NASA Contractor Report 178280

THE ELECTROMAGNETIC MODELING OF THIN APERTURES USING THE FINITE-DIFFERENCE TIME-DOMAIN TECHNIQUE

Kenneth R. Demarest

THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
Lawrence, Kansas

Grant NAG1-514
June 1987

Date for general release February 28, 1989

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report discusses the implementation of an enhancement of the Finite-difference Time-Domain (FDTD) technique to compute the transient electromagnetic response of complex conducting scatterers containing apertures that are narrow with respect to the wavelength contained within the power spectrum of excitation. An analytical technique is developed which utilizes Babinet's principle to model the aperture, and a computer code, called THNAPP, is described and documented that utilizes this technique.

The computer code THNAPP is an extension of previously developed FDTD codes that were not capable of modeling thin apertures or seams. This code possesses the same basic structure as the code G3DXL which was written for the System Validation Methods Branch of the Information Systems Division at Langley Research Center. THNAPP has been successfully executed on the CDC 203 computer system at Langley Research Center and is currently operational.

I. INTRODUCTION

A vital part of the lightning research conducted by NASA Langley Research Center [1-3] has been the development of computer codes capable of calculating the electromagnetic responses of aircraft subjected to lightning events. The importance of these codes stems from the fact that the direct measurement of aircraft responses is very difficult and expensive. The capability to predict interior and exterior responses numerically provides the opportunity to probe the nature of the responses, the causes of various aspects of the responses, and, most importantly, to act as a test bed for various aircraft changes designed to reduce the susceptibility of aircraft to electrical upset.

One of the electromagnetic code types investigated by NASA during the past few years utilizes the Finite-Difference Time-Domain (FDTD) technique [4]. This technique possesses a number of very attractive attributes with respect to the NASA lightning effort. First, the solutions are calculated directly in the time domain. Since a lightning event is a transient phenomenon, it is desirable to calculate responses directly in the time domain, rather than transforming frequency domain results back into the time domain. Second, the FDTD approach is very flexible in the types of scatterers (i.e., aircraft) and electromagnetic sources (i.e., lightning events) that can be handled. In particular, these codes can handle any shape scatterer that can be roughly modeled as a collection of rectangular cells or plates (either metal or dielectric). Also, both direct and indirect lightning strikes can be modeled.

In spite of their many advantages, a distinct shortcoming of many FDTD codes is their inability to model geometries that contain rapid geometrical variations (e.g., changes in radii of curvature or material composition). This deficiency stems from the fact the FDTD technique solves scattering problems by calculating the fields only at a finite number of points that lie on a numerical grid. Thus, surface details that are smaller than this grid cannot be accounted for in the electromagnetic calculations without additional analysis. Fortunately, there are a number of situations in which FDTD codes have been augmented so that they can handle such geometries. Kunz and Simpson [5] have developed a technique whereby a complicated scatterer can be first analyzed using a course grid, and then a portion of it can be re-analyzed using a finer grid that can resolve the fine detail. As another example,

Holland and Simpson [6] have developed a technique for modeling wires and struts which are much smaller than the grid.

An important example of a type of fine detail often encountered in aircraft that cannot at present be handled easily by FDTD codes is the long thin aperture. This type of sub-structure is encountered in many areas of an airplane fuselage. Examples are door seals and the seams between either metal or composite panels. Only when FDTD codes are capable of handling such geometries will one be able to accurately predict the interior electromagnetic responses of aircraft when subjected to threats such as lightning or nuclear electromagnetic pulse (NEMP).

This report presents an FDTD computer code, called THNAPP, that is capable of modeling conducting scatterers that contain thin, cavity backed apertures. This code represents an update of an FDTD code called G3DXL that was written by Kunz Associates for NASA Langley Research Center [7] for use in predicting the responses of aircraft struck by lightning. THNAPP retains the general form of G3DXL, but adds two very important capabilities not present in G3DXL. This first is of course the ability to model aircraft that contain thin apertures. The second is the ability to model complex configurations of thin wires involving wires of different radii and wire junctions with other wires or surfaces.

The approach used in THNAPP to model thin apertures utilizes Babinet's principle to obtain a prediction of the fields in the aperture by analyzing a complementary geometry in which the thin aperture becomes a thin magnetic conductor [8]. These aperture fields are then used to drive the fields in the cavity behind the aperture. THNAPP can model a great variety of scattering geometries. In particular, more than one aperture can be modeled simultaneously if they occur in the same plane. Also, these apertures may intersect and be of difference sizes and widths.

The structure of this report is as follows. Chapter II introduces the theoretical aspects of the application of Babinet's principle to thin aperture coupling. Chapter III presents the computer code, THNAPP, that implements this theory. This chapter also contains a description of each of the subroutines present in THNAPP that were either not present in or have been radically changed in the parent code G3DXL. Chapter IV presents typical results from THNAPP for geometries containing complex wire and geometries and cavity backed apertures. Finally, Chapter IV presents concluding comments.

II. THIN APERTURE FORMULATION

Figure 1a shows a perfectly conducting scatterer containing a cavity behind a thin, conducting, flat surface with a narrow aperture.

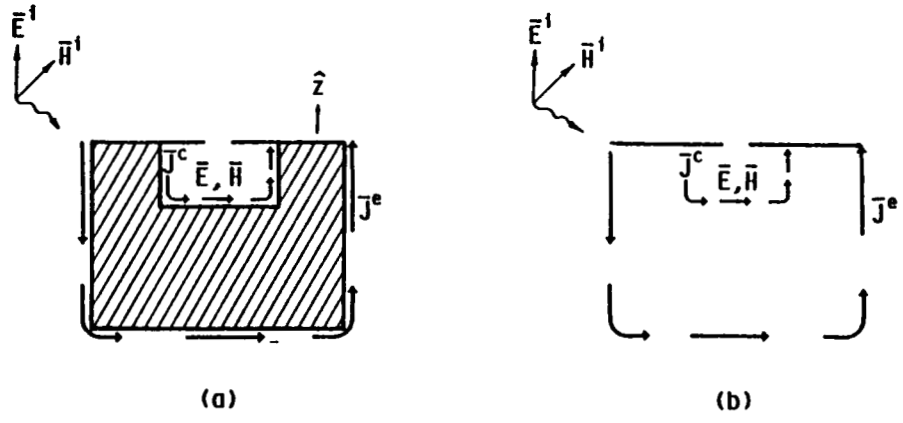

Fig. 1. (a) A conducting scatterer containing a narrow aperture in a thin conducting plate, backed by a cavity. $\bar{J}^c$ and $\bar{J}^e$ are physical currents. (b) Equivalent geometry. All surfaces except those on the aperture plane have been replaced by impressed equivalent currents $\bar{J}^c$ and $\bar{J}^e$.

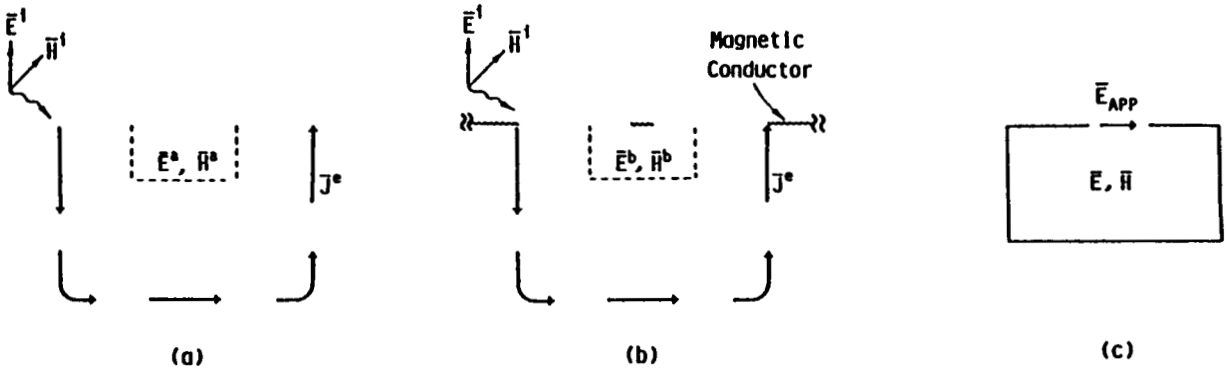

Fig. 2. Sub-geometries analyzed when using Babinet's principle. (a) Aperture plane removed and surface currents $\bar{J}^e$ impressed. (b) Aperture plane replaced with its complement and surface currents $\bar{J}^e$ impressed. (c) Aperture electric field impressed upon the cavity only.

In this development, it will be assumed that the scatterer is illuminated by a known, externally applied field $(\bar{E}^i, \bar{H}^i)$ which may be either localized in

nature (as in the case of a voltage or current source) or global (such as a plane wave). Thus, the total fields are given by:

$$\begin{aligned} \bar{E}(z) &= \bar{E}^i(z) + \bar{E}^s(z) \\ \bar{H}(z) &= \bar{H}^i(z) + \bar{H}^s(z) \quad , \end{aligned} \tag{1}$$

where the superscripts "i" and "s" denote incident and scattered fields, respectively. It will be assumed that the aperture lies in the $z = 0$ plane and the cavity lies below this plane (i.e., $z < 0$). In this equation, as with those to follow, only the z dependence of the fields will be enumerated, where z is the normal to the aperture plane. The currents $\bar{J}^e$ and $\bar{J}^c$ are the physical surface currents that are flowing on the external and cavity surfaces, respectively. Note that, by definition, neither $\bar{J}^e$ or $\bar{J}^c$ include those currents located on the aperture plane.

An equivalent situation to that of Figure 1a, both inside and outside the scatterer, is shown in Figure 1b. Here, the entire obstacle, except for the conducting plane that contains the aperture, has been removed while retaining the surface currents $\bar{J}^e$ and $\bar{J}^c$. This equivalence follows from the Schelkunoff equivalence principle [9]. Note that the conducting sheet has imaged out the impressed equivalent currents above and below it. The sources in this figure are thus the incident fields and the free space fields of the impressed currents $\bar{J}^e$ and $\bar{J}^c$. Also, note that the only material body in Figure 1b is the (perfectly) conducting plane which contains the aperture.

The purpose of transforming the geometry of Figure 1a into that of Figure 1b is to express the problem in a form where Babinet's principle [10] can be utilized to break the geometry down to where the FDTD technique can be utilized in its analysis. Even though the aperture is assumed to be much more narrow than can be handled directly by the FDTD technique, the complement of the aperture plane, a magnetic strip in free space, can be handled by using the electrical dual of an existing technique for directly modeling electrically conducting wires and struts in FDTD codes [6].

In order to invoke Babinet's principle in the solution of the fields in Figure 1b using FDTD technique, the question of the a priori knowledge of the currents $\bar{J}^e$ and $\bar{J}^c$ must be addressed. First, although the presence of the aperture will have some effect on all the currents flowing on and within the

scatterer, it will in general have little effect on $\bar{J}^e$ since (1) its capacitance is high due to its narrow width, and (2) these currents are not located on the aperture plane itself and thus are somewhat distant from the aperture. Thus, an excellent approximation of $\bar{J}^e$ can be obtained by short-circuiting the aperture and using any technique well suited to the external scatterer geometry -- possibly the FDTD technique or the method-of-moments [11].

On the other hand, the currents $\bar{J}^c$ cannot be known a priori since our assumption has been that the aperture is too small to be handled directly by techniques such as FDTD or the method of moments. Fortunately, this lack of a priori knowledge (and thus the tacit initial assumption that they are zero) will not have a significant effect on the early time responses of Figure 1b for two reasons. First, there is a natural delay between the starting times of $\bar{J}^e$ and $\bar{J}^c$ due to the propagation delay of the fields into the cavity. But more importantly, their effect will in most cases be negligible to observers near the aperture for as long as the external response is much stronger than the internal response. This, of course, will be the case for the entire temporal response when the aperture is small and the cavity Q is low, but will also occur for the early portions of the response even when the cavity Q is high. Even for points of observation deeper into the cavity, there will be at least some clear window in the transient responses that are not appreciably affected by the lack of a priori knowledge of $\bar{J}^c$.

From Babinet's principle, the fields of Figure 1b (minus the effects of $\bar{J}^c$) can be determined from those of Figures 2a and 2b. In Figure 2a, the conducting plate and aperture of Figure 1b have been replaced by free space.

Thus, the fields of this configuration are:

$$\begin{aligned} \bar{E}^a &= \bar{E}^i + \bar{E}^J \\ \bar{H}^a &= \bar{H}^i + \bar{H}^J \quad , \end{aligned} \tag{2}$$

where ($\bar{E}^J$, $\bar{H}^J$) are the fields radiated by the surface currents $\bar{J}^e$ when radiating in free space. In Figure 2b, the plane containing the aperture of Figure 1b has been replaced with its complement: free space where the electrical conductor had been and (perfect) magnetic conductor where free space had been. The fields of this configuration are:

$$\bar{E}^b = \bar{E}^i + \bar{E}^J + \bar{E}^{dif}$$

$$\bar{H}^b = \bar{H}^i + \bar{H}^J + \bar{H}^{dif} \quad , \tag{3}$$

where ($\bar{E}^{dif}$, $\bar{H}^{dif}$) are the diffracted fields due to the magnetic currents on the magnetic conductors, radiating into free space.

From Babinet's principle, the fields of Figure 1b for $z < 0$ can now be written in terms of those of Figures 2a and 2b as [10]

$$\bar{E} = \bar{E}^a_u(z) - \bar{E}^b_u(z) + \bar{E}^a_\ell(z) \pm E^b_\ell(-z)$$

$$\bar{H} = \bar{H}^a_u(z) - \bar{H}^b_u(z) + \bar{H}^a_\ell(z) \pm \bar{H}^b_\ell(-z) \quad , \tag{4}$$

where the fields due to sources above and below the image plane (z=o) have, of necessity, been identified separately; the subscripts "u" and "ℓ" denote fields due to impressed fields and surface currents (i.e., $\bar{J}^e$) which are located above and below the image plane, respectively. The correct sign to be used for the $\pm$ sign depends upon the component of the field in question -- the + sign for normal electric and tangential magnetic and the - sign for the tangential electric and normal magnetic fields. A similar expression can be written for $z > 0$.

The above expression is useful for the calculation of the cavity fields in two ways. First, it can be evaluated within the cavity to determine the initial transient responses directly from the FDTD solutions of the geometries of Figures 2a and 2b. Obviously, the length of time for which they are valid depends upon the proximity of the point of observation to the aperture and the cavity walls. A particularly attractive attribute of this expression is that since the incident fields due to impressed sources above the image plane appear in both ($\bar{E}^a_u$, $\bar{H}^a_{u)}$ and ($\bar{E}^b_u$, $\bar{H}^b_u$), they exactly cancel. Thus, if the incident field is a global field incident from above this plane, this expression for the fields is not subject to the noise problem often encountered in FDTD codes where the calculated scattered fields are perturbed by numerical noise so that, when added to the unperturbed incident fields, they do not correctly cancel to obtain the nearly zero total cavity fields [6]. In the above formulation, all of the terms are subject to the same sources of numerical noise and thus will cancel much more effectively.

Equation (4) can also be evaluated within the aperture. In this way, the early to mid-time aperture fields can be found from the FDTD analysis of the geometries of Figures 2a and 2b and then used as the impressed fields of a final FDTD analysis of the cavity alone. Substituting equations (2) and (3) into equation (4) and evaluating in the limit as $z \to 0^+$, we obtain:

$$\bar{E}(0^+) = \bar{E}_{App} = \bar{E}^J_\ell(0) - \left[\bar{E}^J_\ell(0) + \bar{E}^{dif}_\ell(0^+)\right] + \bar{E}^J_\mu(0) - \left[\bar{E}^J_\mu(0) - \bar{E}^{dif}_\mu(0^-)\right], \quad (5)$$

where the sign corresponding to a tangential electric field has been used. Since the incident fields are continuous across the aperture, they completely cancel and are not present in this expression. Also, although the fields generated by the surface currents alone are also continuous across the aperture and do indeed cancel in this expression, they have not been removed because they are embedded in the FDTD solutions of both Figures 2a and 2b respectively (as indicated by the brackets), and must be subtracted numerically.

The fields of equation (5) can be used as the impressed aperture fields in an FDTD analysis of the cavity alone to yield solutions that include the effects of the internal cavity currents. This situation is depicted in Figure 2c. These calculations will remain valid much longer than those obtained from the direct application of equation (4). Only after the aperture fields become dominated by the response of the cavity will these calculations fail. The point at which this happens will depend upon the cavity Q.

Since it is the aperture field that is forced during this calculation, there is no need to advance the fields outside the cavity as would be the case if it were an equivalent current, rather than the field itself, that drove the solution. Also, the spatial grid used in even this final FDTD calculation need not be small enough to resolve the aperture. This follows from the fact that the major inaccuracy encountered when the aperture is smaller than the grid results from the difficulty in determining the average fields in cells containing the aperture. However, since the early time response magnitudes inside the cavity are already known from the application of equation (4), the final FDTD results of the cavity modeled alone can be scaled to agree with these known amplitudes. No scaling is needed when the grid size used to obtain the aperture fields is the same as that used in the cavity analysis. This will be demonstrated in the next section.

III. CODE DESCRIPTION

A. Introduction

The purpose of this chapter is to describe a code, called THNAPP, that implements the thin aperture coupling concepts derived in the previous chapter. This code is basically a derivative of the code G3DXL written by Karl Kunz of Kunz Associates for the System Validation Methods Branch of the Information Systems Division at NASA Langley Research Center [7]. G3DXL is in turn a derivative of the well known code THREDE [12], and was developed to satisfy NASA's need for an FDTD code capable of calculating the responses of aircraft to a direct or indirect lightning event.

Given the fact that the thin aperture code to be described herein is an outgrowth of these two well documented codes, it is not the intent of this chapter to give a complete description of this code. Rather, it is assumed that the reader is familiar with G3DXL and thus this chapter will act as an appendix to the documentation of this code since the basic framework and philosophy of this code has been retained in THNAPP.

In the sections to follow, a basic overview of THNAPP will be given that will convey the basic philosophy of the code. This will be followed by a series of sections that describe each subroutine that has been added to G3DXL.

B. Overview of Operation

As with G3DXL, THNAPP is designed to allow a particular geometry to be analyzed using serial calculation runs (or loops, using Kunz's nomenclature). However, whereas G3DXL is designed for two loops (analysis of the entire structure using course grid, and re-analysis of a small portion using an expanded grid [7]), THNAPP retains these two loops and adds three more in order to accommodate the use of the generalized Babinet's principle to solve problems involving thin apertures.

Figure 3 depicts the geometry to be analyzed and the sequence of sub-geometries that are analyzed by THNAPP in order to determine the fields in an internal cavity backed by a thin aperture. Figure 3a depicts the entire geometry to be analyzed, which contains an aperture backed cavity. Figure 3b shows the geometry used to calculate the external surface currents, $\bar{J}^e$. Here, the aperture has been short circuited and only the external characteristics of the geometry need be modeled. The surface currents

evaluated during this calculation are stored and used as sources for the sub-geometries of figures 3c and 3d.

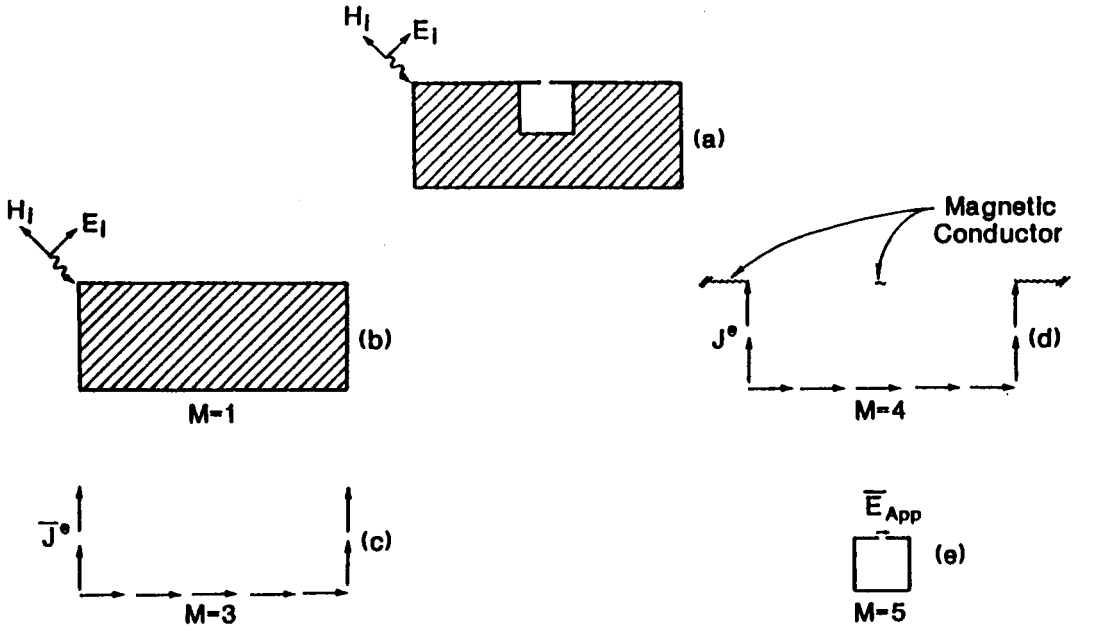

Fig. 3. Sequence of geometries modeled by THNAPP to calculate thin seam coupling using Babinet's principle. (a) Original geometry to be analyzed, which contains a cavity backed thin aperture, (b) M=1 geometry, aperture shorted (c) M=3 geometry, impressed surface currents radiate in free space, (d) M=4 geometry, impressed surface currents radiate in the presence of the aperture plane complement, and (e) M=5 geometry, aperture field forced in the aperture and cavity fields calculated.

In figure 3c, the surface currents are allowed to radiate in free space. Thus, the fields of this figure are the "incident" fields discussed in the previous chapter. In figure 3d, the surface currents radiate in the presence of the electrical complement of the aperture plane. The aperture fields are obtained by taking the difference between the fields at the aperture position of figures 3c and 3d and then using them as the sources for the geometry of figure 3e which need not contain the external portions of the scatterer.

The following is a description of each of the analysis loops contained in the main program, called DRIVER, in THNAPP. A printout of DRIVER is presented at the end of this overview section.

M=1: As with G3DXL, this loop uses a course grid to analyze the entire geometry under consideration except for the aperture (Figure 3b). If a thin seam is present in the geometry, the variable IBAB will be set equal to 1 and will automatically store the entire time histories of the external surface currents on the obstacle for later use in the loops M=3 and 4.

M=2: This loop re-analyzes a small portion of the obstacle (defined in M=1) to allow for more spatial resolution in the solutions. This loop in THNAPP is identical to that in G3DXL. This loop is not necessary for the analysis of structures with thin apertures.

M=3: The purpose of this loop is to calculate the fields in free space of the forced surface currents ($\bar{J}^e$) calculated during M=1 (Figure 3c). In general, there will be no material bodies specified during this loop.

M=4: During this loop, the external currents ($\bar{J}^e$) are again forced and the entire obstacle is removed, except for the aperture plane (Figure 3d). This entire plane is replaced by its complement - magnetic conductor where free space had been and free space where electric conductor had been. A magnetic wire is used to model the magnetic strip, which is the compliment of the aperture.

