UNIVERSITY OF ÇUKUROVA
INSTITUTE OF NATURAL AND
APPLIED SCIENCES

Hasan YAVUNCU **MSc THESIS**

SOLUTION OF THE SCATTERING OF A PLANE WAVE FROM AN INHOMOGENEOUS SPHERE WITH FDTD METHOD

DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ADANA, 2006

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SOLUTION OF THE SCATTERING OF A PLANE WAVE FROM AN INHOMOGENEOUS SPHERE WITH FDTD METHOD

HASAN YAVUNCU

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK ANA BİLİM DALI

Bu Tez 29/12/2006 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından
Oybirliği/Oyçokluğu İle Kabul Edilmiştir.

İmza………………	İmza……………………..…	İmza………………
Doç.Dr.Turgut İKİZ	Prof.Dr.V.PROGREBNYAK	Yrd.Doç.Dr. Faruk KARADAĞ
DANIŞMAN	ÜYE	ÜYE

Bu tez Enstitümüz Elektrik-Elektronik Anabilim Dalında hazırlanmıştır.

Kod No:

Prof. Dr. Aziz ERTUNÇ

Enstitü Müdürü

ABSTRACT

MSc THESIS

SOLUTION OF THE SCATTERING OF PLANE WAVE FROM AN INHOMOGENEOUS SPHERE WITH FDTD METHOD

Hasan YAVUNCU

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF CUKUROVA

Supervisor : Assoc. Prof. Dr. Turgut İKİZ
: Year 2006, Pages 64
Jury : Assoc. Prof. Dr.Doç. Dr. Turgut İKİZ
Prof. Dr. Victor PROGREBNYAK
Asst. Prof. Dr. Faruk KARADAĞ

It is necessary to investigate the scattering of electromagnetic waves from various geometries and the effects of these geometries to scattering and radiation for analysis and synthesis of the electromagnetic scatterers and sources. In this study, scattering from an inhomogeneous canonical structure is investigated. Scattering from a spherical structure is a canonical problem. Finite difference time domain (FDTD) will be used as the numerical method for the solution of the problem. The scattered field from the sphere will be determined in the specified space and after for obtaining the field expressions outside this space perfectly matched layer (PML) will be used for the boundary condition.

Key Words: FDTD, Numeric, Sphere, Scattering

ÖZ

YÜKSEK LİSANS TEZİ

HOMOJEN OLMAYAN BİR KÜREDEN BİR DÜZLEMSEL DALGANIN SAÇILIMININ FDTD ÇÖZÜMÜ

HASAN YAVUNCU

ÇUKUROVA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK ANABİLİM DALI

Danışman : Doç.Dr. Turgut İKİZ
Yıl 2006, Sayfa 64
Jüri : Doç. Dr. Turgut İKİZ
Prof. Dr. Victor PROGREBNYAK
Yrd. Doç. Dr. Faruk KARADAĞ

Elektromanyetik dalgaların çeşitli geometrilere sahip nesnelerden saçılmasının incelenmesi ve bu nesnelerin ışınıma olan etkilerinin araştırılması elektromanyetik saçıcı ve yayıcıların analiz ve sentezi için zorunludur. Bu çalışmada homojen olmayan bir üç boyutlu kanonik yapıdan saçılma problemi FDTD metodu kullanılarak incelenmiştir. Belirlenen uzay için küreden saçılan alan hesaplanmış ve daha sonra bu uzayın dışındaki alan ifadelerinin elde edilebilmesi için sınır koşulu olarak mükemmel uyumlu tabaka uygulanmıştır.

Anahtar Kelimeler: FDTD, Nümerik, Küre, Saçılma

ACKNOWLEDGEMENTS

First of all I would like to express my appreciation to my supervisor Assoc. Prof. Dr. Turgut İKİZ for all of his supports, guidance, suggestions, patience and encouragement on initiating, improving and completing this study.

I would like to express my appreciation to Prof. Dr. A. Hamit SERBEST for all of his guidance and suggestions.

I would like to extend my heartfelt appreciation to my parents for their constant love, moral and supports.

I would like to thank to Ali ARINÇ and Hüseyin ÖZKAN for their moral supports and thanks to Mustafa K. ZATEROĞLU, Duygu ÇAKIR, Adem BİLGİLİ and Asil DEDEOĞLU for their great encouragement and cooperates on improvement of this thesis during the study.

Lastly, I would like to express my appreciation to Bostan YÜCE and Zeliha ER for their supports.

CONTENTS **PAGE**

LIST OF FIGURES **PAGE**

1. INTRODUCTION

It is necessary to investigate the scattering of electromagnetic waves from various geometries and the effects of these geometries to scattering and radiation for the analysis and synthesis of the electromagnetic scatterers and sources. Electromagnetic computational encompasses the electromagnetic modeling, simulation, and analysis of the electromagnetic responses of complex systems to various electromagnetic stimuli. It provides an understanding of the system response that allows for the better design or modification of the system.

Some examples of the practical applications of electromagnetic scattering are radar, meteorology, biomedical, antenna design and radar cross section reduction.

In radar, a body is illuminated by an electromagnetic wave, and the wave is scattered and received by an antenna. The received signal is used to identify the characteristics of the object, such as its position and motion.

The scattering of waves may be used to probe atmospheric conditions, such as the size, density and motion of rain, fog, smog and cloud particles which gives useful information on the environment and for weather prediction.

In biomedical applications, microwaves, optical waves, or acoustic waves are propagated through biological media and the scattering from various portion of the body is used to identify the objects for diagnostic purposes.

In antenna applications, the presence of the scattering body affects the radiation pattern of the antenna. For example the radiation pattern of a direction finding antenna of an aircraft changes due to the scattering effect of the aircraft body. Again the support structure of a reflector antenna degrades the radiation pattern of the antenna. Therefore the effect of the scattering structure near the antenna must be considered in the antenna design.

The radar cross section (RCS) of a body is a measure of the amount energy scattered in a particular direction for a given illumination. In the case of RCS reduction, the objective is to design the body in such a way as the minimize the amount of energy scattered in a particular direction (stealth technology). Thus, the calculation of radar cross section is important in the design of potential radar targets.

The four forces in nature are strong, weak, electromagnetic, and gravitational. The electromagnetic force is the most technologically pervasive. The three methods of predicting electromagnetic effects are experiment, analysis and computation. Computation is the newest and fastest-growing approach. Of the many approaches to electromagnetic computation, including method of moments, finite difference time domain, finite element, geometric theory of diffraction, and physical optics. Finite difference time domain (FDTD) technique is applicable to the widest range of problems.

Finite – differencing was introduced by Yee in the 1966 as an efficient way of solving Maxwell’s time-dependent curl equations. His method involved sampling a continuous electromagnetic field in a finite region at equidistant points in a spatial lattice, and also at equidistant time intervals. Spatial and time intervals have been chosen to avoid aliasing and to provide stability for the time – marching system. The propagation of waves from a source, assumed to be turned on at time $t = 0$, is computed at each of the spatial lattice points by using the finite difference equations to march forward in time. This prosess continues until a desired final state has been reached (usually the steady state). This method has been demonstrated to be accurate for solving for hundreds of thousands of field unknowns in a relatively efficient manner on a vector – processing computer.

There are seven primary reasons for the expansion of interest in FDTD and related computational solution approaches for Maxwell’s equations:

1. FDTD uses no linear algebra. Being a fully explicit computation, FDTD avoids the difficulties with linear algebra that limit the size of frequency – domain integral – equation and finite – element electromagnetic models to generally fewer than 10^6 electromagnetic field unknowns. FDTD models with as many as 10^9 field unknowns have been run; there is no intrinsic upper bound to this number.

2. FDTD is accurate and robust. The sources of error in FDTD calculations are well understood and can be bounded to permit accurate models for a very large variety of electromagnetic wave interaction problems.

3. FDTD treats impulsive behavior naturally. Being a time – domain technique, FDTD directly calculates the impulse response of an electromagnetic

system. Therefore, a single FDTD simulation can provide either ultrawideband temporal waveforms or the sinusoidal steady – state response at any frequency within the excitation spectrum.

4. FDTD treats nonlinear behavior naturally. Being a time – domain technique, FDTD directly calculates the nonlinear response of an electromagnetic system.

5. FDTD is a systematic approach. With FDTD, specifying a new structure to be modeled is reduced to a problem of mesh generation rather than the potentially complex reformulation of an integral equation. For example, FDTD requires no calculation of structure – dependent Green's functions.

6. Computer memory capacities are increasing rapidly. While this trend positively influences all numerical techniques, it is of particular advantage to FDTD methods which are founded on discretizing space over a volume, and therefore inherently require a large random access memory.

7. Computer visualization capabilities are increasing rapidly. While this trend positively influences all numerical techniques, it is of particular advantage to FDTD methods which generate time – marched arrays of field quantities suitable for use in color videos to illustrate the field dynamics.

We can begin to develop an appreciation of the basis, technical development, and possible future of FDTD numerical techniques for Maxwell's equations by first considering their history.

In 1966 Yee described the basis of the FDTD method for solving Maxwell's equations directly in time domain, using a set of finite difference equations for the system of partial differential equations. In this study, the obstacle is considered as a perfectly conducting square, and the scattering of an incoming pulse is determined for TM case at different grid space and at different time steps. This problem was a two – dimensional problem due to the consideration of the independence of the field components from z-coordinate. This assumption facilitate the solution, but is not failed the generality of the method.

In 1980 Taflove used FDTD method for predicting the sinusoidal steady – state electromagnetic fields penetrating an arbitrary dielectric or conducting body. He introduced a small air-dielectric loss factor to improve the lattice truncation conditions and to accelerate the convergence of cavity interior fields to the sinusoidal steady – state. His result was in agreement with those, obtained using classical moment method for a dielectric sphere and a cylindrical metal cavity.

In 1982 Taflove and Umashankar developed a hybrid moment method / FDTD approach to electromagnetic coupling and aperture penetration into complex geometries. The method was based on the determination of an equivalent short – circuit current excitation in the aperture region of the structure using moment method for a given external illumination firstly. Then, the computed equivalent current excitation over the aperture was used to excite the complex loaded interior region, and the penetrating fields and induced current were computed by FDTD method. The significant advantage of this hybrid method was that no Green's function need be calculated for the interior region.

In 1983 Taflove and Umashankar determined the induced surface current and radar cross section for the three – dimensional canonical case of a conducting metal cube illuminated by a plane wave , using moment method and FDTD.

In 1987 Kriegsmann and Taflove developed a new approach for scattering problems named as On – Surface Radiation Boundary Condition Approach. They showed that application of a suitable radiation condition directly on the surface of a conducting scatterer, instead of the application of the radiation condition at some distance from the scatterer, can lead to substantial simplification of the frequency – domain integral equation.

In 1988 Sullivan et al. used the FDTD method for calculating electromagnetic absorption rate (SAR) for the frequencies of 100 and 350 MHz for three different cases: a homogeneous man model in free space, an in homogeneous man model in free space, an inhomogeneous man model in free space and an inhomogeneous man model standing on a ground plane. With respect to their results, the most realistic model was the inhomogeneous man model on a ground plane.

In 1991 Katz et al. analyzed the electromagnetic wave radiation from system containing horn antennas. This was the first application of FDTD in the systems involving a radiating object.

In 1990 Shibata and Sano carried out the analysis of metal-insulator-semiconductor (M) is structure transmission lines on doped semiconductor substrates by FDTD method. In this study they took metal conduction loss into account and obtained the line parameters as well as electromagnetic field distributions. This was one of the first application of FDTD in circuits including microstrip lines.

In 1992 Beggs et al. used the surface impedance boundary conditions for the analysis of scattering from lossy - dielectric objects, in FDTD formulation. As a result they reduced the solution volume by ignoring the analysis in lossy dielectric scatterer.

In 1994, Thomas et al. introduced a Norton's equivalent circuit for the FDTD space lattice which permits the SPICE circuit analysis tool to implement accurate sub-grid models of nonlinear electronic components or complete circuits embedded within the lattice.

In 1995, Gedney and Lansing introduced the planar-generalized Yee algorithm which permits efficient unstructured-grid FDTD modeling of microwave and digital circuits.

In 1998 Popovic et al. use FDTD to determine the SAR of the liquid biological media for determining the exposure standards for wireless communications.

In 1996 Dunn and Kortum computed the 3-dimensional light scattering from cells containing multiple organelles. They found the scattering cross section of a cell as a function of volume fraction of melanin granules and mitochondria. Mean that, the scattering cross section can be used to identify the content of the cells for diagnostic purposes.

In 1998 Maloney and Kesler, developed an FDTD based method to analysis the antenna arrays. They modeled a single antenna element and used the periodic boundary condition to represent the array geometry.

In 1999, Schneider and Wagner introduced a comprehensive analysis of FDTD grid dispersion based upon complex wavenumbers.

In 1999, Painter et al. used FDTD to design, construct, and successfully test the world's smallest microcavity laser based upon a two-dimensional photonic bandgap structure.

In 2001 Eleiwa and Elsherbeni, for accurate FDTD simulation of biological tissues, a numerical technique is proposed to derive the Debye coefficients from the measured frequency – dependent permittivity of different biological tissues, and then a scattered field FDTD formulation is developed for such dispersive media characterized by multi – term Debye expressions.

In 2004 Neven Simicevic and Donald T Haynie have presented a series of results of FDTD calculations on nanopulse penetration of biological matter. Calculations included a detailed geometrical description of the material exposed to nanopulses, and a state-of-the-art description of the physical properties of the material. The length of a side of the Yee cell was set at ¼ mm, smaller than the value required by the cut-off frequency of 100 GHz in vacuum. To minimize computation time, the Cole-Cole parametrization of the dielectric properties of tissue in the frequency range ≤100 GHz was reformulated in terms of the Debye parametrization with no loss of accuracy of description. In two-dimensional FDTD, the decreased computation time enable comparison of different materials exposure nonopulse.

In 2005 Neven Simicevic have extended our previous FDTD calculations on nanopulse penetration into biological matter from two to three dimensions. Calculations included the same detailed geometrical description of the material exposed to nanopulses, the same accurate description of the physical properties of the material, the same spatial resolution of ¼ mm side length of the Yee cell and same cut-off frequency of ≈100 GHz in vacuum and ≈ 15 GHz in the dielectric. To minimize computation time, the dielectric properties of a tissue in the frequency range ≤ 100 GHz were formulated in terms Debye parametrization which we have shown in a previous paper to be, for the materials studied, as accurate as the Cole-Cole parametrization.

In our study, it is used FORTRAN for an FDTD computer code. The code is 3 – D. İt is capable of including perfect conductors and lossy dielectric materials. The scattering object is selected as an inhomogeneous sphere which has three layers. And each layer is composed of lossy dielectric with relative permittivity 4 and different conductivity. The material property of inner sphere is selected to be almost perfect conductor (conductivity 5 S/m) and the sphere has a radius of two cells. The material property of second layer is selected to be almost free space (conductivity 0.000005 S/m) and has a radius of five cells. The material property of third layer is selected to be lossy dielectric (conductivity 0.005 S/m) and the sphere has a radius of eight cells.

A Bandpass Gaussian pulse plane wave is incident on the sphere. The β value for the Bandpass Gaussian pulse is set at 64 to compensate for the shorter wavelenght inside the dielectric sphere. Β is the temporal width of a bandbass gaussian pulse specified in time steps.

The sphere is centered in a problem space. The problem space size is 34 by 34 by 34 cells in the x,y,z directions. Cell size : delx = 0.058824, delx = 0.058824, delx = 0.058824 meters. Time step is 0.113283×10^{-9} seconds. And the incident bandpass gaussian pulse amplitude is selected 1 V/m. Decay factor alpha is $0.304 \times 10^{+18}$.

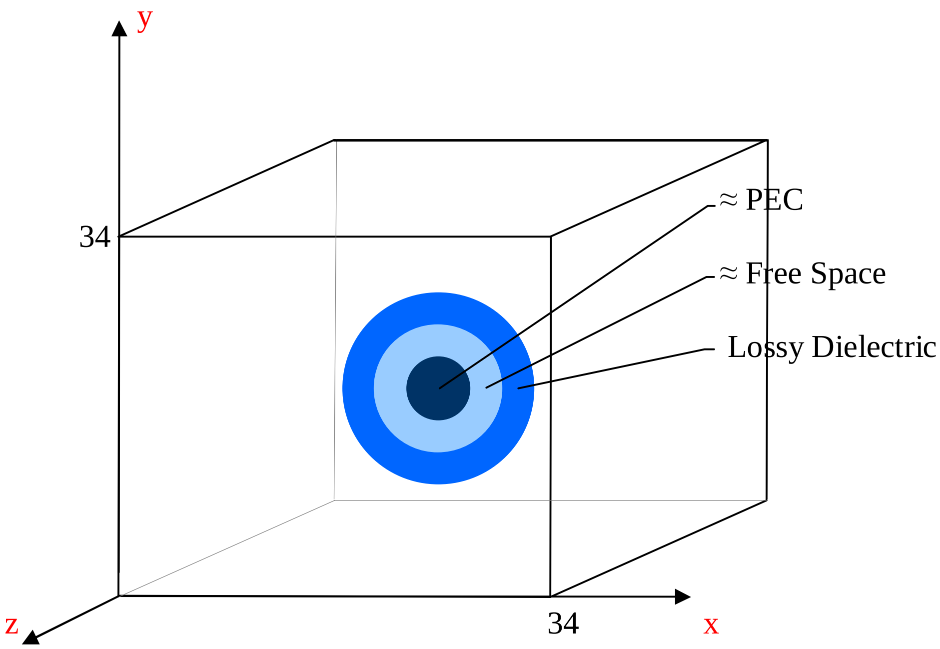

Figure 1.1. The geometry of the problem

2. FORMULATION OF THE PROBLEM

2.1. Finite Differences

Consider a function u(x, t) about a space point x_i, and keeping time fixed at t_n. The Taylor's series expansion of u(x, t_n) from the space point x_i to $x_i+\Delta x$ will be,

$$u(x_i+\Delta x)\Big|_{t_n} = u\Big|_{x_i,t_n} + \Delta x.\frac{\partial u}{\partial x}\Big|_{x_i,t} + \frac{(\Delta x)^2}{2}\cdot\frac{\partial^2 u}{\partial x^2}\Big|_{x_i,t_n} + \frac{(\Delta x)^3}{6}\cdot\frac{\partial^3 u}{\partial x^3}\Big|_{x_i,t_n} + \frac{(\Delta x)^4}{24}\cdot\frac{\partial^4 u}{\partial x^4}\Big|_{z_1,t_n} \quad (2.1)$$

Where z_1 is a space point located somewhere in the interval ($x_i, x_i+\Delta x$) and the last term is the remainder or the error term.

