CONTENTS

ABSTRACT ............................................................................................................. 1
SECTION 1 - SCOPE AND OBJECTIVES ................................................................. 2
SECTION 2 - PROBLEM DEFINITION ..................................................................... 3
  2.1 THE PROBLEM ......................................................................................... 3
  2.2 THE SOLUTION ......................................................................................... 3
  2.3 NUMERICAL IMPLEMENTATION ............................................................... 4
  2.4 PROGRAM ARCHITECTURAL DESCRIPTION ........................................... 11
  2.5 PROGRAM EXECUTION .............................................................................. 14
  2.6 LIMITATIONS ............................................................................................ 19
SECTION 3 - PROGRAMMING INFORMATION ....................................................... 21
  3.1 PROGRAM FLOW ....................................................................................... 21
  3.2 GLOBAL VARIABLES ............................................................................... 28
  3.3 DATA FILES ............................................................................................... 28
  3.4 ERROR MESSAGES .................................................................................... 28
  3.5 SUBROUTINES ........................................................................................... 37
    3.5.1 SUBROUTINE SETUP ......................................................................... 37
    3.5.2 SUBROUTINE BUILD ........................................................................... 52
    3.5.3 SUBROUTINE EADV ......................................................................... 62
    3.5.4 SUBROUTINE EBC ............................................................................ 68
    3.5.5 SUBROUTINE ERAD ......................................................................... 82
    3.5.6 SUBROUTINE HADV ........................................................................ 91
    3.5.7 FIELDS ................................................................................................ 96
    3.5.8 SUBROUTINE SAVESB ....................................................................... 101
    3.5.9 SUBROUTINE OUTBND ..................................................................... 106
    3.5.10 SUBROUTINE DATASAV ................................................................. 114
    3.5.11 SUBROUTINE PRINOUT ................................................................. 136

i
FIGURES - 1. Location of the Six Field Evaluation Points in a Typical Cell ................................ 7
1-a. Unexpanded F-106 Model ............................................. 12
2. Expanded F-106 Model .................................................. 13
2-b. First Pass, Diffusion Calculation ..................................... 23
2-c. Second Pass, Expanded Diffusion Calculation .......................... 24
3. Flow Chart for Subroutine SETUP ...................................... 39
4. Flow Chart for Subroutine BUILD ....................................... 43
5. Flow Chart for Subroutine EADV ....................................... 64
6. Determining E-Field Boundary Condition ................................ 69
7. Generalized Flow Chart of Subroutine EBC ............................... 71
8. Flow Chart for Subroutine EBC, Perfectly Conducting Portion Only .... 72
9. Field Components at the Outer Boundary of the Cell Space ............... 83
10. Flow Chart for Subroutine ERAD ...................................... 88
11. Flow Chart for Subroutine HADV ...................................... 93
12. SAVESB Flow Chart .................................................... 102
13. OUTBND Flow Chart .................................................... 107
14. Flow Chart for Subroutine DATASAV ................................... 115
15. Distribution of Field Component Locations About Test Points .......... 116
16. Example Field Component Arrangement about a Test Point ............... 118
17. Interpolated Field Values about a Test Point .......................... 119
18. Interpolated Field Values Used to Extrapolate to Test Point Location .... 119
19. Graphical Description of THETA ....................................... 120
(Angle defining sensor rotation).
ABSTRACT

The computer code G3DXL computes time domain scattered fields and surface currents and charges induced by a driving function on and within a complex scattering object which may be perfectly conducting or a lossy dielectric. This is accomplished by modeling the object with cells within a three-dimensional, rectangular problem space, enforcing the appropriate boundary conditions and differencing Maxwell's equations in time. In the present version of the program, the driving function can be either the field radiated by a lightning strike or a direct lightning strike. The scattering object is the F-106 B aircraft.

The program has been successfully executed on the CDC 203 at NASA Langley where it is currently operational.
SECTION 1

SCOPE AND OBJECTIVES

This document presents information concerning the programming, maintenance and operation of the computer code G3DXL. This information is intended to facilitate 1) effective use of the program as is and 2) conversion and modification of the code for other test items and excitation sources.

The theoretical basis for the code is discussed, the numerical implementation of the theory is presented, and the limitations are enumerated. This discussion is followed by a presentation of the program architecture, including flow charts, global variable definitions, input and output requirements. A discussion of each individual subroutine is given which includes a subroutine description, local variable definition (where needed), flow chart, and a listing of the subroutine. The reader is provided in this manual with an understanding of the theoretical basis of the code and its practical implementation.
SECTION 2

PROBLEM DEFINITION

2.1 THE PROBLEM

G3DXL addresses the problem of calculating in the time domain the currents and charges induced on and within a composite aircraft, along with the resulting scattering fields, by an excitation source, in this case a direct lightning strike or the radiated lightning field. The scatterer may be imbedded in a homogeneous media (an aircraft in flight, for example) or may be located over a lossy ground plane (an aircraft over a runway, for example).

2.2 THE SOLUTION

G3DXL finds the induced currents and charges on the scatterer along with the scattered fields by differencing Maxwell's equations in time using an algorithm first described by Yee [1]. This algorithm is applied to a problem space with a mesh of dimensions of 29x29x29. Even larger problem spaces on the order of 50x50x50 are possible on certain large computers, such as the DEC VAX 11/780. The problem of echoes from the finite problem space's outer boundaries is circumvented by the application of Merewether's radiating boundary condition [2], which is based on the fact that, far from the scatterer, the scattered fields must behave as $f(\theta,\phi)g(t-r/c)/r$. Here, $f(\theta,\phi)$ represents an arbitrary variation in $\theta$ and $\phi$, $g(t-r/c)$ expresses the retarded time behavior of the fields and the $1/r$ describes the far field dependence on $r$. It is the $1/r$ dependence that is of important here, as it allows fields near the outer boundary to be extrapolated to the outer boundary, thereby affecting the outer radiating boundary condition.
2.3 Numerical Implementation

The three-dimensional, finite-difference formalism employed by G3DXL separates the fields into incident and scattered fields and solves directly for the scattered fields, using Maxwell's equations. The incident field is used only to establish the value of the scattered field throughout the scatterer's volume, where the fields are related by what will be referred to here as a volume boundary condition.

The formalism (Appendix B gives more details) starts with Maxwell's equations for the total fields:

\[ \nabla \times \mathbf{E} = \mu \frac{\partial \mathbf{H}}{\partial t} \]

\[ \nabla \times \mathbf{H} = \varepsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E}. \]

Subtracting the Maxwell equations for the incident field propagating in free space, i.e. the free space form of the Maxwell equations:

\[ \nabla \times \mathbf{E}^i = -\mu_0 \frac{\partial \mathbf{H}^i}{\partial t} \]

\[ \nabla \times \mathbf{H}^i = \varepsilon_0 \frac{\partial \mathbf{E}^i}{\partial t} \]

yields the generalized equations:

\[ \frac{\partial \mathbf{E}^s}{\partial t} = -\nabla \times \mathbf{E}^s - (\mu - \mu_0) \frac{\partial \mathbf{H}^i}{\partial t} \]

\[ \varepsilon \frac{\partial \mathbf{H}^s}{\partial t} + \sigma \mathbf{E}^s = \nabla \times \mathbf{H}^s - \mathbf{E}^i - (\varepsilon - \varepsilon_0) \frac{\partial \mathbf{E}^i}{\partial t} \]
The generalized equations, as they stand, are completely general and may be applied over all space. In free space, the terms with \((\mu-\mu_0), (\varepsilon-\varepsilon_0)\) and \(\sigma\) as multipliers vanish yielding the free space form of the equations. On a perfectly conducting surface, where \(\sigma=0\) the generalized equations reduce to:

\[ E^s_T = -E^i_T \]

where \(T\) refers to the tangential components of the \(E\)-field on the perfectly conducting surface.

Prior work with finite difference codes typically used the free field formulation with the perfectly conducting boundary condition. G3DXL uses the generalized equations throughout the volume of a lossy dielectric scatterer, the free field equations in free space and the perfectly conducting boundary condition for a perfect conductor. This approach was deemed more efficient than the use of the generalized equations throughout all space.

The actual conversion of the differential equations into a finite difference form will be presented here for a perfectly conducting scatterer about which the free space Maxwell equations for the scattered field hold, i.e.,

\[ \nabla \times E^s = -\frac{\partial H^s}{\partial t} \]

\[ \nabla \times H^s = \varepsilon \frac{\partial E^s}{\partial t} + \sigma E^s \]

but are written in a finite-difference form using two point central differencing in a rectangular coordinate system. For example:

\[ \frac{\partial E^\text{scat}}{\partial y} - \frac{\partial E^\text{scat}}{\partial z} = \frac{\partial E^\text{scat}}{\partial t} + \sigma E^\text{scat}_x \]
becomes

\[
\begin{align*}
\left( H_z^{\text{scat}} \right)^n (I,J,K) - \left( H_z^{\text{scat}} \right)^n (I,J-1,K) &= Y(J) - Y(J-1) \\
\left( H_y^{\text{scat}} \right)^n (I,J,K) - \left( H_y^{\text{scat}} \right)^n (I,J,K-1) &= Z(K) - Z(K-1) \\
\left( E_x^{\text{scat}} \right)^{n+1/2} (I,J,K) - \left( E_x^{\text{scat}} \right)^{n-1/2} (I,J,K) &= \frac{\varepsilon}{\Delta t} \\
\left( E_x^{\text{scat}} \right)^{n+1/2} (I,J,K) + \left( E_x^{\text{scat}} \right)^{n-1/2} (I,J,K) &= \frac{\sigma}{\Delta t}
\end{align*}
\]

where \( \Delta t, Y(J) - Y(J-1) \), etc. are discrete time and spatial increments used in the calculation. The superscript \( n \) is the time index and \( I,J,K \) label the nodes in the three-dimensional mesh of the problem space. Similar expressions can be derived for the remaining five equations, involving the remaining components. Note that the \( H \)-field components at times \((n+1)\Delta t\) can be evaluated from their own value at the earlier time \( n\Delta t \) and from the \( E \)-field components at the earlier times \((n+1/2)\Delta t\). Similarly the \( E \)-field components at times \((n+1/2)\Delta t\) are evaluated from the \( H \)-field components at times \( n\Delta t \) and from their own values at earlier times \((n-1/2)\Delta t\). This results in a fully explicit and automatically central differencing scheme that reduces the truncation or round-off error [3,4].

The difference equations are applied, at successive discrete time steps, to a cell space; in this case, a three-dimensional, rectangular cell space. That is, the equations are assumed to hold for all the cells of the space. A single cell and the field components associated with it are shown in Figure 1. The size of the individual cell determines the time step \( \Delta t \). For stability, \( \Delta t \) must satisfy the Courant stability condition, which may be expressed as [5].
Figure 1. Location of the Six Field Evaluation Points in a Typical Cell
Δt = \min_{I,J,K} \{(X^O(I+1) - X^O(I))^2(εμ)^{-1} \}

\ + (Y^O(J+1) - Y^O(J))^2(εμ)^{-1} + (Z^O(K+1) - Z^O(K))^2(εμ)^{-1})^{-1/2}

If ε or μ are inhomogenous, their dependence on (I,J,K) must be included in evaluating the minimal time steps of the above equation. If the time step is greater than Δt, the predicted responses grow without bound. Note that Δt is approximately the transit time across the cell. The cell size also determines the frequency range of valid data. When a quarter wavelength of the frequency component of interest is smaller than the cell dimensions, the answer returned is in error.

Two boundary conditions must be applied to this cell space. The first defines the object for which the scattered fields are to be calculated. We shall only consider a perfect conductor here for simplicity. For each cell face that defines a portion of aircraft surface, the total tangential E-field vanishes so that

\[ E^\scat(t) = -E^\inc(t) \tan(t) \]

At t=0, \( E^\inc(t) \) is assumed zero and so \( E^\scat(t) \) is also zero. At later times the source or driving field \( E^\inc(t) \) appears at the scatterer's surface with non-zero value determining \( E^\scat(t) \) and, through the difference equations, determining the scattered fields throughout the cell space.

The second boundary condition is on the outer surface of the cell space where the radiation condition is imposed [2]. This boundary condition requires the scattered E-field to behave as \( E^\scat(t) = f(θ,ϕ) g(t-r/c)/r \) at the outer surface, the same field behavior as occurs far from the scatter. By imposing this behavior on the surface of the cell space, a finite cell space can be made to approximate an infinite cell space. In effect, the outermost E-
fields are made to behave nearly the same as they would have had the cell space continued on beyond the outermost cells indefinitely, leaving the inner cells seemingly imbedded in an infinite cell space. As a rule of thumb, this approximation is successful when the cell space is at least twice the scatter’s dimensions in each direction.

As an example of this procedure, let $J_m$ be the maximum index in the $y$-coordinate for the cell space. The radiation condition applied to $E_z^{n+1}$ in the $y$-direction is:

$$E_z^{n+1}(I, J_m, K) = \frac{R_z(I, J_m, K)}{2R_z(I, J_m, K)} \times [(\theta_{zy}(\theta_{zy}-1)E_z^{n-1}(I, J_m-1, K)$$

$$+ 2(1-\theta_{zy})E_z^n(I, J_m-1, K) + \theta_{zy}(\theta_{zy}+1)E_z^{n+1}(I, J_m-1, K)]]$$

where

$$R_z(I, J_m, K) = \sqrt{X_o(I)^2 + Y_o(J_m)^2 + Z_o(K)^2}$$

$$\theta_{zy} = 1 - \frac{R_z(I, J_m, K) - R_z(I, J_m-1, K)}{c\Delta t}$$

and $X_o(I)$, $Y_o(J)$, $Z_o(K)$ are coordinate values for the points considered, $c$ is the velocity of light. As can be seen, the fields at the outer boundary are found from interpolating, in time, the fields that are one cell in from the outer boundary. The times are the present program cycle, and the two prior program cycles. This interpolation is based on a constant cell size. If an expanding mesh is used, a somewhat more complex expression must be employed [5].

Similar conditions (there are eleven more) can be written down for $E_x$, $E_y$, and $E_z$ at the six surfaces of the box.
By assigning appropriate constitutive parameters (\(\varepsilon, \mu, \) and \(\sigma\)) to the first few tiers of cells at the base of the cell space, a ground can be modeled from which the scattered fields are partially reflected. Since the radiation condition is applied to all six faces, only the desired ground reflection will be present; the fields transmitted into the ground will not be reflected back from the outer surface of the cell space.

The cell size used need not be uniform. By enlarging the cell size outside the immediate region of the aircraft, the outer boundary can be further removed from the aircraft. (Expansion factors much greater than 1.3 are not recommended. The reason is that severe discontinuity effects can be seen.) Reflections from the outer boundary, that occur because the radiation condition is imposed at moderate distance \(R\) from the origin, are thereby lessened.

The surface current density and surface charge density at any position on the aircraft are found from

\[
\bar{J} = \bar{n} \times (\bar{H}^{\text{scat}} + \bar{H}^{\text{inc}}) \quad \text{and} \quad \rho_s = \bar{E}^{\text{scat}}_{\text{normal}} + \bar{E}^{\text{inc}}_{\text{normal}}
\]

respectively, where \(\bar{n}\) is the outward normal of the scatterer's surface. The values of the scattered fields at the position of interest are found from the adjacent nodal point values using a combination of interpolation and extrapolation. These time-domain responses can be transformed to yield frequency domain data over a broad frequency range.
G3DXL is designed to predict exterior and interior responses on complex scattering objects, such as aircraft. To achieve the needed degree of spatial resolution for interior coupling predictions the expansion technique is employed. A preliminary run with a coarsely modeled aircraft (Figure 1-a) is used to obtain the response about a portion of the aircraft. This response imposed on the outer boundary of the problem space allows a second run to be made in which only the portion of the aircraft is modeled (Figure 2), but in finer detail and with the entire aircraft response preserved. The use of the expansion technique requires what is, in effect, two separate runs.

The scattering object can also be a perfect conductor, a restriction that some prior finite difference codes, such as THREDE [5,6], had to adhere to; or it could be a lossy dielectric, as in a composite aircraft. This last and more general case is treated using the generalized finite difference technique as embodied in G3DXL. A thin plate algorithm (Appendix A) even allows composite panels to be treated. The generalized finite difference technique and thin plate algorithm do not require more runs, but each run using them is more complex. Such a run must be able to treat perfectly conducting solids and surfaces or plates and must be able to treat lossy dielectric solids and plates.
Figure 1-a. Unexpanded F-106B Model
Figure 2. Expanded F-106 Model
2.5 Program Execution

The following steps are a concise guide to the execution of the code.

1) Always maintain two original backup copies: the first, a direct access file resident in the computer's memory and the second, a card deck, the same as the file, resident with the user in charge of maintaining the code.

2) From the original file in the computer, make a new file under a new name. It is this file that will be modified (using the XEDIT capability) to model the correct problem.

3) When the new file is modified and running properly, it is prudent to make a copy of the file. This copy serves as a secondary backup. If further modifications are made and they result in a nonrunning program, one can always return to the secondary backup and try again.

4) Modifications to the original code, as embodied in the original backup, can be made only after the selection of an excitation source and an interaction object.

5) The excitation source can be either a radiated lightning field or a direct lightning strike. To select one or the other, the parameter IDLS is set equal to 0 for a radiated lightning field or 1 for a direct lightning strike. IDLS appears as a data statement in subroutine SETUP.

6) The excitation source specifications are set as follows. For a radiated lightning field a double exponential waveform \( A(e^{-\alpha t} - e^{-\beta t}) \) is assumed. The parameters \( A, \alpha \) and \( \beta \) are represented by AMP, ALPHA and BETA in the code. They are set in subroutine SETUP. Current values are AMP = 5.92E4, ALPHA = 4.08E6, BETA = 1.0E8.
7) The actual calculation of the radiated lightning fields is performed in the functions $E_{INCX}(I,J,K), \ldots , H_{INCZP}(XP,YP,ZP)$. At present the fields are for broadside incidence from above. The electric field points in the $x$ direction and the magnetic field points in the $z$ direction.

8) The interaction object, typically an aircraft, must be modeled within the problem space boundaries set by a space of 28 x 28 x 28 cells. Further, the outer radiating boundary condition can be expected to work reasonably well only if the interaction object occupies, at most, 20 cells across the problem space in any direction. Thus, the interaction object can be divided by no more than 20 in any linear dimension. If one dimension is $L$, then the cell used in the problem space would have dimension $L/20$ in this direction. A typical aircraft is on the order to 20 meters long. Thus, cells approximately 1 meter long are used in the $M=1$ loop where the entire aircraft must be modeled. Generally, cells on the order of $1m \times 0.5m \times 0.5m$ are used. This allows for a reasonably accurate rendition of the aircraft fuselage. Expanding the cells where the wings lie is, however, necessary. This allows all of the wings to be modeled in the problem space at a sacrifice in detail in the region of the wings, where it is often not needed.

9) For the $M=1$ loop, when the entire interaction object must be modeled, a set of scale drawings or the equivalent are needed. They define the interaction object geometry. Additionally, the constitutive parameters ($\sigma$ and $\varepsilon$) of the material comprising the interaction object are needed along with the material thickness. The model is input into the code by:

- setting the cell size ($\Delta x, \Delta y, \Delta z$) through the parameters DELX, DELY and DELZ in subroutine SETUP.
- setting the expansion rates of the cells (XPANX, XPANY, XPANZ) and the indices where cell expansion starts, (IUP, ... KDOWN) in subroutine SETUP.
- setting the NOPE($I,J,K$) array in subroutine BUILD. Nonzero values of NOPE($I,J,K$) place a material media in the ($I,J,K$)th cells. When NOPE($I,J,K$) = 0 the cell has the properties of free space. The option exists to select a perfectly conducting material that fills
the cell (NOPE(I,J,K) = 4) or forms a plate on one side of the cell (NOPE(I,J,K) = 1 for the (Y-Z) plane, 2 for the (X-Z) plane and 3 for the (X-Y) plane). (All sides are the near side of the cell as defined along the x,y,z axis that follow the increase in the indices I, J, and K respectively.) The option also exists to replace the perfectly conducting material with a lossy dielectric material. The same approach is used as for the perfectly conducting material, except 1,2,3,4 are replaced by 5,6,7,8 in the values that may be assigned to NOPE(I,J,K). Cells are set to nonzero values so as to fill the problem with material following the geometry of the interaction object and having the same material properties ($\varepsilon$ and $\sigma$).

- Setting $\varepsilon$ and $\sigma$ for any lossy dielectric material in the interaction model via a data statement for EPS and SIGMA in the subroutine SETUP.

- Setting plate thickness TX for lossy dielectric plates in the (Y-Z) plane, TY for lossy dielectric plates in (X-Z) plane and TZ for lossy dielectric plates in the (X-Y) plane. (Presently EPS, SIGMA, TX, TY and TZ do not vary as a function of position). At a large cost in memory, but within the capabilities of the Cyber 203, these parameters can be input as a function of position, i.e. EPS+EPS(I,J,K), etc. If this change is required, then an array, such as EPS(I,J,K), must be specified in subroutine SETUP and the CDC XEDIT capability used to replace EPS by EPS(I,J,K) everywhere, including common statements. Related quantities to EPS, such as EPEFFX, EPEFFY, EPEFFZ, EXPON, EXPDX, EXPDY, EXPDZ, XEXP, XEXPX, XEXPZ, and to SIGMA, such as RSIGMA, must also be written as three dimensional arrays and dimensioned as such or, more appropriately, written out explicitly in terms of EPX(I,J,K), SIGMA(I,J,K), TX(I,J,K), etc. at the cost of an increased computational burden with the savings of a very large amount of memory.

10) Interaction object models need only be constructed for those loops desired, i.e.

- $M=1$ for unexpanded model where only surface responses are required;

- $M=2$ for expanded model where only aperture coupling and diffusion may be treated;

11) The desired permutation of the two loops is set by MM. Allowed values are MM=1, 12, where, for example, MM=12 results in the $M=1$ and 2 loops being exercised. MM is set in program DRIVER.
12) The number of time steps, \( N \), must also be set in DRIVER. Dimension restrictions limit \( N \) at present to 2000. Further, the expansion factor \( \text{EXPFAC} \) times \( N \) for the \( M=1 \) loop should yield \( N \) for the \( M=2 \) loop. Thus, when an expanded run is made, \( N \leq 2000 \) and \( N \leq 2000/\text{EXPFAC} \) (2000/4 = 500 at present) for unexpanded runs made in conjunction with an expanded run.

13) For expanded runs \((M=2)\) the necessary subboundary information on the preceding unexpanded run \((M=1)\) is saved via subroutine SAVESB and interpolated for use in the expanded run in subroutine OUTBND. Presently, the expansion factor, \( \text{EXPFAC} \), is four, so that the space surrounded by the subboundary on the unexpanded run is \( 7 \times 7 \times 7 \) cells. The use of interpolation only and the positioning of the tangential \( E \) field components on the subboundary faces requires that an array of \( 8 \times 9 \) field components be saved on each face. As there are two components on each face and six faces, a total of 864 components are saved in each time step in the array ARAY, which has dimensions of \((864,500)\). For different expansion factors the 864 would have to change in ARAY, which appears in subroutines SAVESB and OUTBND. Also, \( \text{CRAY}(8,9) \) appearing in subroutine OUTBND for interpolation purposes would have to be changed, as would the auxiliary array ARRAY.

14) The location of the subboundary is set by a data statement in \( \text{INEAR,...,KFar} \) in subroutines SAVESB and OUTBND.

15) Interior regions are typically modeled for the \( M=2 \) loop where sufficient resolution is available. Wires are electrically conducting \((E_\text{scat} = -E^{\text{inc}})\) for \( M=2 \). They are manually specified in EBC.

16) Lightning channels are modeled as exterior wires running from the outer surface of the problem space to the scattering object. They are electrically conducting for \( M=1 \) and 2 and are manually specified in EBC. A loop of \( H \)-field components drive the electrically conducting lightning channel for \( M=1 \) and is specified manually in HADV. For \( M=2 \) the loop is not presented, only the channel. This is because the loop acts as a source and must be outside the subboundary volume for the expansion technique to work.
17) Test point locations are selected manually in DATASAV. The desired locations (up to 24) are specified by XOBS(NPT), YOBS(NPT), ZOBS(NPT), which are the test point locations in meters from the center of the problem space. Setting NPLANE(NPT) for each location determines the surface on which test point is located or, more precisely, the direction from which the fields are extrapolated to the test point. NPLANE=1 selects a test point on a Y-Z plane with the fields extrapolated from left to right; 2 is the same, except the extrapolation is from right to left. NPLANE=3 selects a test point on a X-Z plane with the fields extrapolated from high to low; 4 is the same except the extrapolation is from low to high. NPLANE=5 selects a test point on a X-Y plane with the fields extrapolated from back to fore; 6 is the same except the extrapolation is from fore to back. A test location on a surface can be for an exterior or interior response depending on the selection of NPLANE, which determines from what direction the extrapolation is made. THETA(NPT) aligns the test point response components, that is for THETA=0, HSTOR1 is the axial current for the test point on the sides of a cylinder and HSTOR2 is the circumferential current. When THETA=90, this assignment is reversed. For THETA somewhere in between, HSTOR1, for example, is a mix of axial and circumferential currents. For most applications THETA can be left at 0.

