NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Three Dimensional Finite Element Method Applied to Study the Penetration of Electromagnetic Fields in Cavities

by
Darcy N. Ladd

A thesis
submitted to the school of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science.

Ottawa-Carleton Institute for Electrical Engineering,
Department of Electrical Engineering,
Faculty of Engineering,
University of Ottawa.

ISBN 0-315-60095-0

Canada

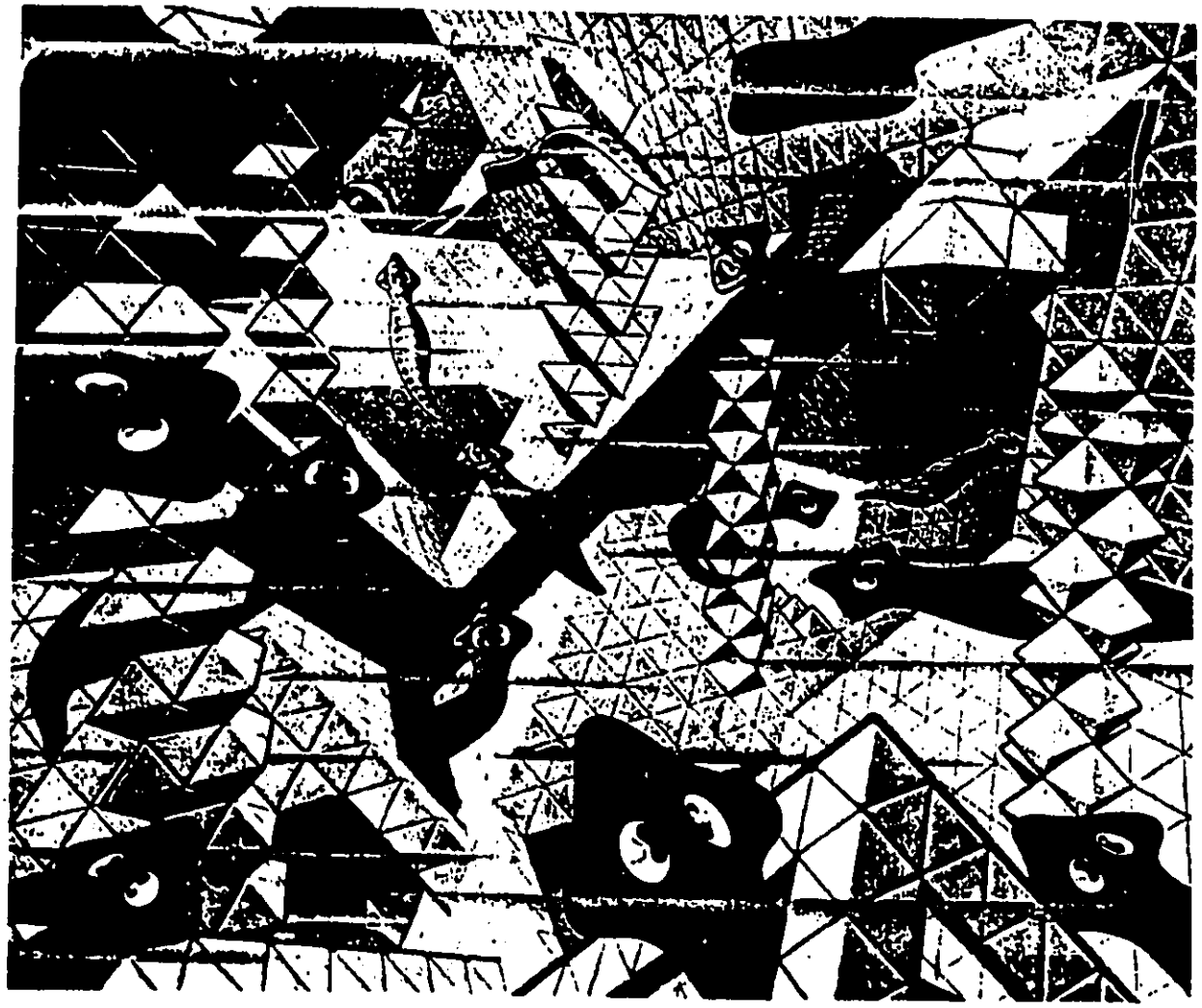
UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

A three dimensional formulation of the finite element method was developed to solve the electromagnetic field distribution in an arbitrary region containing conducting and dielectric materials when the tangential magnetic field was known at the boundaries. The formulation was developed using a three component vector magnetic potential and a scalar electric potential. The displacement current as well as the conduction current term was accounted for. The region of interest was discretized using eight node isoparametric hexahedrons and the potential functions were defined using linear first order basis functions. The frequency domain finite element method program was validated by comparison with closed form solutions for simplified geometries. The algorithm proved to have a convergent solution when solving the diffusion of electromagnetic fields into conducting hollow and solid structures without apertures. The penetration of a steady-state electromagnetic field through an aperture into a simple cavity was analyzed with the 3-D FEM program. The diffusion of a step-impulse magnetic field into a conducting slab was solved directly in the time domain with a time domain finite element program. Conclusions were drawn on the feasibility of using the finite element method as part of an EMI/EMC CAD package.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. G. Costache for his guidance and encouragement throughout this work. I must also acknowledge my appreciation to Dr. S.S. Stuchly for his insight, and to R.R. Goulette of Bell Northern Research for sharing his vast resources on any topic. I am indebted to many of my teachers and colleagues for the knowledge and advice they imparted to me while performing this research. I am also deeply indebted to the technicians and consultants who maintained and upgraded the computers. Finally, I owe a debt of gratitude for my family and friends for their patience and understanding.



FLAT WORMS: a lithograph by M.C. Escher [30].

Contents

List of Figures

List of Symbols

Vector quantities are indicated by bold characters. All complex quantities are underlined. A matrix is indicated by [ ].

$\boldsymbol{B}$ magnetic flux density vector

$\boldsymbol{H}$ magnetic field intensity vector

$\boldsymbol{E}$ electric field intensity vector

$\boldsymbol{J}$ current density vector

ρ charge density

$\boldsymbol{A}$ vector magnetic potential

ϕ scalar electric potential

$\hat{\boldsymbol{A}}$ approximate solution of vector magnetic potential

$\hat{\phi}$ approximate solution of scalar electric potential

$\boldsymbol{H}_t$ tangential magnetic field component

$\boldsymbol{E}_t$ tangential electric field component

x, y, z axes in Cartesian coordinate system

s, t, r axes in natural coordinate system

$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ unit vectors in Cartesian coordinates

$\boldsymbol{r}$ position vector

$\boldsymbol{n}$ unit vector normal

Ω domain of interest

Γ boundary of the domain of interest

$N_j^{(e)}$ interpolatory function of the jth node for element 'e'

$\boldsymbol{w}$ vector weighting function

w scalar weighting function

ϵ permittivity

μ permeability

σ conductivity

∇ the nabla operator

FEM finite element method

EMI electromagnetic interference

EMC electromagnetic compatibility

EMP electromagnetic pulse

ESD electrostatic discharge

CAD computer aided design

2-D two dimensional

3-D three dimensional

Chapter 1

Introduction

1.1 Motivation

There is an ever increasing density of electrical and electronic circuits and devices in modern systems for computation, communication, and control. We rely on these devices for research, business, manufacturing, recreation, security, and numerous other activities, and they have become an integral part of our society. The demand for increased performance and throughput from these complex systems requires higher speed circuitry and broader bandwidths and this has had a major impact on the number and severity of *electromagnetic interference (EMI)*[1] situations. These situations cannot go over-looked because even seemingly minor problems can cause major malfunctions and a loss of reliability; usually manifesting themselves as only a reduction of services, but in some cases, compromising personal safety or national security.

Electrical and electronic devices are electromagnetically compatible when the electrical emissions generated by one device, do not interfere with the performance of any other device. The *electromagnetic compatibility (EMC)* of systems is the desirable situation in which the systems work as intended in their environment. Electromagnetic interference exists when undesirable voltages, currents, or fields are present in a system and influence its performance in an adverse way. The interfering signal or noise enters the victim device by conduction or by radiation.

A device can be shielded within a metal enclosure to eliminate or reduce the incidence of electromagnetic fields radiated from external sources, or to contain the propagation of radiation from its circuits. The integrity of the shielding effectiveness of the enclosure is compromised however, because of the need for apertures in the enclosure (cooling vents, displays, etc.).

If the electromagnetic field distribution inside a shielding enclosure can be predicted numerically, then the behavior of the field as a function of aperture size and location can be studied and therefore, the system designer's ability to minimize EMI problems by suitable selection of aperture sizing and placement is greatly enhanced. An algorithm which allows the prediction of the electromagnetic fields in a three dimensional structure would be an important part of a *computer aided design (CAD)* tool for EMC engineers.

1.2 Objectives

The objectives of the research described in this thesis are to develop a numerical algorithm to calculate the three dimensional distribution of the electromagnetic field inside enclosures and then to use it to model the effects of different apertures in the shield wall on the electromagnetic field distribution.

The problem is to be modelled in three dimensions so that arbitrary, but realistic cases can be solved. The general problem consists of an arbitrarily shaped metal box which has one or several apertures of arbitrary size and shape. An incident electromagnetic field irradiates the exterior of the enclosure and the electromagnetic energy diffuses through the walls and penetrates through the apertures.

The feasibility of a solution by the *Finite Element Method (FEM)* will be explored. The finite element method appears to be suitable for solving this type of problem because it can readily model complex geometries and inhomogeneous regions, and it is fairly simple to program.

1.3 Existing Knowledge of the FEM

Since the late 1960's, when the finite element method was first introduced for use in solving electromagnetic problems, uses and applications of the method have been widely researched and accounts of these works have appeared as countless numbers of papers and monographs.

The inherent complexity of the FEM in three dimensional analysis (or any numerical method in fact) has restricted most work to date to be performed in one or two dimensions. A two dimensional problem must extend infinitely in one dimension, along which axis there is no field variation. Using the Helmholtz or diffusion equation

expressed as a function of two dimensions, the E-polarized field, or the H-polarized field (respectively the TM or TE field) can be solved in terms of E, H, or A.

Since the geometries of the structures in the majority of electrical engineering applications are complex and the sources are multi-dimensional, a truly three dimensional analysis is required to obtain the current and field distribution in the region of interest. With the better understanding of the merits of FEM and with the availability of more powerful computers, it has become feasible to develop three dimensional finite element method programs.

Carpenter [2] examined the range of choice of magnetic and electric vector potentials. He showed that the problem formulation was greatly affected by the choice of the gauge and the scalar potentials and also discussed the differences in the number of functions which have to be computed, the couplings between them, the choice of the divergence, and the regions of computation.

Chari, D'Angelo, and Palmo [3], presented a three component $(2-A,\phi)$ formulation to represent eddy currents fields accurately in a two dimensional geometry. The method was applied to problems of conducting slabs with cracks in them, and they concluded that the scalar electric potential was required to ensure a zero divergence of the induced eddy currents.

Several other papers which examined the use of 3-D FEM to solve electromagnetic diffusion field problems are [4,5,6,7,8]

Salon and Peng [9] presented a general approach to solving three dimensional eddy current problems, using a four component $A-\phi$ formulation. Using the finite element method with eight-noded bricks to solve several examples, they illustrated the validity of their formulation. They also showed that as the conduction current approached the surface of the conductor, it turned away, ensuring the normal component of the current was zero at the boundary. No special boundary conditions were imposed, but the scalar electric potential was essential in ensuring the continuity of the eddy currents.

There are several books [10,11,12,13] which include excellent descriptions of the theory and basics of the finite element method and its application to solving electromagnetic problems.

Chapter 2

The Numerical Technique

There are many practical problems in electromagnetics which cannot be properly modelled with only one or two dimensions. To fully comprehend the scattering effect of a three dimensional object such as a box, the problem must be solved in three dimensions. The following chapter details the development of the three dimensional finite element technique to compute the distribution of electromagnetic fields in three dimensional structures.

Electromagnetic sources can be characterized as continuous wave or transient signals. Continuous wave signals are band limited and are usually described in terms of their frequency components. Transient signals have a broadband frequency characteristic, but can be described as having finite time duration. Broadband radiation from a source such as *electromagnetic pulse (EMP)* (also: nuclear EMP (NEMP), electrostatic discharge (ESD), and lightning) can couple large quantities of noise into a circuit because of the high field levels [kV/m, kA/m], and the fast rise-times [ns] [1,15]. Problems involving transient signals are most readily studied directly in the time domain.

The formulation begins with Maxwell's time-domain equations in the local form and develops the weak form of the equations suitable for use with the finite element method. The problem of interest contains both conducting bodies and dielectric bodies. The FEM has been selected because of the ease with which it can be adapted to the complicated shapes and inhomogeneities of an arbitrary problem.

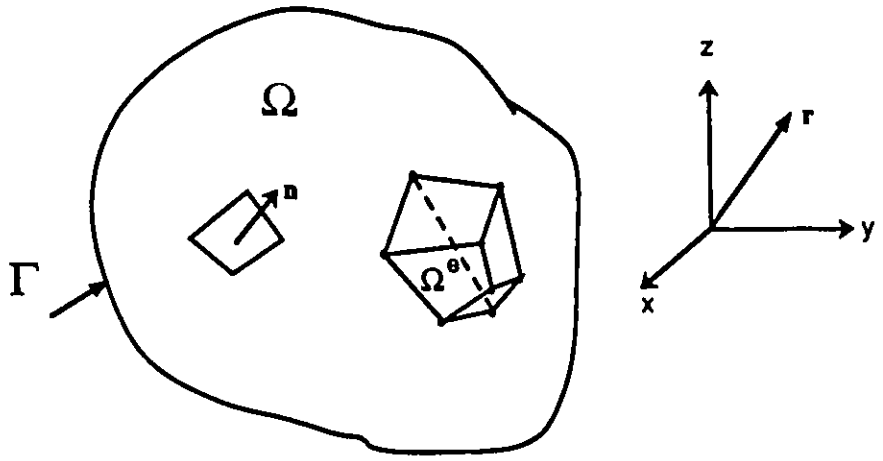


Figure 2.1: The domain and boundary of a 3-D electromagnetic problem

2.1 The Mathematical Formulation

Ampere's Law

The time-domain solution of the electromagnetic fields in the region of interest will be computed by means of the finite element method. Shown in Fig. 2.1 is the domain Ω, which has a boundary Γ. The region of interest can be inhomogeneous, but the properties are assumed to be isotropic and non-dispersive. The region contains no free sources ($\rho_s = 0$, and $\boldsymbol{J}_s = 0$). Within the finite element sub-regions (denoted $\Omega^{(e)}$), the constitutive parameters $\varepsilon^{(e)}$, $\mu^{(e)}$ and $\sigma^{(e)}$ are constant with respect to space and time.

Ampere's equation is

$$\nabla \times \frac{1}{\mu} \boldsymbol{B}(\boldsymbol{r},t) - \boldsymbol{J}(\boldsymbol{r},t) = 0$$

where $\boldsymbol{J} = \boldsymbol{J}_c + \boldsymbol{J}_d$. The conduction current is $\boldsymbol{J}_c(\boldsymbol{r},t) = \sigma \boldsymbol{E}(\boldsymbol{r},t)$. The displacement current, $\boldsymbol{J}_d(\boldsymbol{r},t) = \frac{\partial}{\partial t}\varepsilon \boldsymbol{E}(\boldsymbol{r},t)$ must be included in the solution to account for radiation

occurring at high frequencies. Thus Ampere's equation can be written:

$$\nabla \times \frac{1}{\mu} \boldsymbol{B}(\boldsymbol{r},t) - \left(\sigma + \varepsilon \frac{\partial}{\partial t}\right) \boldsymbol{E}(\boldsymbol{r},t) = 0. \tag{2.1}$$

To insure a continuity of currents across all boundaries, the divergence of the current must be zero everywhere because there are no free charges ($\rho_s = 0$). The current continuity equation is

$$\nabla \cdot \boldsymbol{J}(\boldsymbol{r},t) = 0,$$

or

$$\nabla \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t}\right) \boldsymbol{E}(\boldsymbol{r},t) = 0. \tag{2.2}$$

The electric and magnetic fields, respectively $\boldsymbol{E}(\boldsymbol{r},t)$ and $\boldsymbol{H}(\boldsymbol{r},t)$, can be uniquely determined by knowing their initial conditions and the values of one of $\boldsymbol{E}_t(\boldsymbol{r},t)$ or $\boldsymbol{H}_t(\boldsymbol{r},t)$, the tangential electric or magnetic field component, at every point on the surface Ω at all times. A proof of this theorem is shown in appendix A.

The Potential Functions

The electric and magnetic fields can be defined with the use of a vector magnetic potential $\boldsymbol{A}$, and a scalar electric potential ϕ. The vector magnetic potential $\boldsymbol{A}$ is defined by: $\boldsymbol{B} = \nabla \times \boldsymbol{A}$ and can be expressed as $\boldsymbol{A} = A_x\boldsymbol{x} + A_y\boldsymbol{y} + A_z\boldsymbol{z}$. The scalar electric potential ϕ is defined by $\boldsymbol{E} = -\nabla\phi - \frac{\partial}{\partial t}\boldsymbol{A}$.

The number of unknown scalars has been reduced from six variables (B_x, B_y, B_z, E_x, E_y, E_z), to four variables (A_x, A_y, A_z, ϕ). The scalar electric potential is necessary to ensure the electric field is properly described by accounting for the build up of free charges which occur at surface discontinuities.

Rewrite the two above equations, replacing $\boldsymbol{E}$ and $\boldsymbol{B}$ with the potentials to get:

$$\nabla \times \frac{1}{\mu} \nabla \times \boldsymbol{A} - \left(\sigma + \varepsilon \frac{\partial}{\partial t}\right) \left(-\nabla\phi - \frac{\partial}{\partial t}\boldsymbol{A}\right) = 0, \tag{2.3}$$

and

$$\nabla \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t}\right) \left(-\nabla\phi - \frac{\partial}{\partial t}\boldsymbol{A}\right) = 0. \tag{2.4}$$

These two equations will be solved as a coupled pair.

The Galerkin Integral

Equations (2.3) and (2.4) must be solved for the unknowns $\boldsymbol{A}$ and ϕ in the region Ω. The domain Ω is broken into discrete regions $\Omega^{(e)}$; elements over which the Eqs. (2.3) and (2.4) can be solved.

The potential functions are approximated over each sub-region of interest, which is described in space using M discrete points, using basis functions in the form

$$\psi(\boldsymbol{r}) = \sum_{m=1}^{M} \psi_m N_m(\boldsymbol{r}),$$

where $\psi(\boldsymbol{r})$ is the potential to be evaluated at the point described by $\boldsymbol{r}$, N_m is the set of the M interpolatory functions, and $\{\psi_m; m = 1, M\}$ is the value of the potential at each of the M nodes. The interpolatory functins must be defined such that

$$N_m = \begin{cases} 1, & \text{if node}=m; \\ 0, & \text{otherwise.} \end{cases}$$

The functions used for interpolation must be of at least the order of the highest order derivative of the equation to be solved. This ensures continuity across inter-element boundaries, and avoids any terms becoming infinite. Linear functions will be used to approximate the potential distributions inside of each of the small elements.

The finite element solution here is based upon the weighted residual method[13]. The error, or the residual of each potential is introduced as $R_\Omega = \psi - \hat{\psi}$ where ψ is the exact solution, and $\hat{\psi}$ is an approximate solution. We attempt to reduce the residual over the whole domain Ω by making the weighted integral of the residue over each sub-region $\Omega^{(e)}$ equal to zero. This is imposed as:

$$\langle w_j, \psi - \hat{\psi} \rangle = 0; \qquad j = 1, 2, \ldots, N \tag{2.5}$$

where $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ is the inner product of the two functions $\boldsymbol{u}$ and $\boldsymbol{v}$[1], and $\{w_j; j = 1, N\}$ is an appropriate set of independent weighting functions. The general convergence of $\hat{\psi} \to \psi$, or $\psi - \hat{\psi} \to 0$, requires that Eq. (2.5) be satisfied for all j, as $N \to \infty$.

Apply the inner product to Eqs. (2.3) and (2.4). Multiply (2.3) with a vector weighting function $\boldsymbol{w}_j$, and (2.4) with a scalar weighting function w_j. Integrate both equations over the entire domain Ω and set the resulting equations equal to zero. The approximate solutions of these equations by a numerical method are indicated by $\hat{\boldsymbol{A}}$

[1]The inner product of $\boldsymbol{u}$ and $\boldsymbol{v}$: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \int_{\Omega(e)} \boldsymbol{u} \cdot \boldsymbol{v}\, dV$

and $\hat{\phi}$. We get:

$$\int_{\Omega} \left[\boldsymbol{w}_j \cdot \nabla \times \frac{1}{\mu} \nabla \times \hat{\boldsymbol{A}} + \boldsymbol{w}_j \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t} \right) \nabla \hat{\phi} + \boldsymbol{w}_j \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t} \right) \frac{\partial}{\partial t} \hat{\boldsymbol{A}} \right] dR = 0, \quad (2.6)$$

and

$$\int_{\Omega} \left[w_j \nabla \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t} \right) \nabla \hat{\phi} + w_j \nabla \cdot \left(\sigma + \varepsilon \frac{\partial}{\partial t} \right) \frac{\partial}{\partial t} \hat{\boldsymbol{A}} \right] dR = 0. \quad (2.7)$$

The Galerkin method requires that the weighting functions be the same as the basis functions. The previous step lead to a set of simultaneous equations with $\hat{A}_x$, $\hat{A}_y$, $\hat{A}_z$, and $\hat{\phi}$ being the unknowns.

Equations (2.6) and (2.7) can be reduced by means of some vector identities and integral theorems. Note that $\mu^{(e)}$, $\varepsilon^{(e)}$ and $\sigma^{(e)}$ are constant inside each element region $\Omega^{(e)}$, and can be moved outside the derivative and integral operators. Since the integration is over the volume, the result is a function of time only, and the partial derivatives become total derivatives.

Using the identity $\boldsymbol{A} \cdot (\nabla \times \boldsymbol{B}) = \boldsymbol{B} \cdot (\nabla \times \boldsymbol{A}) - \nabla \cdot (\boldsymbol{A} \times \boldsymbol{B})$, the first term of (2.6) is

$$1^{st} \text{ term of } (2.6) = \frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} \left[\left(\nabla \times \hat{\boldsymbol{A}} \right) \cdot (\nabla \times \boldsymbol{w}_j) - \nabla \cdot \left(\boldsymbol{w}_j \times \left(\nabla \times \hat{\boldsymbol{A}} \right) \right) \right] dR$$

and then using the Divergence theorem[2],

$$= \frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} \left(\nabla \times \hat{\boldsymbol{A}} \right) \cdot (\nabla \times \boldsymbol{w}_j) \, dR - \frac{1}{\mu^{(e)}} \int_{\Gamma^{(e)}} \boldsymbol{w}_j \times \left(\nabla \times \hat{\boldsymbol{A}} \right) \cdot \boldsymbol{n} \, dS,$$

or

$$= \frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} \left(\nabla \times \hat{\boldsymbol{A}} \right) \cdot (\nabla \times \boldsymbol{w}_j) \, dR - \frac{1}{\mu^{(e)}} \int_{\Gamma^{(e)}} \boldsymbol{w}_j \cdot \left(\nabla \times \hat{\boldsymbol{A}} \right) \times \boldsymbol{n} \, dS$$

where $\boldsymbol{n}$ is a unit vector normal to the boundary Γ.

Using Green's first identity[3], the first term of (2.7) can be rewritten:

1^{st} term of (2.7)

$$= - \left(\sigma^{(e)} + \varepsilon^{(e)} \frac{d}{dt} \right) \int_{\Omega^{(e)}} \nabla w_j \cdot \nabla \hat{\phi} \, dR + \left(\sigma^{(e)} + \varepsilon^{(e)} \frac{d}{dt} \right) \int_{\Gamma^{(e)}} \left(w_j \nabla \hat{\phi} \right) \cdot \boldsymbol{n} \, dS.$$

[2]Divergence theorem: $\int_{\Omega} \nabla \cdot \boldsymbol{A} \, dV = \int_{\Gamma} \cdot \boldsymbol{A} \boldsymbol{n} \, dS$

[3]Green's first identity: $\int_{\Omega} \vartheta (\nabla \cdot \nabla \psi) \, dR = - \int_{\Omega} \nabla \vartheta \cdot \nabla \psi \, dR + \int_{\Gamma} (\vartheta \nabla \psi) \cdot \boldsymbol{n} \, dS$

By using the identity $\psi\nabla\cdot\boldsymbol{\theta} = \nabla\cdot(\psi\boldsymbol{\theta}) - \boldsymbol{\theta}\cdot\nabla\psi$, the second term of (2.7) becomes:

$$2^{nd} \text{ term of (2.7)}$$
$$= -\left(\sigma^{(e)} + \varepsilon^{(e)}\frac{d}{dt}\right)\frac{d}{dt}\int_{\Omega^{(e)}} \hat{\boldsymbol{A}}\cdot\nabla w_j\, dR + \left(\sigma^{(e)} + \varepsilon^{(e)}\frac{d}{dt}\right)\frac{d}{dt}\int_{\Omega^{(e)}} \nabla\cdot\left(w_j\hat{\boldsymbol{A}}\right) dR$$

and then using the Divergence theorem:

$$= -\left(\sigma^{(e)} + \varepsilon^{(e)}\frac{d}{dt}\right)\frac{d}{dt}\int_{\Omega^{(e)}} \hat{\boldsymbol{A}}\cdot\nabla w_j\, dR + \left(\sigma^{(e)} + \varepsilon^{(e)}\frac{d}{dt}\right)\frac{d}{dt}\int_{\Gamma^{(e)}} w_j\hat{\boldsymbol{A}}\cdot\boldsymbol{n}\, dS.$$

The weighted integral equation form of Eqs. (2.3) and (2.4) are presented here in their general form:

$$\frac{1}{\mu^{(e)}}\int_{\Omega^{(e)}} (\nabla\times\boldsymbol{w}_j)\cdot\left(\nabla\times\hat{\boldsymbol{A}}\right) dR + \sigma^{(e)}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\nabla\hat{\phi}\, dR + \sigma^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\hat{\boldsymbol{A}}\, dR$$
$$+ \varepsilon^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\nabla\hat{\phi}\, dR + \varepsilon^{(e)}\frac{d^2}{dt^2}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\hat{\boldsymbol{A}}\, dR \qquad (2.8)$$
$$-\frac{1}{\mu^{(e)}}\int_{\Gamma^{(e)}} \boldsymbol{w}_j\cdot\left(\nabla\times\hat{\boldsymbol{A}}\right)\times\boldsymbol{n}\, dS = 0,$$

and

$$\sigma^{(e)}\int_{\Omega^{(e)}} \nabla w_j\cdot\nabla\hat{\phi}\, dR + \sigma^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \nabla w_j\cdot\hat{\boldsymbol{A}}\, dR + \varepsilon^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \nabla w_j\cdot\nabla\hat{\phi}\, dR$$
$$+ \varepsilon^{(e)}\frac{d^2}{dt^2}\int_{\Omega^{(e)}} \nabla w_j\cdot\hat{\boldsymbol{A}}\, dR - \sigma^{(e)}\int_{\Gamma^{(e)}} w_j\left(\nabla\hat{\phi} + \frac{d}{dt}\hat{\boldsymbol{A}}\right)\cdot\boldsymbol{n}\, dS \qquad (2.9)$$
$$-\varepsilon^{(e)}\frac{d}{dt}\int_{\Gamma^{(e)}} w_j\left(\nabla\hat{\phi} + \frac{d}{dt}\hat{\boldsymbol{A}}\right)\cdot\boldsymbol{n}\, dS = 0.$$

Notice that Eqs. (2.8) and (2.9) contain only first order derivatives of the potentials with respect to position, whereas Eqs. (2.6) and (2.7) contained second order derivatives as well. The new equations are in a *weak form* of the original ones. By using Green's first identity, which is essentially an integration by parts, the operators on $\boldsymbol{A}$ and ϕ have been reduced. The price for this is that the operators on the weighting functions $\boldsymbol{w}$ and w are of a higher order. It has been said that the weak form is often more realistic physically than the original differential equation[14].

Interface Conditions

If the surface of the element being integrated ($\Gamma^{(e)}$) lies on the interior of the region Ω, then the contribution from the surface integral of Eq. (2.8) over the common surface of the two adjacent elements will be zero because the normals on the common surface of the two adjacent elements are in opposite directions. However, if the surface of the element being integrated lies on the exterior surface Γ of the region Ω, then the normal is not opposed, and the term in the surface integral of Eq. (2.8) can be replaced with the tangential magnetic field on the surface because

$$\boldsymbol{H}_t = \frac{1}{\mu}\nabla \times \hat{\boldsymbol{A}} \times \boldsymbol{n}.$$

The $\boldsymbol{H}_t$ component on the exterior surface is the source term, and thus its contribution must be computed over the entire surface and it becomes the source vector on the right hand side of the equation.

The term of the surface integral of Eq. (2.9) $\left(\sigma^{(e)} + \varepsilon^{(e)}\frac{d}{dt}\right)\left(-\nabla\hat{\phi} - \frac{d}{dt}\hat{\boldsymbol{A}}\right)\cdot\boldsymbol{n}$ which is $\boldsymbol{J}\cdot\boldsymbol{n}$ and is the normal component of the total current $\boldsymbol{J}$. This term will produce no contribution to the final system because the total current must be conserved, and the two surface normals of the adjacent elements are opposite.

2.1.1 The Time Domain Equations

Finally the set of equations which can be solved to find the approximations of the field potentials $\hat{\boldsymbol{A}}(t)$ and $\hat{\phi}(t)$, are

$$\begin{aligned}\frac{1}{\mu^{(e)}}\int_{\Omega^{(e)}} (\nabla \times \boldsymbol{w}_j)\cdot\left(\nabla\times\hat{\boldsymbol{A}}\right) dR + \sigma^{(e)}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\nabla\hat{\phi}\, dR \\ +\sigma^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\hat{\boldsymbol{A}}\, dR + \varepsilon^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\nabla\hat{\phi}\, dR \\ +\varepsilon^{(e)}\frac{d^2}{dt^2}\int_{\Omega^{(e)}} \boldsymbol{w}_j\cdot\hat{\boldsymbol{A}}\, dR = \int_{\Gamma^{(e)}} \boldsymbol{w}_j\cdot\boldsymbol{H}_t\, dS,\end{aligned} \tag{2.10}$$

and

$$\begin{aligned}\sigma^{(e)}\int_{\Omega^{(e)}} \nabla w_j\cdot\nabla\hat{\phi}\, dR + \sigma^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \nabla w_j\cdot\hat{\boldsymbol{A}}\, dR \\ +\varepsilon^{(e)}\frac{d}{dt}\int_{\Omega^{(e)}} \nabla w_j\cdot\nabla\hat{\phi}\, dR + \varepsilon^{(e)}\frac{d^2}{dt^2}\int_{\Omega^{(e)}} \nabla w_j\cdot\hat{\boldsymbol{A}}\, dR = 0.\end{aligned} \tag{2.11}$$

2.1.2 The Frequency Domain Equations

If the problem is solved in steady state, then the equations can be simplified by replacing $\frac{d}{dt}$ with $j\omega$. The field components become complex and the notation used to denote this is an underlined quantity. The subsequent equations are

$$\frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} (\nabla \times \boldsymbol{w}_j) \cdot \left(\nabla \times \underline{\hat{\boldsymbol{A}}}\right) dR - \left(\omega^2 \varepsilon^{(e)} - j\omega\sigma^{(e)}\right) \int_{\Omega^{(e)}} \boldsymbol{w}_j \cdot \underline{\hat{\boldsymbol{A}}}\, dR + \left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right) \int_{\Omega^{(e)}} \boldsymbol{w}_j \cdot \nabla \underline{\hat{\phi}}\, dR = \int_{\Gamma^{(e)}} \boldsymbol{w}_j \cdot \underline{\boldsymbol{H}}_t\, dS, \tag{2.12}$$

and

$$\left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right) \int_{\Omega^{(e)}} \nabla w_j \cdot \nabla \underline{\hat{\phi}}\, dR - \left(\omega^2 \varepsilon^{(e)} - j\omega\sigma^{(e)}\right) \int_{\Omega^{(e)}} \nabla w_j \cdot \underline{\hat{\boldsymbol{A}}}\, dR = 0. \tag{2.13}$$

2.2 The Finite Element Method Formulation

2.2.1 The Frequency Domain Algorithm

The following formulation develops Eqs. (2.12) and (2.13) derived in section 2.1 for solution in the frequency domain. The formulation for the time domain solution is very similar and is shown in section 2.2.2. Appendix C describes how a fast Fourier transform method can be used to solve time domain problems using the frequency domain response.

