IMPLEMENTATION OF UNSPLIT PERFECTLY MATCHED LAYER ABSORBING BOUNDARY CONDITION IN 3 DIMENSIONAL FINITE DIFFERENCE TIME DOMAIN METHOD

B. U. Musa

(Department of Electrical and Electronics Engineering
University of Maiduguri, Maiduguri, Nigeria)
E-mail address: musa_bu@yahoo.com

Abstract
The C++ programming language was used to implement three-dimensional (3-D) finite-difference time-domain (FDTD) technique to simulate radiation of high frequency electromagnetic waves in free space. To achieve any meaningful results the computational domain of interest should have to be truncated in some way and this is achieved by applying absorbing boundary conditions. A uniaxial perfectly matched layer (UPML) absorbing boundary condition is used in this work. The discretised equations of the UPML in FDTD time stepping scheme were derived and has been successfully implemented using the computer program. Simulation results showed that the UPML behaves as an absorber. This was confirmed by comparing the results with another boundary condition, the Mur ABC.

Keywords: Electromagnetics, 3D Finite Difference Time Domain, 3D Unsplit field perfectly Matched Layer, Absorbing boundary conditions

1. Introduction
Electromagnetic fields can be computed either in free space in enclosed space. The behaviour of the electromagnetic fields distributions at the interface between two or more media is completely different from the field distribution away from the interface. For this reason, it is necessary to define the conditions at the interfaces when solving for the electromagnetic fields. The FDTD technique is based on direct solution of Maxwell’s equations in differential form, and for partial differential equations to be well posed; an appropriate number of boundary conditions need to be specified. As described by the uniqueness theorem (Georgopoulos et al., 2002), it is required to specify the tangential components of the electric field $E$ and magnetic field $H$ (or both) over the boundary of a domain to be sufficient for a unique solution.

Since the introduction of PML to the electromagnetic community by Berenger in 1994, it has been applied widely to the solution of numerous problems. The area of application is too numerous that it is almost impossible to discuss in a single paper. Another important development in the PML is the extensions to Berenger’s original split field algorithm. One such proposal is that the PML can be realized through an anisotropic medium having complex permittivity and permeability tensors, known as Unsplit Field Perfectly Matched Layer (UPML) Formulation. This strategy was first put forward and discussed by (Sacks et al., 1995) and implemented in the FDTD grid by (Gedney, 1996). The solution of maxwell’s equations in differential form using the FDTD method in three dimensions produces 6 electromagnetic field quantities-$E_x$, $E_y$, $E_z$, $H_x$, $H_y$ and $H_z$. Following the derivation of the 3 electric field quantities in (Taflove and Hagness, 2000), the derivation of the 3 magnetic field quantities is covered in this paper. These together with the 3 electric field quantities are then implemented in the FDTD
grid using C++ programme. A region with specified boundaries was chosen and simulated to demonstrate the effectiveness of the UPML scheme.

2. Formulations of the problem

2.1 The FDTD Method

The FDTD method is one of the numerical techniques in computational electromagnetics, which computes electromagnetic signal scattering or radiation from objects of arbitrary shapes. The method relies on discretizing Maxwell’s curl equations in both time and space, where all derivatives are approximated by the central difference equation. The computational region of interest where the solution is sought is divided into cells with the corresponding electric and magnetic fields being located on the edges and the faces. It usually assumed that all fields in the entire solution region are initially zero. The scattering or radiating objects of interest are defined by specifying the dielectric and magnetic material parameters at each discrete location in the FDTD mesh. The computational region must be large enough to enclose the objects to be analyzed. In addition, suitable boundary conditions on the artificial boundary must be imposed to absorb outgoing waves to simulate the extension of the computational region to infinity (Yee, 1966). Finally, to ensure the stability of the finite-difference scheme, the time increment \( \Delta t \) must satisfy the Courant condition (Yee, 1966) as stated in Equation (1),

\[
\Delta t = \frac{1}{v \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}}
\]

where \( v = \sqrt{\mu \varepsilon} \) is the maximum phase velocity in the model, \( \mu \) and \( \varepsilon \) are the permeability and permittivity of the medium respectively, and \( \Delta x, \Delta y \) and \( \Delta z \) are the spatial mesh increments in the \( x, y \) and \( z \) directions respectively.