18) To obtain current responses for late times economically, the data records for the HSTOR1 component are extended analytically LMAX * IP time steps. IP is the number of time steps taken before response data is saved. For example, for IP=2 data is saved every other time step. Both LMAX and IP are set via a data statement in subroutine SETUP. For currents on interior wires a loop of magnetic field about the wire can be manually specified and output as CSTORE. NPLANE must be selected so that each component is stored in HSTOR1, which is analytically extended. The, CSTORE is also of extended duration and the late time currents are known.
2.6 LIMITATIONS

There are three limitations on the operation of G3DXL which are imposed by the mathematical model and the computer facilities. These are listed and discussed below.

1) Run time is limited to \( \leq 2000 \) program cycles for a single loop for typical aircraft scattering problems. A program cycle is typically on the order of 1 ns resulting in 500 to 2000 ns of data. Instabilities in the radiation boundary condition return a growing oscillation in the late-time data that overwhelms the true response. This limitation is lessened by maintaining nearly equal cell sizes on the outer boundaries (a condition that is automatically met when using a constant mesh, but not always when using an expanding mesh.) Differences of greater than a factor of two should be avoided.

2) The model is limited in size to be no more than one-half the cell space in any direction. If the model is not restricted to this size limit, the approximation upon which the radiating boundary condition is based, namely that the scattered fields behave as far fields at the outer boundary, is severely compromised and reflections off the outer boundary will be excessively large.

3) The upper frequency is limited to \( c/4\ell \), where \( \ell \) is the largest dimension of the cells used to define the scatterer. This is equivalent to quarter wavelengths equal to the largest cell dimension. At these wavelengths, the granularity in the scattering model becomes apparent and at shorter wavelengths, i.e., high frequencies, invalidates the data.
4) Only expansion factors of 2, 4, 7, and 14 are allowed, as these are the only integral factors possible with a 28-on-a-side cell space. Further, the factor of 14 is not recommended, as it will involve very crude subboundary field interpolations.
SECTION 3

PROGRAMMING INFORMATION

This section provides information directed toward implementation of the computer program G3DXL on a particular computer, modification or extension to meet particular needs or conversion for a different computer environment.

3.1 PROBLEM FLOW

Flow charts showing the overall program structure for G3DXL are given in Figures 2a-c. The functions performed by each subroutine shown on the chart are described in detail in Section 3.5. The main program inputs a variable indicating how frequently to output results; it sets the starting and ending time for the computations and also increments time. Most importantly, in addition, it calls subprograms SETUP, BUILD, EADV, HADV, SAVESB, and DATASAV. The variables input in the main program are described in Table 1.

Table 1. Variables Input in the Main Program, DRIVER, of G3DXL

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>Desired permutation of the four possible configurations</td>
</tr>
<tr>
<td>IDLS</td>
<td>0=Indirect Radiation  1=Direct Lightning Strike</td>
</tr>
<tr>
<td>TSTART</td>
<td>Time (in seconds) at which to start computations</td>
</tr>
</tbody>
</table>
Figure 2a. G3DXL's Program Flow
Figure 2b. First Pass, Diffusion Calculation
Figure 2c. Second Pass, Expanded Diffusion Calculation
PROGRAM DRIVER INPUT, OUTPUT=OUTKF

 COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
 COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
 COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
 DYX(29),DHY(29),DHZ(29),DXO1(28),DYO1(28),DZO1(28)
 COMMON/EXTRAS/NX,NY,NZ,1JX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPLIM,
 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
 COMMON/PERM/MM

 C LOOP OVER THE DESIRED PERMUTATION OF THE TWO CONFIGURATIONS
 (UNEXPANDED M=1/EXPANDED ,M=2)

 C SET MM VALUE TO DETERMINE WHICH PERMUTATION
 (UNEXPANDED ONLY,MM=1/EXPANDED ,MM=12)

 MM=12

 PRINT 10
 10 FORMAT(*MM INPUT ERROR*)

 100 CONTINUE

 UNEXPANDED(M=1) LOOP

 M=1

 GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=1

 CALL SETUP
 CALL BUILD

 SET TIME LIMITS AND EVERY IP DATA POINT SAVED

 TSTART=0.0
 PRINT 111
 111 FORMAT(*BUILD DONE*)

 T=DT/2.+TSTART
 N=0
 CALL EADV
 IF(MM.NE.1) CALL SAVESB
 PRINT 112
 112 FORMAT(*EADV CALLED*)
 DO 130 N=1,48

 ADVANCE TIME

 T=T+DT/2.

 ADVANCE HFIELD

 CALL HADV

 130 

 25
ADVANCE TIME

T = T + DT/2.

ADVANCE EFIELD

CALL EADV
IF(MM .NE. 1) CALL SAVESB

STORE FIELDS

IF(MOD(N,IP).EQ.0) CALL DATASAV

130 CONTINUE

PRINT 150, T, N
150 FORMAT(*EXIT TIME(M=1)=*E12.3*,AFTER CYCLE*I4)

CALL PRINOUT
IF(MM .NE. 1) GO TO 160
GO TO 500

160 CONTINUE

200 CONTINUE

EXPANDED (M=2) LOOP

M = 2

GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=2

CALL SETUP
CALL BUILD

SET TIME LIMITS AND EVERY IP DATA POINT SAVED

TSTART = 0.0
IP = IP * EXPFAC

T = DT/2. + TSTART
N = 0
CALL EADV
DO 230 N = 1, 192
ADVANCE TIME

T = T + DT/2.

ADVANCE HFIELD

CALL HADV

ADVANCE TIME

T = T + DT/2.
ADVANCE EFIELD

CALL EADV

STORE FIELDS

IF(MOD(N,IP).EQ.0) CALL DATASAV

230 CONTINUE

PRINT 250, T, N

250 FORMAT(*EXIT TIME(M=2)=*E12.3*,AFTER CYCLE*I4)

CALL PRINOUT

IF(1M.IE.12) GO TO 260

GO TO 500

260 CONTINUE

300 CONTINUE

500 CONTINUE

END
3.2  GLOBAL VARIABLES

The global variables for G3DXL are located in labeled common. The
nine commons are labeled EFIELD, HFIELD, EXTRAS, GRID, UGRID, RAD, TSITEM, EBS,
and OUT. The variables are described in Table 2.

3.3  DATA FILES

Input to Program G3DXL is made through data statements and by
manually defining program variables, predominantly in subroutine SETUP. The
variables input are identified in the discussion of the subroutines in which
they are defined (Section 3.5).

There is one output file for Program G3DXL. This printer output is
referred to in the program as OUTPUT and is accessed via formatted print
statements. Descriptions of the output data are furnished in the discussions
of the subroutines in which they are output (Section 3.5).

3.4  ERROR MESSAGES

The only error message in G3DXL is in the main program DRIVER. The
statement "MM INPUT ERROR" is printed if MM is not set to an allowed value.
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Other Using Routines</th>
<th>Defined Routines</th>
<th>Size</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Constant $= 3\times 10^8$</td>
<td>EINCX, EINCYP, EINC, EINCXP, HINCYP, HINC, HIT, HITP, DATAX, DATAY</td>
<td>SETUP</td>
<td>1</td>
<td>EXTRAS</td>
</tr>
<tr>
<td>DT</td>
<td>Speed of light in m/sec</td>
<td>Main Program, EADV, HADV</td>
<td>SETUP</td>
<td>28</td>
<td>GRID</td>
</tr>
<tr>
<td>DX, DY, DZ, DI, DZ, etc.</td>
<td>Cycle time step (seconds)</td>
<td>HADV, DATASAV</td>
<td>SETUP</td>
<td>1</td>
<td>EXTRAS</td>
</tr>
<tr>
<td>EPS0</td>
<td>Individual cell dimensions along $x$, $y$, $z$ direction (meters)</td>
<td>HADV, DATASAV</td>
<td>SETUP</td>
<td>1</td>
<td>EADV, DATASAV</td>
</tr>
<tr>
<td>EPS</td>
<td>Individual cell inverse dimensions along $x$, $y$, $z$ direction (meters)</td>
<td>HADV, DATASAV</td>
<td>SETUP</td>
<td>1</td>
<td>EADV, DATASAV</td>
</tr>
<tr>
<td>ESTORE</td>
<td>Dielectric constant of free space $(\text{coul}/\text{n}-\text{m}^2)$</td>
<td>HADV, DATASAV</td>
<td>SETUP</td>
<td>1</td>
<td>EADV, DATASAV</td>
</tr>
</tbody>
</table>

Electric field intensity at each time value for each test point.
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Common</th>
<th>Size</th>
<th>Defining Routines</th>
<th>Other Using Routines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX</td>
<td>EFIELD</td>
<td>(29, 29)</td>
<td>ERAD, EADV, EBC</td>
<td>HADV, DATASAV</td>
<td>x component of electric field for each cell in the cell space</td>
</tr>
<tr>
<td>EYXU</td>
<td>EBS</td>
<td>(28, 29,3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>Backstored values of x components of E-field used for calculating far field radiation condition</td>
</tr>
<tr>
<td>EXZD</td>
<td>EBS</td>
<td>(28, 29,3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>y component of electric field for each cell in the cell space</td>
</tr>
<tr>
<td>EY</td>
<td>EFIELD</td>
<td>(29, 29,29)</td>
<td>ERAD, EADV, EBC</td>
<td>HADV, DATASAV</td>
<td>Backstored values of y components of E-field used for calculating far field radiation condition</td>
</tr>
</tbody>
</table>
Table 2. (continued)

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Common</th>
<th>Size</th>
<th>Defining Routines</th>
<th>Other Using Routines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EYZD</td>
<td>EBS</td>
<td>(28, 29, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>Backstored values of ( y ) components of E-field used for calculating far field radiation condition</td>
</tr>
<tr>
<td>EYZU</td>
<td>EBS</td>
<td>(28, 29, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>( z ) component of E-field for each cell in the cell space</td>
</tr>
<tr>
<td>EZ</td>
<td>EBS</td>
<td>(29, 29, 29)</td>
<td>ERAD, EADV, EBC</td>
<td>Hadv, Datasav</td>
<td>Backstored values of ( z ) components of E-field used for calculating far field radiation condition</td>
</tr>
<tr>
<td>EZXD</td>
<td>EBS</td>
<td>(29, 28, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>( z ) component of E-field for each cell in the cell space</td>
</tr>
<tr>
<td>EZXU</td>
<td>EBS</td>
<td>(29, 28, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>Backstored values of ( z ) components of E-field used for calculating far field radiation condition</td>
</tr>
<tr>
<td>EZYD</td>
<td>EBS</td>
<td>(29, 28, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>( z ) component of E-field for each cell in the cell space</td>
</tr>
<tr>
<td>EZYU</td>
<td>EBS</td>
<td>(29, 28, 3)</td>
<td>EADV</td>
<td>ERAD</td>
<td>Backstored values of ( z ) components of E-field used for calculating far field radiation condition</td>
</tr>
<tr>
<td>HSTOR1</td>
<td>OUT</td>
<td>(500, 30)</td>
<td>Datasav</td>
<td></td>
<td>Component of current density at each time value for each test point</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Common</td>
<td>Size</td>
<td>Defining Routines</td>
<td>Other Using Routines</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>----------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HSTOR2</td>
<td>OUT</td>
<td>(500, 30)</td>
<td>DATASAV</td>
<td></td>
<td>Component of current density at each time value for each test point</td>
</tr>
<tr>
<td>HX</td>
<td>HFIELD</td>
<td>(29, 29, 29)</td>
<td>HADV</td>
<td>EADV, DATASAV</td>
<td>x component of magnetic field for each cell face in the cell space</td>
</tr>
<tr>
<td>HY</td>
<td>HFIELD</td>
<td>(29, 29, 29)</td>
<td>HADV</td>
<td>EADV, DATASAV</td>
<td>y component of magnetic field for each cell face in the cell space</td>
</tr>
<tr>
<td>HZ</td>
<td>HFIELD</td>
<td>(29, 29, 29)</td>
<td>HADV</td>
<td>EADV, DATASAV</td>
<td>z component of magnetic field for each cell face in the cell space</td>
</tr>
<tr>
<td>IP</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>DRIVER</td>
<td>Number of cycles between calls to DATASAV to save data</td>
</tr>
<tr>
<td>MQ</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>EADV, DATASAV</td>
<td>Index used to advance back-stored values of E-fields</td>
</tr>
<tr>
<td>MU</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>EADV, DATASAV</td>
<td>Constant = (4\pi \times 10^{-7}); absolute permeability of free space ((\mu/\lambda^2))</td>
</tr>
<tr>
<td>N</td>
<td>EXTRAS</td>
<td>1</td>
<td>DRIVER</td>
<td>EADV, ERAD, DATASAV</td>
<td>Counter on the number of cycles that have been computed</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Common</td>
<td>Size</td>
<td>Defining Routines</td>
<td>Other Using Routines</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>------------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOPE</td>
<td>TSITEM</td>
<td>(29, 29, 29)</td>
<td>BUILD</td>
<td>EBC</td>
<td>Array that defines the test item. Each cell is assigned a number to indicate if it is an element of the test item</td>
</tr>
<tr>
<td>NX</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>BUILD, HADV, ERAD, DATASAV</td>
<td>NX1+1</td>
</tr>
<tr>
<td>NX1</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>ERAD, EADV, HADV, DATASAV</td>
<td>Number of cells in the cell space on the x dimension</td>
</tr>
<tr>
<td>NY</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>BUILD, HADV, ERAD, DATASAV</td>
<td>NY1+1</td>
</tr>
<tr>
<td>NY1</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>ERAD, EADV, HADV, DATASAV</td>
<td>Number of cells in the cell space in the y dimension</td>
</tr>
<tr>
<td>NZ</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>BUILD, HADV, ERAD, DATASAV</td>
<td>NZ1+1</td>
</tr>
<tr>
<td>NZ1</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>ERAD, EADV, HADV, DATASAV</td>
<td>Number of cells in the cell space in the z dimension</td>
</tr>
<tr>
<td>N1</td>
<td>EBS</td>
<td>1</td>
<td>EADV</td>
<td>ERAD</td>
<td>Cycle numbers used to compute L1, L2 and L3 (see Subroutine ERAD)</td>
</tr>
<tr>
<td>N2</td>
<td>EBS</td>
<td>1</td>
<td>EADV</td>
<td>ERAD</td>
<td></td>
</tr>
<tr>
<td>N3</td>
<td>EBS</td>
<td>1</td>
<td>EADV</td>
<td>ERAD</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Common</td>
<td>Size</td>
<td>Defining Routines</td>
<td>Other Using Routines</td>
<td>Descriptions</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>----------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PI</td>
<td>EXTRAS</td>
<td>1</td>
<td>SETUP</td>
<td>DATASAV</td>
<td>Constant = π</td>
</tr>
<tr>
<td>RXY</td>
<td>RAD</td>
<td>(28, 29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_x$ component that is to be extrapolated in the y direction</td>
</tr>
<tr>
<td>RXZ</td>
<td>RAD</td>
<td>(28, 29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_x$ component that is to be extrapolated in the z direction</td>
</tr>
<tr>
<td>RYX</td>
<td>RAD</td>
<td>(28, 29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_y$ component that is to be extrapolated in the x direction</td>
</tr>
<tr>
<td>RYZ</td>
<td>RAD</td>
<td>(29, 28,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_y$ component that is to be extrapolated in the z direction</td>
</tr>
<tr>
<td>RZX</td>
<td>RAD</td>
<td>(29, 28,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_z$ component that is to be extrapolated in the x direction</td>
</tr>
<tr>
<td>RZY</td>
<td>RAD</td>
<td>(29, 29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Distance to the $E_z$ component that is to be extrapolated in the y direction</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Common</td>
<td>Size</td>
<td>Defining Routines</td>
<td>Other Using Routines</td>
<td>Descriptions</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>--------</td>
<td>-------------------</td>
<td>----------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>T</td>
<td>EXTRAS</td>
<td>1</td>
<td>Main Program</td>
<td>EINCX,EINCY,EINCY,</td>
<td>Time value (seconds)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EINCP,EINCYP,EINCP,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HINCP,HINCPY,HINCPY,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DATASAV</td>
<td></td>
</tr>
<tr>
<td>THXY</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RXY according to Equation (7)</td>
</tr>
<tr>
<td>THXZ</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RXZ according to Equation (7)</td>
</tr>
<tr>
<td>THYX</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RYX according to Equation (7)</td>
</tr>
<tr>
<td>THYZ</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RYZ according to Equation (7)</td>
</tr>
<tr>
<td>THXZ</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RZX according to Equation (7)</td>
</tr>
<tr>
<td>THZY</td>
<td>RAD</td>
<td>(28,  29,2)</td>
<td>SETUP</td>
<td>ERAD</td>
<td>Angle defined in terms of RZY according to Equation (7)</td>
</tr>
<tr>
<td>TFIT</td>
<td>SETUP</td>
<td>300</td>
<td>SETUP</td>
<td>GFIT</td>
<td>Time coordinates for the simulator field driving function (seconds)</td>
</tr>
</tbody>
</table>
Table 2. (continued)

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Common</th>
<th>Size</th>
<th>Defining Routines</th>
<th>Other Using Routines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTORE</td>
<td>OUT</td>
<td>500</td>
<td>DATASAV</td>
<td></td>
<td>Time values for output data (seconds)</td>
</tr>
<tr>
<td>X</td>
<td>GRID</td>
<td>28</td>
<td>SETUP</td>
<td>EINCX,EADV, DATASAV</td>
<td>x coordinates at centers of cells relative to center of test item (meters)</td>
</tr>
<tr>
<td>XO</td>
<td>GRID</td>
<td>29</td>
<td>SETUP</td>
<td>EINCY,EINCY, DATASAV</td>
<td>x coordinates at ends of cells relative to center of test item (meters)</td>
</tr>
<tr>
<td>Y</td>
<td>GRID</td>
<td>28</td>
<td>SETUP</td>
<td>EINCY,EADV, DATASAV</td>
<td>y coordinates at centers of cells relative to center of test item (meters)</td>
</tr>
<tr>
<td>YO</td>
<td>GRID</td>
<td>29</td>
<td>SETUP</td>
<td>EINCY,EINCY, DATASAV</td>
<td>y coordinates at ends of cells relative to center of test item (meters)</td>
</tr>
<tr>
<td>Z</td>
<td>GRID</td>
<td>28</td>
<td>SETUP</td>
<td>EINCY,EADV, DATASAV</td>
<td>z coordinates at centers of cells relative to center of test item (meters)</td>
</tr>
<tr>
<td>ZO</td>
<td>GRID</td>
<td>29</td>
<td>SETUP</td>
<td>EINCY,EINCY, DATASAV</td>
<td>z coordinates at ends of cells relative to center of test item (meters)</td>
</tr>
</tbody>
</table>
3.5 SUBROUTINES

In this section the subroutines that comprise Program C3DXL are presented. A brief description of the function performed by the subroutine is provided along with a subroutine flow chart and a description of those variables local to the subroutine.

SETUP - Problem space definition, initialization and parameter definitions
BUILD - Scattering geometry definition
EADV - Advances free space E-fields
EBC - Sets electric field boundary conditions, lightning channels, and defines internal wires.
ERAD - Outer radiation boundary condition
HADV - Advances free space H-fields
FIELDS - Defines indirect radiated fields
SAVESB - Saves subboundary tangential E-fields for expansion technique
OUTBND - Interpolates SAVESB fields for expanded calculation
DATASAV - Saves data at selected test point locations
PRINOUT - Outputs data from DATASAV, extends wire current data in time

3.5.1 Subroutine SETUP

The test configuration and conditions are specified in subroutine SETUP. The gridding for the cell space, the time step and the radiation factors are determined. The variables listed in Table 3 are set in the subroutine. A flow chart for Subroutine SETUP is furnished in Figure 3. Table 4 presents the variables local to the subroutine.
Table 3. Variables Input in Subroutine SETUP

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELX</td>
<td>Length of cell where there is no cell expansion (x dimension)(m)</td>
</tr>
<tr>
<td>DELY</td>
<td>Width of cell where there is no cell expansion (z dimension)(m)</td>
</tr>
<tr>
<td>DELZ</td>
<td>Height of cell where there is no cell expansion (y dimension)(m)</td>
</tr>
<tr>
<td>IDOWN</td>
<td>I index of the first cell that is not expanded in the x direction</td>
</tr>
<tr>
<td>IUP</td>
<td>I index of the last cell that is not expanded in the x direction</td>
</tr>
<tr>
<td>JDOWN</td>
<td>J index of the first cell that is not expanded in the y direction</td>
</tr>
<tr>
<td>JUP</td>
<td>J index of the last cell that is not expanded in the y direction</td>
</tr>
<tr>
<td>KDOWN</td>
<td>K index of the first cell that is not expanded in the z direction</td>
</tr>
<tr>
<td>KUP</td>
<td>K index of the last cell that is not expanded in the z direction</td>
</tr>
<tr>
<td>IST, JST, KST</td>
<td>Starting point in the unexpanded grid ((I,J,K) = (IST,JST,KST)), for the expanded grid ((I,J,K) = (0,0,0))</td>
</tr>
</tbody>
</table>
Figure 3. Flow Chart for Subroutine SETUP
Table 4. Local Variables - Subroutine SETUP

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTXI</td>
<td>Minimum cell dimension in x</td>
</tr>
<tr>
<td>DTYI</td>
<td>Minimum cell dimension in y</td>
</tr>
<tr>
<td>DTZI</td>
<td>Minimum cell dimension in z</td>
</tr>
<tr>
<td>XPANX, XPANY</td>
<td>Cell expansion coefficient in x, y, z direction</td>
</tr>
<tr>
<td>XPANZ</td>
<td></td>
</tr>
<tr>
<td>EXPFAC</td>
<td>Expansion factor for M=2,4 loops</td>
</tr>
<tr>
<td>IDOWN</td>
<td>See Table 3</td>
</tr>
<tr>
<td>IUP</td>
<td>See Table 3</td>
</tr>
<tr>
<td>JDOWN</td>
<td>See Table 3</td>
</tr>
<tr>
<td>JUP</td>
<td>See Table 3</td>
</tr>
<tr>
<td>KDOWN</td>
<td>See Table 3</td>
</tr>
<tr>
<td>KUP</td>
<td>See Table 3</td>
</tr>
<tr>
<td>Q</td>
<td>Minimum value of QYZ, QZX, QXY, QZY, QXZ, and QYZ</td>
</tr>
<tr>
<td>QXY</td>
<td>Minimum value of θ_{xy}'s</td>
</tr>
<tr>
<td>QXZ</td>
<td>Minimum value of θ_{xz}'s</td>
</tr>
<tr>
<td>QYX</td>
<td>Minimum value of θ_{yx}'s</td>
</tr>
<tr>
<td>QYZ</td>
<td>Minimum value of θ_{yz}'s</td>
</tr>
<tr>
<td>QZX</td>
<td>Minimum value of θ_{zx}'s</td>
</tr>
<tr>
<td>QZY</td>
<td>Minimum value of θ_{zy}'s</td>
</tr>
<tr>
<td>R1</td>
<td>Temporary storage for distances in computation of radiation factors</td>
</tr>
<tr>
<td>R2</td>
<td></td>
</tr>
</tbody>
</table>
The gridding of the cell space for the unexpanded loops is determined from DELX, DELY, and DELZ (see Table 3) and the expansion coefficients (XPANX, XPANY, and XPANZ). The grid size within the nonexpanding area is simply DELX, DELY, and DELZ in the x, y, and z directions respectively. The majority, if not all, of the test item is usually contained within this area. Outside this area the cell size of each cell as one moves away from the center of the cell space is the previous cell size multiplied by the expansion factor for that coordinate, for example,

\[ \text{DXO}(I+1) = \text{DXO}(I) \times \text{EXPNX} \]

The time step for the computations is defined as

\[ \Delta t = \left\{ \sqrt{\frac{1}{\epsilon \mu} \times \left[ \frac{1}{(dx_{\text{min}})^2} + \frac{1}{(dy_{\text{min}})^2} + \frac{1}{(dz_{\text{min}})^2} \right]} \right\}^{-1} \]

where \( d_{\text{min}} \) is the minimum cell dimension for the \( l \) coordinate direction.

The radiation factors \( \theta_{yx}, \theta_{zx}, \theta_{xy}, \theta_{zy}, \theta_{xz}, \theta_{yz}, R_{yx}, R_{zx}, R_{xy}, R_{zy}, R_{xz}, \) and \( R_{yz} \) (see Subroutine ERAD) are computed and stored for later used in the computations.

