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.

SHIELDING EFFECTIVENESS OF ENCLOSURES PREDICTED VIA THE FINITE ELEMENT METHOD

by

Luc B. Gravelle

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 0-315-59997-9

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

I hereby declare that I am the sole author of this document. I authorize the University of Ottawa to lend this document to other institutions or individuals for the purpose of scholarly research.

Luc B. Gravelle

I further authorize the University of Ottawa to reproduce this document by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Luc B. Gravelle

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. G.I. Costache for his guidance and encouragement during the preparation of this thesis, thank you very much for always being available for your students when we needed you, it is exceptional.

I would also like to extend sincere thanks to Dr. R. Vaillancourt, from the Mathematics departement of the University of Ottawa, for his great help in locating a finite-difference algorithm appropriate to the analysis.

The financial support of the National Science and Engineering Research Council of Canada is gratefully acknowledged.

Abstract

This thesis presents the application of the numerical finite element method (FEM) in solving the electromagnetic shielding effectiveness of enclosures. The emphasis is placed on the solution of time-dependant varying fields, as for example an NEMP or a digital signal, directly in the time domain. Previous numerical prediction of shielding effectiveness on complicated structures have been analyzed with the FEM in the frequency domain whereas no study, to the author's knowledge, has been conducted in the time domain. EMP type varying fields impinging on enclosures are studied with this new CAD tool and compared to the analytical solution of a simple parallel plate shield. Following the validation of the technique for the one-dimensional case, the shielding effectiveness of a more general two dimensional structure is investigated and validated with the introduction of a second numerical technique, TLM. Included in this study are two user-friendly, general, interactive Fortran codes. One analyze the field penetration through rectangular enclosures (with apertures) of various EMP type plane waves (TE or TM) having a certain angle of incidence on enclosures. The other analyze the field radiation from enclosures (with apertures) due to currents in signal lines.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electromagnetic shielding of sensitive equipment is receiving a growing interest in the electromagnetic interference and compatibility (EMI/EMC) community as the density of electronic equipment in the working environment is continuously increasing. The effect of the overloaded environment is to reduce the reliability of devices improperly protected from potentially damaging ambient fields. To remedy this problem, federal agencies throughout the world have regulated the radiation levels of any electronic device to specified limits. In the United States, for example, the Federal Communication Commission (FCC) has listed these requirements in the "FCC Rules and Regulations" under the section "Part 15J" [40]. It is now required, by law, that the radiation levels of an equipment be lower than a specified level. Furthermore natural phenomena such as lightning and electrostatic discharges (ESD) are also very threatening to the performance of sensitive equipment as their associated rise and fall times generate wide frequency band electromagnetic waves. Along with these inherent passive EMI sources (passive in the sense of unprovoked) one more major catastrophic source to be included in the EMI shielding analysis is a possible high-altitude electromagnetic burst. The characteristic of these cataclysmic bursts lie in the fact that with a single nuclear war head explosion at a certain altitude in the atmosphere (typically around 400 km) the entire North-American continent could be

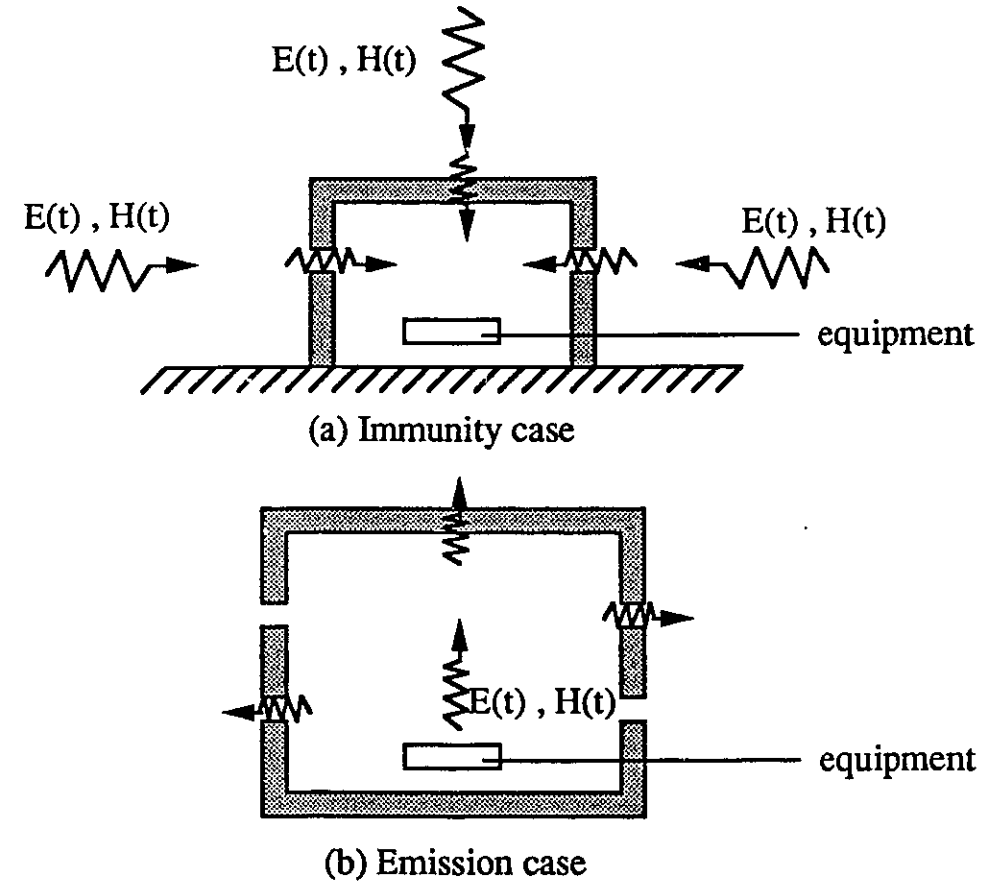


Figure 1.1: Problems to be analysed.

illuminated by this devastating high-altitude nuclear electromagnetic pulse (HNEMP) causing malfunction and even breakdown of improperly shielded sensitive electronic equipment.

The wide interest in assuring the reliability of equipment under normal to severe conditions has focused our attention on the need for an effective, inexpensive and easy to use tool intended to predict the shielding effectiveness of arbitrarily shaped enclosures. As it was implied above, there are two cases to study when refering to the shielding effectiveness of an enclosure. One type of study to include in the tool is the immunity analysis, in which the shield is intended to prevent ambient fields from perturbing the electronic device it contains (see figure 1.1 (a)). The other is the emission analysis, where the shield restrains energy, generated by the device, from affecting nearby equipment (see figure 1.1 (b)).

1.1 Goal

The object of this thesis is to bring forward a novel approach to predict the shielding effectiveness of enclosures. It consists of developing a computer aided design (CAD) tool via the numerical technique of finite element method (FEM). This particular numerical technique was chosen as it is one of the most flexible methods for solving problems as general as the ones presented here.

The CAD tool will have to be applicable to both the emission and the immunity problems as these are two very different situations. In the latter case, the impinging field will be assumed known at a certain distance from the enclosure; where as in the former, a current, flowing in a wire or on a microstrip's ground plane, will generate a magnetic field which will be assumed to be known very close to the conductor.

Furthermore, the tool shall perform the analysis in both the frequency and time domains. The added feature of a time domain analysis will reveal to be extremely important, especially in the study of EMP type fields such as lightning, ESD and NEMP and also in the study of the effect of digital signals on the radiated fields.

To achieve these objectives with assurance, we divide the task into four stages.

- Analytical solution of one-dimensional problem to validate the technique.
- Introduction of a second numerical technique.
- Comparison between two numerical techniques for more involved situations.
- Application of the CAD tool to complex problems.

In the first task, an analytical solution to a diffusion problem is derived for both the frequency and time domains. With the availability of this analytical formulation, the range of operation of the CAD tool is determined. To validate the prediction of the tool for a higher degree of complexity, a completely different numerical technique is introduced. The transmission-line matrix (TLM) method, the new technique, need also to be validated through the analytical problem to determine its operating range.

Once the confidence in solving one-dimensional diffusion problems is established, both techniques are applied to two-dimensional problems and the results are compared. The CAD tool, now validated, may be applied to complex situations that are more practical.

1.2 The Numerical Techniques

1.2.1 FEM

The method of finite elements originates from the variational method which may be traced back to the past century. This method was mainly applied, initially, by civil engineers to calculate structural stress and fluid dynamics. Although it was first attributed the name of "finite element method" by Clough in 1960 [1], it was extensively used in the late fifties.

The finite element method consists of characterizing the equations of a boundary-value problem through an appropriate functional which generally relates to the energy stored in the field (for electrical situations). Once this functional is determined, the region of interest is then discretized into finite elements in which the unknown field is approximated by suitable interpolation functions. Introducing these functions into the appropriate functional and taking the first variation with respect to the unknown, leads to a final system of equation that, in some cases, may be solved directly.

The finite element method (FEM) was first applied in electrical engineering in the middle of the 60's by Winslow [2] for the study of saturation effects in accelerator magnets. Since then, the application of the finite element method in electrical engineering has become a current topic, it is considered one of the fastest growing techniques in this field [43]. Its applications vary from eddy currents and skin effect problems [3], [4], [5] to the study of transformers [6] [7] , electrostatic [7], turbo-alternators [8], anisotropic magnetic problems [9] and much more. The flexibility of

this numerical technique is its driving force, the elements may be adapted to practically any boundary and interface geometry as they may be defined in various shapes and sizes.

1.2.2 TLM

The transmission-line matrix (TLM) method was first introduced by Johns and Beurle [33] in 1971. This method is a direct application of Huygen's principle of light [32] which explains a wavefront propagation as a combination of secondary wave sources on the wave front giving rise to a new front formed by the envelope of the secondary wavefronts. The TLM method consists of discretizing the region of interest into interconnecting transmission lines in which a travelling field would be scattered upon reaching a junction as there is impedance discontinuity. Thus each junction could then be seen as the secondary wave sources mentioned above.