2.2 Boundary Conditions

Boundary conditions applied on the outer boundaries of structures are usually categorized under the following (Juntunen, 2001):

(a) Perfect electric conductor (PEC) or electric wall, which is used to represent ideal conductors.
(b) Perfect magnetic conductor (PMC) or magnetic wall, which may be used for field calculations in structures with symmetry planes in order to reduce the size of the computational volume.
(c) Absorbing boundary condition (ABC), applied in the suppression of waves in open boundaries.

When the problem space is an “open” boundary, for example, those encountered in radiations, scattering, and so on, there are no exact boundary conditions known. However, to apply a boundary to such regions, a computational domain large enough to enclose the structure of interest is simulated. This domain is truncated by applying appropriate artificial boundary conditions on the outer perimeter of the domain that simulates its extension to infinity. The artificial boundary conditions should allow the outgoing propagating waves to exit the domain, and should suppress spurious reflections of this wave to an acceptable level. This is what absorbing boundary is all about. There are numerous absorbing boundary conditions available for use in numerical methods such as the FDTD method. Absorbing boundary conditions (ABC) available for use with numerical methods such as the FDTD are broadly divided into two:
The first one is Analytical ABCs—those derived from differential equations. The idea of the analytical ABCs is to construct a linear partial differential operator from the derivatives of the fields. This differential operator cancels terms of the outgoing wave expansions. This leaves a remainder term, which diminishes to zero as an inverse power of the distance from the observation point to the origin. Examples are Engquist and Majda one-way operator ABC (Enquist and Majda, 1977), Ramahi’s Complementary operator (Ramahi, 1997), Mei-Fang’s super absorption (Mei and Fang, 1992), and Liao’s extrapolation (Liao et al., 1984). One drawback with these ABCs is that it does not account for the fringing fields propagating tangential to the boundaries. This places the need to place the boundaries well away from any fringing fields. To address this problem, more effective and powerful ABCs, known as Material absorbers were introduced. This includes Mur’s ABC (Mur, 1981) and the perfectly matched Layer (PML) ABC (Berenger, 1994).

Consider Figure 1(a) where an electric dipole antenna is placed in some infinite homogeneous medium. Due to time and memory limitations, this infinite space must be truncated such that the electromagnetic fields are solved only at the region of interest (within the dashed box in this case). A material of finite thickness is placed surrounding the dashed box as shown Figure 1(b). The idea is that any propagating wave from the antenna incident on the material is absorbed by the material with no reflections onto the truncated domain. This material absorber is known as the perfectly matched layer (PML).

Figure 1: Radiating dipole in (a) Infinite medium (b) Truncated medium with material absorber

2.3 The UPML Formulation

In the unsplit field perfectly matched layer (UPML) formulation the anisotropic medium is assumed to be uniaxial having permittivity and permeability tensors given in the form:

\[ \bar{\varepsilon} = \varepsilon \bar{s} \]

\[ \bar{\mu} = \mu \bar{s} \]  \hspace{1cm} (2a)

where: \( \bar{s} \) is a three dimensional diagonal tensor expressed as
In which $s_x$, $s_y$, and $s_z$ are any complex numbers given as 

\[
 s_x = k_x + \frac{\sigma_x}{j\omega\varepsilon} \quad (2c) ; \quad s_y = k_y + \frac{\sigma_y}{j\omega\varepsilon} \quad (2d) ; \quad s_z = k_z + \frac{\sigma_z}{j\omega\varepsilon} \quad (2e)
\]

By assigning proper values to $s_x$, $s_y$, and $s_z$ the medium described by equation $(2a)$ is known as uniaxial or unsplit perfectly matched layer (UPML). It was shown in (Sacks et al., 1995) that a plane wave impinging from vacuum into this medium would experience no reflection regardless of frequency or angle of incidence. It should be noted that UPML is uniaxial only in regions where the absorbing layers do not overlap. At the overlap regions, the parameters vary in all directions hence no longer uniaxial. As explained in (Taflove and Highness, 2000), the strategy of implementing UPML throughout the FDT space is as follows: With reference to Figure 2, $\vec{s}$ varies for each region according to the grid position. Assuming the following: 

$(\sigma_{x1}, \sigma_{x1}^*, \sigma_{x2}, \sigma_{x2}^*), (\sigma_{y1}, \sigma_{y1}^*, \sigma_{y2}, \sigma_{y2}^* )$ and $(\sigma_{z1}, \sigma_{z1}^*, \sigma_{z2}, \sigma_{z2}^* )$ are the electric and magnetic conductivities at the faces $x = 1, y = 1,2$ and $z = 1,2$ respectively as shown in Figure 3.