Similarly the Taylor's series expansion from x_i to $x_i-\Delta x$, again keeping time fixed at t_n, will be

$$u(x_i-\Delta x)\Big|_{t_n} = u\Big|_{x_i,t_n} - \Delta x.\frac{\partial u}{\partial x}\Big|_{x_i,t} + \frac{(\Delta x)^2}{2}\cdot\frac{\partial^2 u}{\partial x^2}\Big|_{x_i,t_n} - \frac{(\Delta x)^3}{6}\cdot\frac{\partial^3 u}{\partial x^3}\Big|_{x_i,t_n} + \frac{(\Delta x)^4}{24}\cdot\frac{\partial^4 u}{\partial x^4}\Big|_{z_2,t_n} \quad (2.2)$$

Where z_2 is a space point located somewhere in the interval $(x_i, x_i-\Delta x)$ and again the last term is named as remainder or error term.

Adding (2.1) and (2.2) we obtain;

$$\mathrm{u}(\mathrm{x_i}+\Delta \mathrm{x})\Big|_{\mathrm{t_n}} + \mathrm{u}(\mathrm{x_i}-\Delta \mathrm{x})\Big|_{\mathrm{t_n}} = 2\mathrm{u}\Big|_{\mathrm{x_i},\mathrm{t_n}} + (\Delta \mathrm{x})^2\cdot\frac{\partial^2 \mathrm{u}}{\partial \mathrm{x}^2}\Big|_{\mathrm{x_i},\mathrm{t_n}} + \frac{(\Delta \mathrm{x})^4}{12}\cdot\frac{\partial^4 \mathrm{u}}{\partial \mathrm{x}^4}\Big|_{\zeta_3,\mathrm{t_n}} \quad (2.3)$$

Here z_3 a space point located somewhere in the interval $(x_i-\Delta x, x_i+\Delta x)$. Solving (2.3) in terms of the second derivative of the function $u(x,t)$, we can obtain that,

$$\frac{\partial^2 u}{\partial x^2}\Big|_{x_i,t_n} = [\frac{u(x_i+\Delta x) - 2u(x_i) + u(x_i - \Delta x)}{(\Delta x)^2}]\Big|_{t_n} + O[(\Delta x)^2] \tag{2.4}$$

Where $O[(\Delta x)^2]$ is a short hand notation for the remainder term, and it approach zero as the square of the space increment. Equation (2.4) is commonly referred to as a second order accurate, central-difference approximation to the second partial space derivative of $u(x,t)$. For convenience, we can use a subscript i for the space position and a superscript n for the time observation point. Then (2.4) can be expressed as

$$\frac{\partial^2 u}{\partial x^2}\Big|_{x_i,t_n} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + O[(\Delta x)^2] \tag{2.5}$$

Here u_i^n denotes a field quantity calculated at the space point $i\Delta x$ and time point $t_n = n\Delta t$.

Using the same procedure described above, the second partial time derivative of $u(x,t)$, keeping x_i fixed is obtained as follows:

$$\frac{\partial^2 u}{\partial t^2}\Big|_{x_i,t_n} = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{(\Delta t)^2} + O[(\Delta t)^2] \tag{2.6}$$

Equation (2.6) is named as the second-order accurate, central-difference approximation to the second partial time derivative of $u(x,t)$.

2.2. Maxwell's Equations in Three Dimensions

Let as consider a region of space that has no electric or magnetic current sources, but may have materials that absorb electric or magnetic field energy. Then the time dependent Maxwell's equations in differential and in integral forms are given as followings:

Faraday's Law:

$$\frac{\partial \vec{B}}{\partial t} = -\nabla x\vec{E} - \vec{M} \tag{2.7}$$

$$\frac{\partial}{\partial t}\int_a \vec{B}.d\vec{a} = -\oint_l \vec{E}.d\vec{l} - \int_a \vec{M}.d\vec{a} \tag{2.8}$$

Ampere's Law:

$$\frac{\partial \vec{D}}{\partial t} = \nabla x\vec{H} - \vec{J} \tag{2.9}$$

$$\frac{\partial}{\partial t}\int_a \vec{D}.d\vec{a} = \oint_l \vec{H}.d\vec{l} - \int_a \vec{J}.d\vec{a} \tag{2.10}$$

Gauss's Law for the electric field:

$$\nabla.\vec{D} = 0 \tag{2.11}$$

$$\oint_a \vec{D}.d\vec{a} = 0 \tag{2.12}$$

Gauss's Law for the magnetic field:

$$\nabla.\vec{B} = 0 \tag{2.13}$$

$$\oint_a \vec{B}.d\vec{a} = 0 \tag{2.14}$$

where

$\vec{E}$: electric f ield intensity vector (V/m)

$\vec{D}$: electric flux density vector (C/m^2)

$\vec{H}$: magnetic field intensity vector (A/m)

$\vec{B}$: magnetic flux density vector (W/m^2)

a : arbitrary three-dimensional surface

$d\vec{a}$: differential normal vector

$\mathbf{l}$: closed contour that bounds the surface a

$d\vec{\mathbf{l}}$: differential length vector

$\vec{J}$: electric current density vector (A/m^2)

$\vec{M}$: equivalent magnetic current density vector (V/m^2)

For linear, isotropic and nondispersive materials we can relate $\vec{D}$ to $\vec{E}$ and $\vec{B}$ to $\vec{H}$ using

$$\vec{D} = \varepsilon\vec{E} = \varepsilon_o \varepsilon_r \vec{E} \tag{2.15}$$

$$\vec{B} = \mu\vec{H} = \mu_O \mu_r \vec{H} \tag{2.16}$$

where

ε : electrical permittivity (F/m)

ε_r : relative permittivity

ε_o : free-space permittivity (8.854x10^{-12} F/m)

μ : magnetic permeability (H/m)

μ_r : relative permeability

μ_o : free-space permeability (4π x10^{-7} H/m)

$\vec{J}$ and $\vec{M}$ are the independent sources of electric and magnetic field. Denoting these as $\vec{J}_{source}$ and $\vec{M}_{source}$, and considering the lossy materials which attenuate $\vec{E}$ and $\vec{H}$ fields via conversion to heat energy, we can write that

$$\vec{J} = \vec{J}_{source} + s\vec{E} \tag{2.17}$$

and

$$\vec{M} = \vec{M}_{source} + s^{*}\vec{H} \tag{2.18}$$

where,

s :electric conductivity (S/m)

s^{*} equivalent magnetic Loss (Ω/m)

Then substituting (2.11) and (2.12) into (2.7) and (2.8) we get,

$$\frac{\partial \vec{H}}{\partial t} = -\frac{1}{m}\nabla x\vec{E} - \frac{1}{m}(\vec{M}_{source} + s * \vec{H}) \tag{2.19}$$

and

$$\frac{\partial \vec{E}}{\partial t} = \frac{1}{e}\nabla x\vec{H} - \frac{1}{e}(\vec{J}_{source} + s\vec{E}) \tag{2.20}$$

In terms of the field components in Cartesian coordinates the following system of six coupled equations are obtaining from (2.13) and (2.14) :

$$\frac{\partial H_x}{\partial t} = \frac{1}{m}[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - (M_{source_x} + s^{*}H_x)] \tag{2.21}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{m}[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - (M_{source_y} + s^* H_y)] \tag{2.22}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{m}[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - (M_{source_z} + s^* H_z)] \tag{2.23}$$

and

$$\frac{\partial E_x}{\partial t} = \frac{1}{e}[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - (\bar{J}_{source_x} + sE_x)] \tag{2.24}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{e}[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - (\bar{J}_{source_y} + sE_y)] \tag{2.25}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{e}[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - (\bar{J}_{source_z} + sE_z)] \tag{2.26}$$

This system of six coupled partial differential equations forms the basis of the FDTD algorithm for electromagnetic wave interactions with three dimensional objects.

2.3. The YEE Algorithm

2.3.1. Basic Ideas

In 1966, Kane Yee originated a set of finite-difference equations fort Maxwell’s the time-dependent curl equation system of (2.15) and (2.16) for the lossless materials case s^* = and s =0 .

1. The Yee algorithm solves for both electric and magnetic fields in time and space using the coupled Maxwell’s curl equations rather than solving for the electric field alone (or the magnetic field alone) with a wave equation.

- This is analogous to the combined-field integral equation formulation of MM, where both $\vec{E}$ and $\vec{H}$ boundary conditions are enforced on the surface of a material structure.
- Using both $\vec{E}$ and $\vec{H}$ information, the solution is more robust than using either alone (i.e., it is accurate for a wider class of structures). Both electric and magnetic material properties can be modeled in a straightforward manner. This is especially important when modeling radar cross section mitigation.
- Features unique to each field such as tangential $\vec{H}$ singularities near edges and corners, azimuthal (looping) $\vec{H}$ singularities near thin wires, and radial $\vec{E}$ singularities near points, edges, and thin wires can be individually modeled if both electric and magnetic fields are available.

2. As illustrated in Fig. 2.1, the Yee algorithm centers its $\vec{E}$ and $\vec{H}$ components in three-dimensional space so that every $\vec{E}$ components is surrounded by four circulating $\vec{E}$ components.

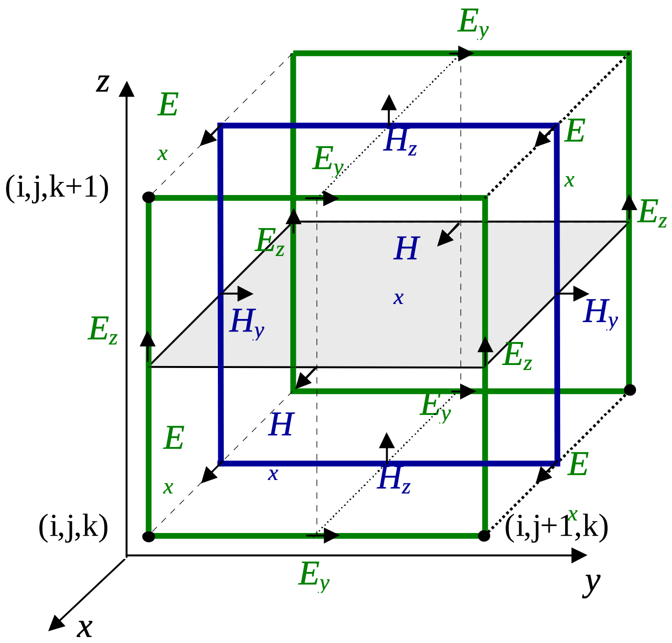

Figure 2.1. Position of the electric and magnetic field vector components about a cubic unit cell of the Yee space lattice.

This provides a beautiful simple picture of three-dimensional space being filled by an interlinked array of Faraday's Law and Ampere's Law contours. For example, it is possible to identify Yee $\vec{E}$ components associated with displacement current flux linking $\vec{E}$ loops. In effect, the Yee algorithm simultaneously simulates the pointwise differential form and macroscopic integral form of Maxwell's equation. The latter is extremely useful in specifying field boundary conditions and singularities.

In addition, we have the following attributes of the Yee space lattice:

- The finite-difference expressions for the space derivatives used in the curl operators are central-difference in nature and second-order accurate.
- Continuity of tangential $\vec{E}$ and $\vec{H}$ is naturally maintained across an interface of dissimilar materials if the interface is parallel to one of the lattice coordinate axes. For this case, there is no need to specially enforce field boundary conditions at the interface. At the beginning of the problem, we simply specify the material permittivity and permeability at each field component location. This yields a stepped or "staircase" approximation of the surface and internal geometry of the structure, with a space resolution set by the size of the lattice unit cell.
- The location of the $\vec{E}$ and $\vec{H}$ components in the Yee space lattice and the central-difference operations on these components implicitly enforce the two Gauss' Law relation. Thus, the Yee mesh is divergence-free with respect to its $\vec{E}$ and $\vec{H}$ fields in the absence of free electric and magnetic charge.

3. As illustrated in Fig. 2.2, the Yee algorithm also centers its $\vec{E}$ and $\vec{H}$ components in time in what is termed a leapfrog arrangement. All of the $\vec{E}$ computations in the modeled space are completed and stored in memory for a particular time point using previously stored $\vec{H}$ data. Then all of the $\vec{H}$ computations in the space are completed and stored in memory using the

$\vec{E}$ data just computed. The cycle beings again with the recomputation of the $\vec{E}$ components based on the newly obtained $\vec{H}$. This process continues until time-stepping is concluded.

§ Leapfrog time-stepping is fully explicit, thereby avoiding problems involved with simultaneous equation and matrix inversion.

§ The finite-difference expressions for the time derivatives are central difference in nature and second-order accurate.

§ The time-stepping algorithm is nondissipative. That is, numerical wave modes propagating in the mesh do not spuriously decay due to a nonphysical artifact of the time-stepping algorithm.

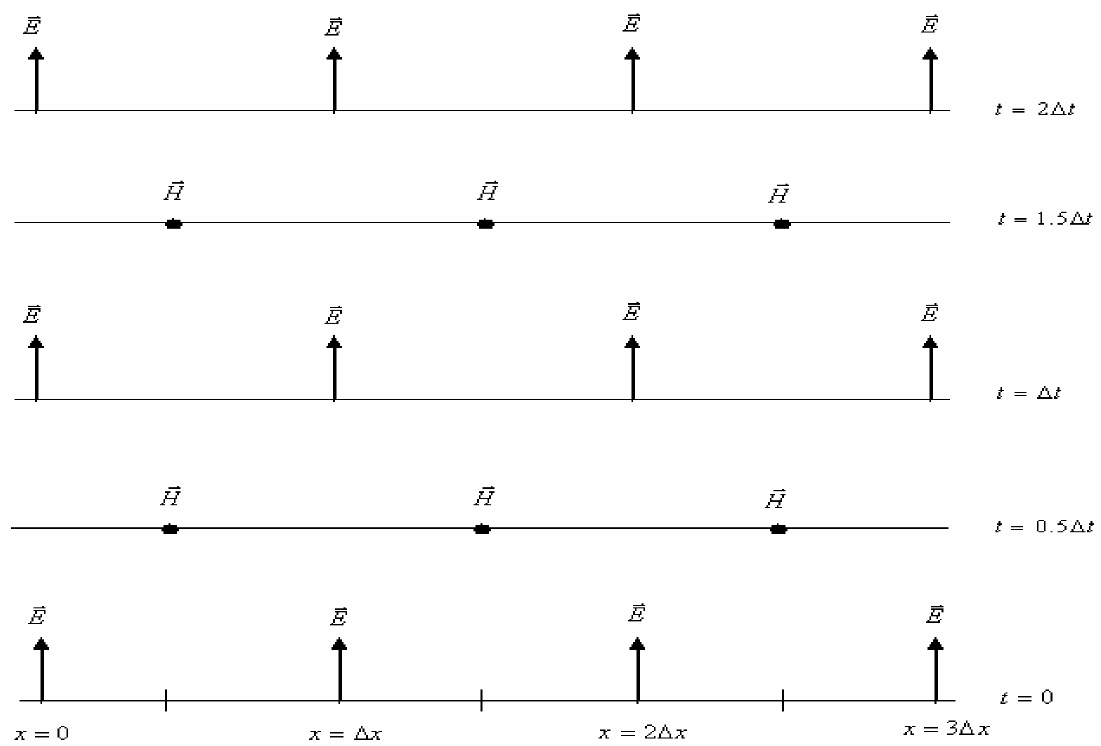

Figure 2.2. Space-time chart of the Yee algorithm for a one-dimensional wave propagation example showing the use of central differences for the space derivatives and leapfrog for the time derivatives. Initial conditions for both electric and magnetic fields are zero everywhere in the grid.

2.4. Finite Differences and Notation

Yee introduced the notation used in section 2.1 (there, in one spatial dimension) for space points and functions of space and time. For convenience, this notation is repeated here and generalized to three spatial dimensions. We denote a space point in a uniform, rectangular lattice as

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z) \tag{2.27}$$

Here, Δx, Δy, Δz are, respectively, the lattice space increments in the x, y and z coordinate directions, and i, j and k are integers. Further, we denote any function u of space and time evaluated at a discrete point in the grid and at a discrete point in time as

$$u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = u^{n}_{i,j,k} \tag{2.28}$$

where Δt is the time increment, assumed uniform over the observation interval, and n is an integer.

Yee used centered finite difference (central difference) expressions for the space and time derivatives that are both simply programmed and second order accurate in the space and time increments. Consider his expression for first partial space derivative of *u* in the *x*-direction, evaluated at the fixed time $t_n = n\Delta t$:

$$\frac{\partial\ u}{\partial x}(i\Delta x, \mathrm{j}\Delta y, \mathrm{k}\Delta\mathrm{z}, \mathrm{n}\Delta\mathrm{t}) = [\frac{u^{n}_{i+1/2,j,k} - u^{n}_{i-1/2,j,k} + u^{n}_{i-1/2,j,k}}{\Delta x}]\Big|_{t_n} + O[(\Delta x)^2] \tag{2.29}$$

We note the $\mp 1/2$ increment in the *i* subscript (*x*-coordinate) of *u*, denoting a space finite-difference over $\mp 1/2\Delta x$.

Yee choose this notation because he wished to interleave his $\vec{E}$ and $\vec{H}$ components in the space lattice at intervals of $\Delta x/2$.