Three Dimensional Elements

Discretizing a three dimensional volume into elements can be labourious and prone to mistakes, depending upon the complexity of the elements used. A cube or a rectangular hexahedron is the easiest shape with which to assemble three dimensional structures; however in a curved or irregular volume, the discretization errors will be very large and the potential functions poorly approximated over curved boundaries. It would require very many such elements to model a cylinder or sphere. Tetrahedrons are very versatile in the way they model complex geometric shapes and by using simplex coordinates, volume integration is straight-forward [10], but assembling the elements is a mind-bending task, best left to a graphic artist like M.C. Escher [30].

The isoparametric eight node solid is an element with eight nodes and having six surfaces. It can be imagined by distorting the sides of a cube so that adjacent walls are not perpendicular, and opposite walls are not parallel. One such element is shown in Fig. 2.2 [12].

The flexible geometry of the *isoparametric hexahedron* allows it to be readily used for modelling irregular regions. To make the interpolatory functions and the volume integrals simpler, a natural coordinate system (designated by axes s, t, and r) is used to describe the irregular shaped solid in terms of a cube. This transformation of coordinates facilitates the derivation of the interpolatory functions, and makes the limits of the volume and surface integrals much easier to define. Figure 2.3 shows the solid element in the natural coordinate system. The local node and surface numbering sequences are defined and must be strictly followed when identifying the global node numbers during element assembly. The x,y,z coordinates are described in terms of first order interpolation polynomials in the natural coordinate system. In summation

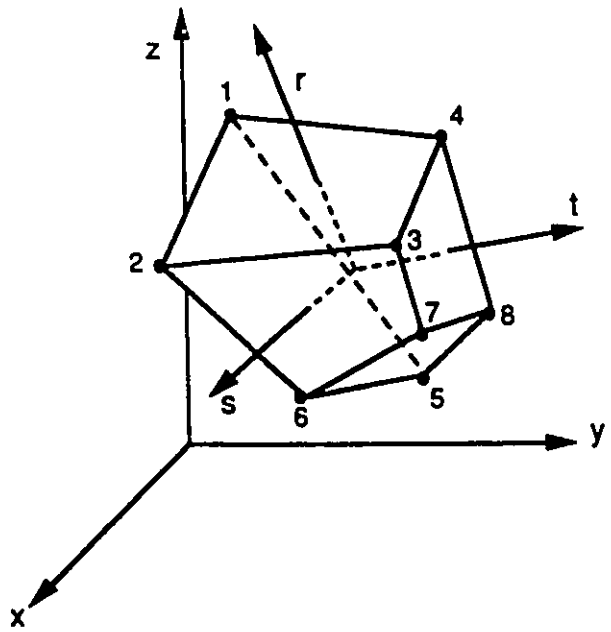


Figure 2.2: The eight node solid in generalized coordinates.

form, we write:

$$\begin{aligned} x &= \sum_{j=1}^{8} N_j^{(e)} x_j^{(e)} \\ y &= \sum_{j=1}^{8} N_j^{(e)} y_j^{(e)} \\ z &= \sum_{j=1}^{8} N_j^{(e)} z_j^{(e)} \end{aligned} \tag{2.14}$$

where $\{x_j, y_j, z_j; j = 1,8\}$ are the local node coordinates of the element 'e' in the general or x,y,z coordinate system, and $\{N_j; j = 1,8\}$ are the first order interpolation functions in terms of variables s, t, r for the element 'e'.

The *interpolatory functions* must be defined such that

$$N_j = \begin{cases} 1, & \text{if node}=j; \\ 0, & \text{otherwise.} \end{cases}$$

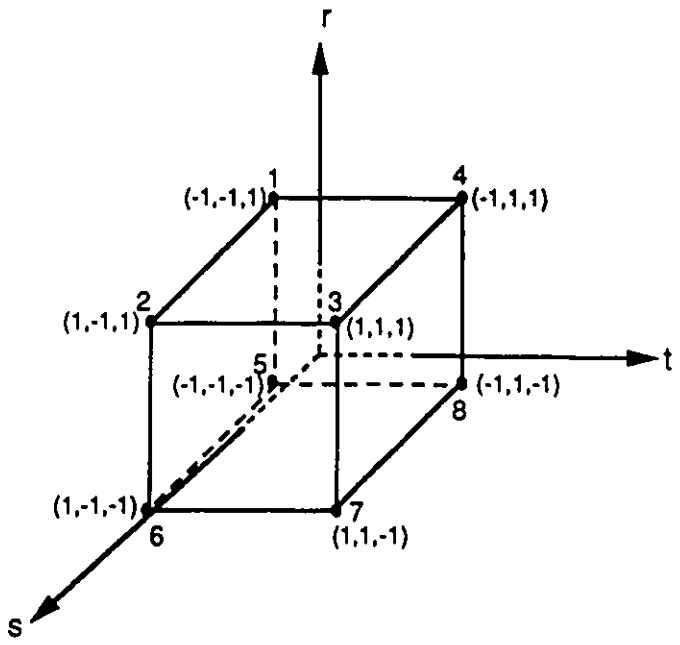


Figure 2.3: An eight node solid in natural coordinates.

The interpolatory functions appropriate for the local node numbering scheme shown in Fig. 2.3 are

$$\begin{aligned}
N_1 &= \tfrac{1}{8}(1-s)(1-t)(1+r) \\
N_2 &= \tfrac{1}{8}(1+s)(1-t)(1+r) \\
N_3 &= \tfrac{1}{8}(1+s)(1+t)(1+r) \\
N_4 &= \tfrac{1}{8}(1-s)(1+t)(1+r) \\
N_5 &= \tfrac{1}{8}(1-s)(1-t)(1-r) \\
N_6 &= \tfrac{1}{8}(1+s)(1-t)(1-r) \\
N_7 &= \tfrac{1}{8}(1+s)(1+t)(1-r) \\
N_8 &= \tfrac{1}{8}(1-s)(1+t)(1-r).
\end{aligned} \tag{2.15}$$

The Basis and Weighting Functions

The potential functions can be expressed by means of shape or basis functions. We will use the same first order interpolation polynomials used above as basis functions

for describing the potentials; thus in summation form:

$$\begin{aligned}
\underline{\hat{A}}_x &= \sum_{j=1}^{8} N_j^{(e)} \underline{\hat{A}}_{xj}^{(e)} \\
\underline{\hat{A}}_y &= \sum_{j=1}^{8} N_j^{(e)} \underline{\hat{A}}_{yj}^{(e)} \\
\underline{\hat{A}}_z &= \sum_{j=1}^{8} N_j^{(e)} \underline{\hat{A}}_{zj}^{(e)} \\
\underline{\hat{\phi}} &= \sum_{j=1}^{8} N_j^{(e)} \underline{\hat{\phi}}_j^{(e)}
\end{aligned} \tag{2.16}$$

where $\{\underline{\hat{A}}_{xj}, \underline{\hat{A}}_{yj}, \underline{\hat{A}}_{zj}, \underline{\hat{\phi}}_j; j = 1,8\}$ are the complex field potential values for the nodes of element 'e'. Written in matrix formulation, they are

$$\underline{\hat{\phi}} = [\, N_1 \quad N_2 \quad N_3 \quad N_4 \quad N_5 \quad N_6 \quad N_7 \quad N_8 \,] \begin{Bmatrix} \underline{\hat{\phi}}_1 \\ \underline{\hat{\phi}}_2 \\ \underline{\hat{\phi}}_3 \\ \underline{\hat{\phi}}_4 \\ \underline{\hat{\phi}}_5 \\ \underline{\hat{\phi}}_6 \\ \underline{\hat{\phi}}_7 \\ \underline{\hat{\phi}}_8 \end{Bmatrix} = [N][\underline{\hat{\phi}}], \tag{2.17}$$

and

$$\begin{aligned}
\underline{\hat{\bar{A}}} &= \underline{\hat{A}}_x \boldsymbol{x} + \underline{\hat{A}}_y \boldsymbol{y} + \underline{\hat{A}}_z \boldsymbol{z} \\
&= [N][\underline{\hat{A}}_x]\boldsymbol{x} + [N][\underline{\hat{A}}_y]\boldsymbol{y} + [N][\underline{\hat{A}}_z]\boldsymbol{z} \\
&= [N]\,([\underline{\hat{A}}_x] \quad [\underline{\hat{A}}_y] \quad [\underline{\hat{A}}_z]) = \begin{pmatrix} [N] & 0 & 0 \\ 0 & [N] & 0 \\ 0 & 0 & [N] \end{pmatrix} \begin{Bmatrix} [\underline{\hat{A}}_x] \\ [\underline{\hat{A}}_y] \\ [\underline{\hat{A}}_z] \end{Bmatrix}.
\end{aligned} \tag{2.18}$$

The Galerkin–weighted residual method uses weighting functions which are the same as the basis functions, that is, $w_j = N_j$. The scalar weighting function is written in vector notation as:

$$[w] = [N]^T, \tag{2.19}$$

and the vector weighting function in matrix notation as:

$$[\boldsymbol{w}] = \begin{pmatrix} [N]^T & 0 & 0 \\ 0 & [N]^T & 0 \\ 0 & 0 & [N]^T \end{pmatrix} \begin{pmatrix} [1] \\ [1] \\ [1] \end{pmatrix}. \tag{2.20}$$

By definition (Eq. 2.16), $\underline{\hat{\phi}} = \sum_{j=1}^{8} N_j \underline{\hat{\phi}}_j$, and by extension of this and the fact that a derivative is a linear operator, we see that:

$$\begin{aligned} \frac{\partial \underline{\hat{\phi}}}{\partial x} &= \sum_{j=1}^{8} \frac{\partial N_j \underline{\hat{\phi}}_j}{\partial x} \\ &= \sum_{j=1}^{8} \frac{\partial N_j}{\partial x} \underline{\hat{\phi}}_j, \qquad \text{since } \underline{\hat{\phi}}_j \text{ is a constant.} \end{aligned}$$

The basis and weighting interpolatory functions have been written in terms of the natural coordinates, but their partial derivatives with respect to the spatial displacements must be performed in terms of the generalized (x, y, z) coordinates. The derivative operators in the generalized system must be transformed to the natural system. Using the chain rule, we expand the partial derivatives as:

$$\begin{Bmatrix} \frac{\partial \phi}{\partial s} \\ \frac{\partial \phi}{\partial t} \\ \frac{\partial \phi}{\partial r} \end{Bmatrix} = \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \\ \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \end{pmatrix} \begin{Bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{Bmatrix},$$

$$\text{or,} \qquad K = ML,$$

where K and L are vectors, and M is a matrix. We can solve for L by solving $L = M^{-1}K$, or $L = \frac{1}{|M|} adjoint(M)K$. Thus, we see that the partial derivatives of ϕ with respect to x, y, and z are given by

$$\begin{Bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{Bmatrix} = \frac{[m]}{|M|} \begin{Bmatrix} \frac{\partial \phi}{\partial s} \\ \frac{\partial \phi}{\partial t} \\ \frac{\partial \phi}{\partial r} \end{Bmatrix},$$

where the $adjoint(M)$ is

$$[m] = \begin{pmatrix} \frac{\partial y}{\partial t}\frac{\partial z}{\partial r} - \frac{\partial y}{\partial r}\frac{\partial z}{\partial t} & \frac{\partial y}{\partial r}\frac{\partial z}{\partial s} - \frac{\partial y}{\partial s}\frac{\partial z}{\partial r} & \frac{\partial y}{\partial s}\frac{\partial z}{\partial t} - \frac{\partial y}{\partial t}\frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial r}\frac{\partial z}{\partial t} - \frac{\partial x}{\partial t}\frac{\partial z}{\partial r} & \frac{\partial x}{\partial s}\frac{\partial z}{\partial r} - \frac{\partial x}{\partial r}\frac{\partial z}{\partial s} & \frac{\partial x}{\partial t}\frac{\partial z}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial z}{\partial t} \\ \frac{\partial x}{\partial t}\frac{\partial y}{\partial r} - \frac{\partial x}{\partial r}\frac{\partial y}{\partial t} & \frac{\partial x}{\partial r}\frac{\partial y}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial y}{\partial r} & \frac{\partial x}{\partial s}\frac{\partial y}{\partial t} - \frac{\partial x}{\partial t}\frac{\partial y}{\partial s} \end{pmatrix}. \tag{2.21}$$

The $determinant(M)$ is defined as:

$$|M| = m_{11}\frac{\partial x}{\partial s} + m_{21}\frac{\partial y}{\partial s} + m_{31}\frac{\partial z}{\partial s},$$

where m_{11}, m_{21}, and m_{31} are terms in $[m]$.

Matrix elements

It is now possible to write out each term of the Eqs. (2.12) and (2.13), using the matrix form of the basis and weighting functions. Thus the first term of Eq. (2.12) is

$$\frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} (\nabla \times \boldsymbol{w}_j) \cdot \left(\nabla \times \hat{\underline{A}}\right) dR = \int_{\Omega^{(e)}} [\underline{a}'] \begin{pmatrix} [\hat{\underline{A}}_x] \\ [\hat{\underline{A}}_x] \\ [\hat{\underline{A}}_x] \end{pmatrix} dR,$$

where $[\underline{a}'] = \frac{1}{\mu^{(e)}}[a]$ and the terms of $[a]$ are as follows:

$$[a] = \tag{2.22}$$
$$\begin{pmatrix} \left[\frac{\partial N}{\partial z}\right]^T \left[\frac{\partial N}{\partial z}\right] + \left[\frac{\partial N}{\partial y}\right]^T \left[\frac{\partial N}{\partial y}\right] & -\left[\frac{\partial N}{\partial y}\right]^T \left[\frac{\partial N}{\partial x}\right] & -\left[\frac{\partial N}{\partial x}\right]^T \left[\frac{\partial N}{\partial x}\right] \\ -\left[\frac{\partial N}{\partial x}\right]^T \left[\frac{\partial N}{\partial y}\right] & \left[\frac{\partial N}{\partial x}\right]^T \left[\frac{\partial N}{\partial x}\right] + \left[\frac{\partial N}{\partial z}\right]^T \left[\frac{\partial N}{\partial z}\right] & -\left[\frac{\partial N}{\partial z}\right]^T \left[\frac{\partial N}{\partial y}\right] \\ -\left[\frac{\partial N}{\partial x}\right]^T \left[\frac{\partial N}{\partial z}\right] & -\left[\frac{\partial N}{\partial y}\right]^T \left[\frac{\partial N}{\partial z}\right] & \left[\frac{\partial N}{\partial x}\right]^T \left[\frac{\partial N}{\partial x}\right] + \left[\frac{\partial N}{\partial y}\right]^T \left[\frac{\partial N}{\partial y}\right] \end{pmatrix}.$$

Note that $[N]^T$ represents the transpose matrix of $[N]$.

The second term of Eq. (2.12) is

$$-\left(\omega^2 \varepsilon^{(e)} - j\omega\sigma^{(e)}\right) \int_{\Omega^{(e)}} \boldsymbol{w}_j \cdot \hat{\underline{A}}\, dR = \int_{\Omega^{(e)}} [\underline{b}'] \begin{pmatrix} [\hat{\underline{A}}_x] \\ [\hat{\underline{A}}_x] \\ [\hat{\underline{A}}_x] \end{pmatrix} dR,$$

where $[\underline{b}'] = \left(-\omega^2 \varepsilon^{(e)} + j\omega\sigma^{(e)}\right)[b]$ and $[b]$ is

$$[b] = \begin{pmatrix} [N]^T[N] & 0 & 0 \\ 0 & [N]^T[N] & 0 \\ 0 & 0 & [N]^T[N] \end{pmatrix}. \tag{2.23}$$

The third term of Eq. (2.12) is

$$\left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right) \int_{\Omega^{(e)}} \boldsymbol{w}_j \cdot \nabla\hat{\underline{\phi}}\, dR = \int_{\Omega^{(e)}} [\underline{c}'][\hat{\underline{\phi}}]\, dR,$$

where $[\underline{c}'] = \left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right)[c]$ and $[c]$ is

$$[c] = \begin{pmatrix} [N]^T \left[\frac{\partial N}{\partial x}\right] \\ [N]^T \left[\frac{\partial N}{\partial y}\right] \\ [N]^T \left[\frac{\partial N}{\partial z}\right] \end{pmatrix}. \tag{2.24}$$

The first term of Eq. (2.13) is

$$\left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right) \int_{\Omega^{(e)}} \nabla w_j \cdot \nabla\hat{\underline{\phi}}\, dR = \int_{\Omega^{(e)}} [\underline{d}'][\hat{\underline{\phi}}]\, dR,$$

where $[\underline{d}'] = \left(\sigma^{(e)} + j\omega\varepsilon^{(e)}\right)[d]$ and $[d]$ is

$$[d] = \left(\left[\frac{\partial N}{\partial x}\right]^T\left[\frac{\partial N}{\partial x}\right] + \left[\frac{\partial N}{\partial y}\right]^T\left[\frac{\partial N}{\partial y}\right] + \left[\frac{\partial N}{\partial z}\right]^T\left[\frac{\partial N}{\partial z}\right]\right). \tag{2.25}$$

The second term of Eq. (2.13) is

$$-\left(\omega^2\varepsilon^{(e)} - j\omega\sigma^{(e)}\right)\int_{\Omega^{(e)}} \nabla w_j \cdot \underline{\hat{A}}\, dR = \int_{\Omega^{(e)}} [e']\begin{pmatrix}[\hat{A}_x]\\ [\hat{A}_x]\\ [\hat{A}_x]\end{pmatrix} dR,$$

where $[\underline{e}'] = \left(-\omega^2\varepsilon^{(e)} + j\omega\sigma^{(e)}\right)[e]$ and $[e]$ is

$$[e] = \begin{pmatrix}\left[\frac{\partial N}{\partial x}\right]^T[N] & \left[\frac{\partial N}{\partial y}\right]^T[N] & \left[\frac{\partial N}{\partial z}\right]^T[N]\end{pmatrix}. \tag{2.26}$$

The right hand side term of Eq. (2.12) is

$$\int_{\Gamma^{(e)}} \boldsymbol{w}_j \cdot \underline{\boldsymbol{H}}_t\, dS = \int_{\Gamma^{(e)}} [\underline{f}]\, dS,$$

where $[\underline{f}]$ is

$$\begin{pmatrix}[N]^T\left(\underline{H}_y n_z - \underline{H}_z n_y\right)\\ [N]^T\left(\underline{H}_z n_x - \underline{H}_x n_z\right)\\ [N]^T\left(\underline{H}_x n_y - \underline{H}_y n_x\right)\\ [0]\end{pmatrix}. \tag{2.27}$$

The values for $\mu^{(e)}$, $\sigma^{(e)}$, and $\varepsilon^{(e)}$ are specified when defining the problem geometry. The magnetic field, $\underline{\boldsymbol{H}}_0 = \underline{H}_x\boldsymbol{x} + \underline{H}_y\boldsymbol{y} + \underline{H}_z\boldsymbol{z}$ must also be specified for each subsurface $\Gamma^{(e)}$ on the exterior of $\Omega^{(e)}$. For solution at a particular frequency, $\underline{\boldsymbol{H}}$ is assumed to be $\underline{\boldsymbol{H}} = \underline{\boldsymbol{H}}_0 e^{j\omega t}$.

Numerical Integration

Integration over the volume $\Omega^{(e)}$ is performed in the natural (s,t,r) coordinate system because the limits of integration are simple $(x:[-1,1], y:[-1,1], z:[-1,1])$. It is necessary to change the variable of integration from a differential volume in one system $(dxdydz)$ to the other $(dsdtdr)$ using the Jacobian determinant. Thus

$$dR(x,y,z) = dR(s,t,r) = |J|dsdtdr,$$

where $|J|$ is the Jacobian determinant, and

$$|J| = m_{11}\frac{\partial x}{\partial s} + m_{21}\frac{\partial y}{\partial s} + m_{31}\frac{\partial z}{\partial s};$$

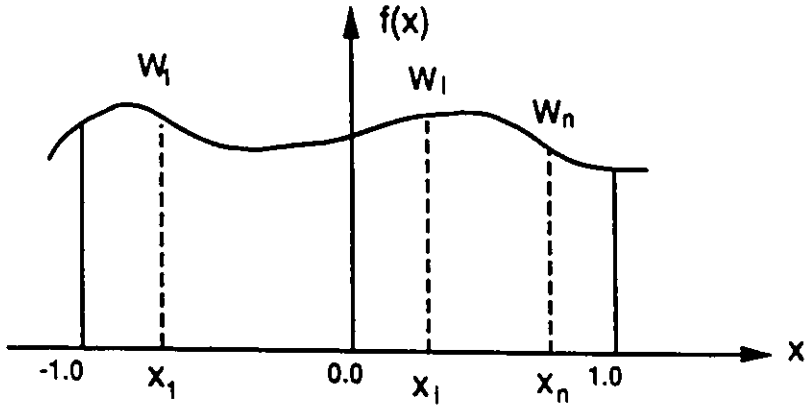


Figure 2.4: An arbitrary function $f(x)$ integrated over the interval $[-1, 1]$ using the Gauss quadrature method

the terms m_{ij} having been defined previously in Eq. (2.21).

The terms of Eqs. (2.12) and (2.13) are fully integrable (no singularities) over the sub-domains $\Omega^{(e)}$, but due to the complexity added by the transformation of coordinates, numerical integration is preferred over evaluating the integrals analytically.

The Gauss quadrature method is ideal for the numerical evaluation of integrals generated by the finite element method. A third order polynomial function can be evaluated exactly over the limits [-1,1] by summing the function at two points $(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ and multiplying by two weighting factors $(1, 1)$. The method can be extended to any polynomial $f(x)$ of order $2n-1; n > 0$ (An example is shown in Fig. 2.4) by using the general rule:

$$\begin{aligned} \int_{-1}^{1} f(x)\,dx &= \int_{-1}^{1} \left[a_0 + a_1 x + a_2 x^2 + \cdots + a_{2n-1} x^{2n-1}\right] dx \\ &= \sum_{i=1}^{n} W_i f(x_i), \end{aligned}$$

where the appropriate locations and weighting factors are found in Gauss quadrature data tables [17]. Integration of a third order polynomial function of three variables over a volume requires evaluation at just eight points using the Gauss quadrature method.

We can write the full expression for the integration over one solid element for the first term in Eq. (2.12) as:

$$\frac{1}{\mu^{(e)}} \int_{\Omega^{(e)}} (\nabla \times \boldsymbol{w}_j) \cdot \left(\nabla \times \underline{\hat{A}}\right) dR \tag{2.28}$$

$$= \frac{1}{\mu^{(e)}} \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \left([a(s_i^{(e)}, t_j^{(e)}, r_k^{(e)})] \left(\frac{[m(s_i^{(e)}, t_j^{(e)}, r_k^{(e)})]}{|M(s_i^{(e)}, t_j^{(e)}, r_k^{(e)})|} \right)^2 |J| W_i W_j W_k \right) \begin{Bmatrix} [\hat{A}_x] \\ [\hat{A}_x] \\ [\hat{A}_x] \end{Bmatrix}.$$

The term $[m]/|M|$ is squared because both the derivatives of $\underline{\hat{A}}$ and $\boldsymbol{w}$ must be transformed. The other terms of Eqs. (2.12) and (2.13) can be described similarly.

The right hand side term of Eq. (2.12) is a surface integral which is evaluated over one face of the solid element in the s, t, r domain. One variable (s, t, or r) is kept constant during the integration. The Jacobian determinant becomes to the ratio of the area of the surface in the x, y, z domain over the area of the surface in the s, t, r domain (which always equals four). The function only needs to be evaluated at four points when using the Gauss quadrature method.

The Final System of Equations

Once the integrals have been evaluated, the resulting contributions are assembled into a matrix equation for each element which can be written in compact form as:

$$[\underline{S}^{(e)}] \begin{pmatrix} [\hat{A}_x^{(e)}] \\ [\hat{A}_y^{(e)}] \\ [\hat{A}_z^{(e)}] \\ [\hat{\phi}^{(e)}] \end{pmatrix} = \begin{pmatrix} [\underline{F}_x^{(e)}] \\ [\underline{F}_y^{(e)}] \\ [\underline{F}_z^{(e)}] \\ [0] \end{pmatrix}, \tag{2.29}$$

where

$$[\underline{S}^{(e)}] = \begin{pmatrix} \underline{a}'_{11} + \underline{b}'_{11} & \underline{a}'_{12} & \underline{a}'_{13} & \underline{c}'_1 \\ \underline{a}'_{21} & \underline{a}'_{22} + \underline{b}'_{22} & \underline{a}'_{23} & \underline{c}'_2 \\ \underline{a}'_{31} & \underline{a}'_{32} & \underline{a}'_{33} + \underline{b}'_{33} & \underline{c}'_3 \\ \underline{e}'_1 & \underline{e}'_2 & \underline{e}'_3 & \underline{d}' \end{pmatrix}$$

is a 32×32 matrix, and the terms $\underline{a}_{ij}$, $\underline{b}_{ij}$, $\underline{c}_i$, $\underline{d}$, $\underline{e}_j$ have been defined previously in Eqs. (2.22) to (2.26) above. Once the contributions from each element have been computed, and summed together in a system of global matrix equations of order $4N \times 4N$:

$$[\underline{GS}] \begin{pmatrix} [\hat{A}_x] \\ [\hat{A}_y] \\ [\hat{A}_z] \\ [\hat{\phi}] \end{pmatrix} = \begin{pmatrix} [\underline{GF}_x] \\ [\underline{GF}_y] \\ [\underline{GF}_z] \\ [0] \end{pmatrix}, \tag{2.30}$$

then the system of equations can be solved for the unknowns
$\{\underline{\hat{A}}_{x1}, \underline{\hat{A}}_{x2}, \cdots \underline{\hat{A}}_{xN}; \underline{\hat{A}}_{y1}, \underline{\hat{A}}_{y2}, \cdots \underline{\hat{A}}_{yN}; \underline{\hat{A}}_{x1}, \underline{\hat{A}}_{x2}, \cdots \underline{\hat{A}}_{xN}; \underline{\hat{\phi}}_1, \underline{\hat{\phi}}_2, \cdots \underline{\hat{\phi}}_N\}$.

2.2.2 The Time Domain Algorithm

Development of Eq. (2.10) and (2.11) for solution in the time domain by the finite element method yields a final system of equations of the following form (compact form):

$$[R]\frac{d^2}{dt^2}\begin{pmatrix}[\hat{A}_x]\\[\hat{A}_y]\\[\hat{A}_z]\\[\hat{\phi}]\end{pmatrix} + [T]\frac{d}{dt}\begin{pmatrix}[\hat{A}_x]\\[\hat{A}_y]\\[\hat{A}_z]\\[\hat{\phi}]\end{pmatrix} + [S]\begin{pmatrix}[\hat{A}_x]\\[\hat{A}_y]\\[\hat{A}_z]\\[\hat{\phi}]\end{pmatrix} = \begin{pmatrix}[F_x(t)]\\[F_y(t)]\\[F_z(t)]\\[0]\end{pmatrix}. \tag{2.31}$$

where

$$[R] = \begin{bmatrix} b'''_{11} & 0 & 0 & 0 \\ 0 & b'''_{22} & 0 & 0 \\ 0 & 0 & b'''_{33} & 0 \\ e'''_1 & e'''_2 & e'''_3 & 0 \end{bmatrix}, \quad \text{and} \quad \begin{matrix}[b'''] = \varepsilon^{(e)}[b] \\ [e'''] = \varepsilon^{(e)}[e]\end{matrix}; \tag{2.32}$$

$$[T] = \begin{bmatrix} b''_{11} & 0 & 0 & c'''_1 \\ 0 & b''_{22} & 0 & c'''_2 \\ 0 & 0 & b''_{33} & c'''_3 \\ e''_1 & e''_2 & e''_3 & d''' \end{bmatrix}, \quad \text{and} \quad \begin{matrix}[b''] = \sigma^{(e)}[b] \\ [c'''] = \varepsilon^{(e)}[c] \\ [d'''] = \varepsilon^{(e)}[d] \\ [e''] = \sigma^{(e)}[e]\end{matrix}; \tag{2.33}$$

$$[S] = \begin{bmatrix} a''_{11} & a''_{12} & a''_{13} & c''_1 \\ a''_{21} & a''_{22} & a''_{23} & c''_2 \\ a''_{31} & a''_{32} & a''_{33} & c''_3 \\ 0 & 0 & 0 & d'' \end{bmatrix}, \quad \text{and} \quad \begin{matrix}[a''] = \frac{1}{\mu^{(e)}}[a] \\ [c''] = \sigma^{(e)}[c] \\ [d''] = \sigma^{(e)}[d]\end{matrix}. \tag{2.34}$$

The matrices $[S]$, $[T]$, and $[R]$ are 32×32 matrices, and the terms $a_{ij}, b_{ij}, c_i, d, e_j$ have been defined previously in Eqs. (2.22) to (2.26) above. These matrices are combined together as $[GS]$, $[GT]$, and $[GR]$, which are $4N \times 4N$ matrices.

The system of differential equations can be solved using a routine designed to solve large systems of differential equations. However, notice that $[R]$ is a singular matrix. This is because there is no second derivative of ϕ with respect to time. This condition may pose a problem in trying to reduce the matrices.

First order ODE solver

Many popular ordinary differential equation solvers use single or multi-stepping techniques, and require that the system of equations be in the form [17]:

$$\dot{u} = f(t, u)$$

where $\dot{u} = \frac{d}{dt}u$. Thus, we must reduce our system of second order differential equations to a system of first order ones. This involves a substitution of variables ($B = \dot{A}$) and a matrix inversion.

By making several simplifications in the original set of equations, the final system of equations can be reduced to a non-singular set of first order differential equations. These two assumptions are

1. the displacement current is negligible everywhere (done by making $\epsilon = 0$ everywhere), thus $\nabla \times \frac{1}{\mu}\boldsymbol{B} = \boldsymbol{J}_c$.
2. the gradient of the scalar electric potential is zero ($\nabla\phi = 0$), thus $\boldsymbol{E} = -\frac{d}{dt}\boldsymbol{A}$.

So that we now have:

$$[T]\left\{\dot{\boldsymbol{A}}\right\} + [S]\{\boldsymbol{A}\} - f(t) = 0 \tag{2.35}$$

We can justify the above assumptions if the problem we are solving has the following characteristics:

- the region is conductive and the permittivity is that of free-space, and the object is electrically small (for the range of dominant frequencies). This means the solution is restricted to low frequencies and that the term $\frac{d}{dt}\boldsymbol{E}$ becomes negligible compared to the term $\sigma\boldsymbol{E}$. (the dominant type of current is conduction, $\nabla \cdot \boldsymbol{J}_c = 0$),
- the geometry and the incident field are such that there are no disruptions in the conduction current flow (a large disruption in an eddy current implies a buildup of charges around the discontinuity). This means the induced charge distribution will be constant everywhere.