3. Method of Solution

Computational space chosen for implementation and demonstration of the UPML is shown in Figure 4. The computational domain was set as $50 \times 50 \times 50$ cells and the spatial increments $\Delta x = \Delta y = \Delta z$ are set to 0.5m which gives a discrete time step $\Delta t$ as $9.141 \times 10^{10}$. A 10-layer thick perfectly matched layer is used. A differentiated Gaussian pulse is placed at the centre of the box oriented in the positive y direction as shown. The equation describing the current is: 

\[
 I = I_m e^{-\left(\frac{t}{\beta}\right)^2}
\]

with $I_m = 1000$ A, the peak magnitude of the current, $\beta = 5 \times 10^{-7}$.

The flow chart for computation of electric and magnetic fields everywhere within the computational space is shown in Figure 5.
Musa: Implementation of unsplit perfectly matched layer absorbing boundary condition in 3 dimensional finite difference time domain method. 
AZOJETE, 13(2):238-251. ISSN 1596-2490; e-ISSN 2545-5818, www.azojete.com.ng

Figure 4: Computational space for implementation of Unsplit Perfectly Matched Layer

![Diagram](image1)

Figure 5: Flow diagram for the computation of Electromagnetic Fields

The derivation of electromagnetic fields and the computer implementation of the field vectors in C++ is given in appendix.

4 Results and Discussion

Figure 6 shows the x-z plane view of the simulated electromagnetic fields advancing towards the boundary of the computational region described in Figure 4. It can be seen that as the wave reaches the edges of the computational space, there is little or no reflection at the boundaries.

![Images](image2)

Figure 6: Propagation of Electromagnetic waves towards the boundary of the computational region
Thus the unwanted spurious wave reflections at the boundaries, which would have added to fields the region, have been absorbed. The absorption superiority of the UPML over other ABCs has been established in the literature (Berenger, 1994). However, to demonstrate the workability of the implemented UPML FDTD using the C++, Murs ABC is compared with UPML ABC using the same computational environment set up above. Electromagnetic fields were computed at a point close to one of the boundaries to observe how the fields are absorbed. Figures 7 and 8 show the components of the electric field vectors computed applying the two absorbing boundary conditions.

It is clearly seen that the PML ABC gives a better absorption than the Mur ABC. Consider the $E_x$ components of the Electric fields, for example, the magnitude of this field computed using Murs ABC, it was about 12 kV/m at that point. At this same point, the magnitude was reduced to about 6 kV/m using the UPML. Similar observations can also be for the $E_y$ and $E_z$ fields from
the Figures. This strategy has been applied in the field of power system electromagnetic transients computation by the author in (Musa, 2009; Musa 2010).

5. Conclusion
The introduction of perfectly matched layer (PML) ABC is one of the most important developments in the field of computational electromagnetics in recent years. The original split field proposed by Berenger was extended and re-formulated in stretched coordinate form. This makes it possible to extend the concepts to other coordinates such as cylindrical and spherical coordinate systems.

Another formulation of the PML is the uniaxial anisotropic medium first put forward and implemented in the FDTD grid by Gedney (Gedney, 1996) known as Uniaxial Perfectly Matched Layer (UPML). The implementation of the PML within the FDTD grid and computer program for execution of the time stepping scheme for the fields have been demonstrated and found that electromagnetic fields were absorbed as required for an absorbing boundary. Also, comparison and another ABC has shown that the UPML ABC is a better than absorber. The field of applications are enormous, for example the study of antennas, photonics, bioengineering and so on.

