World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:3, 2012

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,

$$\mu\frac{\partial H_x}{\partial t}+\sigma_y^* H_x=-\frac{\partial\left(E_{zx}+E_{zy}\right)}{\partial y} \tag{1}$$

$$\mu\frac{\partial H_y}{\partial t}+\sigma_x^* H_y=\frac{\partial\left(E_{zx}+E_{zy}\right)}{\partial x} \tag{2}$$

$$\varepsilon\frac{\partial E_{zx}}{\partial t}+\sigma_x E_{zx}=\frac{\partial H_y}{\partial x} \tag{3}$$

$$\varepsilon\frac{\partial E_{zy}}{\partial t}+\sigma_y E_{zy}=-\frac{\partial H_x}{\partial y} \tag{4}$$

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

$$\begin{aligned}&H_x^{n+1}\left(i+1/2,j\right)=e^{-\sigma_y^*(i+1/2,j)\delta t/\mu}H_x^n\left(i+1/2,j\right)\\&-\frac{\left(1-e^{-\sigma_y^*(i+1/2,j)\delta t/\mu}\right)}{\sigma_y^*\left(i+1/2,j\right)\delta x}\begin{bmatrix}E_{zx}^{n+1/2}\left(i+1/2,j+1/2\right)+E_{zy}^{n+1/2}\left(i+1/2,j+1/2\right)\\-E_{zx}^{n+1/2}\left(i+1/2,j-1/2\right)-E_{zy}^{n+1/2}\left(i+1/2,j-1/2\right)\end{bmatrix}\end{aligned} \tag{5}$$

$$\begin{aligned}&E_{zx}^{n+1/2}\left(i+1/2,j+1/2\right)=e^{-\sigma_x^*(i+1/2,j+1/2)\delta t/\varepsilon}E_{zx}^{n-1/2}\left(i+1/2,j+1/2\right)\\&-\frac{\left(1-e^{-\sigma_x^*(i+1/2,j+1/2)\delta t/\varepsilon}\right)}{\sigma_x^*\left(i+1/2,j+1/2\right)\delta x}\left[H_y^n\left(i,j+1/2\right)-H_y^n\left(i+1,j+1/2\right)\right]\end{aligned} \tag{6}$$

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 ($\varepsilon_r = 4.0$, $\sigma = 0.12$) of radius 6 cm is simulated in this section, shown in Fig. 1. A 1 mW/cm^2 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 :

*ezy(:,jebc+2) = ezy(:,jebc+2) + 61.4*sin(omega*n*dt)*

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

G. Akram is with the Faculty of Health Science & Biomedical Eng., , Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia (phone: 607-55-35785; fax: 607-55-66272; e-mail: akram@ fke.utm.my).

Y. Jasmy is with the Faculty of Health Science & Biomedical Eng., Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia.

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 o the cylinder. The simulation was made with a code attached in Appendix.

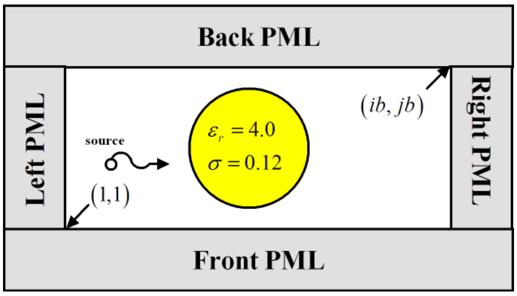

Fig. 1 Schematic representation of the FDTD computational domain

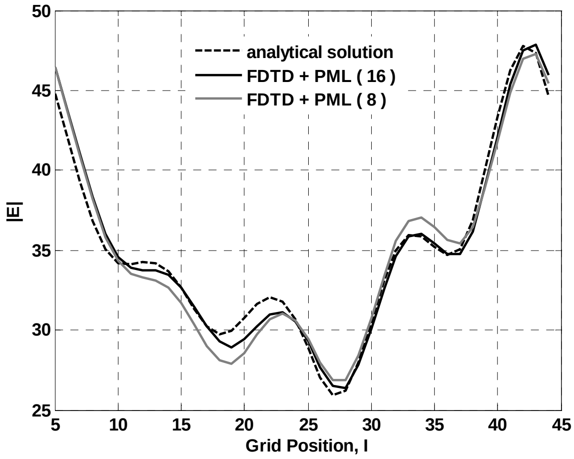

Fig. 2 Waveform comparison of the FDTD results (with different PML thicknesses) and the analytical solution for a plane wave source