The initial application of the TLM method was directed to the modeling of field propagation into a waveguide and the study of the scattered fields due to a waveguide discontinuity [33]. The method was then rapidly extended to analyse inhomogeneous [34] and lossy waveguides [10]. Then the interest was shifted to three-dimensional implementation of the technique [39], [11], [12]. Following this, some refinements on the two-dimensional formulation were brought forward such as non-uniform mesh [36], [13], and coarseness error elimination [14]. The applications of TLM have been reported in the analysis of fin lines [15], transients in transmission-lines [16], anisotropic media [17], dielectric loaded cavities [39], and many more. An excellent review paper on TLM, by Hoefer, can be found in [18]. The merits of TLM are also its flexibility and most definitely its versatility. The fact that the analysis is conducted in the time domain directly is also very interesting.

Chapter 2

One-Dimensional Test Enclosure

The evaluation of any type of measurement or prediction schemes is problematic; one has to be extremely careful when "conceiving" a test problem aimed at determining their proper range of operation. When evaluating a computer aided design (CAD) tool, an individual may choose among three options: either he(she) compares the results from the tool with measurements, another tool, or an analytical solution. The latter option, in our opinion, is advantageous when available, as it yields more information in less time at a very low cost and particularly because the analytical (closed-form) results are absolutely accurate.

When predicting the field penetration into an enclosure, a limiting factor arises from the diffusion process. In particular, one would ask the following question, will the tool be able to follow the exponential decay of the field? If so, up to what point? A novel approach is presented here which predicts the shielding effectiveness of enclosures; it consists of the application of two numerical techniques, namely the finite element method (FEM) and the transmission line matrix (TLM) method. To validate these CAD tools, an analytical solution is derived for both the frequency and time domains.

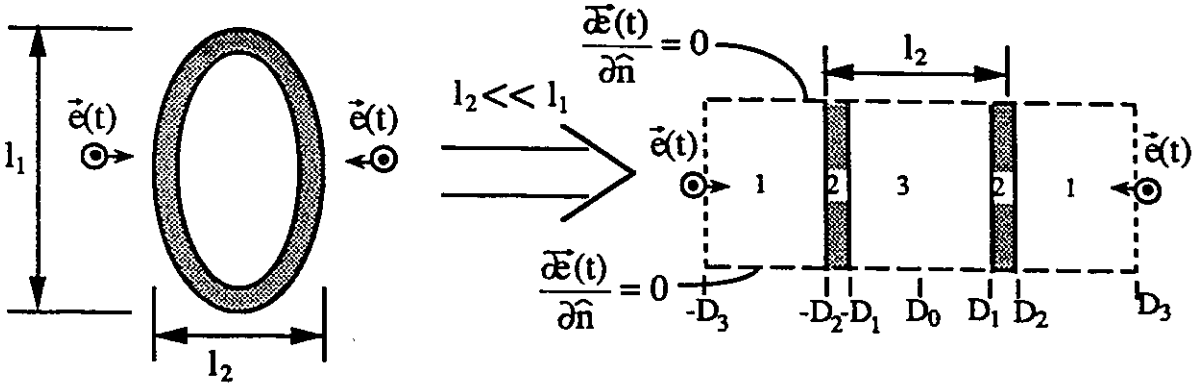


Figure 2.1: One-dimensional approximation.

2.1 Analytical Formulation

The analytical diffusion test problem may be regarded as a two-dimensional enclosure that extends much more in one direction than in the other thus making a one-dimensional analysis possible, see figure 2.1. The field is assumed to impinge on both sides of the enclosure and to be of equal magnitude and phase.

The full equation describing the field propagation through air and conducting materials is obtained directly from Maxwell's equations and given as

$$\frac{\partial^2}{\partial x^2}\vec{e}(x,t) - \mu\sigma\frac{\partial}{\partial t}\vec{e}(x,t) - \mu\epsilon\frac{\partial^2}{\partial t^2}\vec{e}(x,t) = 0 \tag{2.1}$$

for the one-dimensional problems. Now taking the Laplace transform of equation (2.1) leads to

$$\begin{aligned}\frac{\partial^2}{\partial x^2}\vec{E}(x,s) - s\mu\sigma\vec{E}(x,s) + \mu\sigma\vec{e}(x,t=0^-) - \\ s^2\mu\epsilon\vec{E}(x,s) + s\mu\epsilon\vec{e}(x,t=0^-) + \mu\epsilon\frac{\partial}{\partial t}\vec{e}(x,t=0^-) \;\; = \;\; 0\end{aligned} \tag{2.2}$$

but as the impinging field is initially zero, thus equation (2.2) becomes of the following form

$$\frac{\partial^2}{\partial x^2}\vec{E}(x,s) - s\mu\sigma\vec{E}(x,s) - s^2\mu\epsilon\vec{E}(x,s) = 0 \tag{2.3}$$

where μ, ϵ and σ are respectively the permeability, permittivity and conductivity of the region.

2.1.1 Frequency Domain

The wave equation in the frequency domain is written as (from equation (2.3))

$$\frac{\partial^2}{\partial x^2}\underline{\vec{E}}(x) + (\omega^2\mu\epsilon - j\omega\mu\sigma)\underline{\vec{E}}(x) = 0 \tag{2.4}$$

in which ω is a frequency variable. Referring to figure 2.1, the equation to be solved in region 1 and 3 (in air) is

$$\frac{\partial^2}{\partial x^2}\underline{\vec{E}}_{1/3}(x) + \omega^2\mu_0\epsilon_0\underline{\vec{E}}_{1/3}(x) = 0 \tag{2.5}$$

and in region 2 (in the conducting walls)

$$\frac{\partial^2}{\partial x^2}\underline{\vec{E}}_2(x) - j\omega\mu\sigma\underline{\vec{E}}_2(x) = 0 \tag{2.6}$$

where in the conducting walls the displacement current term ($\omega^2\mu\epsilon\underline{\vec{E}}(x)$) may be neglected. From equations (2.5) and (2.6), the general solution for the three different regions is given in the following

$$\underline{\vec{E}}_1(x) = \hat{z}(Ae^{jK_1x} + Be^{-jK_1x}) \tag{2.7}$$

$$\underline{\vec{E}}_2(x) = \hat{z}(Ce^{jK_2x} + De^{-jK_2x}) \tag{2.8}$$

$$\underline{\vec{E}}_3(x) = \hat{z}Fcosh(K_1x) \tag{2.9}$$

where $K_1 = \sqrt{\omega^2\mu_0\epsilon_0}$ and $K_2 = \sqrt{-j\omega\mu\sigma}$. The constants A through F are derived by imposing the boundary conditions on equations (2.7)-(2.9).
Boundary conditions:

1. $\underline{\vec{E}}_1(\pm D_3) = \hat{z}E_0$
2. $\frac{\partial}{\partial x}\underline{\vec{E}}_3(D_0) = 0$
3. Matching the electric and magnetic field at the interfaces,
   i.e. at $x = \pm D_1$ and $x = \pm D_2$.

Applying these boundary conditions, for positive x, to equations (2.7), (2.8) and (2.9) leads to the following global system of equations

$$\begin{pmatrix} e^{jK_1D_3} & e^{-jK_1D_3} & 0 & 0 & 0 \\ e^{jK_1D_2} & e^{-jK_1D_2} & -e^{jK_2D_2} & -e^{-jK_2D_2} & 0 \\ k_1e^{jK_1D_2} & -K_1e^{-jK_1D_2} & -\frac{K_2}{\mu_r}e^{jK_2D_2} & \frac{K_2}{\mu_r}e^{-jK_2D_2} & 0 \\ 0 & 0 & e^{jK_2D_1} & e^{-jK_2D_1} & \cosh(K_1D_1) \\ 0 & 0 & -\frac{jK_2}{\mu_r}e^{jK_2D_1} & \frac{jK_2}{\mu_r}e^{-jK_2D_1} & K_1\sinh(K_1D_1) \end{pmatrix} \times$$

$$\begin{pmatrix} A \\ B \\ C \\ D \\ F \end{pmatrix} = \begin{pmatrix} E_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.10}$$

This system of equation may be solved analytically or via a matrix reduction technique once the parameters are known.

2.1.2 Time Domain

To obtain the time domain analytical solution, the formulation is first conducted in Laplacian domain and transformed, in the last stage of analysis, back to the time domain via Heaviside's formulation [19]. To decrease the complexity of the problem, the impinging field will be assumed known on the outside wall of the enclosure, at $\pm D_2$ (see figure 2.1). The time signature of the incident field is modeled as

$$\vec{e}(t) = \hat{z}E_0\{e^{-\alpha t} - e^{-\beta t}\} \tag{2.11}$$

which may represent a nuclear electromagnetic pulse (NEMP), an electrostatic discharge (ESD) or lightning [20]. The mathematical model describing the wave propagation through air and conducting materials in the Laplace domain is given in (2.3). Thus, neglecting the displacement current in the conducting material, the equation to be solved in region 2 becomes

$$\frac{\partial^2}{\partial x^2}\vec{E}_2(x,s) - s\mu\sigma\vec{E}_2(x,s) = 0 \tag{2.12}$$

and in region 3 (in the air)

$$\frac{\partial^2}{\partial x^2}\vec{E}_3(x,s) - s^2\mu_0\epsilon_0\vec{E}_3(x,s) = 0 \tag{2.13}$$

where s is the Laplace variable. The general solution in each region is given as

$$\vec{E}_2(x,s) = \hat{z}[A\cos(\sqrt{-s\mu\sigma}\,x) + B\sin(\sqrt{-s\mu\sigma}\,x)] \tag{2.14}$$

$$\vec{E}_3(x,s) = \hat{z}C\cosh(\sqrt{s^2\mu_0\epsilon_0}\,x) \tag{2.15}$$

where A, B and C are constants to be determined. To obtain these constants we need to impose the boundary conditions on equations (2.14) and (2.15).
Boundary conditions:

1. $\vec{E}_2(\pm D_2, s) = \mathcal{L}\{\hat{z}E_0(e^{-\alpha t} - e^{-\beta t})\} = \hat{z}\frac{E_0(\beta-\alpha)}{(s+\alpha)(s+\beta)} = \hat{z}L(s)$
2. $\frac{\partial}{\partial x}\vec{E}_3(D_0, s) = 0$
3. $\vec{E}_2(\pm D_1, s) = \vec{E}_3(\pm D_1, s)$
4. $\vec{H}_2(\pm D_1, s) = \vec{H}_3(\pm D_1, s)$.