References


APPENDIX A

DERIVATION OF ELECTRIC AND MAGNETIC FIELDS IN THE UPML LATTICE

Introducing the diagonal tensor in (2), the general form of Maxwell’s curl equation for the UPML space in the matched condition can be written as

\[ \Delta \times \vec{H} = j \omega \varepsilon \vec{E} \tag{4a} \]
\[ \Delta \times \vec{E} = -j \omega \mu \vec{H} \tag{4b} \]

Equation (4) applies to space regions with dielectric and permeability material properties only. Conductive material mediums are not represented within the computational space. Hence, the equations have to be modified in some way to account for representation of conductive materials that may be located within the lattice space. The author therefore seeks to concentrate on equations involving conductive media for implementation in the FDTD space.

Introducing the conductive term into (4a) therefore results in (Taflove and Highness, 2000)

\[ \Delta \times \vec{H} = (j \omega \varepsilon + \sigma) \vec{E} \tag{5} \]

Starting with Maxwell’s curl equations and using equation (2), equation (5) can be written as

\[ \begin{bmatrix} \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \end{bmatrix} = \begin{bmatrix} s_x s_z \\ s_y \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ s_x s_z \\ s_y \\ 0 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} \tag{6} \]

In which \( s_x, s_y \) and \( s_z \)are new variables introduced to implement the variations of conductivities within the PML as described earlier.

THEELECTRIC FIELDS

Discretising (6) and after some manipulations, yields the following auxiliary equations. The full derivation is given in (Taflove and Highness, 2000), chapter 7.

\[ p_x^{n+1}(i + 1/2, j, k) = AA_1 p_x^n(i + 1/2, j, k) + AA_2 \{ H_z^{n+1/2}(i + 1/2, j + 1/2, k) - H_z^{n+1/2}(i + 1/2, j - 1/2, k) \} - H_y^{n+1/2}(i + 1/2, j, k + 1/2) + H_y^{n+1/2}(i + 1/2, j, k - 1/2) \} \]

\[ AA_1 = \frac{2 \varepsilon - \sigma \Delta t}{2 \varepsilon + \sigma \Delta t} AA_2 = \frac{2 \varepsilon \Delta t}{\Delta w(2 \varepsilon + \sigma \Delta t)} \Delta w = \Delta x = \Delta y = \Delta z \tag{7a} \]

\[ p_y^{n+1}(i, j + 1/2, k) = BB_1 p_y^n(i, j + 1/2, k) + BB_2 \{ H_x^{n+1/2}(i, j + 1/2, k + 1/2) - H_x^{n+1/2}(i, j + 1/2, k - 1/2) \} - H_z^{n+1/2}(i + 1/2, j + 1/2, k) + H_z^{n+1/2}(i - 1/2, j + 1/2, k) \} \]

\[ BB_1 = \frac{2 \varepsilon - \sigma \Delta t}{2 \varepsilon + \sigma \Delta t} BB_2 = \frac{2 \varepsilon \Delta t}{\Delta w(2 \varepsilon + \sigma \Delta t)} \Delta w = \Delta x = \Delta y \tag{7b} \]
\[ P_{z}^{n+1}(i, j, k + 1/2) = CC_{1}P_{z}^{n}(i, j, k + 1/2) + CC_{2}[H_{y}^{n+1/2}(i + 1/2, j, k + 1/2) - H_{y}^{n+1/2}(i - 1/2, j, k - 1/2) - H_{x}^{n+1/2}(i, j + 1/2, k + 1/2) + H_{x}^{n+1/2}(i, j - 1/2, k + 1/2)] \]

\[ CC_{1} = \frac{2\varepsilon - \sigma \Delta t}{2\varepsilon + \sigma \Delta t}, \quad CC_{2} = \frac{2\varepsilon \Delta t}{\Delta w(2\varepsilon + \sigma \Delta t)} \Delta w = \Delta x = \Delta y \] (7c)

\[ P_{x}^{n+1}(i + 1/2, j, k) = A_{11}P_{x}^{n}(i + 1/2, j, k) + A_{22}\left\{ P_{x}^{n+1}(i + 1/2, j, k) - P_{x}^{n}(i + 1/2, j, k) \right\} \]

\[ A_{11} = \frac{2\varepsilon_{0}k_{y} - \sigma_{y}\Delta t}{2\varepsilon_{0}k_{y} + \sigma_{y}\Delta t}, \quad A_{22} = \frac{2\varepsilon_{0}}{2\varepsilon_{0} + \sigma_{y}\Delta t} \] (8a)

