Simulation of the Finite Difference Time Domain in Two Dimension

Akram G., Jasmy Y.

Abstract—The finite-difference time-domain (FDTD) method is one of the most widely used computational methods in electromagnetic. This paper describes the design of two-dimensional (2D) FDTD simulation software for transverse magnetic (TM) polarization using Berenger's split-field perfectly matched layer (PML) formulation. The software is developed using Matlab programming language. Numerical examples validate the software.

Keywords—Finite difference time domain (FDTD) method, perfectly matched layer (PML), split-filed formulation, transverse magnetic (TM) polarization.

I. INTRODUCTION

The finite-difference time-domain (FDTD) method is a flexible and powerful solution tool for solving Maxwell's equations [1-5]. Further, the efficient and accurate solution of electromagnetic wave interaction problems in unbounded regions is one of the greatest challenges of the FDTD method. For such problems, an absorbing boundary condition (ABC) must be introduced at the outer grid boundary to simulate the extension of the grid to infinity.

Recently, among a few important studies on ABC was J.P. Berenger's Perfectly Matched Layer (PML) method [6]. In the PML, he created an artificial medium with magnetic conductivity (σ*) that will make the boundary condition to work as wave absorber region. This paper describes the design of a two-dimensional (2D) FDTD simulation software for transverse magnetic (TM) polarized incident wave using Berenger's split-field perfectly matched layer (PML) formulation. The software is developed using Matlab programming language.

II. TWO-DIMENSIONAL TM POLARIZATION

The Maxwell's curl equations as modified by Berenger are expressed in their time-dependent form as,

$$\mu \frac{\partial H_y}{\partial t} + \sigma_*^* H_y = -\frac{\partial (E_{cx} + E_{cz})}{\partial y}$$

(1)

In order to solve these equations numerically, they are discretized according to central differencing techniques. The central difference approximation is used in this case. For example, in equations (1) and (3), central differencing results for homogeneous media against the analytical solution in the following equations.

$$H_{i+1/2, j}^{n+1} = e^{j \omega (t+n \cdot \delta t)} H_{i+1/2, j}^n + \frac{1}{\sigma_*^* (i+1/2, j) \delta x} [E_{cx}^{n+1/2} (i+1/2, j+1/2) + E_{cx}^{n-1/2} (i+1/2, j-1/2) - E_{cx}^{n+1/2} (i+1/2, j-1/2) - E_{cx}^{n-1/2} (i+1/2, j+1/2)]$$

(5)

$$E_{cz}^{n+1/2} (i+1/2, j+1/2) = e^{-j \omega (t+n \cdot \delta t) / \delta y} E_{cz}^{n+1/2} (i+1/2, j+1/2)$$

$$= \frac{1}{\sigma_*^* (i+1/2, j+1/2) \delta x} [H_y^* (i, j+1/2) - H_y^* (i+1, j+1/2)]$$

(6)

Based on the above formulation, a 2-D FDTD code has been written in Matlab programming language to simulate the fields of a plane wave source in lossy media. The formulation and the FDTD code are validated by checking the numerical results for homogeneous media against the analytical solution. The code for 2D FDTD is shown in Appendix.

In order to test and validate the FDTD code, an electromagnetic wave source interacting with dielectric cylinder (ε = 4.0, σ = 0.12) of radius 6 cm is simulated in this section, shown in Fig. 1. A 1 mW/cm² plane wave is used as source, and the frequency is setup to 2.5 GHz. The reason for using dielectric cylinder is there exist an analytical solution to this problem [7]. A y-directed plane wave source:

$$e^{y(\cdot \cdot jebc+2)} = e^{y(\cdot \cdot jebc+2)} + 61.4 \sin(\omega \cdot n \cdot \delta t)$$

is used to generate the incident wave at \(j = jebc + 2\) and \(n = 1\). Where jebc is thickness of the PML region, \(n\) is time step.
in FDTD code and \( dt \) is 5.0 psec. Fig. 2 shows a comparison of FDTD calculated results with analytic solution along the center axis of the cylinder. The simulation was made with a code attached in Appendix.