Direct single-step DE solver

Another approach for solving systems of differential equations is a direct single-step algorithm, originally demonstrated by Crank and Nicholson [18], and later refined by Zienkiewicz, Wood, and Hines [19]. This algorithm can be used to solve Eq. (2.31) directly in the form [7]

$$[R]\left\{\begin{matrix}\ddot{\boldsymbol{A}} \\ \ddot{\phi}\end{matrix}\right\} + [T]\left\{\begin{matrix}\dot{\boldsymbol{A}} \\ \dot{\phi}\end{matrix}\right\} + [S]\left\{\begin{matrix}\boldsymbol{A} \\ \phi\end{matrix}\right\} - f(t) = 0.$$

2.3 The Boundary Conditions

Tangential Magnetic Field

The finite element equations derived in section 2.1 are used to solve for the electromagnetic field distribution (in terms of the potential functions) inside the region Ω, which is bounded by Γ. On the right hand side of Eq. (2.10), or (2.12), there is provision for a magnetic field source: $\underline{\boldsymbol{H}}_t$. The direction and phase of the tangential magnetic field must be imposed on the surface regions of the boundary Γ. By placing the outer boundary sufficiently far away, we can neglect contributions on the surface due to the field scattered from the structure (assume they decay to zero).

If there is no phase variation of the field between different physical locations on the surface of the region where the source term is applied, then the field is like the one that would be produced between the plates of a parallel plate capacitor (a TEM field). The object can re-oriented with respect to the problem geometry thus changing the polarization of the illuminating field. Shown in Fig. 2.5 is an example of how the tangential magnetic field boundary condition is applied.

Dirichlet Boundaries

If the conductivity inside an element region $\Omega^{(e)}$ is made infinite, then it can be shown that the vector magnetic potential inside is constant ($\boldsymbol{A} = constant$). The potential distribution inside an element is described by the eight node potentials of the element. If the vector magnetic potentials of the eight nodes are fixed to zero, then the vector magnetic potential inside the element will be zero, and thus we have effectively imposed $\sigma^{(e)} = \infty$.

By making the element infinitesimally thin in one dimension, and placing it next to another regular element, it would force the potentials on that element to zero and produce the effect of an infinitely conducting surface. Thus we can see that by fixing the vector magnetic potential nodes to zero (creating a Dirichlet boundary condition) on a surface[4], we can model an infinitesimally thin, infinitely conducting wall. This simplification will be a good approximation for problems where the metal wall is a very good conductor, and is very thin as compared to its other dimensions. This

[4]The tangential electric field on the conductor must be zero, thus the scalar electric potential must be constant. To keep the algorithm simple, we will arbitrarily make it zero since there are no free charges.

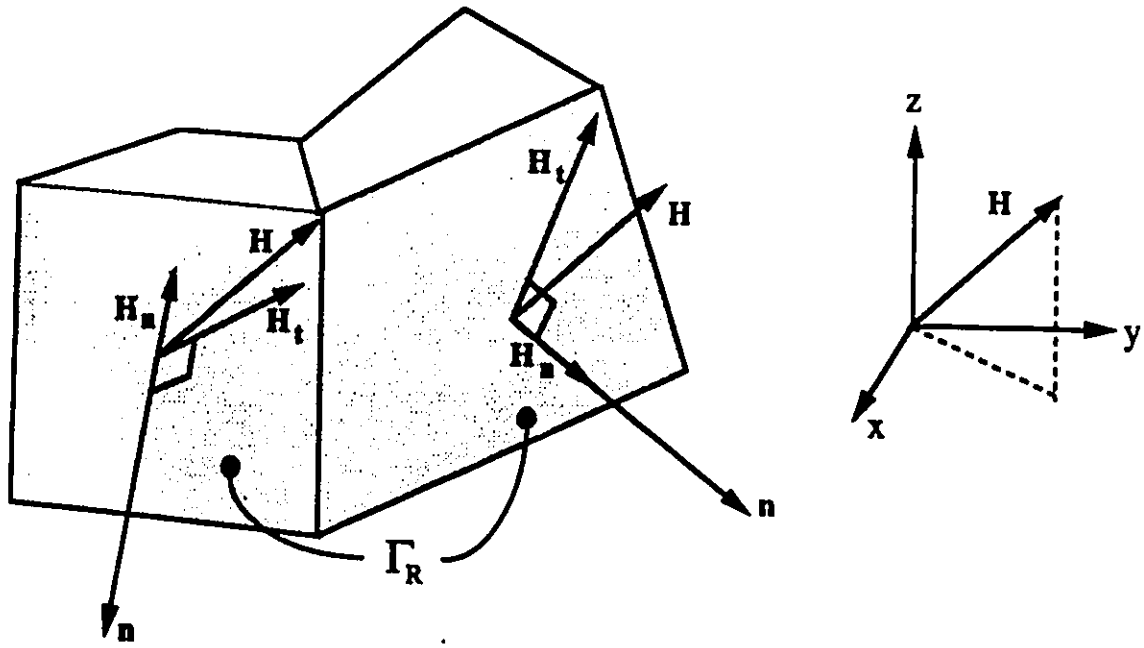
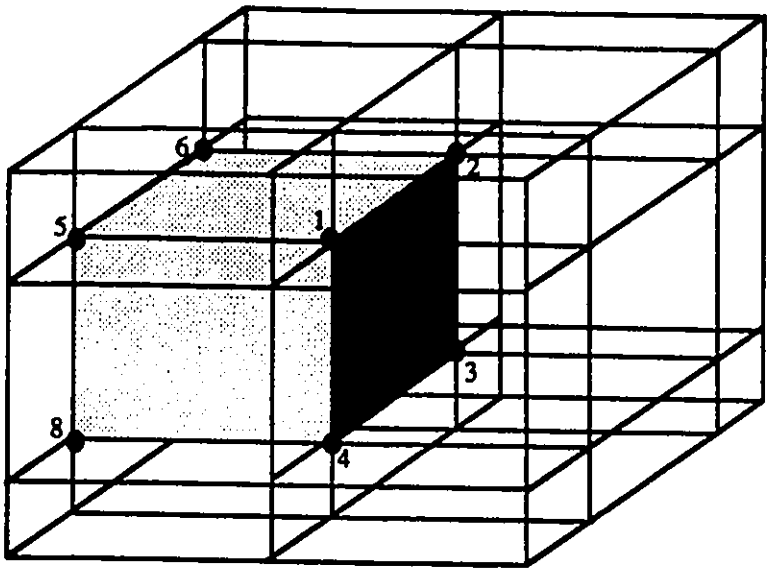


Figure 2.5: The tangential magnetic field on the boundary
The two shaded surfaces represent part of the external boundary of the region of interest. The blank surfaces have adjacent elements which are not shown. $\boldsymbol{H}$ is the incident magnetic field, $\boldsymbol{H}_t$ is the tangential magnetic field, and $\boldsymbol{H}_n$ is the normal magnetic field



This volume is a hexahedron finite element.

This surface has infinite conductivity and is created by fixing the potential of nodes 1 to 4 to zero.

Figure 2.6: An infinitesimally thin, infinitely conducting wall created using a Dirichlet boundary condition

approximation reduces the number of unknown potentials, and the final system of equations is better conditioned. Figure 2.6 is a case where an infinitesimally thin, infinitely conducting wall had been created using a Dirichlet boundary condition.

2.4 Electromagnetic Field Computations

The electric and magnetic field vectors can be computed from the original definitions of $\boldsymbol{A}$ and ϕ by application of the basis functions and their derivatives to the solution values of $\hat{\boldsymbol{A}}$ and $\hat{\phi}$. The potential functions are linear inside each element, thus the first derivative will be constant. Because of this averaging, we can only compute the electromagnetic field for the centre point of each element. It is necessary to use a higher order interpolation to find the field values at the edges of an element.

We can use the algorithms described above to provide information about the distribution of the electromagnetic fields inside the region of interest due to a magnetic

field applied on the boundaries.

2.5 Description of the Computer Program

For every node in the finite element discretized region Ω, there are four degrees of freedom: symbolically, A_x, A_y, A_z, and ϕ. The final system of equations produced after the assembly of the contributions from each node are stored in a $4N \times 4N$ matrix where N is the number of nodes. If Dirichlet boundary conditions are imposed, then the number of unknowns is fewer.

Frequency domain

When the problem is solved in the steady–state, the unknown variables are complex, and require twice the memory real variables. It is preferred to use double precision variables (REAL*8 and COMPLEX*16), to minimize the errors introduced by round-off and discretization, and thus the memory requirement simply for the coefficients of the system of equations is $4N \times 4N \times 16 = 256N^2$ bytes. If the problem being solved has 250 nodes, then there are 1000 unknowns and the memory requirement will be at least 16 Mbytes. A sample discretization of a rectangular solid with 250 nodes into $5 \times 5 \times 10$ nodes gives an indication of the limited resolution available with 16 Mbytes of computer memory.

The FORTRAN coded implementation of the frequency domain finite element algorithm is called *DORS* (*D*ynamics *O*f *R*adiating *S*tructures). The program uses double precision variables and functions and has been written using FORTRAN to run on a Digital Electronic Corporation (DEC) VAX 3500 computer. (Note: a copy of the source code listing has been included in Appendix D.)

The structure of the program can be broken up into five major parts. The five parts are:

1. *geometry data input* – the structure's mesh information (element node numbering and node coordinates, the conductivity and permittivity of the finite element regions, Dirichlet boundaries, exterior boundary faces, the solution frequencies) is read in from a file of filetype *.GEO*.

2. *computation of volume integral contributions* – Gauss quadrature integration (8 points) is used to calculate the contributions of each term of the integral

expressions by summing over the volume of each element, and then combining them to generate the global coefficient matrix.

3. *computation of surface integral contributions* – the tangential magnetic field (the magnetic field on the boundary surface times (cross product) the outward surface normal) is multiplied by the interpolatory functions and integrated over the surface to produce the source vector terms on the right hand side.

4. *solution of the system of equations* – the system of equations stored in matrix form is reduced and solved by a Gaussian elimination algorithm.

5. *output of results* – the complex values of $\underline{A_x}$, $\underline{A_y}$, $\underline{A_z}$, and $\underline{\phi}$ at each node and for each desired frequency are written as real and imaginary components to a file of filetype *.RES*. These values can be used to compute the electric and magnetic fields.

The most suitable technique for solving the system of complex equations is Gaussian elimination applied with partial pivoting. The program DORS uses a pair of routines, ZGECO and ZGESL from LINPACK [20]. The routine ZGECO factors the coefficient matrix using Gaussian elimination and it estimates the condition of the matrix, and then ZGESL solves the system.

A condition number [21] for a system of equations $Ax = b$ is a measure of its nearness to singularity. If cond(A) is large, then A is close to being singular, and thus small changes in A and b will produce large changes in x. For a well conditioned matrix, cond(A) is close to one, and small changes in A and b will only result in corresondingly small changes in x.

The condition number can be seen as a relative error magnification factor, where changes in the right hand side b may cause changes cond(A) times as large in the solution x.

The final system of equations is not symmetric, and regardless of the way the nodes are numbered, there will only be a minimal amount of banding. The finite element method does produce a sparse matrix (very few non-zero elements). Several algorithms for solving systems of sparse equations were examined [22,23,24,25]; however no suitable sparse matrix equation solver was found for large systems of complex numbers.

The program *FIELDING* (the FORTRAN code is included in Appendix D) reads the geometry information from the .GEO file and the potential solutions from the .RES file. It then uses the basis functions and their derivatives to compute the curl of $\boldsymbol{A}$ and the gradient of $\underline{\phi}$ to determine $\boldsymbol{E}$ and $\boldsymbol{H}$ at the centre of each element. The magnitude of each component, and a total field value for the $\underline{E}$ and $\underline{H}$ fields are written to a file of filetype *.CUR*.

Time domain

The time domain algorithm uses only REAL*8 variables and does not use any COMPLEX*16 variables, however a $4N \times 4N$ matrix is required for each of $\frac{d^2}{dt^2}[\boldsymbol{A}, \phi]$, $\frac{d}{dt}[\boldsymbol{A}, \phi]$, and $[\boldsymbol{A}, \phi]$.

The FORTRAN coded implementation of the time domain finite elememt algorithm *APROXHP.FOR* was developed based upon the simplified system of equations discussed in section 2.2.2. There were only three degrees of freedom (A_x, A_y, and A_z) and two matrices ($\frac{d}{dt}[\boldsymbol{A}]$, and $[\boldsymbol{A}]$). This meant that $2 \times (3N \times 3N) = 144N^2$ bytes of computer memory were required to store the system of differential equations.

The structure of the program APROXHP differs from DORS only in the way the global matrices are assembled and solved, and the format of the output. The first order differential equation solver is based on Hamming's modified predictor–corrector method, and it solved the final system in the form

$$\left\{\dot{\boldsymbol{A}}\right\} = [T]^{-1}\left(f(t) - [S]\left\{\boldsymbol{A}\right\}\right)$$

The output of the program are the values of the magnetic field [$\boldsymbol{H}_x$, $\boldsymbol{H}_y$, and $\boldsymbol{H}_z$] as functions of time at the positions specified at the input.

Chapter 3

Results

3.1 Program Validation

Before the programmed formulation (DORS.FOR) can be used to solve an arbitrary diffusion or penetration problem, it must first be properly tested to verify its operation. The program is validated by comparison with the exact solution of a representative example with the finite element method solution. It is also important to determine what limits and restrictions must be applied at the input to obtain valid and acceptable output. The problem geometries used to validate the program were designed to approach one and two dimensional cases, so that simple analytical solutions were available for comparison, but also so that the test geometries were easy to generate.

3.1.1 Diffusion Problem in One Dimension

A simple problem to solve is the penetration or diffusion of a magnetic field into a slab of conducting material which extends infinitely in the y and z directions, but has finite thickness in the x direction. If the magnetic field source has no x components, then the analytical solution is a function of x only. We will model it as a three dimensional problem by creating a rectangular solid mesh whose large dimensions are ten times the thickness. The slab geometry is shown in Fig. 3.1. It is assumed that the field variation in the y and z directions is negligibly small, so that large linear elements will adequately model the field in those dimensions.

In Fig. 3.2, the results of the FEM solution for the diffusion of a magnetic field into a 1.0 mm slab (a =0.5 mm) of material with conductivity 5.6×10^7 S/m (copper) at a frequency 1.0 MHz are plotted for different numbers of elements N in the x

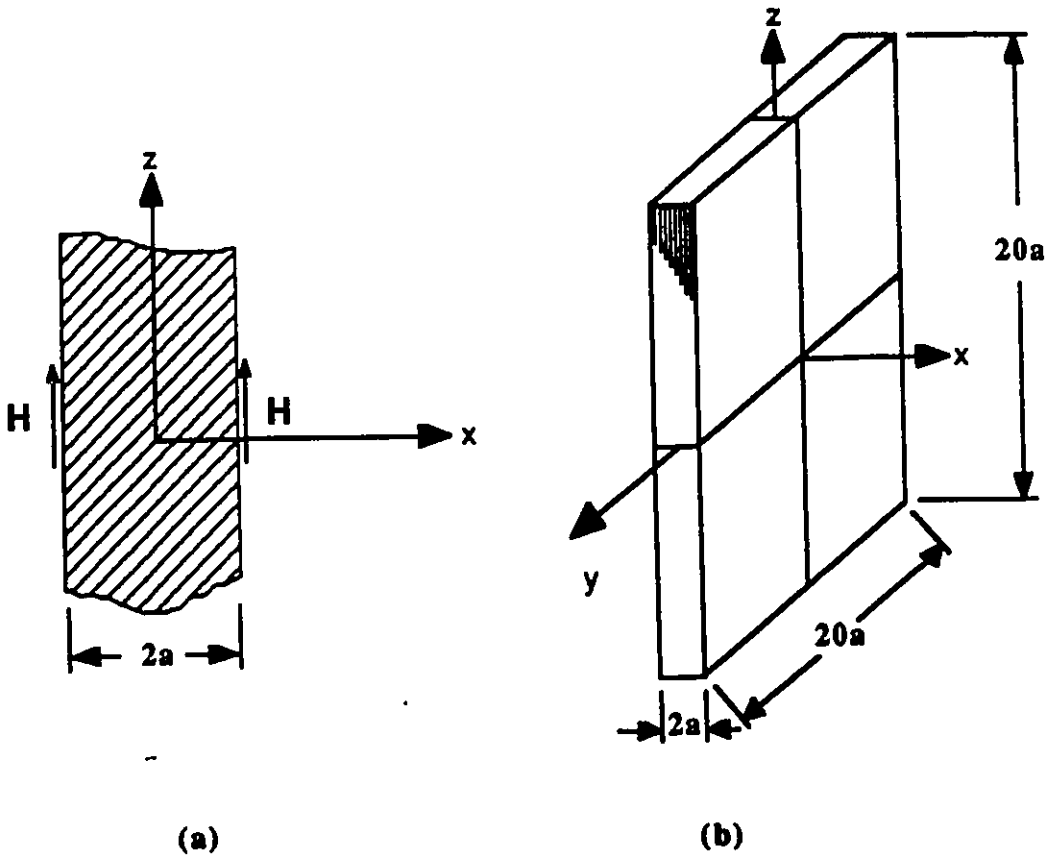


Figure 3.1: Geometry of the slab
(a) cross-section showing finite thickness in x–dimension, (b) orthographic projection of the 3–D geometry used by DORS

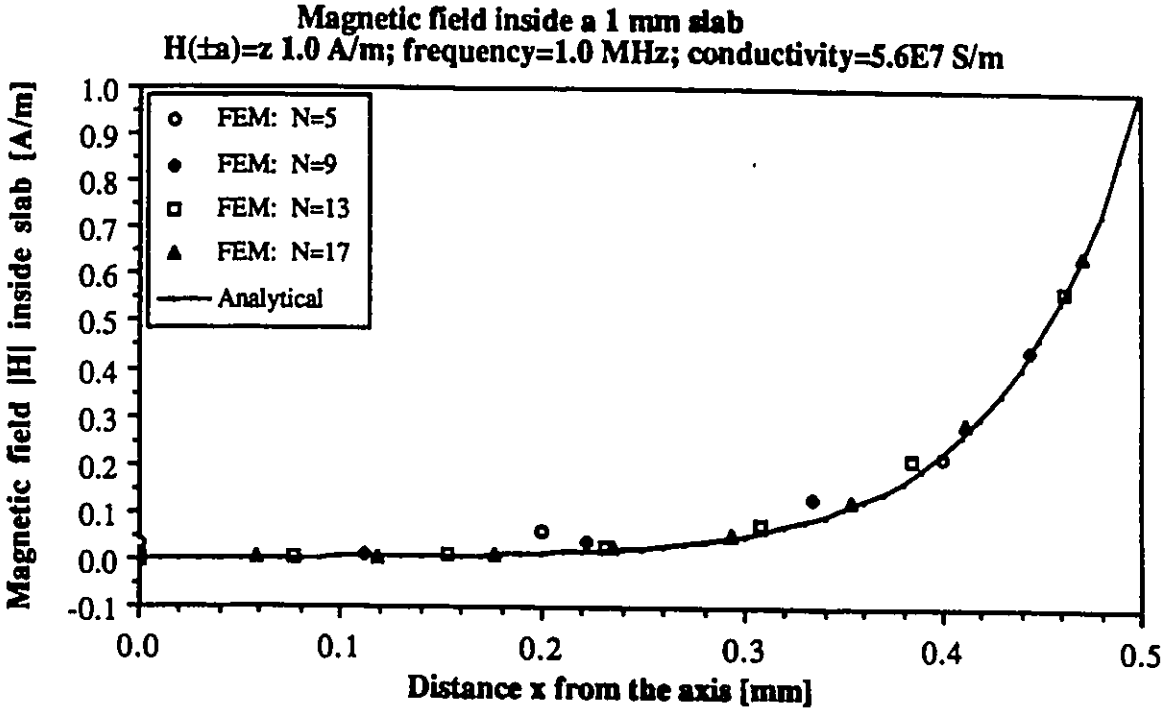


Figure 3.2: Convergence of the 1-D diffusion problem as a function of the number of elements in the x-dimension (due to the symmetry about $x = 0$, only half of the solution is shown)

dimension. The incident field strength at the slab boundary was 1.0 A/m. Only the results for $x = 0$ to a are shown because of the symmetry of the solution. Also note that the FEM solves field at the centre of each element. The estimate of the matrix condition number was typically 10^{24}. We observe that as the number of elements in the cross-section increases, the results converge to the analytical solution, which has also been plotted in Fig. 3.2.

3.1.2 Diffusion Problems in Two Dimensions

Solid Cylinder

The cross-sectional view and the orthographic projection of a cylindrical rod discretized into finite elements, are shown in Fig. 3.3. This geometry is an example of how a region with curved boundaries can be modelled using a small number of eight node hexahedron elements.

The penetration of a magnetic field into an infinitely long cylinder (along the

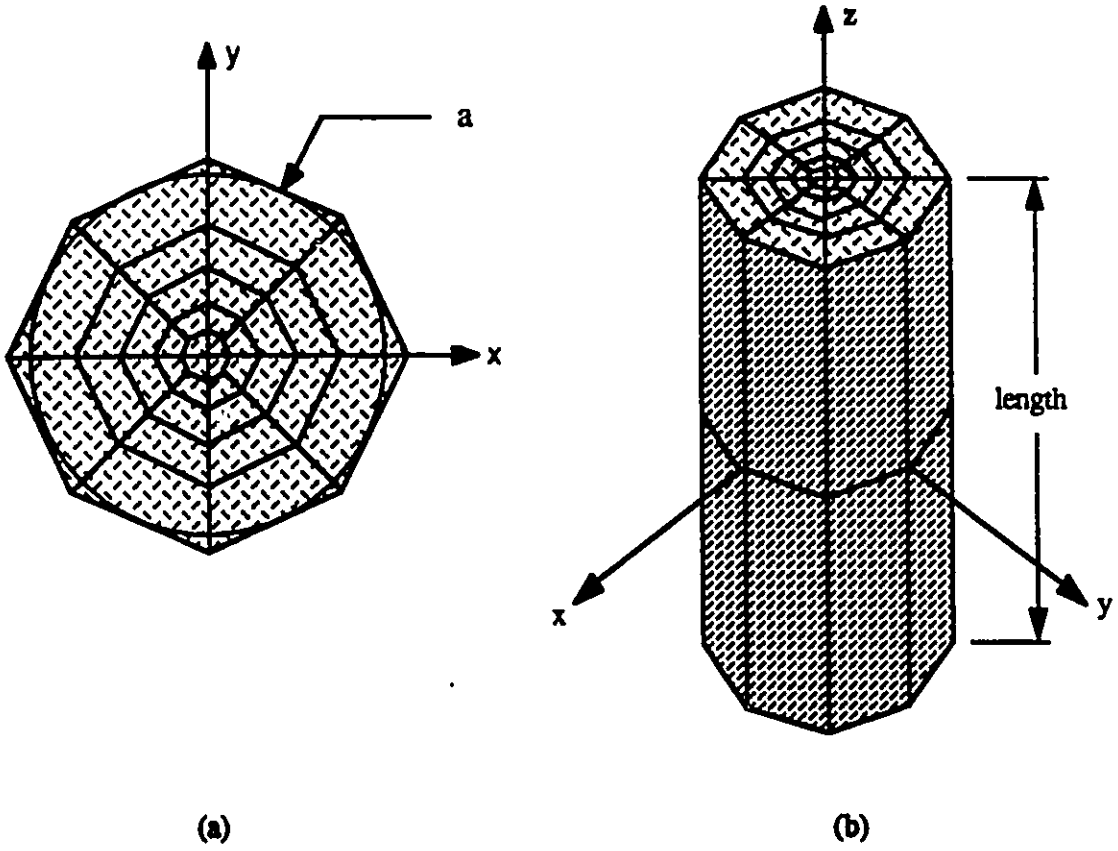


Figure 3.3: Rod geometry of the finite element mesh used to approximate a circular cylinder
(a) cross-section (radius a), (b) orthographic projection of the 3–D geometry used by DORS

z–axis) can be solved using two dimensional analysis. If the cylinder is round, then the field variation is dependent only upon the radius, and can be solved analytically using cylindrical coordinates. For the case where the applied tangential field is along the axis of the cylinder, the analytical solution is

$$\boldsymbol{H}(r,\omega) = H_0 \frac{J_0(\sqrt{j\omega\mu\sigma}r)}{J_0(\sqrt{j\omega\mu\sigma}a)}\boldsymbol{z},$$

where H_0 is the applied axial magnetic field on the surface of the cylinder, and J_0 is a zero order Bessel function of the first kind. The radius of the cylinder (a), is taken to be the radial length which is perpendicular to the outer element surface [see Fig. 3.3(a)].

If a cylinder is very long compared to its diameter, then in the middle length of the rod, the magnetic field will be invariant to changes in ϕ and z. If the incident magnetic field has only an axial component, then the induced current will flow circumferentially around the cylinder, but if there is a circumferential component of the magnetic field, then the current will flow along the axis, and will be disrupted by the discontinuities at the ends of the rod, thus the 2–D approximation is no longer valid. Only the case where the applied field is parallel to the axis was analyzed.

The FEM solution and the analytical solution of the magnitude of the magnetic field as a function of r are shown together in Fig. 3.4 for a 9.24 cm diameter (a = 4.62 cm) rod of conductivity 10^6 S/m for several different frequencies.

The raised contour mesh in Fig. 3.5 is the magnetic field distribution inside a square cylinder. Only one quarter of the cross–section is shown because the solution is symmetrical. The square rod has a dimension 1.0 mm and is made of a material with conductivity 10^6 S/m. The surface magnetic field was 1.0 A/m at 1.0 MHz in the direction of the cylinder axis.

Hollow Cylinder

A hollow cylinder or pipe is a simple two dimensional enclosure. To solve the problem with FEM, a long cylinder was created by extending the cross–section shown in Fig. 3.6 to produce long elements. The bold line denotes the pipe wall (thickness d), which was 0.924 mm thick and was formed with four layers of elements, whose conductivity was 10^6 S/m (see the inset enlargement).

The pipe radius (a) is 27.76 cm and its length is 10.0 m. The magnetic field excitation is applied 92.39 cm away from the axis of the pipe in the direction of

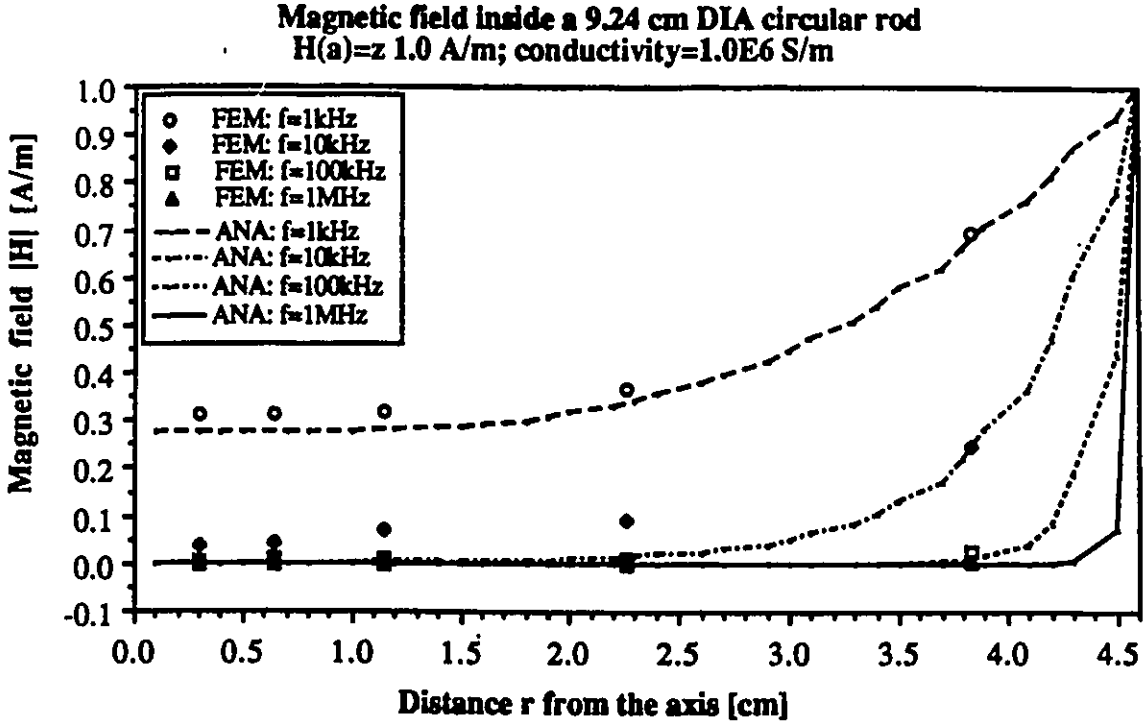


Figure 3.4: Magnetic field diffusion in a circular rod for different frequencies (solved by DORS)

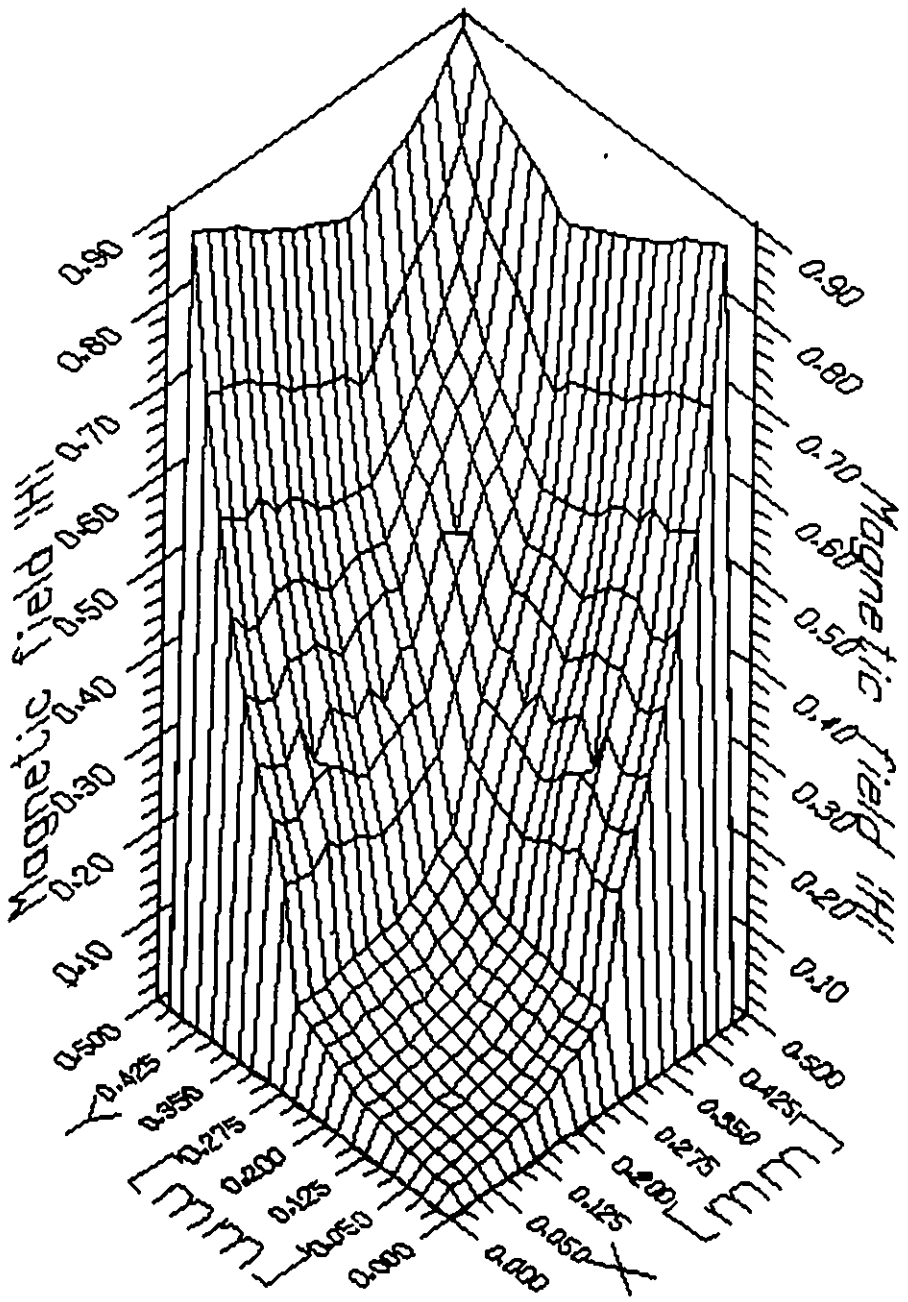


Figure 3.5: 3–D graph of the magnetic field diffusion in a 1.0 mm square rod solved by DORS: $H_i = x1.0$ A/m, frequency= 1.0 MHz, conductivity=10^6 S/m. (due to symmetry, only one quarter of the solution is shown)

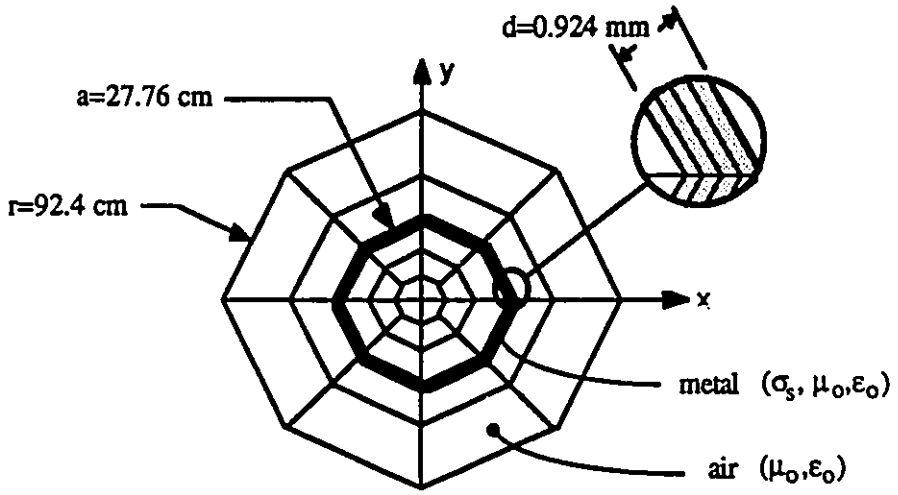


Figure 3.6: Cross-sectional view of the finite element mesh for the pipe geometry used to study penetration into a hollow cylinder with DORS

the pipe axis. In Fig. 3.7, we see the magnetic field intensity as a function of the observation radius. The pipe wall begins at $r = 27.67$ cm and ends at $r = 27.76$ cm. On either side of the wall for the low frequencies solved, the magnetic field is constant. For a frequency of 1.0 MHz, we observe that the field has dropped from a constant 1.0 A/m outside, to 0.001 A/m inside. The field is attenuated by 60 dB as it penetrates the wall of the circular enclosure. Figure 3.8 is an expanded view of the field inside the wall.