Yee's expression for the first time partial derivative of *u*, evaluated at the fixed space point (i, j, k), follows by analogy:

$$\frac{\partial\ u}{\partial t}(i\Delta x, \mathrm{j}\Delta y, \mathrm{k}\Delta\mathrm{z}, \mathrm{n}\Delta\mathrm{t}) = [\frac{u^{n+1/2}_{i,j,k} - u^{n-1/2}_{i,j,k}}{\Delta t}]\Big|_{t_n} + O[(\Delta x)^2] \tag{2.30}$$

Now the $\mp 1/2$ increment is in the *n* superscript (time-coordinate) of u, denoting a time finite-difference over $\mp 1/2\Delta x$. Yee choose this notation because he wished to interleave his $\vec{E}$ and $\vec{H}$ components in the time at intervals of $1/2\Delta t$ for purposes of implementing a leapfrog algorithm.

2.4. Finite-Difference Expressions for Maxwell's Equations in Three Dimensions

For lossy dielectric media:

We now apply the above ideas and notation to achieve a numerical approximation of the Maxwell's curl equation in three dimensions given by the system of equations of (2.21 - 2.26). We begin by considering the E_x field-component equation (2.24), repeated here for convenience:

$$\frac{\partial E_x}{\partial t} = \frac{1}{e}\left[\frac{\partial H_z}{\partial_y} - \frac{\partial H_y}{\partial_z} - \left(J_{source_x} + sE_x\right)\right] \tag{2.31}$$

Referring to Fig. 2.1, consider a typical substitution of central differences for the time and space derivatives in (2.24), for example at E_x(i, j+1/2, k+1/2, n). Here, we have initially

$$\frac{E_x\big|_{i,j+1/2,k+1/2}^{n+1/2} - E_x\big|_{i,j+1/2,k+1/2}^{n-1/2}}{\Delta t} = \frac{1}{e_{i,j+1/2,k+1/2}} \cdot \left[\begin{array}{l} \dfrac{H_z\big|_{i,j+1/2,k+1/2}^{n} - H_z\big|_{i,j,k+1/2}^{n}}{\Delta y} - \dfrac{H_{y_{i,j+1/2,k+1}} - H_y\big|_{i,j+1/2,k}^{n}}{\Delta z} \\ - J_{source_x}\big|_{i,j+1/2,k+1/2} - s_{\,i,j+1/2,k+1/2}\, E_x\big|_{i,j+1/2,k+1/2}^{n} \end{array}\right] \tag{2.32}$$

Note that all field quantities on the right-hand side are evaluated at time-step n, including the electric field E_x appearing due to material conductivity s . Since Ex values at time-step n are not assumed to be stored in the computer's memory (only the previous values of E_x at time-step n-1/2 are assumed to be in memory), we need some way to estimate such terms. A very good way is as follows, using what we call a semi-implicit approximation:

$$E_x\Big|_{i,j+1/2,k+1/2}^{n} = \frac{E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2} + E_x\Big|_{i,j+1/2,k+1/2}^{n-1/2}}{2} \tag{2.33}$$

Here E_x values at time-step n are assumed to be simply the arithmetic average of the stored values of E_x at time-step n-1/2 and the yet –to-be computed new values of E_x at time-step n+1/2. Substituting (2.33) into (2.32) after multiplying both sides by Δt, we obtain

$$\frac{E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2} - E_x\Big|_{i,j+1/2,k+1/2}^{n-1/2}}{\Delta t} = \frac{1}{e_{i,j+1/2,k+1/2}} \cdot \left[\begin{array}{l} \dfrac{H_z\Big|_{i,j+1/2,k+1/2}^{n} - H_z\Big|_{i,j,k+1/2}^{n}}{\Delta y} - \dfrac{H_{y\,i,j+1/2,k+1} - H_y\Big|_{i,j+1/2,k}^{n}}{\Delta z} \\ - J_{source_x}\Big|_{i,j+1/2,k+1/2} - s_{\,i,j+1/2,k+1/2} E_x\Big|_{i,j+1/2,k+1/2}^{n} \end{array} \right] \tag{2.34}$$

We note that the terms $E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2}$ and $E_x\Big|_{i,j+1/2,k+1/2}^{n-1/2}$ appear on both sides of (2.34). Collecting all terms of these two types and isolating $E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2}$ on the left-hand side yields

$$\left(1 + \frac{s_{\,i,j+1/2,k+1/2}\Delta t}{2e_{i,j+1/2,k+1/2}}\right) E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2} = \left(1 - \frac{s_{\,i,j+1/2,k+1/2}\Delta t}{2e_{i,j+1/2,k+1/2}}\right) E_x\Big|_{i,j+1/2,k+1/2}^{n-1/2} + \frac{\Delta t}{e_{i,j+1/2,k+1/2}} \cdot$$
$$\left(\frac{H_z\Big|_{i,j+1,k+1/2}^{n} - H_z\Big|_{i,j,k+1/2}^{n}}{\Delta y} - \frac{H_y\Big|_{i,j+1/2,k+1}^{n} - H_y\Big|_{i,j+1/2,k}^{n}}{\Delta z} - J_{source_x}\Big|_{i,j+1/2,k+1/2}^{n} \right) \tag{2.35}$$

Dividing both sides by $\left(1 + s_{\,i,j+1/2,k+1/2}\Delta t / 2e_{i,j+1/2,k+1/2}\right)$ yields the desired explicit time-stepping relation for $E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2}$:

$$E_x\Big|_{i,j+1/2,k+1/2}^{n+1/2} = \left(\frac{1 - \dfrac{s_{i,j+1/2,k+1/2}\Delta t}{2e_{i,j+1/2,k+1/2}}}{1 + \dfrac{s_{i,j+1/2,k+1/2}\Delta t}{2e_{i,j+1/2,k+1/2}}} \right) E_x\Big|_{i,j+1/2,k+1/2}^{n-1/2} + \left(\frac{\dfrac{\Delta t}{e_{i,j+1/2,k+1/2}}}{1 + \dfrac{s_{i,j+1/2,k+1/2}\Delta t}{2e_{i,j+1/2,k+1/2}}} \right)$$
$$\cdot \left(\frac{H_z\Big|_{i,j+1,k+1/2}^{n} - H_z\Big|_{i,j,k+1/2}^{n}}{\Delta y} \right) - \frac{H_y\Big|_{i,j+1/2,k}^{n} - H_y\Big|_{i,j+1/2,k}^{n}}{\Delta z} - J_{source_x}\Big|_{i,j+1/2,k+1/2}^{n} \quad (2.36)$$

The semi-implicit assumption of (2.33) has been found to yield numerically stable and accurate results for values of s from zero to infinity. As we have seen above, this assumption fortunately allows us to avoid simultaneous equations for $E_x\Big|^{n+1/2}$. The term of this type introduced on the right –hand side of (2.34) can be grouped with a like term on the left-hand side and then solved explicitly.

Similarly, we can derive finite-difference expressions based on Yee's algorithm for the E_y and E_z field components given by Maxwell's equations (2.25) and (2.26). Referring again to Fig. 2.1, we have for example the following time-stepping expressions for the E components for the E components normal to the remaining visible faces of the unit cell:

$$E_y\Big|_{i-1/2,j+1,k+1/2}^{n+1/2} = \left(\frac{1 - \dfrac{s_{i-1/2,j+1,k+1/2}\Delta t}{2e_{i-1/2,j+1,k+1/2}}}{1 + \dfrac{s_{i-1/2,j+1,k+1/2}\Delta t}{2e_{i-1/2,j+1,k+1/2}}} \right) E_x\Big|_{i-1/2,j+1,k+1/2}^{n-1/2} + \left(\frac{\dfrac{\Delta t}{e_{i-1/2,j+1,k+1/2}}}{1 + \dfrac{s_{i-1/2,j+1,k+1/2}\Delta t}{2e_{i-1/2,j+1,k+1/2}}} \right)$$
$$\cdot \left(\frac{H_z\Big|_{i-1/2,j+1,k+1}^{n} - H_z\Big|_{i-1/2,j+1,k}^{n}}{\Delta z} \right) - \frac{H_z\Big|_{i,j+1,k+1/2}^{n} - H_z\Big|_{i-1,j+1,k+1/2}^{n}}{\Delta x} - J_{source_y}\Big|_{i-1/2,j+1,k+1/2}^{n} \quad (2.37)$$

$$E_z\Big|_{i-1/2,\,j+1/2,k+1}^{n+1/2}=\left(\frac{1-\dfrac{s_{\,i-1/2,\,j+1/2,k+1}\Delta t}{2e_{\,i-1/2,\,j+1/2,k+1}}}{1+\dfrac{s_{\,i-1/2,\,j+1/2,k+1}\Delta t}{2e_{\,i-1/2,\,j+1/2,k+1}}}\right)E_z\Big|_{i-1/2,\,j+1/2,k+1}^{n-1/2}+\left(\frac{\dfrac{\Delta t}{e_{\,i-1/2,\,j+1/2,k+1}}}{1+\dfrac{s_{\,i-1/2,\,j+1/2,k+1}\Delta t}{2e_{\,i-1/2,\,j+1/2,k+1}}}\right)$$
$$\cdot\left(\frac{H_y\Big|_{i,\,j+1/2,k+1}^{n}-H_y\Big|_{i-1,\,j+1/2,k+1}^{n}}{\Delta x}\right)-\frac{H_x\Big|_{i-1/2,\,j+1,k+1}^{n}-H_x\Big|_{i-1/2,\,j,k+1}^{n}}{\Delta y}-J_{source_z}\Big|_{i-1/2,\,j+1/2,k+1}^{n}\quad(2.38)$$

By analogy we can derive finite-difference equations for (2.21)-(2.23) to time-step H_x, H_y and H_z. Here s^*H represents a magnetic loss term on the right-hand side of each equation, which is estimated using a semi-implicit procedure analogous to (2.34). This results in three equations having a form similar to that of the E equations above. Referring again to Fig. 2.1, we have for example the following time-stepping expression for the H_x component located at the upper right corner of the unit cell:

$$H_x\Big|_{i-1/2,\,j+1,k+1}^{n+1}=\left(\frac{1-\dfrac{s^*_{\,i-1/2,\,j+1/2,k+1}\Delta t}{2m_{i-1/2,\,j+1/2,k+1}}}{1+\dfrac{s^*_{\,i-1/2,\,j+1/2,k+1}\Delta t}{2m_{i-1/2,\,j+1/2,k+1}}}\right)H_x\Big|_{i-1/2,\,j+1,k+1}^{n}+\left(\frac{\dfrac{\Delta t}{m_{i-1/2,\,j+1,k+1}}}{1+\dfrac{s^*_{\,i-1/2,\,j+1,k+1}\Delta t}{2m_{i-1/2,\,j+1,k+1}}}\right)$$
$$\cdot\left(\frac{E_y\Big|_{i-1/2,\,j+1,k+3/2}^{n+1/2}-E_y\Big|_{i-1,\,j+1,k+1/2}^{n+1/2}}{\Delta z}-\frac{E_z\Big|_{i-1/2,\,j+3/2,k+1}^{n+1/2}-E_z\Big|_{i-1/2,\,j+1/2,k+1}^{n+1/2}}{\Delta y}-\mathrm{M}_{source_x}\Big|_{i-1/2,\,j+1,k+1}^{n+1/2}\right)\quad(2.39)$$

Similarly, we have the following time-stepping expression for the H_y component located at the upper front corner of the unit cell:

$$H_y\Big|_{i,\,j+1/2,k+1}^{n+1}=\left(\frac{1-\dfrac{s^*_{\,i,\,j+1/2,k+1}\Delta t}{2m_{i,\,j+1/2,k+1}}}{1+\dfrac{s^*_{\,i,\,j+1/2,k+1}\Delta t}{2m_{i,\,j+1/2,k+1}}}\right)H_y\Big|_{i,\,j+1/2,k+1}^{n}+\left(\frac{\dfrac{\Delta t}{m_{i,\,j+1/2,k+1}}}{1+\dfrac{s^*_{\,i,\,j+1/2,k+1}\Delta t}{2m_{i,\,j+1/2,k+1}}}\right)$$

$$\cdot\left(\frac{E_z\big|_{i+1/2,j+1/2,k+1}^{n+1/2}-E_z\big|_{i-1,j+1/2,k+1}^{n+1/2}}{\Delta x}-\frac{E_x\big|_{i,j+1/2,k+3/2}^{n+1/2}-E_z\big|_{i,j+1/2,k+1/2}^{n+1/2}}{\Delta z}-\mathrm{M}_{source_y}\Big|_{i,j+1/2,k+1}^{n+1/2}\right) \quad (2.40)$$

Finally we have the following time-stepping expression for the H_z component located at the right front corner of the unit cell:

$$H_z\big|_{i,j+1,k+1/2}^{n+1}=\left(\frac{1-\dfrac{s^*{}_{i,j+1,k+1/2}\Delta t}{2m_{i,j+1k+1/2}}}{1+\dfrac{s^*{}_{i,j+1/2,k+1}\Delta t}{2m_{i,j+1,k+1/2}}}\right)H_z\big|_{i,j+1,k+1/2}^{n}+\left(\frac{\dfrac{\Delta t}{m_{i,j+1,k+1/2}}}{1+\dfrac{s^*{}_{i,j+1,k+1/2}\Delta t}{2m_{i,j+1,k+1/2}}}\right)$$

$$\cdot\left(\frac{E_x\big|_{i,j+3/2,k+1/2}^{n+1/2}-E_x\big|_{i,j+1/2,k+1/2}^{n+1/2}}{\Delta y}-\frac{E_y\big|_{i,j+3/2,k+1/2}^{n+1/2}-E_y\big|_{i-1/2,j+1,k+1/2}^{n+1/2}}{\Delta x}-\mathrm{M}_{source_z}\Big|_{i,j+1,k+1/2}^{n+1/2}\right) \quad (2.41)$$

With the system of finite-difference of (2.36) - (2.41), the new value of an electromagnetic field vector component at any lattice point depends only on its previous value, the previous values of the components of the other field vector at adjacent points, and the known electric and magnetic current sources. Therefore, at any given time step, the computation of a field vector can proceed either one point at a time, or, if p parallel processors are employed concurrently, p points at a time.

2.6. Radiation Boundary Condition

An outer radiation boundary condition (ORBC) may not always be necessary when applying FDTD. If the FDTD problem space is bounded by a condition that can be implemented directly into the finite difference equations, an ORBC is not necessary. For example, if we are modeling electromagnetic phenomena inside a closed waveguide system, the tangential electric field at the walls is zero, and we implement this appropriately.

Consider that we are at the x = 0 limit of our FDTD computational space. We decide that on this plane we will locate E_y and E_z field components. Using these field components we can evaluate the finite difference curl operations needed to update the H_x magnetic field components at x = 0, and of course all nearest-neighbor field components will be available for updating the field components located at x = Δx/2 and beyond (at least to the maximum x dimension included in the problem space, at which location we must again apply an ORBC). However, we cannot update the E_y and E_z field components at x = 0 with the usual FDTD equations because the magnetic fields at x = -Δx/2 are not available.

We may update them, however, with the Mur expressions. Let us consider the Ez component located at x = 0, y = jΔy, and z = (k+1/2)Δz. The first order Mur estimate of this field component is

$$E_z^{n+1}(0,j,k+1/2) = E_z^{n}(1,j,k+1/2) + \frac{c\Delta t - \Delta x}{c\Delta t + \Delta x}\left(E_z^{n+1}(1,j,k+1/2) - E_z^{n}(0,j,k+1/2)\right) \quad (2.42)$$

The second order estimate for Ez at the boundary x = 0 is

$$\begin{aligned} E_z^{n+1}(0,j,k+1/2) = {} & E_z^{n-1}(1,j,k+1/2) + \frac{c\Delta t - \Delta x}{c\Delta t + \Delta x}\left(E_z^{n+1}(1,j,k+1/2) + E_z^{n-1}(0,j,k+1/2)\right) + \ldots \\ & \frac{2\Delta x}{c\Delta t + \Delta x}\left(E_z^{n}(0,j,k+1/2) + E_z^{n}(1,j,k+1/2)\right) + \frac{\Delta x (c\Delta t)^2}{2(\Delta y)^2 (c\Delta t + \Delta x)} + \ldots \\ & (E_z^{n}(0,j+1,k+1/2) - 2\,E_z^{n}(0,j,k+1/2) + E_z^{n}(0,j-1,k+1/2) + \ldots \\ & E_z^{n}(1,j+1,k+1/2) - 2E_z^{n}(1,j+1,k+1/2) + E_z^{n}(1,j-1,k+1/2)) + \\ & \frac{\Delta x (c\Delta t)^2}{2(\Delta y)^2 (c\Delta t + \Delta x)} \,.\, (E_z^{n}(0,j,k+3/2) - 2E_z^{n}(0,j,k+1/2) + E_z^{n}(0,j,k- \\ & 1/2) + \ldots\ E_z^{n}(1,j,k+3/2) - 2E_z^{n}(1,j,k+1/2) + E_z^{n}(1,j,k-1/2)) \end{aligned} \quad (2.43)$$

Considering the first order Mur approximation, we see that the current value of ez at x = 0 is estimated from the previous and current values at x = Δx and the same y and z positions. The second order estimate uses previous values from the preceding two time steps, and values at the adjacent y and z positions. The equations needed to determine other field components at other limiting surfaces of the FDTD space are readily determined by modification of (2.42) and (2.43).