![Back PML](image1)

**Fig. 1** Schematic representation of the FDTD computational domain

![Grid Position, I](image2)

**Fig. 2** Waveform comparison of the FDTD results (with different PML thicknesses) and the analytical solution for a plane wave source

### III. Procedure for Paper Submission

In this section, various numerical simulations are performed to illustrate the effectiveness of PML code [8]. Fig. 3 shows the two FDTD computational domains used in the experiments: a test domain, \( \Omega_T \), and the much larger benchmark domain \( \Omega_B \). By calculating the difference between the FDTD solutions in the two domains at each grid-point at each time-step, a measure of the spurious reflection caused by the test ABC is obtained. Initially, a 2-D test domain 100×50 cells long was examined. A square cell was used, with \( \delta = 0.015 \) m, and the time step was chosen based on the stability criterion \( \delta t = \delta^2 / 2c \), where \( c \) is the speed of light in vacuum.

In all the cases examined here, the excitation was a pulse exhibiting a very smooth transition to zero, as used in [8], and defined as follows,

\[
E_{z,T}^{1}\left[ n \right] = \begin{cases} 
10 - 15 \cos \left( \pi n / 20 \right) + 6\cos \left( 2\pi n / 20 \right) - \cos \left( 3\pi n / 20 \right) & n \leq 40 \\
0 & n > 40 
\end{cases} 
\]

(7)

Then, the local error of the computed field in \( \Omega_T \), due to reflections from the PML, is obtained by subtracting the field at any point within \( \Omega_T \) at a given time step from the field at the corresponding point in \( \Omega_B \). Here, \( E_i \) is used to define the error

\[
e_{\text{local}} \left[ n \right] = E_{z,T}^{\text{B}} \left[ n \right] - E_{z,T}^{\text{B}} \left[ n \right]
\]

(8)

where \( E_{z,T} \) and \( E_{z,B} \) are, respectively, the FDTD computed E-fields within the test and benchmark domains. Further, the global error was defined as

\[
e_{\text{global}} = \sum \sum |E_{z,T}^{\text{B}} \left[ n \right] - E_{z,B}^{\text{B}} \left[ n \right]|^2
\]

(9)

In Fig. 4 and 5, different numbers of PML are used outside the FDTD simulation region to test the efficiency of the PML ABC for the lossy media. The accuracy of the PML ABC is compared with that of standard second-order Mur ABC [9]. The numbers of PML used are 16, 8 and 4, respectively. As is evident from the result, 16 PML are good enough to absorb most of the incident waves at the boundaries.

### IV. Conclusion

The FDTD method is widely used because it is simple to implement numerically. It provides a flexible means for directly solving Maxwell’s time-dependent curl equations by using finite differences to discretize them. This paper describes the design of two-dimensional (2D) FDTD simulation software for transverse magnetic (TM) polarized incident wave using Berenger's split-field perfectly matched layer (PML) formulation. The software is developed using Matlab programming language. The numerical results agree very well with the analytical results for homogeneous media.
Fig. 4 Global error within a 100x50 cell two-dimensional TM test grid

Fig. 5 Comparative local error measures for Mur and PML ABCs along the test grid outer boundary at time-step n = 100.

APPENDIX

% Fundamental constants
cc = 2.99792458e8;            %speed of light in free space
muz = 4.0*pi*1.0e-7;         %permeability of free space
epsz = 1.0/(cc*cc*muz);    %permittivity of free space
freq = 2.5e+9;                  %frequency of source excitation
omega = 2.0*pi*freq;

% Grid parameters
ie = 50;           %number of grid cells in x-direction
je = 50;            %number of grid cells in y-direction
iebc = 8;           %thickness of left and right PML region
jebc = 8;           %thickness of front and back PML region
nmax = 500;        %total number of time steps
iebc = ie + 2*iebc;
jebc = je + 2*jebc;
ib = ie + 1;
jb = je + 1;
dx = 3.0e-3;        %space increment of square lattice
dt = dx/(2.0*cc);   %time step

