

1-D Implementation of Maxwell's Equations in MATLAB to Study the Effect of Absorption Using PML

¹Vikas Rathi, ²Pranav ³K. Shrivastava, ³Hemant S Pokhariya

ECE Department
Graphic Era University
Dehradun, India
Email: vikas.rth@gmail.com

Abstract: The Finite Difference Time Domain method (FDTD) uses centre-difference representations of the continuous partial differential equations to create iterative numerical models of wave propagation. First we study the propagation behavior of the wave in single dimension without PML and in second part we study the absorption using PML for the same wave using MATLAB environment.

I. INTRODUCTION

Finite-difference time-domain (FDTD) is a popular computational electrodynamics modeling technique. Since it is a time-domain method, solutions can cover a wide frequency range with a single simulation run.

The FDTD method belongs in the general class of grid-based differential time-domain numerical modeling methods. The time-dependent Maxwell's equations (in partial differential form) are discretized using central-difference approximations to the space and time partial derivatives. The resulting finite-difference equations are solved in either software or hardware in a leapfrog manner: the electric field vector components in a volume of space are solved at a given instant in time; then the magnetic field vector components in the same spatial volume are solved at the next instant in time; and the process is repeated over and over again until the desired transient or steady-state electromagnetic field behavior is fully evolved. When Maxwell's differential equations are examined, it can be seen that the change in the E-field in time (the time derivative) is dependent on the change in the H-field across space (the curl). This results in the basic FDTD time-stepping relation that, at any point in space, the updated value of the E-field in time is dependent on the stored value of the E-field and the numerical curl of the local distribution of the H-field in space.^[1] The H-field is time-stepped in a similar manner. At any point in space, the updated value of the H-field in time is dependent on the stored

value of the H-field and the numerical curl of the local distribution of the E-field in space. The 3D source free (J=0) Maxwell's curl equations of a homogeneous medium are:

$$\nabla \times \vec{E} = -\mu \frac{d\vec{H}}{dt} \rightarrow \begin{cases} \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t} \\ \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t} \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t} \end{cases}$$

$$\nabla \times \vec{H} = \epsilon \frac{d\vec{E}}{dt} \rightarrow \begin{cases} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \epsilon \frac{\partial E_x}{\partial t} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \epsilon \frac{\partial E_y}{\partial t} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \epsilon \frac{\partial E_z}{\partial t} \end{cases}$$

$$\frac{\partial E_x}{\partial z} = -\mu \frac{\partial H_y}{\partial t}$$

$$\frac{\partial H_y}{\partial z} = -\epsilon \frac{\partial E_x}{\partial t}$$

II. FDTD TREATMENT TO MAXWELL'S EQUATIONS IN MATLAB

To create arrays for E field and H field, a spatial width of say 100 points and temporal width of 2 for current and previous values

```
S_Width=100; T_Width=2;
e=zeros(S_Width,T_Width);
h=zeros(S_Width,T_Width);
```

If our model is to represent a waveguide of a given length (say 2m) we can evaluate the spatial resolution with

```
length=2;
dx=length/S_Width;
```

Hence we can determine the temporal resolution using the Courant limit

$C = \frac{u \Delta t}{\Delta x}$, which is 1 for 1D case. Hence type

```
c=23;
dt=dx/c;
```

Courant–Friedrichs–Lewy(CFL) condition

In mathematics, the Courant–Friedrichs–Lewy condition (CFL condition) is a necessary condition for convergence while solving certain partial differential equations (usually hyperbolic PDEs) numerically. (It is not in general a sufficient condition.) It arises when explicit time-marching schemes are used for the numerical solution. As a consequence, the timestep must be less than a certain time in many explicit time-marching computer simulations, otherwise the simulation will produce wildly incorrect results. The condition is named after Richard Courant, Kurt Friedrichs, and Hans Lewy who described it in their 1928 paper.

For example, if a wave is crossing a discrete grid, then the timestep must be less than the time for the wave to travel adjacent grid points. As a corollary, when the grid point separation is reduced, the upper limit for the time step also decreases. In essence, the numerical domain of dependence must include the analytical domain of dependence in order to assure that the scheme can access the information required to form the solution.