M=5: The aperture fields as calculated by the M=3 and 4 loops are forced in the aperture, thus allowing the cavity fields to be calculated as shown in Figure 3e. The calculations during this loop can utilize an unexpanded (M5EXP = 0) or an expanded (M5EXP = 1) grid, depending upon the dimensions of the cavity backing the aperture.

The version of THNAPP described here assumes that the plane containing the aperture is located such that the remainder of the obstacle is entirely on or below this plane. It is also assumed that the source of any impressed field in this problem is also on this same side of the aperture plane. Extensions of this code to include cases more general than this are straightforward and can be accomplished by tayloring the code after equation (5).

THNAPP is capable of analyzing structures with more than one aperture present. If all of the apertures exist on the same plane, they can all be modeled simultaneously on the same sequence of loops. If, however, they exist on separate planes, they must be analyzed one plane at a time. Thus, the fields inside a cavity that contains two or more apertures that are on different planes are found by adding the contributions from each aperture plane separately. This could be accomplished by making separate runs of THNAPP for each of the aperture planes and summing the M=5 outputs of these runs.

As with G3DXL, the scattering geometry in THNAPP is specified in subroutine BUILD by filling the array NOPE with a cell by cell description of the geometry. The rules by which the NOPE array is filled in THNAPP are exactly the same as for G3DXL. At present, however, the code is restricted to the case of perfectly conducting M=1 (but not M=5) geometries made up of solid cubes and thin plates when the thin aperture capability is used (i.e. IBAB = 1). The plates used to build up the scatterer may be of any size and orientation and may intersect solid cubes, but the intersection of plates of different orientations in the external geometry is at present not permitted in the code (due to limitations in subroutine FCBLD). The generalization of this code to model dielectrics and conducting plates that intersect at right angles could be incorporated later.

During the M=1 loop, the $\bar{H}$ fields adjacent to the external conducting surfaces are monitored and the surface currents are computed at each time step. These current values are calculated in subroutine STOCUR and stored in the array FCUR. The positions of these surface currents are determined in subroutine FCBLD, which is called by subroutine BUILD and basically uses the information in the NOPE array to determine the surface current locations.

After the M=1 loop, in which the entire scatterer (minus the aperture) is analyzed and the surface currents on all external surfaces (except for those on the plane containing the aperture) have been calculated and stored, the M=3 and 4 loops can commence. For the M=3 loop, the NOPE array is filled entirely with zeros since the purpose of this calculation is to find the fields radiated by the surface source currents in free space. During this loop, the external surface currents that were calculated during the M=1 loop are forced at these locations by adding the current density term to the curl $\bar{H}$ equation (in subroutine EADV).

For the M=4 loop, the only non-zero entries in the NOPE array occur on the plane containing the aperture. Here, magnetic conductor is placed where free space is present in the actual geometry, and free space is placed where electrical conductor had been present. Since the aperture is much smaller than a cell dimension, a magnetic wire of appropriate radius is placed at the position of the aperture. The placement of this magnetic wire is accomplished simply by declaring the endpoint positions and radius of the wire in subroutine BUILD. As in M=4, the external currents calculated during M=1 are again forced.

The fields at each test point location (see [7] for a description of the test point declarations in subroutine DATASAV) are stored in the output arrays as each loop cycles through all of its time increments. At the end of each loop, these arrays are printed out by subroutine PRINOUT. In order to calculate the desired fields either inside or outside the cavity, these fields must be point for point added or subtracted according to the rules given in equation (5). Care must be taken to note which field component is being calculated so that the correct signs are used in this formula. Also, note that it may be necessary to define the test points at complementary positions (i.e. image positions with respect to the aperture plane) during the M=3 and 4 runs to account for the E (-z) terms in the formula. The use of these rules is more clearly seen in the chapter on numerical results.

In the following section those subroutines contained in THNAPP that do not occur in G3DXL or have been significantly modified are presented. For a description of the remaining subroutines, see [7].

```
      PROGRAM DRIVER(INPUT,TAPE3=INPUT,OUTPUT=OUTKF,TAPE4=INFILE,
     +    TAPE5=PZFILE,TAPE6=PLOTDAT)
C
      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),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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      COMMON/OUTLIST/DELX,DELY,DELZ,XPANX,XPANY,XPANZ,
     1   IUP,JUP,KUP,IDOWN,JDOWN,KDOWN,INEAR,JNEAR,KNEAR,
     2   IFAR,JFAR,KFAR,XOBS(6),YOBS(6),ZOBS(6),TEST,
     3   NPLANE(6)
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      COMMON/PERM/MM
      CHARACTER*30 TITLE
      INTEGER TEST
C
C     LOOP OVER THE DESIRED PERMUTATION OF THE FOUR CONFIGURATIONS
C     (DIFFUSION,M=1/EXPANDED DIFFUSION,M=2/"INCIDENT FIELDS",M=3/
C     "MAGNETIC" FIELDS,M=4/
C
C     SET MM VALUE TO DETERMINE WHICH PERMUTATION
C     (DIFFUSION ONLY,MM=1/DIFFUSION,EXPANDED DIFFUSION,MM=12/
C     SOURCE CURRENTS,"INCIDENT FIELDS",MM=13/
C     SOURCE CURRENTS,"INCIDENT FIELDS,"MAGNETIC" FIELDS,MM=134/
C
C
C  IDLS = 0 FOR RADIATED FIELDS ; IDLS = 1 FOR DIRECT STRIKES
C
C
C  TO USE THE 20-POINT AVERAGING TECHNIQUE ,
C  SET IAVG=1. TO TURN OFF THE AVERAGING, SET IAVG=0.
C
C
C  READ CONSTANT PARAMETERS FROM INPUT FILE
C
      READ(4,1003) TITLE
      READ(4,1004) MM,IDLS,TEST,IAVG
      READ(4,1005) AMP,ALPHA,BETA,SIGMA,EPS
      READ(4,1004) N130,N230,N330,N430,N530
C
      IF(MM.EQ.1.OR.MM.EQ.12.OR.MM.EQ.13.OR.MM.EQ.14
     < .OR.MM.EQ.134.OR.MM.EQ.1345)GO TO 100
      PRINT 10
   10 FORMAT(*MM INPUT ERROR*)
C
  100 CONTINUE
C
C     DIFFUSION(M=1) LOOP
C
```

```
      M=1
C
C     GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=1
C
      CALL SETUP
      CALL BUILD
C
C
      TSTART=0.0
      PRINT 111
  111 FORMAT(*BUILD DONE*)
C
      T=DT/2.+TSTART
      N=0
      DO 130 N=1,N130
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE HFIELD
C
      CALL HADV
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE EFIELD
C
      CALL EADV
      IF(MM.EQ.12) CALL SAVESB
C
C     STORE FIELDS
C
      IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
  130 CONTINUE
      N=N130
C
      PRINT 150,T,N
  150 FORMAT(*1EXIT TIME(M=1)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT(IAVG,TITLE)
      IF(MM.EQ.12) GO TO 160
      IF(MM.EQ.13 .OR. MM.EQ.134 .OR. MM.EQ.1345)GO TO 300
      IF(MM.EQ.14)GO TO 400
      GO TO 600

  160 CONTINUE
C
  200 CONTINUE
C
C     EXPANDED DIFFUSION(M=2) LOOP
```

```
C
C
      M=2
C
C     GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=2
C
      CALL SETUP
      CALL BUILD
C
C     SET TIME LIMITS AND EVERY IP DATA POINT SAVED
C
      TSTART=0.0
      IP=IP*EXPFAC
C
      T=DT/2.+TSTART
      N=0
      DO 230 N=1,N230
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE HFIELD
C
      CALL HADV
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE EFIELD
C
      CALL EADV
C
C     STORE FIELDS
C
      IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
  230 CONTINUE
      N=N230
C
      PRINT 250,T,N
  250 FORMAT(*1EXIT TIME(M=2)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT(IAVG,TITLE)
      GO TO 600
C
  260 CONTINUE
C
  300 CONTINUE
C
C     "INCIDENT FIELDS"(M=3) LOOP
C
      M=3
C
```

```
C     GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=3
C
      CALL SETUP
      CALL BUILD
C
C     SET TIME LIMITS AND EVERY IP DATA POINT SAVED
C
      TSTART=0.0
C
      T=DT/2.+TSTART
      N=0
      DO 330 N=1,N330
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE HFIELD
C
      CALL HADV
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE EFIELD
C
      CALL EADV
C
C     STORE FIELDS
C
      IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
  330 CONTINUE
      N=N330
C
      PRINT 350,T,N
  350 FORMAT(*1EXIT TIME(M=3)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT(IAVG,TITLE)
      IF(MM.NE.134 .AND. MM.NE.1345)GO TO 600
C
  360 CONTINUE
C
  400 CONTINUE
C
C     "MAGNETIC FIELDS"(M=4) LOOP
C
      M=4
C
C     GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=4
C
      CALL SETUP
      CALL BUILD
```

```
C
C     SET IN TIME LIMITS AND EVERY IP DATA POINT SAVED
C
      TSTART=0.0
C
      T=DT/2.+TSTART
      N=0
      DO 430 N=1,N430
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE HFIELD
C
      CALL HADV
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE EFIELD
C
      CALL EADV
      IF(M5EXP.EQ.1) CALL SAVESB
C
C     STORE FIELDS
C
      IF(MOD(N,IP).EQ.0) CALL DATASAV
C
C
  430 CONTINUE
      N=N430
C
      PRINT 450,T,N
  450 FORMAT(*1EXIT TIME(M=4)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT(IAVG,TITLE)
  500 CONTINUE
      IF(MM.NE.1345)GO TO 600
C     IN THIS LOOP(M=5), THE APERTURE FIELDS ARE USED AS SOURCE
C     TO CALCULATE THE FIELDS INSIDE THE CAVITY.
C
      M=5
C
C     GENERATE PROBLEM SPACE AND INTERACTION OBJECT FOR M=5
C
      CALL SETUP
      CALL BUILD
C
C     SET IN TIME LIMITS AND SAVE DATA EVERY IP
C
      TSTART=0.0
      IF(M5EXP.EQ.1) THEN
          IP=IP*EXPFAC
      END IF
```

```
      T=DT/2.+TSTART
      N=0
      DO 530 N=1,N530
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE HFIELD
C
      CALL HADV
C
C     ADVANCE TIME
C
      T=T+DT/2.
C
C     ADVANCE EFIELD
C
      CALL EADV
C
C     STORE FIELDS
C
      IF(MOD(N,IP) .EQ. 0)CALL DATASAV
  530 CONTINUE
      N=N530
      PRINT 550,T,N
  550 FORMAT(*1EXIT TIME(M=5)=*E12.3*,AFTER CYCLE*I4)
      CALL PRINOUT(IAVG,TITLE)
  600 CONTINUE
      STOP
C
C  FORMAT SPECIFICATIONS
C
 1003 FORMAT(A30)
 1004 FORMAT(6I10)
 1005 FORMAT(6E20.10)
      END
```

C. Subroutines

This section contains individual descriptions of each of the subroutines that are contained in THNAPP that are not contained in G3DXL. Subroutines WIREBLD and WIREADV pertain to thin wires, subroutines SLOTBLD and SLOTADV pertain to magnetic wires (the complement of apertures), and subroutines FCBLD and STOCUR pertain to the use of Babinet's principle for thin apertures. Other subroutines or functions that have been either modified or added for THNAPP included BUILD, EADV, ITRAN, IRTRAN, PULSE, SAVESB, ABSORB and HBC.

Subroutine BUILD

Subroutine BUILD has two functions. The first is to allow the user to build the M=1 through M=5 geometries by filling the NOPE array with values appropriate for their geometries. The second is to allow the user to specify at the beginning of the M=1 loop the position of the aperture and aperture plane so that the code can decide which surface currents should be stored during the M=1 loop and later forced during the M=3 and 4 loops.

The rules for filling the array NOPE are basically the same as they are in G3DXL. However, when the thin aperture formalism is to be used, external geometries (M=1,3,4) must be restricted to perfectly conducting structures composed of solid cubes and plates (although during M=5, aperture backed cavities may contain dielectrics). Although these plates may be of any size and may intersect solid sections, they cannot intersect other plates at right angles.

The first two variables to be defined in BUILD are IBAB and M5EXP. Both of these are specified before the M=1 loop for easy visual identification, although they could also be defined as the first statements in the M=1 loop. The parameter IBAB indicates whether or not thin seams will be present in the geometry to be analyzed. Thus, IBAB controls whether or not the external currents will be sensed during M=1 and then forced during M=3 and 4. The acceptable values of IBAB are 1 and 0, which indicate that the thin seam formalism will or will not be used during the loop, respectively. In order to be consistent, IBAB must equal 1 for MM=1345. The values of M5EXP indicate whether the M=5 (internal cavity) loop is to use an unexpanded (M5EXP=0) or expanded (M5EXP=1) grid.

When building the M=1 version of a scatterer geometry containing a thin aperture, the entire scatter is built using the normal rules of the NOPE array [7], remembering that the geometry specified during M=1 must consist of solid conductor blocks with no hollow sections, with the possibility of thin plates intersecting the solid sections (but not each other). Also, the aperture must be short circuited (i.e. covered with metal) during this loop. The orientation of the plane that contains the aperture is indicated by defining the parameter LPDIR to have a value of 1, 2, or 3 - corresponding to yz, xz, and xy orientations, respectively. The elevation of this plane is indicated by the value of the parameter LP. For example, parameter values of LPDIR=2

and LP=15 indicate that the aperture is in the xz plane, and this plane appears at the bottom of the J=15 cells.

If any wires are present in the M=1 geometry, only their endpoints and radii must be specified. The rules for defining wires are the same for any value of M and is independent of whether or not the thin aperture formalism is in use. Wires can be of any length or radius, and they may intersect other wires, conducting plates, or conducting solids. Although wires may intersect other wires that are either parallel or perpendicular, they may not intersect plates or solid sections directly on an edge. Wires must be defined along the cell lattice lines (i.e. through $\overline{E}$ field points) and their endpoints must coincide with cell lattice points (i.e. cell apex points).

The number of wires to be defined is set by the parameter NWRS. For three wires, the value of NWRS would be 3. The endpoints of each wire are indicated by the arrays IA and IB. The radius of each wire is indicated by the value of the array WRAD. For example, if the 2nd wire to be modeled is specified as:

```
IA(2) = ITRAN (13, 8, 17)
IB(2) = ITRAN (13, 17, 17)
WRAD(2) = .001
```

this wire has endpoints at the apexes of cells (13, 8, 17) and (13, 17, 17) and has a radius of .001 meters. Note that although this particular wire has its endpoint at the apex of the J=17 cell, it only runs through the 16th cell. On the other hand, its other endpoint is in the J=8 th cell and it also runs through this cell. ITRAN is a function that assigns a unique number for every triplet of numbers (I, J, K) and allows all of the parameters concerning wire currents to be stored in singular subscripted arrays, rather than triplicate subscripted arrays. This results in a large reduction in computer storage since wires will generally appear in a relatively small number of cells.

For M=3, there will usually be no material bodies present during this loop since it is the purpose of this loop to calculate the fields of the forced surface currents radiating into free space. Thus, all elements of the NOPE array are filled with zero's during this loop and the value of NWRS is maintained at zero. The position of the aperture, however, must be specified in this loop (or before) so that the fields at these positions can be stored during this loop and subtracted from those of the M=4 loop in order to

determine the aperture fields. Since the apertures are assumed to be small, only the fields across their narrow dimension are generally stored (this is done automatically in subroutine EADV). The number of cells that contain the aperture(s) are indicated by the value of NLOC, the component of the electric field in the aperture is indicated by the value of IDIRE (1, 2 or 3 for x, y, or z directed fields, respectively), and the locations of the cells that contain the aperture are indicated by the values of the array LOCE. Note that IDIRE is not an array and thus all apertures must have the same orientation (although this could easily be changed by modifying subroutine EADV). For example, an aperture position indicated by the parameters

NLOC = 4

IDIRE = 1

LOCE(1-4) = ITRAN (10, 12, 11 - 14)

represents an aperture lying along the x axis, and there are 4 cells that contain this aperture at positions (10, 12, 11 - 14). These parameters need not be re-specified for the M=4 or M=5 loops.

For M=4, the only material bodies that should be present are magnetic wires and plates since it is the purpose of this loop to calculate the fields radiated by the forced currents in the presence of the complement of the aperture. Magnetic plates are specified by much the same rules as those for electrically conducting plates, except that the NOPE values are the negatives of their electrically conducting counterparts, and their positions within the cells are somewhat different. Thus, the appropriate NOPE values for magnetic plates are -1, -2, -3, -12, -13, -23, -123. Also, since when modeling thin apertures the magnetic plates will all be in the same plane, only the first three values will be needed (i.e. there will be no seams). Basically, these negative NOPE values instruct subroutine HBC where to zero the total tangential $\bar{H}$ fields. As a result, the rules for the placement of magnetic plates are somewhat different from those of electric plates in that magnetic plates must run along $\bar{H}$ field lines rather than $\bar{E}$ field lines. Thus, magnetic surfaces are defined along the centers of cells, rather than along their edges. In specifying the M=4 geometry, a decision must be made as to whether the magnetic surface is to be placed either one half cell above (i.e. away from the scatterer) or one half cell below the actual aperture location (i.e. inside the scatterer), recognizing that either situation is an approximation

of the actual problem and also that a ½ cell displacement will have little effect on the calculated fields. Usually, the former is chosen so that the surfaces lie just on top of the forced surface current locations.

Although the complement of the aperture is a thin strip of magnetic conductor, THNAPP models this strip as a round magnetic wire. Even though wires and strips are obviously different geometries, the spatial average of the fields close to their surfaces are very similar if their cross sectional dimensions are small with respect to wavelength. As a basic rule of thumb, the diameter of the wire should be chosen to approximately equal the width of the aperture. The specification of the magnetic wires which represent the aperture during the M=4 loop is very similar to the specification of electric wires. In this case, however, the wires must lie along $\bar{H}$ field lines and thus are defined along the middle of cells, rather than along their edges. Also, wire endpoints must be in cell centers rather than apex points. The number of magnetic wires to be specified is indicated by the parameter NSLT, and the endpoints of each wire and its radius must be specified in the arrays SLTRAD, IMA, and IMB, respectively. As in the case of magnetic plates, the positions of the magnetic wires will have to be specified one half cell either above (outside) or below (inside) the actual aperture location. The later choice is generally best since the fields calculated at the actual aperture plane represent the limit of the fields as approached from the outside ($Z \to 0$; see equation 5). In light of this, subroutine EADV, which, among other things calculates the aperture fields needed for the M=5 loop, assumes that the magnetic wire has been specified one half cell inside the cavity (or below the aperture plane). Note that the version of THNAPP specified here assumes that the entire scatterer lies below (i.e., smaller values of x, y, or z) the aperture plane. Thus it is assumed that the forced currents are below the aperture plane. See also the description of EADV.

For M=5, the aperture fields calculated during the M=3 and M=4 loops are forced and the cavity fields are calculated. At this point, the programmer has a choice of running the M=5 loop using an unexpanded or expanded grid. The choice of whether an unexpanded or expanded grid is used will usually be dictated by the size of the cavity in back of the aperture. Small cavities will generally dictate the use of an expanded grid, whereas large cavities will need an unexpanded grid. If an unexpanded grid is used, the NOPE array should be filled using the normal rules, but realizing that those parts of the

scatterer that are not a part of the cavity need not be specified. Also, the rules for specifying the aperture itself during this loop are discussed in the results section.

If an expanded grid is to be used during the M=5 loop, the variable M5EXP must first have been set to 1 (one) during or before the M=1 loop in subroutine BUILD. Also, the variables indicating the locations of the bounding planes of the sub-volume to be analyzed during this loop (INEAR, IFAR, JNEAR, JFAR, KNEAR, KFAR - see [7]) need to be specified in the input file. Care must be taken to ensure that one of the surfaces of this sub-volume (of the unexpanded grid) correspond exactly with the aperture plane and contains the aperture and that the cavity to be analyzed during the M=5 loop lies entirely within the sub-volume (otherwise fields within the cavity will be incorrectly zeroed by subroutine OUTBND). The geometry specified via the array NOPE can contain as much internal detail as is consistent with the grid size, with no restrictions on the use of dielectrics or plates. The cavity wall in the aperture plane must be specified as a thin plate although other portions of the interior geometry may contain solid conducter or dielectric regions, as well as thin wires. Since the aperture plane must correspond to one of the sides of the expanded grid, these thin plate elements will reside in either the "bottom" (e.g. I=1) or "top" (e.g. I=29) of the expanded numerical grid.