\[ P_{y}^{n+1}(i + 1/2, j, k) = B_{11}P_{y}^{n}(i + 1/2, j, k) + B_{22}\left\{ P_{y}^{n+1}(i + 1/2, j, k) - P_{y}^{n}(i + 1/2, j, k) \right\} \]

\[ B_{11} = \frac{2\varepsilon_{0}k_{z} - \sigma_{z}\Delta t}{2\varepsilon_{0}k_{z} + \sigma_{z}\Delta t}, \quad B_{22} = \frac{2\varepsilon_{0}}{2\varepsilon_{0} + \sigma_{z}\Delta t} \] (8b)

\[ P_{z}^{n+1}(i + 1/2, j, k) = C_{11}P_{z}^{n}(i + 1/2, j, k) + C_{22}\left\{ P_{z}^{n+1}(i + 1/2, j, k) - P_{z}^{n}(i + 1/2, j, k) \right\} \]

\[ C_{11} = \frac{2\varepsilon_{0}k_{x} - \sigma_{x}\Delta t}{2\varepsilon_{0}k_{x} + \sigma_{x}\Delta t}, \quad C_{22} = \frac{2\varepsilon_{0}}{2\varepsilon_{0} + \sigma_{x}\Delta t} \] (8c)

\[ P_{x}, P_{y}, P_{z}, P_{x}', P_{y}', \text{and } P_{z}' \] are the time domain transformations of the auxiliary variable introduced in (Taflove and Hagness, 2000)

Further manipulations of the above equations yields explicit time stepping schemes for \( E_{x}, E_{y} \) and \( E_{z} \) as follows:

\[ E_{x}^{n+1}(i + 1/2, j, k) = A_{1}E_{x}^{n}(i + 1/2, j, k) + A_{2}[A_{3}P_{x}^{n+1}(i + 1/2, j, k) - A_{4}P_{x}^{n}(i + 1/2, j, k)] \]

\[ A_{1} = \frac{2\varepsilon_{0}k_{z} - \sigma_{z}\Delta t}{2\varepsilon_{0}k_{z} + \sigma_{z}\Delta t}, \quad A_{2} = \frac{1}{2\varepsilon_{0}k_{z} + \sigma_{z}\Delta t}, \quad A_{3} = 2\varepsilon_{0}k_{x} + \sigma_{x}\Delta tA_{4} = \frac{2\varepsilon_{0}k_{x} - \sigma_{x}\Delta t}{2\varepsilon_{0}k_{x} + \sigma_{x}\Delta t} \] (9a)

\[ E_{y}^{n+1}(i + 1/2, j, k) = B_{1}E_{y}^{n}(i + 1/2, j, k) + B_{2}[B_{3}P_{y}^{n+1}(i + 1/2, j, k) - B_{4}P_{y}^{n}(i + 1/2, j, k)] \]

\[ B_{1} = \frac{2\varepsilon_{0}k_{x} - \sigma_{x}\Delta t}{2\varepsilon_{0}k_{x} + \sigma_{x}\Delta t}, \quad B_{2} = \frac{1}{2\varepsilon_{0}k_{x} + \sigma_{x}\Delta t}, \quad B_{3} = 2\varepsilon_{0}k_{y} + \sigma_{y}\Delta tB_{4} = \frac{2\varepsilon_{0}k_{y} - \sigma_{y}\Delta t}{2\varepsilon_{0}k_{y} + \sigma_{y}\Delta t} \] (9b)

\[ E_{z}^{n+1}(i + 1/2, j, k) = C_{1}E_{z}^{n}(i + 1/2, j, k) + C_{2}[C_{3}P_{z}^{n+1}(i + 1/2, j, k) - C_{4}P_{z}^{n}(i + 1/2, j, k)] \]
Musa: Implementation of unsplit perfectly matched layer absorbing boundary condition in 3
dimensional finite difference time domain method.
AZOJETE, 13(2):238-251. ISSN 1596-2490; e-ISSN 2545-5818, www.azojete.com.ng

\[ C_1 = \frac{2\varepsilon_0 k_y - \sigma_y \Delta t}{2\varepsilon_0 k_y + \sigma_y \Delta t}, \quad C_2 = \frac{1}{2\varepsilon_0 k_y + \sigma_y \Delta t}, \quad C_3 = 2\varepsilon_0 k_z + \sigma_z \Delta t \]
\[ = 2\varepsilon_0 k_z - \sigma_z \Delta t \]  
(9c)

\(\sigma_x, \sigma_y, \sigma_z, k_x, k_y, \) and \(k_z\) vary spatially along their axes within the PML.