Figure 3.9 shows the level of the diffusion field inside a metal pipe (it is constant everywhere) for different frequencies. The metal pipe had a radius of 27.76 cm, a wall thickness of $d = 0.924$ mm, and a conductivity 10^6 S/m. The results plotted were computed by the FEM (solved with DORS using the geometry shown in Fig. 3.6), and the following approximation formula found in [15]:

$$H_{int}(\omega) = \frac{H_{ext}}{\cosh(\sqrt{j\omega\tau_d}) + \frac{a}{2d\mu_r}\sqrt{j\omega\tau_d}\sinh(\sqrt{j\omega\tau_d})},$$

The above formula describes the level of the magnetic inside a circular cylinder of radius a, and wall thickness d, with $\tau_d = \mu\sigma d^2$.

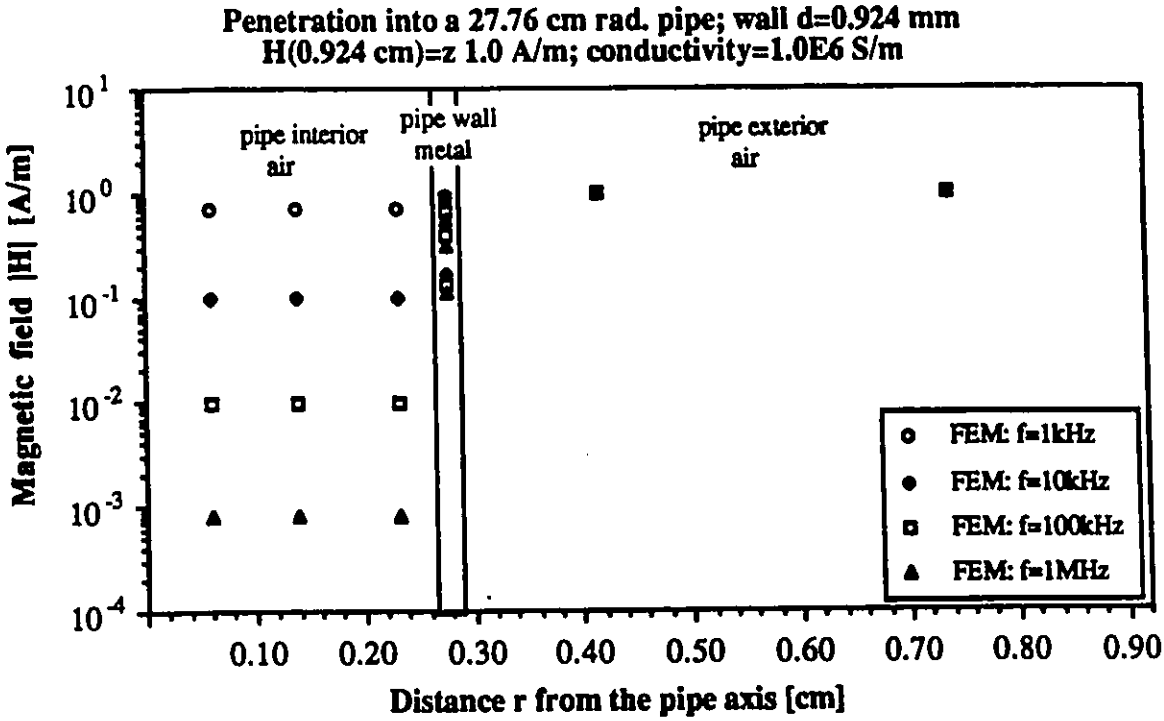


Figure 3.7: Magnetic field diffusion into a pipe for different frequencies (solved by DORS)

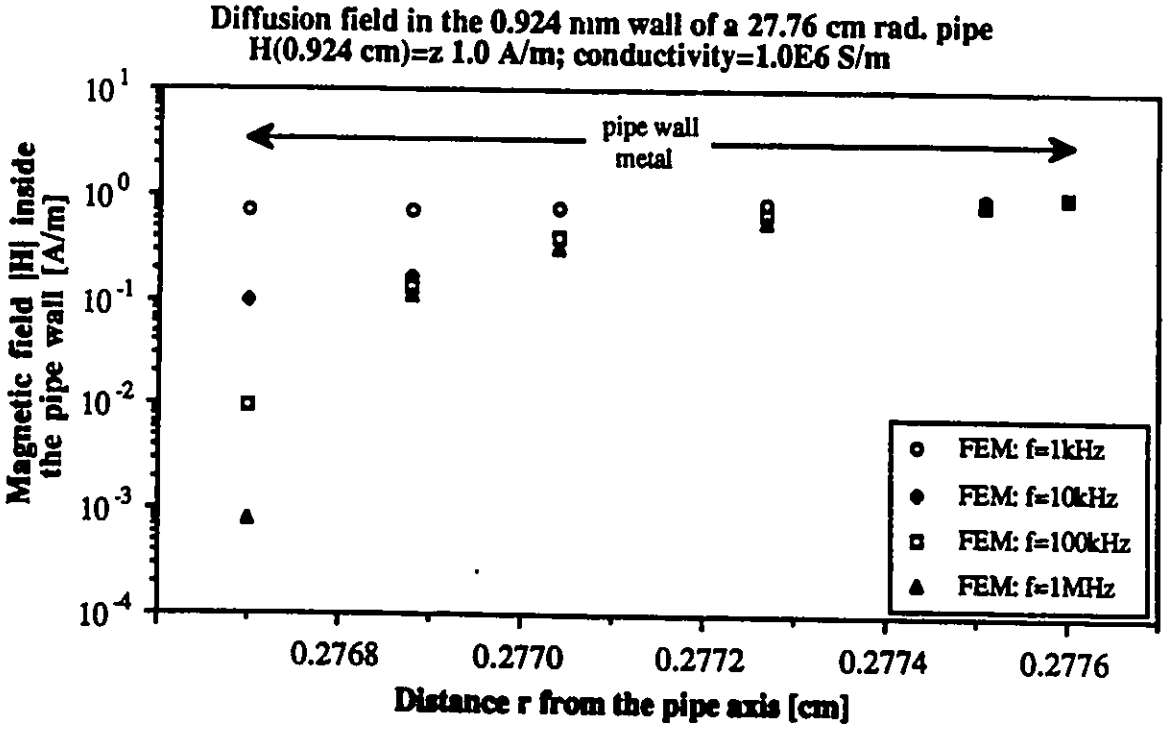


Figure 3.8: Magnetic field inside the pipe wall for different frequencies. Expanded view of Fig. 3.7. (solution by DORS)

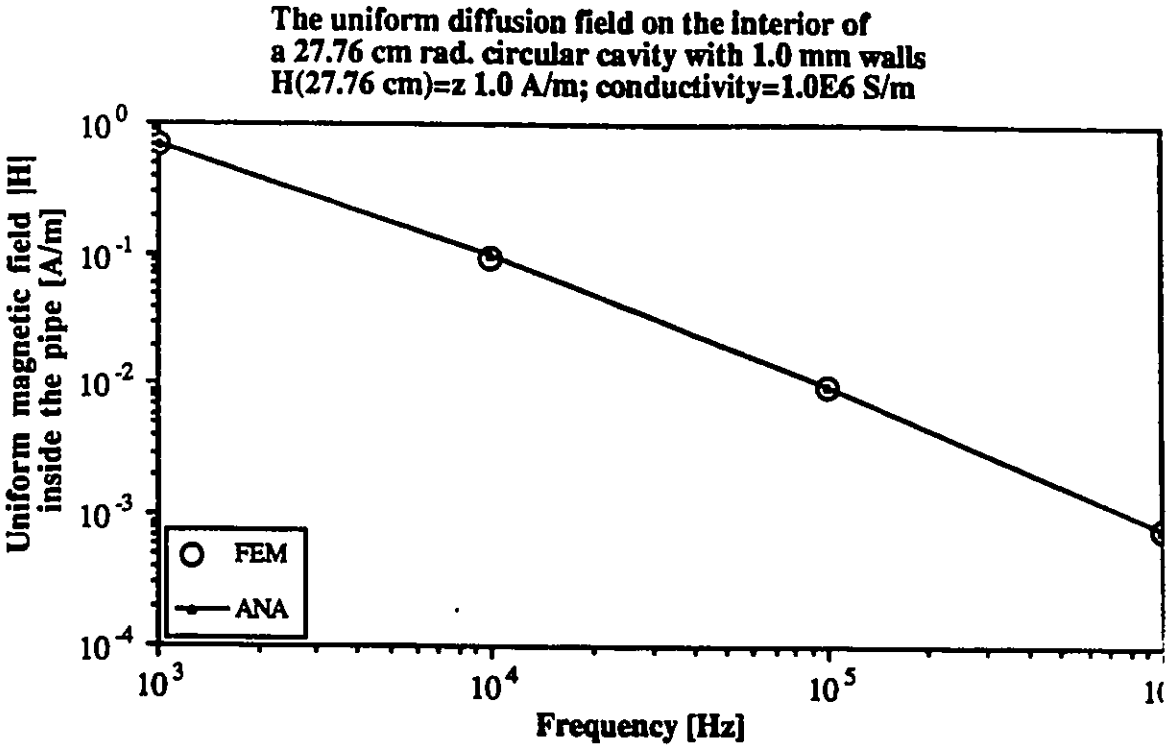


Figure 3.9: Diffusion field inside an infinitely long circular pipe for different frequencies

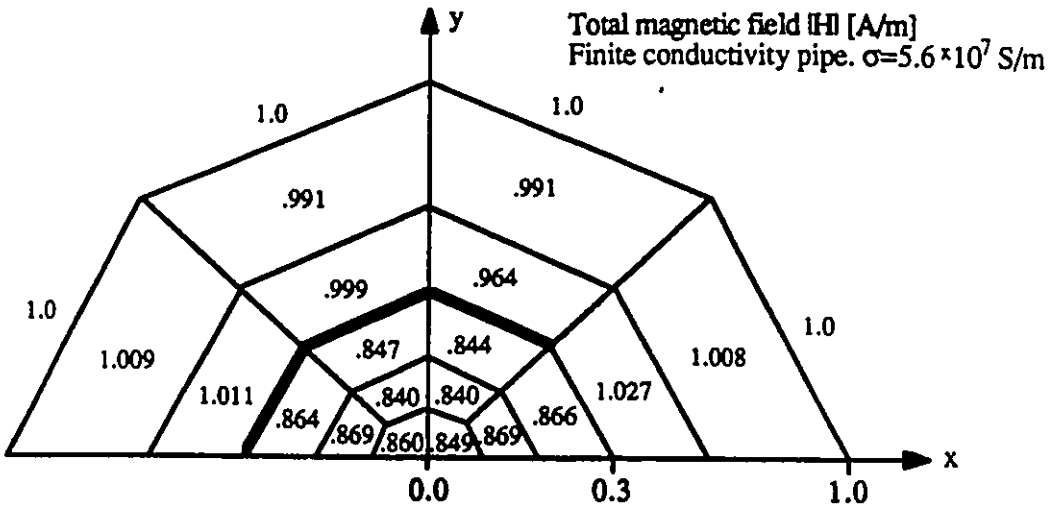


Figure 3.10: Magnetic field distribution on the cross-section of a long pipe with a 90° aperture: $\boldsymbol{H}_t = \hat{\boldsymbol{x}}1.0$ A/m, frequency= 1.0 MHz, conductivity=5.6×10^7 S/m. (due to symmetry, only one half of the solution is shown)

3.2 Two Dimensional Aperture Problems

The first effort at an aperture problem in an enclosure was to introduce a slot along the axis of the long cylinder. The slot was created by making the conductivity zero in a portion of the wall. The problem geometry is shown in Fig. 3.10. It is the same as in Fig. 3.6, but only half the problem is shown because of its symmetry. The bold line denotes the pipe wall which has a conductivity of 5.6×10^7 S/m. The pipe was irradiated with a 1.0 A/m magnetic field with a component along the pipe axis. The magnetic field intensity is shown for the center of each element. The field on the interior of the enclosure is no longer constant and is significantly higher than when there was no aperture. The shielding effectiveness of the enclosure at 1.0 MHz has been reduced from 60 dB to only 1.5 dB.

It is obvious that the dominant part of the penetrating field is entering through the aperture, and therefore that the contribution diffusing through the pipe wall has become negligible. With this consideration, we can use the Dirichlet boundary condition discussed in section 2.3 to create walls having no thickness, but with infinite conductivity. The same geometry is used (the enclosure being defined by $\boldsymbol{A} = 0, \phi =$

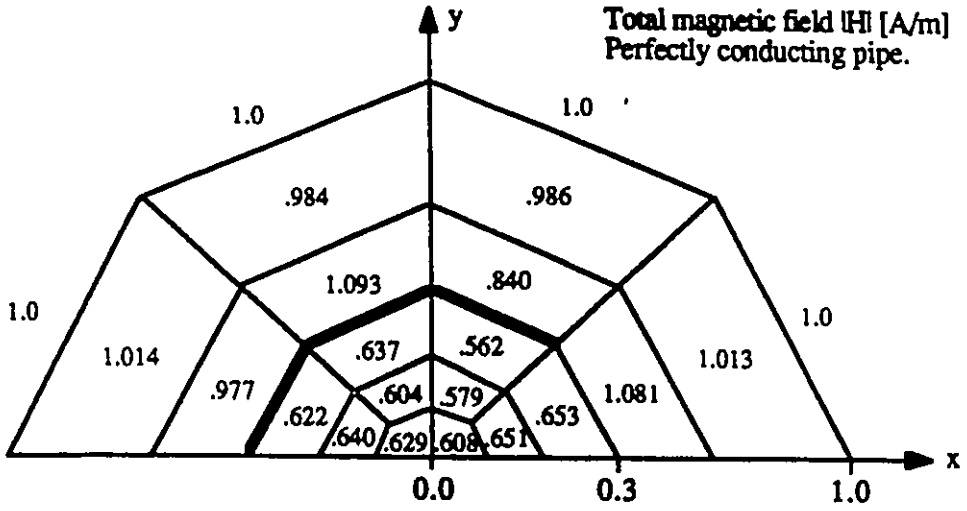


Figure 3.11: Magnetic field distribution on the cross–section of a long pipe $\sigma_{wall} = \infty$ with a 90° aperture: $H_t = z1.0$ A/m, frequency= 1.0 MHz. (due to symmetry, only one half of the solution is shown)

0)[1], and the magnetic field distribution is shown in Fig. 3.11. Note that the estimate of the condition number at 1.0 MHz of the system of equations for the problem having conductivity of 5.6×10^7S/m was 10^{23}, while it was only 10^{12} for the system of equations developed for the geometry with perfectly conducting walls.

Figure 3.12 is a third case of a slot aperture in a pipe, created using the Dirichlet boundary. The slot is smaller, and hence less energy is coupled into the cavity (attenuation of 25 dB).

3.3 Three Dimensional Aperture Problems

In the previous section, the three dimensional finite element algorithm (DORS) was used to find the approximate solutions to some simple one and two dimensional diffusion and penetration problems.

In this section, a three dimensional problem is discussed. In three dimensions, the simplest problem for which an analytical solution can be derived, is the diffusion of a magnetic field into a sphere. However, because of the limited number of nodes avail-

[1]Only 123 unknowns were required as opposed to 219 for the geometry where the wall thickness is modelled explicitly.

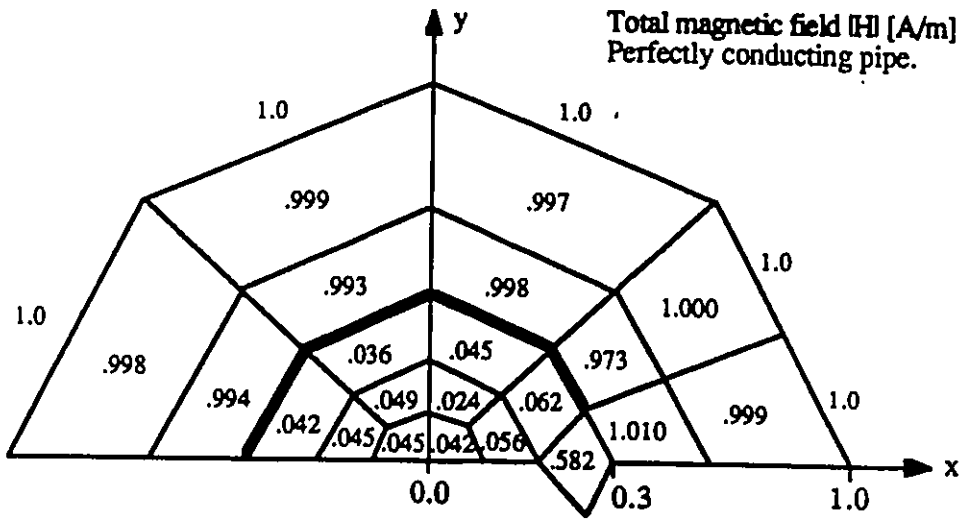


Figure 3.12: Magnetic field distribution on the cross-section of a long pipe $\sigma_{wall} = \infty$ with a 45° aperture: $H_t = z1.0$ A/m, frequency= 1.0 MHz. (due to symmetry, only one half of the solution is shown)

able with which to discretize the geometry, a sphere mesh would not have adequate resolution for comparison with the exact solution. Therefore, based on the fact that the program produced excellent results in one and two dimensions, the program is assumed to work for three dimensional problems.

A simple three dimensional enclosure with an aperture is a rectangular box with a rectangular hole in one side. Figure 3.13 shows such an enclosure.

If the walls of the box were to be modelled using elements made of a finite conducting material, then there must be many thin layers so that the diffusion field is properly interpolated. The mesh for such a problem would require at least 2744 bricks with 3375 nodes (4 elements to describe the wall, and 6 element resolution inside the box interior). The number of unknowns for this problem would be 13 500 which far exceeds the capacity of the program DORS.

By making several simplifying assumptions about the geometry of the enclosure, the number of unknowns can be reduced to a workable level.

The *first and most fundamental assumption* was that the walls of the box were made of a perfectly conducting material. The assumption can be validated by the fact that if the enclosure were made with sheet copper ($\sigma = 5.6 \times 10^7$ S/m) and had wall thickness of 1.0 mm (much smaller than the box dimensions), then an electromagnetic

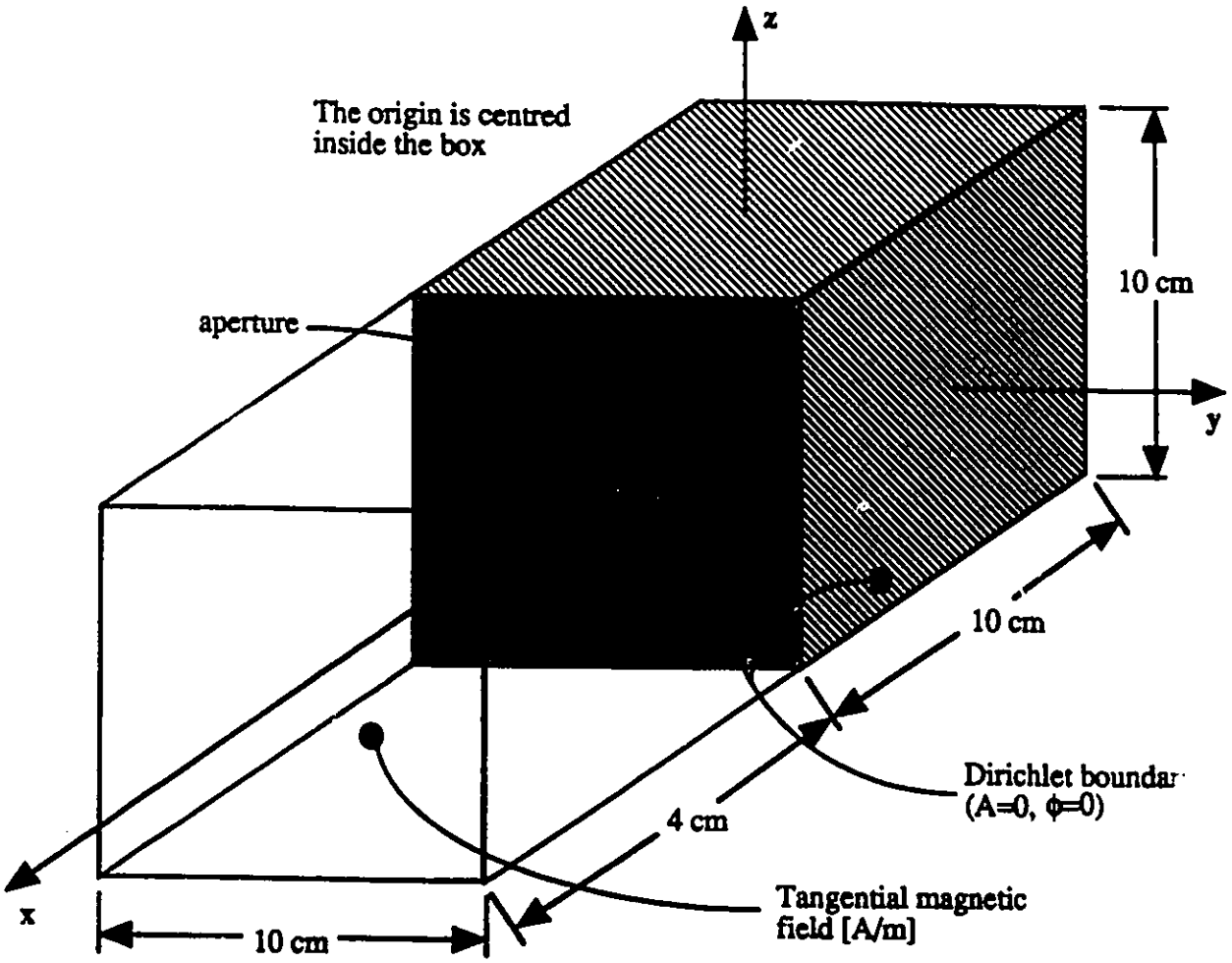


Figure 3.13: A metal box with a hole

field would be attenuated by about 129 dB at 1.0 MHz. This diffusion field on the cavity interior would be negligible when compared with the field which would penetrate through an aperture.

This assumption was necessary because it meant that the Dirichlet boundary condition ($\boldsymbol{A}$=0) discussed in section 2.3, could be used to define the conductor surface, eliminating the elements and nodes defining the walls, and significantly reducing the number of unknown potentials. It also eliminated the associated frequency limitation imposed by the finite conductivity and thickness of the wall elements.

The *second assumption* was that the enclosure was located in a uniform transverse electromagnetic field and that there are no obstacles near the box to perturb the field. The tangential magnetic field component was applied on a boundary far away, such that any scattered fields from the box became negligible at the boundary.

Since the field could not diffuse or penetrate through the shield wall, except via an aperture, it was only necessary to impose the tangential magnetic field on the surface of the region directly in front of the aperture. Figure 3.13 indicates where the Dirichlet potentials and the tangential magnetic fields were applied.

Several cases of the electromagnetic distribution inside a 3–D enclosure were examined.

Case I

In the first case, the enclosure was a cube of dimension 10.0 cm that had a 2.5 cm by 3.33 cm rectangular aperture centred on the front face. The box was discretized into 240 elements [5(in x) by 8(in y) by 6(in z)]. The origin of the Cartesian coordinate system is centred in the box.

The irradiating magnetic field had only a component parallel to the z axis, an intensity of 1.0 A/m, and a frequency of 1.0 MHz. The field was applied 4.0 cm in front of the enclosure.

The vector magnetic potential ($\boldsymbol{A}$) and the scalar electric potential (ϕ) were fixed to zero at the nodes on the surface of the enclosure walls. The aperture was created by leaving the potentials of one node in the centre of the front face floating (free to assume any non-zero value). The number of unknowns to be solved was 996 potentials.

The electromagnetic field distribution of vector fields in three dimensions is not easily illustrated in two dimensions. One effective way is to look at a cross–section of the region of interest, and to plot the magnitude of the total electric and magnetic

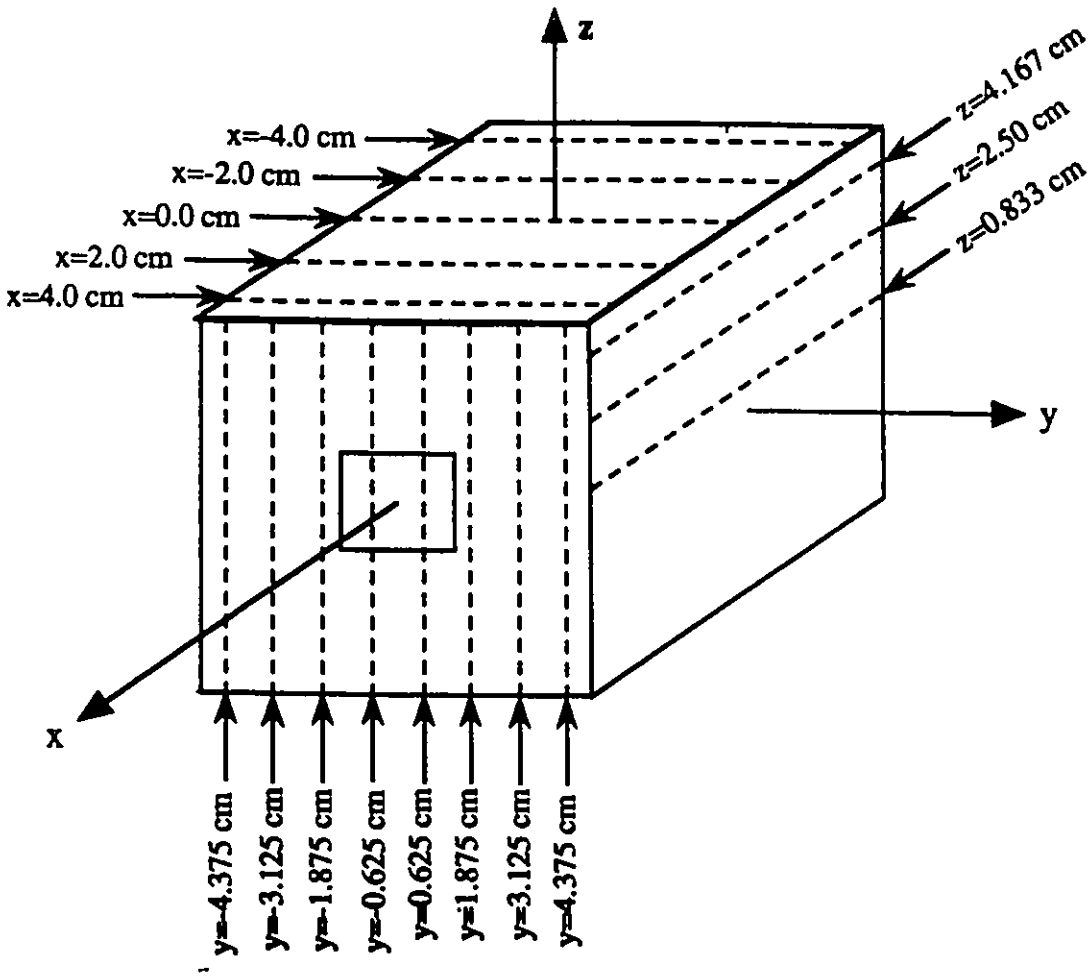


Figure 3.14: The 10 cm cube illustrating the various plane cuts used to depict the electromagnetic field distribution

fields separately. Figure 3.14 presents the different plane cuts used to depict the 3-D electromagnetic field distribution. By plotting only the magnitude, there is no information about the direction of the field, but variations in the field can be seen, and the information is useful for solving worst case approximations for a given EMI situation.

Shown in Fig. 3.15 (a)–(c) is the magnitude of the total magnetic field for the problem of case I as a function of y for different x on three different plane surfaces of constant z. The field for each x is maximum at $y = 0$, which demonstrates that the field has penetrated through the aperture. The intensity of the field decreases as we move away from the aperture (x getting smaller).