The printout of BUILD that appears on the next page contains the complete specification for a test geometry used to obtain results presented in the results section. This geometry is specified in the "test" portion of this subroutine (i.e., Test=1 - see the description of BUILD in [7]). This printout does not contain a geometry specification for the "non-test" or "production" option (Test = 0) - this must be specified by the user.

```
      SUBROUTINE BUILD
C
      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
      COMMON/TSITEM/NOPE(29,29,29)
      COMMON/OUTLIST/DELX,DELY,DELZ,XPANX,XPANY,XPANZ,
     1    IUP,JUP,KUP,IDOWN,JDOWN,KDOWN,INEAR,JNEAR,KNEAR,
     2    IFAR,JFAR,KFAR,XOBS(6),YOBS(6),ZOBS(6),TEST,
     3    NPLANE(6)
      COMMON/WIRE/IA(50),IB(50),IDIR(400),MAPQI(400,6),MAPIQ(400,2),NI,
     1    NQ,NWRS,LOCQ(400),LOCI(400),DELS(400),CUR(400),Q(400),EWD(400),
     2    DIL(400),WRAD(50),SRAD(400),AIND(400),CAP(400),QFAC(400,2)
      COMMON/SLOT/IMA(50),IMB(50),MDIR(400),MMAPQI(400,6),
     1    MMAPIQ(400,2),NMI,NMQ,NSLT,LOCMQ(400),LOCMI(400),DELSM(400),
     2    CURM(400),QM(400),HMD(400),DILM(400),SLRAD(50),SMRAD(400),
     3    AMIND(400),CAPM(400),QMFAC(400,2)
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1   ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1   ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      COMMON/PERM/MM
      INTEGER TEST
C
      DO 50 I=1,NX
      DO 50 J=1,NY
      DO 50 K=1,NZ
      NOPE(I,J,K)=0
   50 CONTINUE
C
C      CYLINDER TEST
C
      IF(TEST .EQ. 0)GO TO 8888
C
C      DEFINE FORCED CURRENT SURFACES FOR THIN SEAM RUNS
C
C
C      IBAB=1 FOR THIN SEAM RUNS
C      IBAB=0 FOR NO THIN SEAMS
C
      IBAB=1
      M5EXP=0
C
C      BUILD M=1 GEOMETRY
C
   65 IF(M.NE.1)GO TO 200
      DO 101 I=9,23
      DO 101 J=12,20
      DO 101 K=12,17
      IF(I.LE.13 .AND. J.GT.17)GO TO 101
      NOPE(I,J,K)=4
  101 CONTINUE
C
C     INDICATE THE APERATURE PLANE
C
      LPDIR=2
```

```
      LP=21
      IF(IBAB.EQ.1) CALL FCBLD
      NWRS=1
      IA(1)=ITRAN(4,15,15)
      IB(1)=ITRAN(9,15,15)
      WRAD(1)=.01
      GO TO 500
C
C      M=2 GEOMETRY
C
  200 IF(M .NE. 2)GO TO 300
C
C      M=3 GEOMETRY (INCIDENT FIELDS)
C
  300 IF(M.NE.3.AND.M.NE.4) GO TO 400
      NLOC=3
      DO 350 I=1,NLOC
      LOCE(I)=ITRAN(20,21,13+I)
  350 CONTINUE
      IDIRE=1
      GO TO 400
C
C      M=4 GEOMETRY (MAGNETIC FIELDS CASE)
C
  400 IF(M.NE.4)GO TO 449
C
C      MAGNETIC CONDUCTING SHEET
C
      DO 401 I=1,NX
             J=21
      DO 401 K=1,NZ
      IF(I.GE.14 .AND. I.LE.23 .AND. K.GE.12 .AND. K.LE.17)GO TO 401
      NOPE(I,J,K)=-2
  401 CONTINUE
C
C      MAGNETIC WIRE
C
      NSLT=1
      IMA(1)=ITRAN(20,20,13)
      IMB(1)=ITRAN(20,20,16)
      SLRAD(1)=0.1
C
  449 CONTINUE
      IF(M .NE. 5)GOTO 500
      IF(M5EXP.EQ.1) GOTO 490
             I=9
      DO 441 J=12,17
      DO 441 K=12,17
      NOPE(I,J,K)=1
  441 CONTINUE
             I=14
      DO 442 J=18,20
      DO 442 K=12,17
      NOPE(I,J,K)=1
```

```
  442 CONTINUE
      DO 443 I=9,13
             J=18
      DO 443 K=12,17
      NOPE(I,J,K)=2
  443 CONTINUE
      DO 444 I=9,23
             J=12
      DO 444 K=12,17
      NOPE(I,J,K)=2
  444 CONTINUE
             I=24
      DO 445 J=12,20
      DO 445 K=12,17
      NOPE(I,J,K)=1
  445 CONTINUE
      DO 446 I=9,23
      DO 446 J=12,20
      IF(I .LE. 13.AND.J.GT.17) GO TO 446
      NOPE(I,J,12)=3
      NOPE(I,J,18)=3
  446 CONTINUE
      DO 447 I=12,23
             J=21
      DO 447 K=12,17
      IF(I .EQ. 20 .AND. (K.GE.13 .AND. K.LE..16) GO TO 447
      NOPE(I,J,K)=2
  447 CONTINUE
  448 CONTINUE
C
C     SEAMS
C
      DO 453 K=12,17
      NOPE(9,12,K)=12
  453 CONTINUE
      DO 454 J=12,17
      NOPE(9,J,12)=13
  454 CONTINUE
      DO 455 J=18,20
      NOPE(14,J,12)=13
  455 CONTINUE
      NOPE(9,12,12)=123
      GO TO 500
C
C     BUILD THE GEOMETRY FOR THE EXPANDED RUN
C
  490 DO 491 J=17,28
      DO 491 K=1,24
      NOPE(5,J,K)=1
      NOPE(25,J,K)=1
  491 CONTINUE
      DO 492 I=5,24
      DO 492 J=17,28
      NOPE(I,J,1)=3
```

```
      NOPE(I,J,25)=3
  492 CONTINUE
      DO 493 I=5,24
      DO 493 K=1,24
      NOPE(I,17,K)=2
  493 CONTINUE
      DO 897 I=5,24
      DO 897 K=1,24
      IF((I.EQ.14.OR.I.EQ.15).AND.(K.GE.7.AND.K.LE.18)) GO TO 897
      NOPE(I,29,K)=2
  897 CONTINUE
      DO 494 K=1,24
      NOPE(5,17,K)=12
  494 CONTINUE
      DO 895 J=17,28
      NOPE(5,J,1)=13
  895 CONTINUE
      DO 896 I=5,24
      NOPE(I,17,1)=12
  896 CONTINUE
      NOPE(5,17,1)=123
  500 CONTINUE
      GO TO 700
 8888 CONTINUE
C
C     BUILD "PRODUCTION" SCATTERER
C
  700 CONTINUE
      IF(NWRS.NE.0)CALL WIREBLD
      IF(NSLT.NE.0)CALL SLOTBLD
C
  497 FORMAT(4(10X,20I3/))
      RETURN
      END
```

Subroutine WIREBLD

The purpose of subroutine WIREBLD is to determine all of the points at which wire currents and charges will exist and all of the parameters needed by subroutine WIREADV to advance the currents, charges, and electric fields on and near the wires. All of the variables generic to wires are stored in the common block WIRE. WIREBLD is called automatically whenever wires are present (i.e., NWRS≠0).

This subroutine proceeds by first checking to see if all of the endpoint values IA are closer to the origin than the IB endpoint values, and correcting them if they aren't. The subroutine then proceeds to march sequentially along each cell of each wire to identify all the positions of the currents, charges, and electric fields of the wires. Care is taken to make sure that all wire intersections are sensed and accounted for.

Each wire is broken up into a series of segments, each one cell long. Current points are defined at each point along the wire that is coincident with a tangential electric field point. Charge points are defined at cell apex points (which straddle the current locations), including the wire ends. Since wires intersect only at charge points, there will be as many current locations as there are wire segments. On the other hand, the same charge point will be shared by all wires intersecting at that point. WIREBLD automatically checks for intersections and therefore does not redefine charge points as it marches down each of the wires. The description of subroutine WIREADV contains more information on the relationship between the wire segments and the numerical grid.

The following is a summary of the parameters and arrays filled by WIREBLD and stored in common block WIRE. Note, however, that the programmer need not be aware of the values of these variables in order to use THNAPP since they are used by WIREADV automatically.

NI - The total number of current positions defined.

NQ - The total number of charge positions defined.

LOCI(N) - The (coded) location of the N th current.

LOCQ(N) - The (coded) location of the N th charge.

CUR(N) - The current at the N th current position for the previous time step (not filled by WIREBLD).

Q(N) - The charge at the N th charge position for the previous time step (not filled by WIREBLD).

EWD(N) - The tangential electric field at the position of the N th current for the previous time step.

IDIR(N) - The orientation of the N th current element. 1, 2, or 3 for x, y, or z orientation, respectively.

CAP(N) - The capacitance of the N th segment.

AIND(N) - The inductance of the N th segment.

SRAD(N) - The radius of the N th segment.

DELS(N) - The cross sectional area of cell that the N th segment passes through.

MAPQI (M,N) - A list of the (coded) locations of all current points that are adjacent to the M th charge point. $1 < N < 6$. A zero value indicates no current location. This array is filled for the lowest values of N first.

MAPIQ(M,N) - The (coded) locations of the two charge locations that are adjacent to the M th current point. N=1 is the "left" most value (i.e. lowest value of I, J, or K), and N=2 is the "right" most location.

DIL(N) - The total length of all segments attached to the N th charge point. This number is needed by the continuity equation in order to advance the charges is time.

QFAC(M,N) - A list of two parameters (N=1 or 2) used to determine $\partial Q/\partial Z$ at the position of the M th current. These parameters contain information about the percentage of charge stored on either side of a junction of wires with different radii. N=1 or 2 are for charges on the "left" or "right" sides of the mth current, respectively.

```
      SUBROUTINE WIREBLD
C
      COMMON/WIRE/IA(50),IB(50),IDIR(400),MAPQI(400,6),MAPIQ(400,2),NI,
     1   NQ,NWRS,LOCQ(400),LOCI(400),DELS(400),CUR(400),Q(400),EWD(400),
     2   DIL(400),WRAD(50),SRAD(400),AIND(400),CAP(400),QFAC(400,2)
      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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      COMMON/TSITEM/NOPE(29,29,29)
      COMMON/PERM/MM
C
      DO 100 IWR=1,NWRS
      IF(IA(IWR).LT.IB(IWR))GO TO 100
      ITMP=IA(IWR)
      IA(IWR)=IB(IWR)
      IB(IWR)=ITMP
  100 CONTINUE
C
C  GENERATE WIRE CURRENT AND CHARGE POINTS
C
      II=0
      IQ=0
      DO 500 IWR=1,NWRS
      IF(IRTRAN(IA(IWR),1) .NE. IRTRAN(IB(IWR),1)) THEN
          IDIRW=1
          J=IRTRAN(IA(IWR),2)
          K=IRTRAN(IA(IWR),3)
          MIN=IRTRAN(IA(IWR),1)
          MAX=IRTRAN(IB(IWR),1)
      ELSE IF(IRTRAN(IA(IWR),2) .NE. IRTRAN(IB(IWR),2)) THEN
          IDIRW=2
          I=IRTRAN(IA(IWR),1)
          K=IRTRAN(IA(IWR),3)
          MIN=IRTRAN(IA(IWR),2)
          MAX=IRTRAN(IB(IWR),2)
      ELSE
          IDIRW=3
          J=IRTRAN(IA(IWR),2)
          I=IRTRAN(IA(IWR),1)
          MIN=IRTRAN(IA(IWR),3)
          MAX=IRTRAN(IB(IWR),3)
      END IF
      DO 500 IT=MIN,MAX
      IF(IDIRW .EQ. 1) I=IT
      IF(IDIRW .EQ. 2) J=IT
      IF(IDIRW .EQ. 3) K=IT
      JP=ITRAN(I,J,K)
      IF(IT .NE. MAX) THEN
          II=II+1
          LOCI(II)=JP
```

```
          IDIR(II)=IDIRW
          SRAD(II)=WRAD(IWR)
          IF(IDIRW .EQ. 1) THEN
              A=DY(J)/2.
              B=DZ(K)/2.
          ELSE IF(IDIRW .EQ. 2) THEN
              A=DX(I)/2.
              B=DZ(K)/2.
          ELSE
              A=DX(I)/2.
              B=DY(J)/2.
          END IF
          DELS(II)=4.*A*B
          AIND(II)=XMU/(4.*PI)*(ALOG((A**2+B**2)/SRAD(II)**2)
     <      +A/B*ATAN(B/A)+B/A*ATAN(A/B)+(SRAD(II)**2)*PI/(4.*A*B)-3.)
          CAP(II)=EPSO*XMU/AIND(II)
        END IF
        DO 120 IV=1,IQ
        IF(LOCQ(IV) .EQ. JP) THEN
          DO 125 IW=1,6
          IF(MAPQI(IV,IW) .EQ. 0)GO TO 126
  125     CONTINUE
  126     IQC=IV
          IF(IT .NE. MIN) THEN
              MAPQI(IQC,IW)=IIP
              IF(IT .NE. MAX) THEN
                  MAPQI(IQC,IW+1)=II
                  GO TO 130
              ELSE
                  GO TO 130
              END IF
          ELSE
              MAPQI(IQC,IW)=II
              GO TO 130
          END IF
        END IF
  120   CONTINUE
        IQ=IQ+1
        IQC=IQ
        LOCQ(IQ)=JP
        IF(IT.NE.MIN)THEN
          MAPQI(IQ,1)=IIP
          IF(IT .NE. MAX) THEN
              MAPQI(IQ,2)=II
          END IF
        ELSE
          MAPQI(IQ,1)=II
        END IF
  130   IF(IT .NE. MAX) MAPIQ(II,1)=IQC
        IF(IT .NE. MIN) MAPIQ(IIP,2)=IQC
        IIP=II
  500   CONTINUE
        NI=II
        NQ=IQ
C
```

```
C     CHARGE ADVANCE PARAMETERS
C
      DO 200 IQ=1,NQ
      DIL(IQ)=0.
      DO 200 IV=1,6
      II=MAPQI(IQ,IV)
      IF(II .NE. 0) THEN
          JP=LOCI(II)
          IF(IDIR(II) .EQ. 1)DIL(IQ)=DIL(IQ)+0.5*DX0(IRTRAN(JP,1))
          IF(IDIR(II) .EQ. 2)DIL(IQ)=DIL(IQ)+0.5*DY0(IRTRAN(JP,2))
          IF(IDIR(II) .EQ. 3)DIL(IQ)=DIL(IQ)+0.5*DZ0(IRTRAN(JP,3))
      END IF
  200 CONTINUE
C
C     CURRENT ADVANCE PARAMETERS
C
      DO 300 II=1,NI
      DO 300 IS=1,2
      IQ=MAPIQ(II,IS)
      JP=LOCQ(IQ)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      R0=0.5*(DX(I)*DY(J)*DZ(K))**(1./3.)
      CC=1./ALOG(R0/SRAD(II))
      DENOM=0.
      ANUM=0.
      DO 320 IV=1,6
      JJ=MAPQI(IQ,IV)
      IF(JJ.EQ.0)GO TO 320
      JP=LOCI(JJ)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      IF(IDIR(JJ).EQ.1)DEL2=DX0(I)/2.
      IF(IDIR(JJ).EQ.2)DEL2=DY0(J)/2.
      IF(IDIR(JJ).EQ.3)DEL2=DZ0(K)/2.
      DENOM=DENOM+DEL2/CC/ALOG(R0/SRAD(JJ))
      ANUM=ANUM+DEL2
  320 CONTINUE
      QFAC(II,IS)=ANUM/DENOM
  300 CONTINUE
C
C     DETERMINATION OF WIRE INTERSECTIONS WITH PLATES
C
      DO 340 IWR=1,NWRS
      DO 340 IS=1,2
      IF(IS.EQ.1)IPQ=IA(IWR)
      IF(IS.EQ.2)IPQ=IB(IWR)
      DO 350 IQ=1,NQ
      IF(LOCQ(IQ).EQ.IPQ)GO TO 360
  350 CONTINUE
  360 IF(MAPQI(IQ,2).NE.0)GO TO 340
      II=MAPQI(IQ,1)
      IPI=LOCI(II)
```

```
      I=IRTRAN(IPI,1)
      J=IRTRAN(IPI,2)
      K=IRTRAN(IPI,3)
      IF(IDIR(II).EQ.1 .AND. MAPIQ(II,1).EQ.IQ)THEN
          IF(NOPE(I-1,J,K).EQ.4 .OR. NOPE(I,J,K).EQ.1)LOCQ(IQ)=-IPQ
      ELSE IF(IDIR(II).EQ.1 .AND. MAPIQ(II,2).EQ.IQ)THEN
          IF(NOPE(I,J,K).EQ.4 .OR. NOPE(I+1,J,K).EQ.1)LOCQ(IQ)=-IPQ
      ELSE IF(IDIR(II).EQ.2 .AND. MAPIQ(II,1).EQ.IQ)THEN
          IF(NOPE(I,J-1,K).EQ.4 .OR. NOPE(I,J,K).EQ.2)LOCQ(IQ)=-IPQ
      ELSE IF(IDIR(II).EQ.2 .AND. MAPIQ(II,2).EQ.IQ)THEN
          IF(NOPE(I,J,K).EQ.4 .OR. NOPE(I,J+1,K).EQ.2)LOCQ(IQ)=-IPQ
      ELSE IF(IDIR(II).EQ.3 .AND. MAPIQ(II,1).EQ.IQ)THEN
          IF(NOPE(I,J,K-1).EQ.4 .OR. NOPE(I,J,K).EQ.3)LOCQ(IQ)=-IPQ
      ELSE IF(IDIR(II).EQ.3 .AND. MAPIQ(II,2).EQ.IQ)THEN
          IF(NOPE(I,J,K).EQ.4 .OR. NOPE(I,J,K+1).EQ.3)LOCQ(IQ)=-IPQ
      END IF
  340 CONTINUE
      IF(IBAB.EQ.1 .AND. M.EQ.1 .AND. MM.GT.12)THEN
      DO 370 II=1,NI
      LOCFC(NFC+II)=LOCI(II)
      IDIRFC(NFC+II)=IDIR(II)
  370 CONTINUE
      NFCT=NI
      ENDIF
      DO 400 II=1,NI
      WRITE(5,600)II,LOCI(II),IDIR(II),MAPIQ(II,1),MAPIQ(II,2)
  400 CONTINUE
      DO 410 IQ=1,NQ
      WRITE(5,600)IQ,LOCQ(IQ),(MAPQI(IQ,IV),IV=1,6)
  410 CONTINUE
      DO 412 II=1,NI
      WRITE(5,700)II,QFAC(II,1),QFAC(II,2)
  412 CONTINUE
      DO 413 IQ=1,NQ
      WRITE(5,700)IQ,DIL(IQ)
  413 CONTINUE
  700 FORMAT(I8,4E12.4)
  600 FORMAT(8I12)
      RETURN
      END
```

Subroutine WIREADV

The purpose of WIREADV is to advance the currents, charges, and electric fields in those cells that contain thin wires. It also stores the forced wire currents for use in subroutine EADV during the M=3 and 4 loops if thin apertures are to be analyzed. It is called EADV during every time cycle whenever wires are defined after the $\overline{E}$ fields have been first advanced under the assumption that no wires are present. WIREADV then readvances the $\overline{E}$ fields (along with the charges and currents) only in those cells that contain wires. The technique used to advance these quantities is basically the same as that developed by Holland and Simpson [6]. In this discussion, this technique will be briefly outlined and the code written for this technique will also be discussed.

Shown in Figure 4a and 4b are two orthogonal views of a wire, running along a cell lattice line. The wire is of radius "a" and lies along the z axis. Although the tangential electric field along the surface of the wire and the total electric field inside the wire is zero, the fields at lattice points along the wire center will in general not be zero since the fields calculated by a FDTD code represent average fields in the cell surrounding each point. Thus, since the wire radius is assumed to be small as compared to the cell dimension, the fields along these points will not be zero. Only as the wire dimensions start to approach those of the cell cross section will these fields indeed tend towards zero.

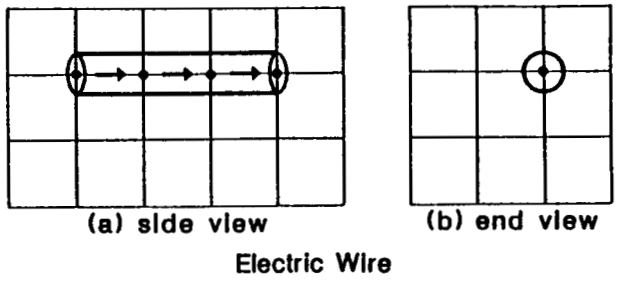

Fig. 4. Two views of the placement of an electric wire in the numerical grid. Current and charge locations are shown as arrows and dots, respectively.

From [6], the average tangential electric field in the cells containing the wire segments is related to the charge and current on the wire by:

$$E_z + E_i = L \frac{\partial I}{\partial t} + \frac{1}{C} \frac{\partial Q}{\partial z} \quad , \tag{6}$$

where I is the current and Q is the charge per unit length along the wire. E_z and E_i denote scattered and incident (if present) components of the field, respectively. Also, L and C are effective cell inductances and capacitances per meter, respectively. Appropriate expressions for L and C are given by [6]:

$$L = \frac{\mu}{4\pi} \left\{ \text{Ln}\left[\frac{\frac{\Delta x^2}{2} + \frac{\Delta y^2}{2}}{a^2} \right] + \frac{\Delta x}{\Delta y} \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) + \frac{\Delta y}{\Delta x} \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) + \frac{\pi a^2}{\Delta x \Delta y} - 3 \right\} , \tag{7}$$

and

$$C = \frac{\mu\varepsilon}{L} \quad , \tag{8}$$

where Δx and Δy are the cross sectional dimensions of the cell.

In WIREADV, currents (shown as arrows in Figure 4a) are evaluated along the grid lattice lines at the same locations as the electric field tangent to the wire, and the charges (shown as dots in Figure 4a) are evaluated in between the current locations. I is evaluated at the same points in time as is $\bar{E}$, whereas Q is evaluated at interlaced points in time (the same as $\bar{H}$). This scheme appears to yield good numerical stability. Thus, the differenced form of equation (6) is:

$$\tfrac{1}{2}\left[E_z^{n+1} + E_z^{n}\right] + E_i^{n+\frac{1}{2}} = \frac{L}{\Delta t}\left[I^{n+1} - I^{n}\right] + \frac{1}{C}\frac{\partial}{\partial z} Q^{-n+\frac{1}{2}} \quad . \tag{9}$$

The superscripts in the above quantities indicate the points in time where they are evaluated. The different versions of other pertinent equations relating the field quantities I, Q, E, and H are 1) Maxwell's curl $\bar{H}$ equation:

$$\left(\nabla \times \bar{H}^{n+1/2}\right)_z = \frac{1}{2\Delta s}\left(I^{n+1/2} + I^{n}\right) + \frac{\varepsilon}{\Delta t}\left(E^{n+1/2} - E^{n}\right), \qquad (10)$$

and 2), the continuity equation

$$\frac{1}{\Delta t}\left(Q^{n+1/2} - Q^{n-1/2}\right) + \frac{\partial}{\partial z} I^{n} = 0 \qquad (11)$$

In equation (10), the term $(\nabla \times \bar{H})_z$ denotes the component of curl $\bar{H}$ that is tangent to the wire. Also, Δs is the crossed sectional area of the cell. Since both equations (4) and (5) contain the variables E and I, E can be eliminated, yielding

$$I^{n+1} = \frac{(\Delta t)^2}{4\varepsilon \Delta s\, L + (\Delta t)^2}\left\{2\Delta s\left(\nabla \times H^{n+1/2}\right)_z + \frac{4\varepsilon \Delta s}{\Delta t}\left(E^{n} + E_i^{n+1/2}\right) - \frac{4\varepsilon \Delta s}{c\Delta t}\frac{\partial}{\partial z} Q^{n+1/2}\right\} \qquad (12)$$

Once equation (12) has been used to advance I, E can be advanced from the application of equation (9) or (10).

The overall order in which the field quantities relation to wires are evaluated in THNAPP and the subroutine WIREADV is:

(1) Subroutine EADV advances E in all cells under the assumption that no wires are present in the geometry.

(2) WIREADV calculates I in cells containing wires using equation (12).

(3) WIREADV calculates E in cells containing wires using equation (9) or (10).

(4) WIREADV calculates Q from equation (11).

(5) EBC is called to set the correct fields on and within conducting sheets and solids.

(6) HADV is called to advance the H fields (same as in G3DXL).

(7) SLOTADV is called to modify the H field in cells containing magnetic wires (if present).

(8) HBC is called to set the correct fields on magnetic surfaces (if present).

When dealing with straight wires that have no changes in radius, WIREADV proceeds by directly utilizing equations (10), (11), and (12), to advance $\bar{E}$, I, and Q. However, when dealing with the junctions of wires of different radii and/or multiple wire junctions, modifications of these expressions must

be used in order to correctly evaluate the terms $\frac{\partial}{\partial z} I$ (which in its most general case is actually $\nabla \bullet \bar{J}$) and $\frac{\partial}{\partial z} Q$ [6]. Shown in Figure 5 is a junction of wires, each having a different radius. The charge points are

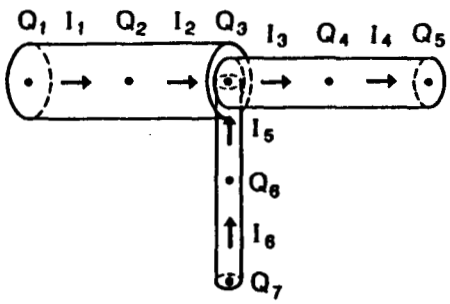

Fig. 5. Three wires intersecting wires, each of different radii. Current and charge locations are shown as arrows and dots, respectively.

shown as dots and the current points are shown as arrows. The charges and currents have been given arbitrary numbers for the sake of illustration. For instance, in order to advance the charge density Q(3), $\nabla \bullet \bar{J}$ must be evaluated at this point taking into account all the current flowing into this junction. This is accomplished by summing all of the currents flowing into this node and dividing by the sum of the segment lengths. The array MAPQI contains the list of all currents that flow into this node. Once the total current flowing into this node has been summed, the charge density is advanced by the formula

$$Q^{n+\frac{1}{2}}(M) = Q^{n-\frac{1}{2}}(M) + \Delta t \frac{\sum_i I_i}{\frac{1}{2}\sum_i \Delta \ell_i} = Q^{n-\frac{1}{2}}(M) + \frac{\Delta t \sum_i I_i}{DIL(M)} , \qquad (13)$$

where $\Delta \ell_i$ is the length of the ith segment and DIL(M) is the sum of the half lengths of wire segments attached to the M th charge.