Applying the boundary conditions 1-4, for positive x, to equations (2.14) and (2.15), the constants A, B and C may be derived by solving the following global system of equations:

$$\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ \cos\gamma & \sin\gamma & -\cosh\phi \\ \sin\gamma & -\cos\gamma & \delta\sinh\phi \end{pmatrix}\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} L(s) \\ 0 \\ 0 \end{pmatrix} \tag{2.16}$$

where

$$\delta = \left(\eta_0\sqrt{\frac{\sigma}{-s\mu}}\right)^{-1}\ ;\ \theta = \sqrt{-s\mu\sigma}\,D_2\ ;\ \gamma = \sqrt{-s\mu\sigma}\,D_1\ ;\ \phi = \sqrt{s^2\mu_0\epsilon_0}\,D_1$$

and

$$\eta_0 = \sqrt{\frac{\epsilon_0}{\mu_0}}.$$

Solving (2.16) for A, B, and C leads to:

$$A = L(s)(\cosh\phi\cos\gamma - \delta\sinh\phi\sin\gamma)/R(s) \tag{2.17}$$

$$B = L(s)(\cosh\phi\sin\gamma + \delta\sinh\phi\cos\gamma)/R(s) \tag{2.18}$$

$$C = L(s)/R(s) \tag{2.19}$$

where $R(s)$ is the determinant of the 3x3 matrix in (2.16), given as :

$$R(s) = [\cosh\phi\cos(\gamma-\theta) - \delta\sinh\phi\sin(\gamma-\theta)]$$

and

$$L(s) = \mathcal{L}\{E_0(e^{-\alpha t} - e^{-\beta t})\} = \frac{E_0(\beta-\alpha)}{(s+\alpha)(s+\beta)}.$$

To obtain the time domain counterpart of equations (2.14) and (2.15) one may use Heaviside's formulation as stated below, if

$$G(s) = \frac{P(s)}{Q(s)} \tag{2.20}$$

then

$$g(t) = \sum_{k=1}^{\infty} \frac{P(s_k)}{\dot{Q}(s_k)} e^{s_k t} \tag{2.21}$$

where s_k are the poles of $Q(s)$ and $\dot{Q}$ is the derivative of Q with respect to the variable s. As the derivation of $\vec{e}(t)$, the time domain electric field, is quite involved, it will not be written here (the reader is referred to Appendix A for more details).

Chapter 3

Finite Element Method

3.1 Introduction

The finite element method is a numerical technique based on the variational method [21]. It consists of describing the mathematical model (i.e. the equation defining the particular situation), along with the boundary conditions, with an appropriate functional

$$\mathcal{F}(\phi) = \int_{\Omega} L\phi d\Omega \tag{3.1}$$

where ϕ is the unknown quantity, L, the operator and Ω is the region of interest. The functional, which frequently coincides with the energy stored in the field for electrical problems, as given in (3.1), is an integral equation which will be made stationary by taking its first variation with respect to the unknown

$$\frac{\partial \mathcal{F}(\phi)}{\partial \phi} = 0 \tag{3.2}$$

thus yielding the solution to the unknown ϕ (approximated by an interpolation function in the region Ω). Now to better approximate the field variations, the region is divided into subregions, such as triangles, of various shapes and sizes as shown in figure 3.1. In each of these subregions Ω_i, the unknown quantity is represented by suitable interpolation functions. Thus the functional becomes a summation over the

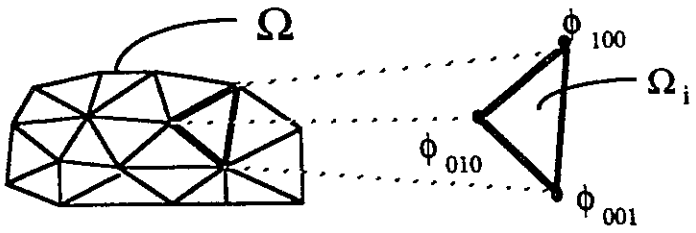


Figure 3.1: Finite element subregions.

total number of subregions of the integrals

$$\mathcal{F}(\phi) = \sum_{i=1}^{NSUB} \int_{\Omega_i} L\hat{\phi} d\Omega_i \tag{3.3}$$

where NSUB is the total number of subregions and

$$\hat{\phi} = \underline{\phi}_j \underline{\zeta}(x, y)$$

in which $\underline{\phi}_j$ is the unknown nodal vector $[\underline{\phi}_j = (\phi_{100}\ \phi_{010}\ \phi_{001})]$ of the corresponding subregion and $\underline{\zeta}(x, y)$, the interpolation function. Equation (3.2) is then satisfied by taking the first variation of (3.3) with respect to all unknown nodal field quantities $(\phi_1, \phi_2, ...\phi_m)$ which in turn produces a system of equations

$$S\underline{\phi} = \underline{b} \tag{3.4}$$

which when linear may be solved via a matrix reduction technique.

3.2 Mathematical Formulation

The idea behind the finite element method is to find an energy related functional describing the particular situation and make it stationary by taking its first variation with respect to the unknown. The functional associated with the problem of interest is determined by both the mathematical model (i.e. the equation describing the field propagation through conducting material and air), as given in the following

$$\nabla^2 \vec{g}(t) - \mu\sigma \frac{\partial}{\partial t} \vec{g}(t) - \mu\epsilon \frac{\partial^2}{\partial t^2} \vec{g}(t) = 0 \tag{3.5}$$

and the boundary conditions such as inhomogeneous Neumann boundaries. In equation (3.5), $\vec{g}(t)$ represents either the electric or the magnetic field, or even the magnetic vector potential which will be derived later in section 5.3. Equation (3.5) may be rewritten in a more general form as

$$\nabla^2 \vec{g}(t) - k\vec{g}(t) = 0 \tag{3.6}$$

in which k is a time differentiation operator ($k \equiv \mu\sigma\frac{\partial}{\partial t} + \mu\epsilon\frac{\partial^2}{\partial t^2}$).

The functional associated with the mathematical model given in (3.6), is [22]

$$\mathcal{F}(\vec{g}(t)) = \int_\Omega \vec{g}(t) \cdot \nabla^2 \vec{g}(t) d\Omega - k \int_\Omega \vec{g}(t) \cdot \vec{g}(t) d\Omega \tag{3.7}$$

where the integrals cover the total region of interest, Ω. To include Neumann boundary conditions, the first integral in the functional may be modified by Green's first formula [23]

$$\int_\Omega \left[\vec{f} \cdot \nabla^2 \vec{h} + \nabla \vec{f} \cdot \nabla \vec{h}\right] d\Omega = \int_S \vec{f} \cdot \frac{\partial \vec{h}}{\partial \hat{n}} dS \tag{3.8}$$

where the right hand side integral is performed on the surface of the region with the normal pointing outward. The use of Green's identity will also reduce the Laplacian operator, ∇^2 (a second order derivative), to a vector product of two gradients, first order derivatives, making the analysis more stable. Equation (3.7) then becomes

$$\mathcal{F}(\vec{g}(t)) = -\int_\Omega \nabla \vec{g}(t) \cdot \nabla \vec{g}(t) d\Omega - k \int_\Omega \vec{g}(t) \cdot \vec{g}(t) d\Omega + 2 \int_S \vec{g}(t) \cdot \frac{\partial \vec{g}(t)}{\partial \hat{n}} dS. \tag{3.9}$$

Now to verify that (3.9) is an appropriate functional associated to equation (3.6), let's take the first variation of the functional with respect to the unknown, $\vec{g}(t)$, and set it equal to zero as it is a sufficient condition for the solution to be stationary [24].

$$\Delta\mathcal{F}(\vec{g}(t)) = 0. \tag{3.10}$$

The first variation of (3.9) is

$$\begin{aligned} \Delta\mathcal{F}(\vec{g}(t)) \;=\; & -\int_\Omega 2\nabla \vec{g}(t) \cdot \Delta(\nabla \vec{g}(t)) d\Omega - k \int_\Omega 2\vec{g}(t) \cdot \Delta\vec{g}(t) d\Omega + \\ & 2 \int_S \left[\Delta\vec{g}(t) \cdot \frac{\partial \vec{g}(t)}{\partial \hat{n}} + \vec{g}(t) \cdot \Delta\left(\frac{\partial \vec{g}(t)}{\partial \hat{n}}\right)\right] dS. \end{aligned} \tag{3.11}$$

Making use of Green's identity once more along with the knowledge that both operators, Δ and ∇, are interchangeable, the first integral in the previous equation is modified to

$$-2\int_{\Omega}\nabla\vec{g}(t)\cdot\nabla(\Delta\vec{g}(t))d\Omega = 2\int_{\Omega}\nabla^2\vec{g}(t)\cdot\Delta\vec{g}(t)d\Omega - 2\int_{S}\Delta\vec{g}(t)\cdot\frac{\partial\vec{g}(t)}{\partial\hat{n}}dS \quad (3.12)$$

thus (3.11) may be written as

$$\begin{aligned}\Delta\mathcal{F}(\vec{g}(t)) &= 2\int_{\Omega}\nabla^2\vec{g}(t)\cdot\Delta\vec{g}(t)d\Omega - 2\int_{S}\Delta\vec{g}(t)\cdot\frac{\partial\vec{g}(t)}{\partial\hat{n}}dS - k\int_{\Omega}2\vec{g}(t)\cdot\Delta\vec{g}(t)d\Omega + \\ &\quad 2\int_{S}\Delta\vec{g}(t)\cdot\frac{\partial\vec{g}(t)}{\partial\hat{n}}dS + 2\int_{S}\vec{g}(t)\cdot\Delta\left(\frac{\partial\vec{g}(t)}{\partial\hat{n}}\right)dS. \quad (3.13)\end{aligned}$$

The last surface integral in (3.13) reduces to zero as, in the problems solved here, the Neumann boundary conditions are dependent on an external source and not on the unknown, $\vec{g}(t)$

$$\Delta\left(\frac{\partial\vec{g}(t)}{\partial\hat{n}}\right) = 0. \quad (3.14)$$

Then the first variation of the functional with respect to the unknown, $\vec{g}(t)$, is

$$\Delta\mathcal{F}(\vec{g}(t)) = 2\int_{\Omega}\left[\nabla^2\vec{g}(t)\cdot\Delta\vec{g}(t) - k\vec{g}(t)\cdot\Delta\vec{g}(t)\right]d\Omega. \quad (3.15)$$

But since the vector product is distributive the following is obtained

$$\Delta\mathcal{F}(\vec{g}(t)) = 2\int_{\Omega}\left[\nabla^2\vec{g}(t) - k\vec{g}(t)\right]\cdot\Delta\vec{g}(t)d\Omega. \quad (3.16)$$

Returning to the necessary condition given in (3.10), we must then have

$$\nabla^2\vec{g}(t) - k\vec{g}(t) = 0 \quad (3.17)$$

which is the original mathematical model describing the field propagation through a medium. We therefore see that the functional in (3.9) is appropriate for this type of problem.