Subroutine SETUP is called from DRIVER, the main program of G3DXL. Variables input and computed in Subroutine SETUP are printed before it returns to the program.
SUBROUTINE SETUP

DIMENSION DDXO(210),DDYO(210),DDZO(210)
COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DX1(29),DY1(29),DZ1(29),DX01(28),DY01(28),DZ01(28)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),YO(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DX1(29),DY1(29),DZ1(29),DX01(28),DY01(28),DZ01(28)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),YO(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DX1(29),DY1(29),DZ1(29),DX01(28),DY01(28),DZ01(28)
CONMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29),
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,
2 BETA,IDLS
COMMON/RAD/RXY(28,29,2),RZ(28,29,2),RYZ(28,29,2),RZY(29,28,2),
1 RXZ(28,29,2),RYZ(28,29,2),THYX(28,29,2),THXZ(29,28,2),
2 THXY(28,29,2),THYZ(29,28,2),THXY(28,29,2),THYZ(29,28,2)
COMMON/EBS/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZUX(29,28,3)
1 ,EXYD(28,29,3),EXYU(28,29,3),EZXU(29,28,3),
2 EYXD(28,29,3),EYXU(28,29,3),EZXU(29,28,3),EZYD(29,28,3),
3 N1,N2,N3
COMMON/CUMUL/CUMM(24),LOC(24)
SET CYCLES BETWEEN PRINTS AND RECORD LENGTH

DATA LMAX/250/

SET EXPANSION FACTOR

DATA EXPFAC/4.0/

SET PARAMETERS

DATA PI,EPSO,XMU,C/3.1415926536,8.854E-12,1.256637E-6,3.8E/

SET PROBLEM SPACE DIMENSIONS

DATA NX,NY,NZ,NX1,NY1,NZ1/NX,29,29,29,28,28,28/

SET AIRCRAFT SIGMA AND EPSILON

DATA EPS,SIGMA/3.1E-11,1.0E2/

IDLS=0 FOR RADIATED FIELDS,IDLS=1 FOR DIRECT STRIKES

DATA IDLS/1/

IP=2

FIRST INITIALIZE FIELDS TO ZERO

DO 10 I=1,29
DO 10 J=1,29
DO 10 K=1,29
EX(I,J,K)=0.0
EY(I,J,K)=0.0
EZ(I,J,K)=0.0
HX(I,J,K)=0.0
HY(I,J,K)=0.0
HZ(I,J,K)=0.0
10 CONTINUE
C
DO 20 I=1,28
DO 20 J=1,29
DO 20 K=1,3
C
EXYD(I,J,K)=0.0
EXYU(I,J,K)=0.0
EXZD(I,J,K)=0.0
EXZU(I,J,K)=0.0
EYXD(I,J,K)=0.0
EYXU(I,J,K)=0.0
20 CONTINUE
C
DO 25 I=1,29
DO 25 J=1,28
DO 25 K=1,3
C
EYZD(I,J,K)=0.0
EYZU(I,J,K)=0.0
EZLD(I,J,K)=0.0
EZXU(I,J,K)=0.0
EZYD(I,J,K)=0.0
EZYU(I,J,K)=0.0
25 CONTINUE
C
N1=0
N2=0
N3=0
C
DO 30 LL=1,24
CUMM(LL)=0.0
30 CONTINUE
C
AMP=5.92E4
ALPHA=4.08E6
BETA=1.0E8
C
TX=0.002
TY=0.005
TZ=0.005
C
C INPUT GRID FOR M=1
C
SET THE UNEXPANDED CELL DIMENSIONS(DELX,DELY,DELZ) IN METERS,
EXPANSION FACTORS(XPANX,XPANY,XPANZ) AND THE LIMITS OVER
WHICH THE CELLS REMAIN CONSTANT(IUP,IDOWN,JUP,JDOWN,KUP,KDOWN)
DELX=1.2
DELY=0.48
DELZ=0.48
C XPANX=1.3
XPANY=1.15
XPANZ=1.15
C
IUP=25
IDOWN=5
JUP=25
JDOWN=5
KUP=25
KDOWN=5
C
DO 100 I=1,IDOWN
DXO(I)=DELX*(XPANX**(IDOWN-I))
100 CONTINUE
C
DO 105 I=IDOWN,IUP
DXO(I)=DELX
105 CONTINUE
C
DO 110 I=IUP,NX1
DXO(I)=DELX*(XPANX**(I+1-IUP))
110 CONTINUE
C
DO 115 J=1,JDOWN
DYO(J)=DELY*(XPANY**(JDOWN-J))
115 CONTINUE
C
DO 120 J=JDOWN,JUP
DYO(J)=DELY
120 CONTINUE
C
DO 125 J=JUP, NY1
DYO(J)=DELY*(XPANY**(J+1-JUP))
125 CONTINUE
C
DO 130 K=1,KDOWN
DZO(K)=DELZ*(XPANZ**(KDOWN-K))
130 CONTINUE
C
DO 135 K=KDOWN,KUP
DZO(K)=DELZ
135 CONTINUE
C
DO 140 K=KUP,NZ1
DZO(K)=DELZ*(XPANZ**(K+1-KUP))
140 CONTINUE
C
IF(M.EQ.1) GO TO 205
C
INPUT GRID FOR M=2
C
NX1=EXPFAC*NX1
NY1=EXPFAC*NY1

44
NZ1 = EXPFAC * NZ1
NX = NX1 + 1
NY = NY1 + 1
NZ = NZ1 + 1

C
IST = 7
JST = 9
KST = 12

C
IDOWN = (EXPFAC * (IDOWN - 1)) + 1
IUP = (EXPFAC * (IUP - 1)) + 1
XPANX = XPANX / EXPFAC

C
JDOWN = (EXPFAC * (JDOWN - 1)) + 1
JUP = (EXPFAC * (JUP - 1)) + 1
XPANY = XPANY / EXPFAC

C
KDOWN = (EXPFAC * (KDOWN - 1)) + 1
KUP = (EXPFAC * (KUP - 1)) + 1
XPANZ = XPANZ / EXPFAC

C
DELX = DELX / EXPFAC
DELY = DELY / EXPFAC
DELZ = DELZ / EXPFAC

C
DO 145 I = 1, IDOWN
DDX0(I) = DELX * (XPANX**(IDOWN - I))
145 CONTINUE

C
DO 150 I = IDOWN, IUP
DDX0(I) = DELX
150 CONTINUE

C
DO 155 I = IUP, NX1
DDX0(I) = DELX * (XPANX**(I + 1 - IUP))
155 CONTINUE

C
DO 160 J = 1, JDOWN
DDY0(J) = DELY * (XPANY**(JDOWN - J))
160 CONTINUE

C
DO 165 J = JDOWN, JUP
DDY0(J) = DELY
165 CONTINUE

C
DO 170 J = JUP, NY1
DDY0(J) = DELY * (XPANY**(J + 1 - JUP))
170 CONTINUE

C
DO 175 K = 1, KDOWN
DDZ0(K) = DELZ * (XPANZ**(KDOWN - K))
175 CONTINUE

C
DO 180 K = KDOWN, KUP
DDZ0(K) = DELZ

45
CONTINUE
DO 185 K=KUP,NZ1
DDZ0(K)=DELZ*(XPANZ**(K1-KUP))
185 CONTINUE

IEXPST=(EXPFAC*(IST-1))+1
IEXPEND=IEXPST+27
DO 190 I=IEXPST,IEXPEND
DXO(I-IEXPST+1)=DDXO(I)
190 CONTINUE

JEXPST=(EXPFAC*(JST-1))+1
JEXPEND=JEXPST+27
DO 195 J=JEXPST,JEXPEND
DYO(J-JEXPST+1)=DDYO(J)
195 CONTINUE

KEXPST=(EXPFAC*(KST-1))+1
KEXPEND=KEXPST+27
DO 200 K=KEXPST,KEXPEND
DZO(K-KEXPST+1)=DDZO(K)
200 CONTINUE

NX1=NX1/EXPFA
NY1=NY1/EXPFA
NZ1=NZ1/EXPFA
NX=NX1+1
NY=NY1+1
NZ=NZ1+1

IDOWN=((IDOWN-1)/EXPFA)+1
IUP=((IUP-1)/EXPFA)+1
JDOWN=((JDOWN-1)/EXPFA)+1
JUP=((JUP-1)/EXPFA)+1
KDOWN=((KDOWN-1)/EXPFA)+1
KUP=((KUP-1)/EXPFA)+1

DO 210 I=1,NX1
D=D+DXO(I)
210 CONTINUE

DO 215 I=2,NX
X0(I)=-(D/2.)
IF(M.EQ.2) XO(I)=UXO(IST)
X0(I)=X0(I-1)+DXO(I-1)
IF(M.EQ.2) GO TO 215
UXO(I)=X0(I)
UXO(I)=X0(I)
215 CONTINUE

C
W=0.0
DO 220 J=1, NY1
W=W+DY0(J)
220 CONTINUE

C
DO 225 J=2, NY
YO(1)=-(W/2.)
IF(M.EQ.2) YO(1)=UY0(JST)
YO(J)=Y0(J-1)+DY0(J-1)
IF(M.EQ.2) GO TO 225
UY0(1)=Y0(1)
UY0(J)=Y0(J)
225 CONTINUE

C
H=0.0
DO 230 K=1, NZ1
H=H+DZO(K)
230 CONTINUE

C
DO 235 K=2, NZ
ZO(1)=-(H/2.)
IF(M.EQ.2) ZO(1)=UZO(KST)
ZO(K)=ZO(K-1)+DZO(K-1)
IF(M.EQ.2) GO TO 235
UZO(1)=ZO(1)
UZO(K)=ZO(K)
235 CONTINUE

C
DO 240 I=1, NX1
X(I)=(X0(I)+X0(I+1))/2.
IF(M.EQ.2) GO TO 240
UX(I)=X(I)
240 CONTINUE

C
DO 245 J=1, NY1
Y(J)=(YO(J)+YO(J+1))/2.
IF(M.EQ.2) GO TO 245
UY(J)=Y(J)
245 CONTINUE

C
DO 250 K=1, NZ1
Z(K)=(ZO(K)+ZO(K+1))/2.
IF(M.EQ.2) GO TO 250
UZ(K)=Z(K)
250 CONTINUE

C
DO 260 I=2, NX1
DX(I)=X(I)-X(I-1)
260 CONTINUE

C
DX(1)=DX(2)*XPANX
DX(29)=DX(28)*XPANX
C
DO 265 J=2, NY1
DY(J) = Y(J) - Y(J-1)

265 CONTINUE
DY(1) = DY(2) * XPANY
DY(29) = DY(28) * XPANY

DO 270 K = 2, NZ1
DZ(K) = Z(K) - Z(K-1)
270 CONTINUE
DZ(1) = DZ(2) * XPANZ
DZ(29) = DZ(28) * XPANZ

C
DO 275 I = 1, NX1
DXOI(I) = 1. / DXO(I)
275 CONTINUE
C
DO 280 J = 1, NY1
DYOI(J) = 1. / DYO(J)
280 CONTINUE
C
DO 285 K = 1, NZ1
DZOI(K) = 1. / DZO(K)
285 CONTINUE
C
DO 290 I = 1, NX
DXI(I) = 1. / DX(I)
290 CONTINUE
C
DO 295 J = 1, NY
DYI(J) = 1. / DY(J)
295 CONTINUE
C
DO 300 K = 1, NZ
DZI(K) = 1. / DZ(K)
300 CONTINUE
C
DTXI = C / DXO(NX1/2)
DTYI = C / DYO(NY1/2)
DTZI = C / DZO(NZ1/2)
DT = 1. / SQRT(DTXI**2 + DTYI**2 + DTZI**2)
C
IF (M.EQ.2) GO TO 199
C
C EYX RADIATION FACTORS
QYX = 1.
DO 61 K = 1, NZ
DO 61 J = 1, NY1
X1 = X0(1)**2
Y2 = Y(J)**2
Z2 = Z0(K)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THYX(J, K, 1) = 1. - (R1 - R2) / (C*DT)
IF (THYX(J, K, 1).LT.QYX) QYX = THYX(J, K, 1)
RYX(J, K, 1) = R2 / R1

48
X1 = X0(NX)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(NX1)**2
R2 = SQRT(X2 + Y2 + Z2)
THYX(J,K,2) = 1. - (R1-R2)/(C*DT)
IF(THYX(J,K,2).LT.QYX) QYX = THYX(J,K,2)
RYX(J,K,2) = R2/R1
CONTINUE

EZX RADIATION FACTORS

QZX = 1.
DO 62 K=1,NZ
DO 62 J=1,NY
X1 = X0(1)**2
Y2 = Y0(J)**2
Z2 = Z(K)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THZX(J,K,1) = 1. - (R1-R2)/(C*DT)
IF(THZX(J,K,1).LT.QZX) QZX = THZX(J,K,1)
RZX(J,K,1) = R2/R1
X1 = X0(NX)**2
R1 = SQRT(X1 + Y2 + Z2)
X2 = X0(NX1)**2
R2 = SQRT(X2 + Y2 + Z2)
THZX(J,K,2) = 1. - (R1-R2)/(C*DT)
IF(THZX(J,K,2).LT.QZX) QZX = THZX(J,K,2)
RZX(J,K,2) = R2/R1
CONTINUE

EXY RADIATION FACTORS

QXY = 1.
DO 63 K=1,NZ
DO 63 I=1,NX
X2 = X(I)**2
Y1 = Y0(1)**2
Z2 = ZO(K)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THXY(I,K,1) = 1. - (R1-R2)/(C*DT)
IF(THXY(I,K,1).LT.QXY) QXY = THXY(I,K,1)
RXY(I,K,1) = R2/R1
Y1 = Y0(NY)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(NY1)**2
R2 = SQRT(X2 + Y2 + Z2)
THXY(I,K,2) = 1. - (R1-R2)/(C*DT)
IF(THXY(I,K,2).LT.QXY) QXY = THXY(I,K,2)
RXY(I,K,2) = R2/R1
CONTINUE

CONTINUE
EZY RADIATION FACTORS

QZY = 1.
DO 64 K=1,NZ1
DO 64 I=1,NX
X2 = XO(I)**2
Y1 = YO(1)**2
Z2 = Z(K)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THZY(I,K,1) = 1. - (R1-R2)/(C*DT)
IF(THZY(I,K,1) .LT. QZY) QZY = THZY(I,K,1)
RZY(I,K,1) = R2/R1
Y1 = YO(NY)**2
R1 = SQRT(X2 + Y1 + Z2)
Y2 = YO(NY1)**2
R2 = SQRT(X2 + Y2 + Z2)
THZY(I,K,2) = 1. - (R1-R2)/(C*DT)
IF(THZY(I,K,2) .LT. QZY) QZY = THZY(I,K,2)
RZY(I,K,2) = R2/R1
CONTINUE

EXZ RADIATION FACTORS

QXZ = 1.
DO 65 I=1,NX1
DO 65 J=1,NY
X2 = X(I)**2
Y2 = YO(J)**2
Z1 = ZO(1)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THXZ(I,J,1) = 1. - (R1-R2)/(C*DT)
IF(THXZ(I,J,1) .LT. QXZ) QXZ = THXZ(I,J,1)
RXZ(I,J,1) = R2/R1
Z1 = ZO(NZ)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(NZ1)**2
R2 = SQRT(X2 + Y2 + Z2)
THXZ(I,J,2) = 1. - (R1-R2)/(C*DT)
IF(THXZ(I,J,2) .LT. QXZ) QXZ = THXZ(I,J,2)
RXZ(I,J,2) = R2/R1
CONTINUE

EYZ RADIATION FACTORS

QYZ = 1.
DO 66 J=1,NY1
DO 66 I=1,NX
X2 = XO(I)**2
Y2 = Y(J)**2
Z1 = ZO(1)**2
R1 = SQRT(X2 + Y2 + Z1)
Z2 = ZO(2)**2
R2 = SQRT(X2 + Y2 + Z2)
THYZ(I,J,1)=1.-(R1-R2)/(C*DT)
IF(THYZ(I,J,1).LT.QYZ) QYZ=THYZ(I,J,1)
RYZ(I,J,1)=R2/R1
Z1=ZO(NZ)**2
R1=SQRT(X2+Y2+Z1)
Z2=ZO(NZ1)**2
R2=SQRT(X2+Y2+Z2)
THYZ(I,J,2) = 1. - (R1-R2)/(C*DT)
IF(THYZ(I,J,2).LT.QYZ) QYZ = THYZ(I,J,2)
RYZ(I,J,2) = R2/R1
66 CONTINUE
Q = AMIN1(QYX,QZX,QXY,QZY,QXZ,QYZ)
MQ = -Q/2.

PRINT VARIABLES

199 PRINT 201
PRINT 202, DT, EPS, SIGMA
PRINT 99
PRINT 104, NX, NX1, NY, NY1, NZ, NZ1
PRINT 101
PRINT 98, (XO(I), I = 1,NX)
PRINT 102
PRINT 98, (YO(J), J = 1, NY)
PRINT 107
PRINT 98, (DXO(I), I = 1, NX1)
PRINT 103
PRINT 98, (ZO(K), K = 1, NZ)
PRINT 106
PRINT 98, QYX, QZX, QXY, QZY, QXZ, QYZ, Q

FORMAT STATEMENTS

98 FORMAT (10F10.4)
101 FORMAT (*OXO GRID*)
102 FORMAT (*OYO GRID*)
103 FORMAT (*UZO GRID*)
104 FORMAT (2015)
99 FORMAT(5HU IJX,ZX,*NX1 NY NY1 NZ NZ1 MQ*)
106 FORMAT(11H0,5X,*QYX*,6X,*QZX*,6X,*QXY*,6X,*QZY*,6X,*QXZ*,6X,*QYZ*,
1 7X,*Q*)
107 FORMAT (*UDXO GRID*)
202 FORMAT (3E12.4)
201 FORMAT (*1DT, EPS, SIGMA*)
390 FORMAT(2F10.3,2E12.4)
400 FORMAT(3F10.0)
440 FORMAT(615)
RETURN
END
3.5.2 Subroutine BUILD

The NOPE array, which defines the test time, is filled in Subroutine BUILD. Each cell in the cell space is identified by three indices \((I,J,K)\) (see Figure 1). The entire array is initialized to zero. For perfect conductors, if the \(I,J,K\)th cell is a three-dimensional cell in the test time, NOPE\((I,J,K)\) is set equal to four \((4)\). If the YZ plane adjacent to vertex \(I,J,K\) of the cell is a two-dimensional piece of the test time, NOPE\((I,J,K)\) is set equal to one \((1)\). For the XZ plane, NOPE\((I,J,K)\) is set to two \((2)\) and to three \((3)\) if the XY plane is a two-dimensional part of the test item. Where planes meet NOPE is set to 12 (YZ and XZ planes), 13 (YZ and XY planes), 23 (XZ and XY planes) and 123 (YZ, XZ and XY planes). When an imperfect conductor is modeled \(4+8\) and \(1,2,3+5,6,7\). The values of the NOPE array for each XZ plane cut of the test item are printed and can be used to check the test item structure. This routine is replaced in its entirety whenever a computation on a new structure is required.

The version of Subroutine BUILD included here builds the F-106B aircraft. A flow chart for the routine is given in Figure 4. Subroutine BUILD is called by the main program DRIVER.
Figure 4. Flow Chart for Subroutine BUILD

Start

Initialize Array to Zero

Build M=1 Geometry

Build M=2 Geometry

Print NOPE Array

Return
SUBROUTINE BUILD

COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,ILMAX,SIGMA,C,T,PI,EXPFA,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/TSITEM/NOPE(29,29,29)

DO 100 I=1,NX
DO 100 J=1,NY
DO 100 K=1,NZ
NOPE(I,J,K)=0

100 CONTINUE

CYLINDER TEST CASE FOR CODE OPERATION CHECK

TEST=0
IF(TEST.EQ.0) GO TO 8888

BUILD M=1 GEOMETRY
IF(M.NE.1) GO TO 200

DO 110 I=5,24
DO 110 J=13,16
DO 110 K=13,16
NOPE(I,J,K)=8

110 CONTINUE
GO TO 700

DO 1101 I=5,25,20
DO 1101 J=13,16
DO 1101 K=13,16
NOPE(I,J,K)=5

1101 CONTINUE
GO TO 700

DO 1102 I=5,24
DO 1102 J=13,17,4
DO 1102 K=13,16
IF(NOPE(I,J,K).EQ.1) GO TO 1111
NOPE(I,J,K)=6
GO TO 1102

1111 NOPE(I,J,K)=56

1102 CONTINUE
DO 1103 I=5,24
DO 1103 J=13,16
DO 1103 K=13,17,4
IF(NOPE(I,J,K).EQ.1) GO TO 1112
IF(NOPE(I,J,K).EQ.2) GO TO 1113
IF(NOPE(I,J,K).EQ.12) GO TO 1114
NOPE(I,J,K)=7
GO TO 1103

1112 NOPE(I,J,K)=57

1113 NOPE(I,J,K)=67
GO TO 1103

1114 NOPE(I,J,K)=567
1103 CONTINUE
GO TO 700

C

54
C BUILD M=2 GEOMETRY
200 CONTINUE
   IF(M.NE.2) GO TO 300
   DO 120 I=1,28
   DO 120 J=5,20
   DO 120 K=5,20
      NOPE( I,J,K)=8
   120 CONTINUE
C
   GO TO 700
C
8888 CONTINUE
C
C BUILD M=1 F 106 B GEOMETRY
C
C IF(M.NE.1) GO TO 1200
C
C FUSELAGE
C
   DO 10 I=8,20
   DO 10 J=11,12
   DO 10 K=13,16
      NOPE( I,J,K)=4
   10 CONTINUE
C
   DO 11 I=11,22
   DO 11 J=13,14
   DO 11 K=13,16
      NOPE( I,J,K)=4
   11 CONTINUE
C
      NOPE(9,13,14)=4
      NOPE(9,13,15)=4
C
      NOPE(10,13,13)=4
      NOPE(10,13,14)=4
      NOPE(10,13,15)=4
      NOPE(10,13,16)=4
C
      NOPE(10,14,14)=4
      NOPE(10,14,15)=4
C
   DO 12 I=21,22
       J=12
   DO 12 K=13,16
      NOPE( I,J,K)=4
   12 CONTINUE
C
   DO 13 I=13,15
   DO 13 J=12,13
   DO 13 K=12,17,5
      NOPE( I,J,K)=4
   13 CONTINUE
C
C WINGS
C
DO 14 I=14,20
   J=12
   DO 14 K=9,20
   NOPE(I,J,K)=4
  14 CONTINUE

NOPE(14,12,10)=0
NOPE(14,12,19)=0
NOPE(14,12,9)=0
NOPE(14,12,20)=0
NOPE(15,12,9)=0
NOPE(15,12,20)=0

DO 15 I=18,20
   J=12
   DO 15 K=7,22
   NOPE(I,J,K)=4
  15 CONTINUE

NOPE(18,12,7)=0
NOPE(18,12,22)=0

STABILIZER

DO 16 I=19,20
   DO 16 J=15,17
       K=15
   NOPE(I,J,K)=3
  16 CONTINUE

NOPE(19,17,15)=0

DO 17 I=21,22
   DO 17 J=15,21
       K=15
   NOPE(I,J,K)=3
  17 CONTINUE

NOPE(21,20,15)=0
NOPE(21,21,15)=0

GO TO 700

1200 CONTINUE

BUILD M=2 GEOMETRY

IF(M.NE.2) GO TO 1300

BUILD SOLID PORTION FIRST

TREAT SOLID PORTION ON SUBBOUNDARY

I=28
DO 20 J=9,24
DO 20 K=5,20
NOPE(I,J,K)=4
20 CONTINUE

C
  DO 21 J=13,20
  DO 21 K=1,24
  NOPE(I,J,K)=4
21 CONTINUE

C
  TREAT SOLID UPPER BODY
C
  DO 22 I=17,27
  DO 22 J=15,24
  DO 22 K=6,19
  NOPE(I,J,K)=4
22 CONTINUE

C
  DO 23 I=13,16
  DO 23 J=13,20
  DO 23 K=6,19
  NOPE(I,J,K)=4
23 CONTINUE

C
  DO 24 I=13,16
  DO 24 J=21,24
  DO 24 K=7,18
  NOPE(I,J,K)=4
24 CONTINUE

C
  DO 25 I=10,12
  DO 25 J=13,18
  DO 25 K=6,19
  NOPE(I,J,K)=4
25 CONTINUE

C
  I=12
  DO 26 J=19,23
  DO 26 K=8,17
  NOPE(I,J,K)=4
26 CONTINUE

C
  I=11
  DO 27 J=19,22
  DO 27 K=9,16
  NOPE(I,J,K)=4
27 CONTINUE

C
  I=10
  DO 28 J=19,20
  DO 28 K=10,15
  NOPE(I,J,K)=4
28 CONTINUE

C
  I=9
DO 29 J=18,19
DO 29 K=11,14
NOPE(I,J,K)=4
29 CONTINUE

C
DO 30 I=6,9
    J=17
DO 30 K=7,18
NOPE(I,J,K)=4
30 CONTINUE

C
DO 31 I=4,5
    J=16
DO 31 K=8,17
NOPE(I,J,K)=4
31 CONTINUE

C
TREAT Y-Z THIN PLANES(NOPE(I,J,K)=1)

C
BULKHEAD B1

C
I=4
DO 32 J=9,15
DO 32 K=8,17
NOPE(I,J,K)=1
32 CONTINUE

C
HOLES IN B1

C
DO 33 J=9,14
DO 33 K=12,13
NOPE(I,J,K)=0
33 CONTINUE

C
BULKHEAD B2

C
I=6
DO 34 J=9,16
DO 34 K=7,18
NOPE(I,J,K)=1
34 CONTINUE

C
HOLES IN B2

C
DO 35 J=9,10
DO 35 K=14,17
NOPE(I,J,K)=0
35 CONTINUE

C
BULKHEAD B4

C
I=30
DO 36 J=9,12
DO 36 K=6,19
NOPE(I,J,K)=1
36 CONTINUE
36 CONTINUE
C
C HOLES IN B4
C
DO 37 J=9,10
DO 37 K=17,18
NOPE(I,J,K)=0
37 CONTINUE
C
C BULKHEAD B6
C
I=16
DO 38 J=9,14
DO 38 K=6,19
NOPE(I,J,K)=1
38 CONTINUE
C
C HOLES IN B6
C
DO 39 J=9,10
DO 39 K=18,19
NOPE(I,J,K)=0
39 CONTINUE
C
C BULKHEAD B8
C
I=20
DO 40 J=9,14
DO 40 K=6,19
NOPE(I,J,K)=1
40 CONTINUE
C
C HOLES IN B8
C
DO 41 J=13,14
DO 41 K=18,19
NOPE(I,J,K)=0
41 CONTINUE
C
C TREAT X-Z THIN PLANES(NOPE(I,J,K)=2)
C
DO 42 I=4,27
J=9
DO 42 K=6,19
IF(NOPE(I,J,K).EQ.1) GO TO 420
NOPE(I,J,K)=2
GO TO 42
420 NOPE(I,J,K)=12
42 CONTINUE
C
DO 43 I=4,9
DO 43 K=6,19,13
NOPE(I,J,K)=0
43 CONTINUE
C
DO 44 I=4,5
DO 44 K=7,18,11
NOPE(I,J,K)=0
44 CONTINUE