The correct evaluation of equation (12) demands that the discontinuity of the charge density on different sides of multiple wire junctions of different radii be accounted for. The size of this discontinuity is dependent upon the relative values of the capacitance of each of these segments. Thus, if one wants to evaluate I(2) in Figure 5, one cannot assume that the charge density on the right side of this current element is Q(3) since Q(3) represents only an average charge density at the junction. In WIREADV, the correct charge

densities that straddle a particular current I(M) are found from the product of the charge densities stored in the array Q times the factors QFAC(M,1) or QFAC(M,2), depending upon whether the charge is to the "left" (i.e. small values of I, J, or K) or the "right" side, respectively. The formula for determining the $\frac{\partial Q}{\partial z}$ term at the M th current is thus

$$\frac{\partial Q}{\partial z} = \frac{1}{\Delta z} \left[Q(B) * QFAC(M,2) - Q(A) * QFAC(M,2) \right] \tag{14}$$

where Q(A) and Q(B) are the charge points to the "right" and "left" of I(M), respectively. Remember also that the array MAPIQ contains the list of charges that are adjacent to each current. The expression for QFAC(M,N) is

$$QFAC(M,N) = \frac{\sum_i \Delta \ell_i/2}{\frac{1}{CAP(M)} \sum_j \left(CAP\ (j) * (\Delta \ell_j/2) \right)} \tag{15}$$

where the sums include all segments attached to the "right" (N=1) or "left" (M=2) hand charge that straddles the current I(M).

The computer variables that are used in WIREADV in the above formula are DX0, DY0, or DZ0, depending upon whether the segments are x, y, or z directed. For current elements that are not adjacent to junctions, the elements of the QFAC array are equal to 1, meaning that the charge density at those points is the same as that stored in the Q array.

Finally, during the M=1 loop WIREADV calculates and stores the time histories of the external wire currents (for use in subroutine EADV during the M=3 and 4 loops) if the parameter IBAB = 1. The values stored in the array FCUR are calculated from the Maxwell's curl equation:

$$\bar{J} = \nabla \times \bar{H} - \varepsilon \frac{\partial}{\partial t} \bar{E} \tag{16}$$

It should be noted that the subscripts of the FCUR array that are specified in this portion of the subroutine (just above the 12 Continue statements) are FCUR (II+NFC,N). The second subscript, N, indicates the time cycle. The first subscript, II+NFC, indicates that it is the II'th wire current that is being stored, but it is located in the II+NFC'th location in FCUR. The reason for the shift term NFC is that the array FCUR is used for both forced surface and wire currents. The surface currents are stored in the first NFC locations (NFC being the number of forced surface current locations) and the forced wire currents are stored in the remaining NI locations.

```
      SUBROUTINE WIREADV
C
      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,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,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),DYI(29),DZI(29),DX0I(28),DY0I(28),DZ0I(28)
      COMMON/WIRE/IA(50),IB(50),IDIR(400),MAPQI(400,6),MAPIQ(400,2),NI,
     1   NQ,NWRS,LOCQ(400),LOCI(400),DELS(400),CUR(400),Q(400),EWD(400),
     2   DIL(400),WRAD(50),SRAD(400),AIND(400),CAP(400),QFAC(400,2)
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      COMMON/PERM/MM
C
      DTE=DT/EPSO
      T=T-DT/2.
      DO 12 II=1,NI
C
C     ADVANCE WIRE CURRENT
C
      C1=(2.*DELS(II)*DT*DT)/(4.*AIND(II)*EPSO*DELS(II)+DT**2)
      C2=(4.*AIND(II)*EPSO*DELS(II)-DT**2)/(2.*DELS(II)*DT**2)
      C3=2.*EPSO/DT
      C4=2.*DT**2/(4.*AIND(II)*EPSO*DELS(II)+DT**2)
      PLSE=PULSE(II)
      JP=LOCI(II)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      DQ=Q(MAPIQ(II,2))*QFAC(II,2)-Q(MAPIQ(II,1))*QFAC(II,1)
      OLDCUR=CUR(II)
      EWDD=EWD(II)
        IF(IDIR(II) .EQ. 1) THEN
           CUR(II)=C1*(C2*CUR(II)+C3*(EWD(II)+EINCX(I,J,K)+PLSE
     <      -1./(CAP(II)*DX(I))*DQ)
     <      +(HZ(I,J,K)-HZ(I,J-1,K))*DYI(J)
     <      -(HY(I,J,K)-HY(I,J,K-1))*DZI(K))
        ELSE IF(IDIR(II) .EQ. 2) THEN
           CUR(II)=C1*(C2*CUR(II)+C3*(EWD(II)+EINCY(I,J,K)+PLSE
     <      -1./(CAP(II)*DY(J))*DQ)
     <      +(HX(I,J,K)-HX(I,J,K-1))*DZI(K)
     <      -(HZ(I,J,K)-HZ(I-1,J,K))*DXI(I))
        ELSE
           CUR(II)=C1*(C2*CUR(II)+C3*(EWD(II)+EINCZ(I,J,K)+PLSE
     <      -1./(CAP(II)*DZ(K))*DQ)
     <      +(HY(I,J,K)-HY(I-1,J,K))*DXI(I)
     <      -(HX(I,J,K)-HX(I,J-1,K))*DYI(J))
        END IF
C
C     ADVANCE E
C
```

```
      IF(IDIR(II) .EQ. 1) THEN
          EX(I,J,K)=2.*AIND(II)/DT*(CUR(II)-OLDCUR)-EWD(II)-2.*PLSE
     <     -2.*EINCX(I,J,K)+2./(CAP(II)*DX(I))*DQ
          EWD(II)=EX(I,J,K)
      ELSE IF(IDIR(II) .EQ. 2) THEN
          EY(I,J,K)=2.*AIND(II)/DT*(CUR(II)-OLDCUR)-EWD(II)-2.*PLSE
     <     -2.*EINCY(I,J,K)+2./(CAP(II)*DY(J))*DQ
          EWD(II)=EY(I,J,K)
      ELSE
          EZ(I,J,K)=2.*AIND(II)/DT*(CUR(II)-OLDCUR)-EWD(II)-2.*PLSE
     <     -2.*EINCZ(I,J,K)+2./(CAP(II)*DZ(K))*DQ
          EWD(II)=EZ(I,J,K)
      END IF
      IF(IBAB .EQ. 0 .OR. M.NE.1 .OR. MM.LE.12)GO TO 12
      IF(IDIR(II) .EQ. 1) THEN
        FCUR(II+NFC,N)=(HZ(I,J,K)-HZ(I,J-1,K))*DYI(J)
     <                 -(HY(I,J,K)-HY(I,J,K-1))*DZI(K)
     <                 +(1/DTE)*(EWDD-EWD(II))
      ELSE IF(IDIR(II) .EQ. 2) THEN
        FCUR(II+NFC,N)=(HX(I,J,K)-HX(I,J,K-1))*DZI(K)
     <                 -(HZ(I,J,K)-HZ(I-1,J,K))*DXI(I)
     <                 +(1/DTE)*(EWDD-EWD(II))
      ELSE IF(IDIR(II) .EQ. 3) THEN
        FCUR(II+NFC,N)=(HY(I,J,K)-HY(I-1,J,K))*DXI(I)
     <               -(HX(I,J,K)-HX(I,J-1,K))*DYI(J)
     <             +(1/DTE)*(EWDD-EWD(II))
        ENDIF
   12 CONTINUE
      T=T+DT/2.
C
C     ADVANCE CHARGE
C
      DO 20 IQ=1,NQ
      IF(LOCQ(IQ).LT.0)GO TO 20
      SUM=0.
      DO 30 IV=1,6
      II=MAPQI(IQ,IV)
      IF(II .EQ. 0)GO TO 30
      IF(MAPIQ(II,2) .EQ. IQ)SUM=SUM+CUR(II)
      IF(MAPIQ(II,1) .EQ. IQ)SUM=SUM-CUR(II)
   30 CONTINUE
      Q(IQ)=Q(IQ)+DT/DIL(IQ)*SUM
   20 CONTINUE
      RETURN
      END
```

Function PULSE

Function PULSE is called by WIREADV. Its purpose is to allow a wire to be excited at some point or points by a pre-defined pulse function. It is intended for situations in which the geometry is to be excited by a current pulse initiated on one of the wires. Thus, PULSE returns nonzero values only when IDLS = 1 and the wire segment number matches the one(s) specified in PULSE.

When dealing with geometries involving thin apertures, care must be taken to ensure that PULSE returns nonzero values only for the M=1 loop. Otherwise, PULSE will attempt to excite cavity wires directly, rather than allowing them to be excited by the aperture fields alone.

```
      FUNCTION PULSE(II)
C
      COMMON/UGRID/UX(28),UY(28),UZ(28),UX0(29),UY0(29),UZ0(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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
C            THREAT I
C
      PULSE=0.0
      IF(II .NE. 3 .OR. IDLS .EQ. 0) GO TO 1
      TAU = T - (UY0(23) - Y0(22))/C
      IF (TAU.LE.0.0)GO TO 1
      PULSE =AMP/2.*(1.-COS(ALPHA*TAU))
      IF(TAU .GE. PI/ALPHA)PULSE=AMP
    1 CONTINUE
      RETURN
      END
```

Subroutine SLOTBLD

Subroutine SLOTBLD performs the same function for magnetic wires that WIREBLD performs for electric wires. Like WIREBLD, it starts with the defined values of the magnetic wire endpoints (which must be in cell centers) and segments the entire magnetic wire structure, defining magnetic currents and charges along each segment. Figure 6 shows the orientation of a typical magnetic wire in the spatial grid.

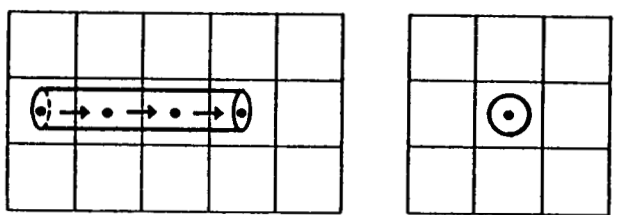

Fig. 6. Two views of the placement of a magnetic wire in the numerical grid. Current and charge locations are shown as arrows and dots, respectively.

Except for the fact that magnetic wires must be defined along $\bar{H}$ field points rather than $\bar{E}$ field points as in the case of electric wires, the algorithm and variables in SLOTBLD are nearly the same as that of WIREBLD. For this reason, the variable names generated by SLOTBLD are nearly the same as those of WIREBLD, with the addition of a preceding "M" for most. The following is a table that indicates a variable in the common block SLOT and its electric wire counterpart in common block WIRE:

SLOTBLD Variable	WIREBLD Variable
NMI	NI
NMQ	NQ
LOCMI	LOCI
LOCMQ	LOCQ
CURM	CUR
QM	Q

HMD . . . . . .EMD

MDIR. . . . . .IDIR

CAPM. . . . . .CAP

AMIND . . . . . .AIND

SMRAD . . . . . .SRAD

DELSM . . . . . .DELS

MMAPQI. . . . . .MAPQI

MMAPIQ. . . . . .MAPIQ

DILM. . . . . .DIL

QMFAX . . . . . .QFAC

The function of each of the above variables can be deduced by noting the function of its electric wire counterpart in the description of WIREBLD.

```
      SUBROUTINE SLOTBLD
C
      COMMON/SLOT/IMA(50),IMB(50),MDIR(400),MMAPQI(400,6),
     1   MMAPIQ(400,2),NMI,NMQ,NSLT,LOCMQ(400),LOCMI(400),DELSM(400),
     2   CURM(400),QM(400),HMD(400),DILM(400),SLRAD(50),SMRAD(400),
     3   AMIND(400),CAPM(400),QMFAC(400,2)
      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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      COMMON/TSITEM/NOPE(29,29,29)
C
      DO 100 ISL=1,NSLT
      IF(IMA(ISL).LT.IMB(ISL))GO TO 100
      ITMP=IMA(ISL)
      IMA(ISL)=IMB(ISL)
      IMB(ISL)=ITMP
  100 CONTINUE
C
C  GENERATE MAGNETIC WIRE CURRENT AND CHARGE POINTS
C
      II=0
      IQ=0
      DO 500 ISL=1,NSLT
      IF(IRTRAN(IMA(ISL),1) .NE. IRTRAN(IMB(ISL),1)) THEN
          MDIRW=1
          J=IRTRAN(IMA(ISL),2)
          K=IRTRAN(IMA(ISL),3)
          MIN=IRTRAN(IMA(ISL),1)
          MAX=IRTRAN(IMB(ISL),1)
      ELSE IF(IRTRAN(IMA(ISL),2) .NE. IRTRAN(IMB(ISL),2)) THEN
          MDIRW=2
          I=IRTRAN(IMA(ISL),1)
          K=IRTRAN(IMA(ISL),3)
          MIN=IRTRAN(IMA(ISL),2)
          MAX=IRTRAN(IMB(ISL),2)
      ELSE
          MDIRW=3
          J=IRTRAN(IMA(ISL),2)
          I=IRTRAN(IMA(ISL),1)
          MIN=IRTRAN(IMA(ISL),3)
          MAX=IRTRAN(IMB(ISL),3)
      END IF
      DO 500 IT=MIN,MAX
      IF(MDIRW .EQ. 1) THEN
          I=IT
          JP1=ITRAN(I+1,J,K)
      ELSE IF(MDIRW .EQ. 2) THEN
          J=IT
          JP1=ITRAN(I,J+1,K)
      ELSE
          K=IT
          JP1=ITRAN(I,J,K+1)
      END IF
      JP=ITRAN(I,J,K)
```

```
      IF(IT .NE. MAX) THEN
          II=II+1
          LOCMI(II)=JP1
          MDIR(II)=MDIRW
          SMRAD(II)=SLRAD(ISL)
          IF(MDIRW .EQ. 1) THEN
              A=DY0(J)/2.
              B=DZ0(K)/2.
          ELSE IF(MDIRW .EQ. 2) THEN
              A=DX0(I)/2.
              B=DZ0(K)/2.
          ELSE
              A=DX0(I)/2.
              B=DY0(J)/2.
          END IF
          DELSM(II)=4.*A*B
          AMIND(II)=EPSO/(4.*PI)*(ALOG((A**2+B**2)/SMRAD(II)**2)
     <      +A/B*ATAN(B/A)+B/A*ATAN(A/B)+(SMRAD(II)**2)*PI/(4.*A*B)-3.)
          CAPM(II)=EPSO*XMU/AMIND(II)
      END IF
      DO 120 IV=1,IQ
      IF(LOCMQ(IV) .EQ. JP) THEN
          DO 125 IW=1,6
          IF(MMAPQI(IV,IW) .EQ. 0)GO TO 126
  125     CONTINUE
  126     IQC=IV
          IF(IT .NE. MIN) THEN
              MMAPQI(IQC,IW)=IIP
              IF(IT .NE. MAX) THEN
                  MMAPQI(IQC,IW+1)=II
                  GO TO 130
              ELSE
                  GO TO 130
              END IF
          ELSE
              MMAPQI(IQC,IW)=II
              GO TO 130
          END IF
      END IF
  120 CONTINUE
      IQ=IQ+1
      IQC=IQ
      LOCMQ(IQ)=JP
      IF(IT.NE.MIN)THEN
          MMAPQI(IQ,1)=IIP
          IF(IT .NE. MAX) THEN
              MMAPQI(IQ,2)=II
          END IF
      ELSE
          MMAPQI(IQ,1)=II
      END IF
  130 IF(IT .NE. MAX) MMAPIQ(II,1)=IQC
      IF(IT .NE. MIN) MMAPIQ(IIP,2)=IQC
      IIP=II
  500 CONTINUE
      NMI=II
```

```
      NMQ=IQ
C
C     MAGNETIC CHARGE ADVANCE PARAMETERS
C
      DO 200 IQ=1,NMQ
      DILM(IQ)=0.
      JP=LOCMQ(IQ)
      DO 200 IV=1,6
      II=MMAPQI(IQ,IV)
      IF(II .NE. 0) THEN
          IF(MDIR(II) .EQ. 1)DILM(IQ)=DILM(IQ)+0.5*DX0(IRTRAN(JP,1))
          IF(MDIR(II) .EQ. 2)DILM(IQ)=DILM(IQ)+0.5*DY0(IRTRAN(JP,2))
          IF(MDIR(II) .EQ. 3)DILM(IQ)=DILM(IQ)+0.5*DZ0(IRTRAN(JP,3))
      END IF
  200 CONTINUE
C
C     MAGNETIC CURRENT ADVANCE PARAMETERS
C
      DO 300 II=1,NMI
      DO 300 IS=1,2
      IQ=MMAPIQ(II,IS)
      JP=LOCMQ(IQ)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      R0=0.5*(DX0(I)*DY0(J)*DZ0(K))**(1./3.)
      CC=1./ALOG(R0/SMRAD(II))
      DENOM=0.
      ANUM=0.
      DO 320 IV=1,6
      JJ=MMAPQI(IQ,IV)
      IF(JJ.EQ.0)GO TO 320
      IF(MDIR(JJ).EQ.1)DEL2=DX0(I)/2.
      IF(MDIR(JJ).EQ.2)DEL2=DY0(J)/2.
      IF(MDIR(JJ).EQ.3)DEL2=DZ0(K)/2.
      DENOM=DENOM+DEL2/CC/ALOG(R0/SMRAD(JJ))
      ANUM=ANUM+DEL2
  320 CONTINUE
      QMFAC(II,IS)=ANUM/DENOM
  300 CONTINUE
      DO 400 II=1,NMI
      WRITE(5,600)II,LOCMI(II),MDIR(II),MMAPIQ(II,1),MMAPIQ(II,2)
  400 CONTINUE
      DO 410 IQ=1,NMQ
      WRITE(5,600)IQ,LOCMQ(IQ),(MMAPQI(IQ,IV),IV=1,6)
  410 CONTINUE
      DO 412 II=1,NMI
      WRITE(5,700)II,QMFAC(II,1),QMFAC(II,2),AMIND(II),CAPM(II)
  412 CONTINUE
      DO 413 IQ=1,NMQ
      WRITE(5,700)IQ,DILM(IQ)
  413 CONTINUE
  700 FORMAT(I8,4E12.4)
  600 FORMAT(8I12)
      RETURN
      END
```

Subroutine SLOTADV

This subroutine is called by HADV during every time cycle after the $\overline{H}$ fields have first been advanced under the assumption that no magnetic wires are present. Its function is to correctly advance the H fields along each wire and also the magnetic currents and charges.

The technique used to advance these quantities is the electrical dual of the wire technique used in WIREADV. Thus, one can transform the equations that advance the H field, magnetic currents, and magnetic charges (equations in the WIREADV section) by making the following substitutions [13]:

WIREADV variable	SLOTADV variable
E	H
H	-E
I	I_m

Performing these substitutions on equations in the WIREADV, the following advancing equations are obtained:

$$I_m^{n+1/2} = \frac{(\Delta t)^2}{4\varepsilon\Delta SL + (\Delta t)^2} \left\{ -2\Delta S\left(\nabla \times E^n\right)_z + \frac{4\varepsilon\Delta S}{\Delta t}\left(H^{n-1/2} + H_i^n\right) - \frac{4\varepsilon\Delta S}{C\Delta t}\frac{\partial}{\partial z} Q_M^n \right\} \quad (17)$$

and

$$Q_M^{n+1} = Q_M^n - \Delta t \frac{\partial}{\partial t} I_M^{n+1/2} \quad (18)$$

SLOTADV advances each of the magnetic wire currents, $\overline{H}$ fields, and charges in the same order as the above equations. The cases of magnetic wire junctions are handled exactly the same as their dual quantities in WIREADV.

```
      SUBROUTINE SLOTADV
C
      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,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,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),DYI(29),DZI(29),DX0I(28),DY0I(28),DZ0I(28)
      COMMON/SLOT/IMA(50),IMB(50),MDIR(400),MMAPQI(400,6),
     1   MMAPIQ(400,2),NMI,NMQ,NSLT,LOCMQ(400),LOCMI(400),DELSM(400),
     2   CURM(400),QM(400),HMD(400),DILM(400),SLRAD(50),SMRAD(400),
     3   AMIND(400),CAPM(400),QMFAC(400,2)
C
      T=T-DT/2.
      DO 12 II=1,NMI
C
C     ADVANCE MAGWIRE H
C
      C1=(4.*AMIND(II)*DELSM(II)*DT)/(4.*AMIND(II)*XMU*DELSM(II)+DT**2)
      C2=(4.*AMIND(II)*XMU*DELSM(II)-DT**2)/(4.*AMIND(II)*XMU*DELSM(II)
     <  +DT**2)
      C3=DT/(2.*AMIND(II)*CAPM(II)*DELSM(II))
      C4=2.*DT**2/(4.*AMIND(II)*XMU*DELSM(II)+DT**2)
      JP=LOCMI(II)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      DQ=QM(MMAPIQ(II,2))*QMFAC(II,2)-QM(MMAPIQ(II,1))*QMFAC(II,1)
      IF(MDIR(II) .EQ. 1) THEN
          HX(I,J,K)=C1*(-CURM(II)/DELSM(II)+C3/DX(I)*DQ
     <      -(EZ(I,J+1,K)-EZ(I,J,K))*DY0I(J)
     <      +(EY(I,J,K+1)-EY(I,J,K))*DZ0I(K))
     <      +C2*HMD(II)-C4*HINCX(I,J,K)
      ELSE IF(MDIR(II) .EQ. 2) THEN
          HY(I,J,K)=C1*(-CURM(II)/DELSM(II)+C3/DY(J)*DQ
     <      -(EX(I,J,K+1)-EX(I,J,K))*DZ0I(K)
     <      +(EZ(I+1,J,K)-EZ(I,J,K))*DX0I(I))
     <      +C2*HMD(II)-C4*HINCY(I,J,K)
      ELSE
          HZ(I,J,K)=C1*(-CURM(II)/DELSM(II)+C3/DZ(K)*DQ
     <      -(EY(I+1,J,K)-EY(I,J,K))*DX0I(I)
     <      +(EX(I,J+1,K)-EX(I,J,K))*DY0I(J))
     <      +C2*HMD(II)-C4*HINCZ(I,J,K)
      END IF
C
C     ADVANCE WIRE CURMRENT
C
      IF(MDIR(II) .EQ. 1) THEN
         CURM(II)=CURM(II)+DT/2./AMIND(II)*(HX(I,J,K)+HMD(II))
     <     -DT/(AMIND(II)*CAPM(II)*DX(I))*DQ+DT/AMIND(II)*HINCX(I,J,K)
          HMD(II)=HX(I,J,K)
      ELSE IF(MDIR(II) .EQ. 2) THEN
         CURM(II)=CURM(II)+DT/2./AMIND(II)*(HY(I,J,K)+HMD(II))
```

```
     <     -DT/(AMIND(II)*CAPM(II)*DY(J))*DQ+DT/AMIND(II)*HINCY(I,J,K)
           HMD(II)=HY(I,J,K)
      ELSE
         CURM(II)=CURM(II)+DT/2./AMIND(II)*(HZ(I,J,K)+HMD(II))
     <     -DT/(AMIND(II)*CAPM(II)*DZ(K))*DQ+DT/AMIND(II)*HINCZ(I,J,K)
           HMD(II)=HZ(I,J,K)
      END IF
   12 CONTINUE
      T=T+DT/2.
C
C     ADVANCE CHARGE
C
      DO 20 IQ=1,NMQ
      SUM=0.
      DO 30 IV=1,6
      II=MMAPQI(IQ,IV)
      IF(II .EQ. 0)GO TO 30
      IF(MMAPIQ(II,2) .EQ. IQ)SUM=SUM+CURM(II)
      IF(MMAPIQ(II,1) .EQ. IQ)SUM=SUM-CURM(II)
   30 CONTINUE
      QM(IQ)=QM(IQ)+DT/DILM(IQ)*SUM
   20 CONTINUE
      RETURN
      END
```

Subroutine FCBLD

The purpose of subroutine FCBLD is to determine the positions of the forced surface current locations when the thin seam option is used. FCBLD operates automatically, using only the NOPE array as its input. The output of this subroutine is a set of arrays (in common block FORCE) that contain the locations and spatial orientations of these surface currents, as well as a list of the magnetic field locations that drive them.