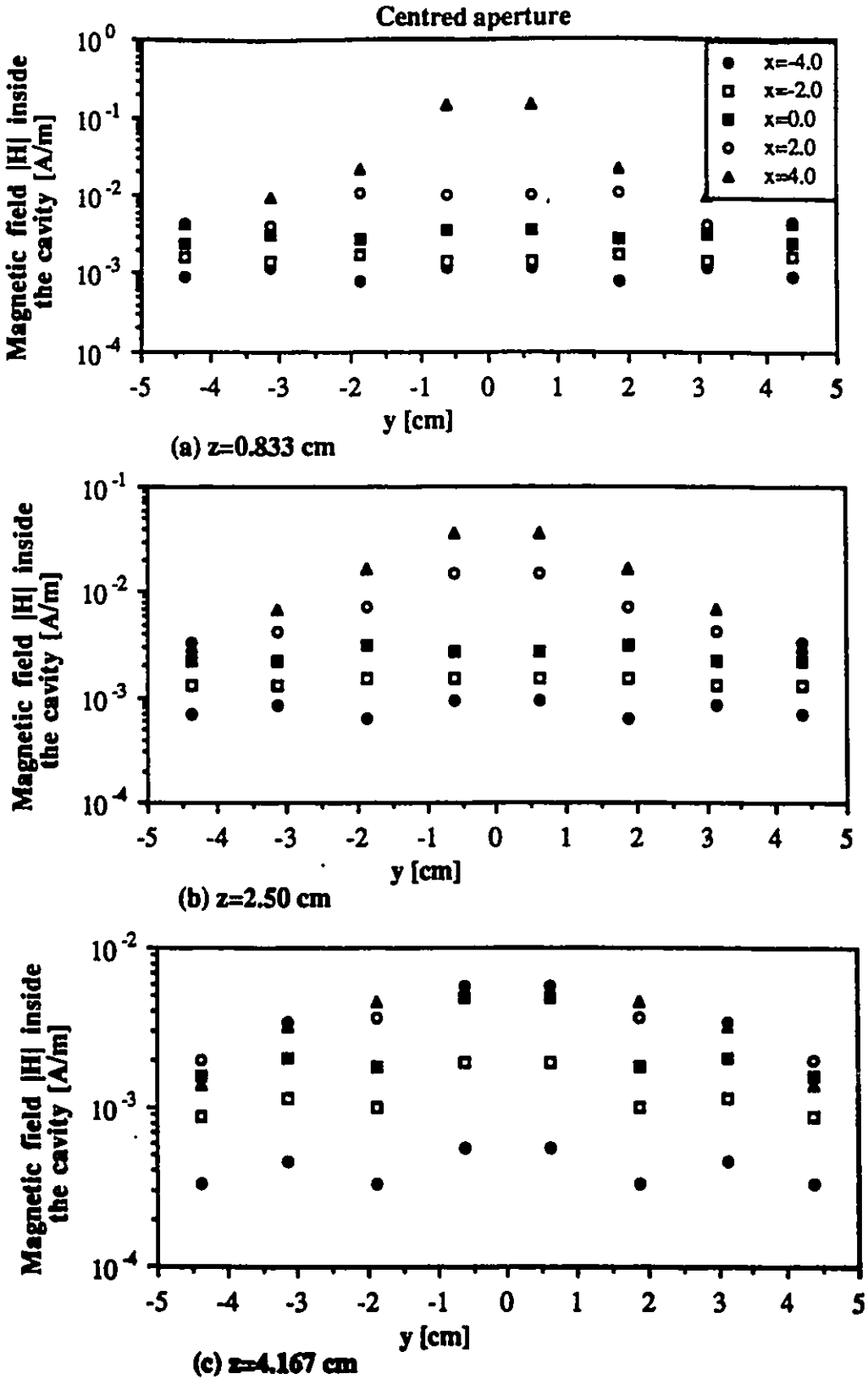


Figure 3.15: Magnetic field distribution for case I: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,0,0), $H_t = z1.0$ A/m, frequency=1.0 MHz

In Fig. 3.16 (a)-(c), the magnitude of the total electric field is shown for case I.

Case II

In this case, the only difference as compared with the first case was that the problem was solved at a frequency of 1.0 GHz. Figures 3.17 and 3.18 present the distribution of the total magnetic and electric fields, respectively, for a plane cut $z = 0.833$cm (Note: the plane cut is made at 0.833cm instead of 0.0 because the z dimension of the cube was broken into 6 elements and 0.833 cm is the closest element centre where the potential is known).

The FEM solution of the problem described in case I for a range of frequencies: 1.0 kHz to 10.0 GHz, revealed that the results do not change significantly for frequencies below 100.0 MHz. As the frequency increases above 100.0 MHz, however, the wavelength becomes comparable to the dimensions of the enclosure, and the distribution can no longer be considered quasi-static. Above 1.0 GHz, the solution breaks down because the element size becomes comparable to the wavelength, and the field cannot be accurately interpolated.

Case III

The case III problem was identical to case I, except that the aperture on the front face of the enclosure was offset to the right of the centre by 1.25 cm [moved to (5,1.25,0)]. In Figs. 3.19 and 3.20, the magnitude of the magnetic and electric field distributions for case III are plotted for $z = 0.833$cm at $x = 4.0$cm. It can be seen that there is no longer symmetry about the $y = 0$.

Case IV

To demonstrate the solution of an inhomogeneous problem with the FEM algorithm, the bottom half of the box ($z < 0$) described in case I was filled with water ($\varepsilon_r = 80, \sigma = 0$). This was easily implemented by making the relative permittivity of the elements in the lower half of the mesh equal to 80. The magnetic field applied on the boundary was in the y-direction and had a frequency of 1.0 MHz. Figure 3.21 contains plots of the magnitude of the magnetic and electric fields for case IV at $y = 0.625$ cm, $x = 4.0$ cm. We see that the fields are perturbed by the presence of the water.

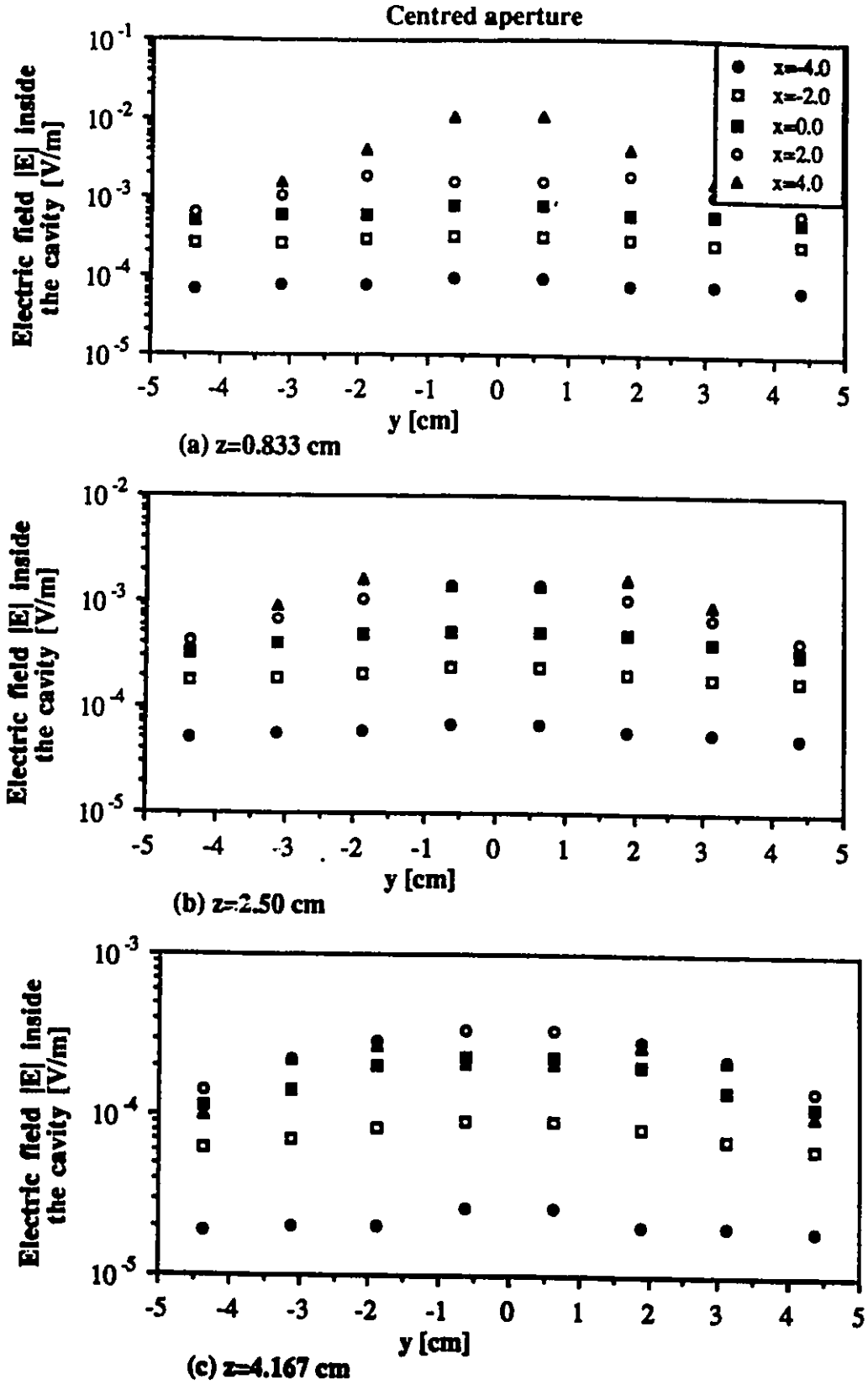


Figure 3.16: Electric field distribution for case I: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,0,0), $H_t = z1.0$ A/m, frequency=1.0 MHz

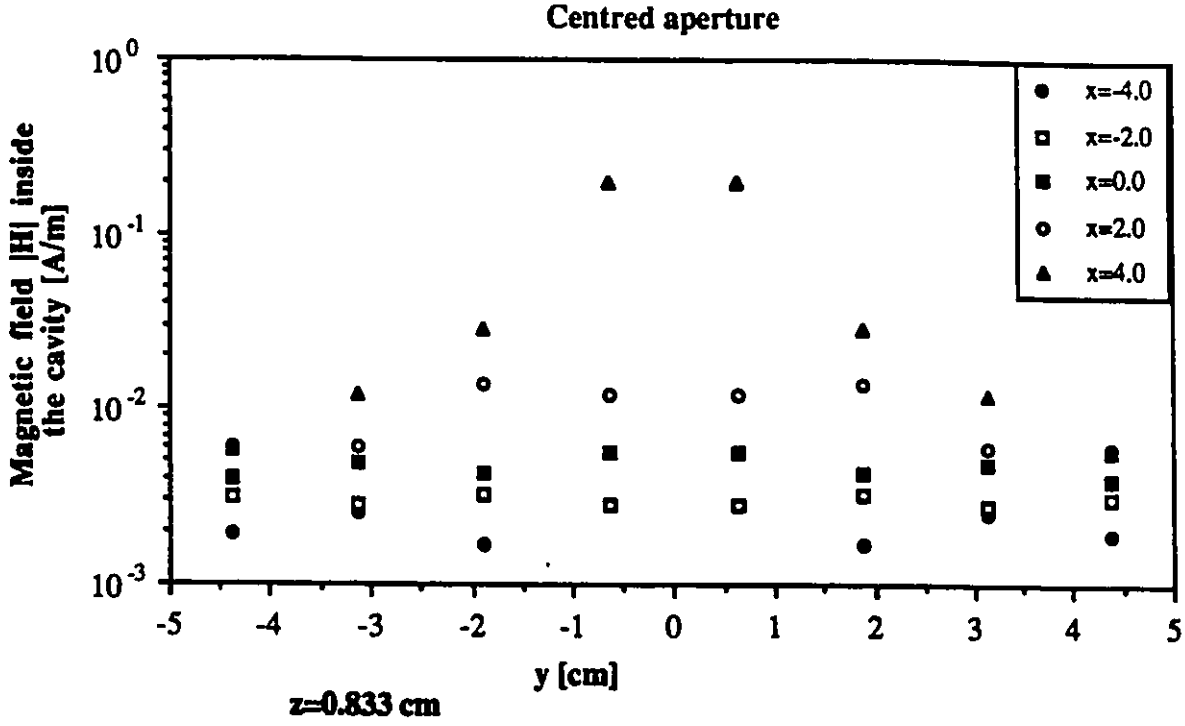


Figure 3.17: Magnetic field distribution for case II: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,0,0), $H_t = z1.0$ A/m, frequency=1.0 GHz

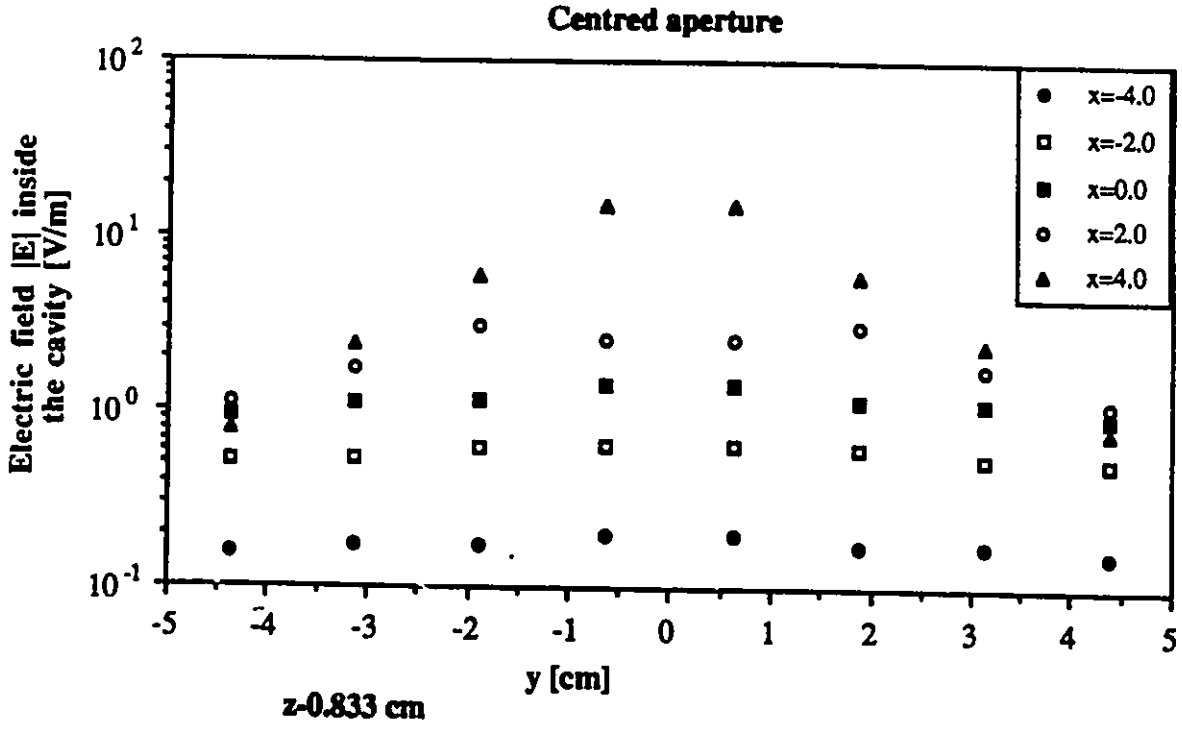


Figure 3.18: Electric field distribution for case II: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,0,0), $H_t = z1.0$ A/m, frequency=1.0 GHz

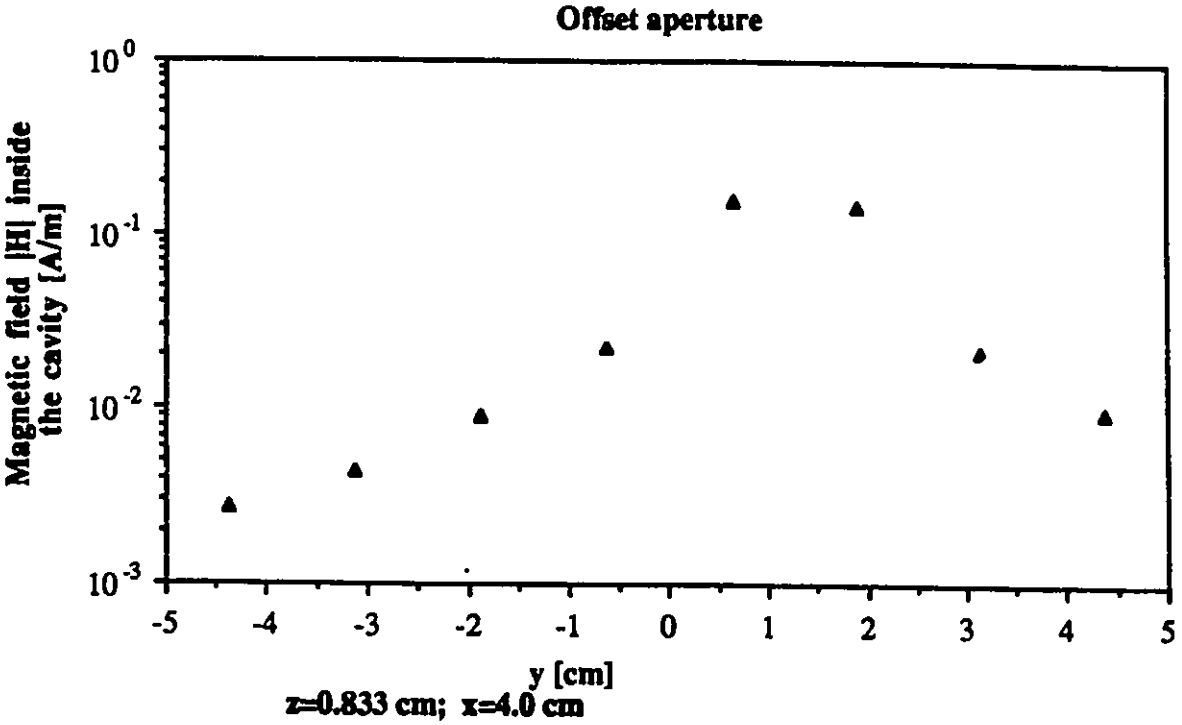


Figure 3.19: Magnetic field distribution for case III: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,1.25,0), $H_t = z1.0$ A/m, frequency=1.0 MHz

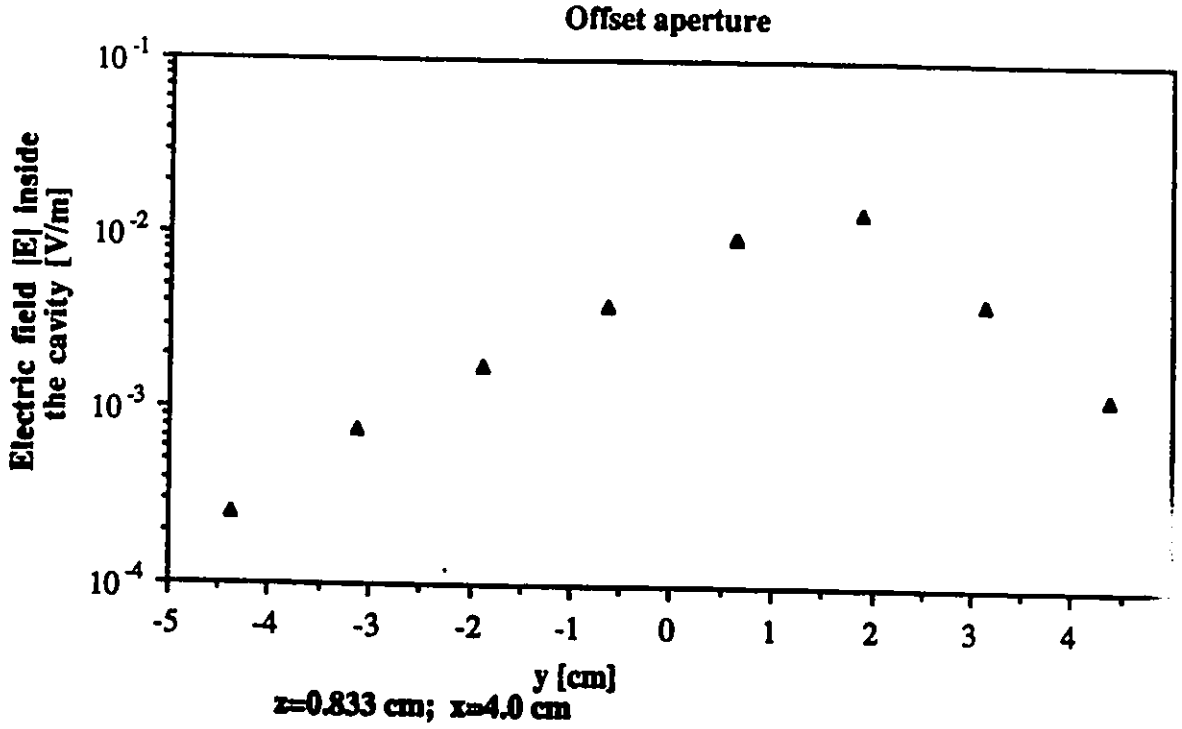


Figure 3.20: Electric field distribution for case III: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,1.25,0), $H_t = z1.0$ A/m, frequency=1.0 MHz

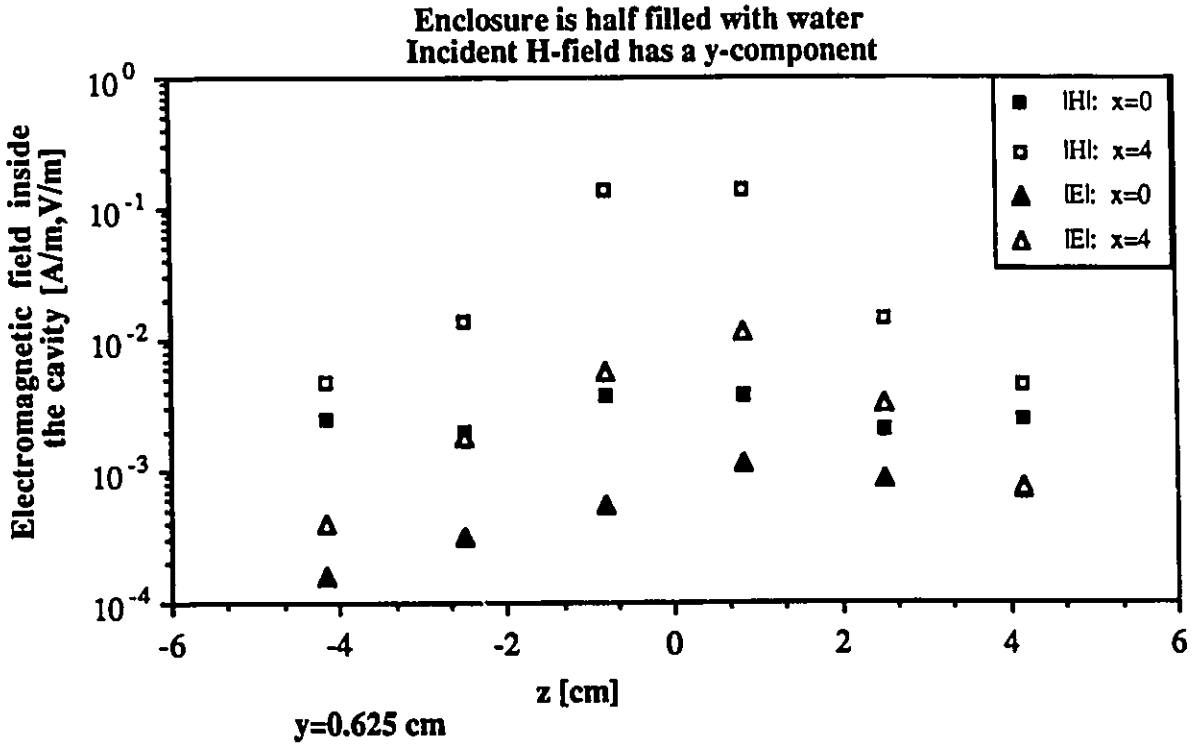


Figure 3.21: Electromagnetic field distribution for case IV: a 10 cm cubic cavity, with a 2.50 cm by 3.33 cm aperture at (5,0,0), $-5 \leq z \leq 0; \varepsilon_r = 80$ water , $0 \leq z \leq 5; \varepsilon_r = 0$ air , $\boldsymbol{H}_i = \boldsymbol{y}1.0$ A/m, frequency=1.0 MHz

3.4 Time domain Diffusion Problem

In this section, results are presented for a diffusion problem solved in the time domain with the three dimensional time domain finite element program (APROXHP).

The test geometry is a conducting slab (see Fig. 3.1) with thickness $d = 10$ cm, and conductivity of 5.6×10^7S/m. The slab was divided up into 8 elements across.

The input signal for the test sample was a unit step function[2].

$$1(t) = \begin{cases} 0, & t \leq 0; \\ 1, & t > 0. \end{cases}$$

The results shown in Fig. 3.22 are the total magnetic field as a function of time at different points inside the slab for both the analytical and the finite element solutions. The derivation of the analytical solution is shown in Appendix B.

[2]The derivation of the analytical solution is shown in Appendix B

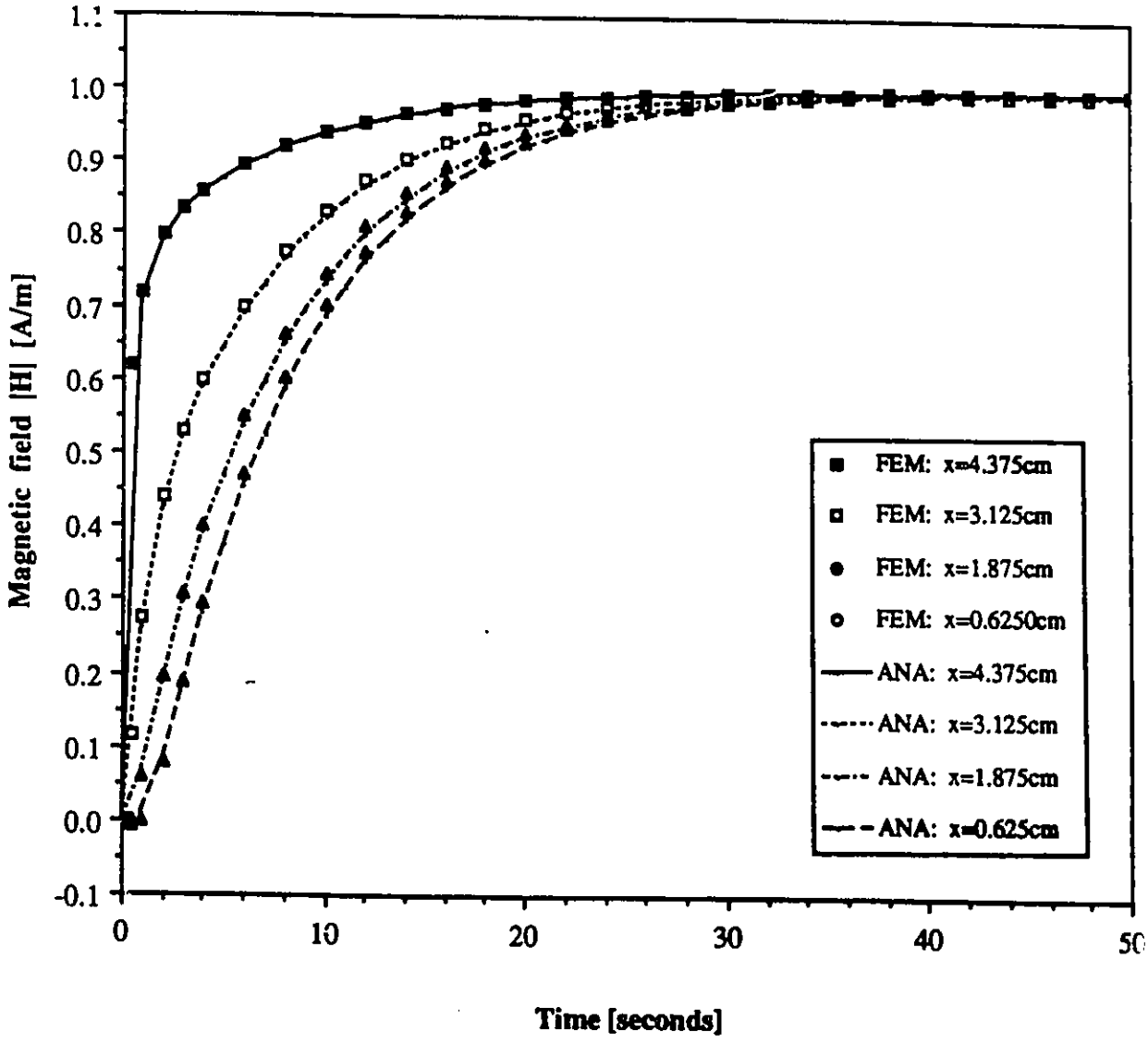


Figure 3.22: Time domain solution of the diffusion of a step–input magnetic field into a 10.0 cm slab, conductivity=5.6×10^7 S/m (solved by APROXHP for different values of x)

Chapter 4

Discussion

4.1 Frequency Domain Analysis

In chapter 3, the results of various electromagnetic problems solved via a three dimensional finite element program were presented. The program used was the one based on the formulation developed in chapter 2 and for which the code (DORS) is included in Appendix D.

The one dimensional problem of a magnetic field diffusing into a conducting slab is a simple electromagnetic problem. Because of its simplicity, it proved to be an ideal test problem for the 3-D FEM program. The geometry is described with identically sized rectangular brick elements which use a minimal number of nodes and it is very easy to increase the number of elements to study the convergence of the solution. The finite element geometry approximates the one dimensional problem by making two dimensions very large.

With 17 elements across a 1.0 mm thick, 5.6×10^7 S/m conducting slab, the element size is 87% of the skin-depth at 1.0 MHz. From Fig. 3.2, the error at 0.44 skin-depths from the surface (x =0.4706 mm) is 1.2%, and at 4.4 skin-depths from the surface (x =0.2941 mm) the error is 21%. The magnetic field is computed from the curl of the vector magnetic potential. Since the basis functions are only first order, the potential function is piece-wise continuous across inter-element boundaries and the derivative is discontinuous. Because the derivative and therefore the magnetic field is discontinuous across the boundaries, as we move away from the edge of the slab where the tangential magnetic field is defined, the error in the solution is cumulative. The use of second order basis functions would allow piece-wise continuity of the magnetic field and therefore would reduce the error in the middle region.

The tangential magnetic field must be defined explicitly everywhere on the outer boundary of the region of interest. If the magnetic field is left undefined on a surface, then a value of zero is implicitly imposed. The 3-D approximation of a one dimensional slab required uniform irradiation with a magnetic field all around the structure (even on the cross-section of the slab ends where we seek the distribution). However, since the brick elements were large (at least ten times the slab thickness and much larger than the skin-depth) in the directions which the slab was infinitely extended, the magnetic field applied at the ends decayed to zero in the middle region. Therefore, at the point where the solution was desired, the contributions from the end field were negligible.

As an electromagnetic field diffuses into a conductor, its intensity decays exponentially. The rate of decay is dependent upon the conductivity and permeabilty of the material, and the frequency of the propagating field. The skin-depth (δ) is the distance the incident wave must propagate into a material for the incident field to decay by a factor of $e \simeq 2.718$.

Figure 4.1 presents the error in the FEM solution for the magnetic field computed at the centre of the element on the slab boundary for different values of the ratio h/δ, where δ is the penetration depth, and h is the thickness of the finite element. The skin-depth was varied by solving for various frequencies. We see that the error can be significant ($> 1\%$) if the element size is too large, but that good accuracy can be obtained (error$\approx 0.01\%$) if the ratio h/δ is at most 0.1 .