3.NUMERICAL CONCLUSIONS

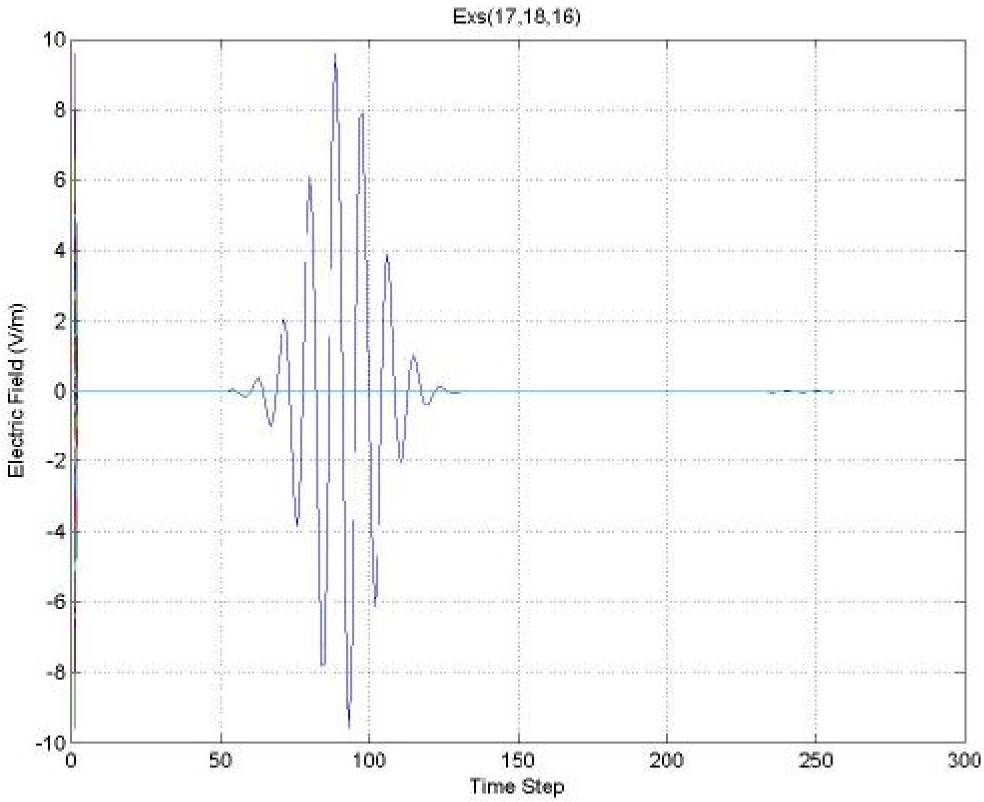

Figure 3.1. Electric field value versus time step in the Ex(17,18,16)

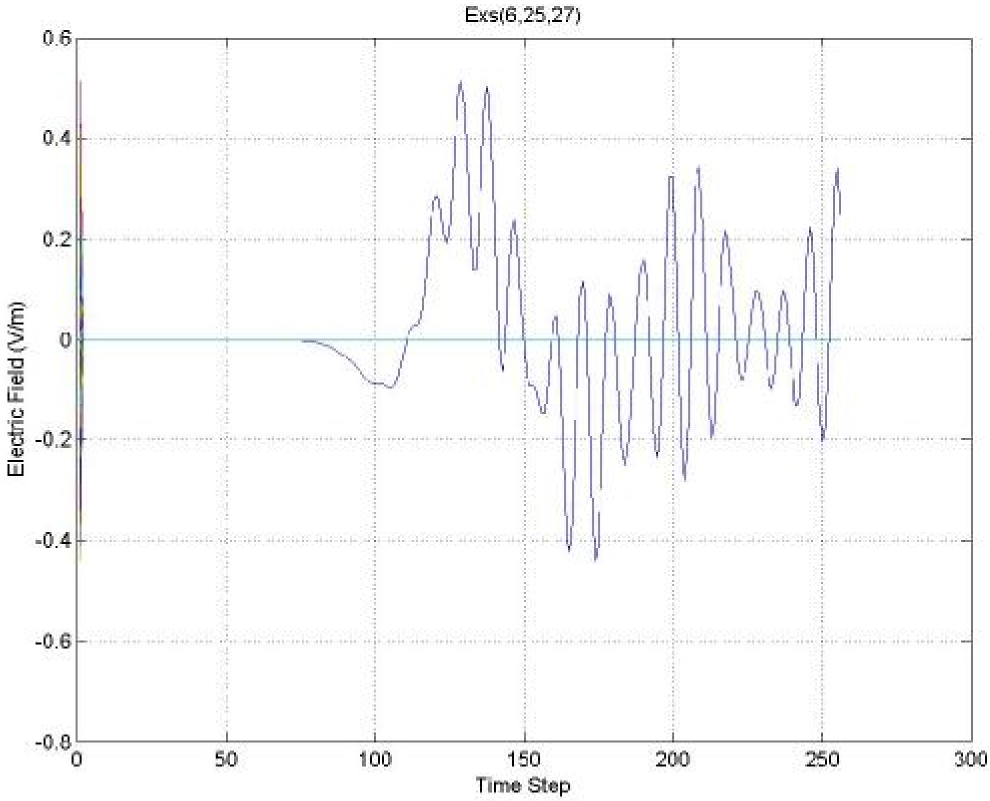

Figure 3.2. Electric field value versus time step in the Ex(6,25,27)

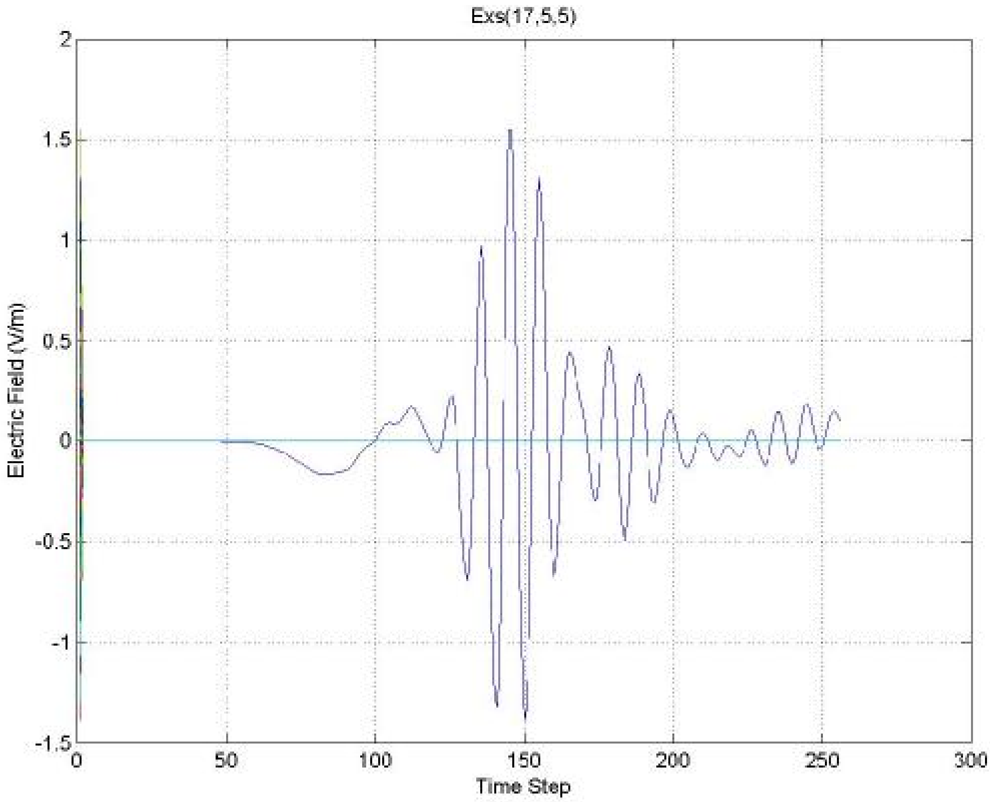

Figure 3.3. Electric field value versus time step in the Ex(17,5,5)

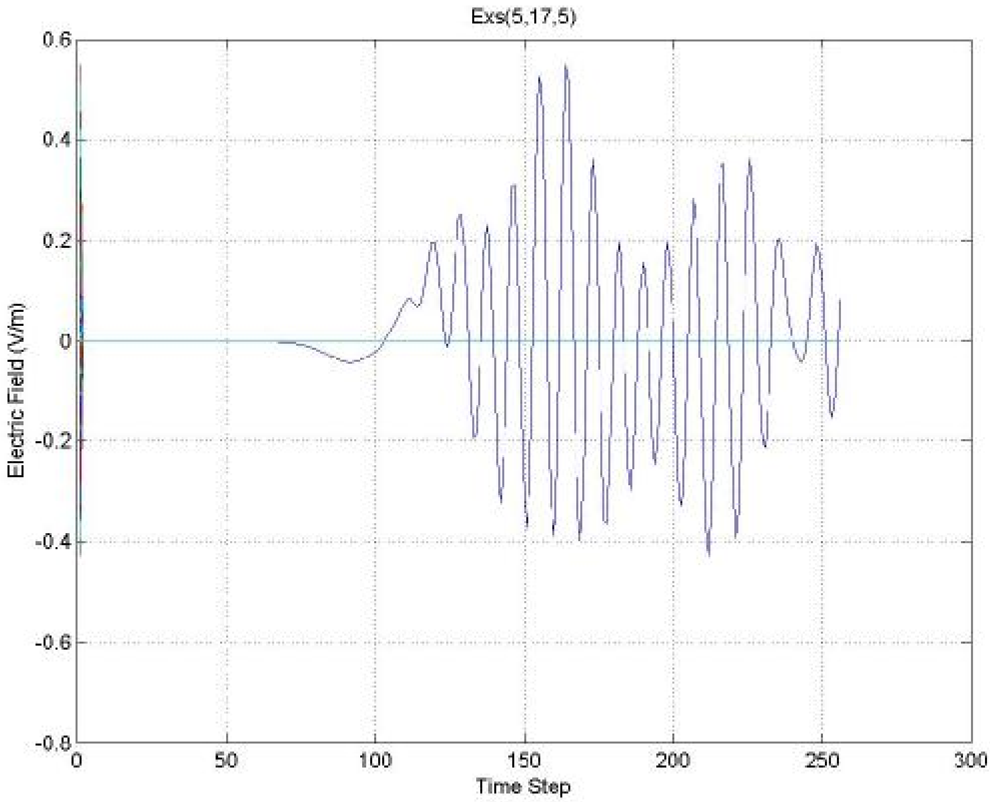

Figure 3.4. Electric field value versus time step in the Ex(5,17,5)

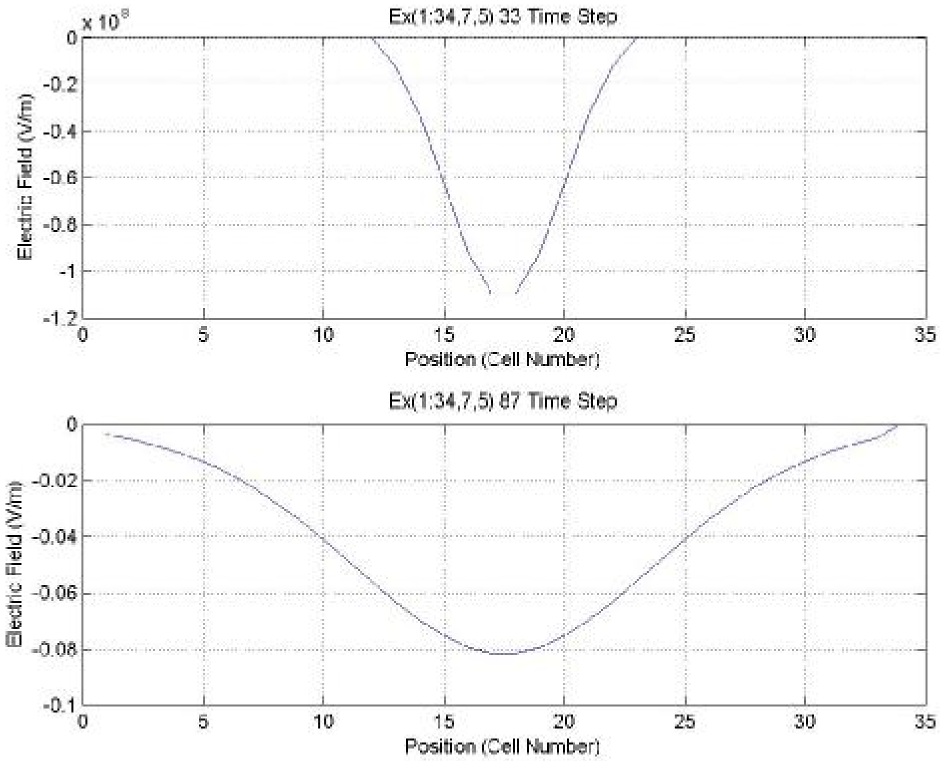

Figure 3.5. Electric field value versus position (cell number) in the Ex(1:34,7,5) for 33. and 87. time steps

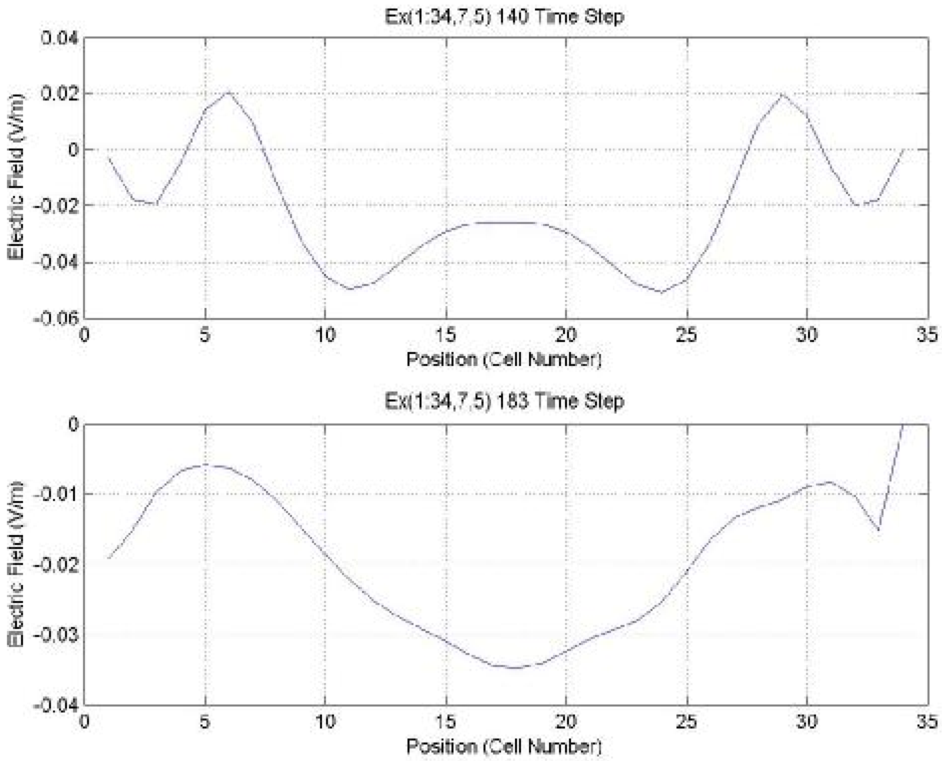

Figure 3.6. Electric field value versus position (cell number) in the Ex(1:34,7,5) for 140. and 183. time steps

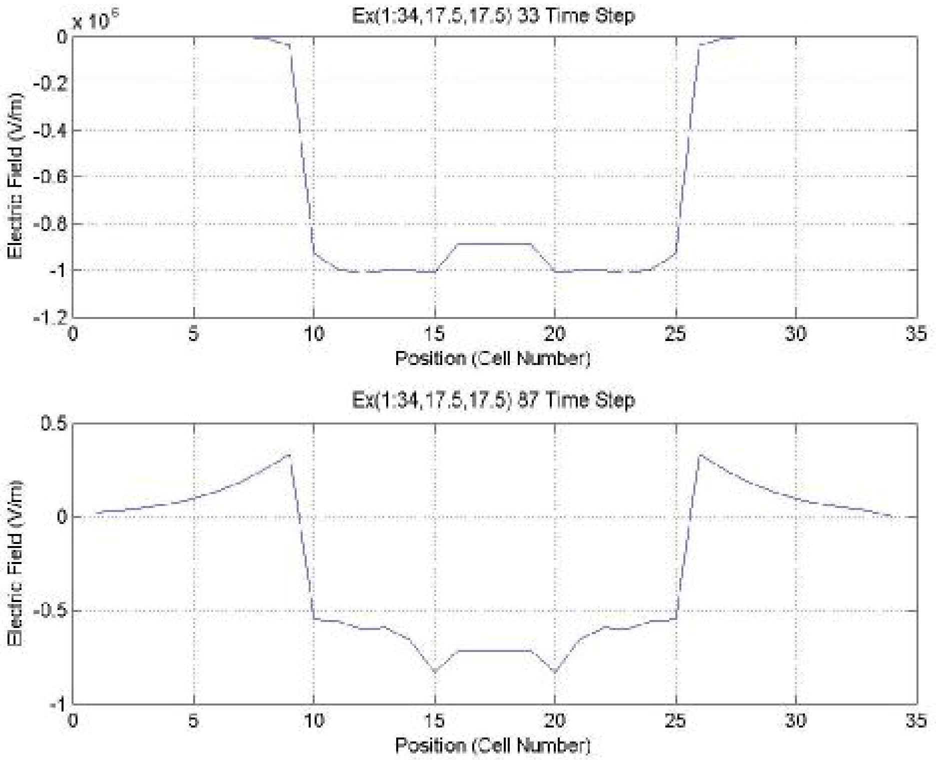

Figure 3.7. Electric field value versus position (cell number) in the Ex(1:34,17.5,17.5) for 33. and 87. time steps

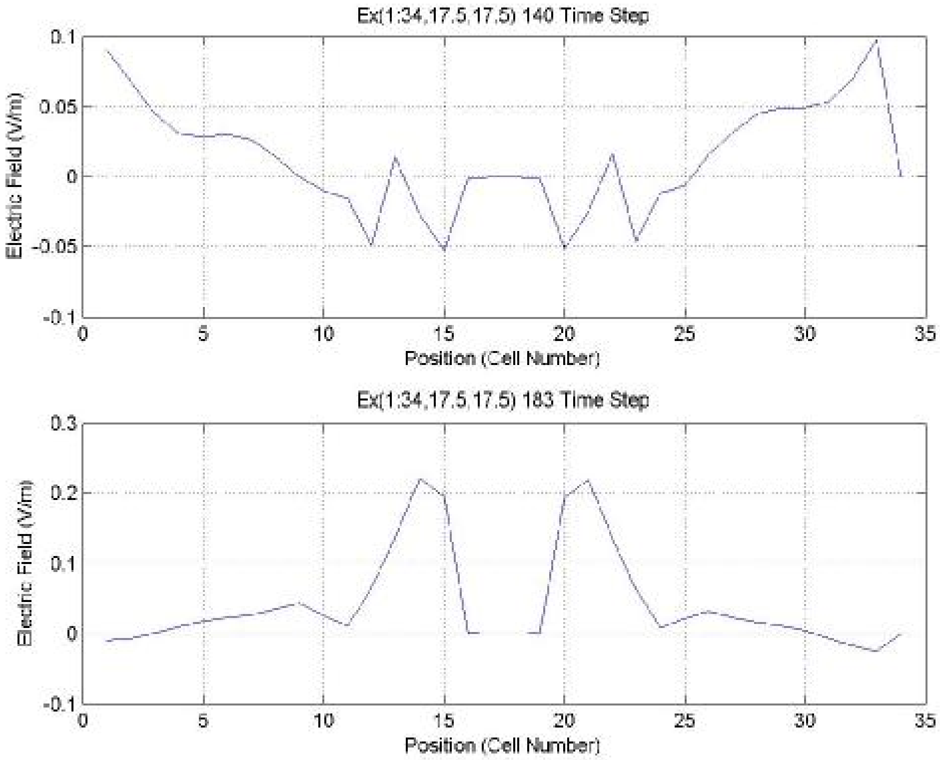

Figure 3.8. Electric field value versus position (cell number) in the Ex(1:34,17.5,17.5) for 140. and 183. time steps

4. CONCLUSION

As an important canonical structure, the scattering from an inhomogeneous sphere by using FDTD is investigated. Scattering from a canonical structure is a basic problem and it can be used for analyzing the scattering from more complex structure. The sphere is a canonical structure and so scattering from sphere constitutes a foundation for solving the scattering from more complex structure.