FCBLD assumes that in the proceeding call of subroutine BUILD, the geometry to be analyzed has been built using only solid conducting blocks (NOPE = 4) and conducting surfaces (NOPE = 1, 2, or 3). It is further assumed that there are no hollow sections within the geometry, and the conducting surfaces do not intersect with each other perpendicularly, although intersections with the solid sections are allowed (thus allowing the modeling of structures such as aircraft wings, etc.). It is also assumed that the direction (LPDIR) and elevation (LP) of the aperture plane have also been specified at or before the M=1 section of BUILD.

FCBLD proceeds by first identifying the surfaces of the geometry. This is accomplished by marching through the numerical grid, cell by cell, and at each cell checking the values of the NOPE array in rings of four adjacent cells in all three orthogonal orientations about each cell. These three orthogonal directional searches correspond to the three possible orientations of the surface currents. The cell by cell search through the numerical grid is accomplished in the DO 1 loop, and the three orthogonal directions within this search are performed in the DO 20 loop.

The first test performed for each of the orthogonal loops about each cell is whether or not any of these cells contains a conducting plane. If none are present, FCBLD then sums the values of NOPE for the four cells in the loop. If this sum (IADD) equals zero, the cell is in free space and not adjacent to the scatterer. If it equals 16, it is in the interior of the scatterer. In either of these cases, there is no surface current along this direction at this cell position. All other values of IADD indicate that a surface current does exist at this location. FCBLD then tests which of the cells adjacent to the surface current location contain sold conductor and free space and from this is able to determine which H field(s) control each of these surface currents.

Figure 7 depicts a portion of an example object to help show the output of FCBLD. Shown is a conducting solid section on the left, with a thin conducting plane on the right that butts against the solid section. Several (of the many) currents that would be identified by FCBLD are shown as the numbered currents. Also shown in this figure is the numerical grid, complete with I, J, and K cell number identifiers.

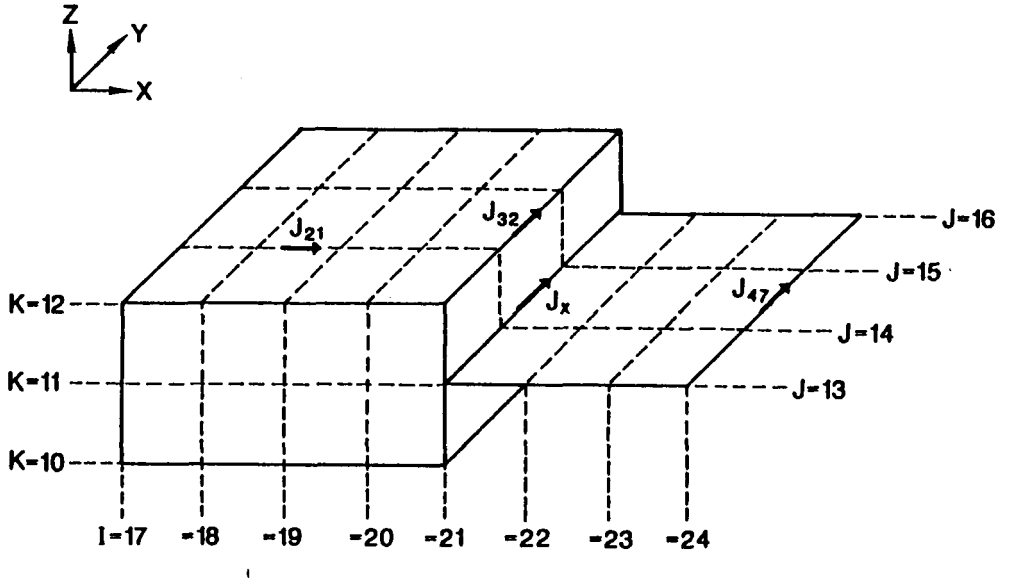

Fig. 7. A representative scattering geometry showing various surface current locations identified by subroutine FCBLD for use as forced currents in the M=3 and M=4 loops.

An example of the simplest type of current location for FCBLD to identify is J_{21} which is a $\hat{x}$ directed current on the solid section at cell position (18, 14,12), away from any bends of seams. In this case, of the four $\overline{H}$ fields surrounding this location that could control this current, only the $\hat{y}$ directed $\overline{H}$ field in cell (18,14,12) is nonzero (since the $\overline{H}$ fields perpendicular to and inside a perfect conductor are zero). Thus, this current is specified by the following list of variables:

```
LOCFC(21) = ITRAN(18,14,12)
IDIRFC(21) = 1
IHVAL(21,1) = 0
IHVAL(21,2) = 0
IHVAL (21,3) = 1
IHVAL (21,4) = 0
```

where the values of LOCFC and IDIRFC indicate that this current is located at position (18,14,12) and is $\hat{x}$ directed.

The purpose array IHVAL derives from the fact that STOCUR uses the expression

$$\nabla \times \bar{H} = \bar{J} + \cancel{\frac{\partial \bar{E}}{\partial t}}^{\,0} \qquad (19)$$

to calculate the surface currents to be forced in later loops. Although it is certainly easy to calculate $\nabla \times \bar{H}$ at each forced current location and, physically, the $\bar{H}$ fields perpendicular to and inside a perfectly conducting surface are identically zero, it is possible (when the "scattered field mode", IDLS=0, is used) that the values actually calculated by the code may not be zero. These nonzero fields do not in any way cause problems in the calculation of the fields outside the conductor surfaces, but they would cause the incorrect calculation of the forced surface currents as calculated by equation 19. To circumvent this, the values of IHVAL act as binary multipliers in the $\nabla \times \bar{H}$ operator of subroutine STOCUR so that only those fields that will be physically nonzero are actually used. The numbering system used (i.e. 1,2,3,4) coincides with the order in which the four H field values are used in the curl operator in subroutine STOCUR.

Current locations that lie along "exterior" seams, such as J_{32} are different in that there are two $\bar{H}$ field values that control them. For J_{32} the identifying variables are

```
LOCFC(32) = ITRAN(21,14,12)
IDIRFC(32) = 2
IHVAL(32,1) = 1
IHVAL (32,2) = 0
IHVAL(32,3) = 1
IHVAL(32,4) = 0
```

On the other hand, locations along "interior" seams such as J_x always will have zero current since the four $\bar{H}$ field locations surrounding this location are identically zero. As a result, FCBLD will not establish a forced current at these locations.

For forced currents established along thin surfaces (e.g. NOPE = 1,2,...etc), testing procedure used by FCBLD to determine the locations of

these currents and their controlling $\overline{H}$ fields is more involved (since surfaces can have either $\hat{x}$, $\hat{y}$, or $\hat{z}$ normal directions), but is essentially the same. Such a current is depicted as J_{47} in Figure 7. As can be seen, this particular current is controlled by three (one on each side of the plate, plus the component adjacent to and normal to the plate). For such a case, the identifying variables are

```
LOCFC(47) = ITRAN(24,14,11)
IDIRFC(47) = 2
IHVAL(47,1) = 1
IHVAL(47,2) = 1
IHVAL(47,3) = 1
IHVAL(47,4) = 0
```

Finally, FCBLD checks to see if any forced surface current location coincides with the aperture plane indicated by the parameters LP and LPDIR. This test is performed at the end of the DO 1 loop. If indeed there is a coincidence, FCBLD will not establish a forced surface current location at these locations.

```
      SUBROUTINE FCBLD
C
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      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
      COMMON/WIRE/IA(50),IB(50),IDIR(400),MAPQI(400,6),MAPIQ(400,2),NI,
     1   NQ,NWRS,LOCQ(400),LOCI(400),DELS(400),CUR(400),Q(400),EWD(400),
     2   DIL(400),WRAD(50),SRAD(400),AIND(400),CAP(400),QFAC(400,2)
      COMMON/TSITEM/NOPE(29,29,29)
      DO 1 I=1,NX
      DO 1 J=1,NY
      DO 1 K=1,NZ
      DO 20 L=1,3
      ISTCR=0
      IF(L.NE.1) GO TO 30
      IF(NOPE(I,J,K-1).EQ.1.OR.NOPE(I,J,K).EQ.1.OR.NOPE(I,J-1,K).EQ.1
     <    .OR.NOPE(I,J-1,K-1).EQ.1) GO TO 23
      IF(NOPE(I,J,K-1).EQ.2.OR.NOPE(I,J,K).EQ.2.OR.NOPE(I,J-1,K).EQ.2
     <    .OR.NOPE(I,J-1,K-1).EQ.2) GO TO 24
      IF(NOPE(I,J,K-1).EQ.3.OR.NOPE(I,J,K).EQ.3.OR.NOPE(I,J-1,K).EQ.3
     <    .OR.NOPE(I,J-1,K-1).EQ.3) GO TO 25
      IADD=NOPE(I,J,K)+NOPE(I,J-1,K)+NOPE(I,J-1,K-1)+NOPE(I,J,K-1)
      IF(IADD.EQ.16.OR.IADD.EQ.0) GO TO 30
      IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) THEN
          ISTCR=1
          NFC=NFC+1
          IHVAL(NFC,3)=1
      END IF
      IF(NOPE(I,J-1,K).EQ.0.AND.NOPE(I,J-1,K-1).EQ.0) THEN
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,2)=1
      END IF
      IF(NOPE(I,J,K-1).EQ.0.AND.NOPE(I,J-1,K-1).EQ.0) THEN
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,4)=1
      END IF
      IF(NOPE(I,J,K-1).EQ.0.AND.NOPE(I,J,K).EQ.0) THEN
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,1)=1
      END IF
      IF(ISTCR.EQ.1) THEN
          LOCFC(NFC)=ITRAN(I,J,K)
          IDIRFC(NFC)=1
      END IF
      GO TO 21
   23 IADD=NOPE(I,J,K)+NOPE(I,J-1,K)+NOPE(I,J-1,K-1)+NOPE(I,J,K-1)
      IF(IADD.NE.9.AND.IADD.NE.10) GO TO 20
```

```
      IF(NOPE(I,J,K).NE.4.AND.NOPE(I,J-1,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,3)=1
          GO TO 26
      END IF
      IF(NOPE(I,J-1,K).NE.4.AND.NOPE(I,J-1,K-1).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,2)=1
          GO TO 26
      END IF
      IF(NOPE(I,J-1,K-1).NE.4.AND.NOPE(I,J,K-1).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,4)=1
          GO TO 26
      END IF
      IF(NOPE(I,J,K-1).NE.4.AND.NOPE(I,J,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,1)=1
          GO TO 26
      END IF
      GO TO 20
   24 IADD=NOPE(I,J,K)+NOPE(I,J-1,K)+NOPE(I,J-1,K-1)+NOPE(I,J,K-1)
      IF(IADD.EQ.10) THEN
          IF(NOPE(I,J,K).EQ.2.OR.NOPE(I,J,K-1).EQ.2) GO TO 20
          IF(NOPE(I,J-1,K).EQ.2) THEN
              NFC=NFC+1
              IHVAL(NFC,3)=1
              GO TO 26
          END IF
          IF(NOPE(I,J-1,K-1).EQ.2) THEN
              NFC=NFC+1
              IHVAL(NFC,4)=1
              GO TO 26
          END IF
      END IF
      IF(IADD.EQ.4) THEN
          IF(NOPE(I,J,K-1).EQ.0.AND.NOPE(I,J,K).EQ.0) GO TO 20
          NFC=NFC+1
          IHVAL(NFC,1)=1
          IHVAL(NFC,2)=1
          GO TO 26
      END IF
      IF(IADD.EQ.2) THEN
          IF(NOPE(I,J,K-1).EQ.0.AND.NOPE(I,J,K).EQ.0) GO TO 20
          IF(NOPE(I,J,K-1).EQ.2) THEN
              NFC=NFC+1
              IHVAL(NFC,1)=1
              IHVAL(NFC,2)=1
              IHVAL(NFC,3)=1
              GO TO 26
          END IF
          IF(NOPE(I,J,K).EQ.2) THEN
              NFC=NFC+1
              IHVAL(NFC,1)=1
              IHVAL(NFC,2)=1
              IHVAL(NFC,4)=1
              GO TO 26
          END IF
```

```
      END IF
      GO TO 20
   25 IADD=NOPE(I,J,K)+NOPE(I,J-1,K)+NOPE(I,J-1,K-1)+NOPE(I,J,K-1)
      IF(IADD.EQ.11) THEN
           IF(NOPE(I,J,K).EQ.3.OR.NOPE(I,J-1,K).EQ.3) GO TO 20
           IF(NOPE(I,J,K-1).EQ.3) THEN
                NFC=NFC+1
                IHVAL(NFC,3)=1
                GO TO 26
           END IF
           IF(NOPE(I,J-1,K-1).EQ.3) THEN
                NFC=NFC+1
                IHVAL(NFC,2)=1
                GO TO 26
           END IF
      END IF
      IF(IADD.EQ.6) THEN
           IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) GO TO 20
           NFC=NFC+1
           IHVAL(NFC,3)=1
           IHVAL(NFC,4)=1
           GO TO 26
      END IF
      IF(IADD.EQ.3) THEN
           IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) GO TO 20
           IF(NOPE(I,J,K).EQ.3) THEN
                NFC=NFC+1
                IHVAL(NFC,3)=1
                IHVAL(NFC,2)=1
                IHVAL(NFC,4)=1
                GO TO 26
           END IF
           IF(NOPE(I,J-1,K).EQ.3) THEN
                NFC=NFC+1
                IHVAL(NFC,3)=1
                IHVAL(NFC,4)=1
                IHVAL(NFC,1)=1
                GO TO 26
           END IF
      END IF
      GO TO 20
   26 LOCFC(NFC)=ITRAN(I,J,K)
      IDIRFC(NFC)=1
   30 IF(L.NE.2) GO TO 40
      IF(NOPE(I,J,K).EQ.1.OR.NOPE(I,J,K-1).EQ.1.OR.NOPE(I-1,J,K-1).EQ.1
     <     .OR.NOPE(I-1,J,K).EQ.1) GO TO 31
      IF(NOPE(I,J,K).EQ.2.OR.NOPE(I,J,K-1).EQ.2.OR.NOPE(I-1,J,K-1).EQ.2
     <     .OR.NOPE(I-1,J,K).EQ.2) GO TO 32
      IF(NOPE(I,J,K).EQ.3.OR.NOPE(I,J,K-1).EQ.3.OR.NOPE(I-1,J,K-1).EQ.3
     <     .OR.NOPE(I-1,J,K).EQ.3) GO TO 33
      IADD=NOPE(I,J,K)+NOPE(I,J,K-1)+NOPE(I-1,J,K-1)+NOPE(I-1,J,K)
      IF(IADD.EQ.16.OR.IADD.EQ.0) GO TO 40
      IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J,K-1).EQ.0) THEN
           ISTCR=1
           NFC=NFC+1
           IHVAL(NFC,3)=1
      END IF
      IF(NOPE(I,J,K-1).EQ.0.AND.NOPE(I-1,J,K-1).EQ.0) THEN
```

```
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,2)=1
      END IF
      IF(NOPE(I-1,J,K-1).EQ.0.AND.NOPE(I-1,J,K).EQ.0) THEN
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,4)=1
      END IF
      IF(NOPE(I-1,J,K).EQ.0.AND.NOPE(I,J,K).EQ.0) THEN
          IF(ISTCR.EQ.0) THEN
              ISTCR=1
              NFC=NFC+1
          END IF
          IHVAL(NFC,1)=1
      END IF
      IF(ISTCR.EQ.1) THEN
          LOCFC(NFC)=ITRAN(I,J,K)
          IDIRFC(NFC)=2
      END IF
      GO TO 21
   32 IADD=NOPE(I,J,K)+NOPE(I,J,K-1)+NOPE(I-1,J,K-1)+NOPE(I-1,J,K)
      IF(IADD.NE.10.AND.IADD.NE.12) GO TO 20
      IF(NOPE(I,J,K).NE.4.AND.NOPE(I,J,K-1).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,3)=1
          GO TO 36
      END IF
      IF(NOPE(I,J,K-1).NE.4.AND.NOPE(I-1,J,K-1).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,2)=1
          GO TO 36
      END IF
      IF(NOPE(I-1,J,K-1).NE.4.AND.NOPE(I-1,J,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,4)=1
          GO TO 36
      END IF
      IF(NOPE(I-1,J,K).NE.4.AND.NOPE(I,J,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,1)=1
          GO TO 36
      END IF
      GO TO 20
   31 IADD=NOPE(I,J,K)+NOPE(I,J,K-1)+NOPE(I-1,J,K-1)+NOPE(I-1,J,K)
      IF(IADD.EQ.9) THEN
          IF(NOPE(I,J,K).EQ.1.OR.NOPE(I,J,K-1).EQ.1) GO TO 20
          IF(NOPE(I-1,J,K).EQ.1) THEN
              NFC=NFC+1
              IHVAL(NFC,1)=1
              GO TO 36
          END IF
          IF(NOPE(I-1,J,K-1).EQ.1) THEN
              NFC=NFC+1
```

```
            IHVAL(NFC,2)=1
            GO TO 36
        END IF
    END IF
    IF(IADD.EQ.2) THEN
        IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J,K-1).EQ.0) GO TO 20
        NFC=NFC+1
        IHVAL(NFC,3)=1
        IHVAL(NFC,4)=1
        GO TO 36
    END IF
    IF(IADD.EQ.1) THEN
        IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J,K-1).EQ.0) GO TO 20
        IF(NOPE(I,J,K-1).EQ.1) THEN
            NFC=NFC+1
            IHVAL(NFC,4)=1
            IHVAL(NFC,2)=1
            IHVAL(NFC,3)=1
            GO TO 36
        END IF
        IF(NOPE(I,J,K).EQ.1) THEN
            NFC=NFC+1
            IHVAL(NFC,1)=1
            IHVAL(NFC,3)=1
            IHVAL(NFC,4)=1
            GO TO 36
        END IF
    END IF
    GO TO 20
33  IADD=NOPE(I,J,K)+NOPE(I,J,K-1)+NOPE(I-1,J,K-1)+NOPE(I-1,J,K)
    IF(IADD.EQ.11) THEN
        IF(NOPE(I,J,K).EQ.3.OR.NOPE(I-1,J,K).EQ.3) GO TO 20
        IF(NOPE(I,J,K-1).EQ.3) THEN
            NFC=NFC+1
            IHVAL(NFC,3)=1
            GO TO 36
        END IF
        IF(NOPE(I-1,J,K-1).EQ.3) THEN
            NFC=NFC+1
            IHVAL(NFC,4)=1
            GO TO 36
        END IF
    END IF
    IF(IADD.EQ.6) THEN
        IF(NOPE(I,J,K).EQ.0.AND.NOPE(I-1,J,K).EQ.0) GO TO 20
        NFC=NFC+1
        IHVAL(NFC,1)=1
        IHVAL(NFC,2)=1
        GO TO 36
    END IF
    IF(IADD.EQ.3) THEN
        IF(NOPE(I,J,K).EQ.0.AND.NOPE(I-1,J,K).EQ.0) GO TO 20
        IF(NOPE(I,J,K).EQ.3) THEN
            NFC=NFC+1
            IHVAL(NFC,1)=1
            IHVAL(NFC,2)=1
            IHVAL(NFC,4)=1
            GO TO 36
```

```
      END IF
      IF(NOPE(I-1,J,K).EQ.3) THEN
          NFC=NFC+1
          IHVAL(NFC,3)=1
          IHVAL(NFC,2)=1
          IHVAL(NFC,1)=1
          GO TO 36
      END IF
   END IF
   GO TO 20
36 LOCFC(NFC)=ITRAN(I,J,K)
   IDIRFC(NFC)=2
40 IF(L.NE.3) GO TO 20
   IF(NOPE(I,J,K).EQ.1.OR.NOPE(I-1,J,K).EQ.1.OR.NOPE(I-1,J-1,K).EQ.
  <    1.OR.NOPE(I,J-1,K).EQ.1) GO TO 41
   IF(NOPE(I,J,K).EQ.2.OR.NOPE(I-1,J,K).EQ.2.OR.NOPE(I-1,J-1,K).EQ.
  <    2.OR.NOPE(I,J-1,K).EQ.2) GO TO 42
   IF(NOPE(I,J,K).EQ.3.OR.NOPE(I-1,J,K).EQ.3.OR.NOPE(I-1,J-1,K).EQ.
  <    3.OR.NOPE(I,J-1,K).EQ.3) GO TO 43
   IADD=NOPE(I,J,K)+NOPE(I-1,J,K)+NOPE(I-1,J-1,K)+NOPE(I,J-1,K)
   IF(IADD.EQ.0.OR.IADD.EQ.16) GO TO 21
   IF(NOPE(I,J,K).EQ.0.AND.NOPE(I-1,J,K).EQ.0) THEN
       ISTCR=1
       NFC=NFC+1
       IHVAL(NFC,3)=1
   END IF
   IF(NOPE(I-1,J,K).EQ.0.AND.NOPE(I-1,J-1,K).EQ.0) THEN
       IF(ISTCR.EQ.0) THEN
           ISTCR=1
           NFC=NFC+1
       END IF
       IHVAL(NFC,2)=1
   END IF
   IF(NOPE(I-1,J-1,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) THEN
       IF(ISTCR.EQ.0) THEN
           ISTCR=1
           NFC=NFC+1
       END IF
       IHVAL(NFC,4)=1
   END IF
   IF(NOPE(I,J-1,K).EQ.0.AND.NOPE(I,J,K).EQ.0) THEN
       IF(ISTCR.EQ.0) THEN
           ISTCR=1
           NFC=NFC+1
       END IF
       IHVAL(NFC,1)=1
   END IF
   IF(ISTCR.EQ.1) THEN
       LOCFC(NFC)=ITRAN(I,J,K)
       IDIRFC(NFC)=3
   END IF
   GO TO 21
43 IADD=NOPE(I,J,K)+NOPE(I-1,J,K)+NOPE(I-1,J-1,K)+NOPE(I,J-1,K)
   IF(IADD.NE.11.AND.IADD.NE.14) GO TO 20
   IF(NOPE(I,J,K).NE.4.AND.NOPE(I-1,J,K).NE.4) THEN
       NFC=NFC+1
       IHVAL(NFC,3)=1
       GO TO 46
```

```
      END IF
      IF(NOPE(I-1,J,K).NE.4.AND.NOPE(I-1,J-1,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,2)=1
          GO TO 46
      END IF
      IF(NOPE(I-1,J-1,K).NE.4.AND.NOPE(I,J-1,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,4)=1
          GO TO 46
      END IF
      IF(NOPE(I,J-1,K).NE.4.AND.NOPE(I,J,K).NE.4) THEN
          NFC=NFC+1
          IHVAL(NFC,1)=1
          GO TO 46
      END IF
      GO TO 20
   41 IADD=NOPE(I,J,K)+NOPE(I-1,J,K)+NOPE(I-1,J-1,K)+NOPE(I,J-1,K)
      IF(IADD.EQ.9) THEN
          IF(NOPE(I,J,K).EQ.1.OR.NOPE(I-1,J,K).EQ.1) GO TO 20
          IF(NOPE(I-1,J,K).EQ.1) THEN
              NFC=NFC+1
              IHVAL(NFC,3)=1
              GO TO 46
          END IF
          IF(NOPE(I-1,J-1,K).EQ.1) THEN
              NFC=NFC+1
              IHVAL(NFC,4)=1
              GO TO 46
          END IF
      END IF
      IF(IADD.EQ.2) THEN
          IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) GO TO 20
          NFC=NFC+1
          IHVAL(NFC,1)=1
          IHVAL(NFC,2)=1
          GO TO 46
      END IF
      IF(IADD.EQ.1) THEN
          IF(NOPE(I,J,K).EQ.0.AND.NOPE(I,J-1,K).EQ.0) GO TO 20
          IF(NOPE(I,J-1,K).EQ.1) THEN
              NFC=NFC+1
              IHVAL(NFC,1)=1
              IHVAL(NFC,2)=1
              IHVAL(NFC,3)=1
              GO TO 46
          END IF
          IF(NOPE(I,J,K).EQ.1) THEN
              NFC=NFC+1
              IHVAL(NFC,1)=1
              IHVAL(NFC,2)=1
              IHVAL(NFC,4)=1
              GO TO 46
          END IF
      END IF
      GO TO 20
   42 IADD=NOPE(I,J,K)+NOPE(I-1,J,K)+NOPE(I-1,J-1,K)+NOPE(I,J-1,K)
      IF(IADD.EQ.10) THEN
```

```
      IF(NOPE(I,J,K).EQ.2.OR.NOPE(I-1,J,K).EQ.2) GO TO 20
      IF(NOPE(I,J-1,K).EQ.2) THEN
          NFC=NFC+1
          IHVAL(NFC,1)=1
          GO TO 46
      END IF
      IF(NOPE(I-1,J-1,K).EQ.2) THEN
          NFC=NFC+1
          IHVAL(NFC,2)=1
          GO TO 46
      END IF
   END IF
   IF(IADD.EQ.4) THEN
      IF(NOPE(I,J,K).EQ.0.AND.NOPE(I-1,J,K).EQ.0) GO TO 20
      NFC=NFC+1
      IHVAL(NFC,3)=1
      IHVAL(NFC,4)=1
      GO TO 46
   END IF
   IF(IADD.EQ.2) THEN
      IF(NOPE(I,J,K).EQ.0.AND.NOPE(I-1,J,K).EQ.0) GO TO 20
      IF(NOPE(I,J,K).EQ.2) THEN
          NFC=NFC+1
          IHVAL(NFC,3)=1
          IHVAL(NFC,2)=1
          IHVAL(NFC,4)=1
          GO TO 46
      END IF
      IF(NOPE(I-1,J,K).EQ.2) THEN
          NFC=NFC+1
          IHVAL(NFC,3)=1
          IHVAL(NFC,4)=1
          IHVAL(NFC,1)=1
          GO TO 46
      END IF
   END IF
   GO TO 20
46 LOCFC(NFC)=ITRAN(I,J,K)
   IDIRFC(NFC)=3
   GO TO 20
21 IF(ISTCR.EQ.1) THEN
      IADD=IHVAL(NFC,1)+IHVAL(NFC,2)+IHVAL(NFC,3)+IHVAL(NFC,4)
      IF(LPDIR.EQ.1.AND.LP.EQ.I.AND.IADD.EQ.1) GO TO 22
      IF(LPDIR.EQ.2.AND.LP.EQ.J.AND.IADD.EQ.1) GO TO 22
      IF(LPDIR.EQ.3.AND.LP.EQ.K.AND.IADD.EQ.1) GO TO 22
      GO TO 20
22    IHVAL(NFC,1)=0
      IHVAL(NFC,2)=0
      IHVAL(NFC,3)=0
      IHVAL(NFC,4)=0
      NFC=NFC-1
   END IF
20 CONTINUE
 1 CONTINUE
   RETURN
   END
```