Having determined the functional to be made stationary throughout the region of interest, the region is then subdivided into finite elements, such as triangles, making

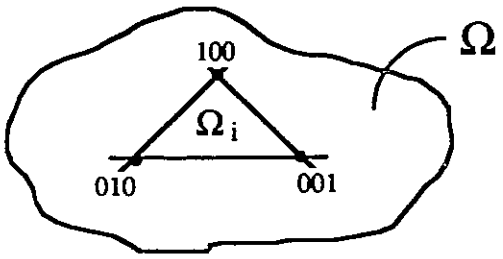


Figure 3.2: Basic triangular element.

it easier to follow the field variations. The field in each subregion is then approximated by n^{th} order polynomials. The derivation of the two-dimensional first order polynomial is given in (3.18) to (3.21). As the extension to higher orders is fairly straightforward when using simplex coordinates [25], it will not be shown here.

A first order polymonial approximating a field oriented in the $\hat{z}$ direction may be given as

$$\vec{g}(t) = (a(t)x + b(t)y + c(t))\hat{z} \tag{3.18}$$

where a, b and c are unknowns and (x,y) are the physical positions of the field. Now when approximating the field by equation (3.18), the solution to the problem would be determined in terms of the unknowns a, b and c of each subregions. But since the solution of these unknowns is insufficient by itself (a substitution is needed to calculate the field), it is in our interest to represent a, b and c as a function of the node potentials (at each vertex of a triangle), and then solve directly for the field at each node.

The field potential at each vertex of a triangle, g_{100}, g_{010} and g_{001}, may be related to the unknowns a, b and c of the same triangle (figure 3.2) by the following system of equations

$$\begin{pmatrix} g_{100}(t) & g_{010}(t) & g_{001}(t) \end{pmatrix} = \begin{pmatrix} a(t) & b(t) & c(t) \end{pmatrix} \begin{pmatrix} x_{100} & x_{010} & x_{001} \\ y_{100} & y_{010} & y_{001} \\ 1 & 1 & 1 \end{pmatrix}. \tag{3.19}$$

The unknowns a, b and c may then be represented by the node potentials g_{100}, g_{010}

and g_{001} as

$$\begin{pmatrix} a(t) & b(t) & c(t) \end{pmatrix} = \begin{pmatrix} g_{100}(t) & g_{010}(t) & g_{001}(t) \end{pmatrix} \begin{pmatrix} x_{100} & x_{010} & x_{001} \\ y_{100} & y_{010} & y_{001} \\ 1 & 1 & 1 \end{pmatrix}^{-1} . \quad (3.20)$$

Then equation (3.18) becomes

$$\vec{g}(t) = \begin{pmatrix} a(t) & b(t) & c(t) \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \hat{z} = \underline{g}(t)\underline{\zeta}(x,y)\hat{z} \quad (3.21)$$

where $\underline{g}(t)$ is a 1x3 row vector with the node potentials as its elements and $\underline{\zeta}(x,y)$ is a 3x1 column vector given as follow

$$\underline{\zeta}(x,y) = \begin{pmatrix} x_{100} & x_{010} & x_{001} \\ y_{100} & y_{010} & y_{001} \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.22)$$

in which -1 is the symbol for the inverse and T for the transpose and (x_{---}, y_{---}) are the physical position of the triangle vertices.

Introducing the shape functions of equation (3.21) into the functional and taking its first variation with respect to the node potential yields a system of equations since the functional must be satisfied *simultaneously* in each subregion. The overall functional then becomes the summation of each subregion integrals

$$\begin{aligned} \mathcal{F}(\vec{g}(t)) \;=\; & \sum_{i=1}^{NSUB} \Bigg[- \int_{\Omega_i} \left\{ \nabla(\underline{g}(t)\underline{\zeta}(x,y)\hat{z}) \cdot \nabla(\underline{g}(t)\underline{\zeta}(x,y)\hat{z}) \right\} d\Omega_i - \\ & k_i \int_{\Omega_i} \left\{ (\underline{g}(t)\underline{\zeta}(x,y)\hat{z}) \cdot (\underline{g}(t)\underline{\zeta}(x,y)\hat{z}) \right\} d\Omega_i \Bigg] + \\ & \sum_{l=1}^{NS} \left[2 \int_{S_l} \left\{ \underline{g}(t)\underline{\zeta}(x,y)\hat{z} \cdot \frac{\partial(\underline{g}(t)\underline{\zeta}(x,y)\hat{z})}{\partial \hat{n}} \right\} dS_l \right] \end{aligned} \quad (3.23)$$

where $NSUB$ is the total number of subregions, NS is the number of Neumann boundaries and k_i is given as $k_i \equiv \mu_i \sigma_i \frac{\partial}{\partial t} + \mu_i \epsilon_i \frac{\partial^2}{\partial t^2}$. Now to solve (3.21), the first

variation with respect to the node potentials is set to zero which produces the following system of equations

$$(S+T+U)\underline{g}=0 \tag{3.24}$$

where the elements of matrices S, T and U are given as

$$\begin{aligned}
S_{mn} &= -\frac{\partial}{\partial g_m}\left\{\sum_{i=1}^{NSUB}\int_{\Omega_i}\left[\nabla(\underline{g}(t)\underline{\zeta}(x,y))\right]^2 d\Omega_i\right\} \\
T_{mn} &= -\frac{\partial}{\partial g_m}\left\{\sum_{i=1}^{NSUB} k_i\int_{\Omega_i}\left(\underline{g}(t)\underline{\zeta}(x,y)\right)^2 d\Omega_i\right\} \\
U_{mn} &= \frac{\partial}{\partial g_m}\left\{\sum_{l=1}^{NS} 2\int_{S_l}\left[\underline{g}(t)\underline{\zeta}(x,y)\hat{z}\cdot\frac{\partial(\underline{g}(t)\underline{\zeta}(x,y)\hat{z})}{\partial\hat{n}}\right] dS_l\right\}
\end{aligned}$$

where m varies from one to the total number of unknowns, n is related to the unknown node potentials inside the integrals and $\underline{g}$ is a column vector including all the node potentials at time t.

3.3 Frequency Domain

The frequency domain is a good starting position to evaluate a numerical technique for a diffusion problem. The major limiting factor is due to the field's attenuation as it propagates into a conducting material. The rate at which the field decays is dictated by the conductivity and permeability of the material as well as the wavelength components of the field in relation to the thickness of the shield's wall. To have a better understanding of their influence, they (with the exception of the wall thickness) have been incorporated into a single parameter, the skin depth δ

$$\delta=\sqrt{\frac{2}{\omega\mu\sigma}} \tag{3.25}$$

where $\omega=2\pi f$. The skin depth defined as in (3.25) is the required thickness of a material with parameters μ & σ, to attenuate an incident field, at frequency f, by a factor of $e^{-1}\simeq 0.368$ [26].

3.3.1 Solving the System of Equations

For the frequency domain analysis, the Fourier transform needs to be performed on all time dependent variables defined in section 3.2. Therefore, the final system of equations to be solved becomes the following

$$(Q + R)\underline{G} = 0 \tag{3.26}$$

where the matrix Q is the Fourier transform of the summation of S and T matrices (see equation (3.24)), given as

$$Q_{mn} = -\frac{\partial}{\partial G_m}\left\{\sum_{i=1}^{NSUB}\left[\int_{\Omega_i}[\nabla(\underline{G\zeta}(x,y))]^2 d\Omega_i + K_i\int_{\Omega_i}[\underline{G\zeta}(x,y)]^2 d\Omega_i\right]\right\}$$

and R is the transform of matrix U in (3.24),

$$R_{mn} = \frac{\partial}{\partial G_m}\left\{\sum_{i=1}^{NS} 2\int_{S_i}\underline{G\zeta}(x,y)\hat{z}\cdot\frac{\partial(\underline{G\zeta}(x,y)\hat{z})}{\partial\hat{n}}dS_l\right\}$$

in which G defines the Fourier transform of g and the variable K_i is the transform of k_i, given as

$$K_i = \mathcal{F}\left\{\mu_i\epsilon_i\frac{\partial^2}{\partial t^2} + \mu_i\sigma_i\frac{\partial}{\partial t}\right\} = -\omega^2\mu_i\epsilon_i + j\omega\mu_i\sigma_i \tag{3.27}$$

in which j denotes the imaginary part of a complex variable. As the final system of equations is linear, it may be directly solved by a matrix reduction technique such as the Gaussian elimination [27].

3.3.2 Realistic Frequency Diffusion Problems and Validation

In the FEM, the field variations are approximated by shape functions in each discrete subregion. As the skin depth increases, it becomes harder for the shape functions associated with the subregions to follow the field variation. Therefore, to make the task possible and easier, one may increase the order of the polynomials used in the shape functions and/or decrease the area of the subregions. Considering the computer

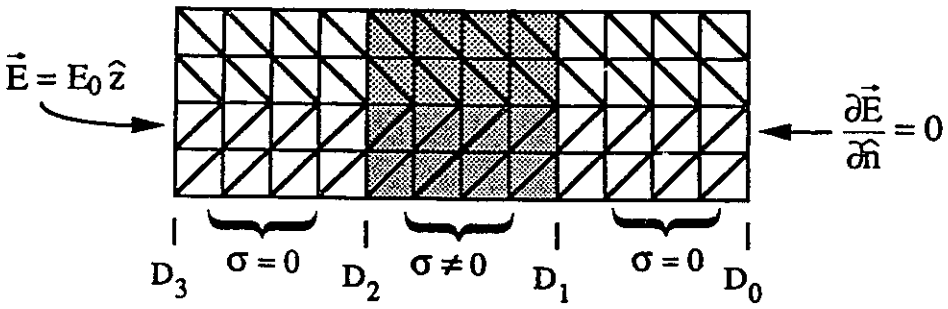


Figure 3.3: Division into FEs for frequency domain.