The CFL condition is commonly prescribed for those terms in PDEs which represent advection (hyperbolic part of the PDE). For one-dimensional case, the CFL condition is given by

$$C = \Delta T u / \Delta x;$$

Where u is the velocity (L/T)
 ΔT is the time step (T)
 Δx is the length interval (L),

and the constant C depends on the particular equation to be solved and not on Δt and Δx . The number is called the Courant number.

In the two-dimensional case this becomes

The CFL condition can be a very limiting constraint on the time step Δt , to the extent that for certain fourth-order nonlinear partial differential equations it can be of the form

and efforts are often made to avoid it by using implicit methods

We also need to set permittivity and permeability
 $\epsilon_0 = 8.8541878 \times 10^{-12}$

```
mu0 = 4e-7 * pi
and Q=1/mu0
```

Let's say we want to run the simulation for five seconds then we will need declare and set 'duration' and 'iterations' variables with

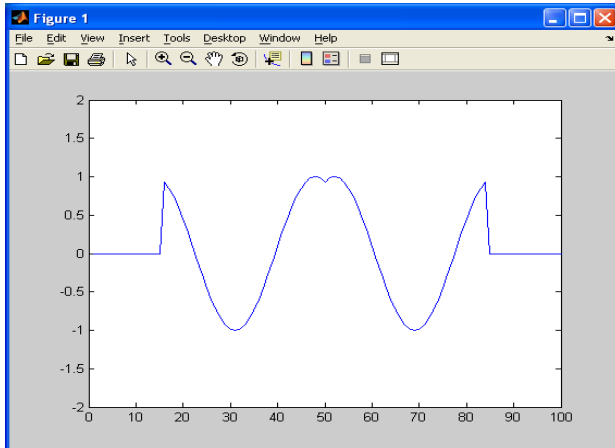
```
duration=5;
iterations=duration/dt;
```

Let's decide on an excitation point half way along the waveguide with

```
excitationPoint=S_Width/2;
```

Hence given the necessary parameters we can construct a loop that iterates across the waveguide to set pressures and velocities. This loop is then nested within another loop that iterates for successive time steps. Add to your matlab source code the following

```
for
n=2:
iterat
ions
t=n*
dt;
for i=2:
S_Width
-1
e(i,1)=e(i,2)-(1/eps0)*dt/dx*(h(i+1,2)-
h(i,2));
if i==excitatiopoint
p(i,1)=cos(2*pi*500*t); sigma=0.0005;
t0=3*sigma;
fs=exp( - ((t-t0)/0.0005)^2 ); e(i,1)=
e(i,1)+fs+fsprev; fsprev=fs;
end
h(i,1)=h(i,2)-(1/mu0)*dt/dx*(e(i,1)-e(i-
1,1));
e(i,2)=e(i,1);
h(i,2)=h(i,1);
end
plot(e(:,
2))
axis([0 S_Width -2 2]);
frame =
getframe();
end
```



III. GRID TRUNCATION

The most commonly used grid truncation techniques for open-region FDTD modeling problems are the Mur absorbing boundary condition (ABC), the Liao ABC, and various perfectly matched layer (PML) formulations. The PML (which is technically an absorbing region rather than a boundary condition) can provide orders-of-magnitude lower reflections. The PML concept was introduced by J.-P. Berenger in a seminal 1994 paper in the Journal of Computational Physics. Since 1994, Berenger's original split-field implementation has been modified and extended to the uniaxial PML (UPML), the convolutional PML (CPML), and the higher-order PML. The latter two PML formulations have increased ability to absorb evanescent waves, and therefore can in principle be placed closer to a simulated scattering or radiating structure than Berenger's original formulation.

The PML, as a lossy medium, is characterized by an electrical conductivity σ and a magnetic conductivity σ^* . the conductivities are related as

$$\frac{\sigma}{\epsilon_1} = \frac{\sigma^*}{\mu_1}$$

This relationship ensures a required level of attenuation and forces the wave impedance of the PML to be equal to that the free space. Thus a reflectionless transmission of a plane wave propagation across the interface is permitted. For oblique incidence it is required the Hz component must be split into two subcomponents., H_{zx} and H_{zy}. It leads to four components E_x, E_y, H_{zx} and H_{zy} and

four (rather than the usual three) coupled field equations:

$$\begin{aligned} \epsilon_0 \frac{\partial E_x}{\partial t} + \sigma_y E_x &= \frac{\partial (H_{zx} + H_{zy})}{\partial y} \\ \epsilon_0 \frac{\partial E_y}{\partial t} + \sigma_x E_y &= - \frac{\partial (H_{zx} + H_{zy})}{\partial x} \\ \mu_0 \frac{\partial H_{zx}}{\partial t} + \sigma_x^* E_x &= - \frac{\partial E_y}{\partial x} \\ \mu_0 \frac{\partial H_{zy}}{\partial t} + \sigma_y^* H_{zy} &= - \frac{\partial E_x}{\partial y} \end{aligned}$$

Implementing Perfectly Matched Layers

When a wave reaches the ends of the waveguide it is reflected back into the medium. This is a problem for most simulations that can be solved with the implementation of perfectly match layers (PMLs). Originally developed by Berenger, the technique specifies a new region that surrounds the FDTD domain where a set of non physical equations are applied giving a high attenuation. Here we used our 1D case equations.

Edit the previous matlab file to include a PML to the right hand side of the simulation. Key changes are in bold

```
S_Width=100; T_Width=2;
e=zeros(S_Width+1,T_Width);
h=zeros(S_Width+1,T_Width); length=2;
dx=length/S_Width;
c=22.5;
dt=dx/c;
eps0 = 8.8541878e-12
mu0 = 4e-7 * pi
duration=5;
iterations=duration/dt;
excitationPoint=S_Width/2;
xPML=10;
```

```
for
n=
2:

t=n
*dt
;
for i=2: S_Width-1

if i>(S_Width-xPML)
```

```

xi=xPML-(S_Width-i)
a0=log(10)/(K*dt);
a1=a0*(xi/xPML)^2;
a2=a0*((xi-1/2)/xPML)^2;
e(i,1)=exp(-a1*Q*dt)*e(i,2)-(1-
exp(-
(a1*Q*dt))/(a1*Q)*1/dx*(h(i+1,2)-
h(i,2)));
h(i,1)=exp(-a2*Q*dt)*h(i,2)-(1-exp(-
(a2*Q*dt))/(a2*Q)*(1/mu0)*1/dx*(e(i,1)-e(i-1,1)));
else
e(i,1)=e(i,2)-(1/eps0)*dt/dx*(h(i+1,2)-h(i,2));
if i==excitationPoint
p(i,1)=cos(2*pi*500*t);
sigma=0.0005;
t0=3*sigma;
fs=exp(-((t-t0)/0.0005)^2);
e(i,1)=e(i,1)+fs+fsprev;
fsprev=fs;
end
h(i,1)=h(i,2)-(1/mu0)*dt/dx*(e(i,1)-e(i-1,1));
e(i,2)=e(i,1);
h(i,2)=h(i,1);
end
plot(e(:,2))
axis([0 S_Width -2 2]);
frame = getframe();
end;

```

Note two absorption coefficients a_1 and a_2 are created to take account of E field and H field being $\frac{1}{2}$ element apart in the staggered grid.

IV. SIMULATION RESULTS

The output of the above program using PML concept is shown as a sequence of following three figures:

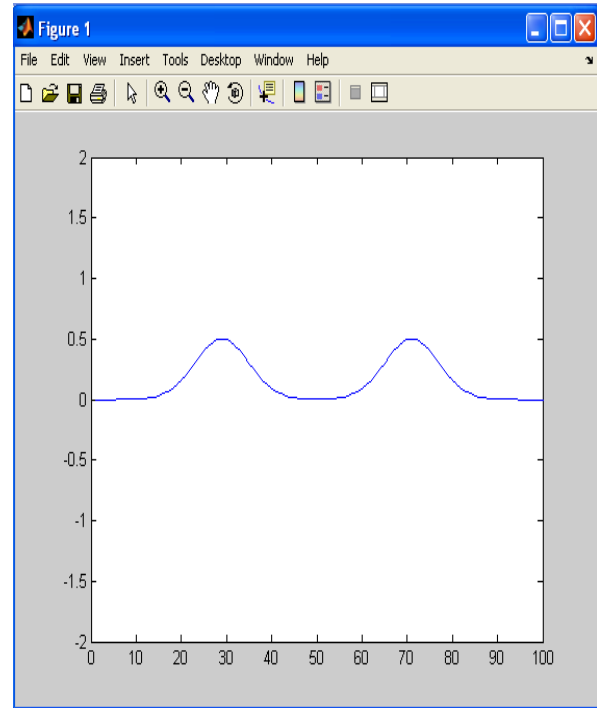


Fig 3(a): Excitation of the Wave Propagation at half points

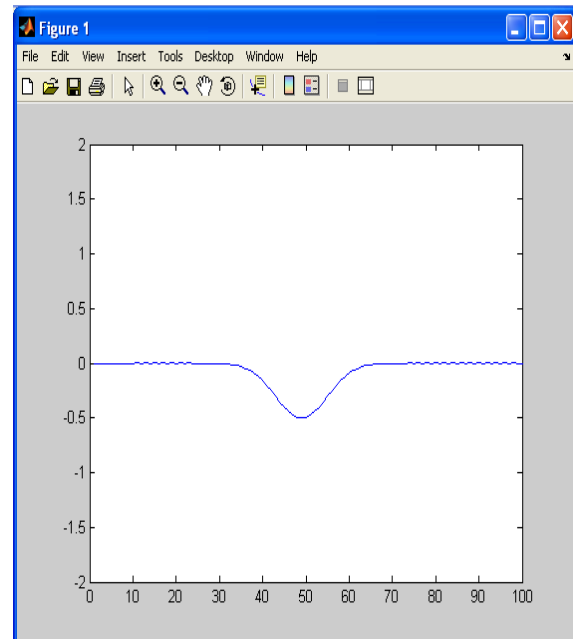


Fig 3(b) Continue wave propagation after first strike with wall

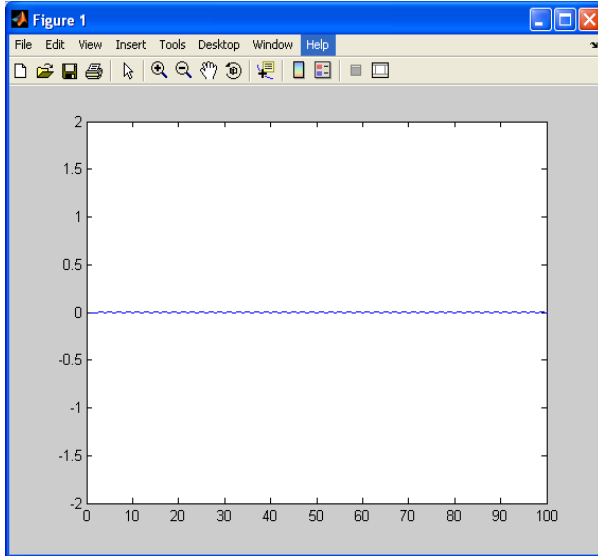


Fig 3(c) Reflection less transmission at PML

V. CONCLUSION

The output of the above PML program is a continue propagating wave which having no reflection from the absorptive boundaries,. Here three consecutive state of the wave just before and after first striking with the boundary are . This wave is starting with hard Gaussian source at the middle i.e. half point as an excitation point (depicting in Fig 3(a)).

REFERENCES

- [1] Berekger, Jean-Pierre, *A Perfectly Matched layer for the Absorption of Electromagnetic Waves*, July 2,1993, Centre d'Analyse de defense, 16 bis, Avenue prieur de la cote d'or,9411 Aruceil, France
- [2] Sadiku, Matthew N O, *Numerical Techniques in Electromagnetic*, 2nd ed. CRC Press NY, 2001, 3.8.1 and 3.9
- [3]. David B. Davidson *Computational Electromagnetic for RF and Microwave Engineering*, , Cambridge UniversityPress 2005, pp 94-105 *difference time domain method*, 2nd ed., Norwood, MA. Artech House, 2000, pp. 197-201
- [4]. Anders Bondeson, Thomas Rylander Par Ingelstrom, *Computational Electromagnetic* Springer, June 27, 2005, pp57-68

- [5] Loui, Hung, Student Member, IEEE, *1D-FDTD using MATLAB*, ECEN-6006 NUMERICAL METHODS IN PHOTONICS PROJECT-4, September 2004