The finite element method relies on polynomial interpolatory functions to model the field variation inside each element. As the skin-depth decreases, the field varies too quickly with position for the interpolatory functions to follow it accurately and thus the error increases. To increase the accuracy, it is necessary to decrease the size of the elements further, or to increase the order of the basis function. Both these options require an increase of the number of nodes to discretize the same region. There is an upper frequency limitation to the FEM solution of a particular problem, which is dependent on the size of the elements and the skin-depth and it is determined by the memory capacity of the computer.

From Fig. 4.1, a rule of thumb for determining the size of the finite elements can be developed. If we require that the error of the magnitude of the magnetic field be less than 1%, then the element size should be smaller than the skin-depth.

$$h < \delta\Big|_{error<1\%}$$

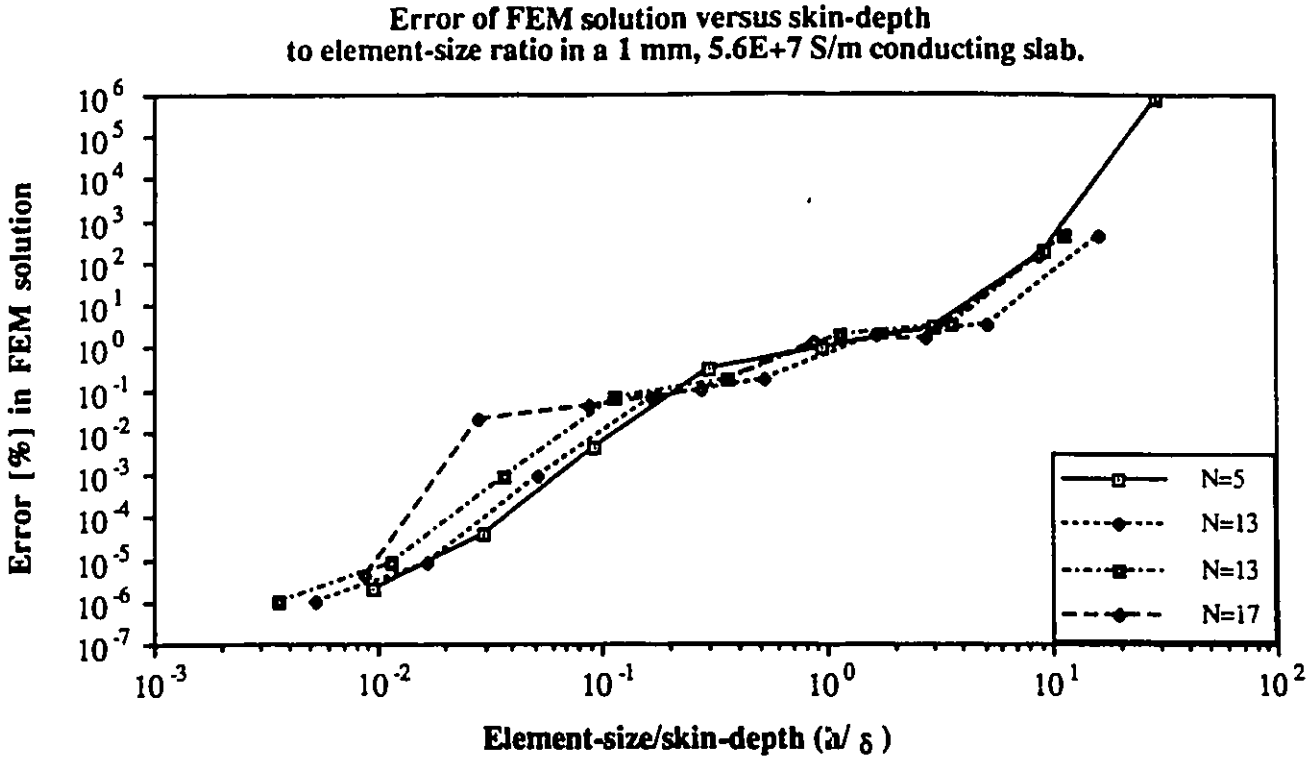


Figure 4.1: Error analysis of FEM for various h/δ

For high frequencies and conductivities, the skin-depth is very small, and thus the number of nodes required to achieve acceptable convergence will exceed the limit determined by the available memory of the computer.

In Fig. 3.6 the geometry of a circular conducting rod was approximated by an octogonal structure. Using this mesh, there was error in the FEM solution because the element discretization only minimized the error of the magnetic field propagating in the thin dimension of the elements. The field diffusing radially from angles other than the ones perpendicular to the element boundary should decay quicker, but the field was averaged over the large elements. The circular cross-section could be resolved better and the errors reduced by using a discretization mesh with more segments on the circumference and in the radial direction.

It would not be possible to solve a one dimensional enclosure problem (two infinite slabs separated by a non-conducting region) with the 3–D FEM program, since the magnetic field in the region between the walls is constant and it would simply assume the value of the tangential field applied on the slab ends.

The magnitude, phase, and direction of the applied magnetic field are constant

over each sub-surface, but they can be varied from surface to surface to allow different types of propagating wave fronts to be approximated. Cylindrical wavefronts were used for the diffusion and penetration problems with circular symmetry.

In one and two dimensional analysis, a current can be regarded as having no variation along the infinite dimensions, but although with a three dimensional analysis, the problem can be made large in one or two dimensions to approximate infinity, the discontinuity at the problem boundary will cause the uniform flow of the current will be disrupted. To ensure zero divergence of the total current, surface charges will accumulate around the discontinuity and will result in a re-distribution of the electric field. Because of this, one and two dimensional problems can only be modelled using the 3-D program if the magnetic field components are in the directions of infinite extent.

A slot or aperture in a conducting wall is a discontinuity where surface charges will accumulate and where eddy currents are induced. At these discontinuities, the direction of the electromagnetic field will change rapidly as a function of the position. Unless the finite element mesh has sufficient resolution, the error in the electromagnetic field values around the discontinuities will be very large.

We cannot examine the errors introduced by inadequate resolution in the mesh around discontinuities because there are no three dimensional problems involving discontinuities for which an exact solution is known and which can be adequately modelled by DORS. Nor can we check for the convergence of a solution by adding more nodes around the discontinuities because of the memory restriction. At the present stage, we can only examine the results intuitively and determine whether or not the electromagnetic field is behaving as expected.

In the regions near the walls and the aperture (the local discontinuities), the electromagnetic field components change rapidly, and the poor resolution will produce large errors in these regions. Increasing the number of nodes in the structure would allow for the field to be better resolved, and for the convergence of the solution to be checked.

The results presented in Figs. 3.15 to 3.21, illustrate the electromagnetic field distribution inside a cubic enclosure. Even though the problem geometry is very simple, it is nevertheless a practical problem, since the cabinets for most electronic devices are rectangular boxes.

At 1.0 MHz, the resolution was 15 000 elements per wavelength, but the cavity size

was only 1/3000 wavelengths (the cavity was a 10 cm cube and had $5 \times 8 \times 6$ element resolution), though with this resolution, the results appear reasonable. Energy is definitely penetrating through the aperture (when the aperture is moved, the peak field also moves) and it is diminishing with the distance. At 1.0 GHz, the resolution was 1.5 elements per wavelength, but the cavity size was only 1/3 wavelengths. The field pattern is similar to the results for 1.0 MHz, but the intensity levels are higher. When the bottom of the box is filled with water (see Fig. 3.21), the magnetic field is not perturbed noticeably, yet the magnitude of the electric field drops suddenly across the inteface. This behavior is in accordance with how an electromagnetic field should behave at such an interface. At the air–water boundary, the discontinuity in the tangential magnetic field is equal to the surface current which is zero because the conductivity is zero:

$$\boldsymbol{n} \times (\boldsymbol{H}_{water} - \boldsymbol{H}_{air}) = 0,$$

and the discontinuity in the normal component of the electric flux density is equal to the surface charge density:

$$\boldsymbol{n} \cdot (80\boldsymbol{E}_{water} - \boldsymbol{E}_{air}) = \frac{\rho_s}{\varepsilon_0}.$$

In regions where the conductivity is is very low or zero, the skin-depth is not meaningful, and cannot be used to determine the size of the finite elements. The variation of the field will not be linear, but if it is assumed to be smooth over one wavelength, then to approximate it, the largest dimension of finite element must be several times smaller than the wavelength of the propagating wave. We can develop a general rule of thumb as:

$$Qh \leq \lambda,$$

or

$$h \leq \frac{1}{Qf\sqrt{\mu\varepsilon}}, \tag{4.1}$$

where h is the thickness of the element and Q is a number greater than 1. The number Q has to be determined through analysis of results and intuition.

The system of equations used in the algorithm to describe the electromagnetic potential distributions ($\boldsymbol{A}$, ϕ) is ill-conditioned. If we look at Eqs. 2.12 and 2.13, we see that the terms with ϕ are scaled by "$\sigma + j\omega\varepsilon$", and the terms with $\boldsymbol{A}$ are scaled by "$\omega^2\varepsilon - j\omega\sigma$". Examining the ratio between the two scaling factors, we find it is the angular frequency. Hence, the terms in the matrix due to ϕ are several orders of

magnitude smaller than those of A. As well the real and imaginary components of the scaling factors differ by a factor of $\sigma/\omega\varepsilon$. This scaling produces columns which are nearly singular. If the displacement current term is ignored ($\varepsilon = 0$), or the conduction current term is ignored ($\sigma = 0$), then the matrix coefficients are either purely real or purely imaginary, and the matrix is better conditioned.

The matrix reduction routine ZGECO, computed a worst case approximation for the condition of the matrix. My experiences showed that if the problem region contained elements with a finite conductivity, then the condition number was typically of the order of magnitude 10^{18}, but would increase proportionally to changes in the frequency and conductivity. However, if an infinitely conducting surface described with a Dirichlet boundary were used instead, then the condition number was typically 10^{12}, which is a significant improvement over the previous case, but still indicates a poorly conditioned system. The ill-conditioning of the system indicates this algorithm is best restricted to solving problems with low conductivity and frequency.

A very large condition number would seem to indicate that the results were not meaningful, but as in the case with the infinite slab where the condition number was 10^{24}, it works out that the results made sense physically.

4.2 Time Domain Analysis

Only the results for one problem solved with the time domain finite element program were shown in chapter 3. It was a simple one dimensional slab (modelled in 3-D) irradiated with a step–impulse magnetic field. Only eight elements were used in the cross–section, however because a unit step function contains most of its energy in the low frequency band ($f \ll 1$ MHz), the magnetic field will not decay quickly so that large elements are sufficient for good accuracy. It is observed that the farther one is away from the outer surface, the slower the response of the rising magnetic field.

In the time domain, it is important to consider how far the wave front would travel in one time-step Δt. A wave propagating with velocity v_p will take h/v_p seconds to cross an element of size h. If $\Delta t < h/v_p$, then errors will occur because the field will not propagate properly. The appropriate rule of thumb for relating the element size and the time-step is

$$h = \frac{v_p \Delta t}{Q},$$

where Q is a number which is determined so as to yield good accuracy.

The time domain program shares the same memory limitation problem as the frequency domain counterpart, and as well it requires longer computation times. In attempts to maintain the desired error tolerances, the ordinary differential equation solver reduces the time-step until the error is acceptable, but sometimes the time-step becomes too small, the rule used to discretize the geometry is violated and the algorithm fails.

Chapter 5

Conclusions

In this thesis, a three dimensional formulation of the finite element method was developed to solve the equations describing the electromagnetic field distribution in an arbitrary region containing conducting and dielectric materials when the tangential magnetic field was known at the boundaries. The intent was to explore the feasibility of solving the 3–D problem of the penetration of electromagnetic fields through apertures and walls in shielding enclosures of electronic equipment. The limitation of computer resources (time and memory) restricted the available resolution of the mesh, thus the diffusion and the displacement problems had to be solved separately. A program that could solve the electromagnetic field distribution inside an enclosure would have application in EMI/EMC analysis and design.

The formulation was developed using a three component vector magnetic potential and a scalar electric potential. The displacement current as well as the conduction current was accounted for, thus not restricting the solution at high frequencies. The equations were developed using Galerkin's weighted residual method and solved by means of the finite element method. The bounded region of interest was discretized using eight node isoparametric hexahedrons and the potential functions were defined using linear first order basis functions. These elements minimize the discretization error when modelling complicated geometries. The volume and surface integrals were solved using Gauss quadrature numerical integration. Two programs DORS and APROXHP, respectively for frequency and time domain analysis, were written using the above algorithm to solve the electric and magnetic field distribution.

The frequency domain finite element method program (DORS) was validated by comparison with closed form solutions for simplified geometries. The algorithm proved to have a convergent solution when solving the diffusion of electromagnetic

fields into conducting hollow and solid structures without apertures. The upper frequency at which the algorithm could still work accurately was dependent upon the size of the computer on which the program ran. The maximum size of an element was determined by the frequency and the constitutive parameters of the problem. The size was restricted because the amount of available memory fixed the upper limit on the number of nodes for any problem.

The system of equations generated by the finite element method for the A–ϕ formulation is an ill-conditioned system. The condition number is a measure of the ill-condition or nearness to singularity of a matrix and it is proportional to the frequency and conductivity. Thus even with the availability of an infinitely large memory, at very high frequencies the results would not be reliable, however experience indicated that if sufficient resolution was available, the results made sense physically..

The penetration of a steady–state electromagnetic field through an aperture into a simple cavity was analyzed with the 3–D FEM program. The enclosure was created with infinitesimally thin, infinitely conducting walls using a Dirichlet boundary condition. This condition reduced the number of unknowns significantly and minimized the condition number of the matrix. The electromagnetic field values inside the cavity could neither be verified analytically, nor by checking the convergence, however an intuitive analysis indicated that the distribution behaved correctly.

The ability of the program to model complicated enclosure structures was illustrated by solving different cases: the aperture was moved, a dielectric region was introduced, a different frequency was solved, or even a different wave–type was imposed.

The 3D–FEM program produced results in the frequency domain which were suitable for prediction of field distributions. However, it was very important to pay proper attention to the discretization of the problem into finite elements. The element size should be at least one order of magnitude smaller than the skin–depth in conducting regions or the wavelength in non-conducting regions. As the frequency increases, then the number of unknowns must also increase.

The analysis of problems in the time domain was simplified by assuming that the displacement current was negligible, and that the gradient of the scalar electric potential was zero. These simplifications produced a well conditioned system of equations that had fewer unknowns. The restriction on the number of nodes placed a limit on the minimum size of the time step that was dependent upon the element size.

The validity of the time domain program was illustrated by solving the diffusion of a step-impulse magnetic field into a conducting slab.

The conclusions to be drawn from this thesis are that the A-ϕ formulation produces a poorly conditioned system of equations when solved by the finite element method, that the FEM can be developed to solve electromagnetic problems in both time and frequency domains, and that three dimensional meshes are difficult to construct without an automatic meshing routine and are seriously restricted in resolution by computer memory. No conclusions which would assist an EMC engineer in shielding design can be drawn yet because of the present short-comings of the programs.

The limitations due to computer memory restrictions, and the lengthy computation times involved in using the three dimensional finite element program indicate that until more powerful computers are made available, it is not yet feasible for use in an EMI/EMC CAD package.

Appendix A

A Theorem on the Uniqueness of Electromagnetic Field Problems

This appendix presents a theorem for the uniqueness of the solution of an electromagnetic field problem with respect to the boundary condtions and illustrates its proof[26].

Theorem of Uniqueness

The electric and magnetic fields, respectively $\boldsymbol{E}(\boldsymbol{r},t)$ and $\boldsymbol{H}(\boldsymbol{r},t)$, are uniquely determined in the region Ω by their initial conditions, and by the values of one of $\boldsymbol{E}_t(\boldsymbol{r},t)$ and $\boldsymbol{H}_t(\boldsymbol{r},t)$, at every point on a surface Γ at all times.

Proof of the Theorem of Uniqueness

We define a special surface Γ to enclose the region of interest which is depicted as Ω. The surface Γ is closed and stationary (Fig. A.1). The values of σ and ϵ are positive or zero everywhere.

The following relations must be met at every point on Γ at all times.

$$\boldsymbol{n}\cdot(\nabla\times\boldsymbol{E}(\boldsymbol{r},t)) = 0 \tag{A.1}$$

$$\boldsymbol{n}\cdot(\nabla\times\boldsymbol{H}(\boldsymbol{r},t)) = 0 \tag{A.2}$$

Equation (A.1) states that there are no normal components of the magnetic flux density on the surface Γ. It effectively means there is no inductive coupling between the region enclosed by Γ, and the region outside it.

Equation (A.2) states that there are no normal current components (either conduction of displacement) across the surface Γ. There is no capacitive coupling between

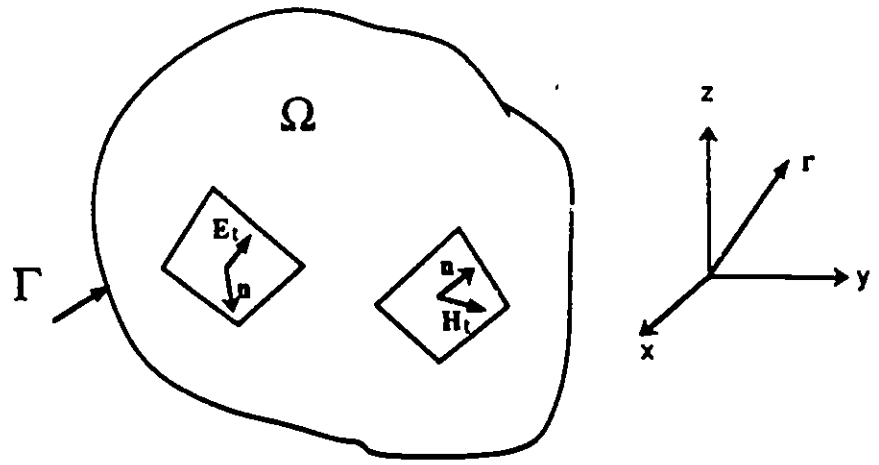


Figure A.1: The domain and boundary of an electromagnetic problem

the regions on either side of Γ.

From Poynting's theorem, the instantaneous electromagnetic power crossing the surface Γ can be expressed as:

$$p_\Gamma \equiv -\oint_\Gamma (\boldsymbol{E} \times \boldsymbol{H}) \cdot \boldsymbol{n}\, dS, \tag{A.3}$$

or in terms of the tangential field components as:

$$p_\Gamma \equiv -\oint_\Gamma (\boldsymbol{E}_t \times \boldsymbol{H}_t)\, dS. \tag{A.4}$$

Expanding Eq. (A.3), we can write:

$$p_\Gamma \equiv p_J(t) + \frac{d}{dt} W_e(t) + \frac{d}{dt} W_m(t), \tag{A.5}$$

where $p_J(t)$ is the power lost due to Joule heating, and $W_e(t)$ and $W_m(t)$ are respectively the energy stored in the electric and magnetic fields.

$$\begin{aligned} p_J(t) &= \int_\Omega \frac{1}{\sigma} \frac{J^2}{2} dV \\ W_e(t) &= \int_\Omega \frac{\varepsilon E^2}{2} dV, \\ W_m(t) &= \int_\Omega \frac{\mu H^2}{2} dV. \end{aligned} \tag{A.6}$$

Based on Eqs. (A.1) to (A.5) the following two corollaries are developed.

Corollary I

The homogeneous problem corresponding to the initial conditions at rest:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{r},0) &= \boldsymbol{F}_e(\boldsymbol{r}) \equiv 0, \\ \boldsymbol{H}(\boldsymbol{r},0) &= \boldsymbol{F}_m(\boldsymbol{r}) \equiv 0, \end{aligned} \tag{A.7}$$

must produce a trivial solution:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{r},t) &\equiv 0, \\ \boldsymbol{H}(\boldsymbol{r},t) &\equiv 0. \end{aligned} \tag{A.8}$$

The following analysis shows this. We see that because the electromagnetic fields are initially zero, the electromagnetic energy must also be zero at time zero.

$$W_e(0) + W_m(0) = 0. \tag{A.9}$$

Since there is no initial electomagnetic energy in the region, and because there is no energy coupled across Γ (relations (A.1) and (A.2)), then the total power in Ω will always be zero:

$$p_\Gamma(t) = 0. \tag{A.10}$$

From Eqs. (A.5) and (A.10), the energies due to electric and magnetic field storage, and Joule heating are related as:

$$-\left(W_e(t) + W_m(t)\right) = \int_0^t p_J(t)\, dt \geq 0, \tag{A.11}$$

which is positive since the power dissipated by Joule heating is positively defined. Likewise the electromagnetic energy, $W_e(t)$ and $W_m(t)$, must also be positively defined, thus the only way Eq. (A.11) can be satisfied is if the stored energy due to the electric and magnetic fields is zero at all times:

$$W_e(t) + W_m(t) \equiv 0 \qquad t \in (0, \infty). \tag{A.12}$$

Thus from Eq. (A.12) and the fact that the integrals of the energies in Eq. (A.6) are positive definite functions of E and H, then Eq. (A.7) must be true.

Corollary II

The homogeneous problem corresponding to initial conditions:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{r},0) &= \boldsymbol{F}_e(\boldsymbol{r}), \\ \boldsymbol{H}(\boldsymbol{r},0) &= \boldsymbol{F}_m(\boldsymbol{r}), \end{aligned} \tag{A.13}$$

produces a particular solution of functions $\boldsymbol{E}(\boldsymbol{r},t)$ and $\boldsymbol{H}(\boldsymbol{r},t)$.

If we have two solutions, $\boldsymbol{E}^{(1)}$ and $\boldsymbol{E}^{(2)}$, $\boldsymbol{H}^{(1)}$ and $\boldsymbol{H}^{(2)}$, for the same initial conditions, then we can write the difference between them as:

$$\begin{aligned} \boldsymbol{E}^{(d)}(\boldsymbol{r},t) &\equiv \boldsymbol{E}^{(1)}(\boldsymbol{r},t) - \boldsymbol{E}^{(2)}(\boldsymbol{r},t), \\ \boldsymbol{H}^{(d)}(\boldsymbol{r},t) &\equiv \boldsymbol{H}^{(1)}(\boldsymbol{r},t) - \boldsymbol{H}^{(2)}(\boldsymbol{r},t). \end{aligned} \tag{A.14}$$

At $t = 0$,

$$\begin{aligned} \boldsymbol{E}^{(d)}(\boldsymbol{r},0) &= 0, \\ \boldsymbol{H}^{(d)}(\boldsymbol{r},0) &= 0, \end{aligned} \tag{A.15}$$

since the two solutions had the same initial conditions.

Based on the linearity of the solutions, $\boldsymbol{E}^{(d)}(\boldsymbol{r},t)$ and $\boldsymbol{H}^{(d)}(\boldsymbol{r},t)$ conform to a homogeneous problem, with zero initial conditions, which according to Corollary I means:

$$\begin{aligned} \boldsymbol{E}^{(d)}(\boldsymbol{r},t) &\equiv 0, \\ \boldsymbol{H}^{(d)}(\boldsymbol{r},t) &\equiv 0,. \end{aligned} \tag{A.16}$$

From Eq. (A.14), we see that Eq. (A.16) is satisfied only if the two solutions are identical.

$$\begin{aligned} \boldsymbol{E}^{(1)}(\boldsymbol{r},t) &= \boldsymbol{E}^{(2)}(\boldsymbol{r},t), \\ \boldsymbol{H}^{(1)}(\boldsymbol{r},t) &= \boldsymbol{H}^{(2)}(\boldsymbol{r},t). \end{aligned} \tag{A.17}$$

We can see that for a particular set of initial conditions, there is only one solution for the fields $\boldsymbol{E}(\boldsymbol{r},t)$ and $\boldsymbol{H}(\boldsymbol{r},t)$, in the region Ω.

Appendix B

The Diffusion Problem in One Dimension

This appendix contains the analytical derivation in the time domain for the solution to the one dimensional diffusion problem. See Fig. 3.1 for a picture of the geometry.

The slab has a finite thickness of $2a$ in the x–dimension, but it extends infinitely in the y and z dimensions. The excitation is a unit step impulse applied on both sides of the slab with the direction of the magnetic field perpendicular to x.

The Helmholtz time domain equation which describes a wave propagating through a conducting material in one dimension is

$$\frac{\partial^2}{\partial x^2}\boldsymbol{H}(x,t) - \mu\sigma\frac{\partial}{\partial t}\boldsymbol{H}(x,t) = 0 \tag{B.1}$$

The Laplace transform of Eq. (B.1) is

$$\frac{\partial^2}{\partial x^2}\mathcal{H}(x,s) - s\mu\sigma\mathcal{H}(x,s) + \mu\sigma\boldsymbol{H}(x,0) = 0, \tag{B.2}$$

where $\mathcal{H}(x,s)$ is the Laplace transform of $\boldsymbol{H}(x,t)$, and $\boldsymbol{H}(x,0)$ is the initial condition of the magnetic field. However, since the field is initially zero, Eq. (B.2) is simply

$$\frac{\partial^2}{\partial x^2}\mathcal{H}(x,s) - s\mu\sigma\mathcal{H}(x,s) = 0, \tag{B.3}$$

The general solution to Eq. (B.3) is

$$\mathcal{H}(x,s) = A\cos(\sqrt{-\mu\sigma s}\,x) + B\sin(\sqrt{-\mu\sigma s}\,x), \tag{B.4}$$

but due to the symmetry of the problem, the B term is zero.

To find the constant A, apply the boundary condition

$$\mathcal{H}(\pm a, s) = \mathcal{L}[1(t)] = \frac{1}{s},$$

as

$$\mathcal{H}(\pm a, s) = A\cos(\sqrt{-\mu\sigma s}\, a) = \frac{1}{s}.$$

Thus A can be found as

$$A = \frac{1}{s}\frac{1}{\cos\sqrt{-\mu\sigma s}\, a}.$$

so that in the Laplace domain, the solution is

$$\mathcal{H}(x, s) = \frac{1}{s}\frac{\cos\sqrt{-\mu\sigma s}\, x}{\cos\sqrt{-\mu\sigma s}\, a}. \tag{B.5}$$

The time domain response can be found by transforming Eq. (B.5) with Heaviside's formula:

$$\begin{aligned} H(x,t) &= \mathcal{L}^{-1}\left[\frac{P(s)}{sQ(s)}\right] \\ &= \frac{P(0)}{Q(0)} + \sum_{k=1}^{\infty}\frac{P(s_k)}{Q'(s_k)}e^{s_k t}, \end{aligned} \tag{B.6}$$

where s_k are the poles of $Q(s)$. For our problem,

$$P(s_k) = \cos\sqrt{-\mu\sigma s}x, \tag{B.7}$$

$$Q(s_k) = \cos\sqrt{-\mu\sigma s a^2}. \tag{B.8}$$

The poles of $Q(s)$ are found by

$$\begin{aligned} \sqrt{-\mu\sigma s_k a^2} &= (2k+1)\frac{\pi}{2}; \qquad k = 0, 1, 2, \ldots \\ s_k &= -\left[(2k+1)\frac{\pi}{2}\right]^2 \frac{1}{\mu\sigma a^2} \end{aligned}$$

The derivative of $Q(s)$ with respect to s is

$$Q'(s) = \sqrt{\mu\sigma a^2}\left(\frac{-1}{2\sqrt{-s}}\right)(\sin\sqrt{-\mu\sigma a^2 s}). \tag{B.9}$$

By combining Eqs. (B.6), (B.7), and (B.9), and simplifying, we find the solution is

$$H(x,t) = 1 - \sum_{k=0}^{\infty}(-1)^k \frac{2}{(2k+1)\frac{\pi}{2}}\cos\left((2k+1)\frac{\pi}{2}\frac{x}{a}\right)e^{-[(2k+1)\frac{\pi}{2}]^2/(\mu\sigma a^2)t}. \tag{B.10}$$

Appendix C

Fourier Transform

An initial approximation to finding the time domain response of a system to a time varying input signal $i(t)$, is to find the inverse Fourier transform of the frequency domain response of the system ($H(\omega) \rightarrow h(t)$), and then to convolve it with the input time domain signal. Thus we write:

$$h(t) = \mathcal{F}^{-1}\{H(\omega)\},$$

and then,

$$o(t) = i(t) * h(t),$$

where $o(t)$ is the time varying output signal. The most efficient way to perform the inverse Fourier transform for a discrete frequency response, is to use the inverse fast Fourier transform (IFFT) technique. One such algorithm is described in [29].

Appendix D

Computer Programs for the Finite Element Method

These programs were written and executed on a Digital Electronic Corporation (DEC) VAX 3500 computer.