Table 3.1: Comparison between FEM and analytical solution for various δ / d.

δ / d (%)	*Attenuation* $\frac{E(D_2)}{E(D_0)}$ *(dB)*	*Analytical*		*FEM*		
		$E(D_2)$	$E(D_0)$	$E(D_2)$	$E(D_0)$	*order of polynomials*
39	1.63E01	3.470	0.527E+00	3.470	0.535E+00	1^{st}
21	3.48E01	1.890	0.343E-01	1.890	0.361E-01	1^{st}
15	5.18E01	1.340	0.346E-02	1.340	0.411E-02	1^{st}
11	7.57E01	0.945	0.156E-03	0.945	0.155E-03	3^{rd}
8	1.03E02	0.707	0.493E-05	0.707	0.371E-05	3^{rd}
6.5	1.28E02	0.577	0.239E-06	0.577	0.805E-06	3^{rd}
5.6	1.48E02	0.500	0.191E-07	0.500	0.693E-06	3^{rd}
5.3	1.66E02	0.447	0.210E-08	0.447	0.412E-06	3^{rd}

limitations, diffusion problems can be solved practically for a certain range of skin depth, δ, and wall thickness, d. It isn't independently the skin depth nor the wall thickness that determines this range but the relation between the two. To have an idea of the limits that the FEM poses to the solution of diffusion problems, the numerical technique is applied to calculate the field penetration through a wall, for various thickness and constitutive parameters, and the results are compared with the analytical solution derived in section 2.1.1.

The results of this analysis are shown in Table 3.1, where the leftmost column refers to the percentage of the skin depth in relation to the wall thickness. The

solution by FEM was set up with first order polynomials and four divisions in the wall for δ/d down to 15% and, for lower percentage, with third order polynomials with the same number of divisions (see figure 3.3 for the finite element division). As seen from Table 3.1, the FEM can be used practically (with the available computer memory limits), in relatively large regions, for skin depth as low as seven percent of the wall thickness. Obviously, for one-dimensional problems the FEM could be used for much lower δ/d percentage as the computer memory limits present no longer a problem.

3.4 Time Domain

The time domain analysis concentrates on the diffusion process of electromagnetic pulse (EMP) types impinging plane waves. EMP type fields are defined here as time varying fields that can be modeled as a double exponential of the following general form

$$e(t) = E_0\{e^{-\alpha t} - e^{-\beta t}\} \tag{3.28}$$

Three different sources of EMP type fields are analysed throughout this chapter, namely lightning, ESD and NEMP. Although the emphasis is on NEMP generated fields, the added analysis of lightning and ESD sources will facilitate the understanding of the implications that NEMP type fields have on the shielding performances of enclosures. Table 3.2 gives the characteristics of the three EMP type fields to be analysed [28] and shown in figure 3.4 are their time signatures.

3.4.1 Solving the System of Equations

The method of solution in the time domain is a combination of two numerical techniques, namely the FEM and a time integration or differentiation method.

The final system of equations, (3.24), may be rewritten in a more practical manner

$$V\underline{g} + W\frac{\partial}{\partial t}\underline{g} + Z\frac{\partial^2}{\partial t^2}\underline{g} = 0 \tag{3.29}$$

Table 3.2: EMP type fields characteristics.

Type of EMP field	E_0 (V/m)	α (s^{-1})	β (s^{-1})
Lightning	E_0	10^4	0.5×10^6
ESD	E_0	1.6×10^6	10^9
NEMP	5.25×10^4	4×10^6	4.76×10^8

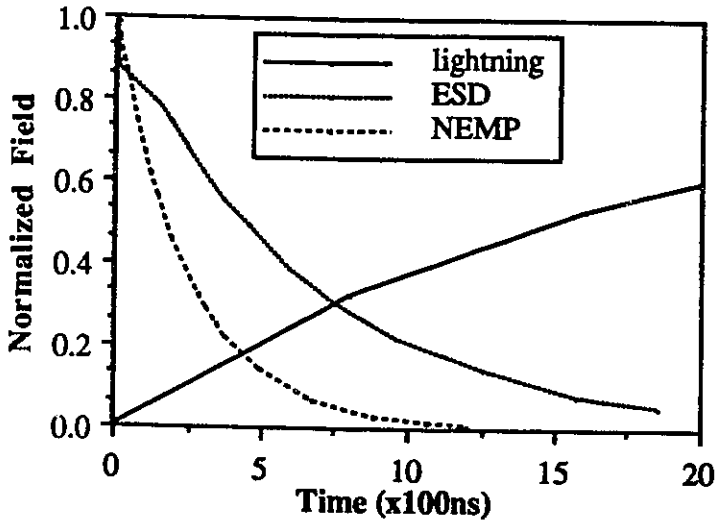


Figure 3.4: Time signatures of the EMP type fields.

with matrix V, being the summation of S and U matrices in (3.24),

$$V_{mn} = -\frac{\partial}{\partial g_m}\left\{\sum_{i=1}^{NSUB}\int_{\Omega_i}[\nabla(\underline{g}(t)\underline{\zeta}(x,y))]^2 d\Omega_i - \sum_{l=1}^{NS} 2\int_{S_l}\left[\underline{g}(t)\underline{\zeta}(x,y)\hat{z}\cdot\frac{\partial(\underline{g}(t)\underline{\zeta}(x,y)\hat{z})}{\partial\hat{n}}\right]dS_l\right\}$$

and matrices W and Z are obtained directly from the T matrix by parting k_i,

$$W_{mn} = -\frac{\partial}{\partial g_m}\left\{\sum_{i=1}^{NSUB}\mu_i\sigma_i\int_{\Omega_i}[\underline{g}(t)\underline{\zeta}(x,y)]^2 d\Omega_i\right\}$$

$$Z_{mn} = -\frac{\partial}{\partial g_m}\left\{\sum_{i=1}^{NSUB}\mu_i\epsilon_i\int_{\Omega_i}[\underline{g}(t)\underline{\zeta}(x,y)]^2 d\Omega_i\right\}.$$

Now the effect of the displacement current (the term related to $\mu\epsilon\frac{\partial^2}{\partial t^2}$ in equation (3.29)) could be neglected for fields varying relatively slowly with time (fields with frequency components lower than about 10 GHz). But as most, possibly all, numerical integration techniques available in software packages require that the ordinary differential equations (ODE) have the following format

$$\begin{aligned}\dot{y} &= f(t,y)\\ y(t_0) &= y_0\end{aligned} \tag{3.30}$$

one would need to find the inverse of the matrix W which is singular, as in some regions $\sigma = 0$. Thus this formulation of an ODE, for the situation of interest, is unattainable when neglecting the displacement current term. To overcome this, one might instinctively reduce equation (3.29) to a differential equation of the first order by setting the first derivative with respect to time of $\underline{g}$ to a vector $\underline{f}$

$$\underline{f} = \dot{\underline{g}} \tag{3.31}$$

and transform equation (3.29) into

$$Z\dot{\underline{f}} = -W\underline{f} - V\underline{g}. \tag{3.32}$$

Now as the matrix Z is a non-singular matrix, this leads to the following system of equations

$$\begin{bmatrix} \dot{\underline{g}} \\ \dot{\underline{f}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -Z^{-1}V & -Z^{-1}W \end{bmatrix} \begin{bmatrix} \underline{g} \\ \underline{f} \end{bmatrix} \tag{3.33}$$

where I is a unit matrix and Z^{-1} is the inverse of Z.

At the first glance, it might seem that the system of equations presented in (3.33) is now ready to be solved by an available ODE solver such as DVERK (found in IMSL Fortran package [29]). However, it turns out that (3.29) is a highly unstable system of equations, it is ill-posed as the elements of the matrix Z are orders of magnitude lower than those of matrices V and W. In order to solve the ODE in (3.33), one would need ridiculously small time steps, and even so, the system would still be very unstable. Thus the solution of (3.29) in a practical time frame requires that the displacement current term be neglected. Since the matrix W is singular, an ODE is out of the question, and the system to be solved becomes a differential/algebraic equation(DAE) system described as [30]

$$\begin{aligned} f(t, y, \dot{y}) &= 0 \\ y(t_0) &= y_0 \\ \dot{y}(t_0) &= \dot{y}_0. \end{aligned} \tag{3.34}$$

After an extensive search, the author came across a program called DDASSL developed by Linda R. Petzold, from Sandia National Laboratories [31]. This program is a DAE solver based on backward differentiation formulas which solves equation (3.34) directly by replacing the derivative with a difference approximation, and then solve the resulting equation using Newton's method at the current time t_n. DDASSL approximates the derivative via k^{th} order backward differentiation formula, where k ranges from one to five. For example, replacing the derivative by the backward difference in (3.34), the first order formula can be written as

$$F\left(t_n, y_n, \frac{y_n - y_{n-1}}{\Delta t_n}\right) = 0. \tag{3.35}$$

Then using Newton's method, the equation is solved at t_n for y_n as

$$y_n^{m+1} = y_n^m - \left(\frac{\partial F}{\partial \dot{y}} + \frac{1}{\Delta t_n}\frac{\partial F}{\partial y}\right)^{-1} F\left(t_n, y_n^m, \frac{y_n^m - y_{n-1}^m}{\Delta t_n}\right) \tag{3.36}$$

where m is the iteration index. As will be seen in section 3.4.3, the method used in DDASSL is very well suited to the type of problems considered herin.

3.4.2 Realistic Time Diffusion Problems

In the time domain investigation, we are confronted with two possible limitations. One is related to the rise and fall times of the incident field; the faster the rise or fall time, the smaller the time steps in the differentiation technique needs to be. As in EMP type fields there is a combination of a fast rise time compared to a much lower fall time, the technique must then have variable time steps to be effective, which is the case for the program DDASSL. The second limitation is again related to the steepness of the rise and fall time but is due to the approximation of the field by the polynomials in the FEM. The steeper they get, the higher the field frequency components will reach. We are then limited by the skin depth related to the higher frequencies and the wall thickness.