C TREAT X-Y THIN PLANES(NOPE(I,J,K)=3)
C
DO 45 I=10,15
DO 45 J=9,12
DO 45 K=6,20,14
IF(NOPE(I,J,K).EQ.1) GO TO 450
IF(NOPE(I,J,K).EQ.2) GO TO 451
IF(NOPE(I,J,K).EQ.12) GO TO 452
NOPE(I,J,K)=3
GO TO 45

450 NOPE(I,J,K)=13
GO TO 45
451 NOPE(I,J,K)=23
GO TO 45
452 NOPE(I,J,K)=123
45 CONTINUE

C
DO 46 I=16,27
DO 46 J=9,14
DO 46 K=6,20,14
IF(NOPE(I,J,K).EQ.1) GO TO 460
IF(NOPE(I,J,K).EQ.2) GO TO 461
IF(NOPE(I,J,K).EQ.12) GO TO 462
NOPE(I,J,K)=3
GO TO 46

460 NOPE(I,J,K)=13
GO TO 46
461 NOPE(I,J,K)=23
GO TO 46
462 NOPE(I,J,K)=123
46 CONTINUE

C
DO 47 I=6,9
DO 47 J=9,16
DO 47 K=7,19,12
IF(NOPE(I,J,K).EQ.1) GO TO 470
IF(NOPE(I,J,K).EQ.2) GO TO 471
IF(NOPE(I,J,K).EQ.12) GO TO 472
NOPE(I,J,K)=3
GO TO 47

470 NOPE(I,J,K)=13
GO TO 47
471 NOPE(I,J,K)=23
GO TO 47
472 NOPE(I,J,K)=123
47 CONTINUE

C
DO 48 I=4,5
DO 48 J=9,15
DO 48 K=8,18,10

60
IF(NOPE(I,J,K).EQ.1) GO TO 480
IF(NOPE(I,J,K).EQ.2) GO TO 481
IF(NOPE(I,J,K).EQ.12) GO TO 482
NOPE(I,J,K)=3
GO TO 48

480 NOPE(I,J,K)=13
GO TO 48

481 NOPE(I,J,K)=23
GO TO 48

482 NOPE(I,J,K)=123
48 CONTINUE

C
GO TO 700
C

401 FORMAT(1hl, ,5x,7hNOPE(I,12,3H,K),//)
402 FORMAT(2014)
700 CONTINUE
RETURN
END
3.5.3 Subroutine EADV

In subroutine EADV, the equation

\[ \nabla \times \mathbf{H} = \varepsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} \]

is applied to the scattered fields and is used to find \( \mathbf{E} \) from \( \mathbf{E} \) and \( \mathbf{H} \) at earlier times. This "time stepping" solution is found for each component of \( \mathbf{E} \), i.e., Equation (1) is written explicitly for each component as

\[
\begin{align*}
\varepsilon \frac{\partial E_x}{\partial t} + \sigma E_x &= \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\
\varepsilon \frac{\partial E_y}{\partial t} + \sigma E_y &= \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\
\varepsilon \frac{\partial E_z}{\partial t} + \sigma E_z &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}
\end{align*}
\]

These equations are cast in their finite-difference form and solved for \( \mathbf{E} \) at the latest time. Thus

\[
\begin{align*}
E^{n+1/2}_{x}(I,J,K) &= E^{n-1/2}_{x}(I,J,K) \\
+ (\Delta t/\varepsilon) \frac{H^{n}_{z}(I,J,K) - H^{n}_{z}(I,J-1,K)}{Y(J) - Y(J-1)} \\
- \frac{H^{n}_{y}(I,J,K) - H^{n}_{y}(I,J,K-1)}{Z(K) - Z(K-1)} \\
E^{n+1/2}_{y}(I,J,K) &= E^{n-1/2}_{y}(I,J,K) + (\Delta t/\varepsilon) \\
\left( \frac{H^{n}_{x}(I,J,K) - H^{n}_{x}(I,J,K-1)}{Z(K) - Z(K-1)} - \frac{H^{n}_{z}(I,J,K) - H^{n}_{z}(I-1,J,K)}{X(I) - X(I-1)} \right)
\end{align*}
\]
\[ E_z^{n+1/2}(I,J,K) = E_z^{n-1/2}(I,J,K) + (\Delta t/\varepsilon) \]  

\[
\left( \begin{array}{c}
\frac{H_y^n(I,J,K) - H_y^{n}(I-1,J,K)}{X(I) - X(I-1)} \\
\frac{H_x^n(I,J,K) - H_x^{n}(I,J-1,K)}{Y(J) - Y(J-1)}
\end{array} \right)
\]

EADV is called every program cycle by the main program, as is the subroutine HADV for advancing the H-fields. It uses the prior values of the fields to calculate the new values. Initially, these prior values are zero. The first nonzero values are those values for E and H that imposed at the scatterer's boundary. These values are accessed by a call to EBC after the equations (2), (3), and (4) have been advanced in EADV. Thus, on the next program cycle these values will be available and nonzero values of the fields off the boundary will start appearing when Equations (2), (3), and (4) are advanced again. Since EBC is always called after the fields are advanced, the proper boundary conditions are maintained.

Additionally, EADV stores the E-field components one cell in from each face of the outer boundary that are used in the calculation of the E-field component values on the outer boundary. This calculation, based on the radiation boundary condition [2], is performed in Subroutine ERAD.
Figure 5. Flow Chart for Subroutine EADV
SUBROUTINE EADV

COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/HFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NXI,NYL,NZL,N,M,D,T,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,P,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DZI(29),DXI(28),DYO(28),DZO(28),
COMMON/EBS/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZXU(29,28,3)
1 EXYD(28,29,3),EXYU(28,29,3),EZYD(29,28,3),EZYE(29,28,3),
2 EXZD(28,29,3),EZXD(29,28,3),EZXY(28,28,3),EZUY(29,28,3),
3 N1,N2,N3
COMMON/TSMEM/NOPE(29,29,29)

DTE=DT/EPSO

ADVANCE EX

DO 1 I = 1,NX1
DO 1 J = 2,NY1
DO 1 K = 2,NZ1
1 GO TO 1
EX(I,J,K)=EX(I,J,K)+DTE*((HZ(I,J,K)-HZ(I,J-1,K))*DZI(K)
1 -(HY(I,J,K)-HY(I,J,K-1))*DXI(I))
1 CONTINUE

ADVANCE EY

DO 2 I = 2,NX1
DO 2 J = 1,NY1
DO 2 K = 2,NZ1
1 GO TO 2
EY(I,J,K)=EY(I,J,K)+DTE*((HX(I,J,K)-HX(I,J,K-1))*DZI(K)
1 -(HY(I,J,K)-HY(I,J,K-1))*DXI(I))
2 CONTINUE

ADVANCE EZ

DO 3 I = 2,NX1
DO 3 J = 2,NY1
DO 3 K = 1,NZ1
1 GO TO 3
EZ(I,J,K)=EZ(I,J,K)+DTE*((HY(I,J,K)-HY(I-1,J,K))*DXI(I)
1 -(HX(I,J,K)-HX(I-1,J,K))*DYI(J))
3 CONTINUE

IF (M.EQ.2) GO TO 10
IF (N2.NE.N3) CALL ERAD
10 CONTINUE
CALL EBC
IF (M.EQ.2) CALL OUTBND
IF (N.EQ.2) RETURN
IF (MOD(N,MQ).NE.0) RETURN
N3=N2
N2=N1
N1=N

C C C ADVANCE EXY
C DO 11 I = 1,NX1
DO 11 K = 2,NZ1
EXYD(I,K,3) = EXYD(I,K,2)
EXYD(I,K,2) = EXYD(I,K,1)
EXYD(I,K,1) = EX(I,2,K)
EXYU(I,K,3) = EXYU(I,K,2)
EXYU(I,K,2) = EXYU(I,K,1)
EXYU(I,K,1) = EX(I,NY1,K)
11 CONTINUE

C C C ADVANCE EXZ
C DO 111 I = 1,NX1
DO 111 J = 2,NY1
EXZD(I,J,3) = EXZD(I,J,2)
EXZD(I,J,2) = EXZD(I,J,1)
EXZD(I,J,1) = EX(I,J,2)
EXZU(I,J,3) = EXZU(I,J,2)
EXZU(I,J,2) = EXZU(I,J,1)
EXZU(I,J,1) = EX(I,J,NZ1)
111 CONTINUE

C C C ADVANCE EYX
C DO 22 J = 1,NY1
DO 22 K = 2,NZ1
EYXD(J,K,3) = EYXD(J,K,2)
EYXD(J,K,2) = EYXD(J,K,1)
EYXD(J,K,1) = EY(2,J,K)
EYXU(J,K,3) = EYXU(J,K,2)
EYXU(J,K,2) = EYXU(J,K,1)
EYXU(J,K,1) = EY(NX1,K)
22 CONTINUE

C C C ADVANCE EYZ
C DO 222 I = 2,NX1
DO 222 J = 1,NY1
EYZD(I,J,3) = EYZD(I,J,2)
EYZD(I,J,2) = EYZD(I,J,1)
EYZD(I,J,1) = EY(I,J,2)
EYZU(I,J,3) = EYZU(I,J,2)
EYZU(I,J,2) = EYZU(I,J,1)
EYZU(I,J,1) = EY(I,NZ1)
222 CONTINUE
C ADVANCE EZX
C
DO 33 J = 2,NY1
DO 33 K = 1,NZ1
EZXD(J,K,3) = EZXD(J,K,2)
EZXD(J,K,2) = EZXD(J,K,1)
EZXD(J,K,1) = EZ(2,J,K)
EZXU(J,K,3) = EZXU(J,K,2)
EZXU(J,K,2) = EZXU(J,K,1)
EZXU(J,K,1) = EZ(NX1,J,K)
33 CONTINUE
C ADVANCE EZY
C
DO 333 I = 2,NX1
DO 333 K = 1,NZ1
EZYD(I,K,3) = EZYD(I,K,2)
EZYD(I,K,2) = EZYD(I,K,1)
EZYD(I,K,1) = EZ(I,2,K)
EZYU(I,K,3) = EzyU(I,K,2)
EZYU(I,K,2) = EzyU(I,K,1)
EZYU(I,K,1) = EZ(I,NY1,K)
333 CONTINUE
C
RETURN
END
3.5.4 Subroutine EBC

Subroutine EBC sets the appropriate electric field boundary conditions for the test item. On the surface of a perfectly conducting aircraft, the total tangential E-field vanishes so

\[ E_{\text{scattered}} = -E_{\text{incident}} \]

The conditions are set in one part of the subroutine for any cell that defines a three-dimensional portion of the test item \(-E_x, E_y\) and then \(E_z\). The setting of boundary conditions for the two dimensional surfaces is done in a separate part of the subroutine.

The values of the NOPE array for the cell and its surrounding cells are used to determine what the E-field value should be. If NOPE(I,J,K) is zero, no action is taken. For a cell with NOPE(I,J,K) = 4, the three surrounding cells are checked. If any one of the three is missing, the field is on the surface and \(E\) is set to \(-E_{\text{incident}}\). If all three cells are present \(E\) is not set, since it is then internal to the test item. Figure 6 indicates the situation for \(E_x\) on cell I,J,K. It is clear that if either cell 2 or 3 is not a part of the test item, \(E_x\) for cell 1 would be on an external surface of the item. Similar situations can be defined for \(E_y\) and \(E_z\).

In addition, if, for \(E_x\), cell I,J,K is the outermost cell in either the J or the K directions, the \(E_x\) values at the far edges of this last cell are set equal to \(-E_{\text{incident}}\). The same is done for \(E_y\), if the cell is the outermost in either the I or K directions and for \(E_z\), in the I or J directions.

If NOPE defines a two-dimensional surface, the appropriate E-field components are set equal to \(-E_{\text{incident}}\). If the surface is in the X-Y plane, for example, \(E_x\) on the two sides of the rectangle defined by the cell and \(E_y\) on the other two sides are set equal to \(-E_{\text{incident}}\).
Figure 6. Determine E-Field Boundary Conditions

For a lossy dielectric aircraft or an aircraft composed, in part, by a lossy dielectric, the boundary condition is given by

\[ \varepsilon \frac{\partial \tilde{E}^s}{\partial t} + \sigma \tilde{E}^s = \tilde{\nabla} \times \tilde{H}^s - \sigma \tilde{E}^l - (\varepsilon - \varepsilon_o) \frac{\partial \tilde{E}^l}{\partial t} \]

This boundary condition holds within the lossy dielectric volume. It may be considered a volume boundary condition. In the limit, \( \sigma \to \infty \), it becomes the surface boundary condition

\[ \varepsilon \frac{\tan \tilde{E}^{\text{scattered}}} {\tan \tilde{E}^{\text{incident}}} = -\varepsilon \tan \]  

for perfectly conducting surfaces.
Solid blocks (3-dimensional bodies) and plates (2-dimensional bodies) are treated much the same as for the perfectly conducting counterparts, except:

1) the volume boundary condition is used
2) the NOPE array values of 1,2,3,12,13,23,123, and 4 are replaced by 5,6,7,56,57,67,567 and 8 respectively
3) the plates have thickness TX, TY and TZ in the x, y and z directions respectively, where the thickness is expressed as a fraction of the cell dimension in the appropriate direction
4) the volume boundary condition is slightly modified, as discussed discussed in Appendix A: Thin Plate Formalism, to more accurately model plates.

The generalized flow of the Subroutine EBC is given in Figure 7. The details of the flow through the perfectly conducting portion of EBC are given in Figure 8. The lossy dielectric portion follows the same approach. Subroutine EBC is called from EADV. EINCX, EINCY, and EINCNZ are called by EBC.

Thin wires representing the lightning channels and interim wires (M=1 and 12) are also input in EBC.
Figure 7. Generalized Flow Chart of Subroutine EBC
Figure 8. Flow Chart of Subroutine EBC
SUBROUTINE EBC

COMMON/EFIELD/EX(29,29,29), EY(29,29,29), EZ(29,29,29)
COMMON/EFIELD/HX(29,29,29), HY(29,29,29), HZ(29,29,29)
COMMON/GRID/X(28), Y(28), Z(28), XO(29), YO(29), ZO(29),
1 DX(29), DY(29), DZ(29), DXO(28), DY0(28), DZO(28),
2 DXI(29), DYOI(29), DZI(29), DXOI(28), DYOI(28), DZOI(28)
COMMON/TITEM/NOPE(29,29,29)
COMMON/EXTRAS/NX, NY, NZ, NX1, NY1, NZ1, N, M, MQ, DT, XMU, EPSO, EPS, NPTLIM,
1 NR, NPTS, LMAX, SIGMA, C, T, EXPFAC, IP, TX, TY, TZ, AMP, ALPHA, BETA, IDLS

IF(IDLS.EQ.0) GO TO 1
IF(M.NE.1) GO TO 2

UNEXPANDED LIGHTNING CHANNELS

DO 3 J=13,28
EY(8,J,15)=0.0
3 CONTINUE

DO 4 J=1,11
EY(22,J,15)=0.0
4 CONTINUE

GO TO 1

2 CONTINUE
IF(M.NE.2) GO TO 5

EXPANDED LIGHTNING CHANNELS

DO 6 J=9,28
EY(5,J,13)=0.0
6 CONTINUE

GO TO 1

5 CONTINUE

1 CONTINUE

LOOP OVER REGION IN WHICH EBC MUST BE SET

DO 500 I=1,NX1
DO 500 J=1,NY1
DO 500 K=1,NZ1

IF(M.EQ.1) GO TO 12
TX=0.008
TY=0.02
TZ=0.02

500 CONTINUE
CONTINUE
C
IF(NOPE(I,J,K).EQ.0)GO TO 500
C
IF (NOPE(I,J,K).NE.4) GO TO 60
C
SET BOUNDARY CONDITIONS ON PERFECTLY CONDUCTING 3-D BODY
C
SET EX
C
IF((J-1).EQ.0.OR.(K-1).EQ.0) GO TO 18
  1 NOPE(I,J-1,K).EQ.4)GO TO 10
EX(I,J,K)=-EINCX(I,J,K)
10 CONTINUE
IF(NOPE(I,J+1,K).EQ.4)GO TO 15
EX(I,J+1,K+1)=-EINCX(I,J+1,K+1)
EX(I,J+1,K)=-EINCX(I,J+1,K)
15 IF(NOPE(I,J,K+1).EQ.4)GO TO 20
EX(I,J,K+1)=-EINCX(I,J,K+1)
EX(I,J+1,K+1)=-EINCX(I,J+1,K+1)

C
C
SET EY
C
18 CONTINUE
IF((I-1).EQ.0.OR.(K-1).EQ.0) GO TO 38
  1 NOPE(I-1,J-1,K).EQ.4)GO TO 30
EY(I,J,K)=-EINCY(I,J,K)
30 CONTINUE
IF(NOPE(I+1,J,K).EQ.4)GO TO 35
EY(I+1,J,K+1)=-EINCY(I+1,J,K+1)
EY(I+1,J,K)=-EINCY(I+1,J,K)
35 IF(NOPE(I,J,K+1).EQ.4)GO TO 40
EY(I,J,K+1)=-EINCY(I,J,K+1)
EY(I+1,J,K+1)=-EINCY(I+1,J,K+1)

C
C
SET EZ
C
38 CONTINUE
IF((I-1).EQ.0.OR.(J-1).EQ.0) GO TO 500
  1 NOPE(I-1,J-1,K).EQ.4)GO TO 50
EZ(I,J,K)=-EINCZ(I,J,K)
50 CONTINUE
IF(NOPE(I+1,J,K).EQ.4)GO TO 55
EZ(I+1,J,K)=-EINCZ(I+1,J,K)
EZ(I+1,J+1,K)=-EINCZ(I+1,J+1,K)
55 IF(NOPE(I,J+1,K).EQ.4)GO TO 500
EZ(I+1,J+1,K)=-EINCZ(I+1,J+1,K)
EZ(I,J+1,K)=-EINCZ(I,J+1,K)
GO TO 500
C
60 CONTINUE
C
1
2

C SET BOUNDARY CONDITIONS ON PERFECTLY CONDUCTING 2-D BODY

C

1
AND.NOPE(I,J,K).NE.123) GO TO 70

C

Y-Z PLANE

C

EY(I,J,K+1)=-EINCY(I,J,K+1)
EY(I,J,K)=-EINCY(I,J,K)
EZ(I,J+1,K)=-EINCY(I,J+1,K)
EZ(I,J,K)=-EINCY(I,J,K)

C

IF(NOPE(I,J,K).EQ.12.OR.NOPE(I,J,K).EQ.123) GO TO 70
IF(NOPE(I,J,K).EQ.13) GO TO 80
GO TO 500

C

1

C

X-Z PLANE

C

EX(I,J,K+1)=-EINCX(I,J,K+1)
EX(I,J,K)=-EINCX(I,J,K)
EZ(I+1,J,K)=-EINCY(I+1,J,K)
EZ(I,J,K)=-EINCY(I,J,K)

C

IF(NOPE(I,J,K).EQ.23.OR.NOPE(I,J,K).EQ.123) GO TO 80
GO TO 500

C

80 CONTINUE

C

X-Y PLANE

C

EX(I,J+1,K)=-EINCX(I,J+1,K)
EX(I,J,K)=-EINCX(I,J,K)
EY(I+1,J,K)=-EINCY(I+1,J,K)
EY(I,J,K)=-EINCY(I,J,K)

C

GO TO 500

C

510 CONTINUE

C SET COMPUTATIONAL PARAMETERS

C

8 RSIGMA=1./SIGMA
EPEFFX=(1.0-TX)*EPSO+(TX*EPS)
EPEFFY=(1.0-TY)*EPSO+(TY*EPS)
EPEFFZ=(1.0-TZ)*EPSO+(TZ*EPS)
EXPON=EXP((-SIGMA*DT)/EPS)
EXPDX=EXP((-SIGMA*TX*DT)/EPEFFX)
EXPDY=EXP((-SIGMA*TY*DT)/EPEFFY)
EXPDZ = EXP((-SIGMA*TZ*DT)/EPEFFZ)
XEXP = 1. - EXPON
XEXPX = 1. - EXPDX
XEXPY = 1. - EXPDY
XEXPZ = 1. - EXPDZ

IF(NOPE(I,J,K).NE.8)GO TO 600

SET BOUNDARY CONDITIONS ON LOSSY DIELECTRIC 3-D BODY

SET EX

IF((J-1).EQ.0.OR.(K-1).EQ.0) GO TO 200
EX(I,J,K) = (EX(I,J,K)*EXPON) + ((XEXP)*(-EINCX(I,J,K))
            1  -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K)
            2  +((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J))
            3  -((HY(I,J,K)-HY(I,J,K-1))*DZI(K))*RSIGMA)

IF(NOPE(I,J+1,K).EQ.8)GO TO 150
EX(I,J+1,K) = (EX(I,J+1,K)*EXPON) + ((XEXP)*(-EINCX(I,J+1,K))
            1  -((EPS-EPSO)*RSIGMA)*DEINCX(I,J+1,K)
            2  +((HZ(I,J+1,K)-HZ(I,J,K))*DYI(J+1))
            3  -((HY(I,J+1,K)-HY(I,J+1,K-1))*DZI(K))*RSIGMA)

150 IF(NOPE(I,J,K+1).EQ.8)GO TO 200
EX(I,J,K+1) = (EX(I,J,K+1)*EXPON) + ((XEXP)*(-EINCX(I,J,K+1))
            1  -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K+1)
            2  +((HZ(I,J,K+1)-HZ(I,J-1,K))*DZI(K))
            3  -((HY(I,J,K+1)-HY(I,J,K))*DZI(K+1))*RSIGMA)

C SET EY

C 200 CONTINUE EY

C IF((I-1).EQ.0.OR.(K-1).EQ.0) GO TO 300
EY(I,J,K) = (EY(I,J,K)*EXPON) + ((XEXP)*(-EINCY(I,J,K))
            1  -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
            2  +((HZ(I,J,K)-HX(I,J,K-1))*DXI(I))
            3  -((HZ(I,J,K)-HZ(I,J,K-1))*DZI(K))*RSIGMA)

C IF(NOPE(I+1,J,K).EQ.8)GO TO 250
EY(I+1,J,K) = (EY(I+1,J,K)*EXPON) + ((XEXP)*(-EINCY(I+1,J,K))
            1  -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J,K)
            2  +((HZ(I,J,K)-HX(I,J,K-1))*DXI(I))
            3  -((HZ(I,J,K)-HZ(I,J,K-1))*DZI(K))*RSIGMA)

C
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J,K+1)
2 +(((HX(I+1,J,K+1)-HX(I+1,J,K))*DZI(K+1))
3 -(HZ(I+1,J,K+1)-HZ(I,J,K+1))*DXI(I+1))*RSIGMA))

C

EY(I,J,K)=EY(I,J,K)*EXPON+((XEXP)*(-EINCY(I,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2 +(((HX(I,J,K+1)-HX(I,J,K))*DZI(K+1))
3 -((HZ(I,J,K)-HZ(I,J,K))*DXI(I+1))*RSIGMA))

C

IF(NOPE(I,J,K+1).EQ.8) GO TO 300
C

EY(I+1,J,K+1)=(EY(I+1,J,K)*EXPON)+((XEXP)*(-EINCY(I+1,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J,K)
2 +(((HX(I+1,J,K)-HX(I+1,J,K-1))*DZI(K))
3 -((HZ(I+1,J,K)-HZ(I,J,K))*DXI(I+1))*RSIGMA))

C

SET EZ
C

300 CONTINUE
C

IF((I-1).EQ.0.OR.(J-1).EQ.0) GO TO 500
EZ(I,J,K)=EZ(I,J,K)*EXPON+((XEXP)*(-EINCY(I,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2 +(((HY(I,J,K)-HY(I-1,J,K))*DXI(I))
3 -((HZ(I,J,K)-HZ(I-1,J,K))*DYI(J))*RSIGMA))

C

IF(NOPE(I+1,J,K).EQ.8) GO TO 350
C

EZ(I+1,J,K)=(EZ(I+1,J,K)*EXPON)+((XEXP)*(-EINCY(I+1,J,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J,K)
2 +(((HY(I+1,J,K)-HY(I,J,K))*DXI(I+1))
3 -((HZ(I+1,J,K)-HZ(I,J,K))*DYI(J))*RSIGMA))