% Material parameters
media = 2;
eps = [1.0 4.0];
sig = [0.0 0.012];
mur = [1.0 1.0];
sim = [0.0 0.0];

% Field arrays
ez = zeros(ibec, jebc);
ez = zeros(ibbc, jbbc);
ezy = zeros(ibec, jebc);
ezy = zeros(ibbc, jbbc);
hx = zeros(iebc, jebc);
hx = zeros(ibbc, jbbc);
hy = zeros(iebc, jebc);
hy = zeros(ibbc, jbbc);

% Updating coefficients
eaf = dt*sig(1)/(2.0*epsz*eps(1));
cai(1) = (1.0-eaf)/(1.0+eaf);
che(1) = dt*eps(1)/dx/(1.0+eaf);
haf = dt*sim(1)/(2.0*muz*mur(1));
dai(1) = (1.0-haf)/(1.0+haf);
dbi(1) = dt/muz/mur(1)/dx/(1.0+haf);
end

% Geometry specification (main grid)

caez = zeros(ibfbc, jbfbc);           %fields in main grid
caezx = zeros(ibfbc, jbfbc); 
caez = zeros(ibbc, jbbc);
caezx = zeros(ibbc, jbbc);

diam = 40;          % diameter of cylinder: 6 cm
rad = diam/2.0;     % radius of cylinder: 3 cm
ic = iefbc/2;   % i-coordinate of cylinder's center
jc = jefbc/2;     % j-coordinate of cylinder's center
for i = iebc+1:ie+iebc-1
    for j = jebc+1:je+jebc-1
        dist2 = (i-ic)^2 + (j-jc)^2;
        if dist2 <= rad^2
            caez(i,j) = ca(2);
            che(i,j) = cb(2);
            caezx(i,j) = ca(2);
            chez(i,j) = cb(2);
            caexz(i,j) = ca(2);
            chez(i,j) = cb(2);
        end
        end
    end
    end

% Add dielectric cylinder
for i = ie+1:ie+iebc-1
    for j = je+1:je+jebc-1
        disi2 = (i-ic)^2 + (j-je+1)^2;
        if disi2 <= rad^2
            caexz(i,j) = ca(2);
            chez(i,j) = cb(2);
            caezx(i,j) = ca(2);
            chez(i,j) = cb(2);
        end
        end
end
end

% Fill the PML regions
delbc = iebc*dx;
sigmam = log(rmax/100.0)/epsz*cc*(orderbc+1)/(2*delbc);
bcfactor = eps(1)*sigmam/(dx*(delbc*orderbc) * ... (orderbc+1));

% FRONT region
caez(1:ibbc, 1) = 1.0;
chez(1:ibbc, 1) = 0.0;
for j = 2:jebc
    y1 = (jebc-j+1.5)*dx;
    y2 = (jebc-j+0.5)*dx;
    sigmay = bcfactor*(y1^(orderbc+1)-y2^(orderbc+1));
    ca1 = exp(-sigmay*dt/(epsz*eps(1)));
    cb1 = (1.0-ca1)/(sigmay*dx);
    caezy(1:ibfbc,j) = ca1;
    cbezy(1:ibfbc,j) = cb1;
end

for j = 1:jebc
    y1 = (jebc-j+1)*dx;
    y2 = (jebc-j)*dx;
    sigmay = bcfactor*(y1^(orderbc+1)-y2^(orderbc+1));
    ca1 = exp(-sigmay*dt/(epsz*eps(1)));
    cb1 = (1-ca1)/(sigmay*dx);
    caezy(1:ibfbc,j) = ca1;
    cbezy(1:ibfbc,j) = cb1;
end

for i = 1:iebc
    x1 = i*dx;
    x2 = (i-1)*dx;
    sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
    da1 = exp(-sigmaxs*dt/muz);
    db1 = (1-da1)/(sigmaxs*dx);
    dahyx(i,1:iefbc) = da1;
    dbhyx(i,1:iefbc) = db1;
end