The results obtained from this study can be used for modeling the scattering problems which are formed by using electromagnetic waves for example in biomedical applications. Because of selected canonical structure can be modeled as the organs in human body it is thought that this work can be interpreted as an application of FDTD in biotechnology and medicine.

To find the scattered field component a computer program which Karl S. Kunz and Raymond J. Luebbers have made in 1992 is modified for our scattering object. And numerical results are obtained.

REFERENCES

TAFLOVE, A.,1995. Computational Electrodynamics:The Finite-Difference Time-Domain Method. Norwood ,MA; Artech House.

SIMICEVIC, N., HAYNIE D.T., 2005. FDTD Simulation of Exposure of Biological Material to Electromagnetic Nanopulses.

SULLIVAN, D.M, 2000. Electromagnetic Simulation Using The FDTD Method New York: Institute of Electrical and Electronics Engineers.

SHAO, Z., HONG, W., 1998. Finite-difference time-domain method with conformal meshes for scattering analysis of conducting cylinders, Antennas and Propagation Society International Symposium.

ZHANG, L.M., 1996. Finite difference solution of EM fields by asyimtotic waveform technique, Microwaves, Antennas and Propagation, IEE Proceedings.

PAINTER, O., LEE, R. K. and KIM, I., 1999. Two-dimensional photonic bandgap defect mode laser.

YEE, K. S. , 1966. Numerical solution of boundary value problems involving Maxwell's equations in isotropic media.

MALONEY, J. ***0.*** and KERLER M. P. , 1998. Finite-Difference Time-Domain Analysis of Periodic Structures.

KRUMPHOLZ, M. and KATEHI, L. P. B., 1996. MRTD: New time domain schemes based on multiresolution analysis.

GEDNEY, S. and LANSING, F., 1995. Explicit Time-Domain Solutions of Maxwell's Equations Using Non-Orthogonal and Unstructured Grids.

SHIBATA, T., HAYASHI, T. and KIMURA, T.,1988. Analysis of microstrip circuits using three-dimensional full-wave electromagnetic field analysis in the time domain.

TAFLOVE, A. and BRODWIN, M. E. ,1975. Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations.

KUNZ, K. S., LUEBBERS, R. J., 1992. The finite difference time domain method for electromagnetics.

BIOGRAPHY

I was born in Adana at 1981. I completed the high-school at 1998 and i entered to Electrical and Electronics Engineering Department of Dumlupınar University in the same year. I graduated from this university as an Electric and Electronics Engineer at 2002 and joined to Electrical and Electronics Engineering Department of Çukurova University as a MS student in the same year.