C

EZ(I+1,J+1,K)=(EZ(I+1,J+1,K)*EXPON)+((XEXP)*(-EINCY(I+1,J+1,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J+1,K)
2 +(((HY(I+1,J+1,K)-HY(I,J+1,K))*DXI(I+1))
3 -((HZ(I+1,J+1,K)-HZ(I,J+1,K))*DYI(J+1))*RSIGMA))

C

350 IF(NOPE(I+1,J+1,K).EQ.8) GO TO 500
C

EZ(I+1,J+1,K)=(EZ(I+1,J+1,K)*EXPON)+((XEXP)*(-EINCY(I+1,J+1,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I+1,J+1,K)
2 +(((HY(I+1,J+1,K)-HY(I,J+1,K))*DXI(I+1))
3 -((HZ(I+1,J+1,K)-HZ(I,J+1,K))*DYI(J+1))*RSIGMA))

C

EZ(I,J+1,K)=(EZ(I,J+1,K)*EXPON)+((XEXP)*(-EINCY(I,J+1,K)
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J+1,K)
2 +(((HY(I,J+1,K)-HY(I,J+1,K))*DXI(I))
3 -((HZ(I,J+1,K)-HZ(I,J,K))*DYI(J+1))*RSIGMA))

C
GO TO 500

600 CONTINUE

SET BOUNDARY CONDITIONS ON LOSSY DIELECTRIC 2-D BODY

IF((I-1).EQ.0.OR.(J-1).EQ.0.OR.(K-1).EQ.0) GO TO 500

Y-Z PLANE

EY(I,J,K+1)=EY(I,J,K+1)*EXPDX+((XEXPX)*(-EINCY(I,J,K+1))
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K+1)
2 +(((HX(I,J,K+1)-HX(I,J,K))*DZI(K+1))
3 -((HZ(I,J,K+1)-HZ(I-1,J,K+1))*DXI(I)))*RSIGMA*TX))

EY(I,J,K)=EY(I,J,K)*EXPDX+((XEXPX)*(-EINCY(I,J,K))
1 -((EPS-EPSO)*RSIGMA)*DEINCY(I,J,K)
2 +(((HX(I,J,K)-HX(I,J,K-1))*DZI(K))
3 -((HZ(I,J,K)-HZ(I-1,J,K))*DXI(I)))*RSIGMA*TX))

EZ(I,J+1,K)=EZ(I,J+1,K)*EXPDX+((XEXPX)*(-EINCZ(I,J+1,K))
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I,J+1,K)
2 +(((HY(I,J+1,K)-HY(I-1,J+1,K))*DZI(I))
3 -((HZ(I,J+1,K)-HZ(I,J,K))*DXI(I))*RSIGMA*TX))

EZ(I,J,K)=(EZ(I,J,K)*EXPDX)+((XEXPX)*(-EINCZ(I,J,K))
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I,J,K)
2 +(((HY(I,J,K)-HY(I,J,K-1))*DZI(K))
3 -((HZ(I,J,K)-HZ(I-1,J,K))*DXI(J)))*RSIGMA*TX))

IF(NOPE(I,J,K).EQ.56) GO TO 700
IF(NOPE(I,J,K).EQ.57) GO TO 800
IF(NOPE(I,J,K).EQ.567) GO TO 700
GO TO 500

X-Z PLANE

EX(I,J,K)=(EX(I,J,K)*EXPDY)+((XEXPY)*(-EINCX(I,J,K))
1 -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K)
2 +(((HY(I,J,K)-HY(I,J,K-1))*DXI(I))
3 -((HZ(I,J,K)-HZ(I,J,K-1))*DZI(K)))*RSIGMA*TY))

EX(I,J,K+1)=(EX(I,J,K+1)*EXPDY)+((XEXPY)*(-EINCX(I,J,K+1))
1 -((EPS-EPSO)*RSIGMA)*DEINCX(I,J,K+1)
2 +(((HY(I,J,K+1)-HY(I,J,K))*DXI(I))
3 -((HZ(I,J,K+1)-HZ(I,J,K))*DZI(K+1)))*RSIGMA*TY))

EZ(I+1,J,K)=(EZ(I+1,J,K)*EXPDY)+((XEXPY)*(-EINCZ(I+1,J,K))
1 -((EPS-EPSO)*RSIGMA)*DEINCZ(I+1,J,K)
2 +(((HY(I+1,J,K)-HY(I,J,K))*DXI(I+1))

78
C 3  
-((HX(I+1,J,K)-HX(I+1,J-1,K))*DYI(J))*RSIGMA*TY))
EZ(I,J,K)=(EZ(I,J,K)*EXPDY)+((XEXPY)*(-EINCY(I,J,K))
1  
-((EPS-EPSEO)*RSIGMA)*DEINCY(I,J,K)
2  
+(((HZ(I,J,K)-HZ(I,J-1,K))*DZI(K))
3  
-((HZ(I,J,K)-HZ(I,J-1,K))*DZI(K)))*RSIGMA*TY))
C
IF(NOPE(I,J,K).EQ.67.OR.NOPE(I,J,K).EQ.567) GO TO 800
GO TO 500
C
800 CONTINUE
C X-Y PLANE
C
EX(I,J,K)=(EX(I,J,K)*EXPDZ)+((XEXPZ)*(-EINCX(I,J,K))
1  
-((EPS-EPSEO)*RSIGMA)*DEINCX(I,J,K)
2  
+(((HZ(I,J,K)-HZ(I,J-1,K))*DZI(K))
3  
-((HZ(I,J,K)-HZ(I,J-1,K))*DZI(K)))*RSIGMA*TZ))
C
EX(I,J+1,K)=(EX(I,J+1,K)*EXPDZ)+((XEXPZ)*(-EINCX(I,J+1,K))
1  
-((EPS-EPSEO)*RSIGMA)*DEINCX(I,J+1,K)
2  
+(((HZ(I,J+1,K)-HZ(I,J,K))*DZI(K))
3  
-((HZ(I,J+1,K)-HZ(I,J,K))*DZI(K)))*RSIGMA*TZ))
C
EY(I,J,K)=(EY(I,J,K)*EXPDZ)+((XEXPZ)*(-EINCY(I,J,K))
1  
-((EPS-EPSEO)*RSIGMA)*DEINCY(I,J,K)
2  
+(((HZ(I,J,K)-HZ(I,J,K-1))*DZI(K))
3  
-((HZ(I,J,K)-HZ(I,J,K))*DZI(K)))*RSIGMA*TZ))
C
EY(I+1,J,K)=(EY(I+1,J,K)*EXPDZ)+((XEXPZ)*(-EINCY(I+1,J,K))
1  
-((EPS-EPSEO)*RSIGMA)*DEINCY(I+1,J,K)
2  
+(((HZ(I+1,J,K)-HZ(I+1,J,K-1))*DZI(K))
3  
-((HZ(I+1,J,K)-HZ(I+1,J,K))*DZI(K)))*RSIGMA*TZ))
C
500 CONTINUE
501 CONTINUE
C
NWIRE=1
IF(NWIRE.EQ.0) GO TO 550
IF(M.NE.2) GO TO 550
C
B1 TO B2
C
I=4
K=13
DO 900 J=10,14
EY(I,J,K)=-EINCY(I,J,K)
900 CONTINUE
C
EX(4,10,13)=-EINCX(4,10,13)
EZ(5,10,13)=-EINCY(5,10,13)
EZ(5,10,14)=-EINCy(5,10,14)
EX(5,10,15)=-EINCX(5,10,15)
EZ(6,10,15)=-EINCY(6,10,15)
EZ(6,10,16)=-EINCY(6,10,16)
B2 THROUGH B4

EX(6,10,17)=-EINCX(6,10,17)
EZ(7,10,17)=-EINCL(7,10,17)

J=10
K=18
DO 901 I=7,10
EX(I,J,K)=-EINCX(I,J,K)
901 CONTINUE

AFT OF B4 THROUGH B6

EZ(11,10,18)=-EINCL(11,10,18)

J=10
K=19
DO 902 I=11,16
EX(I,J,K)=-EINCX(I,J,K)
902 CONTINUE

AFT OF B6 TO B8

DO 903 J=10,13
EY(I,J,K)=-EINCY(I,J,K)
903 CONTINUE

EX(17,14,19)=-EINCX(17,14,19)
EY(18,13,19)=-EINCY(18,13,19)
EX(18,13,19)=-EINCX(18,13,19)
LY(19,13,19)=-EINCY(19,13,19)
EX(19,14,19)=-EINCX(19,14,19)

B8 ON TO END

EX(20,14,19)=-EINCX(20,14,19)
EZ(21,14,18)=-EINCL(21,14,18)
EZ(21,14,17)=-EINCL(21,14,17)

J=14
K=17
DO 904 I=21,25
EX(I,J,K)=-EINCX(I,J,K)
904 CONTINUE

EY(26,13,17)=-EINCY(26,13,17)
EY(26,12,17)=-EINCY(26,12,17)
EY(26,11,17)=-EINCY(26,11,17)

EX(26,11,17)=-EINCX(26,11,17)

EY(27,11,17)=-EINCY(27,11,17)
EY(27,12,17) = -EINCY(27,12,17)
C
EX(27,13,17) = -EINCX(27,13,17)
C
550 CONTINUE
C
RETURN
C
END
3.5.5 Subroutine ERAD

Subroutine ERAD evaluates the expression

\[ E(R, D, EL1, EL2, EL3, L1, L2, L3, TH) = \frac{R}{D^*} \]

\[ = (EL1 \cdot (L2 - L3) \cdot (L2 \cdot L3 + TH \cdot (L2 + L3) + TH^{**2}) \cdot \]

\[ EL2 \cdot (L3 - L1) \cdot (L3 \cdot L1 + TH \cdot (L3 + L1) + TH^{**2}) \cdot \]

\[ EL3 \cdot (L1 - L2) \cdot (L1 \cdot L2 + TH \cdot (L1 + L2) + TH^{**2})) \].

It substitutes the results for the EADV-determined values of the field components which are on the surface of the problem cell space (shown in Figure 9) according to the prescription:

\[ EX(I, 1, K) = E(RXY(I, K, 1), D, EX(I, 2, K), EXYD(I, K, 2), \]

\[ EXYD(I, K, 3), L1, L2, L3, THXY(I, K, 1)) \]

\[ EZ(I, NY, K) = E(RZY(I, K, 2), D, EZ(I, NY1, K), EZYU(I, K, 2), \]

\[ EZYU(I, K, 3), L1, L2, L3, THZY(I, K, 2)) \]

This substitution imposes a far field behavior on the outermost region of the problem cell space. Equation (1) is derived directly from the assumption that the outermost electric field components of Figure 9 and the corresponding electric fields that are one cell in from the outer surface, behave as far fields,\(^*\) that is,

\[ E = f(\theta, \phi)g(t-r/c) / r \] or equivalently

\[ Er = f(\theta, \phi)g(t-r/c) \]

\(^*\)As a result, fields on the outermost edges are not written over, but this does not effect the interior results.
Figure 9. Field Components at the Outer Boundary of the Cell Space
Thus, taking $E^{n+1/2}_y(I, J, K)$ as an example, 

$$R_{yx}(1, J, K) E^{n+1/2}_y(1, J, K) = e_y(J, K, [(n+1/2)\Delta t - R_{yx}(1, J, K)/c])$$  \hspace{1cm} (4)$$

$$R_{yx}(2, J, K) E^{n+1/2}_y(2, J, K) = e_y(J, K, [(n+1/2)\Delta t - R_{yx}(2, J, K)/c])$$  \hspace{1cm} (5)$$

where $R_{yx}$ is the distance to the $E_y$ component.

What is desired is $E^{n+1/2}_y(I, J, K)$ expressed in terms of $E^{n+1/2-\ell_1}_y(2, J, K)$, i.e., the outer boundary field component expressed in terms of an interior corresponding field component at various time offsets($\ell_1$). Equation (1) is, therefore, rewritten as,

$$R_{yx}(1, J, K) E^{n+1/2}_y(1, J, K) = e_y(J, K, [(n-1/2)\Delta t + \Delta t - R_{yx}(2, J, K)/c + \theta_{yx}(J, K)\Delta t])$$  \hspace{1cm} (6)$$

where

$$\theta_{yx} = 1 - \frac{R_{yx}(1, J, K) - R_{yx}(2, J, K)}{c\Delta t}$$  \hspace{1cm} (7)$$

and Equation (2) is rewritten with a time shift $\ell_1$ as,

$$R(2, J, K) E^{n-1/2-\ell_1}_y(2, J, K) = e_y(J, K, [(n-1/2)\Delta t - R_{yx}(2, J, K)/c - \ell_1\Delta t])$$  \hspace{1cm} (8)$$
It can be seen that Equations (6) and (8) are identical except for $l_1$ in place of $\Theta_{yx}$ in Equation (8). Either Equation (6) or (8) can be expanded in terms of the time $l_1\Delta t$ or $\Theta_{yx}\Delta t$. The first three terms of Maclaurin's series are used so that

$$R_{yx}(1,J,K) E_n^{n+1/2} (1,J,K) = A + B (\Theta_{yx}\Delta t) + C (\Theta_{yx}\Delta t)^2 \quad (9)$$

$$R_{yx}(2,J,K) E_n^{n+1/2-l_1}(2,J,K) = A + B (l_1\Delta t) + C (l_1\Delta t)^2. \quad (10)$$

If three different values of $l_1$ are selected, then, from Equation (10), we obtain three equations for the three unknown $A$, $B$ and $C$ in terms of the three values of $E_n^{n+1/2-l_1}$, the three values of $l_1$ and $R_{yx}(2,J,K)$, all of which are known if $l_1 \geq -1$. Therefore, $A$, $B$ and $C$ can be determined and are

$$A = [E_n^{n-1/2-l_1}(2,J,K) l_2 l_3 (l_2 - l_3)] R_{yx}(2,J,K)/D \quad (11)$$

$$B = [E_n^{n-1/2-l_1}(2,J,K) (l_2^2 - l_3^2)] R_{yx}(2,J,K)/D\Delta t \quad (12)$$

$$C = [E_n^{n-1/2-l_1}(2,J,K) (l_2^2 - l_3^2)] R_{yx}(2,J,K)/D\Delta t^2 \quad (13)$$
where $D$ is the determinant

$$D = l_2 l_3 (l_2 - l_3) + l_3 l_1 (l_3 - l_1) + l_1 l_2 (l_1 - l_2) \quad (14)$$

Knowing $A$, $B$ and $C$, Equation (9) immediately yields

$$E_{y}^{n+1/2}(1,J,K) = A + B(\theta_y \Delta t) + C(\theta_y \Delta t)^2$$

$$\frac{R_{yx}(1,J,K)}{R_y(1,J,K)} \quad (15)$$

This is the desired result of $E_{y}^{n+1/2}(I,J,K)$ expressed in terms of $E_{y}^{n-1/2}$ and other known quantities, namely $R_{yx}(I,J,K)$, $R_{yx}(2,J,K)$ and $l_1$, $l_2$ and $l_3$.

The last equation, Equation (15), is identical to the expression, Equation (1), that ERAD evaluates.

For example, the terms $EX(I,2,K)$, $EXYD(I,K,2)$, and $EXYD(I,K,3)$ in Equation (2a) are the backstored values of $EX$ at $J = 2$ that are used to calculate $EX(I,1,K)$. They are found in Subroutine EADV. The quantities $D$, $l_1$, $l_2$, $l_3$, THXY(I,K,1) correspond to $D$, $l_1$, $l_2$, and $\theta_y(I,K)$. $D$ is calculated in ERAD as are $l_1$, $l_2$ and $l_3$ and the corresponding stored values of $E$ one cell in from the outer boundary. The $l_1$'s are spread enough in time to correspond to the propagation time across the expanded cell boundary. Reference 2 details what requirements on the $l_1$ and, hence, on $N_2$ and $N_3$ in EADV must be met. Only when constant mesh size is used can the $l_1 = -1$, $l_2 = 0$, $l_3 = 1$. In this case, the expression in Equation (1) goes over exactly to the expression in Section 2.3. The value of $R_{xy}(I,K,1)$ (which represents $R_{xy}(I,K,2)/R_{xy}(I,K,1)$ in this subroutine) is determined in Subroutine SETUP, as is the value of THXY (I,K,1). Since they are geometry dependent quantities, they need only be found once, hence, their presence in SETUP.

Flow for Subroutine ERAD is as shown in Figure 10. Local variables are described in Table 5. Subroutine ERAD is called by EADV.
Table 5. Local Variables - Subroutine ERAD

<table>
<thead>
<tr>
<th>Variable NAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Determinant</td>
</tr>
<tr>
<td>L1</td>
<td>Time shifts for backstored data in number of cycles</td>
</tr>
<tr>
<td>L2</td>
<td></td>
</tr>
<tr>
<td>L3</td>
<td></td>
</tr>
</tbody>
</table>
Start

Define Function E (R,D,EL1,EL2,EL3,L1,L2,L3,TH)

Calculate L1,L2,L3 Using Values of N2,N3 From EADV

Evaluates E at the Six Outer Faces (for the Two E-Field Components on Each Face) (Uses Backstored Interior E-Field Data from EADV and Geometrical Quantities Determined in SETUP)

Return

Figure 10. Flow Chart for Subroutine ERAD
SUBROUTINE ERAD

COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DX0I(28),DY0I(28),DZ0I(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,N,M,Q,D,D,MU,EPSON,EPS,PNTLim,
1 NN,NPTS,LMAX,SIGMA,C,T,IP,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,
2 BETA,IDLs

COMMON/RAD/RXY(28,29,2),RXZ(29,28,2),RYX(28,29,2),RZY(29,28,2),
1 RXZ(28,29,2),RYZ(29,28,2),THXY(29,28,2),THXZ(28,29,2),THYZ(29,28,2),
2 THXY(28,29,2),THZY(29,28,2),THXZ(28,29,2),THYZ(29,28,2),
COMMON/EBs/EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZXU(29,28,3),
1 EYXD(28,29,3),EYXU(28,29,3),EZXD(29,28,3),EZXU(29,28,3),
2 EXZD(28,29,3),EXZU(28,29,3),EYZD(29,28,3),EYZU(29,28,3),
3 N1,N2,N3

E(R,D,EL1,EL2,EL3,L1,L2,L3,TH) = R/D*
1 EL1*(L2-L3)*(L2*L3+TH*(L2+L3)+TH**2)+
2 EL2*(L3-L1)*(L3*L1+TH*(L3+L1)+TH**2)+
3 EL3*(L1-L2)*(L1*L2+TH*(L1+L2)+TH**2))

L1=-1
L2=N-N2-1
L3=N-N3-1
IF(D.EQ.0.) D=1.E-9

RADIATE EXY

DO 11 I=1,NX1
DO 11 K=2,NZ1
EX(I,K)=E(RXY(I,K,1),D,EX(I,2,K),EYXD(I,K,2),EYXU(I,K,3),
1 L1,L2,L3,THXY(I,K,1))
11 CONTINUE

RADIATE EXZ

DO 111 I=1,NX1
DO 111 J=2,NY1
EX(I,J)=E(RXZ(I,J,1),D,EX(I,J,2),EXZD(I,J,2),EXZU(I,J,3),
1 L1,L2,L3,THXZ(I,J,1))
111 CONTINUE

RADIATE EYX

DO 22 J=1,NY1
DO 22 K=2,NZ1
EY(I,J)=E(RYX(J,K,1),D,EY(2,J,K),EYXD(J,K,2),EYXU(J,K,3),
1 L1,L2,L3,THYX(J,K,1))
EY(NX, J, K) = E(RYX(J, K, 2), D, EY(NX1, J, K), EYXU(J, K, 2), EYXU(J, K, 3),
1 L1, L2, L3, THYX(J, K, 2))
22 CONTINUE
C RADIATE EYZ
C DO 222 I = 2, NX1
DO 222 J = 1, NY1
EY(I, J, 1) = E(RYZ(I, J, 1), D, EY(I, J, 2), EYZD(I, J, 2), EYZD(I, J, 3),
1 L1, L2, L3, THYZ(I, J, 1))
EY(I, J, NZ) = E(RYZ(I, J, 2), D, EY(I, J, NZ1), EYZU(I, J, 2), EYZU(I, J, 3),
1 L1, L2, L3, THYZ(I, J, 2))
222 CONTINUE
C RADIATE EZX
C DO 33 J = 2, NY1
DO 33 K = 1, NZ1
EZ(1, J, K) = E(RZX(J, K, 1), D, EZ(2, J, K), EZXD(J, K, 2), EZXD(J, K, 3),
1 L1, L2, L3, THZX(J, K, 1))
EZ(NX, J, K) = E(RZX(J, K, 2), D, EZ(NX1, J, K), EZXU(J, K, 2), EZXU(J, K, 3),
1 L1, L2, L3, THZX(J, K, 2))
33 CONTINUE
C RADIATE EZY
C DO 333 I = 2, NX1
DO 333 K = 1, NZ1
EZ(I, 1, K) = E(RZY(I, K, 1), D, EZ(I, 2, K), EZYD(I, K, 2), EZYD(I, K, 3),
1 L1, L2, L3, THZY(I, K, 1))
EZ(I, NY, K) = E(RZY(I, K, 2), D, EZ(I, NY1, K), EZYU(I, K, 2), EZYU(I, K, 3),
1 L1, L2, L3, THZY(I, K, 2))
333 CONTINUE
C RETURN
END
3.5.6 Subroutine HADV

In subroutine HADV, the equation
\[ \nabla \times E = -\mu \frac{\partial H}{\partial t} \quad (16) \]
is applied to the scattered fields and is used to find H from E and H at earlier times. This "time stepping" solution is found for each component of H, i.e., Equation (16) is written explicitly for each component as

\[ -\frac{\partial H_s^x}{\partial t} = \frac{\partial E_s^z}{\partial y} - \frac{\partial E_s^y}{\partial z} \]
\[ -\frac{\partial H_s^y}{\partial t} = \frac{\partial E_s^x}{\partial z} - \frac{\partial E_s^z}{\partial x} \]
\[ -\frac{\partial H_s^z}{\partial t} = \frac{\partial E_s^y}{\partial x} - \frac{\partial E_s^x}{\partial y} \]

These equations are cast in their finite-difference form and solved for H at the latest time. Thus

\[ H_s^n(I,J,K) = H_s^{n-1}(I,J,K) - (\Delta t/\mu) \left( \frac{E_s^{n-1/2}(I,J+1,K) - E_s^{n-1/2}(I,J,K)}{Y_o(J+1) - Y_o(J)} \right. \]
\[ - \left. \frac{E_s^{n-1/2}(I,J,K+1) - E_s^{n-1/2}(I,J,K)}{Z_o(K+1) - Z_o(K)} \right) \]

\[ H_s^n(I,J,K) = H_s^{n-1}(I,J,K) - (\Delta t/\mu) \left( \frac{E_s^{n-1/2}(I,J,K+1) - E_s^{n-1/2}(I,J,K)}{Z_o(K+1) - Z_o(K)} \right. \]
\[ - \left. \frac{E_s^{n-1/2}(I+1,J,K) - E_s^{n-1/2}(I,J,K)}{X_o(I+1) - X_o(I)} \right) \]
\[
H_z^N(I,J,K) = H_z^{n-1}(I,J,K) - (\Delta t/\mu) \left( \frac{E_y^{n-1/2}(I+1,J,K) - E_y^{n-1/2}(I,J,K)}{X_o(I+1) - X_o(I)} \right) 
- \frac{E_x^{n-1/2}(I,J+1,K) - E_x^{n-1/2}(I,J,K)}{Y_o(J+1) - Y_o(J)} 
\]  

(19)

It is assumed that \( \mu = \mu_0 \) for all regions of the problem space.

HADV is called every program cycle by DRIVER to advance the H-fields, just as Subroutine EADV is called every program cycle to advance the E-fields. It uses the prior values of the fields to calculate the new values. Initially, these prior values are zero. The first nonzero values are those values for E. These values are accessed by a call to EBC in the Subroutine EADV for advancing the E-fields. Thus, on the next program cycle these values will be available and nonzero values of the fields off the boundary will start appearing when Equations (17), (18), and (19) are advanced again. Since EBC is always called after the fields are advanced, the proper boundary conditions are maintained.

HADV also defines a loop of H fields that drives the lightning channel for M=1.