Subroutine STOCUR

The purpose of subroutine STOCUR is to calculate and store the time histories of the currents on the external planar surfaces (external wire currents are handled by subroutine WIREADV) of the scatterer during the M=1 loop so that they can be used as forced current sources during the M=3 and 4 loops. STOCUR is called during each time cycle of the M=1 loop after the $\bar{H}$ fields are advanced by subroutine HADV.

STOCUR proceeds by marching through each forced surface location and calculating the surface current from Maxwell's equation:

$$\nabla \times \bar{H} = \bar{J} + \frac{\partial \bar{E}}{\partial t} \quad . \tag{20}$$

But since the electric field tangent to the conducting surface is zero, the current is determined just from the curl of $\bar{H}$.

When the curl of $\bar{H}$ is determined at the surface current location, it is important for STOCUR to know which of the four $\bar{H}$ field values normally used in the curl relationship should be used. The problem to be circumvented here is that when using externally applied fields (i.e. IDLS = 0, using EINCX), it is possible that there will be non-zero $\bar{H}$ fields inside the conducting surfaces. Although this does not effect the external calculations, it would be incorrect to blindly use all four $\bar{H}$ field components in the curl expression for $\bar{J}$. Thus, STOCUR uses the elements of the array IHVAL, which have values of either 0 or 1, as filters to make sure that only the correct $\bar{H}$ field values are used. The values of IHVAL are syncronized with the order in which the corresponding $\bar{H}$ field values are used within the curl operation.

The output of STOCUR is the array FCUR (IFC,N) where, IFC is the surface current element number and N is the time step number.

```
      SUBROUTINE STOCUR
C
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
      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,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      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/PERM/MM
      INTEGER SGN
C
      DO 1 IFC=1,NFC
      JP=LOCFC(IFC)
      I=IRTRAN(JP,1)
      J=IRTRAN(JP,2)
      K=IRTRAN(JP,3)
      IV1=IHVAL(IFC,1)
      IV2=IHVAL(IFC,2)
      IV3=IHVAL(IFC,3)
      IV4=IHVAL(IFC,4)
      IF(IDIRFC(IFC).EQ.1) THEN
          FCUR(IFC,N)=(HZ(I,J,K)*IV1-HZ(I,J-1,K)*IV2)*DYI(J)
     <               -(HY(I,J,K)*IV3-HY(I,J,K-1)*IV4)*DZI(K)
      ELSE IF(IDIRFC(IFC).EQ.2) THEN
          FCUR(IFC,N)=(HX(I,J,K)*IV1-HX(I,J,K-1)*IV2)*DZI(K)
     <               -(HZ(I,J,K)*IV3-HZ(I-1,J,K)*IV4)*DXI(I)
      ELSE
          FCUR(IFC,N)=(HY(I,J,K)*IV1-HY(I-1,J,K)*IV2)*DXI(I)
     <               -(HX(I,J,K)*IV3-HX(I,J-1,K)*IV4)*DYI(J)
      END IF
    1 CONTINUE
      RETURN
      END
```

Subroutine EADV

Subroutine EADV in THNAPP serves essentially the same purpose as it does in G3DXL except for the presence of forced currents during M=3 and M=4 runs and the forcing of aperture fields during M=5. This subroutine proceeds by first advancing the fields in each cell under the assumption that there are no forced currents present. After all of the fields have been advanced under this assumption, those cells that contain forced currents and/or forced aperture fields are dealt with separately. This procedure saves computational time since the number of cells that contain forced currents is small as compared to the total number of cells. Thus, it would significantly slow the algorithm down to test each cell for the presence of a forced current or field.

For cells that contain forced currents (during M=3 or 4), the differenced form of Maxwell's curl $\bar{H}$ equation for the m th component of $\bar{E}$ becomes

$$E_m^{n+1} = E_m^n + \frac{\Delta t}{\varepsilon}\left[\left(\nabla \times H^{n+1/2}\right)_m + J_m^{n+1/2}\right] \tag{21}$$

where the subscript m indicates the m^{th} component. Since all but the last term of this expression has already been taken into account in EADV, only the last term need be added to E_m in those cells that contain forced currents. This is performed by sequencing through all of the forced current locations and directions (using the arrays LOCFC and IDIRFC) and adding the above current term to the previously calculated $\bar{E}$ fields. The currents stored in the array FCUR (supplied by subroutine STOCUR for surface currents and WIREADV for wire currents) already have taken the cell sizes into account and thus have units of A/m.

During M=3, 4, and 5, the aperture fields are either calculated (M=3 and 4), or forced (M=5). The number of cells containing the apertures is contained in the variable NLOC, and the positions and orientations of the aperture fields are contained in the arrays LOCE and IDIRE, respectively (see the section on Subroutine BUILD for the rules for specifying these variables).

During M=3 and 4, the aperture fields are (automatically) calculated according the formula

$$E_{App} = E_{tan}^{m=3} - E_{tan}^{m=4} \tag{22}$$

and stored in the array EAPP. Note that equation 22 is the computer version of equation 5 for the case where it is assumed that all of the forced currents are on the same side of the aperture plane as is the cavity and the fields are evaluated in the limit as $z \to 0^+$. It is also assumed here that the magnetic wire used during the M=4 calculation was $\frac{1}{2}$ cell inside the aperture (i.e., below the surface).

The way that EADV forces the aperture fields during M=5 depends on the value of M5EXP. For M5EXP=0, the aperture locations are the same as they were in M=3 and 4 and thus the aperture fields are forced directly in EADV. Note that it is important that the cells specified here are in exactly the same plane as is the aperture.

When an expanded grid is used (M5EXP = 1), Subroutine OUTBND is called to force the correct aperture fields since the aperture fields must be interpolated in both space and time. For this case, SAVESB is called during each cycle to store the tangential $\bar{E}$ fields along the expanded cell subspace. Obviously, care must be taken to make sure that one of the expansion grid sides is coincident with the aperture plane in order for this process to work correctly. (See also the description of BUILD.)

```
      SUBROUTINE EADV
C
      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,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,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),DYI(29),DZI(29),DX0I(28),DY0I(28),DZ0I(28)
      COMMON/EBS/EYXD(29,29),EYXU(29,29),EZXD(29,29),EZXU(29,29)
     1    ,EXYD(29,29),EXYU(29,29),EZYD(29,29),EZYU(29,29),
     2    EXZD(29,29),EXZU(29,29),EYZD(29,29),EYZU(29,29),
     3    N1,N2,N3,EYXDD(29,29),EYXUU(29,29),EZXDD(29,29),
     4    EXYDD(29,29),EXYUU(29,29),EZYDD(29,29),EZYUU(29,29),
     5    EXZDD(29,29),EXZUU(29,29),EYZDD(29,29),EYZUU(29,29),
     6    EZXUU(29,29),CX(2),CY(2),CZ(2)
      COMMON/WIRE/IA(50),IB(50),IDIR(400),MAPQI(400,6),MAPIQ(400,2),NI,
     1    NQ,NWRS,LOCQ(400),LOCI(400),DELS(400),CUR(400),Q(400),EWD(400),
     2    DIL(400),WRAD(50),SRAD(400),AIND(400),CAP(400),QFAC(400,2)
      COMMON/TSITEM/NOPE(29,29,29)
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
C
      DTE=DT/EPSO
C    ADVANCE EX
C
      DO 1 I = 1,NX1
      DO 1 J = 2,NY1
      DO 1 K = 2,NZ1
      IF(NOPE(I,J,K).EQ.4) GO TO 1
      EX(I,J,K)=EX(I,J,K)+DTE*((HZ(I,J,K)-HZ(I,J-1,K))*DYI(J)
     1   -(HY(I,J,K)-HY(I,J,K-1))*DZI(K))
    1 CONTINUE
C
C    ADVANCE EY
C
      DO 2 I = 2,NX1
      DO 2 J = 1,NY1
      DO 2 K = 2,NZ1
      IF(NOPE(I,J,K).EQ.4) GO TO 2
      EY(I,J,K)=EY(I,J,K)+DTE*((HX(I,J,K)-HX(I,J,K-1))*DZI(K)
     1   -(HZ(I,J,K)-HZ(I-1,J,K))*DXI(I))
    2 CONTINUE
C
C    ADVANCE EZ
C
      DO 3 I = 2,NX1
      DO 3 J = 2,NY1
      DO 3 K = 1,NZ1
      IF(NOPE(I,J,K).EQ.4) 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,J-1,K))*DYI(J))
    3 CONTINUE
      IF(NWRS .NE. 0)CALL WIREADV
```

```
      IF (M.EQ.2.OR.M.EQ.5) GO TO 10
      IF(N2.NE.N3)CALL ABSORB
   10 CONTINUE
      IF(M.EQ.2.OR.(M.EQ.5.AND.M5EXP.EQ.1)) THEN
          CALL OUTBND
          CALL EBC
          RETURN
      END IF
      CALL EBC
      IF(IBAB.EQ.1 .AND. M.GT.2 .AND. M.LT.5) THEN
          DO 4 IFC=1,NFC+NFCT
          JP=LOCFC(IFC)
          I=IRTRAN(JP,1)
          J=IRTRAN(JP,2)
          K=IRTRAN(JP,3)
          IIDIR=IDIRFC(IFC)
          IF(IIDIR .EQ. 1)EX(I,J,K)=EX(I,J,K)-DTE*FCUR(IFC,N)
          IF(IIDIR .EQ. 2)EY(I,J,K)=EY(I,J,K)-DTE*FCUR(IFC,N)
          IF(IIDIR .EQ. 3)EZ(I,J,K)=EZ(I,J,K)-DTE*FCUR(IFC,N)
    4     CONTINUE
      END IF
      DO 20 L=1,NLOC
      I=IRTRAN(LOCE(L),1)
      J=IRTRAN(LOCE(L),2)
      K=IRTRAN(LOCE(L),3)
      IF(M.NE.3) GO TO 45
      IF(IDIRE.EQ.1) THEN
          EAPP(N,L)=EX(I,J,K)
      ELSE IF(IDIRE.EQ.2) THEN
          EAPP(N,L)=EY(I,J,K)
      ELSE
          EAPP(N,L)=EZ(I,J,K)
      END IF
   45 CONTINUE
      IF(M.NE.4) GO TO 85
      IF(IDIRE.EQ.1) THEN
          EAPP(N,L)=EAPP(N,L)-EX(I,J,K)
      ELSE IF(IDIRE.EQ.2) THEN
          EAPP(N,L)=EAPP(N,L)-EY(I,J,K)
      ELSE
          EAPP(N,L)=EAPP(N,L)-EZ(I,J,K)
      END IF
   85 CONTINUE
      IF(M.NE.5.OR.M5EXP.EQ.1) GO TO 115
      IF(IDIRE.EQ.1) THEN
          EX(I,J,K)=EAPP(N,L)
      ELSE IF(IDIRE.EQ.2) THEN
          EY(I,J,K)=EAPP(N,L)
      ELSE
          EZ(I,J,K)=EAPP(N,L)
      END IF
  115 CONTINUE
   20 CONTINUE
      N3=N2
      N2=N1
```

```
      N1=N
C
C
C   ADVANCE EXY
C
      DO 11 I = 1,NX
      DO 11 K = 1,NZ
      EXYD(I,K) = EX(I,2,K)
      EXYU(I,K) = EX(I,NY1,K)
      EXYDD(I,K)=EX(I,1,K)
      EXYUU(I,K)=EX(I,NY,K)
   11 CONTINUE
C
C   ADVANCE EXZ
C
      DO 111 I = 1,NX
      DO 111 J = 1,NY
      EXZD(I,J) = EX(I,J,2)
      EXZU(I,J) = EX(I,J,NZ1)
      EXZDD(I,J)=EX(I,J,1)
      EXZUU(I,J)=EX(I,J,NZ)
  111 CONTINUE
C
C   ADVANCE EYX
C
      DO 22 J = 1,NY
      DO 22 K = 1,NZ
      EYXD(J,K) = EY(2,J,K)
      EYXU(J,K) = EY(NX1,J,K)
      EYXDD(J,K)=EY(1,J,K)
      EYXUU(J,K)=EY(NX,J,K)
   22 CONTINUE
C
C   ADVANCE EYZ
C
      DO 222 I = 1,NX
      DO 222 J = 1,NY
      EYZD(I,J) = EY(I,J,2)
      EYZU(I,J) = EY(I,J,NZ1)
      EYZDD(I,J)=EY(I,J,1)
      EYZUU(I,J)=EY(I,J,NZ)
  222 CONTINUE
C
C
C   ADVANCE EZX
C
      DO 33 J = 1,NY
      DO 33 K = 1,NZ
      EZXD(J,K) = EZ(2,J,K)
      EZXU(J,K) = EZ(NX1,J,K)
      EZXDD(J,K)=EZ(1,J,K)
      EZXUU(J,K)=EZ(NX,J,K)
   33 CONTINUE
C
C   ADVANCE EZY
```

```
C
      DO 333 I = 1,NX
      DO 333 K = 1,NZ
      EZYD(I,K) = EZ(I,2,K)
      EZYU(I,K) = EZ(I,NY1,K)
      EZYDD(I,K)=EZ(I,1,K)
      EZYUU(I,K)=EZ(I,NY,K)
  333 CONTINUE
C
      RETURN
      PRINT 666
  666 FORMAT(*EXIT EADV*)
      END
```

Subroutine ABSORB

THNAPP uses a different method than does G3DXL of estimating the tangential electric fields on the outer boundary of the problem space. Whereas G3DXL uses the so called "radiating boundary condition" and implements this technique in the subroutine ERAD, THNAPP utilizes an improved technique developed by Mur [14], which is often called the "absorbing boundary condition" or the "Mur condition". This condition is imposed in subroutine ABSORB, which essentially replaces ERAD.

Basically, the Mur approach is to assume that the fields in the vicinity of the problem space outer boundaries are behaving roughly as a plane wave. Given this assumption, the Mur approach proceeds to estimate the apparent direction of these plane waves at each point along the boundary by observing the fields calculated just within the problem space. Once this direction has been determined, an estimate of the fields just beyond the boundary can be estimated by assuming that the fields progress beyond the boundary as a plane wave. As an example, if we consider the x=0 planar boundary of the problem space, Mur's first approximation to the absorbing boundary condition can be written as

$$\left(\frac{\partial}{\partial x} - \frac{1}{c} \right) E_z \Bigg|_{x = o} = 0 \ , \qquad (23)$$

where c is the speed of light. Note that it has been assumed that the waves incident on the boundary are propagating in the -x direction. Equation 1 can then be expressed in different form in a way consistent with the FDTD approach. Evaluating the spatial derivative at position $x = \frac{1}{2}\Delta x$, $y = (J-1)\Delta y$, $z = (k-1)\Delta z$, and the temporal derivative at time $t = (n+1)st$ yields (where Δx, Δy, and Δz, are the cell widths, and st is the time step):

$$\frac{1}{\Delta x}\left[\left\{E_z^{n+3/2}(2,J,K) - E_z^{n+3/2}(1,J,K)\right\} + \left\{E_z^{n+\frac{1}{2}}(2,J,K) - E_z^{n+\frac{1}{2}}(1,J,K)\right\}\right]$$
$$- \frac{1}{c\Delta t}\left[\left\{E_z^{n+3/2}(2,J,K) - E_z^{n+\frac{1}{2}}(2,J,K)\right\} + \left\{E_z^{n+3/2}(1,J,K) - E_z^{n+\frac{1}{2}}(1,J,K)\right\}\right] = 0 \qquad (24)$$

Note that the time and space indexing used here are the same as that used throughout THNAPP (and G3DXL). Similar analysis can be used to obtain formulas that allow the tangential electric fields at all of the boundary

surfaces to be estimated using the Mur technique.

Equation 24 can be solved to obtain an estimate of the boundary field to obtain

$$E_z^{n+3/2}(1,J,K) = E_z^{n+1/2}(2,J,K) + \frac{c\Delta t - \Delta x}{c\Delta t + \Delta x}\left\{E_z^{n+3/2}(2,J,K) - E_z^{n+1/2}(1,J,K)\right\} \qquad (25)$$

As can be seen from equation (3), the use of this technique demands that the fields along the boundaries, as well as fields one cell away from these boundaries be known. These field quantities are stored by subroutine EADV as they are calculated for use in ABSORB in the arrays with variable names that begin with EXY, EXZ, and EYZ (indicating the plane on which they exist), followed with suffixes of D, DD, V, and VV (indicating that they are either on or within one cell of the boundary respectively.) Using the appropriate variable names, equation 24 becomes

$$EZ\ (1,J,K) = EZXD\ (J,K) + CX\ (1) * \left[EZ\ (2,J,K) - EZXDD\ (J,K)\right] \qquad (26)$$

Subroutine Absorb is called by subroutine EADV after the fields within the problem space have been advanced. The constants that involve the time step, the cell dimensions and the speed of light (for example, see equation 2) are calculated in subroutine SETUP (under the variable names CX(1), CX(2), and CX(3)), for use in ABSORB.

```
      SUBROUTINE ABSORB
C
      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),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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      COMMON/EBS/EYXD(29,29),EYXU(29,29),EZXD(29,29),EZXU(29,29)
     1   ,EXYD(29,29),EXYU(29,29),EZYD(29,29),EZYU(29,29),
     2   EXZD(29,29),EXZU(29,29),EYZD(29,29),EYZU(29,29),
     3   N1,N2,N3,EYXDD(29,29),EYXUU(29,29),EZXDD(29,29),
     4   EXYDD(29,29),EXYUU(29,29),EZYDD(29,29),EZYUU(29,29),
     5   EXZDD(29,29),EXZUU(29,29),EYZDD(29,29),EYZUU(29,29),
     6   EZXUU(29,29),CX(2),CY(2),CZ(2)
C
C     ADVANCE EZX
C
      DO 11 J=2,NY1
      DO 11 K=1,NZ1
  110 EZ(1,J,K)=EZXD(J,K)+CX(1)*(EZ(2,J,K)-EZXDD(J,K))
      EZ(NX,J,K)=EZXU(J,K)+CX(2)*(EZ(NX1,J,K)-EZXUU(J,K))
   11 CONTINUE
C
C     ADVANCE EYX
C
      DO 12 J=1,NY1
      DO 12 K=2,NZ1
  120 EY(1,J,K)=EYXD(J,K)+CX(1)*(EY(2,J,K)-EYXDD(J,K))
      EY(NX,J,K)=EYXU(J,K)+CX(2)*(EY(NX1,J,K)-EYXUU(J,K))
   12 CONTINUE
C
C     ADVANCE EZY
C
      DO 13 I=2,NX1
      DO 13 K=1,NZ1
  130 EZ(I,1,K)=EZYD(I,K)+CY(1)*(EZ(I,2,K)-EZYDD(I,K))
      EZ(I,NY,K)=EZYU(I,K)+CY(2)*(EZ(I,NY1,K)-EZYUU(I,K))
   13 CONTINUE
C
C     ADVANCE EXY
C
      DO 14 I=1,NX1
      DO 14 K=2,NZ1
  140 EX(I,1,K)=EXYD(I,K)+CY(1)*(EX(I,2,K)-EXYDD(I,K))
      EX(I,NY,K)=EXYU(I,K)+CY(2)*(EX(I,NY1,K)-EXYUU(I,K))
   14 CONTINUE
C
C     ADVANCE EXZ
C
      DO 15 I=1,NX1
      DO 15 J=2,NY1
  150 EX(I,J,1)=EXZD(I,J)+CZ(1)*(EX(I,J,2)-EXZDD(I,J))
```

```
      EX(I,J,NZ)=EXZU(I,J)+CZ(2)*(EX(I,J,NZ1)-EXZUU(I,J))
   15 CONTINUE
C
C     ADVANCE EYZ
C
      DO 16 I=2,NX1
      DO 16 J=1,NY1
  160 EY(I,J,1)=EYZD(I,J)+CZ(1)*(EY(I,J,2)-EYZDD(I,J))
      EY(I,J,NZ)=EYZU(I,J)+CZ(2)*(EY(I,J,NZ1)-EYZUU(I,J))
   16 CONTINUE
      RETURN
      END
```

Subroutine SAVESB

The role of SAVESB in THNAPP is an extension of what it is in G3DXL. As in G3DXL, its purpose is to store the tangential electric fields along the predefined sub-boundary during a loop that uses an unexpanded grid, for use in a subsequent expanded run. Subroutine OUTBND is called during that subsequent run to interpolate the fields in both time and space. However, THNAPP demands that SAVESB work in two situations. The first is when using an MM=12 run in which a scatterer (that does not contain any thin apertures) is to be analyzed first using a course grid (M=1), and then re-analyzed using an expanded grid (M=2). This is the case handled in G3DXL. The second is when using an MM=1345 run and the M=5 loop is to use an expanded grid.