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, Ω_T, and the much larger benchmark domain Ω_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 δ = 0.015 m, and the time step was chosen based on the stability criterion $\delta t = \delta / 2c$, 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,

$$E_{z,T}\Big|_{50,25}^{n} = \begin{cases} \dfrac{1}{32}\left[\begin{array}{l} 10 - 15\cos(\pi n/20) + \\ 6\cos(2\pi n/20) - \cos(3\pi n/20) \end{array}\right] & n \le 40 \\ 0 & n > 40 \end{cases} \quad (7)$$

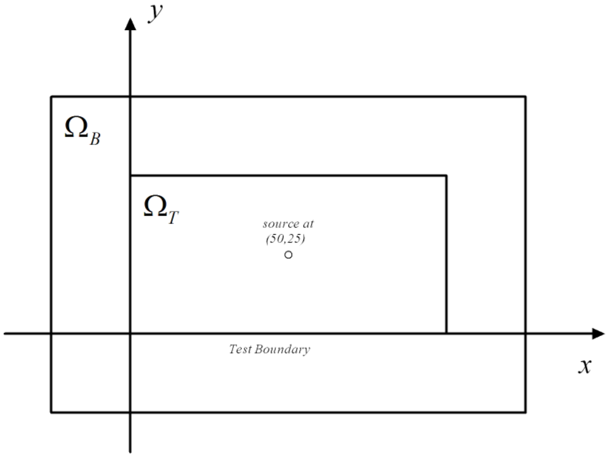

Fig. 3 Test and benchmark computational domains

Then, the local error of the computed field in Ω_T, due to reflections from the PML, is obtained by subtracting the field at any point within Ω_T at a given time step from the field at the corresponding point in Ω_B. Here, E_z is used to define the error

$$e_{local}\Big|_{i,j}^{n} = E_{z,T}\Big|_{i,j}^{n} - E_{z,B}\Big|_{i,j}^{n} \quad (8)$$

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

$$e_{global}\Big|^{n} = \sum_i \sum_j \left| E_{z,T}\Big|_{i,j}^{n} - E_{z,B}\Big|_{i,j}^{n} \right|^2 \quad (9)$$

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.

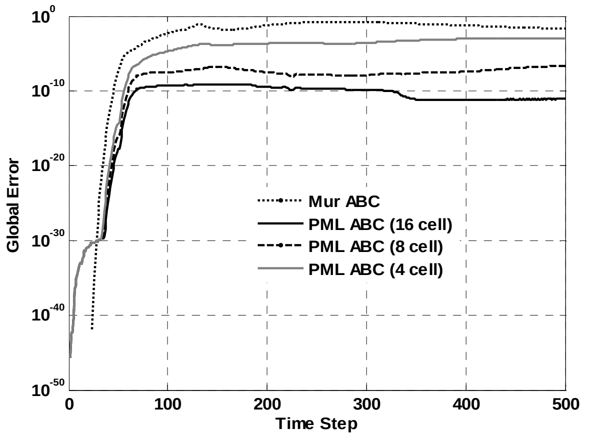

Fig. 4 Global error within a 100×50 cell two-dimensional TM test grid

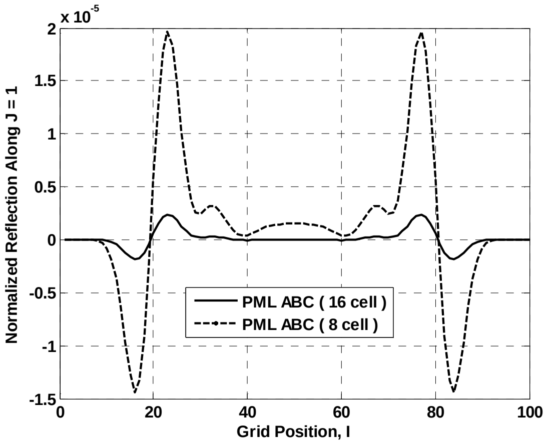

Fig. 5 Comparative local error measures for Mur and PML ABCs along the test grid outer boundary at time-step n = 100.