The effect of the skin depth and wall thickness was demonstrated in section 3.3.2. Now to apply this information to the time-domain problem of interest, the EMP type fields first need to be transformed in the frequency domain

$$\underline{E}(\omega) = \mathcal{F}\{E_0[e^{-\alpha t} - e^{-\beta t}]\} \tag{3.37}$$

given in (3.38) as the magnitude normalized to E_0.

$$\frac{|\underline{E}(\omega)|}{E_0} = \frac{\beta - \alpha}{\sqrt{\omega^4 + \omega^2(\alpha^2 + \beta^2) + \alpha^2\beta^2}}. \tag{3.38}$$

To properly characterize the field penetration, the subregions within the walls should be defined in a way to follow the field variations up to the frequency for which

$$|E(2\pi f_{max})| \leq 0.02 \times |E(0Hz)|. \tag{3.39}$$

Table 3.3: Realistic diffusion problems.

Type of EMP field	f_{max} (MHz)	*Copper* ($\mu_r = 1; \sigma = 0.56E8S/m$)		*Steel* ($\mu_r = 200; \sigma = 0.1E8S/m$)	
		Skin depth for f_{max} (μm)	max. wall thickness (1^{st}ord)(mm)	Skin depth for f_{max} (μm)	max. wall thickness (1^{st}ord)(mm)
Lightning	0.05	301.0	2.010	50.30	0.333
ESD	10.00	21.3	0.142	3.56	0.024
NEMP	20.00	15.0	0.100	2.52	0.017

Here two one hundrets is chosen as the limiting factor since most of the damaging energy is situated in the lower spectrum. Once the maximum frequency, f_{max}, is determined, the lowest skin depth can be calculated for various materials as in Table 3.3.

The code made use of first order polynomials to follow the field variations and four divisions in the walls, therefore realistic diffusion problems in this case relate to situations for which the percentage of skin depth over the wall thickness is greater or equal to fifteen percent (see Table 3.1). Applying this virtual limit to the skin depth calculated in Table 3.3 gives the maximum wall thickness for which the shielding effectiveness can be solved.

3.4.3 Validation

The numerical technique presented in section 3.4 is validated against the analytical solution of a one-dimensional diffusion problem as presented in section 2.1.2. The test problem was analysed for various geometries and impinging fields to show the effect that the constitutive parameters and the thickness of the shield has on the accuracy of the numerical method. The field of interest consists of lightning, ESD and NEMP

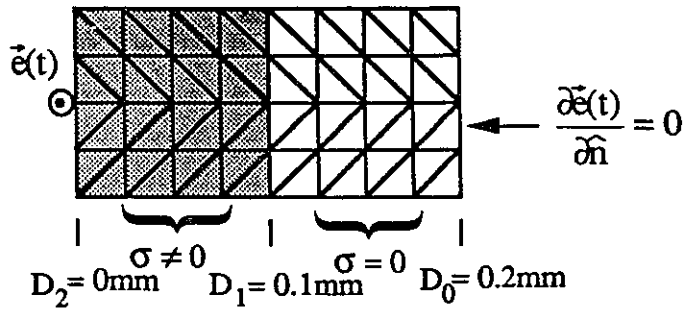


Figure 3.5: Division into FEs for time-domain.

with the characteristics as given in Table 3.2.

The division into triangular finite elements of the one-dimensional test problem (see figure 2.1) is shown in figure 3.5. Figures 3.6(a-c) show the comparison between the analytical and the finite element solution for copper walls of one-tenth of a millimeter thick with three types of impinging fields (the electric field is normalized to the outside peak field). As may be observed, there is a perfect agreement between the FEM and the analytical solution. In figure 3.6(d), both methods were applied to the diffusion of an NEMP field through steel walls of the same thickness. Notice that the diffused field inside the steel walls is more attenuated, has a slower rise time and a flatter response than that of the copper shield (see figure 3.6(c)). It can be seen that the FEM can still follow fairly well the field penetration even when the width of the wall is five times greater than the computed maximum wall thickness (refer to Table 3.3).

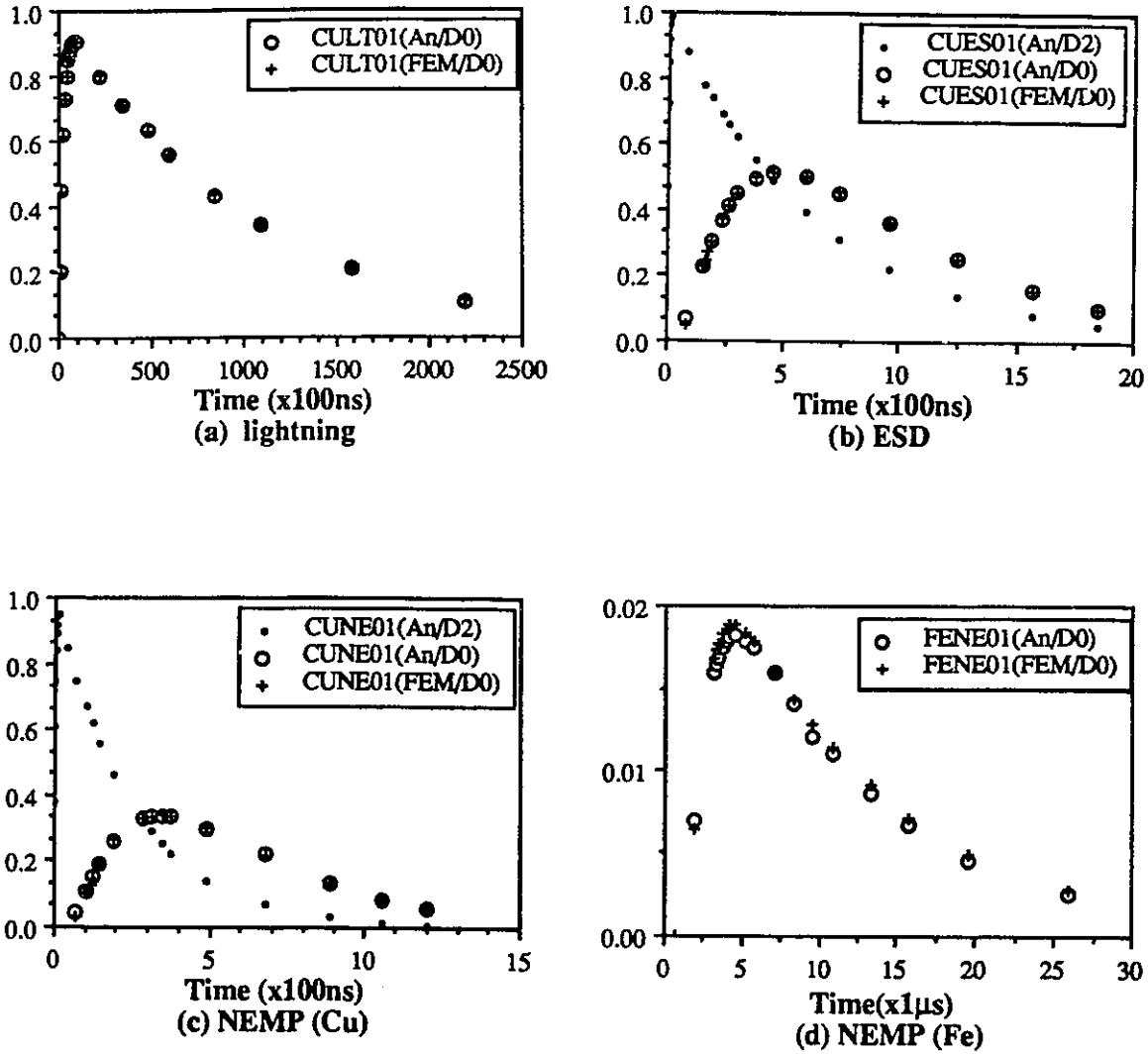


Figure 3.6: One-dimensional time-domain comparison of analytical solution with FEM.

Chapter 4

Transmission-Line-Matrix

4.1 Introduction

The transmission line matrix method finds its origin in Huygen's principle of light [32] which states that a wave front consists of secondary wave sources which in turn radiate thus creating a new wave front with the envelope of these secondary wavelets (see figure 4.1).

John's and Beurle's idea [33] was to implement this principle to solve electromagnetic problems. To accomplish this, they discretized a two-dimensional region into interconnecting transmission lines in which each junction (node) would then become a secondary radiator due to the reflection and transmission of an incoming wave on a node (see figure 4.2).

Thus a field, in this model, travelling along a transmission line would be scattered in four directions upon reaching a junction as there are impedance discontinuities at each junction. In a lossless and homogeneous region with characteristic impedance of Z_c, a field travelling towards a node with an amplitude of $1V$ is reflected back with an amplitude of $-\frac{1}{2}V$ since the node impedance seen from the incident branch is $\frac{Z_c}{3}$,

$$\Gamma = \frac{\frac{Z_c}{3} - Z_c}{\frac{Z_c}{3} + Z_c} = -\frac{1}{2}. \tag{4.1}$$

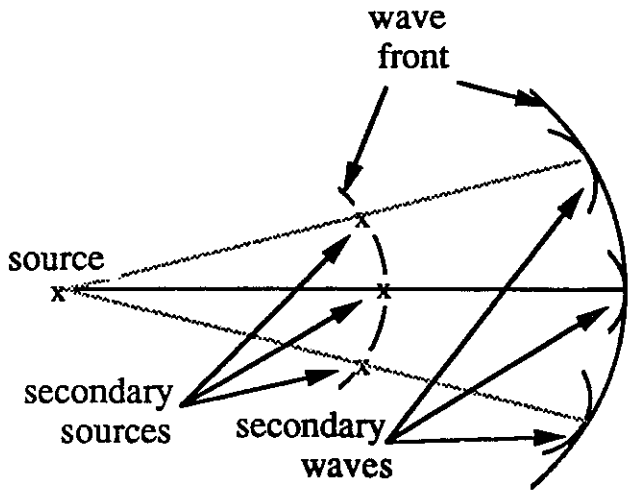
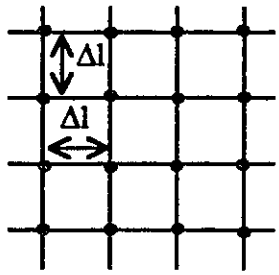
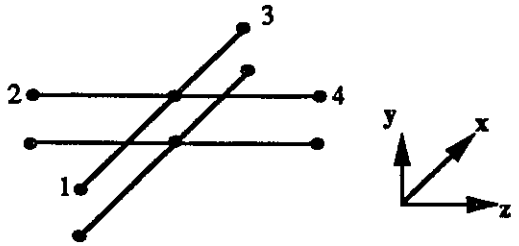


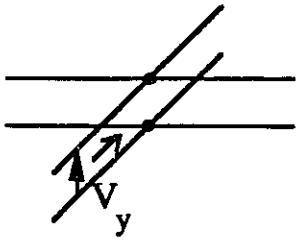
Figure 4.1: Formation of wave fronts as explained by Huygens principle.