It should be noted that equations (7) to (9) are required to update the \(\vec{E}\) components in the
UPML.

DERIVATION OF MAGNETIC FIELDS

The full derivations of the \(\vec{H}\) field components have not been shown in the literature. Therefore
following similar procedure used to derive the \(\vec{E}\) fields, the \(\vec{H}\) fields are derived. This is done by
using (2b) and writing matrix equation as:

\[
\begin{bmatrix}
\frac{\partial \vec{E}_x}{\partial y} - \frac{\partial \vec{E}_y}{\partial z} \\
\frac{\partial \vec{E}_x}{\partial z} - \frac{\partial \vec{E}_z}{\partial x} \\
\frac{\partial \vec{E}_y}{\partial x} - \frac{\partial \vec{E}_x}{\partial y}
\end{bmatrix} = -j \omega \mu
\begin{bmatrix}
s_y s_z & 0 & 0 \\
0 & s_x s_z & 0 \\
0 & 0 & s_x s_y
\end{bmatrix}
\begin{bmatrix}
\vec{H}_x \\
\vec{H}_y \\
\vec{H}_z
\end{bmatrix}
\]
(10)

And choosing a constitutive relationship
\(\vec{B}_x = \mu \frac{s_z}{s_x} \vec{H}_x \) (11a) \(\vec{B}_y = \mu \frac{s_x s_z}{s_y} \vec{H}_y \) (11b) \(\vec{B}_z = \mu \frac{s_y}{s_z} \vec{H}_z \) (11c), Equation (10) becomes

\[
\begin{bmatrix}
\frac{\partial \vec{E}_x}{\partial y} - \frac{\partial \vec{E}_y}{\partial z} \\
\frac{\partial \vec{E}_x}{\partial z} - \frac{\partial \vec{E}_z}{\partial x} \\
\frac{\partial \vec{E}_y}{\partial x} - \frac{\partial \vec{E}_x}{\partial y}
\end{bmatrix} = -j \omega
\begin{bmatrix}
s_y & 0 & 0 \\
0 & s_z & 0 \\
0 & 0 & s_x
\end{bmatrix}
\begin{bmatrix}
\vec{B}_x \\
\vec{B}_y \\
\vec{B}_z
\end{bmatrix}
\]
(12)

Substituting (2c-2e) into (12) and using inverse Fourier transform \(j \omega f(\omega) \rightarrow \left(\frac{\partial}{\partial t}\right) f(t)\), an
equivalent system of differential equation is obtained as shown below.

\[
\begin{bmatrix}
\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \\
\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \\
\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}
\end{bmatrix} = \frac{\partial}{\partial t}
\begin{bmatrix}
k_y & 0 & 0 \\
0 & k_z & 0 \\
0 & 0 & k_x
\end{bmatrix}
\begin{bmatrix}
B_x \\
B_y \\
B_z
\end{bmatrix}
+ \frac{1}{\mu}
\begin{bmatrix}
\sigma_y & 0 & 0 \\
0 & \sigma_z & 0 \\
0 & 0 & \sigma_x
\end{bmatrix}
\begin{bmatrix}
B_x \\
B_y \\
B_z
\end{bmatrix}
\]
(13)

The systems of equation can be discretized in the FDTD lattice similar to the \(\vec{E}\) fields to get explicit
time stepping schemes for \(B_x, B_y\) and \(B_z\). \(B_x\) is given below with additional condition that
the cell sizes are equal in all three directions i.e. \(\Delta_x = \Delta_y = \Delta_z\)
\( B_x^{n+3/2}(i,j+1/2,k+1/2) = D_1 B_x^{n+1/2}(i,j+1/2,k+1/2) - D_2 \{ E_x^{n+1}(i,j+1,k+1/2) - E_x^{n+1}(i,j,k+1/2) - E_y^{n+1}(i,j+1/2,k+1) + E_y^{n+1}(i,j+1/2,k) \} \) (14a)

\[
D_1 = \frac{2\varepsilon k_x - \sigma_y \Delta t}{2\varepsilon k_x + \sigma_x \Delta t}, \quad D_2 = \frac{2\varepsilon \Delta t}{\Delta w(2\varepsilon k_y + \sigma_y \Delta t)}
\]