The only variable used locally in HADV is DTMU, which is equal to \( \Delta t/\mu \). Flow for the subroutine is shown in Figure 11.
Figure 11. Flow Chart for Subroutine HADV
SUBROUTINE HADV

COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/EEFIELD/HX(29,29,29),HY(29,29,29),HZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,M1,DT,XMU,EPS0,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFA,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

DTXMU = DT/XMU

C ADVANCE HX
C
DO 1 I = 2,NX1
DO 1 J = 1,NY1
DO 1 K = 1,NZ1
HX(I,J,K) = HX(I,J,K) - DTXMU*((EZ(I,J+1,K) - EZ(I,J,K))*DYOI(J)
1 - (EY(I,J,K+1) - EY(I,J,K))*DZOI(K))
1 CONTINUE

C ADVANCE HY
C
DO 2 I = 1,NX1
DO 2 J = 2,NY1
DO 2 K = 1,NZ1
HY(I,J,K) = HY(I,J,K) - DTXMU*((EX(I,J,K+1) - EX(I,J,K))*DZOI(K)
1 - (EZ(I+1,J,K) - EZ(I,J,K))*DXOI(I))
2 CONTINUE

C ADVANCE HZ
C
DO 3 I = 1,NX1
DO 3 J = 1,NY1
DO 3 K = 2,NZ1
HZ(I,J,K) = HZ(I,J,K) - DTXMU*((EY(I+1,J,K) - EY(I,J,K))*DXOI(I)
1 - (EX(I,J+1,K) - EX(I,J,K))*DYOI(J))
3 CONTINUE

IF(IDLS.EQ.0) GO TO 1

IF(M.NE.1) GO TO 1

M=1 LOOP,H-LOOP SOURCE

M=2 LOOP NOT TREATED-SOURCE MUST BE OUTSIDE THE PROBLEM SPACE
AND THE WIRES ARE ALSO OUTSIDE

EXPAB=EXP(-ALPHA*T)-EXP(-BETA*T)

HX(8,21,15)=-(33333.)*EXPAB
HZ(8,21,15)=(16667.)*EXPAB
HX(8,21,14)=(33333.)*EXPAB
HZ(7,21,15)=-(16667.)*EXPAB

94
C
C 1 CONTINUE
C
RETURN
END
There are nine functions* used for calculating the six incident E- and H-field components: EINCX, EINCY, EINCZ, EINCXP, EINCYP, EINCP, HINCXP, HINCP, and HINCP. The first letter in the function, E or H, tells what field; the next three letters, INC, indicate that these are incident field components; the next letter, X, Y or Z, states what component is evaluated at a point in space located at XP, YP, ZP and if P is not present the field component is evaluated at the field component location in the cell space labeled by I,J,K. EINCX, EINCY, and EINCZ are called from Subroutine EBC. The others are called by Subroutine DATASAV.

Two different sources for the incident fields are considered:
1) direct strikes;
2) indirect illumination

Direct strikes are modeled by H-fields impressed around a perfectly conducting wire leading to the aircraft that represents the lightning channel. If a lightning channel resistance is desired, then use the method outlined in Appendix D, in the discussion on excitation sources. The specified H-fields determine the lightning channel's current time history. Indirect illumination is modeled by the fields directly illuminating the aircraft, such as EINCX and HINCP broadside illumination from above.

* While these functions are not subroutines per se, they are similar enough in nature to warrant inclusion in this section, as opposed to having a separate short section of their own.
FUNCTION EINCX(I,J,K)
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

THREAT
EINCX = 0.0
IF(IDLS.EQ.1) GO TO 1
TAU = T - (UYO(23) - YO(J))/C
IF (TAU.LE.0.0) GO TO 1
EINCX=AMP*(EXP(-ALPHA*TAU)-EXP(-BETA*TAU))
1 CONTINUE
RETURN
END

FUNCTION DEINCX(I,J,K)
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
DEINCX=0.0
IF(IDLS.EQ.1) GO TO 1
TAU=T-(UYO(23)-YU(J))/C
IF (TAU.LE.0.0) GO TO 1
DEINCX=AMP*(BETA*(EXP(-BETA*TAU))-ALPHA*(EXP(-ALPHA*TAU)))
1 CONTINUE
RETURN
END

FUNCTION EINCY(I,J,K)
COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

THREAT
EINCY = 0.0
RETURN
END

FUNCTION DEINCY(I,J,K)
DEINCY=0.0
RETURN
FUNCTION EINCZ(I,J,K)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYM(29),DZI(29),DXO(28),DYO(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NXI,NY1,NZ1,N,M,Q,DXU,DYU,DZU,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/EXTRAS/NX, NX, NY, NZ, NX1, NY1, NZ1, N, M, Q, DT, XMU, EPSO, EPS, NPTLIM,
1 NN, NPTS, LMAX, SIGMA, C, T, PI, EXPFAC, IP, TX, TY, TZ, AMP, ALPHA, BETA, IDLS
THREAT
EINCZ = 0.0
RETURN
END

FUNCTION DEINCZ(I,J,K)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYM(29),DZI(29),DXO(28),DYO(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NXI,NY1,NZ1,N,M,Q,DXU,DYU,DZU,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
DEINCZ=0.0
RETURN
END

FUNCTION HINCX(I,J,K)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYM(29),DZI(29),DXO(28),DYO(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NXI,NY1,NZ1,N,M,Q,DXU,DYU,DZU,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
THREAT
HINCX=0.0
RETURN
END

FUNCTION HINCY(I,J,K)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYM(29),DZI(29),DXO(28),DYO(28),DZOI(28)
COMMON/EXTRAS/NX,NY,NZ,NXI,NY1,NZ1,N,M,Q,DXU,DYU,DZU,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
THREAT
HINCY=0.0
RETURN
END
FUNCTION HINCZ(I, J, K)

COMMON/UGRID/UX(28), UY(28), UZ(28), UX0(29), UYO(29), UZO(29)
COMMON/GRID/X(28), Y(28), Z(28), XO(29), YO(29), ZO(29),
1 DX(29), DY(29), DZ(29), DX0(28), DYO(28), DZO(28),
2 DXI(29), DYS(29), DZI(29), DX0I(28), DYOI(28), DZOI(28)
COMMON/EXTRAS/NX, NY, NZ, NX1, NY1, NZ1, N, M, HQ, DT, XMU, EPSO, EPS, NPTLIM,
1 NN, NPTS, LMAX, SIGMA, C, T, PI, EXPFAC, IP, TX, TY, TZ, AMP, ALPHA, BETA, IDLS

THREAT

HINCZ = 0.0
RETURN
END

FUNCTION EINCXP(XP, YP, ZP)

COMMON/UGRID/UX(28), UY(28), UZ(28), UX0(29), UYO(29), UZO(29)
COMMON/GRID/X(28), Y(28), Z(28), XO(29), YO(29), ZO(29),
1 DX(29), DY(29), DZ(29), DX0(28), DYO(28), DZO(28),
2 DXI(29), DYS(29), DZI(29), DX0I(28), DYOI(28), DZOI(28)
COMMON/EXTRAS/NX, NY, NZ, NX1, NY1, NZ1, N, M, HQ, DT, XMU, EPSO, EPS, NPTLIM,
1 NN, NPTS, LMAX, SIGMA, C, T, PI, EXPFAC, IP, TX, TY, TZ, AMP, ALPHA, BETA, IDLS

THREAT

EINCXP = 0.0
IF(IDLS.EQ.1) GO TO 1
TAU = T - (UYO(23) - YP)/C
IF (TAU.LE.0) GO TO 1
EINCXP = AMP * (EXP(-ALPHA*TAU) - EXP(-BETA*TAU))
1 CONTINUE
RETURN
END

FUNCTION EINCYP(XP, YP, ZP)

COMMON/UGRID/UX(28), UY(28), UZ(28), UX0(29), UYO(29), UZO(29)
COMMON/GRID/X(28), Y(28), Z(28), XO(29), YO(29), ZO(29),
1 DX(29), DY(29), DZ(29), DX0(28), DYO(28), DZO(28),
2 DXI(29), DYS(29), DZI(29), DX0I(28), DYOI(28), DZOI(28)
COMMON/EXTRAS/NX, NY, NZ, NX1, NY1, NZ1, N, M, HQ, DT, XMU, EPSO, EPS, NPTLIM,
1 NN, NPTS, LMAX, SIGMA, C, T, PI, EXPFAC, IP, TX, TY, TZ, AMP, ALPHA, BETA, IDLS

EINCYP = 0.0
RETURN
END

FUNCTION EINCZP(XP, YP, ZP)

COMMON/UGRID/UX(28), UY(28), UZ(28), UX0(29), UYO(29), UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXO(28),DYO(28),DZO(28),
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
EINCZP = 0.0
RETURN
END

FUNCTION HINCXP(XP,YP,ZP)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXO(28),DYO(28),DZO(28),
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
HINCXP = 0.0
RETURN
END

FUNCTION HINCYP(XP,YP,ZP)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXO(28),DYO(28),DZO(28),
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
HINCYP = 0.0
RETURN
END

FUNCTION HINCZP(XP,YP,ZP)

COMMON/UGRID/UX(28),UY(28),UZ(28),UXO(29),UYO(29),UZO(29)
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DXO(28),DYO(28),DZO(28),
2 DXI(29),DYI(29),DZI(29),DXO(28),DYO(28),DZO(28),
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
THREAT
HINCZP=0.0
RETURN
END
3.5.8 Subroutine SAVESB

The subroutine SAVESB saves the tangential E-field components on the subboundary selected for the M=1 loop, when the model is unexpanded. The subboundary is defined by the values set in a data statement for INEAR, JNEAR, KNEAR, IFAR, JFAR, KFAR. The values must be consistent with the expansion factor value selected. For example, an expansion factor of 4 (2, 4, 7 and 14 are allowed for a problem space 28 cells on a side) require seven cells on a side in the subboundary volume. In this case the values of INEAR and IFAR would be N and N+7, where N is an integer from 1 to 22.

SAVESB treats opposing sides of the subboundary in succession (INEAR and IFAR, JNEAR and JFAR, KNEAR and KFAR sides respectively), saving for each side the two E-field components tangential to the surface (EY and EZ for the INEAR and IFAR sides for example) in the ARAY (L, IT). These components on a seven cell on a side subboundary form 9x8 and 8x9 arrays on a side. This provides field components on the edges of the subboundary (8 field components field components on each cell edge) or bracketing the edges of the subboundary (9 field components across and one cell on either side of the face at each cell center). As a result, in OUTBND it is possible to interpolate field values between the ARAY stored values using a single simple interpolation scheme. If the components running 9 across and about a face were not saved, rather only 7 across the face, then an extrapolation scheme for the fields along the edges would be required, unnecessarily complicating matters.

With 9x8 and 8x9 arrays of two field components on a face and 6 faces, the total number of stored components for a single call to SAVESB is 864, when an expansion factor of 4 is selected. SAVESB is called for each time step, which is labeled by IT in SAVESB, and ARAY is dimensioned as ARAY (864, 500), when the expansion factor is 4 and no more than 500 time steps are required in the M=1 loop. These dimensions must be manually reset for other cases. Figure 13 is a flow chart of SAVESB.
Figure 12. SAVESB Flow Chart
SUBROUTINE SAVESB

COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XNU,EPS0,EPS,NPTLIM,
  N,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDL

COMMON/RAY/ARRAY(864,500)

DATA INEAR,JNEAR,KNEAR,IFAR,JFAR,KFAR/7,9,12,14,16,19/

THIS SUBROUTINE SAVES THE TANGENTIAL E FIELD COMPONENTS ON THE
SUBBOUNDARY

EXPFAC=4.0
LIMIT=EXPACEMENT
L=1
IF(N.NE.0) GO TO 10
IT=1
PRINT 5
5 FORMAT(*SAVESB CALLED*)
10 CONTINUE

IMIN=INEAR
JMIN=JNEAR
KMIN=KNEAR

C EY ON INEAR,J,K SIDE

IL=IMIN-1
JL=JMIN-1
KL=KMIN-1

DO 90 J=JL,JMAX
DO 90 K=KL,KMAX
ARRAY(L,IT)=EY(INEAR,J,K)
L=L+1
90 CONTINUE

C EY ON IFAR,J,K SIDE

DO 110 J=JL,JMAX
DO 110 K=KMIN,KMAX
ARRAY(L,IT)=EY(IFAR,J,K)
L=L+1
110 CONTINUE

C EZ ON INEAR,J,K SIDE

DO 130 J=JMIN,JMAX
DO 130 K=KL,KMAX
AKAY(L, IT) = EZ(INEAR, J, K)
L = L + 1
130 CONTINUE

C EZ ON IFAR, J, K SIDE
C
DO 150 J = JMIN, JMAX
DO 150 K = KMIN, KMAX
ARAY(L, IT) = EZ(IFAR, J, K)
L = L + 1
150 CONTINUE
C
C EX ON I, J, KNEAR SIDE
C
DO 170 I = IL, IMAX
DO 170 J = JMIN, JMAX
NCAY(L, IT) = EX(I, J, KNEAR)
L = L + 1
170 CONTINUE
C
C EX ON I, J, KFAR SIDE
C
DO 190 I = IL, IMAX
DO 190 J = JMIN, JMAX
ARAY(L, IT) = EX(I, J, KFAR)
L = L + 1
190 CONTINUE
C
C EY ON I, J, KNEAR SIDE
C
DO 210 I = IMIN, IMAX
DO 210 J = JMIN, JMAX
AY(L, IT) = EY(I, J, KNEAR)
L = L + 1
210 CONTINUE
C
C EY ON I, J, KFAR SIDE
C
DO 230 I = IMIN, IMAX
DO 230 J = JMIN, JMAX
KAY(L, IT) = EY(I, J, KFAR)
L = L + 1
230 CONTINUE
C
C EX ON I, JNEAR, K SIDE
C
DO 250 I = IL, IMAX
DO 250 K = KMIN, KMAX
ARAY(L, IT) = EX(I, JNEAR, K)
L = L + 1
250 CONTINUE
C
C EX ON I,JFAR,K SIDE
C
DO 270 I=IL,IMAX
DO 270 K=KMIN,KMAX
ARRAY(L,IT)=EX(I,JFAR,K)
L=L+1
270 CONTINUE
C EZ ON I,JNEAR,K SIDE
C
DO 290 I=MIN,IMAX
DO 290 K=KL,KMAX
ARRAY(L,IT)=EZ(I,JNEAR,K)
L=L+1
290 CONTINUE
C C EZ ON I,JFAR,K SIDE
C
DO 310 I=MIN,IMAX
DO 310 K=KL,KMAX
ARRAY(L,IT)=EZ(I,JFAR,K)
L=L+1
IF(IT.NE.1) GO TO 310
310 CONTINUE
C IT=IT+1
RETURN
END
3.5.9 Subroutine OUTBND

Subroutine OUTBND in the M=2 loop reads the data from ARAY acquired in SAVESB in the M=1 loop via a common statement. The data is first interpolated in time with each time step replaced by an expansion factor (EXPFAC or as in the present code 4) of time steps. Next the data is interpolated spatially between the unexpanded grid points (UX, UY, UZ, UXO, UYO, UZO of COMMON/UGRID/...) with a simple linear scheme and the expanded grid point locations (X,Y,Z,XO,YO,ZO of COMMON/GRID/...). The interpolated data is placed on the faces of the problem space in the M=2 loop and acts in lieu of the radiation boundary condition defined in ERAD that is used on the M=1 loop. The same sequence of faces and components as used in SAVESB is used in OUTBND to insure that the interpolated data appears on the correct face in the correct location. Figure 14 is a flow chart of OUTBND.
Figure 13. OUTBND Flow Chart
SUBROUTINE OUTBND

COMMON/EFIELD/EX(29,29,29),EY(29,29,29),EZ(29,29,29)
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
  DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
  DX1(29),DY1(29),DZ1(29),DX01(28),DY01(28),DZ01(28)
COMMON/GRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XNU,EPSO,EPS,NPTLIM,
  NN,NPTS,MAX,SIGNA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS

COMMON/RAY/ARRY(864,500)
DIMENSION ARRAY(864),CRAY(9,9)

DATA INEAR,JNEAR,KNEAR,IFAR,KFAR/7,9,12,14,16,19/
IF(N.NE.0) GO TO 10
PRINT 5
5 FORMAT(*OUTBND CALLED*)
IT=1
10 CONTINUE
E=MOD(N,4)

DO 50 LL=1,864
ARRAY(LL)=ARRY(LL,IT)+(E/EXPFAC)*(ARRY(LL,IT+1)-ARRY(LL,IT))
50 CONTINUE

IF(E.NE.3) GO TO 20
IT=IT+1
20 CONTINUE

LIMIT=28/EXPFAC
LONE=LIMIT+1
LTWO=LIMIT+2
LL=1
DO 80 J=1,LTWO
DO 80 K=1,LONE
CRAY(J,K)=ARRAY(LL)
LL=LL+1
80 CONTINUE

DO 90 J=1,28
JJ=INT((J+1)/EXPFAC)+1
DO 90 K=1,28
KK=INT((K-2)/EXPFAC)+1
JS=JJ+JNEAR-1
KS=KK+KNEAR-1
EY1=CRAY(JJ+1,KK)+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS))
  *(CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
EY2=CRAY(JJ,KK)+((ZO(K)-UZO(KS))/(UZO(KS+1)-UZO(KS))
  *(CRAY(JJ,KK+1)-CRAY(JJ,KK))
EY1(J,K)=EY2+((Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
90 CONTINUE
C
   DO 100 J=1,LTWO
   DO 100 K=1,LONE
   CRAY(J,K)=ARRAY(LL)
   LL=LL+1
   100 CONTINUE

C
   DO 110 J=1,28
      JJ=INT((J+1)/EXPFAC)+1
      DO 110 K=1,28
         KK=INT((K-2)/EXPFAC)+1
         JS=JJ+JNEAR-1
         KS=KK+KNEAR-1
         EY1=CRAY(JJ+1,KK)+((Z0(K)-UZ0(KS))/(UZ0(KS)-UZ0(KS)))
         * (CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
         EY2=CRAY(JJ,KK)+((Z0(K)-UZ0(KS))/(UZ0(KS)+1)-UZ0(KS)))
         * (CRAY(JJ,KK+1)-CRAY(JJ,KK))
         EY(29,J,K)=EY2+((Y(J)-U(1)(JS-1))/(U(1)(JS)-U(1)(JS-1)))
         * (EY1-EY2)
   110 CONTINUE

C
   DO 120 J=1,LONE
   DO 120 K=1,LTWO
   CRAY(J,K)=ARRAY(LL)
   LL=LL+1
   120 CONTINUE

C
   DO 130 J=1,28
      JJ=INT((J-2)/EXPFAC)+1
      DO 130 K=1,28
         JS=JJ+JNEAK-l
         KS=KK+KNEAK-1
         EZ1=CRAY(JJ+1,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
         * (CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))
         EZ2=CRAY(JJ,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
         * (CRAY(JJ,KK+1)-CRAY(JJ,KK))
         EZ(1,J,K)=EZ2+((YO(J)-UYO(JS))/(UYO(JS+1)-UYO(JS)))
         * (EZ1-EZ2)
   130 CONTINUE

C
   DO 140 J=1,LONE
   DO 140 K=1,LTWO
   CRAY(J,K)=ARRAY(LL)
   LL=LL+1
   140 CONTINUE

C
   DO 150 J=1,28
      JJ=INT((J-2)/EXPFAC)+1
      DO 150 K=1,28
         JS=JJ+JNEAK-1
         KS=KK+KNEAK-1
         EZ1=CRAY(JJ+1,KK)+((Z(K)-UZ(KS-1))/(UZ(KS)-UZ(KS-1)))
         * (CRAY(JJ+1,KK+1)-CRAY(JJ+1,KK))


```c
EZ2 = CRAY(JJ, KK) + ((Z(K) - UZ(KS - 1))/(UZ(KS) - UZ(KS - 1)))
    + *(CRAY(JJ, KK + 1) - CRAY(JJ, KK))
EZ(29, J, K) = EZ2 + ((YO(J) - UYO(JS))/(UYO(JS + 1) - UYO(JS)))
    + *(EZ1 - EZ2)
```

150 CONTINUE

```c
DO 160 I = 1, LTWO
    DO 160 J = 1, LONE
    CRAY(I, J) = ARRAY(LL)
    LL = LL + 1
160 CONTINUE
```

```c
DO 170 I = 1, 28
    II = INT((I + 1)/EXPFA C) + 1
    DO 170 J = 1, 28
        JJ = INT((J - 2)/EXPFA C) + 1
        IS = I + INEAR - 1
        JS = J + JNEAR - 1
        EX1 = CRAY(II, JJ + 1) + ((X(I) - UX(IS - 1))/(UX(IS) - UX(IS - 1)))
            + *(CRAY(II + 1, JJ) - CRAY(II, JJ + 1))
        EX2 = CRAY(II, JJ) + ((X(I) - UX(IS - 1))/(UX(IS) - UX(IS - 1)))
            + *(CRAY(II + 1, JJ) - CRAY(II, JJ))
        EX(I, J, 29) = EX2 + ((YO(J) - UYO(JS))/(UYO(JS + 1) - UYO(JS)))
            + *(EX1 - EX2)
170 CONTINUE
```

```c
DO 180 I = 1, LTWO
    DO 180 J = 1, LONE
    CRAY(I, J) = ARRAY(II)
    LL = LL + 1
180 CONTINUE
```

```c
DO 190 I = 1, 28
    II = INT((I + 1)/EXPFA C) + 1
    DO 190 J = 1, 28
        JS = II + INEAR - 1
        JS = J + JNEAR - 1
        EX1 = CRAY(II, JJ + 1) + ((X(I) - UX(IS - 1))/(UX(IS) - UX(IS - 1)))
            + *(CRAY(II + 1, JJ + 1) - CRAY(II, JJ + 1))
        EX2 = CRAY(II, JJ) + ((X(I) - UX(IS - 1))/(UX(IS) - UX(IS - 1)))
            + *(CRAY(II + 1, JJ) - CRAY(II, JJ))
        EX(I, J, 29) = EX2 + ((YO(J) - UYO(JS))/(UYO(JS + 1) - UYO(JS)))
            + *(EX1 - EX2)
190 CONTINUE
```

```c
DO 200 I = 1, LONE
    DO 200 J = 1, LTWO
    CRAY(I, J) = ARRAY(II)
    LL = LL + 1
200 CONTINUE
```

```c
DO 210 I = 1, 28
    II = INT((I - 2)/EXPFA C) + 1
    DO 210 J = 1, 28
```

110
JJ=INT((J+1)/EXPFAC)+1
IS=11+INEAR-1
JS=JJ+JNEAR-1
EY1=CRAY(IJ,JJ+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS))
1*(CRAY(IJ+1,JS+1)-CRAY(IJ,JS+1))
EY2=CRAY(IJ,JS)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS))
1*(CRAY(IJ+1,JS)-CRAY(IJ,JS))
EY(I,J,1)=EY2+(Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1*(EY1-EY2)
CONTINUE
DO 220 I=1,LONE
DO 220 J=1,LTWO
CRAY(I,J)=ARRAY(LL)
220 LL=LL+1
CONTINUE
DO 230 I=1,28
II=INT((I-2)/EXPFAC)+1
DO 230 J=1,28
JS=II+JNEAR-1
JJ=INT((J+1)/EXPFAC)+1
EY1=CRAY(I,JJ+1)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS))
1*(CRAY(I+1,JJ+1)-CRAY(IJ+1))
EY2=CRAY(I,JJ)+((XO(I)-UXO(IS))/(UXO(IS+1)-UXO(IS))
1*(CRAY(I+1,JJ)-CRAY(I,JJ))
EY(I,J,29)=EY2+(Y(J)-UY(JS-1))/(UY(JS)-UY(JS-1)))
1*(EY1-EY2)
CONTINUE
DO 240 I=1,LTWO
DO 240 K=1,LONE
CRAY(I,K)=ARRAY(LL)
240 LL=LL+1
CONTINUE
DO 250 I=1,28
II=INT((I+1)/EXPFAC)+1
DO 250 K=1,28
KS=II+KNEAR-1
EX1=CRAY(I,KK+1)+((X(I)-UX(IK))/(UX(IK)-UX(IK-1))
1*(CRAY(I+1,KK+1)-CRAY(IK+1))
EX2=CRAY(I,KK)+((X(I)-UX(IK))/(UX(IK)-UX(IK-1))
1*(CRAY(I+1,KK)-CRAY(I,KK))
EX(I,1,K)=EX2+(Z0(K)-UZ0(KS))/(UZ0(KS+1)-UZ0(KS))
1*(EX1-EX2)
CONTINUE
DO 260 I=1,LTWO
DO 260 K=1,LONE
CRAY(I,K)=ARRAY(LL)
260 LL=LL+1
DO 270 I = 1, 28
   II = INT((I + 1) / EXPFAC) + 1
   DO 270 K = 1, 28
      KK = INT((K - 2) / EXPFAC) + 1
      IS = II + INEAR - 1
      KS = KK + KNEAR - 1
      EX1 = CRAY(II, KK + 1) + ((X(I) - UX(IS - 1)) / (UX(IS) - UX(IS - 1)))
      EX2 = CRAY(II, KK) + ((X(I) - UX(IS - 1)) / (UX(IS) - UX(IS - 1)))
      EX(I, 29, K) = EX2 + ((ZO(K) - UZ(KS)) / (UZ(KS + 1) - UZ(KS)))
      EZ2 = CRAY(II, KK) + ((XO(I) - UXO(IS)) / (UXO(IS + 1) - UXO(IS)))
      EZ(I, 29, K) = EZ2 + ((Z(K) - UZ(KS - 1)) / (UZ(KS) - UZ(KS - 1)))
   CONTINUE
270 CONTINUE

DO 280 I = 1, LONE
   DO 280 K = 1, LTWO
      CRAY(I, K) = ARRAY(LL)
      LL = LL + 1
280 CONTINUE

DO 300 I = 1, 28
   II = INT((I - 1) / EXPFAC) + 1
   DO 300 K = 1, 28
      KK = INT((K + 1) / EXPFAC) + 1
      IS = II + INEAR - 1
      KS = KK + KNEAR - 1
      EZ1 = CRAY(II, KK + 1) + ((XO(I) - UXO(IS)) / (UXO(IS + 1) - UXO(IS)))
      EZ2 = CRAY(II, KK) + ((XO(I) - UXO(IS)) / (UXO(IS + 1) - UXO(IS)))
      EZ(I, 1, K) = EZ2 + ((Z(K) - UZ(KS - 1)) / (UZ(KS) - UZ(KS - 1)))
      EZ(I, 29, K) = EZ2 + ((Z(K) - UZ(KS - 1)) / (UZ(KS) - UZ(KS - 1)))
   CONTINUE
300 CONTINUE

DO 310 I = 1, 28
   II = INT((I - 2) / EXPFAC) + 1
   DO 310 K = 1, 28
      KK = INT((K + 1) / EXPFAC) + 1
      IS = II + INEAR - 1
      KS = KK + KNEAR - 1
      EZ1 = CRAY(II, KK + 1) + ((XO(I) - UXO(IS)) / (UXO(IS + 1) - UXO(IS)))
      EZ2 = CRAY(II, KK) + ((XO(I) - UXO(IS)) / (UXO(IS + 1) - UXO(IS)))
      EZ(I, 29, K) = EZ2 + ((Z(K) - UZ(KS - 1)) / (UZ(KS) - UZ(KS - 1)))
   CONTINUE
310 CONTINUE

112
RETURN
END
3.5.10 Subroutine DATASAV

Subroutine DATASAV is called by the main program DRIVER for each loop (M=1,2) selected (by MM setting) as frequently as output values of the induced surface charges (\( \hat{n} \cdot \hat{E} \)) and currents (\( \hat{n} \times \hat{H} \)) at the selected test points are required (determined by IP). A flow chart for the M=1 loop is given to Figure 15.

