^{n+1/2}(i, j+1/2) - H_x^{n+1/2}(i, j-1/2)) = \\ & \quad \left(\frac{\sigma_x}{2\epsilon_0} + \frac{\kappa_x}{\Delta t} \right) D_z^{n+1}(i, j) + \left(\frac{\sigma_x}{2\epsilon_0} - \frac{\kappa_x}{\Delta t} \right) D_z^n(i, j) \end{aligned}$$

$$\begin{aligned} & \frac{1}{\Delta_x} (H_y^{n+1/2}(i+1/2, j) - H_y^{n+1/2}(i-1/2, j)) \\ & - \frac{1}{\Delta_y} (H_x^{n+1/2}(i, j+1/2) - H_x^{n+1/2}(i, j-1/2)) = \\ & \quad \left(\frac{\sigma_x \Delta t + 2\epsilon_0 \kappa_x}{2\epsilon_0 \Delta t} \right) D_z^{n+1}(i, j) + \left(\frac{\sigma_x \Delta t - 2\epsilon_0 \kappa_x}{2\epsilon_0 \Delta t} \right) D_z^n(i, j) \end{aligned}$$

$$\begin{aligned} D_z^{n+1}(i, j) = & - \frac{\sigma_x - 2\epsilon_0 \kappa_x}{\sigma_x \Delta t + 2\epsilon_0 \kappa_x} D_z^n(i, j) \\ & + \frac{2\epsilon_0 \Delta t}{\Delta_x (\sigma_x \Delta t + 2\epsilon_0 \kappa_x)} (H_y^{n+1/2}(i+1/2, j) - H_y^{n+1/2}(i-1/2, j)) \\ & - \frac{2\epsilon_0 \Delta t}{\Delta_y (\sigma_x \Delta t + 2\epsilon_0 \kappa_x)} (H_x^{n+1/2}(i, j+1/2) - H_x^{n+1/2}(i, j-1/2)) \end{aligned}$$

Update Components for E

Update Components for B

Update Components for H

1.3 Graded Conductivity

Reflection Factor

$$R(\theta) = \exp^{-2\eta \cos(\theta) \int_0^d \sigma_x(x) dx}$$

Where σ_x is the graded conductivity of the PML material.
 θ is the angle of incidence of the wave. So steeper angles of θ will result in higher values of reflection error.

We want to minimize reflection R but also make sure the wave decays completely in the PML boundary layer.

We are going to compare the error for different types of grading profiles. And/or we can use different values in the grading profile

1.3.1 Polynomial grading

Where the graded conductivity is:

$$\sigma_x = \left(\frac{x}{d}\right)^m \sigma_{x,max}$$

And the graded value for κ_x is:

$$\kappa_x = 1 + (\kappa_{x,max} - 1) \left(\frac{x}{d}\right)^m$$

Reflection factor simplifies to

1.3.2 Geometric grading

Where the graded conductivity is:

$$\sigma_x = (g^{\frac{1}{\Delta}})^x \sigma_{x,0}$$

$\sigma_{x,0}$ is the conductivity at the surface of the PML.
 g is a scaling factor. Nearly optimal: $2 \leq g \leq 3$
 Δ is spacing of FDTD lattice.

And the graded value for κ_x is:

$$\kappa_x = [(\kappa_{max})^{\frac{1}{\Delta}} g^{\frac{1}{\Delta}}]^x$$

2 Code

3 Error Analysis

Insert Error Analysis Here

PMLs are exact for continuous functions, but error is introduced for discrete functions. Having a large step discontinuity can

3.0.1 Error of Polynomial Grading

3.0.2 Error of Geometric Grading

4 Fix me: Bibliography

Susan's book - third edition