In either of the two situations mentioned above, SAVESB is called by the main program DRIVER after EADV is called. The way in which SAVESB operates is determined by the value of M. If M=1, SAVESB proceeds, as in G3DXL, to store all tangential electric field components along the sub-boundary defined by the parameters INEAR, IFAR, JNEAR, JFAR, KNEAR, KFAR. This portion of the subroutine occurs below statement 30.

If M=4, SAVESB assumes that one of the sub-boundary surfaces lies along the aperture plane. Because of this, it would be wasteful for SAVESB to march through all 864 tangential field locations along the sub-boundary (for EXPFAC = 4.0) since the only nonzero electric fields along the sub-boundary will occur in the aperture. Thus, SAVESB fills only those elements of the array ARAY that correspond to the aperture locations. The determinations of which sub-boundary surface contains the aperture(s) is made by comparing the values of INEAR, ..., KFAR (specified in the input file) with LPDIR and LP (specified in BUILD). The exact locations of the array ARAY that receive the aperture field values are then determined from the aperture orientation (IDIRE) and spatial location(s) (LOCE) of the aperture.

```
      SUBROUTINE SAVESB
C
      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,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
C
      COMMON/RAY/ARAY(864,1000)
      COMMON/OUTLIST/DELX,DELY,DELZ,XPANX,XPANY,XPANZ,
     1    IUP,JUP,KUP,IDOWN,JDOWN,KDOWN,INEAR,JNEAR,KNEAR,
     2    IFAR,JFAR,KFAR,XOBS(6),YOBS(6),ZOBS(6),TEST,
     3    NPLANE(6)
      COMMON/FORCE/IBAB,NFC,NFCT,FCUR(1250,1000),LOCFC(1250)
     1  ,IHVAL(1250,4),IDIRFC(1250),LPDIR,LP
     1  ,EAPP(1000,100),LOCE(100),IDIRE,NLOC,M5EXP
C
C     THIS SUBROUTINE SAVES THE TANGENTIAL E FIELD COMPONENTS ON THE
C     SUBBOUNDARY
C
      EXPFAC=4.0
      LIMIT=28/EXPFAC
      L=1
      IF(N.NE.1) GO TO 10
      IT=1
      PRINT 5
    5 FORMAT(*SAVESB CALLED*)
   10 CONTINUE
C
      IMIN=INEAR
      JMIN=JNEAR
      KMIN=KNEAR
C
C
      IMAX=IMIN+LIMIT
      JMAX=JMIN+LIMIT
      KMAX=KMIN+LIMIT
C
      IL=IMIN-1
      JL=JMIN-1
      KL=KMIN-1
C
      IF(M.EQ.1) GO TO 30
      IF(LPDIR.NE.1) GO TO 20
      IF(IDIRE.EQ.2) THEN
          IF(LP.EQ.INEAR) THEN
              L=1
              GO TO 85
          END IF
          IF(LP.EQ.IFAR) THEN
              L=73
              GO TO 95
          END IF
      END IF
      IF(IDIRE.EQ.3) THEN
          IF(LP.EQ.INEAR) THEN
              L=145
              GO TO 115
```

```
          END IF
          IF(LP.EQ.IFAR) THEN
              L=217
              GO TO 135
          END IF
      END IF
   20 IF(LPDIR.NE.3) GO TO 25
      IF(IDIRE.EQ.1) THEN
          IF(LP.EQ.KNEAR) THEN
              L=289
              GO TO 155
          END IF
          IF(LP.EQ.KFAR) THEN
              L=361
              GO TO 175
          END IF
      END IF
      IF(IDIRE.EQ.2) THEN
          IF(LP.EQ.KNEAR) THEN
              L=433
              GO TO 195
          END IF
          IF(LP.EQ.KFAR) THEN
              L=505
              GO TO 215
          END IF
      END IF
   25 IF(LPDIR.NE.2) GO TO 30
      IF(IDIRE.EQ.1) THEN
          IF(LP.EQ.JNEAR) THEN
              L=577
              GO TO 235
          END IF
          IF(LP.EQ.JFAR) THEN
              L=649
              GO TO 255
          END IF
      END IF
      IF(IDIRE.EQ.3) THEN
          IF(LP.EQ.JNEAR) THEN
              L=721
              GO TO 275
          END IF
          IF(LP.EQ.JFAR) THEN
              L=793
              GO TO 295
          END IF
      END IF
C
C     EY ON INEAR,J,K SIDE
C
   85 CONTINUE
      DO 90 J=JL,JMAX
      DO 90 K=KMIN,KMAX
      DO 91 LL=1,NLOC
      IF(J.NE.IRTRAN(LOCE(LL),2).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 91
```

```
      ARAY(L,IT)=EAPP(N,LL)
   91 CONTINUE
      L=L+1
   90 CONTINUE
      GO TO 320
C
C     EY ON IFAR,J,K SIDE
C
   95 CONTINUE
      DO 110 J=JL,JMAX
      DO 110 K=KMIN,KMAX
      DO 111 LL=1,NLOC
      IF(J.NE.IRTRAN(LOCE(LL),2).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 111
      ARAY(L,IT)=EAPP(N,LL)
  111 CONTINUE
      L=L+1
  110 CONTINUE
      GO TO 320
  115 CONTINUE
C
C     EZ ON INEAR,J,K SIDE
C
      DO 130 J=JMIN,JMAX
      DO 130 K=KL,KMAX
      DO 131 LL=1,NLOC
      IF(J.NE.IRTRAN(LOCE(LL),2).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 131
      ARAY(L,IT)=EAPP(N,LL)
  131 CONTINUE
      L=L+1
  130 CONTINUE
      GO TO 320
  135 CONTINUE
C
C     EZ ON IFAR,J,K SIDE
C
      DO 150 J=JMIN,JMAX
      DO 150 K=KL,KMAX
      DO 151 LL=1,NLOC
      IF(J.NE.IRTRAN(LOCE(LL),2).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 151
      ARAY(L,IT)=EAPP(N,LL)
  151 CONTINUE
      L=L+1
  150 CONTINUE
      GO TO 320
  155 CONTINUE
C
C     EX ON I,J,KNEAR SIDE
C
C
      DO 170 I=IL,IMAX
      DO 170 J=JMIN,JMAX
      DO 171 LL=I,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.J.NE.IRTRAN(LOCE(LL),2)) GO TO 171
      ARAY(L,IT)=EAPP(N,LL)
  171 CONTINUE
      L=L+1
```

```
  170 CONTINUE
      GO TO 320
  175 CONTINUE
C
C     EX ON I,J,KFAR SIDE
C
      DO 190 I=IL,IMAX
      DO 190 J=JMIN,JMAX
      DO 191 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.J.NE.IRTRAN(LOCE(LL),2)) GO TO 191
      ARAY(L,IT)=EAPP(N,LL)
  191 CONTINUE
      L=L+1
  190 CONTINUE
      GO TO 320
  195 CONTINUE
C
C     EY ON I,J,KNEAR SIDE
C
C
      DO 210 I=IMIN,IMAX
      DO 210 J=JL,JMAX
      DO 211 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.J.NE.IRTRAN(LOCE(LL),2)) GO TO 211
      ARAY(L,IT)=EAPP(N,LL)
  211 CONTINUE
      L=L+1
  210 CONTINUE
      GO TO 320
  215 CONTINUE
C
C     EY ON I,J,KFAR SIDE
C
      DO 230 I=IMIN,IMAX
      DO 230 J=JL,JMAX
      DO 231 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.J.NE.IRTRAN(LOCE(LL),2)) GO TO 231
      ARAY(L,IT)=EAPP(N,LL)
  231 CONTINUE
      L=L+1
  230 CONTINUE
      GO TO 320
  235 CONTINUE
C
C     EX ON I,JNEAR,K SIDE
C
C
      DO 250 I=IL,IMAX
      DO 250 K=KMIN,KMAX
      DO 251 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 251
      ARAY(L,IT)=EAPP(N,LL)
  251 CONTINUE
      L=L+1
  250 CONTINUE
      GO TO 320
```

```
  255 CONTINUE
C
C     EX ON I,JFAR,K SIDE
C
      DO 270 I=IL,IMAX
      DO 270 K=KMIN,KMAX
      DO 271 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 271
      ARAY(L,IT)=EAPP(N,LL)
  271 CONTINUE
      L=L+1
  270 CONTINUE
      GO TO 320
  275 CONTINUE
C
C     EZ ON I,JNEAR,K SIDE
C
      DO 290 I=IMIN,IMAX
      DO 290 K=KL,KMAX
      DO 291 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 291
      ARAY(L,IT)=EAPP(N,LL)
  291 CONTINUE
      L=L+1
  290 CONTINUE
      GO TO 320
  295 CONTINUE
C
C   C     EZ ON I,JFAR,K SIDE
C
      DO 310 I=IMIN,IMAX
      DO 310 K=KL,KMAX
      DO 311 LL=1,NLOC
      IF(I.NE.IRTRAN(LOCE(LL),1).OR.K.NE.IRTRAN(LOCE(LL),3)) GO TO 311
      ARAY(L,IT)=EAPP(N,LL)
  311 CONTINUE
      L=L+1
  310 CONTINUE
      GO TO 320
C
C     SAVES BOUNDARY FOR M=2 EXPANDED RUN
C
   30 CONTINUE
      DO 92 J=JL,JMAX
      DO 92 K=KMIN,KMAX
      ARAY(L,IT)=EY(INEAR,J,K)
      L=L+1
   92 CONTINUE
      DO 112 J=JL,JMAX
      DO 112 K=KMIN,KMAX
      ARAY(L,IT)=EY(IFAR,J,K)
      L=L+1
  112 CONTINUE
      DO 132 J=JMIN,JMAX
      DO 132 K=KL,KMAX
      ARAY(L,IT)=EZ(INEAR,J,K)
```

```
      L=L+1
  132 CONTINUE
      DO 152 J=JMIN,JMAX
      DO 152 K=KL,KMAX
      ARAY(L,IT)=EZ(IFAR,J,K)
      L=L+1
  152 CONTINUE
      DO 172 I=IL,IMAX
      DO 172 J=JMIN,JMAX
      ARAY(L,IT)=EX(I,J,KNEAR)
      L=L+1
  172 CONTINUE
      DO 192 I=IL,IMAX
      DO 192 J=JMIN,JMAX
      ARAY(L,IT)=EX(I,J,KFAR)
      L=L+1
  192 CONTINUE
      DO 212 I=IMIN,IMAX
      DO 212 J=JL,JMAX
      ARAY(L,IT)=EY(I,J,KNEAR)
      L=L+1
  212 CONTINUE
      DO 232 I=IMIN,IMAX
      DO 232 J=JL,JMAX
      ARAY(L,IT)=EY(I,J,KFAR)
      L=L+1
  232 CONTINUE
      DO 252 I=IL,IMAX
      DO 252 K=KMIN,KMAX
      ARAY(L,IT)=EX(I,JNEAR,K)
      L=L+1
  252 CONTINUE
      DO 272 I=IMIN,IMAX
      DO 272 K=KL,KMAX
      ARAY(L,IT)=EX(I,JFAR,K)
      L=L+1
  272 CONTINUE
      DO 292 I=IMIN,IMAX
      DO 292 K=KL,KMAX
      ARAY(L,IT)=EZ(I,JNEAR,K)
      L=L+1
  292 CONTINUE
      DO 312 I=IMIN,IMAX
      DO 312 K=KL,KMAX
      ARAY(L,IT)=EZ(I,JFAR,K)
      L=L+1
  312 CONTINUE
C
C
  320 CONTINUE
  400 IT=IT+1
      RETURN
      END
```

Subroutine HBC

The purpose of subroutine HBC is similar in a dual sense to that of subroutine EBC. When magnetically conducting surfaces are defined in the NOPE array, HBC zeroes the $\bar{H}$ field components that are tangent to these surfaces or internal to them.

As indicated in the description of subroutine BUILD, magnetic surfaces and solids can be specified using the same rules that apply for electric surfaces and solids, except that negative values are used in the NOPE array. One more difference is that the magnetic surfaces lie along the cell centers, rather than their edges. This is because the $\bar{H}$ fields are evaluated along the centers of the cell faces, rather than along the lattice lines as is the case for electric fields.

Due to the similarities in declaring electric and magnetic surfaces, the operation of HBC is very similar to that of EBC.

```
      SUBROUTINE HBC
C
      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),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,M,MQ,DT,XMU,EPSO,EPS,NPTLIM,
     1 NN,NPTS,LMAX,SIGMA,C,T,PI,EXPFAC,IP,TX,TY,TZ,AMP,ALPHA,BETA,IDLS
      COMMON/TSITEM/NOPE(29,29,29)
C
C
      IF(M.NE.4)RETURN
C
      DO 500 I=1,NX1
      DO 500 J=1,NY1
      DO 500 K=1,NZ1
C
      IF(NOPE(I,J,K).EQ.0) GO TO 499
C
      IF(NOPE(I,J,K).EQ.-4) GO TO 400
C
C     FOR SEAMS(M=4) AND EXPANDED SEAMS(M=5) SET
C     N*H(SCAT)=0 FOR SURFACES BETWEEN SEAMS, SET
C     E=-EINC FOR SEAMS AND DO NOT SET H(SCAT) ADJACENT
C     TO A SEAM
C
      IF(NOPE(I,J,K).NE.-1.AND.NOPE(I,J,K).NE.-12.AND.
     1 NOPE(I,J,K).NE.-13.AND.NOPE(I,J,K).NE.-123)GO TO 70
C
C     Y-Z PLANE
C
      HY(I,J+1,K+1)=-HINCY(I,J+1,K+1)
      HY(I,J+1,K)=-HINCY(I,J+1,K)
      HZ(I,J+1,K+1)=-HINCZ(I,J+1,K+1)
      HZ(I,J,K+1)=-HINCZ(I,J,K+1)
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
C
      GO TO 500
C
   70 IF(NOPE(I,J,K).NE.-2.AND.NOPE(I,J,K).NE.-23.AND.
     1 NOPE(I,J,K).NE.-12.AND.NOPE(I,J,K).NE.-123)
     2 GO TO 80
C
C     X-Z PLANE
C
      HX(I+1,J,K)=-HINCX(I+1,J,K)
      HX(I,J,K)=-HINCX(I,J,K)
      HZ(I,J,K+1)=-HINCZ(I,J,K+1)
      HZ(I,J,K)=-HINCZ(I,J,K)
C
      IF(NOPE(I,J,K).EQ.-23.OR.NOPE(I,J,K).EQ.-123) GO TO 80
```

```
C
      GO TO 500
C
   80 CONTINUE
C
C     X-Y PLANE
C
      HX(I+1,J+1,K)=-HINCX(I+1,J+1,K)
      HX(I+1,J,K)=-HINCX(I+1,J,K)
      HY(I+1,J+1,K)=-HINCY(I+1,J+1,K)
      HY(I,J+1,K)=-HINCY(I,J+1,K)
C
      GO TO 500
C
  400 CONTINUE
C
C     SOLID REGION
C
      HX(I+1,J,K)=-HINCX(I+1,J,K)
      HX(I+1,J+1,K)=-HINCX(I+1,J+1,K)
      HX(I+1,J,K+1)=-HINCX(I+1,J,K+1)
      HX(I+1,J+1,K+1)=-HINCX(I+1,J+1,K+1)
      HY(I,J+1,K)=-HINCY(I,J+1,K)
      HY(I+1,J+1,K)=-HINCY(I+1,J+1,K)
      HY(I,J+1,K+1)=-HINCY(I,J+1,K+1)
      HY(I+1,J+1,K+1)=-HINCY(I+1,J+1,K+1)
      HZ(I,J,K+1)=-HINCZ(I,J,K+1)
      HZ(I+1,J,K+1)=-HINCZ(I+1,J,K+1)
      HZ(I,J+1,K+1)=-HINCZ(I,J+1,K+1)
      HZ(I+1,J+1,K+1)=-HINCZ(I+1,J+1,K+1)
C
  499 CONTINUE
C
  500 CONTINUE
C
      RETURN
      END
```

Function ITRAN

The purpose of ITRAN is to produce a single coded number that indicates the I, J, and K coordinates of a cell. Since only one number is needed to indicate the n-tuple (I, J, K), computer memory can be conserved.

The formula by which this number is produced is

$$ITRAN = 10000*I + 100*J + K \qquad (27)$$

Thus, the number ITRAN is simply the numbers I, J, and K written in sequence as if they were one number.

The function ITRAN need not be changed if the dimension of the numerical grid is changed from the present (28, 28, 28), as long as the number of cells in any one dimension does not exceed 99.

```
      FUNCTION ITRAN(I,J,K)
C
      ITRAN=10000*I+100*J+K
      RETURN
      END
C
```

Function IRTRAN

The purpose of function IRTRAN is to "decode" the coded position numbers that have been produced by function ITRAN. Thus, IRTRAN returns the I, J, and K values for a particular cell.

IRTRAN is called by the statement IRTRAN (IVAL, N). IVAL is the coded number to be decoded. N is the index that tells IRTRAN which index (I, J, or K) is to be returned. N can equal 1, 2, or 3, corresponding to I, J, or K.

```
      FUNCTION IRTRAN(IVAL,N)
C
      IF(N-2)1,2,3
    1 IRTRAN=IVAL/10000
      GO TO 10
    2 IRTRAN=(IVAL-(IVAL/10000)*10000)/100
      GO TO 10
    3 IRTRAN=IVAL-(IVAL/100)*100
   10 RETURN
      END
```

IV. NUMERICAL RESULTS

In this section, example calculations of THNAPP will be presented to demonstrate capabilities that were not present in its predecessor G3DXL. To this end, two examples will be presented: a multiple wire configuration and a solid scatterer containing a thin aperture.

A. Multiple Wire Geometry

Shown in Figure 8 is a configuration of three intersecting wires, each of different lengths and radii. This geometry can be modeled by choosing a spatial grid of dimensions.

DX0 = 2 x DY0 = 2 x DZ0 = 1.12 m

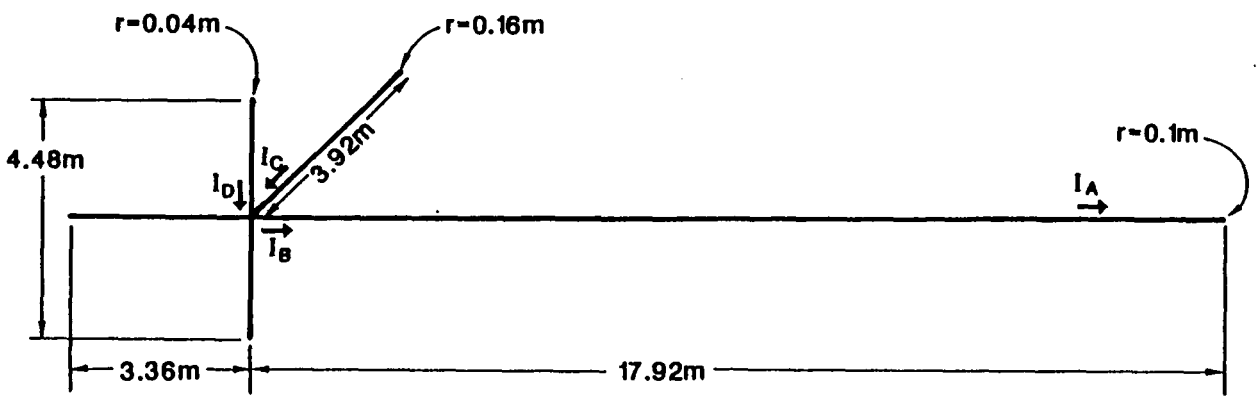

Fig. 8. A geometry of three intersecting wires on radii 0.1 m, 0.04 m, and 0.16 m. This configuration is excited by a pulse at the far right side of the longest wire and the current is monitored at locations I_A, I_B, I_C, and I_D.

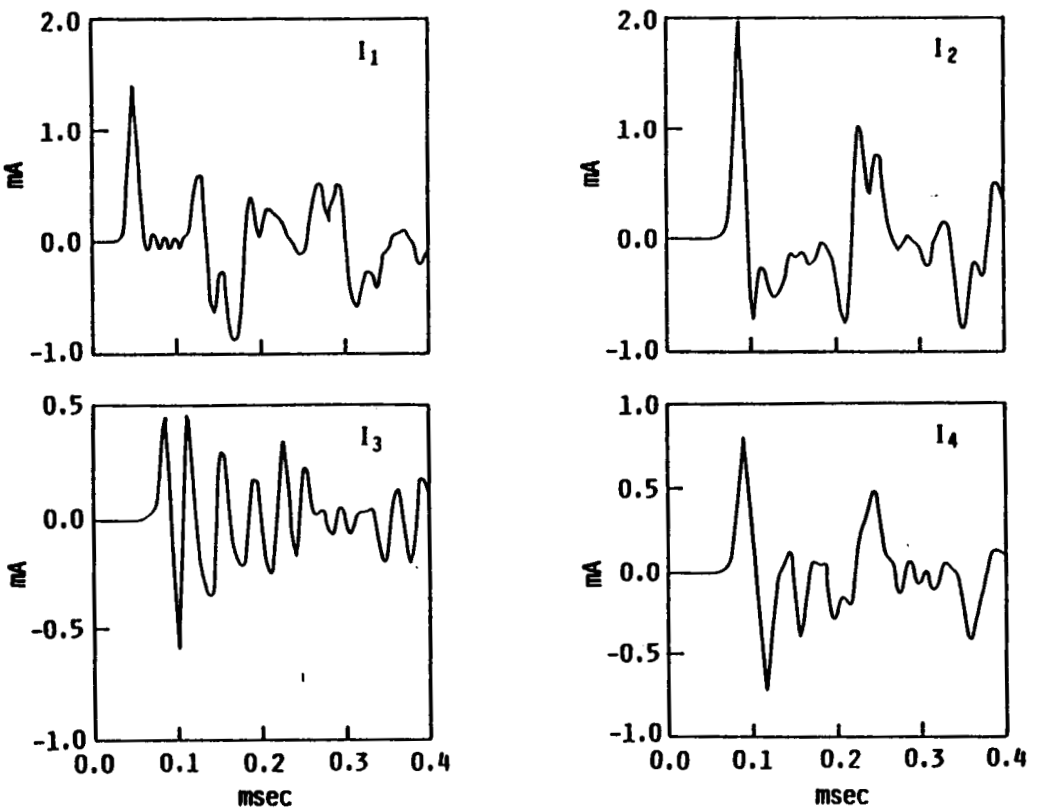

Fig. 9. The transient response of the wire geometry of Figure 8. a) - d), currents I_A, I_B, I_C, I_D, respectively.