On the first call in a particular loop to Subroutine DATASAV parameters that describe the test points are set. These parameters are defined in Table 6. For this and all subsequent calls within the loop, DATASA performs the operations described in the following paragraphs for each of the three field components from which the charge and the two current components are calculated. These operations allow one to define an arbitrary test point location and orientation where current and charge responses can be found from interpolated and extrapolated field responses at the appropriate surrounding I,J,K locations of the calculated fields.

Table 6. Variables Input in Subroutine DATASAV

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPTS</td>
<td>Number of test points</td>
</tr>
<tr>
<td>LOC(NPT)</td>
<td>Test point number for the ( M^{th} ) test point</td>
</tr>
<tr>
<td>NPLANE(NPT)</td>
<td>Scattering Plane indicates the orientation of the plane on which the ( M^{th} ) test point is located</td>
</tr>
<tr>
<td>XOBS(NPT)</td>
<td>( x ) coordinate of the ( M^{th} ) test point</td>
</tr>
<tr>
<td>YOBS(NPT)</td>
<td>( y ) coordinate of the ( M^{th} ) test point</td>
</tr>
<tr>
<td>ZOBS(NPT)</td>
<td>( z ) coordinate of the ( M^{th} ) test point</td>
</tr>
<tr>
<td>THETA(NPT)</td>
<td>Measurement orientation in degrees from one of the principle axes of the problem space</td>
</tr>
</tbody>
</table>

1 - Normal is in the \(+x\) direction  
2 - Normal is in the \(-x\) direction  
3 - Normal is in the \(+y\) direction  
4 - Normal is in the \(-y\) direction  
5 - Normal is in the \(+z\) direction  
6 - Normal is in the \(-z\) direction
Figure 14. Flow Chart for Subroutine DATASAV
Using the test point location and the scattering plane orientation, eight field component locations are selected. These field component locations are distributed about the test point as illustrated below in Figure 16.

![Diagram of field component locations](image)

Figure 15. Distribution of Field Component Locations About Test Point

The selection process compared all the field component locations against the test point location in one of the three dimensions using a criterion of the form

\[
\text{if } X_{\text{OBS}}(\text{NPT}) \leq X(I+1), \text{ then } I_{\text{max}} = (I+1) \text{ and } I_{\text{min}} = (I) \\
\text{(here the x coordinate has been used for illustrative purposes only), and in the remaining dimensions using a criterion of the form}
\]

\[
\text{if } Y_{\text{OBS}}(\text{NPT}) \leq Y(J+1), \text{ the } J_{\text{min}} = (J) \text{ and } J_{\text{max}} = (J+1).
\]

By letting the coordinates by \(X(I)\) or \(X_o(I)\), \(Y(J)\) or \(Y_o(J)\), and \(Z(K)\) or \(Z_o(K)\) in the above criteria, depending on the field component selected, the test point is surrounded by the eight field components.

The eight field component values associated with the eight field component locations are given by:
where \( \xi \) is any one of the six field components. From these field components the value of \( \xi(X_{\text{OBS}(NPT)}, Y_{\text{OBS}(NPT)}, Z_{\text{OBS}(NPT)}) \) can be found by a combination of interpolation and extrapolation. The exact order in which the components are combined depends on the test point location and NPLANE. An example using the field component arrangement shown in Figure 17 is given to illustrate the process.

First an interpolation is performed in the \( z \) direction to yield

\[
\xi_1 = \xi(I_{\text{min}}, J_{\text{min}}, K_{\text{min}}) + \left( \xi(I_{\text{min}}, J_{\text{min}}, K_{\text{max}}) - \xi(I_{\text{min}}, J_{\text{min}}, K_{\text{min}}) \right) \cdot \frac{\Delta z}{\Delta Z}
\]

\[
\xi_2 = \xi(I_{\text{max}}, J_{\text{min}}, K_{\text{min}}) + \left( \xi(I_{\text{max}}, J_{\text{min}}, K_{\text{max}}) - \xi(I_{\text{max}}, J_{\text{min}}, K_{\text{min}}) \right) \cdot \frac{\Delta z}{\Delta Z}
\]

\[
\xi_3 = \xi(I_{\text{min}}, J_{\text{max}}, K_{\text{min}}) + \left( \xi(I_{\text{min}}, J_{\text{max}}, K_{\text{max}}) - \xi(I_{\text{min}}, J_{\text{max}}, K_{\text{min}}) \right) \cdot \frac{\Delta z}{\Delta Z}
\]

\[
\xi_4 = \xi(I_{\text{max}}, J_{\text{max}}, K_{\text{min}}) + \left( \xi(I_{\text{max}}, J_{\text{max}}, K_{\text{max}}) - \xi(I_{\text{max}}, J_{\text{max}}, K_{\text{min}}) \right) \cdot \frac{\Delta z}{\Delta Z}
\]
These interpolated values are distributed about the test point as shown in Figure 18. They are then interpolated in the x direction to yield

$$\xi_{12} = \xi_1 + (\xi_2 - \xi_1) \frac{Ax}{Ax}$$

$$\xi_{34} = \xi_3 + (\xi_4 - \xi_3) \frac{Ax}{Ax}$$

Finally, the values $\xi_{12}$ and $\xi_{34}$, as shown in Figure 19, are extrapolated to the test point location to give $\xi(X_{OBS}(NPT), Y_{OBS}(NPT), Z_{PBS}(NPT))$, using the expression

$$\xi(X_{OBS}(NPT), Y_{OBS}(NPT), Z_{PBS}(NPT)) = \xi_{12} - (\xi_{34} - \xi_{12}) \frac{Ax}{Ay}$$.
To each value of the scattered field component at the test location, $\xi(X_{OBS}(NPT), Y_{OBS}(NPT), Z_{OBS}(NPT))$, there is added the corresponding incident field component at the test point location. This is accomplished by adding the appropriate function value of $E_{INCXP}$, $E_{INCYP}$, $E_{INCP}$, $H_{INCXP}$ or $H_{INCP}$. The resulting total field components correspond to the induced surface charges and currents.
Two current components are calculated on any surface of the scatterer. These components can be combined to yield other orthogonal current component pairs corresponding to those measuring in an experiment where the current sensors are not aligned in the direction of the x, y or z axis. This situation is commonly the case for measurements made on aircraft wings where the current sensors are aligned with the axis of the wing, which generally is canted with respect to one of the principal axes of the problem space.

The angle THETA(NPT) is the sensor rotation at test point with respect to one of the principal axes. It is shown graphically in Figure 20. Simply stated, θ is always defined as falling in the first quadrant of the plane in which the scattering surface falls. As a result of this definition, the two rotated, orthogonal, current components, called H1 and H2, are always given by:

\[
H_1 = A \cos \theta + B \sin \theta \\
H_2 = -A \sin \theta + B \cos \theta
\]

where

\[
A = H_z, B = H_y \text{ for } NPLANE = 1 \text{ or } 2 \\
A = H_z, B = H_x \text{ for } NPLANE = 3 \text{ or } 4 \\
A = H_x, B = H_y \text{ for } NPLANE = 5 \text{ or } 6
\]

Those variables used locally in Subroutine DATASAV are defined in Table 7.

---
front side
---
back side
---

Figure 19. Graphical Description of THETA
(Angle defining sensor rotation).
Table 7. Local Variables - Subroutine DATASAV

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A principle H field component</td>
</tr>
<tr>
<td>B</td>
<td>A principle H field component</td>
</tr>
<tr>
<td>COSAN</td>
<td>Cosine of THETA</td>
</tr>
<tr>
<td>EX1,EX2</td>
<td>Interpolated $E_x$ field components</td>
</tr>
<tr>
<td>EX3,EX4</td>
<td>Interpolated $E_y$ field components</td>
</tr>
<tr>
<td>EX12,EX34</td>
<td>Interpolated $E_z$ field components</td>
</tr>
<tr>
<td>EY1,EY2</td>
<td>Interpolated $H_x$ field components</td>
</tr>
<tr>
<td>EY3,EY4</td>
<td>Interpolated $H_y$ field components</td>
</tr>
<tr>
<td>EY12,EY34</td>
<td>Interpolated $H_z$ field components</td>
</tr>
<tr>
<td>EZ1,EZ2</td>
<td>Interpolated $E_x$ field components</td>
</tr>
<tr>
<td>EZ3,EZ4</td>
<td>Interpolated $E_y$ field components</td>
</tr>
<tr>
<td>EZ12,EZ34</td>
<td>Interpolated $E_z$ field components</td>
</tr>
<tr>
<td>HX1,HX2</td>
<td>Interpolated $H_x$ field components</td>
</tr>
<tr>
<td>HX3,HX4</td>
<td>Interpolated $H_y$ field components</td>
</tr>
<tr>
<td>HX12,HX34</td>
<td>Interpolated $H_z$ field components</td>
</tr>
<tr>
<td>HY1,HY2,</td>
<td>Interpolated $H_x$ field components</td>
</tr>
<tr>
<td>HY3,HY4</td>
<td>Interpolated $H_y$ field components</td>
</tr>
<tr>
<td>HY12,HY34</td>
<td>Interpolated $H_z$ field components</td>
</tr>
<tr>
<td>HZ1,HZ2</td>
<td>Interpolated $E_x$ field components</td>
</tr>
<tr>
<td>HZ3,HZ4</td>
<td>Interpolated $E_y$ field components</td>
</tr>
<tr>
<td>L</td>
<td>Interpolated $E_z$ field components</td>
</tr>
<tr>
<td>LOC</td>
<td>Interpolated $H_x$ field components</td>
</tr>
<tr>
<td>NPLANE</td>
<td>Interpolated $H_y$ field components</td>
</tr>
<tr>
<td>NPTS</td>
<td>Interpolated $H_z$ field components</td>
</tr>
<tr>
<td>IM,IX</td>
<td>Counter on the number of time steps output</td>
</tr>
<tr>
<td>JM,JX</td>
<td>Test point number (array dimensioned on the number of test points)</td>
</tr>
<tr>
<td>KM,KX</td>
<td>Scattering Plane (array dimensioned on the number of test points)</td>
</tr>
<tr>
<td>IMIN,IMAX</td>
<td>Number of test points</td>
</tr>
<tr>
<td>JMIN,JMAX</td>
<td>Values of $I_{min}, I_{max}, J_{min}, J_{max}, K_{min}, K_{max}$</td>
</tr>
<tr>
<td>KMIN,KMAX</td>
<td>Same as above (arrays dimensioned on the number of test points)</td>
</tr>
<tr>
<td>SINAN</td>
<td>Sine of THETA</td>
</tr>
<tr>
<td>TERM1,TERM2</td>
<td>Interpolation and extrapolation terms</td>
</tr>
<tr>
<td>TERM3,TERM4</td>
<td>Interpolation and extrapolation terms</td>
</tr>
<tr>
<td>TERM5,TERM6</td>
<td>Interpolation and extrapolation terms</td>
</tr>
<tr>
<td>TERM7,TERM8</td>
<td>Interpolation and extrapolation terms</td>
</tr>
<tr>
<td>TERM9</td>
<td>Interpolation and extrapolation terms</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>THETA</td>
<td>Angle (degrees) of current sensor rotation (array dimensional on the number of test points)</td>
</tr>
<tr>
<td>XBS, YBS, ZBS</td>
<td>Coordinates of test point location</td>
</tr>
<tr>
<td>XOBS, YOBS, ZOBS</td>
<td>Same as above (arrays dimensioned on the number of test points)</td>
</tr>
</tbody>
</table>
SUBROUTINE DATASAV

COMMON/EFIELD / EX(29,29,29), EY(29,29,29), EZ(29,29,29)
COMMON/HFIELD / HX(29,29,29), HY(29,29,29), HZ(29,29,29)
COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,N,M,MQ,DT,XMU,EPS0,EPS,NPTLIM,
  1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDL
COMMON/GRID/X(28),Y(28),Z(28),XO(29),YO(29),ZO(29),
  1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
  2 DX1(29),DX1(29),DX1(29),DX1(29),DX1(29),DX1(29),
COMMON/OUT/ESTORE(250,24),HSTOR1(2500,24),HSTOR2(250,24),
  1 TSTORE(250)
COMMON/CURIMUL/CUMM(24),LOC(24)

DIMENSION XOBS(30),YOBS(30),ZOBS(30)
DIMENSION SX0(28),SY0(28),SZ0(28)
DIMENSION IMIN(30),IMAX(30),JMIN(30),JMAX(30),KMIN(30),KMAX(30)
DIMENSION THETA(30),NPLANE(30)
DIMENSION SX(28),SY(28),SZ(28),SX(28),SY(28),SZ(28)
DIMENSION SEX(28,28,28),SEY(28,28,28),SEZ(28,28,28)

PIF=3.1415926536/180.0
L=N/IP
TSTORE(L)=T

IF(L.NE.1) GO TO 1120
IF(N.NE.1) GO TO 200
LOC(1)=1
LOC(2)=2
LOC(3)=3
LOC(4)=4
LOC(5)=5
LOC(6)=6
NPLANE(1)=3
NPLANE(2)=4
NPLANE(3)=1
NPLANE(4)=1
NPLANE(5)=1
NPLANE(6)=1
XOBS(1)=-1.8
YOB(1)=0.0
ZOB(1)=0.0
XOBS(2)=-4.8
YOB(2)=-1.92
ZOB(2)=0.0
XOBS(3)=-1.95
YOB(3)=-1.68
ZOB(3)=-0.42
XOBS(4)=-1.95
YOB(4)=-1.62
ZOB(4)=-0.48
XOBS(5)=-1.95
YOB(5)=-1.68
ZOB(5)=-0.54
XOBS(6)=-1.95
YOB(6)=-1.74
ZOBS(6) = 0.48
THETA(1) = 0.0
THETA(2) = 0.0
THETA(3) = 90.0
THETA(4) = 0.0
THETA(5) = 90.0
THETA(6) = 0.0
NPTS = 6
GO TO 499

C
200 CONTINUE
IF(N.NE.2) GO TO 300
LOC(1) = 7
LOC(2) = 8
LOC(3) = 9
LOC(4) = 10
LOC(5) = 11
LOC(6) = 12
NPLANE(1) = 3
NPLANE(2) = 4
NPLANE(3) = 1
NPLANE(4) = 1
NPLANE(5) = 1
NPLANE(6) = 1
XOBS(1) = -1.8
YOBs(1) = 0.0
ZOBS(1) = 0.0
XOBS(2) = -4.8
YOBs(2) = -1.92
ZOBS(2) = 0.0
XOBS(3) = -1.95
YOBs(3) = -1.95
ZOBS(3) = 0.42
XOBS(4) = -1.95
YOBs(4) = -1.62
ZOBS(4) = 0.48
XOBS(5) = -1.95
YOBs(5) = -1.68
ZOBS(5) = 0.54
XOBS(6) = -1.95
YOBs(6) = -1.74
ZOBS(6) = 0.48
THETA(1) = 0.0
THETA(2) = 0.0
THETA(3) = 90.0
THETA(4) = 0.0
THETA(5) = 90.0
THETA(6) = 0.0
NPTS = 6
GO TO 499

C
300 CONTINUE
C
C DETERMINE CELLS FOR EXTRAPOLATION
C

124
DO 1110 NPT=1,NPTS

C

IF(NPLANE(NPT).GE.3) GO TO 1025
DO 1000 I=1,NX1
IF(XOBS(NPT).LT.X(I+1)) GO TO 1020
1000 CONTINUE

1020 CONTINUE
IF(NPLANE(NPT).EQ.1) IMIN(NPT) = I+1
IF(NPLANE(NPT).EQ.2) IMIN(NPT) = I-1
IMAX(NPT) = IMIN(NPT) + 1
GO TO 1110

1025 CONTINUE
IF(NPLANE(NPT).GT.4) GO TO 1070
DO 1040 J=1,NY1
IF(YOBS(NPT).LT.Y(J+1)) GO TO 1060
1025 CONTINUE

1040 CONTINUE

1060 CONTINUE
IF(NPLANE(NPT).EQ.3) JMIN(NPT) = J+1
IF(NPLANE(NPT).EQ.4) JMIN(NPT) = J-1
JMAX(NPT) = JMIN(NPT) + 1
GO TO 1110

1070 CONTINUE
DO 1080 K=1,NZ1
IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1100
1080 CONTINUE
1100 IF(NPLANE(NPT).EQ.5) KMIN(NPT) = K+1
IF(NPLANE(NPT).EQ.6) KMIN(NPT) = K-1
KMAX(NPT) = KMIN(NPT) + 1

1110 CONTINUE
PRINT 407,(XOBS(NPT),NPT=1,NPTS)
PRINT 408,(YOBS(NPT),NPT=1,NPTS)
PRINT 409,(ZOBS(NPT),NPT=1,NPTS)

C

NPLANE = 1 IS THE YZ PLANE, NORMAL AFT (TOWARD INCREASING I)
C
NPLANE = 2 IS THE YZ PLANE, NORMAL FORWARD (TOWARD DECREASING I)
C
NPLANE = 3 IS THE XZ PLANE, NORMAL UP (TOWARD INCREASING J)
C
NPLANE = 4 IS THE XZ PLANE, NORMAL DOWN (TOWARD DECREASING J)
C
NPLANE = 5 IS THE XY PLANE, NORMAL RIGHT (TOWARD INCREASING K)
C
NPLANE = 6 IS THE XY PLANE, NORMAL NEGATIVE (TOWARD DECREASING K)
C

1120 CONTINUE
DO 1400 NPT=1,NPTS
IM = IMIN(NPT)
IX = IMAX(NPT)
JM = JMIN(NPT)
JX = JMAX(NPT)
KM = KMIN(NPT)
KX = KMAX(NPT)
XBS = XOBS(NPT)
YBS = YOBS(NPT)
ZBS = ZOBS(NPT)
ANGLE = THETA(NPT)*PIF

IF(NPLANE(NPT).GT.4) GO TO 1270
IF(NPLANE(NPT).GT.2) GO TO 1155
NPLANE = 1 OR 2 -- YZ PLANES

COMPUTE TERMS FOR CHARGE

DETERMINE JM, JX, KM, KX FOR EX

DO 1125 J=1,NYI
IF(YOBS(NPT).LT.YO(J+1)) GO TO 1128
CONTINUE
JF1 = J
JX = J+1
DO 1130 K=1,NZI
IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1131
CONTINUE
KM = K
KX = K+1
IF(L.EQ.100) PRINT 410, IM, IX, JM, JX, KM, KX
TERM1 = (ZBS - ZO(KM))/DZO(KM)
TERM2 = (YBS - YO(JM))/DYO(JM)
TERM3 = (XBS - XO(IM))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(IN))
TERM5 = 0.5*(DYO(JX)+DYO(JH))
TERM6 = 0.5*(DZO(KX)+DZO(K1))

EX1 = EX(IM, JM, KM)+(EX(IM, JM, KM)-EX(IM, JM, KM))*TERM1
EX2 = EX(IM, JX, KM)+(EX(IM, JX, KM)-EX(IM, JX, KM))*TERM1
EX3 = EX(IX, JM, KM)+(EX(IX, JM, KM)-EX(IX, JM, KM))*TERM1
EX4 = EX(IX, JX, KM)+(EX(IX, JX, KM)-EX(IX, JX, KM))*TERM1
EX12 = EX1 + (EX2-EX1)*TERM2
EX34 = EX3 + (EX4-EX3)*TERM2

DEBUG PRINT

IF(L.NE.100) GO TO 1134
PRINT 514, (((EX(I,J,K), I=IM, IX), J=JM, JX), K=KM, KX)
PRINT 515, EX1, EX2, EX3, EX4, EX12, EX34

COMPUTE TERMS FOR CURRENT DENSITY

DETERMINE KM, KX FOR HY, JM, JX SAME AS FOR EX

DO 1135 K=1,NZI
IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1136

1136 KM = K
KX = K+1
IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
TERM1 = (ZBS-ZO(KM))/DZO(KH)
TERM2 = (YBS-YO(JM))/DYO(JM)
C TERM2 IS SCALE FACTOR FOR H1Y(I,JX,K)
TERM3 = (XBS-XO(IN))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(IM))
TERM5 = 0.5*(DYO(JX)+DYO(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KH))
TERM7 = (ZBS-Z(KH))/TERM6
C TERM7 IS SCALE FACTOR FOR HZ(I,JX,K)

C C C DEBUG PRINT
C IF(L.NE.100) GO TO 1137
PRINT 511,((HY(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
PRINT 512, H1Y1, HY2, HY3, HY4, HY12, HY34
1137 CONTINUE
C C DETERMINE JM, JX, KM, KX FOR HZ
C DO 1138 J=1,NY1
IF(YOBS(NPT).LT.Y(J+1)) GO TO 1140
1138 CONTINUE
1140 JM = J
JX = J+1
DO 1144 K=1,NZ1
IF(ZOBS(NPT).LT.Z0(K+1)) GO TO 1146
1144 CONTINUE
1146 KM = K
KX = K+1
IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
TERM1 = (ZBS-20(KI))/Z20(KM)
C TERM1 IS SCALE FACTOR FOR HZ(I,J,KX)
TERM2 = (YBS-YO(JM))/DYO(JM)
TERM3 = (XBS-XO(IN))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(IM))
TERM5 = 0.5*(DYO(JX)+DYO(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KH))
C TERM8 = (YBS-Y(JM))/TERM5
C TERM8 IS SCALE FACTOR FOR HZ(I,JX,K)
HZ1=HZ(I,JX,KM)+(HZ(IM,JM,KX)-HZ(IM,JM,KM))*TERM1
HZ2=HZ(IM,JX,KM)+(HZ(IM,JX,KX)-HZ(IM,JX,KM))*TERM1
HZ3 = HZ(IX, JM, KM) + (HZ(IX, JM, KX) - HZ(IX, JM, KM)) * TERM1
HZ4 = HZ(IX, JX, KM) + (HZ(IX, JX, KX) - HZ(IX, JX, KM)) * TERM1
HZ12 = HZ1 + (HZ2 - HZ1) * TERM8
HZ34 = HZ3 + (HZ4 - HZ3) * TERM8

DEBUG PRINT

IF(L.NE.100) GO TO 1148
PRINT 504, ((HZ(I, J, K), I=IM, IX), J=JM, JX), K=KM, KX)
PRINT 505, HZ1, HZ2, HZ3, HZ4, HZ12, HZ34
1148 CONTINUE

IF(NPLANE(NPT).EQ.2) GO TO 1150

TERM9 = (X(IM) - XBS) / TERM4
TERM9 IS -SCALE FACTOR FOR (EX, HY, HZ)(IX, J, K)

ESTORE(L, NPT) = EX12 - (EX34 - EX12) * TERM9 + EINCXP(XBS, YBS, ZBS)
A = HZ12 - (HZ34 - HZ12) * TERM9 + HINCZP(XBS, YBS, ZBS)
B = HY12 - (HY34 - HY12) * TERM9 + HINCYP(XBS, YBS, ZBS)
GO TO 1375

1150 CONTINUE

TERM9 = (XBS - X(IX)) / TERM4
TERM9 IS -SCALE FACTOR FOR (EX, HY, HZ)(IM, J, K)

ESTORE(L, NPT) = EX34 + (EX34 - EX12) * TERM9 + EINCXP(XBS, YBS, ZBS)
A = HZ34 + (HZ34 - HZ12) * TERM9 + HINCZP(XBS, YBS, ZBS)
B = HY34 + (HY34 - HY12) * TERM9 + HINCYP(XBS, YBS, ZBS)
GO TO 1375