PROGRAM CODE

```
     PROGRAM FDTDA
C
C    PENN STATE UNIVERSITY FINITE DIFFERENCE TIME DOMAIN
C    ELECTROMAGNETIC ANALYSIS COMPUTER CODE--VERSION A
C
C    THIS CODE IS A SCATTERED FIELD FORMULATION
C
C    VERSION A:  AUGUST 25, 2006
C
C    TO REPORT ANY CODE ERRORS OR FOR ADDITIONAL INFORMATION
C    CONTACT:
C
C    DR. RAYMOND J. LUEBBERS
C    203 E E EAST BLDG.
C    UNIVERSITY PARK, PA 16802
C    INTERNET:  LU4@PSUVM.PSU.EDU, BITNET:  LU4@PSUVM.BITNET
C
C    THIS VERSION INCLUDES:
C    1.) 1ST ORDER E FIELD OUTER RADIATION BOUNDARY CONDITION (ORBC)
C    2.) 2ND ORDER E FIELD OUTER RADIATION BOUNDARY CONDITION (ORBC)
C    3.) CAPABILITY TO SPECIFY DIRECTION AND POLARIZATION OF
C        INCIDENT PLANE WAVE
C    4.) ERROR CHECKING FOR NTYPE AND IOBS, JOBS AND KOBS FOR
C        NEAR ZONE FIELD SAMPLING IN SUBROUTINE DATSAV
C    5.) ERROR CHECKING OF IDONE, IDTWO AND IDTHRE VALUES

C    COMMONA.FOR:
C    NX, NY, NZ DETERMINE NUMBER OF CELLS IN PROBLEM SPACE
C    NTEST IS NUMBER OF QUANTITIES SAMPLED AND WRITTEN VS TIME
C    TIME VARIABLE IS NSTOP.
C
C    THE QUANTITIES TO BE SAMPLED (NEAR ZONE COMPUTATIONS) ARE
C    TO BE SET IN SUBROUTINE DATSAV
C
C    DEFINE OUTPUT FILES
C
C    DIAGS3D.DAT   =>  DIAGNOSTICS OF SOME SETUP PARAMETERS
C    NZOUT3D.DAT   =>  NEAR-ZONE FIELDS OR CURRENTS AS DEFINED IN
DATSAV
C
C    WARNING:  PLEASE READ THE FOLLOWING COMMENTS REGARDING THE
C           INCLUDE STATEMENT!
C
C    INCLUDE COMMON FILE (STATEMENT APPEARS IN EVERY FUNCTION
C    SUBPROGRAM AND SUBROUTINE).  THIS INCLUDE STATEMENT IS
C    APPROPRIATE FOR MOST MACHINES (I.E. VAX, SILICON GRAPHICS,
C    386/486 PC'S WITH LAHEY COMPILER).  THE WATFOR (PC WATFOR)
C    VERSION OF THE INCLUDE STATEMENT IS DIFFERENT AND IS INCLUDED
C    IMMEDIATELY AFTER THE NORMAL INCLUDE STATEMENT.  IF USING THE
```

```
C    WATFOR COMPILER, DELETE THE NORMAL INCLUDE STATEMENTS OR
COMMENT
C    THEM OUT.  THE INCLUDE STATEMENT FOR IBM VS FORTRAN IS
C    DIFFERENT ALSO. THE INCLUDE STATEMENT FOR AN IBM VS FORTRAN
C    COMPILER MUST BE CHANGED TO:  INCLUDE 'COMMONA FORTRAN A1'
C    THE INCLUDE STATEMENT MUST BE CHANGED IN EVERY SUBROUTINE
AND
C    FUNCTION SUBPROGRAM AND IS MOST EFFICIENTLY DONE USING A
GLOBAL
C    SEARCH AND REPLACE WITH AN EDITOR.  IF YOU ARE USING THE WATFOR
C    COMPILER, MAKE SURE THE COMMON FILE HAS EXTENSION '.FOR'.
C
     INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C    OPEN DATA FILES.  THE OPEN STATEMENTS SHOWN BELOW ARE
APPROPRIATE
C    FOR MOST MACHINES (I.E. VAX, SILICON GRAPHICS, 386/486 PC'S).
C    HOWEVER, THE IBM VERSION OF THE OPEN STATEMENTS ARE DENOTED
BY
C    LINES WITH C###. UNCOMMENT APPROPRIATE LINES FOR IBM VERSION
AND
C    COMMENT OUT OTHER LINES.
C
     OPEN(UNIT=17,FILE='DIAGS3D.DAT',STATUS='UNKNOWN')
C###      OPEN (UNIT=17,FILE='/DIAGS3D DATA E1',STATUS='UNKNOWN')
c        OPEN(UNIT=10,FILE='NZOUT3D.DAT',STATUS='UNKNOWN')
C###        OPEN (UNIT=10,FILE='/NZOUT3D DATA E1',STATUS='UNKNOWN')
      OPEN(UNIT=9,FILE='s3mline50.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=7,FILE='s3mline100.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=5,FILE='s3mline150.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=3,FILE='s3mline200.DAT',STATUS='UNKNOWN')
          OPEN(UNIT=1,FILE='s3mline250.DAT',STATUS='UNKNOWN')

C
C    ZERO PARAMETERS, GENERATE PROBLEM SPACE, INTERACTION OBJECT,
C    AND EXCITATION
C
     CALL ZERO
     CALL BUILD
     CALL SETUP
C    ****************************************************************
C    MAIN LOOP FOR FIELD COMPUTATIONS AND DATA SAVING
C    ****************************************************************
     T=0.0
     DO 100 N=1,NSTOP
C
     WRITE (*,*) N
C
C      ADVANCE SCATTERED ELECTRIC FIELD
```

```
      CALL EXSFLD
      CALL EYSFLD
      CALL EZSFLD
C
C      APPLY RADIATION BC (SECOND ORDER)
C
      CALL RADEYX
      CALL RADEZX
      CALL RADEZY
      CALL RADEXY
      CALL RADEXZ
      CALL RADEYZ
C
C      ADVANCE TIME BY 1/2 TIME STEP
C
      T=T+DT/2.
C
C      ADVANCE SCATTERED MAGNETIC FIELD
C
      CALL HXSFLD
      CALL HYSFLD
      CALL HZSFLD
C
C      ADVANCE TIME ANOTHER 1/2 STEP
C
      T=T+DT/2.
C
C      SAMPLE FIELDS IN SPACE AND WRITE TO DISK
C
      CALL DATSAV
C
 100  CONTINUE
     T=NSTOP*DT
     WRITE (17,200) T,NSTOP
  200 FORMAT(T2,'EXIT TIME= ',E14.7,' SECONDS,  AT TIME STEP',I6,/)
C
C    CLOSE DATA FILES
C
     CLOSE (UNIT=10)
     CLOSE (UNIT=17)
     STOP
     END
C    *******************************************************
     SUBROUTINE BUILD
C
     INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C    THIS SUBROUTINE IS USED TO DEFINE THE SCATTERING OBJECT WITHIN
C    THE FDTD SOLUTION SPACE.  USER MUST SPECIFY IDONE, IDTWO AND
```

```
C    IDTHRE AT DIFFERENT CELL LOCATIONS TO DEFINE THE SCATTERING
C    OBJECT.  SEE THE YEE PAPER (IEEE TRANS. ON AP, MAY 1966) FOR
C    A DESCRIPTION OF THE FDTD ALGORITHM AND THE LOCATION OF FIELD
C    COMPONENTS.
C
C    GEOMETRY DEFINITION
C
C    IDONE, IDTWO, AND IDTHRE ARE USED TO SPECIFY MATERIAL IN CELL
C    I,J,K.
C    IDONE DETERMINES MATERIAL FOR X COMPONENTS OF E
C    IDTWO FOR Y COMPONENTS, IDTHRE FOR Z COMPONENTS
C    THUS ANISOTROPIC MATERIALS WITH DIAGONAL TENSORS CAN BE
MODELLED
C
C    SET IDONE,IDTWO, AND/OR IDTHRE FOR EACH I,J,K CELL =
C           0      FOR FREE SPACE
C           1      FOR PEC
C           2-9     FOR LOSSY DIELECTRICS
C
C    SUBROUTINE DCUBE BUILDS A CUBE OF DIELECTRIC MATERIAL BY
SETTING
C    IDONE, IDTWO, IDTHRE TO THE SAME MATERIAL TYPE.  THE MATERIAL
C    TYPE IS SPECIFIED BY MTYPE.  SPECIFY THE STARTING CELL (LOWER
C    LEFT CORNER (I.E. MINIMUM I,J,K VALUES) AND SPECIFY THE CELL
C    WIDTH IN EACH DIRECTION (USE THE NUMBER OF CELLS IN EACH
C    DIRECTION).  USE NZWIDE=0 FOR A INFINITELY THIN PLATE IN THE
C    XY PLANE.  FOR PEC PLATE USE MTYPE=1.  ISTART, JSTART, KSTART ARE
C    USED TO DEFINE THE STARTING CELL AND NXWIDE, NYWIDE AND
NZWIDE EACH
C    SPECIFY THE OBJECT WIDTH IN CELLS IN THE X, Y AND Z DIRECTIONS.
C    INDIVIDUAL IDONE, TWO OR THRE COMPONENTS CAN BE SET MANUALLY
C    FOR WIRES, ETC.  DCUBE DOES NOT WORK FOR WIRES (I.E. NXWIDE=0 AND
C    NYWIDE=0 FOR EXAMPLE)!
C
C   Build sphere with center at (SC,SC,SC) and radius RA.
      MTYPE=2
      RA=8.2
      SC=17.5
      DO 100 I=1,NX
        DO 200 J=1,NY
          DO 300 K=1,NZ
            R=SQRT((I-SC)**2+(J-SC)**2+(K-SC)**2)
            IF (R.LT.RA) CALL DCUBE (I,J,K,1,1,1,MTYPE)
 300      CONTINUE
 200    CONTINUE
 100  CONTINUE

      MTYPE=5
      RA=5.2
      SC=17.5
```

```
     DO 110 I=1,NX
       DO 210 J=1,NY
         DO 310 K=1,NZ
           R=SQRT((I-SC)**2+(J-SC)**2+(K-SC)**2)
           IF (R.LT.RA) CALL DCUBE (I,J,K,1,1,1,MTYPE)
 310     CONTINUE
 210   CONTINUE
 110  CONTINUE

     MTYPE=9
     RA=2.2
     SC=17.5
     DO 120 I=1,NX
       DO 220 J=1,NY
         DO 320 K=1,NZ
           R=SQRT((I-SC)**2+(J-SC)**2+(K-SC)**2)
           IF (R.LT.RA) CALL DCUBE (I,J,K,1,1,1,MTYPE)
 320     CONTINUE
 220   CONTINUE
 120  CONTINUE


C
C     THE FOLLOWING SECTION OF CODE IS USED TO CHECK IF THE USER HAS
C     SPECIFIED THE PROPER MATERIAL TYPES TO THE PROPER IDXXX ARRAYS.
C
C     FOR MATERIAL TYPE = ?
C        0 FOR FREE SPACE                 USE  IDONE-IDTHRE ARRAYS
C        1 FOR PEC                        USE  IDONE-IDTHRE ARRAYS
C        2-9 FOR LOSSY DIELECTRICS        USE  IDONE-IDTHRE ARRAYS
C
     DO 1000 K=1,NZ
       DO 900 J=1,NY
         DO 800 I=1,NX
           IF((IDONE(I,J,K).GE.10).OR.(IDTWO(I,J,K).GE.10).OR.
    $(IDTHRE(I,J,K).GE.10)) THEN
             WRITE (17,*)'ERROR OCCURED. ILLEGAL VALUE FOR'
             WRITE (17,*)'DIELECTRIC TYPE (IDONE-IDTHRE) '
             WRITE (17,*)'AT LOCATION:',I,',',J,',',K
             WRITE (17,*)'EXECUTION HALTED.'
             STOP
           ENDIF
 800     CONTINUE
 900   CONTINUE
 1000 CONTINUE
     RETURN
     END
C     ******************************************************
     SUBROUTINE DCUBE
(ISTART,JSTART,KSTART,NXWIDE,NYWIDE,NZWIDE,MTYPE)
```

```
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE SETS ALL TWELVE IDXXX COMPONENTS FOR ONE CUBE
C     TO THE SAME MATERIAL TYPE SPECIFIED BY MTYPE.  IF NXWIDE,
NYWIDE,
C     OR NZWIDE=0, THEN ONLY 4 IDXXX ARRAY COMPONENTS WILL BE SET
C     CORRESPONDING TO AN INFINITELY THIN PLATE.  THE SUBROUTINE IS
MOST
C     USEFUL CONSTRUCTING OBJECTS WITH MANY CELLS OF THE SAME
MATERIAL
C     (I.E. CUBES, PEC PLATES, ETC.).  THIS SUBROUTINE DOES NOT
C     AUTOMATICALLY DO WIRES!
C
      IMAX=ISTART+NXWIDE-1
      JMAX=JSTART+NYWIDE-1
      KMAX=KSTART+NZWIDE-1
C
      IF (NXWIDE.EQ.0) THEN
        DO 20 K=KSTART,KMAX
          DO 10 J=JSTART,JMAX
            IDTWO(ISTART,J,K)=MTYPE
            IDTWO(ISTART,J,K+1)=MTYPE
            IDTHRE(ISTART,J,K)=MTYPE
            IDTHRE(ISTART,J+1,K)=MTYPE
 10         CONTINUE
 20       CONTINUE
      ELSEIF (NYWIDE.EQ.0) THEN
        DO 40 K=KSTART,KMAX
          DO 30 I=ISTART,IMAX
            IDONE(I,JSTART,K)=MTYPE
            IDONE(I,JSTART,K+1)=MTYPE
            IDTHRE(I,JSTART,K)=MTYPE
            IDTHRE(I+1,JSTART,K)=MTYPE
 30         CONTINUE
 40       CONTINUE
      ELSEIF (NZWIDE.EQ.0) THEN
        DO 60 J=JSTART,JMAX
          DO 50 I=ISTART,IMAX
            IDONE(I,J,KSTART)=MTYPE
            IDONE(I,J+1,KSTART)=MTYPE
            IDTWO(I,J,KSTART)=MTYPE
            IDTWO(I+1,J,KSTART)=MTYPE
 50         CONTINUE
 60       CONTINUE
      ELSE
        DO 90 K=KSTART,KMAX
          DO 80 J=JSTART,JMAX
            DO 70 I=ISTART,IMAX
```

```
          IDONE(I,J,K)=MTYPE
          IDONE(I,J,K+1)=MTYPE
          IDONE(I,J+1,K+1)=MTYPE
          IDONE(I,J+1,K)=MTYPE
          IDTWO(I,J,K)=MTYPE
          IDTWO(I+1,J,K)=MTYPE
          IDTWO(I+1,J,K+1)=MTYPE
          IDTWO(I,J,K+1)=MTYPE
          IDTHRE(I,J,K)=MTYPE
          IDTHRE(I+1,J,K)=MTYPE
          IDTHRE(I+1,J+1,K)=MTYPE
          IDTHRE(I,J+1,K)=MTYPE
 70       CONTINUE
 80     CONTINUE
 90   CONTINUE
      ENDIF
      RETURN
      END
C     ******************************************************
      SUBROUTINE SETUP
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE INITIALIZES THE COMPUTATIONS
C
C     DEFINE PI AND C
C
      C=1.0/SQRT(XMU0*EPS0)
      PI=4.0*ATAN(1.0)
C
C     CALCULATE DT--THE MAXIMUM TIME STEP ALLOWED BY THE
C     COURANT STABILITY CONDITION
C
      DTXI=C/DELX
      DTYI=C/DELY
      DTZI=C/DELZ
C
      DT=1./SQRT(DTXI**2+DTYI**2+DTZI**2)
C
C     PARAMETER ALPHA IS THE DECAY RATE DETERMINED BY BETA.
C
      ALPHA=(1./(BETA*DT/4.0))**2
C
      BETADT =  BETA*DT
      PERIOD = 2.0*BETADT
C
C     SET OFFSET FOR COMPUTING INCIDENT FIELDS
C
      OFF=1.0
C
```

```
C     THE FOLLOWING LINES ARE FOR SMOOTH COSINE INCIDENT FUNCTION
      W1 = 2.0*PI/PERIOD
      W2 = 2.0*W1
      W3 = 3.0*W1
C     FIND DIRECTION COSINES FOR INCIDENT FIELD
      COSTH=COS(PI*THINC/180.)
      SINTH=SIN(PI*THINC/180.)
      COSPH=COS(PI*PHINC/180.)
      SINPH=SIN(PI*PHINC/180.)
C     FIND AMPLITUDE OF INCIDENT FIELD COMPONENTS
      AMPX=AMP*(ETHINC*COSTH*COSPH-EPHINC*SINPH)
      AMPY=AMP*(ETHINC*COSTH*SINPH+EPHINC*COSPH)
      AMPZ=AMP*(-ETHINC*SINTH)
C
C     FIND RELATIVE SPATIAL DELAY FOR X, Y, Z CELL DISPLACEMENT
C
      XDISP=-COSPH*SINTH
      YDISP=-SINPH*SINTH
      ZDISP=-COSTH
C
C     DEFINE CONSTITUTIVE PARAMETERS--EPS,SIGMA
C     THIS CODE ASSUMES THAT THE DIELECTRIC MATERIALS ALL HAVE
C     A PERMEABILITY OF MU0.  IF YOU NEED A MATERIAL WITH MAGNETIC
C     PROPERTIES, USE ANOTHER FDTD CODE SUCH AS FDTDC, FDTDD, FDTDG
C     OR FDTDH.
C
C     THESE CORRESPOND TO MATERIAL TYPES IN IDONE, IDTWO, AND
C     IDTHRE ARRAYS.  SEE COMMENTS IN SUBROUTINE BUILD
C
C     VALID CASES:  IDXXX(I,J,K)  = 0 FOR FREE SPACE IN CELL I,J,K
C                                 = 1 FOR PEC
C                                 = 2-9 FOR LOSSY DIELECTRICS
C     FOR PARTICULAR COMPONENTS OF FIELDS AS DETERMINED BY XXX
C
C     IF IDONE (10,10,10) = 3 THEN THE USER MUST SPECIFY AN EPSILON, MU
C     AND SIGMA FOR MATERIAL TYPE 3.  THAT IS, EPS(3)=EPSILON FOR
C     MATERIAL TYPE 3 AND SIGMA(3)= CONDUCTIVITY FOR MATERIAL TYPE 3.
C     ALL PARAMETERS ARE DEFINED IN MKS UNITS.
C
      DO 10 I=1,9
       EPS(I)=EPS0
       SIGMA(I)=0.0
 10    CONTINUE
C
C     DEFINE EPS AND SIGMA FOR EACH MATERIAL HERE
C
      EPS(2)=4.0*EPS0
      SIGMA(2)=0.000005

          EPS(5)=4.0*EPS0
```

```
      SIGMA(5)=0.005

      EPS(9)=4.0*EPS0
      SIGMA(9)=5



C
C     GENERATE MULTIPLICATIVE CONSTANTS FOR FIELD UPDATE EQUATIONS
C
C     FREE SPACE
C
      DTEDX=DT/(EPS0*DELX)
      DTEDY=DT/(EPS0*DELY)
      DTEDZ=DT/(EPS0*DELZ)
      DTMDX=DT/(XMU0*DELX)
      DTMDY=DT/(XMU0*DELY)
      DTMDZ=DT/(XMU0*DELZ)
C
C     LOSSY DIELECTRICS
C
      DO 20 I=2,9
        ESCTC(I)=EPS(I)/(EPS(I)+SIGMA(I) *DT)
        EINCC(I)=SIGMA(I) *DT/(EPS(I)+SIGMA(I) *DT)
        EDEVCN(I)=DT*(EPS(I)-EPS0)/(EPS(I)+SIGMA(I) *DT)
        ECRLX(I)=DT/((EPS(I)+SIGMA(I) *DT)*DELX)
        ECRLY(I)=DT/((EPS(I)+SIGMA(I) *DT)*DELY)
        ECRLZ(I)=DT/((EPS(I)+SIGMA(I) *DT)*DELZ)
 20     CONTINUE
C
C     FIND MAXIMUM SPATIAL DELAY TO MAKE SURE PULSE PROPAGATES
C     INTO SPACE PROPERLY.
C
      DELAY=0.0
      IF (XDISP.LT.0.) DELAY=DELAY-XDISP*NX1*DELX
      IF (YDISP.LT.0.) DELAY=DELAY-YDISP*NY1*DELY
      IF (ZDISP.LT.0.) DELAY=DELAY-ZDISP*NZ1*DELZ
C
C     COMPUTE OUTER RADIATION BOUNDARY CONDITION (ORBC) CONSTANTS
C
      CXD=(C*DT-DELX)/(C*DT+DELX)
      CYD=(C*DT-DELY)/(C*DT+DELY)
      CZD=(C*DT-DELZ)/(C*DT+DELZ)
C
      CXU=CXD
      CYU=CYD
      CZU=CZD
C
C     COMPUTE 2ND ORDER ORBC CONSTANTS
C
```

```
      CXX=2.*DELX/(C*DT+DELX)
      CYY=2.*DELY/(C*DT+DELY)
      CZZ=2.*DELZ/(C*DT+DELZ)
C
      CXFYD=DELX*C*DT*C*DT/(2.*DELY*DELY*(C*DT+DELX))
      CXFZD=DELX*C*DT*C*DT/(2.*DELZ*DELZ*(C*DT+DELX))
      CYFZD=DELY*C*DT*C*DT/(2.*DELZ*DELZ*(C*DT+DELY))
      CYFXD=DELY*C*DT*C*DT/(2.*DELX*DELX*(C*DT+DELY))
      CZFXD=DELZ*C*DT*C*DT/(2.*DELX*DELX*(C*DT+DELZ))
      CZFYD=DELZ*C*DT*C*DT/(2.*DELY*DELY*(C*DT+DELZ))
C
C     WRITE SETUP DATA TO FILE DIAGS3D.DAT
C
      WRITE (17,400) NX,NY,NZ
      WRITE (17,500) DELX,DELY,DELZ
      WRITE (17,600) DT,NSTOP
      WRITE (17,700) AMP,ALPHA,BETA
      WRITE (17,800) ETHINC,EPHINC
      WRITE (17,900) THINC,PHINC
      WRITE (17,1000) 'THE INCIDENT EX AMPLITUDE = ',AMPX,'V/M'
      WRITE (17,1000) 'THE INCIDENT EY AMPLITUDE = ',AMPY,'V/M'
      WRITE (17,1000) 'THE INCIDENT EZ AMPLITUDE = ',AMPZ,'V/M'
      WRITE (17,1100) 'RELATIVE SPATIAL DELAY = ',DELAY
C
 100  FORMAT (T2,A26,/)
 200  FORMAT (T2,A25,/)
 300  FORMAT (T2,A34,/)
 400  FORMAT (T2, 'THE PROBLEM SPACE SIZE IS', I4,' BY',I4,' BY',I4,