DORS.FOR

```
C*********************************************************************
      PROGRAM DORS
C     3D-FEM
C
C           ***********************************
C           *                                 *
C           *      PROGRAM  DORS              *
C           *                                 *
C           ***********************************
C
C                        D  O  R  S
C
C                DYNAMICS*OF*RADIATING*STRUCTURES
C
C    DESCRIPTION:
C    ------------
C           This program uses the Finite Element Method to solve
C     dynamic (steady-state) boundary value problems in three
C     dimensions.
C
C           The region of interest must be broken into a number of
C     eight noded solids (hexahedrons) and the potentials will be
C     calculated at each node.
C
C           The program solves the Diffusion equation for the scalar
C     electric potential PHI and the vector magnetic potential A.  The
```

```
C     set of equations developed are solved by using Galerkin's
C     method applied to finite elements.
C
C           The information of the structure geometry is read
C     in from an external data file.  The default file name is
C     'BRICK.GEO'.
C
C
C           HEADER INFORMATION  PRECEDES THE DASHED LINE
C'-------------------------------------------------------------'
C     'H_INC_X','H_INC_Y','H_INC_Z'       FORMAT(3(2(1X,1PE12.4)))
C     'NUM_FREQ'                          FORMAT(1X,I8)
C     'FREQUENCY'                         FORMAT(1X,1PE12.4)
C     'NUM_NODE','NUM_BRICK','NUM_SUR'    FORMAT(1X,3I8)
C
C     'BRICK','NODE(1)','(2)','(3)','(4)','(5)','(6)','(7)','(8)',
C          'REL_PERM','CONDUCTIVITY'      FORMAT(1X,9I8,2(1PE12.4))
C
C     'NODE','XX','YY','ZZ','DIRICHLET'   FORMAT(1X,I8,4(1PE12.4),I2)
C
C     'SURFACE','BRICK','FACE'            FORMAT(1X,3I8)
C
C     NUM_FREQ is the number of solutions required.
C     FREQUENCY is the solution frequency.
C     NUM_NODE is the number of nodes in the region.
C     NUM_BRICK  is the number of bricks or elements in the region.
C     NUM_SUR is the number of surfaces that must be integrated over.
C     NODE(1,2,3,4,5,6,7,8) are the 8 nodes of BRICK.
C     X,Y,Z are the coordinates for each node.
C     DIRICHLET  is a logical which indicates whether the node is
C     Dirichlet (has a fixed potential) on not.
C       DIRICHLET=0 indicates that the node is not fixed.
C       DIRICHLET=1 indicates that the node is fixed to potential 0.
C     REL_PERM is the relative permittivity of each brick.
C     CONDUCTIVITY is the conductivity of each brick.
C     FACE is the surface on BRICK that must be integrated over.
C
C           The nodes are assigned to the bricks in a particular
C     fashion.  1:(-1,-1,1),2:(1,-1,1),3(1,1,1),4:(-1,1,1),
C     5:(-1,-1,-1),6:(1,-1,-1),7(1,1,-1),8:(-1,1,-1)
C
C           The nodes can be numbered in any order as long as the
C     correct coordinates are assigned.
C
C           The surfaces can be numbered in any order, but the
C     correct surface representation must be used.
C
C           The output consists of the potential at each of the
C     nodes and is written to a user defined file for reference.
C     The magnetic (H) and electric (E) fields can be computed from
C     the potentials by running the program FIELDING.
```

```
C
C*********************************************************************
C
C     NOTE:  This program has been written using VAX FORTRAN;
C            an extended version of FORTRAN 77.  It cannot be run
C            with a FORTRAN 77 compiler without making changes to
C            the DO -- ENDDO and DO WHILE -- ENDDO structures,
C            and the PARAMETER and variable declarations.
C
C************** Property of D. Ladd **********************************
C
C     written by Darcy N. Ladd for partial fulfillment of M.A.Sc
C      thesis: July 1989
C
C*********************************************************************
C
      IMPLICIT NONE
C
C   DECLARATIONS:
C   -------------
C
      PARAMETER ORDER=252
      PARAMETER POINT=505
      PARAMETER REGION=340
      PARAMETER FACES=292
C
      COMPLEX*16  MATA(ORDER*4,ORDER*4)  ! S matrix
      COMPLEX*16  MATB(ORDER*4)          ! B vector
      COMPLEX*16  AA,BB,CC,DD,EE,FF      ! coefficients of equation
                                         !  terms
      COMPLEX*16  H_INC_X,H_INC_Y,H_INC_Z! illuminating magnetic field
      COMPLEX*16  SM(32,32)              ! S matrix terms
      COMPLEX*16  JC                     ! defined imaginary constant
      COMPLEX*16  COM(POINT*4)           ! work vector used by ZGECO
C
      REAL*8  XX(POINT)            ! X coordinates of each node
      REAL*8  YY(POINT)            ! Y coordinates of each node
      REAL*8  ZZ(POINT)            ! Z coordinates of each node
      REAL*8  REL_PERM(REGION)     ! relative permittivity of each
                                   !  brick element
      REAL*8  REL_MU               ! relative permeability
      REAL*8  CONDUCTIVITY(REGION)! conductivity of each brick element
      REAL*8  X(8),Y(8),Z(8)       ! local coordinates of the nodes
                                   !  in a brick being used to compute
                                   !  interpolatory functions
      REAL*8  S,T,R                ! natural coordinate system points
                                   !  transformed from X,Y,Z
      REAL*8  U,V,W                ! alternate set of coordinates
      REAL*8  E(4),WT(4),WEIGHT    ! evaluation and weighting variables
                                   !  used for Gaussian quadrature
      REAL*8  S1,S2,T1,T2,R1,R2    ! functions of S,T,R
```

```
      REAL*8  N(8)                    ! interpolatory functions for a
                                      !  brick in S,T,R units
      REAL*8  ND(8,3)                 ! derivatives of the interpolatory
                                      !  functions w.r.t S.T,R
      REAL*8  NDX(3),NDY(3),NDZ(3)    ! summation of derivatives times
                                      !  X,Y,Z
      REAL*8  AM(3,3)                 ! A matrix terms
      REAL*8  JACOB                   ! Jacobian for transformation of
                                      !  d(X,Y,Z) to d(S,T,R)
      REAL*8  DETA                    ! determinant of the A matrix
      REAL*8  DX(8),DY(8),DZ(8)       ! the derivatives of the inter-
                                      !  polatory functions w.r.t. X,Y,Z
      REAL*8  ZERO,ONE,GHT            ! parameter definitions
      REAL*8  ENOT,MNOT               ! free-space values of permittivity
                                      !  and permeability
      REAL*8  FREQUENCY(10)           ! the frequencies at which the
                                      !  problem is solved
      REAL*8  OMEGA                   ! the angular frequency at which
                                      !  the present solution is for
      REAL*8  PI                      ! defined constant
      REAL*8  PLANE(3,3)              ! matrix to compute surface normals
      REAL*8  NORMAL(3),MAGN          ! surface normals
      REAL*8  CHECK                   ! a check that the normals point
                                      !   in the correct direction
      REAL*8  RCOND                   ! estimate of the reciprocal
                                      !  condition of a matrix processed
                                      !  by ZGECO
C
      INTEGER   NUM_NODE              ! the total number of nodes in the
                                      !  region being investigated
      INTEGER   NUM_B..               ! the total number of bricks or sub-
                                      !  regions in the region being
                                      !  investigated
      INTEGER   NUM_SUR               ! the total number of surfaces in
                                      !  the region being investigated
      INTEGER   NUM_FREQ              ! the total number of frequencies
                                      !  for which the problem must be
                                      !  solved
      INTEGER   NODE                  ! a node counter
      INTEGER   BRICK                 ! a brick element counter
      INTEGER   SURFACE               ! a surface element counter
      INTEGER   FREQ                  ! a frequency counter
      INTEGER   BRICK_NODE(REGION,8)  ! array containing 8 nodes of each
                                      !  brick element
      INTEGER   SUR_NODE(4)           ! local node info for a surface
                                      !  element
      INTEGER   BACK_NODE(4)          ! local node info for the back
                                      !  surface
      INTEGER   INT_SUR(FACES,2)      ! brick and face information of each
                                      !  surface
      INTEGER   FREE_NODE,DIR_NODE
```

```
      INTEGER   ARRAY_SIZE          ! indicates the size of array MATA
      INTEGER   DIRICHLET(POINT*4,2)
      INTEGER   WORKI(ORDER*4)      ! a vector of pivot indices created
                                    !  by ZGECO
      INTEGER   A,B,C               ! counters used in Gaussian
                                    !  quadrature
      INTEGER   I,J,K,L,M           ! counters
      INTEGER   ROW,COL             ! row and column pointers
C
      CHARACTER*50 FILE             ! input/output file names
      CHARACTER*1  DELIMIT          ! variable used to find the
                                    !  delimiter which indicates the
                                    !  start of the data
C
C
C     EXTERNALS     DGAUSS(SIZE,A,B,N)          ! Gaussian elimination
C                   ZGECO(A,LDA,N,IPVT,RCOND,Z) ! complex Gaussian
C                   ZGESL(A,LDA,N,IPVT,B,JOB)   ! elimination
C
C*********************************************************************
C
C     START
C
      ZERO=0.0D0
      ONE=1.0D0
      GHT=8.0D0
C
      PI=DACOS(-ONE)
C
C   Define the permittivity  and permeability of free space
C
      ENOT=2.7816152D-11/PI
      MNOT=4.0D-7*PI
C
      REL_MU=ONE
C
      JC=DCMPLX(ZERO,ONE)
C
C*********************************************************************
C
C   Open the necessary input/output files.
C
C   Get data from user defined input file.
C    Default input file name is BRICK.GEO.
C
      FILE=' '
      WRITE(*,'(A)') ' What is the input file name?'
      WRITE(*,'(A)') ' The default is "BRICK.GEO"'
      READ(*,'(15A)') FILE
      IF(FILE.EQ.' ') THEN
       FILE='BRICK.GEO'
```

```
      ENDIF
C
      OPEN(UNIT=20,FILE=FILE,STATUS='OLD')
C
C   The output data goes to a user defined file.
C    Default output file name is BRICK.RES.
C
      FILE=' '
      WRITE(*,'(A)') ' What is the output file name?'
      WRITE(*,'(A)') ' The default is "BRICK.RES"'
      READ(*,'(15A)') FILE
C
      IF(FILE.EQ.' ') THEN
       FILE='BRICK.RES'
      ENDIF
C
      OPEN(UNIT=10,FILE=FILE,STATUS='NEW')
C
C*********************************************************************
C
      WRITE(*,'(A)') ' Reading the data in now.'
C
C   Search for the '----------' delimit string which
C   indicates that the data is about to begin.  Ignore
C   anything before this.
C
      DELIMIT=' '
C
      DO WHILE (DELIMIT.NE.'-')
       READ(20,5) DELIMIT
5      FORMAT(1X,1A)
C
      ENDDO
C
C   Read the necessary data from the input file.
C
C     This problem can be solved for different frequencies.
C
C     Get the incident magnetic field.
C
      READ(20,38) H_INC_X,H_INC_Y,H_INC_Z
38    FORMAT(6(1X,1PE12.4))
C
      READ(20,*) NUM_FREQ
40    FORMAT(1X,I8)
C
      IF (NUM_FREQ.GT.10) THEN
       WRITE(*,*) ' There has been a dimensioning error for NUM_FREQ!'
       STOP
      ENDIF
      DO I=1,NUM_FREQ
```

```
        READ(20,*) FREQUENCY(I)
45      FORMAT(1X,1PE12.4)
       ENDDO
C
       READ(20,*) NUM_NODE,NUM_BRICK,NUM_SUR
10     FORMAT(1X,3I8)
C
       IF (NUM_BRICK.GT.REGION) THEN
        WRITE(*,*) 'There has been a dimensioning error for NUM_BRICK!'
        STOP
       ENDIF
C
       IF (NUM_NODE.GT.POINT) THEN
        WRITE(*,*) ' There has been a dimensioning error for NUM_NODE!'
        STOP
       ENDIF
C
       IF (NUM_SUR.GT.FACES) THEN
        WRITE(*,*) ' There has been a dimensioning error for NUM_SUR!'
        STOP
       ENDIF
C
C   Get the node assignments for each brick.
C
       WRITE(*,'(A)') ' Getting brick information'
       DO I=1,NUM_BRICK
        READ(20,*) BRICK,(BRICK_NODE(BRICK,J),J=1,8),REL_PERM(BRICK),
      $                    CONDUCTIVITY(BRICK)
20      FORMAT(1X,9I8,2(1PE12.4))
       ENDDO  ! I
C
C   Read the coordinates and potentials of each node.
C
       WRITE(*,'(A)') ' Getting node information'
       DIR_NODE=ZERO
       DO I=1,NUM_NODE
        READ(20,*) NODE,XX(NODE),YY(NODE),ZZ(NODE),DIRICHLET(NODE,1)
30      FORMAT(1X,I8,3(1PE12.4),1X,I1)
        DIRICHLET(I,2)=DIR_NODE
        DIR_NODE=DIR_NODE+DIRICHLET(NODE,1)
       ENDDO  ! I
       FREE_NODE=NUM_NODE-DIR_NODE
       WRITE(*,*) ' FREE_NODE = ',FREE_NODE
C
       IF (FREE_NODE.GT.ORDER) THEN
        WRITE(*,*) 'There has been a dimensioning error for FREE_NODE!'
        STOP
       ENDIF
C
       WRITE(*,'(A)') ' Getting surface information'
       DO I=1,NUM_SUR
```

```
        READ(20,*) SURFACE,(INT_SUR(SURFACE,J),J=1,2)
35      FORMAT(1X,3I8)
       ENDDO  ! I
C
       CLOSE(UNIT=20)
C
C*************************************************************************
C
C      Solve the Helmholtz equation over the regions of interest.
C      Solve for the frequencies specified.
C
       DO FREQ=1,NUM_FREQ
        OMEGA=2.0D0*PI*FREQUENCY(FREQ)
C

C*************************************************************************
C
C      Initialize the MATA and MATB matrices to zero.
C
       ARRAY_SIZE=ORDER*4
       DO I=1,ARRAY_SIZE
C
        MATB(I)=DCMPLX(ZERO,ZERO)
        DO J=1,ARRAY_SIZE
         MATA(J,I)=DCMPLX(ZERO,ZERO)
C
        ENDDO
       ENDDO
C
C*************************************************************************
C
       WRITE(*,'(A)') ' Building the system of equations'
C
C      Generate an SM matrix for each brick.
C
       DO BRICK=1,NUM_BRICK
C
C*************************************************************************
C
C      Transfer the global node data to local node data.
       DO NODE=1,8
        X(NODE)=XX(BRICK_NODE(BRICK,NODE))
        Y(NODE)=YY(BRICK_NODE(BRICK,NODE))
        Z(NODE)=ZZ(BRICK_NODE(BRICK,NODE))
       ENDDO  ! NODE
C
C      Initialize the SM matrix.
C
       DO J=1,32
        DO I=1,32
         SM(I,J)=DCMPLX(ZERO,ZERO)
```

```
       ENDDO ! I
      ENDDO  ! J
C
C*************************************************************************
C     Derive the expression to be integrated.
C     Use Gaussian quadrature to perform the integration over the
C     volume
C
C*********************************************
C
      E(1)=0.3399810436
      E(2)=0.8611363116
      E(3)=-E(2)
      E(4)=-E(1)
C
      WT(1)=0.6521451549
      WT(2)=0.3478548451
      WT(3)=WT(2)
      WT(4)=WT(1)
C
      DO A=1,4
       DO B=1,4
        DO C=1,4
C*********************************************
c      E(1)=DSQRT(1.0D0/3.0D0)
c      E(2)=-E(1)
cC
c      WT(1)=ONE
c      WT(2)=ONE
cC
c      DO A=1,2
c       DO B=1,2
c        DO C=1,2
C*********************************************
C
         S=E(C)
         T=E(B)
         R=E(A)
C
         WEIGHT=WT(A)*WT(B)*WT(C)
C
C*************************************************************************
C
C     Generate the interpolatory functions.
C
      S1=ONE-S
      S2=ONE+S
      T1=ONE-T
      T2=ONE+T
      R1=ONE-R
      R2=ONE+R
```

```
C
      N(1)=S1*T1*R2/GHT
      N(2)=S2*T1*R2/GHT
      N(3)=S2*T2*R2/GHT
      N(4)=S1*T2*R2/GHT
      N(5)=S1*T1*R1/GHT
      N(6)=S2*T1*R1/GHT
      N(7)=S2*T2*R1/GHT
      N(8)=S1*T2*R1/GHT
C
C     Calculate the derivative terms of the interpolative functions
C     with respect to S, T, R.
C     ND(I,D)  D is the dimension:  d/dS: J=1, d/dT: J=2, d/dR: J=3
C
C     Here are the derivatives w.r.t. S
C
      ND(1,1)=-T1*R2/GHT
      ND(2,1)=T1*R2/GHT
      ND(3,1)=T2*R2/GHT
      ND(4,1)=-T2*R2/GHT
      ND(5,1)=-T1*R1/GHT
      ND(6,1)=T1*R1/GHT
      ND(7,1)=T2*R1/GHT
      ND(8,1)=-T2*R1/GHT
C
C     Here are the derivatives w.r.t. T.
C
      ND(1,2)=-S1*R2/GHT
      ND(2,2)=-S2*R2/GHT
      ND(3,2)=S2*R2/GHT
      ND(4,2)=S1*R2/GHT
      ND(5,2)=-S1*R1/GHT
      ND(6,2)=-S2*R1/GHT
      ND(7,2)=S2*R1/GHT
      ND(8,2)=S1*R1/GHT
C
C     Here are the derivatives w.r.t. R
C
      ND(1,3)=S1*T1/GHT
      ND(2,3)=S2*T1/GHT
      ND(3,3)=S2*T2/GHT
      ND(4,3)=S1*T2/GHT
      ND(5,3)=-S1*T1/GHT
      ND(6,3)=-S2*T1/GHT
      ND(7,3)=-S2*T2/GHT
      ND(8,3)=-S1*T2/GHT
C
C*****************************************************************************
C
C     Compute the AM(I,J) terms.
C     ND*(J)  J is the dimension   d/ds: J=1, d/dT: J=2, d/dR: J=3
```

```
C
      DO J=1,3
       NDX(J)=ZERO
       NDY(J)=ZERO
       NDZ(J)=ZERO
C
       DO I=1,8
C
         NDX(J)=NDX(J)+ND(I,J)*X(I)
         NDY(J)=NDY(J)+ND(I,J)*Y(I)
         NDZ(J)=NDZ(J)+ND(I,J)*Z(I)
C
       ENDDO ! I
      ENDDO  ! J
C
      AM(1,1)=NDY(2)*NDZ(3)-NDZ(2)*NDY(3)
      AM(2,1)=NDZ(2)*NDX(3)-NDX(2)*NDZ(3)
      AM(3,1)=NDX(2)*NDY(3)-NDY(2)*NDX(3)
      AM(1,2)=NDY(3)*NDZ(1)-NDZ(3)*NDY(1)
      AM(2,2)=NDZ(3)*NDX(1)-NDX(3)*NDZ(1)
      AM(3,2)=NDX(3)*NDY(1)-NDY(3)*NDX(1)
      AM(1,3)=NDY(1)*NDZ(2)-NDZ(1)*NDY(2)
      AM(2,3)=NDZ(1)*NDX(2)-NDX(1)*NDZ(2)
      AM(3,3)=NDX(1)*NDY(2)-NDY(1)*NDX(2)
C
C     Compute the determinant of matrix A
C
      DETA=AM(1,1)*NDX(1)+AM(2,1)*NDY(1)+AM(3,1)*NDZ(1)
      JACOB=DETA
C
      IF (DETA.EQ.ZERO) THEN
       WRITE(*,'(A)') ' ERROR: DETERMINANT A IS ZERO'
       STOP
      ENDIF
C
C*********************************************************************
C
C    Compute the derivatives of the interpolatory terms w.r.t. x,y,z
C
       DO I=1,8
        DX(I)=ZERO
        DY(I)=ZERO
        DZ(I)=ZERO
C
        DO K=1,3
C
          DX(I)=DX(I)+AM(1,K)*ND(I,K)/DETA
          DY(I)=DY(I)+AM(2,K)*ND(I,K)/DETA
          DZ(I)=DZ(I)+AM(3,K)*ND(I,K)/DETA
C
        ENDDO
```

```
      ENDDO
C
C*********************************************************************
C
C     Compute the terms that must be integrated over the volume
C
      AA=ONE/(REL_MU*MNOT)
      BB=-(OMEGA**2*REL_PERM(BRICK)*ENOT-JC*OMEGA*CONDUCTIVITY(BRICK))
      CC=CONDUCTIVITY(BRICK)+JC*OMEGA*REL_PERM(BRICK)*ENOT
      DD=CONDUCTIVITY(BRICK)+JC*OMEGA*REL_PERM(BRICK)*ENOT
      EE=-(OMEGA**2*REL_PERM(BRICK)*ENOT-JC*OMEGA*CONDUCTIVITY(BRICK))
C
      DO J=1,8
       DO I=1,8
C
         SM(I,J)=SM(I,J)+(AA*(DZ(I)*DZ(J)+DY(I)*DY(J))
     $             +BB*N(I)*N(J))*JACOB*WEIGHT
         SM(I+8,J)=SM(I+8,J)-AA*DX(I)*DY(J)*JACOB*WEIGHT
         SM(I+16,J)=SM(I+16,J)-AA*DX(I)*DZ(J)*JACOB*WEIGHT
         SM(I+24,J)=SM(I+24,J)+EE*DX(I)*N(J)*JACOB*WEIGHT
C
         SM(I,J+8)=SM(I,J+8)-AA*DY(I)*DX(J)*JACOB*WEIGHT
         SM(I+8,J+8)=SM(I+8,J+8)+(AA*(DX(I)*DX(J)+DZ(I)*DZ(J))
     $                           +BB*N(I)*N(J))*JACOB*WEIGHT
         SM(I+16,J+8)=SM(I+16,J+8)-AA*DY(I)*DZ(J)*JACOB*WEIGHT
         SM(I+24,J+8)=SM(I+24,J+8)+EE*DY(I)*N(J)*JACOB*WEIGHT
C
         SM(I,J+16)=SM(I,J+16)-AA*DZ(I)*DX(J)*JACOB*WEIGHT
         SM(I+8,J+16)=SM(I+8,J+16)-AA*DZ(I)*DY(J)*JACOB*WEIGHT
         SM(I+16,J+16)=SM(I+16,J+16)+(AA*(DX(I)*DX(J)+DY(I)*DY(J))
     $             +BB*N(I)*N(J))*JACOB*WEIGHT
         SM(I+24,J+16)=SM(I+24,J+16)+EE*DZ(I)*N(J)*JACOB*WEIGHT
C
         SM(I,J+24)=SM(I,J+24)+CC*N(I)*DX(J)*JACOB*WEIGHT
         SM(I+8,J+24)=SM(I+8,J+24)+CC*N(I)*DY(J)*JACOB*WEIGHT
         SM(I+16,J+24)=SM(I+16,J+24)+CC*N(I)*DZ(J)*JACOB*WEIGHT
         SM(I+24,J+24)=SM(I+24,J+24)+(DD*(DX(I)*DX(J)+DY(I)*DY(J)
     $                                 +DZ(I)*DZ(J)))*JACOB*WEIGHT
C
       ENDDO ! I
      ENDDO  ! J
C
C*********************************************************************
C
        ENDDO        ! C
       ENDDO         ! B
      ENDDO          ! A
C
C*********************************************************************
C
C     Add the SM matrix terms to the the MATA matrix.
```

```
C
      DO L=1,4
       DO J=1,8
C
         IF (DIRICHLET(BRICK_NODE(BRICK,J),1).NE.1) THEN
C         Not Dirichlet B.C.
C
          ROW=BRICK_NODE(BRICK,J)-DIRICHLET(BRICK_NODE(BRICK,J),2)
     $        +(L-1)*FREE_NODE
C
          DO K=1,4
           DO I=1,8
C
            IF (DIRICHLET(BRICK_NODE(BRICK,I),1).NE.1) THEN
C            Not Dirichlet B.C.
C
             COL=BRICK_NODE(BRICK,I)-DIRICHLET(BRICK_NODE(BRICK,I),2)
     $          +(K-1)*FREE_NODE
C
             MATA(ROW,COL)=MATA(ROW,COL)+SM((L-1)*8+J,(K-1)*8+I)
C
            ELSE
C
C   The Dirichlet values are zero for this problem.
C
             MATB(ROW)=MATB(ROW)-SM((L-1)*8+J,(K-1)*8+I)*ZERO
C
            ENDIF     ! COL
           ENDDO      ! J
          ENDDO       ! L
         ENDIF        ! ROW
        ENDDO         ! I
       ENDDO          ! K
C
C*********************************************************************
C
      ENDDO           ! BRICK
C
C*********************************************************************
C
C    Generate an SM matrix for each surface element
C
      DO SURFACE=1,NUM_SUR
C
C*********************************************************************
C
C    Check to see which brick the surface is on.
C
      BRICK=INT_SUR(SURFACE,1)
C
C    Generate a local set of nodes corresponding to the surface
```

```
C     being integrated.
C
      IF (INT_SUR(SURFACE,2).EQ.1) THEN
C     Surface 1 defined by nodes 1,5,6,2
C
      SUR_NODE(1)=1
      SUR_NODE(2)=5
      SUR_NODE(3)=6
      SUR_NODE(4)=2
C
      BACK_NODE(1)=3
      BACK_NODE(2)=7
      BACK_NODE(3)=8
      BACK_NODE(4)=4
C
      ELSEIF (INT_SUR(SURFACE,2).EQ.2) THEN
C     Surface 2 defined by nodes 2,6,7,3
C
      SUR_NODE(1)=2
      SUR_NODE(2)=6
      SUR_NODE(3)=7
      SUR_NODE(4)=3
C
      BACK_NODE(1)=1
      BACK_NODE(2)=4
      BACK_NODE(3)=8
      BACK_NODE(4)=5
C
      ELSEIF (INT_SUR(SURFACE,2).EQ.3) THEN
C     Surface 3 defined by nodes 1,2,3,4
C
      SUR_NODE(1)=1
      SUR_NODE(2)=2
      SUR_NODE(3)=3
      SUR_NODE(4)=4
C
      BACK_NODE(1)=5
      BACK_NODE(2)=8
      BACK_NODE(3)=7
      BACK_NODE(4)=6
C
      ELSEIF (INT_SUR(SURFACE,2).EQ.4) THEN
C     Surface 4 defined by nodes 5,8,7,6
C
      SUR_NODE(1)=5
      SUR_NODE(2)=8
      SUR_NODE(3)=7
      SUR_NODE(4)=6
C
      BACK_NODE(1)=1
      BACK_NODE(2)=2
```

```
      BACK_NODE(3)=3
      BACK_NODE(4)=4
C
      ELSEIF (INT_SUR(SURFACE,2).EQ.5) THEN
C     Surface 5 defined by nodes 1,4,8,5
C
      SUR_NODE(1)=1
      SUR_NODE(2)=4
      SUR_NODE(3)=8
      SUR_NODE(4)=5
C
      BACK_NODE(1)=2
      BACK_NODE(2)=6
      BACK_NODE(3)=7
      BACK_NODE(4)=3
C
      ELSEIF (INT_SUR(SURFACE,2).EQ.6) THEN
C     Surface 6 defined by nodes 3,7,8,4
C
      SUR_NODE(1)=3
      SUR_NODE(2)=7
      SUR_NODE(3)=8
      SUR_NODE(4)=4
C
      BACK_NODE(1)=1
      BACK_NODE(2)=5
      BACK_NODE(3)=6
      BACK_NODE(4)=2
C
      ENDIF
C
C     Calculate the unit normal to each surface in the x,y,z
C      domain from three corner nodes of each surface.
C      By knowing 3 nodes, it is possible to generate the equation
C      of a plane in the form ax+by+cz+d=0.  (a,b,c) represents a
C      normal vector to that plane.
C
      DO I=1,3
       PLANE(I,1)=XX(BRICK_NODE(BRICK,SUR_NODE(I)))
       PLANE(I,2)=YY(BRICK_NODE(BRICK,SUR_NODE(I)))
       PLANE(I,3)=ZZ(BRICK_NODE(BRICK,SUR_NODE(I)))
       NORMAL(I)=-ONE
      ENDDO    ! I
C
C     Solve the matrix PLANE to find NORMAL, which contains
C     the components of the normal vector.
C
      CALL DGAUSS(3,PLANE,NORMAL,3)
C
      MAGN=DSQRT(NORMAL(1)**2+NORMAL(2)**2+NORMAL(3)**2)
C
```

```
C     What about the direction of the normal??????
C      Create a vector which points out by computing the midpoints
C      of the front and back surfaces.  Dot product this vector with
C      the computed normal, and if the result is negative, the
C      direction of the normal must be reversed.
C
       CHECK=ZERO
       DO I=1,3
        PLANE(I,1)=ZERO
       ENDDO
C
       DO I=1,2
        PLANE(1,1)=PLANE(1,1)+XX(BRICK_NODE(BRICK,SUR_NODE(I)))/2.0D0
      $     -XX(BRICK_NODE(BRICK,BACK_NODE(I)))/2.0D0
        PLANE(2,1)=PLANE(2,1)+YY(BRICK_NODE(BRICK,SUR_NODE(I)))/2.0D0
      $     -YY(BRICK_NODE(BRICK,BACK_NODE(I)))/2.0D0
        PLANE(3,1)=PLANE(3,1)+ZZ(BRICK_NODE(BRICK,SUR_NODE(I)))/2.0D0
      $     -ZZ(BRICK_NODE(BRICK,BACK_NODE(I)))/2.0D0
       ENDDO
C
       DO I=1,3
        CHECK=CHECK+PLANE(I,1)*NORMAL(I)
       ENDDO
C
       IF (CHECK.LT.ZERO) MAGN=-MAGN      ! The normal is the wrong way.
       DO I=1,3
        NORMAL(I)=NORMAL(I)/MAGN
       ENDDO
C
C*********************************************************************
C
C     Transfer the global node data to local node data.
C
       DO NODE=1,8
        X(NODE)=XX(BRICK_NODE(BRICK,NODE))
        Y(NODE)=YY(BRICK_NODE(BRICK,NODE))
        Z(NODE)=ZZ(BRICK_NODE(BRICK,NODE))
       ENDDO     ! NODE
C
C     Initialize the SM matrix.
C
       DO J=1,32
        DO I=1,32
         SM(I,J)=DCMPLX(ZERO,ZERO)
        ENDDO ! I
       ENDDO  ! J
C
C*********************************************************************
C
C     Calculate the Jacobian.  It is (the area of the surface in x,y,z)
C      divided by (the area in s,t,r which equals 4).
```

```
C     The area of the surface is computed by the principle that the
C     cross product of two vectors is the area of the parallelogram
C     they define. 4 vectors are computed using the x,y,z coordinates
C     of the surface.  The area of two triangles (each half the
C     surface) is computed, summed and divided by 4.0D0.
C
      NDX(1)=X(SUR_NODE(4))-X(SUR_NODE(1))
      NDX(2)=Y(SUR_NODE(4))-Y(SUR_NODE(1))
      NDX(3)=Z(SUR_NODE(4))-Z(SUR_NODE(1))
      NDY(1)=X(SUR_NODE(2))-X(SUR_NODE(1))
      NDY(2)=Y(SUR_NODE(2))-Y(SUR_NODE(1))
      NDY(3)=Z(SUR_NODE(2))-Z(SUR_NODE(1))
C
      N(1)=NDX(2)*NDY(3)-NDX(3)*NDY(2)
      N(2)=NDX(3)*NDY(1)-NDX(1)*NDY(3)
      N(3)=NDX(1)*NDY(2)-NDX(2)*NDY(1)
      S1=DSQRT(N(1)**2+N(2)**2+N(3)**2)
C
      NDX(1)=X(SUR_NODE(2))-X(SUR_NODE(3))
      NDX(2)=Y(SUR_NODE(2))-Y(SUR_NODE(3))
      NDX(3)=Z(SUR_NODE(2))-Z(SUR_NODE(3))
      NDY(1)=X(SUR_NODE(4))-X(SUR_NODE(3))
      NDY(2)=Y(SUR_NODE(4))-Y(SUR_NODE(3))
      NDY(3)=Z(SUR_NODE(4))-Z(SUR_NODE(3))
C
      N(1)=NDX(2)*NDY(3)-NDX(3)*NDY(2)
      N(2)=NDX(3)*NDY(1)-NDX(1)*NDY(3)
      N(3)=NDX(1)*NDY(2)-NDX(2)*NDY(1)
      S2=DSQRT(N(1)**2+N(2)**2+N(3)**2)
C
      JACOB=(S1+S2)/8.0D0
C
C*************************************************************************
C
C    Use Gaussian quadrature to integrate over the surface.
C
C*********************************************
      E(1)=0.8611363116
      E(2)=0.3399810436
      E(3)=-E(2)
      E(4)=-E(1)
C
      WT(1)=0.3478548451
      WT(2)=0.6521451549
      WT(3)=WT(2)
      WT(4)=WT(1)
C
      DO A=1,4
       DO B=1,4
C*********************************************
c      E(1)=DSQRT(1.0D0/3.0D0)
```

```
c       E(2)=-E(1)
cC
c       WT(1)=ONE
c       WT(2)=ONE
cC
c       DO A=1,2
c        DO B=1,2
C*********************************************
C
         U=E(B)
         V=E(A)
         W=ONE
C
         WEIGHT=WT(A)*WT(B)
C
C     Set the levels of S,T,R according to the plane of integration
C
       IF (INT_SUR(SURFACE,2).EQ.1) THEN
        S=V
        T=-W
        R=U
       ELSEIF (INT_SUR(SURFACE,2).EQ.2) THEN
        S=W
        T=U
        R=V
       ELSEIF (INT_SUR(SURFACE,2).EQ.3) THEN
        S=U
        T=V
        R=W
       ELSEIF (INT_SUR(SURFACE,2).EQ.4) THEN
        S=U
        T=V
        R=-W
       ELSEIF (INT_SUR(SURFACE,2).EQ.5) THEN
        S=-W
        T=U
        R=V
       ELSEIF (INT_SUR(SURFACE,2).EQ.6) THEN
        S=V
        T=W
        R=U
       ENDIF
C
C*****************************************************************************
C
C     Generate the interpolatory functions.
C
       S1=ONE-S
       S2=ONE+S
       T1=ONE-T
       T2=ONE+T
```

```
      R1=ONE-R
      R2=ONE+R
C
      N(1)=S1*T1*R2/GHT
      N(2)=S2*T1*R2/GHT
      N(3)=S2*T2*R2/GHT
      N(4)=S1*T2*R2/GHT
      N(5)=S1*T1*R1/GHT
      N(6)=S2*T1*R1/GHT
      N(7)=S2*T2*R1/GHT
      N(8)=S1*T2*R1/GHT
C
C     Calculate the derivative terms of the interpolative functions
C     with respect to S, T, R.
C     ND(I,D)  D is the dimension:  d/dS: J=1, d/dT: J=2, d/dR: J=3
C
C     Here are the derivatives w.r.t. S
C
      ND(1,1)=-T1*R2/GHT
      ND(2,1)=T1*R2/GHT
      ND(3,1)=T2*R2/GHT
      ND(4,1)=-T2*R2/GHT
      ND(5,1)=-T1*R1/GHT
      ND(6,1)=T1*R1/GHT
      ND(7,1)=T2*R1/GHT
      ND(8,1)=-T2*R1/GHT
C
C     Here are the derivatives w.r.t. T.
C
      ND(1,2)=-S1*R2/GHT
      ND(2,2)=-S2*R2/GHT
      ND(3,2)=S2*R2/GHT
      ND(4,2)=S1*R2/GHT
      ND(5,2)=-S1*R1/GHT
      ND(6,2)=-S2*R1/GHT
      ND(7,2)=S2*R1/GHT
      ND(8,2)=S1*R1/GHT
C
C     Here are the derivatives w.r.t. R
C
      ND(1,3)=S1*T1/GHT
      ND(2,3)=S2*T1/GHT
      ND(3,3)=S2*T2/GHT
      ND(4,3)=S1*T2/GHT
      ND(5,3)=-S1*T1/GHT
      ND(6,3)=-S2*T1/GHT
      ND(7,3)=-S2*T2/GHT
      ND(8,3)=-S1*T2/GHT
C
C*******************************************************************
C
```

```
C     Compute the AM(I,J) terms.
C     ND*(J)  J is the dimension   d/ds: J=1, d/dT: J=2, d/dR: J=3
C
      DO J=1,3
       NDX(J)=ZERO
       NDY(J)=ZERO
       NDZ(J)=ZERO
C
       DO I=1,8
C
        NDX(J)=NDX(J)+ND(I,J)*X(I)
        NDY(J)=NDY(J)+ND(I,J)*Y(I)
        NDZ(J)=NDZ(J)+ND(I,J)*Z(I)
C
       ENDDO ! I
      ENDDO  ! J
C
      AM(1,1)=NDY(2)*NDZ(3)-NDZ(2)*NDY(3)
      AM(2,1)=NDZ(2)*NDX(3)-NDX(2)*NDZ(3)
      AM(3,1)=NDX(2)*NDY(3)-NDY(2)*NDX(3)
      AM(1,2)=NDY(3)*NDZ(1)-NDZ(3)*NDY(1)
      AM(2,2)=NDZ(3)*NDX(1)-NDX(3)*NDZ(1)
      AM(3,2)=NDX(3)*NDY(1)-NDY(3)*NDX(1)
      AM(1,3)=NDY(1)*NDZ(2)-NDZ(1)*NDY(2)
      AM(2,3)=NDZ(1)*NDX(2)-NDX(1)*NDZ(2)
      AM(3,3)=NDX(1)*NDY(2)-NDY(1)*NDX(2)
C
C     Compute the determinant of matrix A
C
      DETA=AM(1,1)*NDX(1)+AM(2,1)*NDY(1)+AM(3,1)*NDZ(1)
C
      IF (DETA.EQ.ZERO) THEN
       WRITE(*,'(A)') ' ERROR: DETERMINANT A IS ZERO'
       STOP
      ENDIF
C
C*********************************************************************
C
C     Compute the derivatives of the interpolatory terms w.r.t. x,y,z
C
      DO I=1,8
       DX(I)=ZERO
       DY(I)=ZERO
       DZ(I)=ZERO
C
       DO K=1,3
C
        DX(I)=DX(I)+AM(1,K)*ND(I,K)/DETA
        DY(I)=DY(I)+AM(2,K)*ND(I,K)/DETA
        DZ(I)=DZ(I)+AM(3,K)*ND(I,K)/DETA
C
```

```
        ENDDO
       ENDDO
C
C*************************************************************************
C
C     Compute the terms that must be integrated over the surface.
C
       FF=ONE
C
       DO J=1,8
        SM(J,1)=SM(J,1)+FF*N(J)*
     $  (H_INC_Y*NORMAL(3)-H_INC_Z*NORMAL(2))*JACOB*WEIGHT
        SM(J+8,1)=SM(J+8,1)+FF*N(J)*
     $  (H_INC_Z*NORMAL(1)-H_INC_X*NORMAL(3))*JACOB*WEIGHT
        SM(J+16,1)=SM(J+16,1)+FF*N(J)*
     $  (H_INC_X*NORMAL(2)-H_INC_Y*NORMAL(1))*JACOB*WEIGHT
        SM(J+24,1)=ZERO
       ENDDO      ! J
C
C*************************************************************************
C
        ENDDO          ! B
       ENDDO           ! A
C
C*************************************************************************
C
C     Add the SM matrix terms to the the MATB matrix.
C
       DO L=1,4
        DO J=1,8
C
          ROW=BRICK_NODE(BRICK,J)-DIRICHLET(BRICK_NODE(BRICK,J),2)
     $        +(L-1)*FREE_NODE
C
          IF (DIRICHLET(BRICK_NODE(BRICK,J),1).NE.1) THEN
C            Not Dirichlet boundary conditions.
C
           MATB(ROW)=MATB(ROW)+SM((L-1)*8+J,1)
C
          ENDIF        ! ROW
         ENDDO         ! J
        ENDDO          ! L
C
C*************************************************************************
C
       ENDDO           ! SURFACE
C
C*************************************************************************
C
C     Solve the set of equations defined by matrix MATA
C      with solution returned in MATB
```

```
C     Call ZGECO to perform Gaussian elimination with
C     complex variables on the matrix
C
      WRITE(*,'(A)') ' Solving the system of equations'
C
      CALL ZGECO(MATA,ARRAY_SIZE,FREE_NODE*4,WORKI,RCOND,COM)
C
      WRITE(*,*) ' RCOND IS ',RCOND
C
      CALL ZGESL(MATA,ARRAY_SIZE,FREE_NODE*4,WORKI,MATB,0)
C
C     Regenerate the vector of unknowns including the Dirichlet
C      variables.
C
      DO J=1,4
       DO I=1,NUM_NODE
        NODE=(J-1)*NUM_NODE+I
        IF (DIRICHLET(I,1).NE.1) THEN
         COM(NODE)=MATB((J-1)*FREE_NODE+I-DIRICHLET(I,2))
        ELSE
         COM(NODE)=DCMPLX(ZERO,ZERO)    ! Dirichlet is zero for this
        ENDIF                           !  problem
       ENDDO
      ENDDO
C
      WRITE(*,'(A)') ' Printing out the results.'
C
C     Write the node potential information to the output file.
C
      WRITE(10,53) FREQUENCY(FREQ)
53    FORMAT(1X,'The following potentials were solved for an operating',
     $       /,1X,' frequency of ',1PE14.7,' Hz.',/)
      WRITE(10,54) RCOND
54    FORMAT(2X,'The condition number of the matrix is ',1PE14.7,'.')
      WRITE(10,55)
55    FORMAT(1X,'Node',5X,'Scalar potential',38X,'Vector potential',/,
     $              45X,'X',26X,'Y',26X,'Z',/)
C
      DO I=1,NUM_NODE
       WRITE(10,60) I,COM(3*NUM_NODE+I),
     $              (COM((J-1)*NUM_NODE+I),J=1,3)
60     FORMAT(1X,I3,4(2X,1PE14.7,1X,1PE14.7))
      ENDDO
C
C*****************************************************************
C
      ENDDO  ! FREQ
C
C*****************************************************************
C
C   Close any opened data file
```

```
C
      CLOSE(UNIT=10)
C
      STOP
      END
```