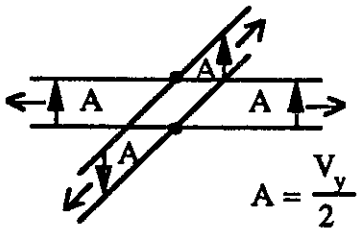
(a) Discretization of the region into interconnecting TLs



(b) Interconnecting TLs



(c) Incident field on junction



(d) Scattered field from junction for the lossless case

Figure 4.2: Interconnecting transmission lines (TLs).

A field is also generated in the three other branches having an amplitude of $(1+\Gamma)\cdot 1V = +\frac{1}{2}V$ in each one of them.

The analysis in the TLM method is conducted directly in the time domain for which the impinging field's or point source's time signature is discretized in time as a succession of Dirac impulses. These impulses are spaced by

$$\Delta t = \Delta l / c_0 \tag{4.2}$$

in a uniformly gridded region.

The reflected voltages in the four branches of a unit cell at the time $(k+1)*\Delta t$ is then related to the incident voltages on the node at time $k*\Delta t$ by the scattering matrix S [33] given below

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}_{k+1}^{r} = S \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}_{k}^{i} \tag{4.3}$$

where

$$S = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

To give a visual representation of the propagation of the field in such a structure, one Dirac impulse is sent from one node in the four directions and the scattered fields are calculated for two iterations (see figure 4.3).

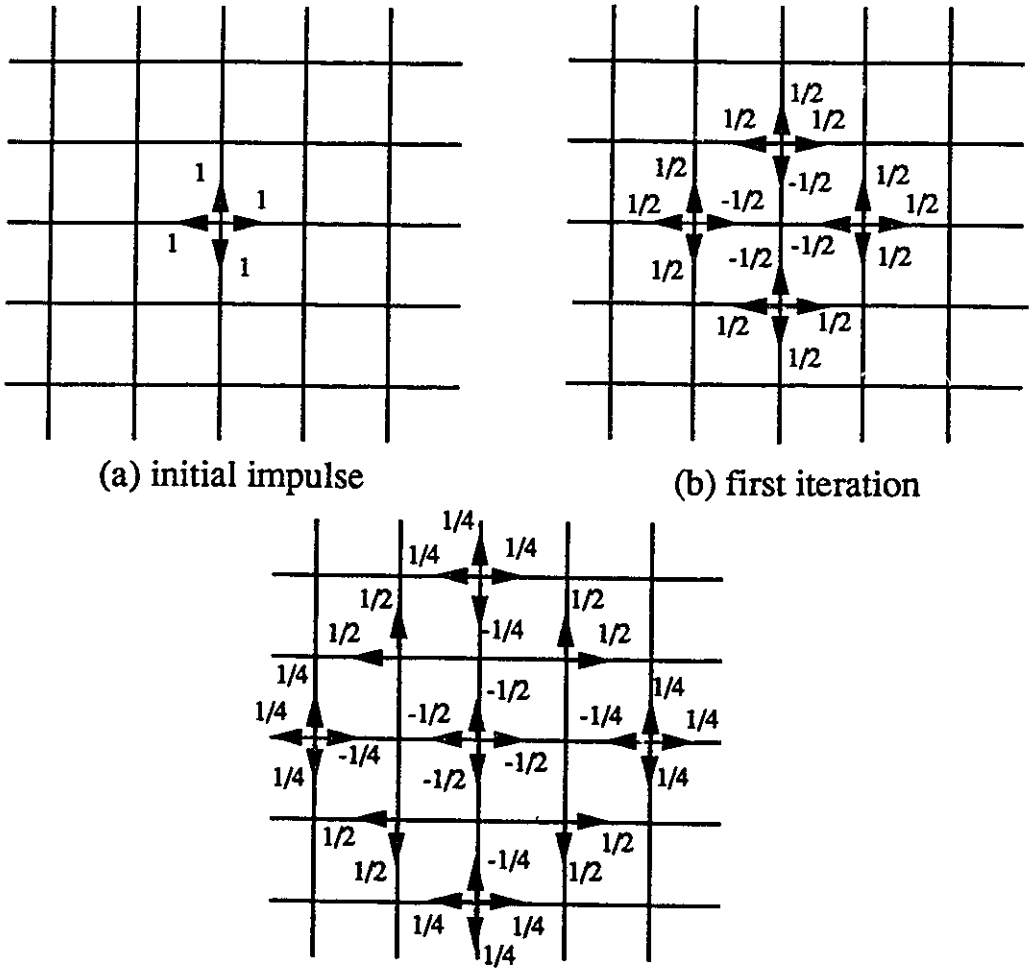


Figure 4.3: Scattered field from one Dirac impulse in a homogeneous region.

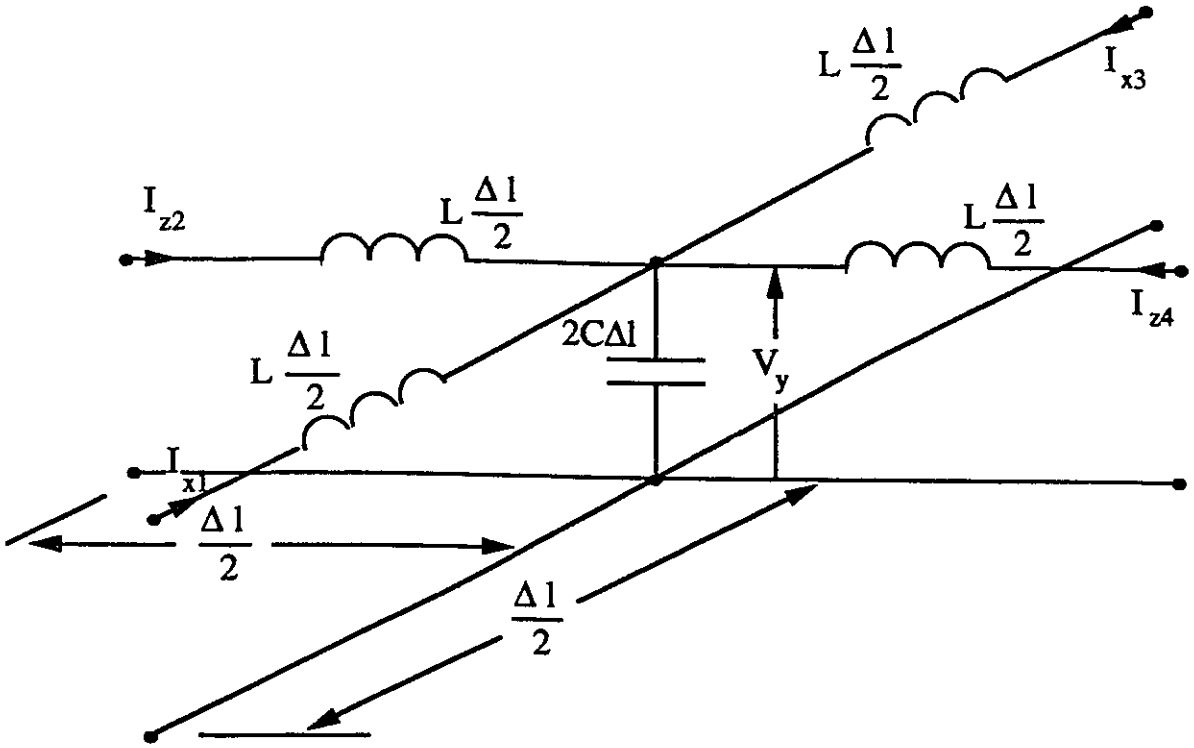


Figure 4.4: Basic shunt network.

4.2 Mathematical Formulation

4.2.1 Lossless Medium

The mapping, for a lossless medium, of the voltages and currents in the transmission lines to the field quantities being modeled is obtained by replacing the transmission lines with their equivalent lumped parameters such as inductances and capacitances. Then, for a certain inductance and capacitance per unit length represented respectively by L and C, the basic network in the two-dimensional analysis may be represented as shown in figure 4.4, known as the shunt configuration.