     $' CELLS IN THE X, Y, Z DIRECTIONS',/)
 500  FORMAT (T2,'CELL SIZE: DELX=',F10.6,' DELY=',F10.6,', DELZ=',
     $F10.6, ' METERS',/)
 600  FORMAT (T2,'TIME STEP IS ',E12.6,' SECONDS, MAXIMUM OF',I6,
     $' TIME STEPS',/)
 700  FORMAT (T2,'INCIDENT GAUSSIAN PULSE AMPLITUDE=',F6.0,' V/M',
     $/,T2,'DECAY FACTOR ALPHA=',E12.3,/,T2,'WIDTH BETA=',F6.0,/)
 800  FORMAT (T2,'INCIDENT PLANE WAVE POLARIZATION:',/,T2,
     $ 'RELATIVE ELECTRIC FIELD THETA COMPONENT=', F4.1,/,T2,
     $ 'RELATIVE ELECTRIC FIELD PHI COMPONENT=', F4.1,/)
 900  FORMAT (T2,'PLANE WAVE INCIDENT FROM THETA=',F8.2,' PHI=',F8.2,
     $' DEGREES',/)
 1000 FORMAT (T2,A28,F12.5,2X,A3)
 1100 FORMAT (/,T2,A25,F10.7,/)
      RETURN
      END
C     ******************************************************
      SUBROUTINE EXSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
```

```
C     THIS SUBROUTINE UPDATES THE EX SCATTERED FIELD
C
      DO 30 K=2,NZ1
        KK = K
        DO 20 J=2,NY1
          JJ = J
          DO 10 I=1,NX1
C
C           DETERMINE MATERIAL TYPE
C
            IF(IDONE(I,J,K).EQ.0) GO TO 100
            IF(IDONE(I,J,K).EQ.1) GO TO 200
            GO TO 300
C
C           FREE SPACE
C
 100        EXS(I,J,K)=EXS(I,J,K)+(HZS(I,J,K)-HZS(I,J-1,K))*DTEDY
     $                     -(HYS(I,J,K)-HYS(I,J,K-1))*DTEDZ
C
            GO TO 10
C
C           PERFECT CONDUCTOR
C
 200        II = I
            EXS(I,J,K)=-EXI(II,JJ,KK)
C
            GO TO 10
C
C           LOSSY DIELECTRIC
C
 300        II = I
            EXS(I,J,K)=EXS(I,J,K)*ESCTC(IDONE(I,J,K))
     $      -EINCC( IDONE(I,J,K))*EXI(II,JJ,KK)
     $      -EDEVCN(IDONE(I,J,K))*DEXI(II,JJ,KK)
     $      +(HZS(I,J,K)-HZS(I,J-1,K))*ECRLY(IDONE(I,J,K))
     $      -(HYS(I,J,K)-HYS(I,J,K-1))*ECRLZ(IDONE(I,J,K))
C
 10         CONTINUE
 20       CONTINUE
 30     CONTINUE
C
      RETURN
      END
C     ***************************************************************
      SUBROUTINE EYSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE UPDATES THE EY SCATTERED FIELD COMPONENTS
```

```
C
      DO 30 K=2,NZ1
        KK = K
        DO 20 J=1,NY1
          JJ = J
          DO 10 I=2,NX1
C
C           DETERMINE MATERIAL TYPE
C
            IF(IDTWO(I,J,K).EQ.0) GO TO 100
            IF(IDTWO(I,J,K).EQ.1) GO TO 200
            GO TO 300
C
C           FREE SPACE
C
 100        EYS(I,J,K)=EYS(I,J,K)+(HXS(I,J,K)-HXS(I,J,K-1))*DTEDZ
     $                    -(HZS(I,J,K)-HZS(I-1,J,K))*DTEDX
C
            GO TO 10
C
C           PERFECT CONDUCTOR
C
 200        II = I
            EYS(I,J,K)=-EYI(II,JJ,KK)
C
            GO TO 10
C
C           LOSSY DIELECTRIC
C
 300        II = I
            EYS(I,J,K)=EYS(I,J,K)*ESCTC(IDTWO(I,J,K))
     $      -EINCC( IDTWO(I,J,K))*EYI(II,JJ,KK)
     $      -EDEVCN(IDTWO(I,J,K))*DEYI(II,JJ,KK)
     $      +(HXS(I,J,K)-HXS(I,J,K-1))*ECRLZ(IDTWO(I,J,K))
     $      -(HZS(I,J,K)-HZS(I-1,J,K))*ECRLX(IDTWO(I,J,K))
C
  10      CONTINUE
  20    CONTINUE
  30  CONTINUE
C
      RETURN
      END
C     ***********************************************************
      SUBROUTINE EZSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE UPDATES THE EZ SCATTERED FIELD COMPONENTS
C
```

```
      DO 30 K=1,NZ1
        KK = K
        DO 20 J=2,NY1
          JJ = J
          DO 10 I=2,NX1
C
C           DETERMINE MATERIAL TYPE
C
            IF(IDTHRE(I,J,K).EQ.0) GO TO 100
            IF(IDTHRE(I,J,K).EQ.1) GO TO 200
            GO TO 300
C
C           FREE SPACE
C
 100        EZS(I,J,K)=EZS(I,J,K)+(HYS(I,J,K)-HYS(I-1,J,K))*DTEDX
     $                       -(HXS(I,J,K)-HXS(I,J-1,K))*DTEDY
C
            GO TO 10
C
C           PERFECT CONDUCTOR
C
 200        II = I
            EZS(I,J,K)=-EZI(II,JJ,KK)
C
            GO TO 10
C
C           LOSSY DIELECTRIC
C
 300        II = I
            EZS(I,J,K)=EZS(I,J,K)*ESCTC(IDTHRE(I,J,K))
     $      -EINCC( IDTHRE(I,J,K))*EZI(II,JJ,KK)
     $      -EDEVCN(IDTHRE(I,J,K))*DEZI(II,JJ,KK)
     $      +(HYS(I,J,K)-HYS(I-1,J,K))*ECRLX(IDTHRE(I,J,K))
     $      -(HXS(I,J,K)-HXS(I,J-1,K))*ECRLY(IDTHRE(I,J,K))
C
 10       CONTINUE
 20     CONTINUE
 30   CONTINUE
C
      RETURN
      END
C     ********************************************************
      SUBROUTINE RADEZX
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 K=1,NZ1
```

```
      J=2
      EZS(1,J,K)=EZSX1(2,J,K)+CXD*(EZS(2,J,K)-EZSX1(1,J,K))
      EZS(NX,J,K)=EZSX1(3,J,K)+CXU*(EZS(NX1,J,K)-EZSX1(4,J,K))
      J=NY1
      EZS(1,J,K)=EZSX1(2,J,K)+CXD*(EZS(2,J,K)-EZSX1(1,J,K))
      EZS(NX,J,K)=EZSX1(3,J,K)+CXU*(EZS(NX1,J,K)-EZSX1(4,J,K))
 10   CONTINUE
C
     DO 20 J=3,NY1-1
      K=1
      EZS(1,J,K)=EZSX1(2,J,K)+CXD*(EZS(2,J,K)-EZSX1(1,J,K))
      EZS(NX,J,K)=EZSX1(3,J,K)+CXU*(EZS(NX1,J,K)-EZSX1(4,J,K))
      K=NZ1
      EZS(1,J,K)=EZSX1(2,J,K)+CXD*(EZS(2,J,K)-EZSX1(1,J,K))
      EZS(NX,J,K)=EZSX1(3,J,K)+CXU*(EZS(NX1,J,K)-EZSX1(4,J,K))
 20   CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
     DO 40 K=2,NZ1-1
      DO 30 J=3,NY1-1
       EZS(1,J,K)=-EZSX2(2,J,K)+CXD*(EZS(2,J,K)+EZSX2(1,J,K))
    $    +CXX*(EZSX1(1,J,K)+EZSX1(2,J,K))
    $    +CXFYD*(EZSX1(1,J+1,K)-2.*EZSX1(1,J,K)+EZSX1(1,J-1,K)
    $    +EZSX1(2,J+1,K)-2.*EZSX1(2,J,K)+EZSX1(2,J-1,K))
    $    +CXFZD*(EZSX1(1,J,K+1)-2.*EZSX1(1,J,K)+EZSX1(1,J,K-1)
    $    +EZSX1(2,J,K+1)-2.*EZSX1(2,J,K)+EZSX1(2,J,K-1))
       EZS(NX,J,K)=-EZSX2(3,J,K)+CXD*(EZS(NX1,J,K)+EZSX2(4,J,K))
    $    +CXX*(EZSX1(4,J,K)+EZSX1(3,J,K))
    $    +CXFYD*(EZSX1(4,J+1,K)-2.*EZSX1(4,J,K)+EZSX1(4,J-1,K)
    $    +EZSX1(3,J+1,K)-2.*EZSX1(3,J,K)+EZSX1(3,J-1,K))
    $    +CXFZD*(EZSX1(4,J,K+1)-2.*EZSX1(4,J,K)+EZSX1(4,J,K-1)
    $    +EZSX1(3,J,K+1)-2.*EZSX1(3,J,K)+EZSX1(3,J,K-1))
 30    CONTINUE
 40   CONTINUE
C
C     NOW SAVE PAST VALUES
C
     DO 60 K=1,NZ1
      DO 50 J=2,NY1
       EZSX2(1,J,K)=EZSX1(1,J,K)
       EZSX2(2,J,K)=EZSX1(2,J,K)
       EZSX2(3,J,K)=EZSX1(3,J,K)
       EZSX2(4,J,K)=EZSX1(4,J,K)
       EZSX1(1,J,K)=EZS(1,J,K)
       EZSX1(2,J,K)=EZS(2,J,K)
       EZSX1(3,J,K)=EZS(NX1,J,K)
       EZSX1(4,J,K)=EZS(NX,J,K)
 50    CONTINUE
 60   CONTINUE
```

```
C
      RETURN
      END
C     *****************************************************
      SUBROUTINE RADEYX
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 K=2,NZ1
        J=1
        EYS(1,J,K)=EYSX1(2,J,K)+CXD*(EYS(2,J,K)-EYSX1(1,J,K))
        EYS(NX,J,K)=EYSX1(3,J,K)+CXU*(EYS(NX1,J,K)-EYSX1(4,J,K))
        J=NY1
        EYS(1,J,K)=EYSX1(2,J,K)+CXD*(EYS(2,J,K)-EYSX1(1,J,K))
        EYS(NX,J,K)=EYSX1(3,J,K)+CXU*(EYS(NX1,J,K)-EYSX1(4,J,K))
 10     CONTINUE
C
      DO 20 J=2,NY1-1
        K=2
        EYS(1,J,K)=EYSX1(2,J,K)+CXD*(EYS(2,J,K)-EYSX1(1,J,K))
        EYS(NX,J,K)=EYSX1(3,J,K)+CXU*(EYS(NX1,J,K)-EYSX1(4,J,K))
        K=NZ1
        EYS(1,J,K)=EYSX1(2,J,K)+CXD*(EYS(2,J,K)-EYSX1(1,J,K))
        EYS(NX,J,K)=EYSX1(3,J,K)+CXU*(EYS(NX1,J,K)-EYSX1(4,J,K))
 20     CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
      DO 40 K=3,NZ1-1
        DO 30 J=2,NY1-1
          EYS(1,J,K)=-EYSX2(2,J,K)+CXD*(EYS(2,J,K)+EYSX2(1,J,K))
     $    +CXX*(EYSX1(1,J,K)+EYSX1(2,J,K))
     $    +CXFYD*(EYSX1(1,J+1,K)-2.*EYSX1(1,J,K)+EYSX1(1,J-1,K)
     $    +EYSX1(2,J+1,K)-2.*EYSX1(2,J,K)+EYSX1(2,J-1,K))
     $    +CXFZD*(EYSX1(1,J,K+1)-2.*EYSX1(1,J,K)+EYSX1(1,J,K-1)
     $    +EYSX1(2,J,K+1)-2.*EYSX1(2,J,K)+EYSX1(2,J,K-1))
          EYS(NX,J,K)=-EYSX2(3,J,K)+CXD*(EYS(NX1 ,J,K)+EYSX2( 4,J,K))
     $    +CXX*(EYSX1(4,J,K)+EYSX1(3,J,K))
     $    +CXFYD*(EYSX1(4,J+1,K)-2.*EYSX1(4,J,K)+EYSX1(4,J-1,K)
     $    +EYSX1(3,J+1,K)-2.*EYSX1(3,J,K)+EYSX1(3,J-1,K))
     $    +CXFZD*(EYSX1(4,J,K+1)-2.*EYSX1(4,J,K)+EYSX1(4,J,K-1)
     $    +EYSX1(3,J,K+1)-2.*EYSX1(3,J,K)+EYSX1(3,J,K-1))
 30       CONTINUE
 40     CONTINUE
C
C     NOW SAVE PAST VALUES
C
```

```
      DO 60 K=2,NZ1
        DO 50 J=1,NY1
          EYSX2(1,J,K)=EYSX1(1,J,K)
          EYSX2(2,J,K)=EYSX1(2,J,K)
          EYSX2(3,J,K)=EYSX1(3,J,K)
          EYSX2(4,J,K)=EYSX1(4,J,K)
          EYSX1(1,J,K)=EYS(1,J,K)
          EYSX1(2,J,K)=EYS(2,J,K)
          EYSX1(3,J,K)=EYS(NX1,J,K)
          EYSX1(4,J,K)=EYS(NX,J,K)
 50       CONTINUE
 60     CONTINUE
C
      RETURN
      END
C     ****************************************************
      SUBROUTINE RADEZY
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 K=1,NZ1
        I=2
        EZS(I,1,K)=EZSY1(I,2,K)+CYD*(EZS(I,2,K)-EZSY1(I,1,K))
        EZS(I,NY,K)=EZSY1(I,3,K)+CYD*(EZS(I,NY1,K)-EZSY1(I,4,K))
        I=NX1
        EZS(I,1,K)=EZSY1(I,2,K)+CYD*(EZS(I,2,K)-EZSY1(I,1,K))
        EZS(I,NY,K)=EZSY1(I,3,K)+CYD*(EZS(I,NY1,K)-EZSY1(I,4,K))
 10     CONTINUE
C
      DO 20 I=3,NX1-1
        K=1
        EZS(I,1,K)=EZSY1(I,2,K)+CYD*(EZS(I,2,K)-EZSY1(I,1,K))
        EZS(I,NY,K)=EZSY1(I,3,K)+CYD*(EZS(I,NY1,K)-EZSY1(I,4,K))
        K=NZ1
        EZS(I,1,K)=EZSY1(I,2,K)+CYD*(EZS(I,2,K)-EZSY1(I,1,K))
        EZS(I,NY,K)=EZSY1(I,3,K)+CYD*(EZS(I,NY1,K)-EZSY1(I,4,K))
 20     CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
      DO 40 K=2,NZ1-1
        DO 30 I=3,NX1-1
          EZS(I,1,K)=-EZSY2(I,2,K)+CYD*(EZS(I,2,K)+EZSY2(I,1,K))
     $    +CYY*(EZSY1(I,1,K)+EZSY1(I,2,K))
     $    +CYFXD*(EZSY1(I+1,1,K)-2.*EZSY1(I,1,K)+EZSY1(I-1,1,K)
     $    +EZSY1(I+1,2,K)-2.*EZSY1(I,2,K)+EZSY1(I-1,2,K))
     $    +CYFZD*(EZSY1(I,1,K+1)-2.*EZSY1(I,1,K)+EZSY1(I,1,K-1)
```

```
     $    +EZSY1(I,2,K+1)-2.*EZSY1(I,2,K)+EZSY1(I,2,K-1))
          EZS(I,NY,K)=-EZSY2(I,3,K)+CYD*(EZS(I,NY1,K)+EZSY2(I,4,K))
     $    +CYY*(EZSY1(I,4,K)+EZSY1(I,3,K))
     $    +CYFXD*(EZSY1(I+1,4,K)-2.*EZSY1(I,4,K)+EZSY1(I-1,4,K)
     $    +EZSY1(I+1,3,K)-2.*EZSY1(I,3,K)+EZSY1(I-1,3,K))
     $    +CYFZD*(EZSY1(I,4,K+1)-2.*EZSY1(I,4,K)+EZSY1(I,4,K-1)
     $    +EZSY1(I,3,K+1)-2.*EZSY1(I,3,K)+EZSY1(I,3,K-1))
 30       CONTINUE
 40     CONTINUE
C
C     NOW SAVE PAST VALUES
C
      DO 60 K=1,NZ1
        DO 50 I=2,NX1
          EZSY2(I,1,K)=EZSY1(I,1,K)
          EZSY2(I,2,K)=EZSY1(I,2,K)
          EZSY2(I,3,K)=EZSY1(I,3,K)
          EZSY2(I,4,K)=EZSY1(I,4,K)
          EZSY1(I,1,K)=EZS(I,1,K)
          EZSY1(I,2,K)=EZS(I,2,K)
          EZSY1(I,3,K)=EZS(I,NY1,K)
          EZSY1(I,4,K)=EZS(I,NY,K)
 50       CONTINUE
 60     CONTINUE
C
      RETURN
      END
C     ********************************************************
      SUBROUTINE RADEXY
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 K=2,NZ1
        I=1
        EXS(I,1,K)=EXSY1(I,2,K)+CYD*(EXS(I,2,K)-EXSY1(I,1,K))
        EXS(I,NY,K)=EXSY1(I,3,K)+CYD*(EXS(I,NY1,K)-EXSY1(I,4,K))
        I=NX1
        EXS(I,1,K)=EXSY1(I,2,K)+CYD*(EXS(I,2,K)-EXSY1(I,1,K))
        EXS(I,NY,K)=EXSY1(I,3,K)+CYD*(EXS(I,NY1,K)-EXSY1(I,4,K))
  10    CONTINUE
C
      DO 20 I=2,NX1-1
        K=2
        EXS(I,1,K)=EXSY1(I,2,K)+CYD*(EXS(I,2,K)-EXSY1(I,1,K))
        EXS(I,NY,K)=EXSY1(I,3,K)+CYD*(EXS(I,NY1,K)-EXSY1(I,4,K))
        K=NZ1
        EXS(I,1,K)=EXSY1(I,2,K)+CYD*(EXS(I,2,K)-EXSY1(I,1,K))
```

```
      EXS(I,NY,K)=EXSY1(I,3,K)+CYD*(EXS(I,NY1,K)-EXSY1(I,4,K))
 20   CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
      DO 40 K=3,NZ1-1
        DO 30 I=2,NX1-1
          EXS(I,1,K)=-EXSY2(I,2,K)+CYD*(EXS(I,2,K)+EXSY2(I,1,K))
     $    +CYY*(EXSY1(I,1,K)+EXSY1(I,2,K))
     $    +CYFXD*(EXSY1(I+1,1,K)-2.*EXSY1(I,1,K)+EXSY1(I-1,1,K)
     $    +EXSY1(I+1,2,K)-2.*EXSY1(I,2,K)+EXSY1(I-1,2,K))
     $    +CYFZD*(EXSY1(I,1,K+1)-2.*EXSY1(I,1,K)+EXSY1(I,1,K-1)
     $    +EXSY1(I,2,K+1)-2.*EXSY1(I,2,K)+EXSY1(I,2,K-1))
          EXS(I,NY,K)=-EXSY2(I,3,K)+CYD*(EXS(I,NY1,K)+EXSY2(I,4,K))
     $    +CYY*(EXSY1(I,4,K)+EXSY1(I,3,K))
     $    +CYFXD*(EXSY1(I+1,4,K)-2.*EXSY1(I,4,K)+EXSY1(I-1,4,K)
     $    +EXSY1(I+1,3,K)-2.*EXSY1(I,3,K)+EXSY1(I-1,3,K))
     $    +CYFZD*(EXSY1(I,4,K+1)-2.*EXSY1(I,4,K)+EXSY1(I,4,K-1)
     $    +EXSY1(I,3,K+1)-2.*EXSY1(I,3,K)+EXSY1(I,3,K-1))
 30     CONTINUE
 40   CONTINUE
C
C     NOW SAVE PAST VALUES
C
      DO 60 K=2,NZ1
        DO 50 I=1,NX1
          EXSY2(I,1,K)=EXSY1(I,1,K)
          EXSY2(I,2,K)=EXSY1(I,2,K)
          EXSY2(I,3,K)=EXSY1(I,3,K)
          EXSY2(I,4,K)=EXSY1(I,4,K)
          EXSY1(I,1,K)=EXS(I,1,K)
          EXSY1(I,2,K)=EXS(I,2,K)
          EXSY1(I,3,K)=EXS(I,NY1,K)
          EXSY1(I,4,K)=EXS(I,NY,K)
 50     CONTINUE
 60   CONTINUE
C
      RETURN
      END
C     ****************************************************
      SUBROUTINE RADEXZ
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 J=2,NY1
        I=1
        EXS(I,J,1)=EXSZ1(I,J,2)+CZD*(EXS(I,J,2)-EXSZ1(I,J,1))
```

```
      EXS(I,J,NZ)=EXSZ1(I,J,3)+CZD*(EXS(I,J,NZ1)-EXSZ1(I,J,4))
      I=NX1
      EXS(I,J,1)=EXSZ1(I,J,2)+CZD*(EXS(I,J,2)-EXSZ1(I,J,1))
      EXS(I,J,NZ)=EXSZ1(I,J,3)+CZD*(EXS(I,J,NZ1)-EXSZ1(I,J,4))
 10   CONTINUE
C
     DO 20 I=2,NX1-1
      J=2
      EXS(I,J,1)=EXSZ1(I,J,2)+CZD*(EXS(I,J,2)-EXSZ1(I,J,1))
      EXS(I,J,NZ)=EXSZ1(I,J,3)+CZD*(EXS(I,J,NZ1)-EXSZ1(I,J,4))
      J=NY1
      EXS(I,J,1)=EXSZ1(I,J,2)+CZD*(EXS(I,J,2)-EXSZ1(I,J,1))
      EXS(I,J,NZ)=EXSZ1(I,J,3)+CZD*(EXS(I,J,NZ1)-EXSZ1(I,J,4))
 20   CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
     DO 40 J=3,NY1-1
      DO 30 I=2,NX1-1
       EXS(I,J,1)=-EXSZ2(I,J,2)+CZD*(EXS(I,J,2)+EXSZ2(I,J,1))
    $  +CZZ*(EXSZ1(I,J,1)+EXSZ1(I,J,2))
    $  +CZFXD*(EXSZ1(I+1,J,1)-2.*EXSZ1(I,J,1)+EXSZ1(I-1,J,1)
    $  +EXSZ1(I+1,J,2)-2.*EXSZ1(I,J,2)+EXSZ1(I-1,J,2))
    $  +CZFYD*(EXSZ1(I,J+1,1)-2.*EXSZ1(I,J,1)+EXSZ1(I,J-1,1)
    $  +EXSZ1(I,J+1,2)-2.*EXSZ1(I,J,2)+EXSZ1(I,J-1,2))
       EXS(I,J,NZ)=-EXSZ2(I,J,3)+CZD*(EXS(I,J,NZ1)+EXSZ2(I,J,4))
    $  +CZZ*(EXSZ1(I,J,4)+EXSZ1(I,J,3))
    $  +CZFXD*(EXSZ1(I+1,J,4)-2.*EXSZ1(I,J,4)+EXSZ1(I-1,J,4)
    $  +EXSZ1(I+1,J,3)-2.*EXSZ1(I,J,3)+EXSZ1(I-1,J,3))
    $  +CZFYD*(EXSZ1(I,J+1,4)-2.*EXSZ1(I,J,4)+EXSZ1(I,J-1,4)
    $  +EXSZ1(I,J+1,3)-2.*EXSZ1(I,J,3)+EXSZ1(I,J-1,3))
 30    CONTINUE
 40   CONTINUE
C
C     NOW SAVE PAST VALUES
C
     DO 60 J=2,NY1
      DO 50 I=1,NX1
       EXSZ2(I,J,1)=EXSZ1(I,J,1)
       EXSZ2(I,J,2)=EXSZ1(I,J,2)
       EXSZ2(I,J,3)=EXSZ1(I,J,3)
       EXSZ2(I,J,4)=EXSZ1(I,J,4)
       EXSZ1(I,J,1)=EXS(I,J,1)
       EXSZ1(I,J,2)=EXS(I,J,2)
       EXSZ1(I,J,3)=EXS(I,J,NZ1)
       EXSZ1(I,J,4)=EXS(I,J,NZ)
 50    CONTINUE
 60   CONTINUE
C
     RETURN
```

```
      END
C     ******************************************************
      SUBROUTINE RADEYZ
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     DO EDGES WITH FIRST ORDER ORBC
C
      DO 10 J=1,NY1
       I=2
       EYS(I,J,1)=EYSZ1(I,J,2)+CZD*(EYS(I,J,2)-EYSZ1(I,J,1))
       EYS(I,J,NZ)=EYSZ1(I,J,3)+CZD*(EYS(I,J,NZ1)-EYSZ1(I,J,4))
       I=NX1
       EYS(I,J,1)=EYSZ1(I,J,2)+CZD*(EYS(I,J,2)-EYSZ1(I,J,1))
       EYS(I,J,NZ)=EYSZ1(I,J,3)+CZD*(EYS(I,J,NZ1)-EYSZ1(I,J,4))
 10   CONTINUE
C
      DO 20 I=3,NX1-1