FIELDING.DORS

```
C*********************************************************************
      PROGRAM FIELDING
C
C           This program compute the magnetic and electric fields
C     from vector magnetic and scalar electric potentials.
C     The function depends upon the information read in.  It computes
C     the curl and divergence using the interpolatory functions
C     generated from the node data of the particular problem being
C     solved.
C           The information of the structure geometry is read
C      in from an external data file.  The default file name is
C      'BRICK.GEO'.
C           The information of the node potentials is read in from
C     an external data file.  The default file name is 'BRICK.RES'.
C
C           The format of the input .GEO file is as follows:
C
C           HEADER INFORMATION  PRECEDES THE DASHED LINE
C'-------------------------------------------------------------'
C     'H_INC_X','H_INC_Y','H_INC_Z'      FORMAT(3(2(1X,1PE12.4)))
C     'NUM_FREQ'                         FORMAT(1X,I8)
C     'FREQUENCY'                        FORMAT(1X,1PE12.4)
C
C     'NUM_NODE','NUM_BRICK','NUM_SUR'   FORMAT(1X,3I8)
C     'BRICK','NODE(1)','(2)','(3)','(4)','(5)','(6)','(7)','(8)',
C           FORMAT(1X,9I8)
C
C     'NODE','XX','YY','ZZ'              FORMAT(1X,I8,3(PE12.4))
C
C           The nodes are assigned to the bricks in a particular
C      fashion.  1:(-1,-1,1),2:(1,-1,1),3(1,1,1),4:(-1,1,1),
C      5:(-1,-1,-1),6:(1,-1,-1),7(1,1,-1),8:(-1,1,-1)
C
C           The nodes can be numbered in any order as long as the
C      correct coordinates are assigned.
C
C           The format of the input .RES file is as follows:
C     Note:  The .RES file must correspond to the .GEO file that is
C           used.  It automatically accounts for solution at all the
C           frequencies.
C
C     There are seven lines preceding each data set and they are
C     ignored.
C
C     'NODE','SCALAR','VECTORX','VECTORY','VECTORZ'
C         FORMAT(1X,I3,4(2X,1PE14.7,1X,1PE14.7))
C
C************** Property of D. Ladd ***********************************
```

```
C
      IMPLICIT NONE
C
C   DECLARATIONS:
C   -------------
C
      PARAMETER ORDER=1000
      PARAMETER REGION=1000
      PARAMETER FACES=1000
C
      COMPLEX*16  MATB(ORDER*4)    ! B vector
      COMPLEX*16  HX,HY,HZ         ! resultant magnetic field
      COMPLEX*16  EX,EY,EZ         ! resultant electric field
      COMPLEX*16  AX(8),AY(8),AZ(8)
      COMPLEX*16  PHI(8)
      COMPLEX*16  JC               ! defined imaginary constant
C
      REAL*8  XPOINT,YPOINT,ZPOINT
      REAL*8  XX(ORDER)            ! X coordinates of each node
      REAL*8  YY(ORDER)            ! Y coordinates of each node
      REAL*8  ZZ(ORDER)            ! Z coordinates of each node
      REAL*8  X(8),Y(8),Z(8)       ! local coordinates of the nodes
                                   !  in a brick being used to compute
                                   !  interpolatory functions
      REAL*8  S,T,R                ! natural coordinate system points
                                   !  transformed from X,Y,Z
      REAL*8  S1,S2,T1,T2,R1,R2    ! functions of S,T,R
      REAL*8  N(8)                 ! interpolatory functions for a
                                   !  brick in S,T,R units
      REAL*8  ND(8,3)              ! derivatives of the interpolatory
                                   !  functions w.r.t S.T,R
      REAL*8  NDX(3),NDY(3),NDZ(3)! summation of derivatives times
                                   !  X,Y,Z
      REAL*8  AM(3,3)              ! A matrix terms
      REAL*8  DETA                 ! determinant of the A matrix
      REAL*8  DX(8),DY(8),DZ(8)    ! the derivatives of the inter-
                                   !  polatory functions w.r.t. X,Y,Z
      REAL*8  FREQUENCY(10),OMEGA  ! the frequencies solved for
      REAL*8  ZERO,ONE,GHT         ! parameter definitions
      REAL*8  ENOT,MNOT            ! free-space values of permittivity
                                   !  and permeability
      REAL*8  PI                   ! defined constant
      REAL*8  DUMR                 ! dummy variable
C
      INTEGER  NUM_NODE            ! the total number of nodes in the
                                   !  region being investigated
      INTEGER  NUM_BRICK           ! the total number of bricks or sub-
                                   !  regions in the region being
                                   !  investigated
      INTEGER  NUM_FREQ            ! the total number of frequencies
                                   !  for which the problem must be
```

```
                                    !  solved
      INTEGER  NODE,COMB            ! a node counter
      INTEGER  BRICK                ! a brick element counter
      INTEGER  FREQ                 ! a frequency counter
      INTEGER  BRICK_NODE(REGION,8)! array containing 8 nodes of each
                                    !  brick element
      INTEGER  A,B,C                ! counters used in Gaussian
                                    !  quadrature
      INTEGER  I,J,K,L,M            ! counters
      INTEGER  DUMI                 ! dummy variable
C
      CHARACTER*50 FILE             ! input/output file names
      CHARACTER*50 FILE_GEO         !
      CHARACTER*1  DUMMY            ! dummy variable
      CHARACTER*1  DELIMIT          ! variable used to find the
                                    !  delimiter which indicates the
                                    !  start of the data
C
C*********************************************************************
C
C      START
C
      ZERO=0.0D0
      ONE=1.0D0
      GHT=8.0D0
C
      PI=DACOS(-ONE)
C
C   Define the permittivity  and permeability of free space
C
      ENOT=2.7816152D-11/PI
      MNOT=4.D-7*PI
C
      REL_MU=ONE
C
      JC=DCMPLX(ZERO,ONE)
C
C*********************************************************************
C
C   Open the necessary input/output files.
C
C   Get data from user defined input file.
C    Default input file name is BRICK.GEO.
C
      FILE=' '
      WRITE(*,'(A)') ' What is the input GEOMETRY file name?'
      WRITE(*,'(A)') ' The default is "BRICK.GEO"'
      READ(*,'(15A)') FILE
      IF(FILE.EQ.' ') THEN
       FILE='BRICK.GEO'
      ENDIF
```

```
C
      FILE_GEO=FILE
      OPEN(UNIT=20,FILE=FILE,STATUS='OLD')
C
C   The output data goes to a user defined file.
C    Default output file name is BRICK.RES.
C
      FILE=' '
      WRITE(*,'(A)') ' What is the input RESULT file name?'
      WRITE(*,'(A)') ' The default is "BRICK.RES"'
      READ(*,'(15A)') FILE
C
      IF(FILE.EQ.' ') THEN
       FILE='BRICK.RES'
      ENDIF
C
      OPEN(UNIT=10,FILE=FILE,STATUS='OLD')
C
      OPEN(UNIT=30,FILE='BRITH.CUR',STATUS='NEW')
      OPEN(UNIT=40,FILE='BRITE.CUR',STATUS='NEW')
C
C**********************************************************************
C
      WRITE(*,'(A)') ' Reading the data in now.'
C
C   Search for the '------------' delimit string which
C   indicates that the data is about to begin.  Ignore
C   anything before this.
C
      DELIMIT=' '
C
      DO WHILE (DELIMIT.NE.'-')
       READ(20,5) DELIMIT
5      FORMAT(1X,1A)
C
      ENDDO
C
C   Read the necessary data from the input file.
C
C     Get the incident magnetic field.
C
      READ(20,*) DUMR
C
C     This problem can be solved for different frequencies.
C
      READ(20,*) NUM_FREQ
      DO I=1,NUM_FREQ
       READ(20,*) FREQUENCY(I)
      ENDDO
C
      READ(20,*) NUM_NODE,NUM_BRICK
```

```
10    FORMAT(1X,3I8)
C
C   Get the node assignments for each brick.
C
      WRITE(*,'(A)') ' Getting brick information'
      DO I=1,NUM_BRICK
       READ(20,*) BRICK,(BRICK_NODE(BRICK,J),J=1,8)
20     FORMAT(1X,9I8,2(PE12.4))
      ENDDO       ! I
C
C   Read the coordinates and potentials of each node.
C
      WRITE(*,'(A)') ' Getting node information'
      DO I=1,NUM_NODE
       READ(20,*) NODE,XX(NODE),YY(NODE),ZZ(NODE)
30     FORMAT(1X,I8,5(PE12.4),I2)
      ENDDO       ! I
C
      CLOSE(UNIT=20)
C
C*********************************************************************
C
      WRITE(*,'(A)') ' Getting the node potential results'
C
      WRITE(30,55) FILE_GEO
      WRITE(40,55) FILE_GEO
55    FORMAT(1X,' These results were generated from the file ',A)
      DO FREQ=1,NUM_FREQ
       OMEGA=2.0D0*PI*FREQUENCY(FREQ)
C
C   First strip off the text at the top of the file
C
      DO I=1,7
       READ(10,5) DUMMY
      ENDDO
C
       DO K=1,NUM_NODE
        READ(10,60) I,MATB(3*NUM_NODE+I),
     $              (MATB((J-1)*NUM_NODE+I),J=1,3)
60      FORMAT(1X,I3,4(2X,PE14.7,1X,PE14.7))
       ENDDO
C
C*********************************************************************
C
      WRITE(*,'(A)') ' Printing out the results.'
C
      WRITE(30,35) FREQUENCY(FREQ)
      WRITE(40,35) FREQUENCY(FREQ)
35    FORMAT(//,2X,'The following potentials were solved for an',
     $       /,2X,'operating frequency of ',PE14.7,' Hz.',/)
      WRITE(30,40)
```

```
      WRITE(40,45)
40    FORMAT(3X,'BRICK',5X,'X',11X,'Y',11X,'Z',11X,
     $             'HX',13X,'HY',13X,'HZ',13X,'|H|'/)
45    FORMAT(3X,'BRICK',5X,'X',11X,'Y',11X,'Z',11X,
     $             'EX',13X,'EY',13X,'EZ',13X,'|E|'/)
C
C*****************************************************************************
C
      DO BRICK=1,NUM_BRICK
C
C*****************************************************************************
C
C      Transfer the global node data to local node data.
      DO NODE=1,8
       X(NODE)=XX(BRICK_NODE(BRICK,NODE))
       Y(NODE)=YY(BRICK_NODE(BRICK,NODE))
       Z(NODE)=ZZ(BRICK_NODE(BRICK,NODE))
       AX(NODE)=MATB(BRICK_NODE(BRICK,NODE))
       AY(NODE)=MATB(NUM_NODE+BRICK_NODE(BRICK,NODE))
       AZ(NODE)=MATB(2*NUM_NODE+BRICK_NODE(BRICK,NODE))
       PHI(NODE)=MATB(3*NUM_NODE+BRICK_NODE(BRICK,NODE))
      ENDDO
C
C*****************************************************************************
C
      S=ZERO
      T=ZERO
      R=ZERO
C
      HX=ZERO
      HY=ZERO
      HZ=ZERO
C
      EX=ZERO
      EY=ZERO
      EZ=ZERO
C
C*****************************************************************************
C
C      Generate the interpolatory functions.
C
      S1=ONE-S
      S2=ONE+S
      T1=ONE-T
      T2=ONE+T
      R1=ONE-R
      R2=ONE+R
C
      N(1)=S1*T1*R2/GHT
      N(2)=S2*T1*R2/GHT
      N(3)=S2*T2*R2/GHT
```

```
      N(4)=S1*T2*R2/GHT
      N(5)=S1*T1*R1/GHT
      N(6)=S2*T1*R1/GHT
      N(7)=S2*T2*R1/GHT
      N(8)=S1*T2*R1/GHT
C
C      Calculate the derivative terms of the interpolative functions
C      with respect to S, T, R.
C      ND(I,D)  D is the dimension:  d/dS: J=1, d/dT: J=2, d/dR: J=3
C
C      Here are the derivatives w.r.t. S
C
      ND(1,1)=-T1*R2/GHT
      ND(2,1)=T1*R2/GHT
      ND(3,1)=T2*R2/GHT
      ND(4,1)=-T2*R2/GHT
      ND(5,1)=-T1*R1/GHT
      ND(6,1)=T1*R1/GHT
      ND(7,1)=T2*R1/GHT
      ND(8,1)=-T2*R1/GHT
C
C      Here are the derivatives w.r.t. T.
C
      ND(1,2)=-S1*R2/GHT
      ND(2,2)=-S2*R2/GHT
      ND(3,2)=S2*R2/GHT
      ND(4,2)=S1*R2/GHT
      ND(5,2)=-S1*R1/GHT
      ND(6,2)=-S2*R1/GHT
      ND(7,2)=S2*R1/GHT
      ND(8,2)=S1*R1/GHT
C
C      Here are the derivatives w.r.t. R
C
      ND(1,3)=S1*T1/GHT
      ND(2,3)=S2*T1/GHT
      ND(3,3)=S2*T2/GHT
      ND(4,3)=S1*T2/GHT
      ND(5,3)=-S1*T1/GHT
      ND(6,3)=-S2*T1/GHT
      ND(7,3)=-S2*T2/GHT
      ND(8,3)=-S1*T2/GHT
C
C**********************************************************************
C
C      Compute the AM(I,J) terms.
C      ND*(J)  J is the dimension   d/ds: J=1, d/dT: J=2, d/dR: J=3
C
      DO J=1,3
       NDX(J)=ZERO
       NDY(J)=ZERO
```

```
        NDZ(J)=ZERO
C
        DO I=1,8
         NDX(J)=NDX(J)+ND(I,J)*X(I)
         NDY(J)=NDY(J)+ND(I,J)*Y(I)
         NDZ(J)=NDZ(J)+ND(I,J)*Z(I)
C
        ENDDO ! I
       ENDDO  ! J
C
       AM(1,1)=NDY(2)*NDZ(3)-NDZ(2)*NDY(3)
       AM(2,1)=NDZ(2)*NDX(3)-NDX(2)*NDZ(3)
       AM(3,1)=NDX(2)*NDY(3)-NDY(2)*NDX(3)
       AM(1,2)=NDY(3)*NDZ(1)-NDZ(3)*NDY(1)
       AM(2,2)=NDZ(3)*NDX(1)-NDX(3)*NDZ(1)
       AM(3,2)=NDX(3)*NDY(1)-NDY(3)*NDX(1)
       AM(1,3)=NDY(1)*NDZ(2)-NDZ(1)*NDY(2)
       AM(2,3)=NDZ(1)*NDX(2)-NDX(1)*NDZ(2)
       AM(3,3)=NDX(1)*NDY(2)-NDY(1)*NDX(2)
C
C       Compute the determinant of matrix A
C
       DETA=AM(1,1)*NDX(1)+AM(2,1)*NDY(1)+AM(3,1)*NDZ(1)
C
       IF (DETA.EQ.ZERO) THEN
        WRITE(*,'(A)') ' ERROR: DETERMINANT A IS ZERO'
        STOP
       ENDIF
C
C*********************************************************************
C
C       Compute the derivatives of the interpolatory terms w.r.t. x,y,z
C
       XPOINT=ZERO
       YPOINT=ZERO
       ZPOINT=ZERO
C
       DO I=1,8
        DX(I)=ZERO
        DY(I)=ZERO
        DZ(I)=ZERO
        XPOINT=XPOINT+N(I)*X(I)
        YPOINT=YPOINT+N(I)*Y(I)
        ZPOINT=ZPOINT+N(I)*Z(I)
C
        DO K=1,3
C
         DX(I)=DX(I)+AM(1,K)*ND(I,K)/DETA
         DY(I)=DY(I)+AM(2,K)*ND(I,K)/DETA
         DZ(I)=DZ(I)+AM(3,K)*ND(I,K)/DETA
        ENDDO
```

```
C
C     COMPUTE THE CURL OF vector potential A w.r.t x,y,z
C
       HX=HX+(DY(I)*AZ(I)-DZ(I)*AY(I))/(REL_MU*MNOT)
       HY=HY+(DZ(I)*AX(I)-DX(I)*AZ(I))/(REL_MU*MNOT)
       HZ=HZ+(DX(I)*AY(I)-DY(I)*AX(I))/(REL_MU*MNOT)
C
       EX=EX-JC*OMEGA*N(I)*AX(I)-DX(I)*PHI(I)
       EY=EY-JC*OMEGA*N(I)*AY(I)-DY(I)*PHI(I)
       EZ=EZ-JC*OMEGA*N(I)*AZ(I)-DZ(I)*PHI(I)
C
      ENDDO
C
      WRITE(30,70) BRICK,XPOINT,YPOINT,ZPOINT,
     $    CDABS(HX),CDABS(HY),CDABS(HZ),
     $    DSQRT(CDABS(HX)**2+CDABS(HY)**2+CDABS(HZ)**2)
      WRITE(40,70) BRICK,XPOINT,YPOINT,ZPOINT,
     $    CDABS(EX),CDABS(EY),CDABS(EZ),
     $    DSQRT(CDABS(EX)**2+CDABS(EY)**2+CDABS(EZ)**2)
70    FORMAT(1X,I5,3(1X,PE11.4),4(1X,PE14.7))
C
C*************************************************************************
C
      ENDDO  ! BRICK
      ENDDO  ! FREQ
C
C*************************************************************************
C
C   Close any opened data file
C
      CLOSE(UNIT=10)
      CLOSE(UNIT=30)
      CLOSE(UNIT=40)
C
      STOP
      END
```

Sample Input File for DORS

```
 For use with DORS.FOR
 This is an example file for the program DORS.  It illustrates
 the format of the input data file .GEO.
 The data presented here is for a single element with dimensions
 1m cubed.

 Note:  everything before the dashed line is ignored by the
        program and is ideal for comments.

 TEST WITH FRUSTRATION BRICK
-------------------------------------------
0.0000E+00 0.0000E+00  0.0000E+00 0.0000E+00  1.0000E+00 0.0000E+00
       1
  1.00E+01
    8    1    4
    1    1    2    3    4    5    6    7    8  1.00E+00  1.00E+07
    1 -5.00E-01 -5.00E-01  5.00E-01  0.00E+00  0.00E+00 0
    2  5.00E-01 -5.00E-01  5.00E-01  0.00E+00  0.00E+00 0
    3  5.00E-01  5.00E-01  5.00E-01  0.00E+00  0.00E+00 0
    4 -5.00E-01  5.00E-01  5.00E-01  0.00E+00  0.00E+00 0
    5 -5.00E-01 -5.00E-01 -5.00E-01  0.00E+00  0.00E+00 0
    6  5.00E-01 -5.00E-01 -5.00E-01  0.00E+00  0.00E+00 0
    7  5.00E-01  5.00E-01 -5.00E-01  0.00E+00  0.00E+00 0
    8 -5.00E-01  5.00E-01 -5.00E-01  0.00E+00  0.00E+00 0
    1    1    1
    2    1    2
    3    1    5
    4    1    6
```

Bibliography

[1] H.W. Ott, *Noise Reduction Techniques in Electronic Systems*, John Wiley and Sons, New York 1976.

[2] C.J. Carpenter, "Comparison of alternative formulations of 3-dimensional magnetic field and eddy current problems at power frequencies", *IEE Proceedings A*, Vol. 124, 11, p. 1026.

[3] M.V.K Chari, J. D'Angelo, and M.A. Palmo, "A finite element analysis of the eddy current diffusion", *IEEE Transactions*, PAS-107, (1984).

[4] M.V.K Chari, A. Konrad, M.A. Palmo, J. D'Angelo, "Three-dimensional vector potential analysis for machine field problems", *IEEE Transactions on Magnetics*, MAG-18, 2 (1982).

[5] D. Rodger, "Finite element method for calculating power frequency three dimensional electromagnetic field distributions", *IEE Proceedings A*, Vol. 130, 5 (1983) pp. 233-238.

[6] N.K. Deshmukh, and K.C. Mukherji "Finite element analysis of three dimensional eddy currents in attractive electromagnetic levitation", *IEE Proceedings A*, Vol. 134, 8 (1987) p. 651.

[7] P.J. Leonard, and D. Rodger, "Some aspects of two and three dimensional transient eddy current modelling using finite elements, and single-step, time-marching algorithms", *IEE Proc. A*, Vol. 135, 3 (1988).

[8] T.W. Preston, and A.B.J. Reece, "Solution of 3-dimensional eddy current problems. The $T-\Omega$ methods" *IEEE Transactions on Magnetics*, MAG-18, 2 (1982).

[9] S.J. Salon, and J.P. Peng, "Three dimensional eddy currents using a four component finite element formulation", *IEEE Trans. on Magnetics*, MAG-20, 5 (1984).

[10] P.P. Silvester, and R.L. Ferrari, *Finite Elements for Electrical Engineers*, Cambridge University Press, New York 1983.

[11] P.P. Silvester, and M.V.K Chari, *Finite Elements in Electric and Magnetic Field Problems*, John Wiley and Sons, New York 1980.

[12] Hartley Grandin Jr., *Fundamentals of the Finite Element Method*, Macmillan Publishing Co., New York, 1986.

[13] O.C. Zienkiewicz, *The Finite Element Method*, 3rd Ed., McGraw Hill Book Co., New York 1977.

[14] O.C. Zienkiewicz, and R.L. Taylor *The Finite Element Method: Volume 1, Basic Formulation and Linear Problems*, 4th Ed., McGraw Hill Book Co., Maidenhead, England 1989.

[15] K.S.H. Lee, Ed., *EMP Interaction 2-1: Principles, Techniques, and Reference Data*, Air Force Weapons Laboratory — TR-80-402, December 1980.

[16] H. Engels, *Numerical Quadrature And Cubature*, Academic Press, London 1980.

[17] M.K. Jain, *Numerical Solution Of Differential Equations*, 2nd Ed., Wiley Eastern Limited 1984.

[18] J. Crank, and P. Nicholson, "Practical method for numerical solution of partial differential equations of heat conduction type", *Proceedings of the Cambridge Philosophical Society*, Vol. 43 (1947), pp. 50-67.

[19] O.C. Zienkiewicz, W.L. Wood, and N.W. Hine, "A unified set of single step algorithms", *International Journal of the Numerical Methods in Engineering*, Vol. 20 (1984), pp. 1529-1552.

[20] J.J Dongarra, C.B. Moler, J.R. Bunch, and G.W Stewart, *LINPACK Users' Guide*, SIAM, U.S.A., 1979.

[21] David Kahaner, Cleve Moler, and Stephen Nash, *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, NJ 1989.

[22] Graham Carey, and J. Tinsley Oden, *Finite Elements: Volume 3, Computational Aspects*, Prentice Hall, Englewood Cliffs, NJ 1981.

[23] J.E. Key, "A computer program for the solution of large, sparse, unsymmetric matrices", *International Journal of the Numerical Methods in Engineering*, July 1973, pp. 497-509.

[24] P. Hood, "Frontal solution program for unsymmetric matrices", *International Journal of the Numerical Methods in Engineering*, Oct. 1976, pp. 379-399.

[25] Z. Zlatev, *Y12M Solution of Large and Sparse Systems of Linear Algebraic Equations*, Springer-Verlag, New York 1981 Vol. 121.

[26] R. Radulet, A. Timotin, A. Tugulea, "Introduction des paramètres transitoires dans l'étude des circuits électriques linéaires ayant des éléments non filiformes et avec pertes supplémentaires", *Rev. Roum. Sci. Techn.-Électrotechn. et Énerg.*, Vol 11, 4 (1966) pp. 565-639.

[27] Erwin Kreyszig, *Advanced Engineering Mathematics*, 5th Ed., John Wiley and Sons, New York 1983.

[28] Alan Jennings, *Matrix Computation for Engineers and Scientists*, John Wiley and Sons, New York 1977.

[29] Nasir Ahmed, and T. Natarajan, *Discrete-Time Signals and Systems*, Reston Publishing Co., Inc., Virginia 1983.

[30] M.C. Escher, *The Graphic Work of M.C. Escher*, Oldbourne Press, London 1961.