\( B_y \) and \( B_z \) are derived following similar procedure and given below

\( B_y^{n+3/2}(i,j+1/2,k+1/2) = E_1 B_y^{n+1/2}(i,j+1/2,k+1/2) - E_2 \{ E_x^{n+1}(i,j+1,k+1) + E_y^{n+1}(i+1/2,j,k+1/2) \} \) (14b)

\[
E_1 = \frac{2\varepsilon k_x - \sigma_y \Delta t}{2\varepsilon k_x + \sigma_x \Delta t}, \quad E_2 = \frac{2\varepsilon \Delta t}{\Delta w(2\varepsilon k_x + \sigma_x \Delta t)}
\]

\( B_z^{n+3/2}(i,j+1/2,k+1/2) = F_1 B_z^{n+1/2}(i,j+1/2,k+1/2) - F_2 \{ E_x^{n+1}(i+1/2,j,k+1) + E_y^{n+1}(i,j+1,1/2) \} \) (14c)

\[
F_1 = \frac{2\varepsilon k_x - \sigma_x \Delta t}{2\varepsilon k_x + \sigma_x \Delta t}, \quad F_2 = \frac{2\varepsilon \Delta t}{\Delta w(2\varepsilon k_x + \sigma_x \Delta t)}
\]

(11a)-(11c) are considered next. For (9a), multiply both sides by \( s_x \) and substitute \( s_x \) and \( s_z \) from (1d) and (1e) to get

\[
\left( k_x + \frac{\sigma_x}{j\omega} \right) \tilde{B}_x = \mu \left( k_z + \frac{\sigma_z}{j\omega} \right) \tilde{H}_x
\] (15)

Multiplying both side by \( j\omega \) and transforming into the time domain yields

\[
\frac{\partial}{\partial t} (k_x B_x) + \frac{\sigma_x}{\varepsilon} B_x = \mu \left[ \frac{\partial}{\partial t} (k_x H_x) + \frac{\sigma_x}{\varepsilon} H_x \right]
\] (16a)

Similar derivations for (20b) and (20c) gives

\[
\frac{\partial}{\partial t} (k_y B_y) + \frac{\sigma_y}{\varepsilon} B_y = \mu \left[ \frac{\partial}{\partial t} (k_y H_y) + \frac{\sigma_y}{\varepsilon} H_y \right]
\] (16b)

\[
\frac{\partial}{\partial t} (k_z B_z) + \frac{\sigma_z}{\varepsilon} B_z = \mu \left[ \frac{\partial}{\partial t} (k_z H_z) + \frac{\sigma_z}{\varepsilon} H_z \right]
\] (16c)

The time derivative (16) are descritized in the standard algorithm yielding explicit time stepping for \( H_x, H_y \) and \( H_z \). They are given below

\( H_x^{n+3/2}(i,j+1/2,k+1/2) = D_3 H_x^{n+1/2}(i,j+1/2,k+1/2) + D_4 \{ D_5 B_x^{n+3/2}(i,j+1/2,k+1/2) - D_6 B_x^{n+1/2}(i,j+1/2,k+1/2) \} \) (17a)

\[
D_3 = \frac{2\varepsilon k_x - \sigma_x \Delta t}{2\varepsilon k_x + \sigma_x \Delta t}, \quad D_4 = \frac{1}{(2\varepsilon k_x + \sigma_x \Delta t)\mu}, \quad D_5 = 2\varepsilon k_x + \sigma_x \Delta t, \quad D_6 = 2\varepsilon k_x - \sigma_x \Delta t
\]