Since that only material bodies in this geometry are wires, the NOPE array is left alone (i.e. the zero values entered in subroutine SETUP are unaltered), and the wire geometry is indicated by the statements

```
NWRS = 3
IA (1) = ITRAN (5, 13, 16)
IB (1) = ITRAN (24, 13, 16)
IA (2) = ITRAN (8, 9, 16)
IB (2) = ITRAN (8, 17, 16)
IA (3) = ITRAN (8, 13, 16)
IB (3) = ITRAN (8, 13, 9)
WRAD (1) = .01
WRAD (2) = .04
WRAD (3) = .16
```

A printout of the variables LOCI, IDIR, and MAPIQ as a function of current element number, and the variables LOCQ and MAPQI as a function of charge point number are shown in Tables 1 and 2, respectively.

I	LOCQ(I)	IDIR(I)	MAPIQ(I,1)	MAPIQ(I,2)
1	51316	1	1	2
2	61316	1	2	3
3	71316	1	3	4
4	81316	1	4	5
5	91316	1	5	6
6	101316	1	6	7
7	111316	1	7	8
8	121316	1	8	9
9	131316	1	9	10
10	141316	1	10	11
11	151316	1	11	12
12	161316	1	12	13
13	171316	1	13	14
14	181316	1	14	15
15	191316	1	15	16
16	201316	1	16	17
17	211316	1	17	18
18	221316	1	18	19
19	231316	1	19	20
20	80916	2	21	22
21	81016	2	22	23
22	81116	2	23	24
23	81216	2	24	4
24	81316	2	4	25
25	81416	2	25	26
26	81516	2	26	27
27	81616	2	27	28
28	81309	3	29	30
29	81310	3	30	31
30	81311	3	31	32
31	81312	3	32	33
32	81313	3	33	34
33	81314	3	34	35
34	81315	3	35	4

Table 1

		MAPQI(I,M)					
I	LOCQ(I)	M=1	M=2	M=3	M=4	M=5	M=6
1	51316	1	0	0	0	0	0
2	61316	1	2	0	0	0	0
3	71316	2	3	0	0	0	0
4	81316	3	4	23	24	34	0
5	91316	4	5	0	0	0	0
6	101316	5	6	0	0	0	0
7	111316	6	7	0	0	0	0
8	121316	7	8	0	0	0	0
9	131316	8	9	0	0	0	0
10	141316	9	10	0	0	0	0
11	151316	10	11	0	0	0	0
12	161316	11	12	0	0	0	0
13	171316	12	13	0	0	0	0
14	181316	13	14	0	0	0	0
15	191316	14	15	0	0	0	0
16	201316	15	16	0	0	0	0
17	211316	16	17	0	0	0	0
18	221316	17	18	0	0	0	0
19	231316	18	19	0	0	0	0
20	241316	19	0	0	0	0	0
21	80916	20	0	0	0	0	0
22	81016	20	21	0	0	0	0
23	81116	21	22	0	0	0	0
24	81216	22	23	0	0	0	0
25	81416	24	25	0	0	0	0
26	81516	25	26	0	0	0	0
27	81616	26	27	0	0	0	0
28	81716	27	0	0	0	0	0
29	81309	28	0	0	0	0	0
30	81310	28	29	0	0	0	0
31	81311	29	30	0	0	0	0
32	81312	30	31	0	0	0	0
33	81313	31	32	0	0	0	0
34	81314	32	33	0	0	0	0
35	81315	33	34	0	0	0	0

Table 2

These variables show how the subroutine WIREBLD has segmented the geometry. Of particular interest in Table 1 is the entry for MAPQI at charge point #4. Here, we see that WIREBLD has correctly sensed the intersection of the three wires and indicates that current element numbers 3, 4, 23, 24, and 35 straddle this charge. A knowledge of these variables is important if it is desired to have a particular current or charge variable monitored by subroutine DATASAV and printed out by subroutine PRINOUT. (A simple way to accomplish this is to modify the end of subroutine DATASAV so that one or more of the ESTORE, or HSTOR1, or HSTOR2 variables are over-ridden with an element of the arrays CUR or Q.)

The excitation used for this geometry is a tangential electric field pulse applied one cell from the free end of the longest ($\hat{x}$ directed wire). The exciting field is defined in subroutine PULSE by the function

$$\text{PULSE} = \tfrac{1}{2}(1 - \cos \alpha t) \ . \qquad (28)$$

where $\alpha = 2.1 \times 10^{8} \ \text{sec}^{-1}$.

The current response of the wire system to the incident current pulse is shown in Figure 9 a-d, which depicts the transient currents at locations A-D (shown in Figure 8), respectively. Figure 9a shows the incident current pulse, followed by the reflection of this pulse off of the junction point. We note that this reflection is positive, corresponding to the fact that the junction is a low impedance load. Notice also that immediately following this positive reflection is a negative, double humped pulse. This is the result of negative reflections of the pulse as it reflects off of the wire ends. The double hump results from the difference of the wire lengths after the junction.

Figure 9b shows the current on the long, $\hat{x}$ directed wire, just before the junction. Here we can easily see the initial positive reflection from the junction. and the subsequent negative reflection from the wire ends. Also, Figures 9c and 9d show the injected currents on the $\hat{y}$ and $\hat{z}$ directed wires, respectively. As can easily be seen, the initial current injected on these two wires is proportional to the wire radius- and would be expected. Other reflection and transmission phenomenon are also contained in these figures but are difficult to identify

individually due to the number of reflection points in this problem.

B. Scatterer Containing a Thin Aperture

Shown in Figure 10 is a "fuselage like", conducting geometry which contains a thin aperture of dimension 3m x 0.1m at its top and a simulated lightning channel at its "nose". The lightning channel is modeled by a 5m long wire of radius 0.01m and is excited by establishing a tangential electric field at the center segment with the same temporal characteristics as in equation (27) and in the printout of Function PULSE.

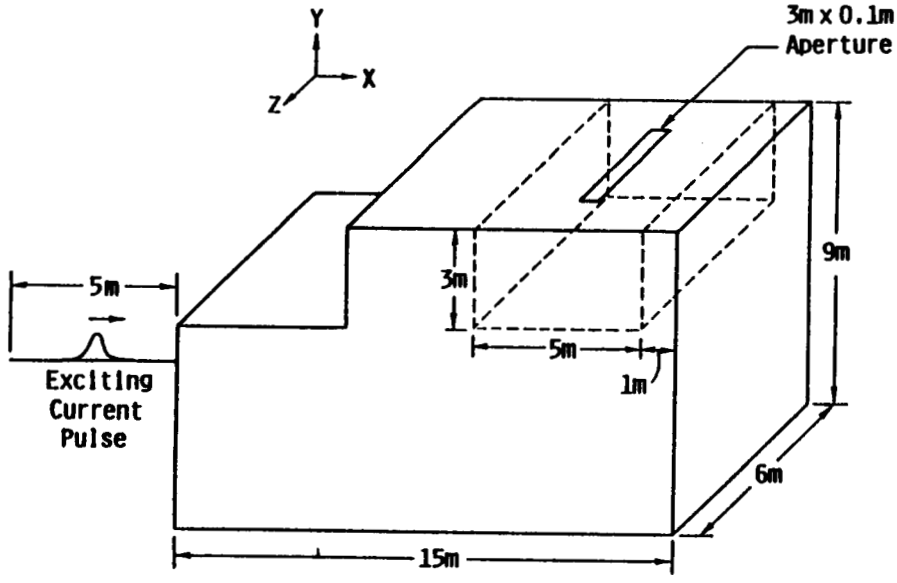

Fig. 10. A "fuselage like" geometry consisting of a conducting surface with a 3m x 0.1m aperture. This aperture is backed by a large or small cavity (shown as a dotted surface). This geometry is transiently excited by a current pulse on a simulated 5m lightning channel.

The coding for the geometry of Figure 10 is contained in the "test" option (i.e. TEST=0) of the printout of SUBROUTINE BUILD. For the unexpanded option (M5EXP=0), the cavity behind the aperture is the entire scatterer. As can be seen from the M=5, M5EXP=0 section of the printout, a line of cells centered about the aperture is absent in the plate containing the aperture.

For the expanded case (M5EXP=1), the cavity is that shown as the hidden lines in Figure 10. In order that the expanded geometry specified in the M=5, M5EXP=1 section of BUILD is correctly oriented with the unexpanded geometries of M=1,3, and 4, the appropriate values of INEAR, IFAR, JNEAR, JFAR, KNEAR, and KFAR that must be read into the program in the input file are

```
INEAR=17
IFAR=24
JNEAR=14
JFAR=21
KNEAR=1
KFAR=8
```

A line of cells, 2 cells wide, 12 cells long, and centered about the aperture has been left open for the aperture. Although a single cell line could also be used, the effect on the response will be small since the correct aperture fields are forced in the open cells of the aperture and thus the exact width of the open section is not critical.

Figures 11 and 12 show the current at the short circuited aperture (calculated during M=1) and the aperture electric field (obtained by filling the array ESTORE in subroutine DATASVE with the values of the array EAPP). The short circuit current waveform contains no information about the aperture, and thus exhibits the response characteristics expected of the external geometry of the scatterer. As would be expected, this current waveform indicates the presence of a current pulse eminating from the lightning channel that reflects back and forth over the aperture.

Comparing this waveform with Figure 11 we see that the aperture initially responds to the incident current pulse with a positive ($\hat{x}$ directed) electric field, but immediately swings negative due to the fact that the aperture starts to discharge. As a result, the aperture waveform contains high frequency components not seen in the short circuit current waveform. The aperture field of Figure 12, however, does however contain information about the details of the cavity.

C-2

Figures 13 and 14 show examples of the external and internal electric field ($\hat{x}$ directed) response in the presence of the aperture and the large cavity (M5EXP=0), respectively. In the case of these waveforms, the internal cavity

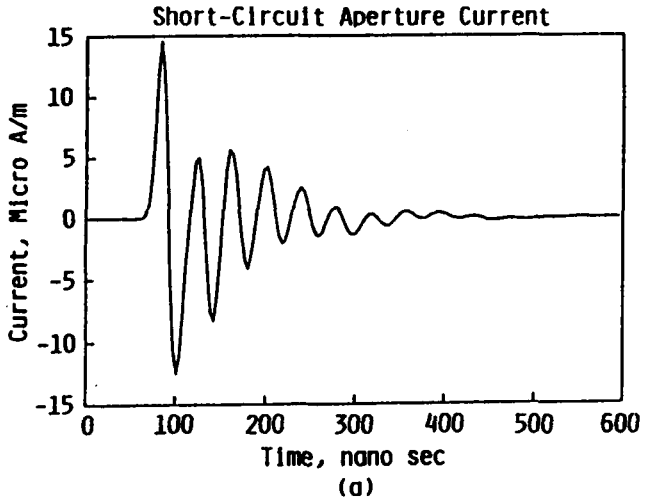

Fig. 11. Surface current excited at the center of the short circuited aperture of Figure 10.

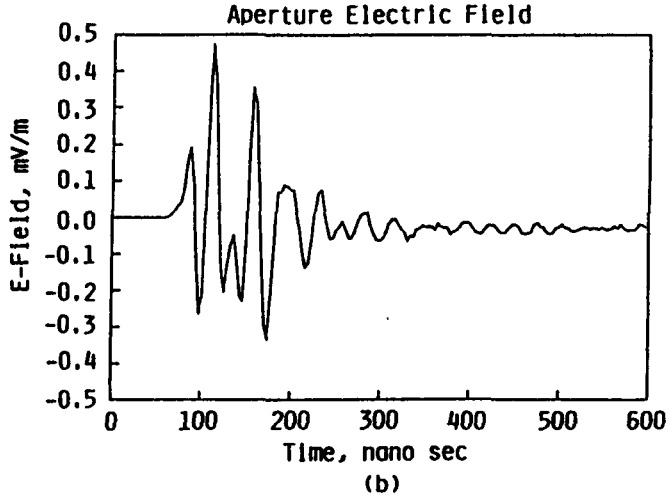

Fig. 12. The $\hat{x}$ directed electric field in the aperture of Figure 10, calculated by taking the difference between the fields calculated by the M=3 and M=4 loops.

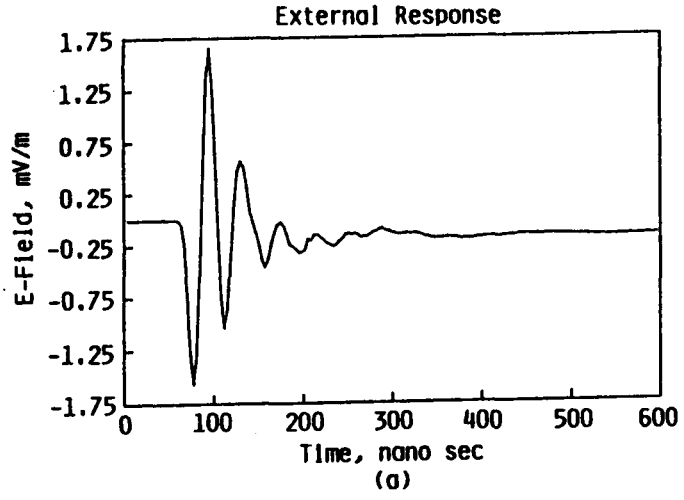

Fig. 13. The $\hat{x}$ directed electric field 3 m above the aperture of Figure 10 calculated by m=1 loop.

consists of the entire interior section of the scatterer. The points at which these waveforms are measured are 3 m above and below the aperture, respectively. As would be expected, the interior cavity response in weaker than the external response and there is a phase reversal of the initial response.

A comparison of the dotted and solid curves of Figure 14 is an important test of the self consistency of the calculated solutions. The solid curve is the interior response calculated directly from Babinet's principle and thus is obtained by taking the difference between the M=3 and 4 waveforms. Since these calculations do not contain information about the cavity, they are valid only for the opening

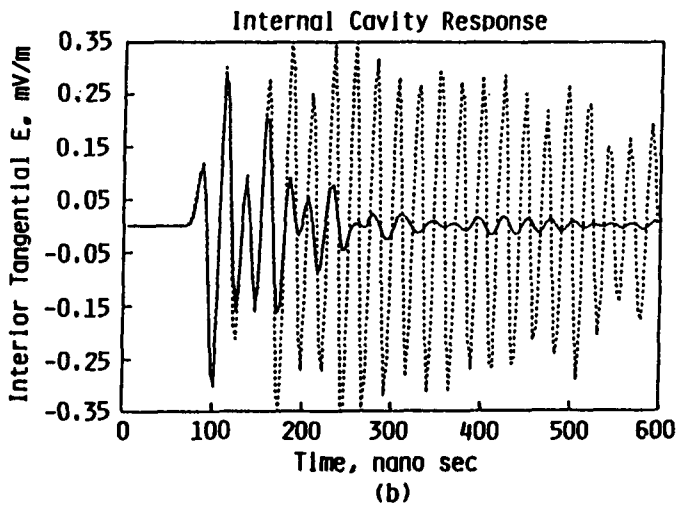

Fig. 14. The $\hat{x}$ directed electric field 3m below the aperture (and inside the small cavity) of Figure 10. Solid Curve- response calculated from M=3 and M=4 loops only. Dotted Curve- response calculated by the M=5 loop by forcing the aperture field.

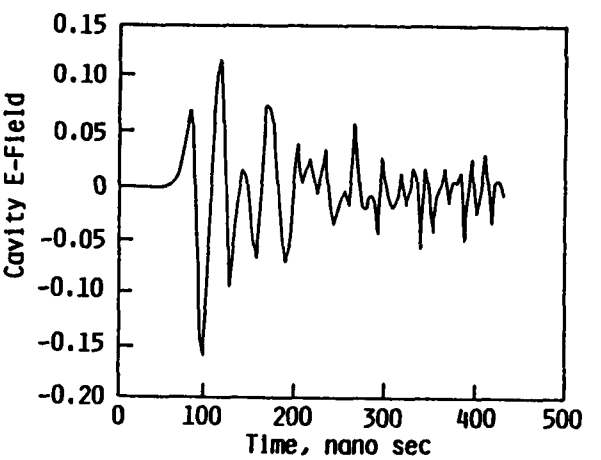

Fig. 15. The $\hat{x}$ directed electric field 2 m below the aperture (and inside the small cavity) of Figure 10. Response calculated by the M=5 loop.

portion of the response. The dotted curve, on the other hand, is obtained from the M=5 calculations where the aperture field calculated during the previous runs is forced in the aperture and only the cavity is modeled. As expected, this waveform agrees with the fields calculated directly from Babinet's principle for early times, but disagrees later since the dotted curve more correctly represents the high Q response of the cavity.

Figure 13 was calculated during the M=1 loop when the aperture was short circuited (at a distance 3m above the aperture, the effect of the aperture is small). Figure 14, however, was obtained from calculations performed during the M=3 and 4 loops, according to the rules of Equation 5. Since in this case the field point is on the same side of the aperture as are the forced surface currents, this waveform was obtained via

$$E(-3) = E^{m=3}(-3) - E^{m=4}(+3) \quad , \tag{29}$$

where the notation -3 and +3 indicates fields 3m below and 3m above the aperture plane, respectively. Thus, the test points defined in the input file for use by subroutine DATASVE were set appropriately.

Finally, shown in Figure 15 is the $\hat{x}$ directed electric field inside the small cavity (i.e., for M5EXP=1) indicated in Figure 10, 2 meters below the aperture. As would be expected, this waveform is of smaller magnitude than is that of Figure 14 since the cavity resonances are now quite high in frequency as compared to the power spectrum of the incident lightning pulse.

V. CONCLUSION

This report has presented a Finite-Difference Time-Domain electromagnetic computer code, called THNAPP, that is capable of analyzing metal structures such as airplanes and missiles that contain long apertures that are much thinner than a wavelength. This code utilizes a technique that employs Babinet's principle to effectively separate the regions on both sides of the aperture so that numerical grids most appropriate to each region can be used. This code retains the basic architecture of the code G3DXL [7], thus demanding a minimum of additional effort to run this code. The preceding sections have provided a complete description of the theory utilized in this code, a complete description of the operation of the code as well as the use and philosophy utilized in the subroutines, and numerical examples demonstrating the operation of the code.

The numerical examples shown in this report have demonstrated the basic capabilities of this code, but have been deliberately kept simple so as to aid in the readers understanding of the use of the code and the interpretation of the results. The modeling of more "real life" problems such as aircraft and multiple apertures and complex cavities is a straightforward application of the instructions given in the sections describing the code.

REFERENCES

[1] Zaepfel, K.P., Fisher, B.D., Ott, M.S., "Direct-Strike Lightning Photographs, Swept-Flash Attachment Patterns, and Flight Conditions for Storm Hazards '82," NASA Technical Memorandum 86347, February 1985.

[2] Turner, C.D., Trost, T.F., "Laboratory Modeling and Analysis of Aircraft-Lightning Interactions," Report on NASA Grant NAG-1-28, August 1982. (Available as NASA CR-169455.)

[3] Rudolph, T.H., Perala, R.A., McKenna, P.M., Parker, S.L., "Investigations into the Triggered Lightning Response of the F106B Thunderstorm Research Aircraft," NASA Contractor Report 3902, June 1985.

[4] Yee, K.S., "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," IEEE Trans. Ant. Prop., vol. AP-14, no. 3, pp. 302-307, May 1966.

[5] Kunz, K.S., Simpson, L., "A Technique for Increasing the Resolution of Finite-Difference Solutions of the Maxwell Equation," IEEE Trans. Electro. Compat., vol. EMC-23, no. 4, pp. 419-422, November 1981.

[6] Holland, R., and Simpson L., "Finite-Difference Analysis of EMP Coupling to Thin Struts and Wires," IEEE Trans. Electro. Compat., vol. EMC-23, no. 2, pp. 88-97, May 1982.

[7] Kunz, K.S., "Generalized Three-Dimensional Experimental Lightning Code (G3DXL) User's Manual," NASA Contractor Report 166079, February, 1986.

[8] Demarest, K.R., "A Finite Difference-Time Domain Technique for Modeling Narrow Apertures in Conducting Scatterers," accepted for publication in IEEE Trans. Ant. Prop., July, 1987.

[9] Schelkunoff, S.A., Electromagnetic Waves, D. Van Nostrand Co., Inc., 1943.

[10] Jones, D.S., The Theory of Electromagnetism, The MacMillan Co., pp. 569-572, 1964.

[11] Harrington, R.F., Field Computations by Moment Methods, The MacMillan Co., 1968.

[12] Holland, R., "THREDE: A Free-Field EMP Coupling and Scattering Code," IEEE Trans. on Nuclear Science, vol. NS-24, no. 6, pp. 2416-2421, December 1977.

[13] Harrington, R.F., Time-Harmonic Electromagnetic Fields, The McGraw Hill Book Co., 1961.

[14] Mur., G., "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic Field Equations," IEEE Trans. on Electro. Compat., vol. EMC-23, no. 4, pp. 377-382, November 1981.

Standard Bibliographic Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-178280		

4. Title and Subtitle	5. Report Date
The Electromagnetic Modeling of Thin Apertures Using the Finite-Difference Time-Domain Technique	June 1987
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Kenneth R. Demarest	
9. Performing Organization Name and Address	10. Work Unit No.
The University of Kansas Center for Research, Inc. 2291 Irving Hill Drive - Campus West Lawrence, KS 66045	505-66-21-04
	11. Contract or Grant No.
	NAG1-514
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, DC 20546	Contractor Report 1985-1986
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Klaus P. Zaepfel
Final Report

16. Abstract

A technique which computes transient electromagnetic responses of narrow apertures in complex conducting scatterers was implemented as an extension of previously developed Finite-Difference Time-Domain (FDTD) computer codes. Although these apertures are narrow with respect to the wavelengths contained within the power spectrum of excitation, this technique dos not require significantly more computer resources to attain the increased resolution at the apertures. In the report, an analytical technique which utilizes Babinet's principle to model the apertures is developed, and an FDTD computer code which utilizes this technique is described.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Finite difference code, lightning, LEMP, computer modeling, thin seams	[redacted] Subject category - 47

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	106	

[redacted]