       J=1
       EYS(I,J,1)=EYSZ1(I,J,2)+CZD*(EYS(I,J,2)-EYSZ1(I,J,1))
       EYS(I,J,NZ)=EYSZ1(I,J,3)+CZD*(EYS(I,J,NZ1)-EYSZ1(I,J,4))
       J=NY1
       EYS(I,J,1)=EYSZ1(I,J,2)+CZD*(EYS(I,J,2)-EYSZ1(I,J,1))
       EYS(I,J,NZ)=EYSZ1(I,J,3)+CZD*(EYS(I,J,NZ1)-EYSZ1(I,J,4))
 20   CONTINUE
C
C     NOW DO 2ND ORDER ORBC ON REMAINING PORTIONS OF FACES
C
      DO 40 J=2,NY1-1
       DO 30 I=3,NX1-1
        EYS(I,J,1)=-EYSZ2(I,J,2)+CZD*(EYS(I,J,2)+EYSZ2(I,J,1))
     $   +CZZ*(EYSZ1(I,J,1)+EYSZ1(I,J,2))
     $   +CZFXD*(EYSZ1(I+1,J,1)-2.*EYSZ1(I,J,1)+EYSZ1(I-1,J,1)
     $   +EYSZ1(I+1,J,2)-2.*EYSZ1(I,J,2)+EYSZ1(I-1,J,2))
     $   +CZFYD*(EYSZ1(I,J+1,1)-2.*EYSZ1(I,J,1)+EYSZ1(I,J-1,1)
     $   +EYSZ1(I,J+1,2)-2.*EYSZ1(I,J,2)+EYSZ1(I,J-1,2))
        EYS(I,J,NZ)=-EYSZ2(I,J,3)+CZD*(EYS(I,J,NZ1)+EYSZ2(I,J,4))
     $   +CZZ*(EYSZ1(I,J,4)+EYSZ1(I,J,3))
     $   +CZFXD*(EYSZ1(I+1,J,4)-2.*EYSZ1(I,J,4)+EYSZ1(I-1,J,4)
     $   +EYSZ1(I+1,J,3)-2.*EYSZ1(I,J,3)+EYSZ1(I-1,J,3))
     $   +CZFYD*(EYSZ1(I,J+1,4)-2.*EYSZ1(I,J,4)+EYSZ1(I,J-1,4)
     $   +EYSZ1(I,J+1,3)-2.*EYSZ1(I,J,3)+EYSZ1(I,J-1,3))
 30    CONTINUE
 40   CONTINUE
C
C     NOW SAVE PAST VALUES
C
      DO 60 J=1,NY1
       DO 50 I=2,NX1
```

```
        EYSZ2(I,J,1)=EYSZ1(I,J,1)
        EYSZ2(I,J,2)=EYSZ1(I,J,2)
        EYSZ2(I,J,3)=EYSZ1(I,J,3)
        EYSZ2(I,J,4)=EYSZ1(I,J,4)
        EYSZ1(I,J,1)=EYS(I,J,1)
        EYSZ1(I,J,2)=EYS(I,J,2)
        EYSZ1(I,J,3)=EYS(I,J,NZ1)
        EYSZ1(I,J,4)=EYS(I,J,NZ)
 50     CONTINUE
 60   CONTINUE
C
      RETURN
      END
C     ****************************************************************
      SUBROUTINE HXSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE UPDATES THE HX SCATTERED FIELD COMPONENTS
C
      DO 30 K=1,NZ1
        DO 20 J=1,NY1
          DO 10 I=2,NX1
            HXS(I,J,K)=HXS(I,J,K)-(EZS(I,J+1,K)-EZS(I,J,K))*DTMDY
     $                    +(EYS(I,J,K+1)-EYS(I,J,K))*DTMDZ
 10         CONTINUE
 20       CONTINUE
 30     CONTINUE
C
      RETURN
      END
C     ********************************************************
      SUBROUTINE HYSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE UPDATES THE HY SCATTERED FIELD COMPONENTS
C
      DO 30 K=1,NZ1
        DO 20 J=2,NY1
          DO 10 I=1,NX1
            HYS(I,J,K)=HYS(I,J,K)-(EXS(I,J,K+1)-EXS(I,J,K))*DTMDZ
     $                    +(EZS(I+1,J,K)-EZS(I,J,K))*DTMDX
 10         CONTINUE
 20       CONTINUE
 30     CONTINUE
C
      RETURN
```

```
      END
C     ******************************************************
      SUBROUTINE HZSFLD
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS SUBROUTINE UPDATES THE HZ SCATTERED FIELD COMPONENTS
C
      DO 30 K=2,NZ1
        DO 20 J=1,NY1
          DO 10 I=1,NX1
            HZS(I,J,K)=HZS(I,J,K)-(EYS(I+1,J,K)-EYS(I,J,K))*DTMDX
     $                 +(EXS(I,J+1,K)-EXS(I,J,K))*DTMDY
 10       CONTINUE
 20     CONTINUE
 30   CONTINUE
C
      RETURN
      END
C     ****************************************************************
      SUBROUTINE DATSAV
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C ------------------------------------------------------------------
C     To completely specifiy a sampling point, the user must specify
C       4 parameters:  NTYPE, IOBS, JOBS, and KOBS.
C       NTYPE for point L is specified by NTYPE(L).
C         NTYPE controls what quantities are sampled as follows:
C           1 = EXS (scattered x-component of electric field)
C           2 = EYS (scattered y-component of electric field)
C           3 = EZS (scattered z-component of electric field)
C           4 = HXS (scattered x-component of magnetic field)
C           5 = HYS (scattered y-component of magnetic field)
C           6 = HZS (scattered z-component of magnetic field)
C           7 = IX (x-component of current through rectangular loop of H)
C           8 = IY (y-component of current through rectangular loop of H)
C           9 = IZ (z-component of current through rectangular loop of H)
C       IOBS(L), JOBS(L) AND KOBS(L) specify the I, J and K coordinates
C         of the point L.  Thus for point 1, the user would specify
C         NTYPE(1), IOBS(1), JOBS(1) and KOBS(1).  For point 2, the
C         user would specify NTYPE(2), IOBS(2), JOBS(2), KOBS(2).
C         The exact locations of a component within a cell follows that
C         of the Yee cell.
C       As an example, the following 4 lines save EYS at the point
C         (33,26,22):
C           NTYPE(1)=2
C           IOBS(1)=33
```

```
C          JOBS(1)=26
C          KOBS(1)=22
C
C     To sample total field, issue a statement to add in the appropriate
C       incident field.  These total field save statements should be added
C       directly after the "3000 CONTINUE" statement near the end of this
C       SUBPROGRAM to override any previous assignment to STORE(NPT).
C       Make certain to back up the incident field by 1 time step (DT)
C       for sampling total E fields and by 1/2 time step (DT/2.0) for
C       sampling total H fields.  For example, to store total EX field
C       at sample point 3, issue the following 3 statements:
C         T=T-DT
C         STORE(3)=EXS(IOBS(3),JOBS(3),KOBS(3))+
C    2      EXI(IOBS(3),JOBS(3),KOBS(3))
C         T=T+DT
C ---------------------------------------------------------------------
C
C  For the first time through (N=1), do some initializations...
      IF (N.NE.1) GO TO 10
C
C   User-defined test point cell location(s).  There should be NTEST
C     occurences of NTYPE and [I,J,K]OBS defined.  Unless total fields
C     are desired, this is probably the only part of this SUBROUTINE
C     the user will need to change.
C
C
      NTYPE(1)=1
      NTYPE(2)=1
      NTYPE(3)=5
      NTYPE(4)=5
C
C     DEFINE NTEST TEST POINT CELL LOCATION(S) HERE
C
      IOBS(1)=17
      JOBS(1)=5
      KOBS(1)=7
      IOBS(2)=17
      JOBS(2)=18
      KOBS(2)=18
      IOBS(3)=17
      JOBS(3)=18
      KOBS(3)=24
      IOBS(4)=17
      JOBS(4)=18
      KOBS(4)=17
C
C   Initialize some variables.
      DO 5 II=1,NTEST
        STORE (II)=0.
 5    CONTINUE
```

```
      NPTS=NTEST
C
C   Print header info and check [I,J,K]OBS to see that they are in
C     range.  Write header to data file and to DIAGS3D.DAT.
c      WRITE (10,1400) DELX,DELY,DELZ,DT,NSTOP,NPTS
      WRITE (17,1500) NPTS
 1400 FORMAT (T2,E12.5,2X,E12.5,2X,E12.5,3X,E14.7,3X,I6,3X,I3)
 1500 FORMAT (T2,'QUANTITIES SAMPLED AND SAVED AT',I4, ' LOCATIONS',/)
      DO 1700 NPT = 1,NPTS
        WRITE (17,1800) NPT,NTYPE(NPT),IOBS(NPT),JOBS(NPT),KOBS(NPT)
        IF (IOBS(NPT).GT.NX) GO TO 1600
        IF (IOBS(NPT).LT.1) GO TO 1600
        IF (JOBS(NPT).GT.NY) GO TO 1600
        IF (JOBS(NPT).LT.1) GO TO 1600
        IF (KOBS(NPT).GT.NZ) GO TO 1600
        IF (KOBS(NPT).LT.1) GO TO 1600
        GO TO 1700
 1600   WRITE (17,*) 'ERROR IN IOBS, JOBS OR KOBS FOR SAMPLING'
        WRITE (17,*) 'POINT ',NPT
        WRITE (17,*) 'EXECUTION HALTED.'
        CLOSE (UNIT=10)
        CLOSE (UNIT=17)
        CLOSE (UNIT=30)
        STOP
 1700 CONTINUE
 1800 FORMAT (T2,'SAMPLE:',I4,',  NTYPE=',I4,', SAMPLED AT CELL I=',
     2  I4,', J=',I4,', K=',I4,/)
C
C   Check for ERROR in NTYPE specification.
      DO 20 NPT=1,NTEST
        IF ((NTYPE(NPT).GE.10).OR.(NTYPE(NPTS).LE.0)) THEN
          WRITE (17,*) 'ERROR IN NYPE FOR SAMPLING POINT ',NPT
          WRITE (17,*) 'EXECUTION HALTED.'
          CLOSE (UNIT=10)
          CLOSE (UNIT=17)
          CLOSE (UNIT=30)
          STOP
        ENDIF
 20   CONTINUE
C  Finished with initialization.
C
C  Cycle through all NTEST sample locations.
 10   DO 3000 NPT=1,NTEST
C
C   Put desired quantity for this sample location into STORE(NPT).
      I=IOBS(NPT)
      J=JOBS(NPT)
      K=KOBS(NPT)
C    Scattered fields
      IF (NTYPE(NPT).EQ.1) STORE(NPT)=EXS(I,J,K)
```

```
      IF (NTYPE(NPT).EQ.2) STORE(NPT)=EYS(I,J,K)
      IF (NTYPE(NPT).EQ.3) STORE(NPT)=EZS(I,J,K)
      IF (NTYPE(NPT).EQ.4) STORE(NPT)=HXS(I,J,K)
      IF (NTYPE(NPT).EQ.5) STORE(NPT)=HYS(I,J,K)
      IF (NTYPE(NPT).EQ.6) STORE(NPT)=HZS(I,J,K)
C    Current loops
C     X-directed current
      IF (NTYPE(NPT).EQ.7) THEN
        STORE(NPT)=0.
        DO 711 KK=K,K+1
          DO 710 JJ=J,J+1
            STORE(NPT)=STORE(NPT)+(-HYS(I,JJ,KK)+HYS(I,JJ,KK-1))*DELY+
     2      (HZS(I,JJ,KK)-HZS(I,JJ-1,KK))*DELZ
 710      CONTINUE
 711    CONTINUE
      ENDIF
C     Y-directed current
      IF (NTYPE(NPT).EQ.8) THEN
        STORE(NPT)=0.
        DO 811 KK=K,K+1
          DO 810 II=I,I+1
            STORE(NPT)=STORE(NPT)+(-HZS(II,J,KK)+HZS(II-1,J,KK))*DELZ+
     2      (HXS(II,J,KK)-HXS(II,J,KK-1))*DELX
 810      CONTINUE
 811    CONTINUE
      ENDIF
C     Z-directed current
      IF (NTYPE(NPT).EQ.9) THEN
        STORE(NPT)=0.
        DO 911 JJ=J,J+1
          DO 910 II=I,I+1
            STORE(NPT)=STORE(NPT)+(-HXS(II,JJ,K)+HXS(II,JJ-1,K))*DELX+
     2      (HYS(II,JJ,K)-HYS(II-1,JJ,K))*DELY
 910      CONTINUE
 911    CONTINUE
      ENDIF
C
 3000 CONTINUE
C
C   If total fields are desired, place total field save statements
C     here.  Example for saving total EX field at sample point 3:
C     T=T-DT
C     STORE(3)=EXS(IOBS(3),JOBS(3),KOBS(3))+
C    2  EXI(IOBS(3),JOBS(3),KOBS(3))
C     T=T+DT
C
C   Write sampled values to disk
c     WRITE (10,*) (STORE(II),II=1,NPTS)
```

```
        a=5
        b=7

      IF (N.EQ.33) THEN
C       do 4 J=1,NY
        do 9 I=1,NX
      WRITE (9,*) EXS(I,b,a)
9       continue
C4      continue
        ENDIF

      IF (N.EQ.87) THEN
C       do 4 J=1,NY
        do 7 I=1,NX
      WRITE (7,*) EXS(I,b,a)
7       continue
C4      continue
        ENDIF

      IF (N.EQ.140) THEN
C       do 4 J=1,NY
        do 4 I=1,NX
      WRITE (5,*) EXS(I,b,a)
4       continue

C4      continue
        ENDIF

      IF (N.EQ.183) THEN
C       do 4 J=1,NY
        do 3 I=1,NX
      WRITE (3,*) EXS(I,b,a)
3       continue
C4      continue
        ENDIF


      IF (N.EQ.250) THEN
C       do 4 J=1,NY
        do 1 I=1,NX
      WRITE (1,*) EXS(a,b,I)
1       continue
C4      continue
        ENDIF



C
      RETURN
      END
```

```
C     ***************************************************************
      FUNCTION EXI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS FUNCTION COMPUTES THE X COMPONENT OF THE INCIDENT
C     ELECTRIC FIELD
C
      DIST=((I-1)*DELX+0.5*DELX*OFF)*XDISP+((J-1)*DELY)*YDISP+
     $((K-1)*DELZ)*ZDISP + DELAY
      EXI = AMPX*SOURCE(DIST)
      RETURN
      END
C     ************************************************************
      FUNCTION EYI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS FUNCTION COMPUTES THE Y COMPONENT OF THE INCIDENT
C     ELECTRIC FIELD
C
      DIST=((I-1)*DELX)*XDISP+((J-1)*DELY+0.5*DELY*OFF)*
     $YDISP+((K-1)*DELZ)*ZDISP + DELAY
      EYI = AMPY*SOURCE(DIST)
      RETURN
      END
C     *********************************************************
      FUNCTION EZI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS FUNCTION COMPUTES THE Z COMPONENT OF THE INCIDENT
C     ELECTRIC FIELD
C
      DIST=((I-1)*DELX)*XDISP+((J-1)*DELY)*YDISP+((K-1)*DELZ+
     $0.5*DELZ*OFF)*ZDISP + DELAY
      EZI = AMPZ*SOURCE(DIST)
      RETURN
      END
C     *****************************************************
      FUNCTION SOURCE(DIST)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     THIS FUNCTION DEFINES THE FUNCTIONAL FORM OF THE INCIDENT
C     FIELD
```

```
C
      SOURCE = 0.0
C
C     EXECUTE FOLLOWING FOR INCIDENT E FIELD
C
      TAU=T-DIST/C
      IF(TAU.LT.0.0) GO TO 10
      IF (TAU.GT.PERIOD) GO TO 10
C
C         Sinusoidal
      SOURCE=EXP(-ALPHA*((TAU-BETADT)**2))
      SOURCE=SOURCE*SIN(6.28*10**9*(TAU-BETADT))
c         Gaussian
c     SOURCE=EXP(-ALPHA*((TAU-BETADT)**2))
C
C
 10   RETURN
      END
C     *************************************************************
      FUNCTION DEXI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     TIME DERIVATIVE OF INCIDENT EX FIELD FOR DIELECTRICS
C
      DIST=((I-1)*DELX+0.5*DELX*OFF)*XDISP+((J-1)*DELY)*YDISP+
     $((K-1)*DELZ)*ZDISP + DELAY
      DEXI=AMPX*DSRCE(DIST)
C
      RETURN
      END
C     *************************************************************
      FUNCTION DEYI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C     TIME DERIVATIVE OF INCIDENT EY FIELD FOR DIELECTRICS
C
      DIST=((I-1)*DELX)*XDISP+((J-1)*DELY+0.5*DELY*OFF)*YDISP+
     $((K-1)*DELZ)*ZDISP + DELAY
      DEYI=AMPY*DSRCE(DIST)
      RETURN
      END
C     *******************************************************
      FUNCTION DEZI(I,J,K)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
```

```
C
C    TIME DERIVATIVE OF INCIDENT EZ FIELD FOR DIELECTRICS
C
      DIST=((I-1)*DELX)*XDISP+((J-1)*DELY)*YDISP+((K-1)*DELZ+
     $0.5*DELZ*OFF)*ZDISP + DELAY
      DEZI=AMPZ*DSRCE(DIST)
      RETURN
      END
C    ***************************************************************
      FUNCTION DSRCE(DIST)
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C    TIME DERIVATIVE OF INCIDENT FIELD FOR DIELECTRICS
C
      DSRCE=0.0
C
      TAU=T-DIST/C
      IF(TAU.LT.0.0) GO TO 10
      IF (TAU.GT.PERIOD) GO TO 10
C
        DSRCE=EXP(-ALPHA*((TAU-BETADT)**2))*(-2.*ALPHA*(TAU-BETADT))
C
C
 10   RETURN
      END
C    ***************************************************************
      SUBROUTINE ZERO
C
      INCLUDE 'COMMONA.FOR'
C$INCLUDE COMMONA
C
C    THIS SUBROUTINE INITIALIZES VARIOUS ARRAYS AND CONSTANTS TO
ZERO.
C
      T=0.0
      DO 30 K=1,NZ
        DO 20 J=1,NY
          DO 10 I=1,NX
            EXS(I,J,K)=0.0
            EYS(I,J,K)=0.0
            EZS(I,J,K)=0.0
            HXS(I,J,K)=0.0
            HYS(I,J,K)=0.0
            HZS(I,J,K)=0.0
            IDONE(I,J,K)=0
            IDTWO(I,J,K)=0
            IDTHRE(I,J,K)=0
 10         CONTINUE
```

```
20     CONTINUE
30   CONTINUE
     DO 60 K=1,NZ1
       DO 50 J=1,NY1
         DO 40 I=1,4
           EYSX1(I,J,K)=0.0
           EYSX2(I,J,K)=0.0
           EZSX1(I,J,K)=0.0
           EZSX2(I,J,K)=0.0
40       CONTINUE
50     CONTINUE
60   CONTINUE
     DO 90 K=1,NZ1
       DO 80 J=1,4
         DO 70 I=1,NX1
           EXSY1(I,J,K)=0.0
           EXSY2(I,J,K)=0.0
           EZSY1(I,J,K)=0.0
           EZSY2(I,J,K)=0.0
70       CONTINUE
80     CONTINUE
90   CONTINUE
     DO 120 K=1,4
       DO 110 J=1,NY1
         DO 100 I=1,NX1
           EXSZ1(I,J,K)=0.0
           EXSZ2(I,J,K)=0.0
           EYSZ1(I,J,K)=0.0
           EYSZ2(I,J,K)=0.0
100      CONTINUE
110    CONTINUE
120  CONTINUE
     DO 130 L=1,9
       ESCTC(L)=0.0
       EINCC(L)=0.0
       EDEVCN(L)=0.0
       ECRLX(L)=0.0
       ECRLY(L)=0.0
       ECRLZ(L)=0.0
130  CONTINUE
     RETURN
     END
```