\( H_y^{n+3/2}(i,j+1/2,k+1/2) = E_3 H_y^{n+1/2}(i,j+1/2,k+1/2) + E_4 \{ E_5 B_y^{n+3/2}(i,j+1/2,k+1/2) - E_6 B_y^{n+1/2}(i,j+1/2,k+1/2) \} \) (17b)

\[
E_3 = \frac{2\varepsilon k_x - \sigma_x \Delta t}{2\varepsilon k_x + \sigma_x \Delta t}, \quad E_4 = \frac{1}{(2\varepsilon k_x + \sigma_x \Delta t)\mu}, \quad E_5 = 2\varepsilon k_y + \sigma_y \Delta t, \quad E_6 = 2\varepsilon k_y - \sigma_y \Delta t
\]
\[ H_z^{n+3/2}(i + 1/2, j + 1/2, k) \]
\[ = F_3 H_z^{n+1/2}(i + 1/2, j + 1/2, k) \]
\[ + F_4 [F_5 B_z^{n+1/2}(i + 1/2, j + 1/2, k) - F_6 B_z^{n+1/2}(i + 1/2, j + 1/2, k)] \]  
\[ F_3 = \frac{2 \varepsilon k_y - \sigma_y \Delta t}{2 \varepsilon k_y + \sigma_y \Delta t} \]
\[ F_4 = \frac{1}{(2 \varepsilon k_y + \sigma_y \Delta t) \mu} \]
\[ F_5 = 2 \varepsilon k_z + \sigma_z \Delta t \]
\[ F_6 = 2 \varepsilon k_z - \sigma_z \Delta t \]

Thus overall, computing the electromagnetic fields in the UPML strategy requires updating equations (7) through (17) at each time step.
APPENDIX B

The computer implementation of the field vectors in C++ is illustrated for the $E_x$ components expressions given in (7), (8) and (9). Implementation of rest of the fields follows the similar procedure. At the beginning of the program, eight coefficients arrays for use with the field update equations are computed. The arrays in (18) hold the permittivity and conductivity properties of the materials within the computational space.

The arrays are given below:

$$A1 = \frac{2\varepsilon - \sigma \Delta t}{2\varepsilon + \sigma \Delta t}$$  \hspace{1cm} (18a)

$$A2 = \frac{2\varepsilon \Delta t}{\Delta x (2\varepsilon + \sigma \Delta t)}$$  \hspace{1cm} (18b)

$$A3[j] = \frac{2\varepsilon_0 k_y[j] - \sigma_y[j] \Delta t}{2\varepsilon_0 k_y[j] + \sigma_y[j] \Delta t}$$  \hspace{1cm} (19a)

$$A4[j] = \frac{2\varepsilon_0}{2\varepsilon_0 + \sigma_y[j] \Delta t}$$  \hspace{1cm} (19b)

$$A5[k] = \frac{2\varepsilon_0 k_x[k] - \sigma_x[k] \Delta t}{2\varepsilon_0 k_x[k] + \sigma_x[k] \Delta t}$$  \hspace{1cm} (20a)

$$A6[k] = \frac{1}{2\varepsilon_0 k_x[k] + \sigma_x[k] \Delta t}$$  \hspace{1cm} (20b)

$$A7[i] = 2\varepsilon_0 k_x[i] + \sigma_x[i] \Delta$$  \hspace{1cm} (21a)

$$A8[i] = 2\varepsilon_0 k_x[i] - \sigma_x[i] \Delta$$  \hspace{1cm} (22b)

The C++ program segment that computes $E_x$ everywhere in the FDTD space is presented below:

For (i = 0; i<Nx-1; i++)

For (j = 0; j<Ny-1; j++)

For (k = 0; k<Nz-1; k++)

{ Pxstore = (*Px) [i] [j] [k];
  (*Px) [i] [j] [k] = A1*(Px) [i] [j] [k]
  + A2*((Hz) [i] [j] [k] - (Hz) [i] [j-1] [k] - (Hy) [i] [j] [k] - (Hy) [i] [j] [k-1]);
  Dxstore = (*Dx) [i] [j] [k];
  (*Dx) [i] [j] [k] = A3[j]>(*Dx) [i] [j] [k]
  + A4[j](*(Px) [i] [j] [k] - Pxstore);
  (*Ex) [i] [j] [k] = A5[k](*(Ex) [i] [j] [k] - Exstore);
  (*Ex) [i] [j] [k] = A5[k](*(Ex) [i] [j] [k] - Exstore);
}