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

ib  =  ie + 1;
jb  =  je + 1;

dx = 3.0e-3;       %space increment of square lattice
dt = dx/(2.0*cc);   %time step

nmax  =  500;        %total number of time steps

iebc  =  8;          %thickness of left and right PML region
jebc  =  8;          %thickness of front and back PML region
rmax  =  0.00001;
orderbc  =  2;
ibbc  =  iebc+1;
jbbc  =  jebc+1;
iefbc  =  ie + 2*iebc;
jefbc  =  je + 2*jebc;
ibfbc  =  iefbc + 1;
jbfbc  =  jefbc + 1;
%***********************************************
%     Material parameters
%***********************************************
media = 2;
eps  = [1.0 4.0];
sig  = [0.0 0.12];
mur = [1.0 1.0];
sim  = [0.0 0.0];
%***********************************************
%     Field arrays
%***********************************************
ez = zeros(ibfbc,jbfbc);          %fields in main grid
ezx = zeros(ibfbc,jbfbc);
ezy = zeros(ibfbc,jbfbc);
hx = zeros(iefbc,jefbc);
hy = zeros(iefbc,jefbc);
%***********************************************
%     Updating coefficients
%***********************************************
for i = 1:media
  eaf = dt*sig(i)/(2.0*epsz*eps(i));
  ca(i) = (1.0-eaf)/(1.0+eaf);
  cb(i) = dt/epsz/eps(i)/dx/(1.0+eaf);
  haf = dt*sim(i)/(2.0*muz*mur(i));
  da(i) = (1.0-haf)/(1.0+haf);
  db(i) = dt/muz/mur(i)/dx/(1.0+haf);
end
%***********************************************
%     Geometry specification (main grid)
%***********************************************
%     Initialize entire main grid to free space
caezx(1:ibfbc,1:jbfbc) = ca(1);
cbezx(1:ibfbc,1:jbfbc) = cb(1);
caezy(1:ibfbc,1:jbfbc) = ca(1);
cbezy(1:ibfbc,1:jbfbc) = cb(1);
dahxy(1:iefbc,1:jefbc) = da(1);
dbhxy(1:iefbc,1:jefbc) = db(1);
dahyx(1:iefbc,1:jefbc) = da(1);
dbhyx(1:iefbc,1:jefbc) = db(1);

%     Add dielectric cylinder
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
            caezy(i,j) = ca(2);
            cbezy(i,j) = cb(2);
            caezx(i,j) = ca(2);
            cbezx(i,j) = cb(2);
        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
caezy(1:ibfbc,1) = 1.0;
cbezy(1:ibfbc,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
sigmay = bcfactor*(0.5*dx)^(orderbc+1);
ca1 = exp(-sigmay*dt/(epsz*eps(1)));
cb1 = (1-ca1)/(sigmay*dx);
caezy(1:ibfbc,jbbc) = ca1;
cbezy(1:ibfbc,jbbc) = cb1;

for j=1:jebc
  y1 = (jebc-j+1)*dx;
  y2 = (jebc-j)*dx;
  sigmay = bcfactor*(y1^(orderbc+1)-y2^(orderbc+1));
  sigmays = sigmay*(muz/(epsz*eps(1)));
  da1 = exp(-sigmays*dt/muz);
  db1 = (1-da1)/(sigmays*dx);
  dahxy(1:iefbc,j) = da1;
  dbhxy(1:iefbc,j) = db1;
end

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

for j = 1:jebc
  y1 = j*dx;
  y2 = (j-1)*dx;
  sigmay = bcfactor*(y1^(orderbc+1)-y2^(orderbc+1));
  sigmays = sigmay*(muz/(epsz*eps(1)));
  da1 = exp(-sigmays*dt/muz);
  db1 = (1-da1)/(sigmays*dx);
  dahxy(1:iefbc,jefbc-jebc+j) = da1;
  dbhxy(1:iefbc,jefbc-jebc+j) = db1;
end