% BACK region
caezy(1:ibfbc,jbfbc) = 1.0;
cbezy(1:ibfbc,jbfbc) = 0.0;
for i = 1:iebc
    x1 = (iebc-i+1.5)*dx;
    x2 = (iebc-i+0.5)*dx;
    sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
    ca1 = exp(-sigmaxs*dt/muz);
    cb1 = (1-ca1)/(sigmaxs*dx);
    caezx(i,1:jbfbc) = ca1;
    cbezx(i,1:jbfbc) = cb1;
end

% LEFT region
caezx(1,1:jbfbc) = 1.0;
cbezx(1,1:jbfbc) = 0.0;
for i = 1:iebc
    x1 = (iebc-i+1)*dx;
    x2 = (iebc-i)*dx;
    sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
    da1 = exp(-sigmaxs*dt/muz);
    db1 = (1-da1)/(sigmaxs*dx);
    dahyx(i,1:iefbc) = da1;
    dbhyx(i,1:iefbc) = db1;
end

% RIGHT region
caezx(ibfbc,1:jbfbc) = 1.0;
cbezx(ibfbc,1:jbfbc) = 0.0;
for i = 1:iebc
    x1 = i*dx;
    x2 = (i-1)*dx;
    sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
    da1 = exp(-sigmaxs*dt/muz);
    db1 = (1-da1)/(sigmaxs*dx);
    dahyx(i,1:iefbc) = da1;
    dbhyx(i,1:iefbc) = db1;
end

% BEGIN TIME-STEPPING LOOP
for n = 1:nmax
    % Update electric field EZ in main grid
    ezy(:,jebc+2) = ezy(:,jebc+2) + 61.4*sin(omega*n*dt);
    ezx(2:iefbc,2:jefbc) = caezx(2:iefbc,2:jefbc).*ezx(2:iefbc,2:jefbc) + ...
                         cbezx(2:iefbc,2:jefbc).*(hyx(2:iefbc,2:jefbc)-...
                         hyx(1:iefbc-1,2:jefbc));
    ezy(2:iefbc,2:jefbc) = caezy(2:iefbc,2:jefbc).*ezy(2:iefbc,2:jefbc) + ...
                         cbezy(2:iefbc,2:jefbc).*(hxy(2:iefbc,1:jefbc-1)-...
                         hxy(2:iefbc,2:jefbc));
    ez = ezx + ezy;
    % Update magnetic fields (Hx and Hy) in main grid
    hxy(:,:) = dahyx(:,:).*hxy(:,:)+dbhyx(:,:)*...
               (ezx(1:iefbc,2:jefbc) + ezy(1:iefbc,2:jefbc) -...
               ezx(1:iefbc,1:jefbc) - ezy(1:iefbc,1:jefbc));
end

% END TIME-STEPPING LOOP
REFERENCES


G. Akram (M’06) Gasmolseed Akram received his B.Sc. degree in Electrical Engineering and Informatics – major in Computer Engineering – and M.Sc degree in Electrical Engineering and Informatics – major in Biomedical Engineering – from Budapest University of Technology and Economics (BME), Budapest, Hungary, in 1993 and 1999, respectively. He received the PhD degree in Electrical Engineering – major in Biomedical Engineering – from Universiti Teknologi Malaysia (UTM), Malaysia, in 2009. He is the founder and chair of IEEE EMBS Society. His biography has listed in the 25th Silver Anniversary Edition of Who’s Who in the World 2009 and 2011.

Y. Jasmy is a Professor at the Faculty of Health Science & Biomedical Engineering, Universiti Teknologi Malaysia (UTM). He obtained his MSc and PhD in Electronics from the University of Kent at Canterbury, UK. He has been involved in developing the Medical Electronics area in the faculty, heading the computer aided rehabilitation engineering group. His research is in speech therapy, electromagnetic effects in hospital environment, and computers as aids for the handicapped.