1155 CONTINUE

NPLANE = 3 OR 4 -- XZ PLANES

COMPUTE TERMS FOR CHARGE

DETERMINE IM, IX, KM, KX FOR EY

1155 CONTINUE
DO 1160 I=1,NX1
IF(XOBS(NPT).LT.XO(I+1)) GO TO 1165
1160 CONTINUE
1165 IM = I
IX = I+1
DO 1170 K=1,NZ1
IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1175
1170 CONTINUE
1175 KM = K
KX = K+1
IF(L.EQ.100) PRINT 410, IM, IX, JM, JX, KM, KX
TERM1 = (ZBS - ZO(KM))/DZO(KM)
C TERM1 IS SCALE FACTOR FOR EY(I,J,KX)
TERM2 = (YBS - YO(JM))/DYO(JM)
TERM3 = (XBS - XO(IM))/DXO(IM)
C TERM3 IS SCALE FACTOR FOR EY(I,J,K)
TERM4 = 0.5*(DXO(IX)+DXO(IM))
TERM5 = 0.5*(DYO(JX)+DYO(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KM))
EY1 = EY(IM,JM,KM)+(EY(IM,JX,KX)-EY(IM,JX,KM))*TERM1
EY2 = EY(IX,JM,KM)+(EY(IX,JX,KX)-EY(IX,JX,KM))*TERM1
EY3 = EY(IM,JM,KM)+(EY(IM,JX,KX)-EY(IM,JX,KM))*TERM1
EY4 = EY(IX,JX,KM)+(EY(IX,JX,KX)-EY(IX,JX,KM))*TERM1

EY12 = EY1 + (EY2-EY1)*TERM3
EY34 = EY3 + (EY4-EY3)*TERM3

DEBUG PRINT
IF(L.NE.100) GO TO 1202
PRINT 500,(((EY(I,J,K),I=II,IX),J=JM,IX),K=KM,IX)
PRINT 501, EY1,EY2,EY3,EY4,EY12,EY34
1202 CONTINUE

C COMPUTE TERMS FOR CURRENT DENSITY
C DETERMINE KM,IX FOR HX IM,IX SAME AS FOR EY
DO 1210 K=1,NZ1
IF(ZOBS(NPT).LT.Z(K+1)) GO TO 1220
1210 CONTINUE
1220 KM = K
IX = K+1
IF(L.EQ.100) PRINT 410,IX,IX,IX,IX,IX,IX
TERM1 = (ZBS - ZO(KM))/DZO(KM)
TERM2 = (YBS - YO(JM))/DYO(JM)
TERM3 = (XBS - XO(IM))/DXO(IM)
C TERM3 IS SCALE FACTOR FOR HX(IX,J,K)
TERM4 = 0.5*(DXO(IX)+DXO(IM))
TERM5 = 0.5*(DYO(JX)+DYO(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KM))
TERM7 = (ZBS-ZO(KM))/TERM6
C TERM7 IS SCALE FACTOR FOR HX(I,J,KX)
HX1 = HX(IM,JI,KM)+(HX(IM,JI,KX)-HX(IM,JI,KM))*TERM7
HX2 = HX(IX,JI,KM)+(HX(IX,JI,KX)-HX(IX,JI,KM))*TERM7
HX3 = HX(IM,JX,KM)+(HX(IM,JX,KX)-HX(IM,JX,KM))*TERM7
HX4 = HX(IX,JX,KM)+(HX(IX,JX,KX)-HX(IX,JX,KM))*TERM7
C HX12 = HX1 + (HX2-HX1)*TERM3
HX34 = HX3 + (HX4-HX3)*TERM3
C DEBUG PRINT
IF(L.NE.100) GO TO 1222
PRINT 502,(((HX(I,J,K),I=IM,IX),J=JM,IX),K=KM,IX)
PRINT 503, HX1, HX2, HX3, HX4, HX12, HX34
1222 CONTINUE

C
C DETERMINE IM, IX, KM, KX FOR HZ
C
DO 1225 I=1, NX1
IF(XOBS(NPT).LT.X(I+1)) GO TO 1230
1225 CONTINUE
1230 IM = I
IX = I+1
DO 1235 K=1, NZ1
IF(ZOBS(NPT).LT.ZO(K+1)) GO TO 1240
1235 CONTINUE
1240 KM = K
KX = K+1
IF(L.EQ.100) PRINT 410, IM, IX, JM, JX, KM, KX
TERM1 = (ZBS-ZO(KM))/DZO(KM)
C TERM1 IS SCALE FACTOR FOR HZ(I,J,KX)
TERM2 = (YBS-YO(JM))/DYO(JM)
TERM3 = (XBS-XO(IN))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(IN))
TERM5 = 0.5*(DYO(JX)+DYO(JH))
TERM6 = 0.5*(DZO(KX)+DZO(KM))
HZ1 = HZ(IM, JM, KM) + (HZ(IM, JM, KX) - HZ(IM, JM, KM)) * TERM1
HZ2 = HZ(IX, JH, KM) + (HZ(IX, JM, KX) - HZ(IX, JM, KM)) * TERM1
HZ3 = HZ(IM, JX, KM) + (HZ(IM, JX, KX) - HZ(IM, JX, KM)) * TERM1
HZ4 = HZ(IX, JX, KM) + (HZ(IX, JX, KX) - HZ(IX, JX, KM)) * TERM1
C
TERM8 = (XBS-X(IM))/TERM4
C TERM8 IS SCALE FACTOR FOR HZ(IX,J,K)
HZ12 = HZ1 + (HZ2-HZ1)*TERM8
HZ34 = HZ3 + (HZ4-HZ3)*TERM8
C
DEBUG PRINT
C
IF(L.NE.100) GO TO 1242
PRINT 504, (((HZ(I,J,K), I=IM, IX), J=JM, JX), K=KM, KX)
PRINT 505, HZ1, HZ2, HZ3, HZ4, HZ12, HZ34
1242 CONTINUE
C
IF(NPLANE(NPT).EQ.4) GO TO 1250
C
TERM9 = (Y(JH)-YBS)/TERM5
C TERM9 IS -SCALE FACTOR FOR (EY,HZ,HX)(I,JX,K)
C
PRINT 995
995 FORMAT(*CHK1*)
C
PRINT 996
996 FORMAT(*CHK2*)
C
ESTORE(L, NPT) = EY12 - (EY34-EY12)*TERM9 + EINCYP(XBS, YBS, ZBS)
PRINT 997
A = HZ12 - (HZ34-HZ12)*TERM9 + HINCZP(XBS,YBS,ZBS)

B = HX12 - (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)

PRINT 997

998 FORMAT(*CHK4*)
B = HX12 - (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
PRINT 999

999 FORMAT(*CHK5*)

DEBUG PRINT

IF(L.NE.100) GO TO 1375
EINC = EINCYP(XBS,YBS,ZBS)
H1INC = HINCXP(XBS,YBS,ZBS)
H2INC = HINCZP(XBS,YBS,ZBS)
PRINT 513, EINC,H1INC,H2INC
PRINT 506,ESTORE(L,NPT),A,B
GO TO 1375

TERM9 = (YBS-Y(JX))/TERM5
TERM9 IS -SCALE FACTOR FOR (EY,HZ,HX)(I,JM,K)

ESTORE(L,NPT)=EY34 + (EY34-EY12)*TERM9 + EINCYP(XBS,YBS,ZBS)
A = iX34 + (iX34-iX12)*TERM9 + HINCZP(XBS,YBS,ZBS)
B = iX34 + (iX34-iX12)*TERM9 + HINCXP(XBS,YBS,ZBS)

DEBUG PRINT

IF(L.NE.100) GO TO 1375
EINC = EINCYP(XBS,YBS,ZBS)
H1INC = HINCXP(XBS,YBS,ZBS)
H2INC = HINCZP(XBS,YBS,ZBS)
PRINT 513, EINC,H1INC,H2INC
PRINT 506,ESTORE(L,NPT),A,B
GO TO 1375

NPLANE = 5 OR 6 -- XY PLANES

COMPUTE TERMS FOR CHARGE

DETERMINE IM,IX,JM,JX FOR EZ

1270 CONTINUE
DO 1275 I=1,NX1
IF(XOBS(NPT).LT.XO(I+1)) GO TO 1280
1275 CONTINUE
1280 IM = I
IX = I+1
DO 1285 J=1,MY1
IF(YOBS(NPT).LT.YO(J+1)) GO TO 1290
1285 CONTINUE
1290 JM=J
JX = J+1

IF(L.EQ.100) PRINT 410,IM,IX,JH,JX,KM,KX
TERM1 = (ZBS - ZO(KM))/DZ0(KH)
TERM2 = (YBS - YO(JM))/DY0(JM)
C
TERM2 IS SCALE FACTOR FOR EZ(I,JX,K)
TERM3 = (XBS - XO(IM))/DX0(IM)
C
TERM3 IS SCALE FACTOR FOR EZ(I,JX,K)
TERM4 = 0.5*(DX0(IX)+DX0(IM))
TERM5 = 0.5*(DY0(JX)+DY0(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KM))

1300 EZ1=EZ(IH,JH,KH)+(EZ(IX,JN,KH)-EZ(IM,JN,KH))*TERM3
EZ2=EZ(IM,JX,KH)+(EZ(IX,JX,KH)-EZ(IM,JX,KH))*TERM3
EZ3=EZ(IM,JH,KX)+(EZ(IX,JH,KX)-EZ(IM,JH,KX))*TERM3
EZ4=EZ(IM,JX,KX)+(EZ(IX,JX,KX)-EZ(IM,JX,KX))*TERM3
C
EZ12 = EZ1 + (EZ2-EZ1)*TERM2
EZ34 = EZ3 + (EZ4-EZ3)*TERM2
C
DEBUG PRINT
C
IF(L.NE.100) GO TO 1302
PRINT 509,(((EZ(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
PRINT 510, EZ1,EZ2,EZ3,EZ4,EZ12,EZ34
1302 CONTINUE
C
COMPUTE TERMS FOR CURRENT DENSITY
C
C DETERMINE JH,JX FOR HX, IN,IX SAME AS FOR EZ
C
DO 1310 J=1,NY1
IF(YOBS(NPT).LT.Y(J+1)) GO TO 1315
1310 CONTINUE
1315 JM = J
JX = J+1

IF(L.EQ.100) PRINT 410,IM,IX,JH,JX,KM,KX
TERM1 = (ZBS - ZO(KM))/DZ0(KH)
TERM2 = (YBS - YO(JM))/DY0(JM)
TERM3 = (XBS - XO(IM))/DX0(IM)
C
TERM3 IS SCALE FACTOR FOR HX(IX,J,K)
TERM4 = 0.5*(DX0(IX)+DX0(IM))
TERM5 = 0.5*(DY0(JX)+DY0(JM))
TERM6 = 0.5*(DZO(KX)+DZO(KM))

HX1=HX(IH,JM,KH)+(HX(IX,JH,KM)-HX(IM,JH,KH))*TERM3
HX2=HX(IM,JX,KH)+(HX(IX,JX,KM)-HX(IM,JX,KH))*TERM3
HX3=HX(IM,JH,KX)+(HX(IX,JH,KX)-HX(IM,JH,KX))*TERM3
HX4=HX(IM,JX,KX)+(HX(IX,JX,KX)-HX(IM,JX,KX))*TERM3
C
TERM7 = (YBS-Y(JM))/TERM5
C
TERM7 IS SCALE FACTOR FOR HX(I,J,K)
HX12 = HX1 + (HX2-HX1)*TERM7
HX34 = HX3 + (HX4-HX3)*TERM7
C
DEBUG PRINT
IF(L.NE.100) GO TO 1318
PRINT 502,(((HX(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
PRINT 503, HX1,HX2,HX3,HX4,HX12,HX34
1318 CONTINUE

C DETERMINE IM,IX,JM,JX FOR HY
C
DO 1320 I=1,NX1
IF(XOBS(NPT).LT.X(I+1)) GO TO 1325
1320 CONTINUE
1325 IM= I
IX = I+1
DO 1330 J=1,NY1
IF(YOBS(NPT).LT.YO(J+1)) GO TO 1335
1330 CONTINUE
1335 JM = J
JX = J+1
IF(L.EQ.100) PRINT 410,IM,IX,JM,JX,KM,KX
TERM1 = (ZBS - ZO(KM))/DZO(KM)
TERM2 = (YBS - YO(JM))/DYO(JM)
TERM3 = (XBS - XO(IM))/DXO(IM)
TERM4 = 0.5*(DXO(IX)+DXO(I+1))
TERM5 = 0.5*(DYO(JX)+DYO(JN))
TERM6 = 0.5*(DZO(KX)+DZ0(KN))
TERM8 = (XBS-X(IM))/TERM14
TERM9 = (Z(KM)-ZBS)/TERM6
TERM9 IS SCALE FACTOR FOR (EZ,HX,HY)(I,J,K)
HY1=HY(IM,JM,KM)+(HY(IX,JX,KM)-HY(IM,JM,KM))*TERM8
HY2=HY(IM,JX,KM)+(HY(IX,JX,KM)-HY(IM,JX,KM))*TERM8
HY3=HY(IM,JM,KX)+(HY(IX,JM,KX)-HY(IM,JM,KX))*TERM8
HY4=HY(IM,JX,KX)+(HY(IX,JX,KX)-HY(IM,JX,KX))*TERM8
HY12 = HY1 + (HY2-HY1)*TERM2
HY34 = HY3 + (HY4-HY3)*TERM2

C DEBUG PRINT
C
IF(L.NE.100) GO TO 1340
PRINT 511,(((HY(I,J,K),I=IM,IX),J=JM,JX),K=KM,KX)
PRINT 512, HY1,HY2,HY3,HY4,HY12,HY34
1340 CONTINUE
C
IF(NPLANE(NPT).EQ.6) GO TO 1350
TERM9 = (Z(KM)-ZBS)/TERM6
TERM9 IS SCALE FACTOR FOR (EZ,HX,HY)(I,J,KX)
C
ESTORE(L,NPT)=EZ12 - (EZ34-EZ12)*TERM9 + EINCZP(XBS,YBS,ZBS)
A = HX12 - (HX34-HX12)*TERM9 + HINCXP(XBS,YBS,ZBS)
B = HY12 - (HY34-HY12)*TERM9 + HINCYP(XBS,YBS,ZBS)

C DEBUG PRINT
IF (L.NE.100) GO TO 1375
EINC = EINCZP(XBS,YBS,ZBS)
H1INC = HINCP(XBS,YBS,ZBS)
H2INC = HINCP(XBS,YBS,ZBS)
PRINT 513,EINC,H1INC,H2INC
PRINT 506,ESTORE(L,NPT),A,B
GO TO 1375

1350 TERM9 = (ZBS-Z(KX))/TERM6
TERM9 IS SCALE FACTOR FOR (EZ,HX,HY)(I,J,KM)

ESTORE(L,NPT)=EZ34 + (EZ34-EZ12)*TERM9 + EINCZP(XBS,YBS,ZBS)
A = HX34 + (HX34-HX12)*TERM9 + HINCP(XBS,YBS,ZBS)
B = HY34 + (HY34-HY12)*TERM9 + HINCP(XBS,YBS,ZBS)

DEBUG PRINT

IF (L.NE.100) GO TO 1375
EINC = EINCZP(XBS,YBS,ZBS)
H1INC = HINCP(XBS,YBS,ZBS)
H2INC = HINCP(XBS,YBS,ZBS)
PRINT 513,EINC,H1INC,H2INC
PRINT 506,ESTORE(L,NPT),A,B
GO TO 1375

1375 CONTINUE
SINAN = SIN(ANGLE)
COSAN = COS(ANGLE)
HSTOR1(L,NPT) = A*COSAN+B*SINAN
HSTOR2(L,NPT) = B*COSAN-A*SINAN
CUM(NPT) = CUM(NPT)+HSTOK1(L,NPT)

IF (L.EQ.1) GO TO 1390
IF (L.NE.100) GO TO 1400

1390 CONTINUE
PRINT 507, HSTOR1(L,NPT),HSTOR2(L,NPT),ESTORE(L,NPT),ANGLE
PRINT 508,SINAN,COSAN

1400 CONTINUE

FORMAT STATEMENTS

401 FORMAT(15)
402 FORMAT(2I5,4F10.3)
407 FORMAT(7H1,XOBS,/3(10(F8.2,4X),/))
408 FORMAT(7H0,YOBS,/3(10(F8.2,4X),/))
409 FORMAT(7H0,ZOBS,/3(10(F8.2,4X),/))
410 FORMAT(10H0,ILX,8X,7HJM JX,8X,7HXM KX,/3(2I5,5X))
500 FORMAT(6H0,EY ,8E12.4,/)  
501 FORMAT(1H0,4X,3HEY1,9X,3HEY2,9X,3HEY3,9X,3HEY4,8X,4HEY12,8X,
  1 4HEY34,/,6E12.4,/)  
502 FORMAT(6H0,HX ,8E12.4,/)  
503 FORMAT(1H0,4X,3HHX1,9X,3HHX2,9X,3HHX3,9X,3HHX4,8X,4HHX12,8X,
1 4HHX34,/,6E12.4,/
504 FORMAT(6HO HZ ,8E12.4,/
505 FORMAT(1HO,4X,3HHZ1,9X,3HHZ2,9X,3HHZ3,9X,3HHZ4,8X,4HHZ12,8X,
1 4HHZ34,/,6E12.4,/
506 FORMAT(9HO E,H1,H2,5X,3E12.4)
507 FORMAT(16HO H1,H2,E,ANGLE,5X,4E12.4)
508 FORMAT(14HO SIN(ANGLE) =,F12.3,5X,12HCOS(ANGLE) =,F12.3)
509 FORMAT(6HO EZ ,8E12.4,/
510 FORMAT(1HO,4X,3HEZ1,9X,3HEZ2,9X,3HEZ3,9X,3HEZ4,8X,4HEZ12,8X,
1 4HEZ34,/,6E12.4,/
511 FORMAT(6HO HY ,8E12.4,/
512 FORMAT(1HO,4X,3HHY1,9X,3HHY2,9X,3HHY3,9X,3HHY4,8X,4HHY12,8X,
1 4HHY34,/,6E12.4,/
513 FORMAT(1HO,8H EINCP =,E12.4,2X,8HINCP =,E12.4,2X,
1 8HINCP =,E12.4)
514 FORMAT(6HO EX ,8E12.4,/
515 FORMAT(1HO,4X,3HEY1,9X,3HEY2,9X,3HEY3,9X,3HEY4,8X,4HEY12,8X,
1 4HEY34,/,6E12.4,/
C
RETURN
END
3.5.11 Subroutine PRINOUT

PRINOUT prints out the data saved in DATASAV. It also extends the data record for H-field loops that define wire currents. The data is summed over time, which averages out any resonances (these behave as $A_n e^{-a_n t} \sin \omega t$ and when summed over several cycles yield approximately zero.) This leaves the sum of the decaying exponential response $A_o e^{-\alpha_0 t} dt$, which is

$$S = \int_0^T A_o e^{-\alpha_0 t} dt.$$ 

The decaying exponential response is the portion of the total response that tracks the driving function which decays as $e^{-\alpha_0 t}$. Thus, $\alpha_0$ is known from the excitation source and $A_o$ can be found from $S$, the numerically derived sum, and the expression $S = \int_0^T A_o e^{-\alpha_0 t} dt$. With $A_o$ and $\alpha_0$ known, the data record is then extended in time. This allows transforms to be made for data records as long as 50 μsec or a lower frequency limit of 10 KHz in the spectrum.
SUBROUTINE PRINOUT

COMMON/EXTRAS/NX,NY,NZ,NX1,NY1,NZ1,M,MQ,DT,XMU,EPS0,EPS,NPTLIM,
1 NN,NPTS,LMAX,SIGMA,C,T,P1,EXPFAC,IP,FX,TY,TZ,AMP,ALPHA,BETA,IDLS
COMMON/GRID/X(28),Y(28),Z(28),X0(29),Y0(29),Z0(29),
1 DX(29),DY(29),DZ(29),DX0(28),DY0(28),DZ0(28),
2 DXI(29),DYI(29),DZI(29),DXOI(28),DYOI(28),DZOI(28)
COMMON/OUT/ESTORE(250,24),HSTORI(2500,24),HSTOR2(250,24),
1 TSTORE(250)
COMMON/CUMUL/CUMH(24),LOC(24)
COMMON/PERM/PP
DIMENSION CSTOKE(2500,8)
REAL PP

PRINT OUTPUT

DO 1500 NPT = 1,NPTS

L=N/IP
PP=IP

PRINT 24,LOC(NPT)
24 FORMAT(8HI LOC = ,I3)
PRINT 27,L
27 FORMAT (80X/36X, I4,40X)
PRINT 23, (TSTORE(LL),LL=1,L,3)
23 FORMAT(1X,1P6E13.6,1X)
PRINT 27,L
PRINT 23, (ESTORE(LL,NPT),LL=1,L,3)
PRINT 27,L
PRINT 23, (HSTOR1(LL,NPT),LL=1,L,3)
PRINT 27,L
PRINT 23, (HSTOR2(LL,NPT),LL=1,L,3)

TT=T

DO 1501 LL=L,LMAX

TT=TT+DT*8.

HSTORI(LL,NPT)=CUMH(NPT)*DT*8.*
1 (1.0/(1.0-EXP(-ALPHA*T))/ALPHA-(1.0-EXP(-BETA*T))/BETA)*
2 EXP(-ALPHA*TT)

1501 CONTINUE

1500 CONTINUE

IF(N.NE.1.OR.NM.NE.1) GO TO 1600

DO 1510 LL=1,LMAX

CSTORE(LL,1)=HSTORI(LL,1)
CSTORE(LL,2)=HSTORI(LL,2)

1510 CONTINUE

PRINT 24,LOC(1)
PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)
PRINT 24,LOC(2)
PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
GO TO 500
C
1600 CONTINUE
C
IF(M.NE.2.OR.NH.NE.12) GO TO 1700
C
DO 1610 LL=1,LMAX
CSTORE(LL,1)=HSTORE(LL,1)
CSTORE(LL,2)=HSTORE(LL,2)
1610 CONTINUE
C
PRINT 24,LOC(1)
PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)
PRINT 24,LOC(2)
PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
GO TO 500
C
1700 CONTINUE
C
IF(M.NE.2) GO TO 1800
C
DO 1710 LL=1,LMAX
CSTORE(LL,1)=HSTORE(LL,1)
CSTORE(LL,2)=HSTORE(LL,2)
1710 CONTINUE
C
PRINT 24,LOC(1)
PRINT 23,(CSTORE(LL,1),LL=1,LMAX,3)
PRINT 24,LOC(2)
PRINT 23,(CSTORE(LL,2),LL=1,LMAX,3)
C
GO TO 500
C
1800 CONTINUE
C
C
500 CONTINUE
C
RETURN
END
/EOR
APPENDIX A

The governing equations for the lossy dielectric finite difference code are in the form

\[
\varepsilon \frac{\partial E_x^S}{\partial t} + \sigma E_x^S = \left( \frac{\partial H_y^S}{\partial y} - \frac{\partial H_z^S}{\partial z} \right) - \sigma E_x^i - (\varepsilon - \varepsilon_0) \frac{\partial E_x^i}{\partial t}
\]

where the x component of the E-field is used, for example. Consider now a plate of fractional thickness T in the x direction, i.e. the plate is \( T \Delta x \) thick, so that T is a fraction of the cell dimension in the desired direction. The effective difference equation for the cell in which the plate is present can be found from the average of the above difference equation for a region in which the plate is present (\( \sigma = 0, \varepsilon = \varepsilon_0 \)) and where there is only free space (\( \sigma = 0, \varepsilon = \varepsilon_0 \)). More precisely

\[
(1-T) \left[ \frac{\partial E_x^S}{\partial t} + \sigma E_x^S = \left( \frac{\partial H_y^S}{\partial y} - \frac{\partial H_z^S}{\partial z} \right) - \sigma E_x^i - (\varepsilon - \varepsilon_0) \frac{\partial E_x^i}{\partial t} \right] + T \left[ \frac{\partial E_x^S}{\partial t} + \sigma E_x^S = \left( \frac{\partial H_y^S}{\partial y} - \frac{\partial H_z^S}{\partial z} \right) - \sigma E_x^i - (\varepsilon - \varepsilon_0) \frac{\partial E_x^i}{\partial t} \right]
\]

In explicit finite difference form this is expressed as

\[
E_x^S(I,J,K)^{n+1/2} = E_x^S(I,J,K)^{n-1/2} e^{-\sigma T \Delta t / \varepsilon_{eff}} \left[ -E_x^i(I,J,K)^{n} + \frac{\varepsilon - \varepsilon_0}{\sigma} E_x^i(I,J,K)^{n-1} \right] + (1-e^{-\sigma T \Delta t / \varepsilon_{eff}}) \chi
\]

\[
+ \left[ -E_x^i(I,J,K)^{n} + \frac{\varepsilon - \varepsilon_0}{\sigma} E_x^i(I,J,K)^{n-1} \right]
\]

\[
+ \frac{H_y^S(I,J,K)^n - H_y^S(I,J-1,K)^n}{\sigma T(Y(J) - Y(J-1))}
\]

\[
+ \frac{H_y^S(I,J,K)^n - H_y^S(I,J,K-1)^n}{\sigma T(Z(K) - Z(K-1))}
\]

where \( \varepsilon_{eff} = [(1-T)\varepsilon_0 + T\varepsilon] \).

Similar expressions hold in the outer directions for the thin plate formalism.
REFERENCES


A new electromagnetic coupling computer code is an integration of several finite-difference codes based on THREDE, an established electromagnetic computer modeling technique. For a given lightning model, the new integrated code will be capable of predicting, with better resolution than was hitherto available, EM fields caused by lightning at several locations outside or inside metal or composite aircraft. Increased resolution is obtained by making two computer runs: total (unexpanded) geometry and expanded geometry. The total response is found and then is expanded, allowing greater resolution in geometry and higher frequency response.