%     LEFT region
caezx(1,1:jbfbc) = 1.0;
cbezx(1,1:jbfbc) = 0.0;
for i = 2:iebc
  x1 = (iebc-i+1.5)*dx;
  x2 = (iebc-i+0.5)*dx;
  sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
  ca1 = exp(-sigmax*dt/(epsz*eps(1)));
  cb1 = (1-ca1)/(sigmax*dx);
  caezx(i,1:jbfbc) = ca1;
  cbezx(i,1:jbfbc) = cb1;
end
sigmax = bcfactor*(0.5*dx)^(orderbc+1);
ca1 = exp(-sigmax*dt/(epsz*eps(1)));
cb1 = (1-ca1)/(sigmax*dx);
caezx(ibbc,1:jbfbc) = ca1;
cbezx(ibbc,1:jbfbc) = cb1;

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

%     RIGHT region
caezx(ibfbc,1:jbfbc) = 1.0;
cbezx(ibfbc,1:jbfbc) = 0.0;
for i = 1:iebc-1
  x1 = (i+0.5)*dx;
  x2 = (i-0.5)*dx;
  sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
  ca1 = exp(-sigmax*dt/(epsz*eps(1)));
  cb1 = (1-ca1)/(sigmax*dx);
  caezx(ibfbc-iebc+i,1:jbfbc) = ca1;
  cbezx(ibfbc-iebc+i,1:jbfbc) = cb1;
end
sigmax = bcfactor*(0.5*dx)^(orderbc+1);
ca1 = exp(-sigmax*dt/(epsz*eps(1)));
cb1 = (1-ca1)/(sigmax*dx);
caezx(ibfbc-iebc+1,1:jbfbc) = ca1;
cbezx(ibfbc-iebc+1,1:jbfbc) = cb1;

for i = 1:iebc
  x1 = i*dx;
  x2 = (i-1)*dx;
  sigmax = bcfactor*(x1^(orderbc+1)-x2^(orderbc+1));
  sigmaxs = sigmax*(muz/(epsz*eps(1)));
  da1 = exp(-sigmaxs*dt/muz);
  db1 = (1-da1)/(sigmaxs*dx);
  dahyx(iefbc-iebc+i,1:jefbc) = da1;
  dbhyx(iefbc-iebc+i,1:jefbc) = 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(:,:) = dahxy(:,:).*hxy(:,:) + dbhxy(:,:).*...
        (ezx(1:iefbc,1:jefbc) + ezy(1:iefbc,1:jefbc) - ...
        ezx(1:iefbc,2:jbfbc) - ezy(1:iefbc,2:jbfbc));

    hyx(:,:) = dahyx(:,:).*hyx(:,:) + dbhyx(:,:).*...
        (ezx(2:ibfbc,1:jefbc) + ezy(2:ibfbc,1:jefbc) - ...
        ezx(1:iefbc,1:jefbc) - ezy(1:iefbc,1:jefbc));

%************************************************
    %     END TIME-STEPPING LOOP
%************************************************
end
```

REFERENCES

[1] K.S. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media,” *IEEE Trans. Antennas & Propag.*, vol. 14, 1966, pp: 302-307.

[2] A. Taflove, and M.E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Trans. on Microwave Theo. & Tech.*, vol. 23, no. 8, 1975, pp.:623–630.

[3] A. Taflove, ” Application of the finite-difference time-domain method to sinusoidal steady state electromagnetic penetration problems,” *IEEE Trans. Electromag. Compatibil.*, vol. 22, 1980, pp.: 191-202.

[4] A. Taflove, S. Hagness, “ *Computational Electrodynamics: The Finite-Difference Time-Domain Method*,” 3rd Ed.. Artech House Publishers. 2005.

[5] K. Umashankar, and A. Taflove, "A novel method to analyze electromagnetic scattering of complex objects," *IEEE Trans. Electromag. Compatibil.*, vol. 24, 1982, pp.:397–405.

[6] J.P. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves,” *J. Comput. Phys.*, vol. 114, 1994, pp.: 185-200.

[7] R. Harrington, “*Time Harmonic Electromagnetic Fields*,” New York: McGraw-Hill. 1961.

[8] T. Moore, J. Blaschak, A. Taflove, and G. Kriegsmann, “Theory and application of radiation boundary operators,” *IEEE Trans. Antennas & Propagat.*, vol. 36, 1988, pp. 1797-1812.

[9] G. Mur, “Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations,” *IEEE Trans. Electromag. Compatibil.*, vol. 23, no. 4, 1981, pp. 377-382.

G. Akram (M’06) Gasmelseed 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 Malasysia (UTM), Malaysia, in 2009. He is the founder and chair of IEEE EMBS Scoiety. 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.