The simulation of a transverse electric (TE) field propagating in a certain medium is conducted by discretizing the region with the use of the basic shunt network. In such a model, the voltages and currents may be approximated by the following equations

when the spacing between the junctions, Δl, is relatively small with respect to the wavelength [33]

$$-\frac{\partial}{\partial x}(I_{x_1} - I_{x_3}) - \frac{\partial}{\partial z}(I_{z_2} - I_{z_4}) = 2C\frac{\partial V_y}{\partial t} \tag{4.4}$$

$$-\frac{\partial V_y}{\partial x} = L\frac{\partial}{\partial t}(I_{x_1} - I_{x_3}) \tag{4.5}$$

$$-\frac{\partial V_y}{\partial z} = L\frac{\partial}{\partial t}(I_{z_2} - I_{z_4}) \tag{4.6}$$

which may be combined to give a wave equation

$$\frac{\partial^2 V_y}{\partial x^2} + \frac{\partial^2 V_y}{\partial z^2} = 2LC\frac{\partial^2 V_y}{\partial t^2}. \tag{4.7}$$

The corresponding field equations for a propagating TE wave are given by Maxwell's equations as

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \epsilon\frac{\partial E_y}{\partial t} \tag{4.8}$$

$$\frac{\partial E_y}{\partial z} = \mu\frac{\partial H_x}{\partial t} \tag{4.9}$$

$$\frac{\partial E_y}{\partial x} = -\mu\frac{\partial H_z}{\partial t} \tag{4.10}$$

which combined give the wave equation

$$\frac{\partial^2 E_y}{\partial x^2} + \frac{\partial^2 E_y}{\partial z^2} = \mu\epsilon\frac{\partial^2 E_y}{\partial t^2}. \tag{4.11}$$

By relating equations (4.4)–(4.7) to equations (4.8)–(4.11) the following mapping may be established, for lossless transmission lines,

$$E_y \equiv V_y \quad ; \quad H_z \equiv (I_{x_1} - I_{x_3}) \quad ; \quad H_x \equiv -(I_{z_2} - I_{z_4})$$

$$\mu \equiv L \quad ; \quad \epsilon \equiv 2C.$$

The implication of this mapping is well understood when considering wave propagation in free space ($\mu_r = 1$ and $\epsilon_r = 1$). As the inductance and capacitance per unit length are related by

$$\frac{1}{\sqrt{LC}} = \frac{1}{\sqrt{\mu_0\epsilon_0}} = c_0,$$

a field propagating in free space at the speed of light would then be simulated, in TLM, as propagating in a medium with a relative permittivity twice that of free space. To account for this inherent error, one has to be careful of how they relate the propagation constant of the network to that of the actual situation.

In the lossless case, the reflected fields are related to the incident fields by the scattering matrix as given in equation (4.3).

4.2.2 Lossy and Dielectric Medium

To represent a TE field propagating in a lossy and/or dielectric medium, the shunt transmission lines are loaded with infinite loss shunt stubs and/or $\Delta l/2$ open stubs respectively (see figure 4.5). In such a medium, equations (4.4) and (4.8) become [34]-[35]

$$-\frac{\partial}{\partial x}(I_{x_1} - I_{x_3}) - \frac{\partial}{\partial z}(I_{z_2} - I_{z_4}) = 2C(1 + Y_0/4)\frac{\partial V_y}{\partial t} + \frac{G_0 C c_0}{\Delta l} V_y \quad (4.12)$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \epsilon_0 \epsilon_r \frac{\partial E_y}{\partial t} + \sigma E_y \quad (4.13)$$

where G_0 is the characteristic admittance of the infinite loss stub normalized to the characteristic admittance of the mainlines. The mapping of the field quantities to those of the model is then

$$E_y \equiv V_y \;\; ; \;\; H_z \equiv (I_{x_1} - I_{x_3}) \;\; ; \;\; H_x \equiv -(I_{z_2} - I_{z_4})$$

$$\mu_0 \equiv L \;\; ; \;\; \epsilon_0 \equiv 2C \;\; ; \;\; \epsilon_r \equiv 1 + Y_0/4 \;\; ; \;\; \sigma \equiv \frac{G_0 C c_0}{\Delta l} \equiv \frac{G_0}{\Delta l}\sqrt{\frac{\epsilon_0}{\mu_0}}$$

when a lossy and dielectric medium is present. With the added representation of losses, the basic shunt network is now a six port network.

In this model, the incident fields are discretized in time and injected in the mesh, at certain nodes, as impulses of different amplitude. The reflected voltages from a node at time $(k+1) * \Delta t$ are related to the incident voltages on the same node at

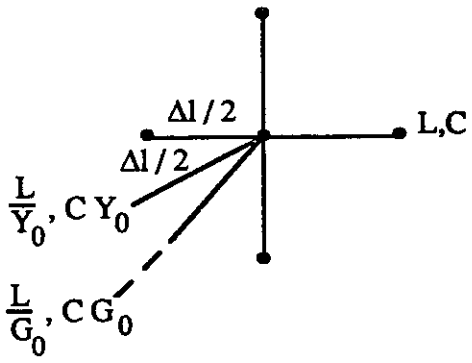


Figure 4.5: Top view of node loaded with $\frac{\Delta l}{2}$ open stub and infinite loss stub.

time $k * \Delta t$ by the scattering matrix S [35]

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}^r_{k+1} = S \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}^i_k \tag{4.14}$$

where

$$S = \frac{2}{Y} \begin{bmatrix} 1 & 1 & 1 & 1 & Y_0 \\ 1 & 1 & 1 & 1 & Y_0 \\ 1 & 1 & 1 & 1 & Y_0 \\ 1 & 1 & 1 & 1 & Y_0 \\ 1 & 1 & 1 & 1 & Y_0 \end{bmatrix} - I$$

and $Y = 4 + Y_0 + G_0$; $Y_0 = 4(\epsilon_r - 1)$; $G_0 = \sigma \Delta l \sqrt{\mu_0/\epsilon_0}$, and I is a unit matrix.

Equation (4.14) gives the generalized scattering matrix, S, when modeling a dielectric and lossy medium. Notice that the voltage scattered into the infinite loss shunt (V_0) is not needed in the calculation as none of the lost energy is reflected back to the node.

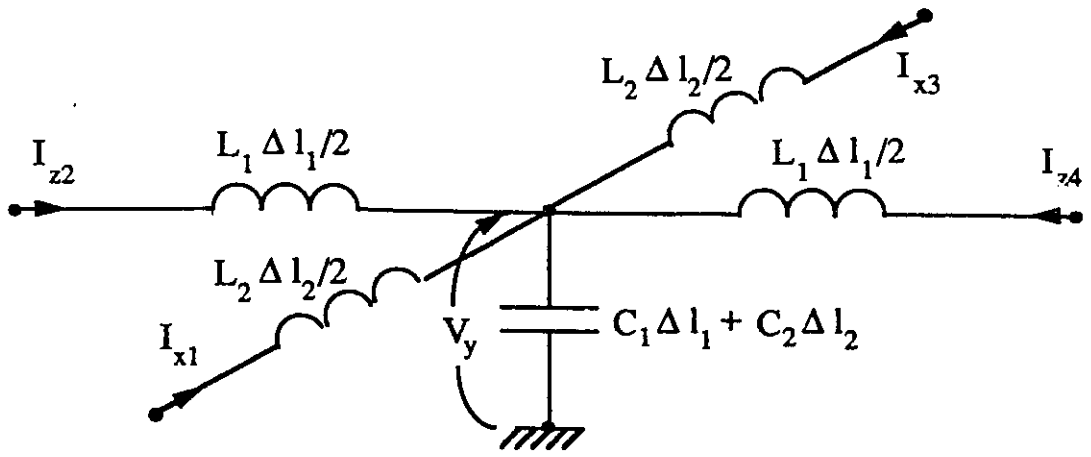


Figure 4.6: Shunt network with variable branch length.

4.2.3 Variable Mesh Size

In order to analyse a large region in which small discontinuities are present, one would need to discretize the whole region in small square meshes, when using the conventional TLM method. This would then lead to a great number of nodes in regions where it is not needed, thus immensely increasing the memory requirements and more importantly, the processing time. To overcome this problem, rectangular mesh (or variable mesh size) have been introduced by Saguet and Pic [36].

In a variable mesh configuration, the basic cell, as shown in figure 4.4, is modified to include different branch length and impedances. An intersection of two transmission lines of length Δl_1 and Δl_2 and of inductance and capacitance per unit length of L_1, C_1, and L_2, C_2 is then described as (see figure 4.6)

$$-\Delta l_1 \frac{\partial}{\partial x}(I_{x_1} - I_{x_3}) - \Delta l_2 \frac{\partial}{\partial z}(I_{z_2} - I_{z_4}) = (C_1 \Delta l_1 + C_2 \Delta l_2) \frac{\partial V_y}{\partial t} \quad (4.15)$$

$$-\frac{\partial V_y}{\partial x} = L_1 \frac{\partial}{\partial t}(I_{x_1} - I_{x_3}) \quad (4.16)$$

$$-\frac{\partial V_y}{\partial z} = L_2 \frac{\partial}{\partial t}(I_{z_2} - I_{z_4}). \quad (4.17)$$

Now to alleviate the complexity of the network Δl_2 is chosen to be an integer multiplier (N) of Δl_1. Thus by setting $\Delta l_2 = N \Delta l_1$, equations (4.15) to (4.17) reduce

to

$$-\frac{\partial}{\partial x}(I_{x_1} - I_{x_3}) - N\frac{\partial}{\partial z}(I_{z_2} - I_{z_4}) = 2C_1\frac{\partial V_y}{\partial t} \tag{4.18}$$

$$-\frac{\partial V_y}{\partial x} = L_1\frac{\partial}{\partial t}(I_{x_1} - I_{x_3}) \tag{4.19}$$

$$-\frac{\partial V_y}{\partial z} = NL_1\frac{\partial}{\partial t}(I_{z_2} - I_{z_4}) \tag{4.20}$$

since $L_2 = NL_1$ and $C_2 = C_1/N$. When combined, equations (4.18) to (4.20) represent a wave equation

$$\frac{\partial^2 V_y}{\partial x^2} + \frac{\partial^2 V_y}{\partial z^2} = 2L_1C_1\frac{\partial^2 V_y}{\partial t^2}. \tag{4.21}$$

The speed of propagation in a variable mesh configuration is conserved, $\sqrt{L_1C_1} = \sqrt{L_2C_2}$, but the characteristic impedances are different

$$\sqrt{\frac{L_1}{C_1}} = \frac{1}{N}\sqrt{\frac{L_2}{C_2}}. \tag{4.22}$$

This difference in the characteristic impedances means that the scattering matrix will differ from the uniform mesh division. For the rectangular mesh, the reflected voltages are related to the incident ones by the following equation

$$V^r_{n+1} = \frac{2\sum_m Y_m V^i_m}{\sum_m Y_m} - V^i_n \tag{4.23}$$

or

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}^r_{k+1} = S \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}^i_k \tag{4.24}$$

with

$$S = \frac{2}{\sum_{m=1}^4 Y_m}\begin{bmatrix} Y_1 & Y_2 & Y_3 & Y_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \end{bmatrix} - I$$

where $Y_3 = Y_1$ and $Y_4 = Y_2$.

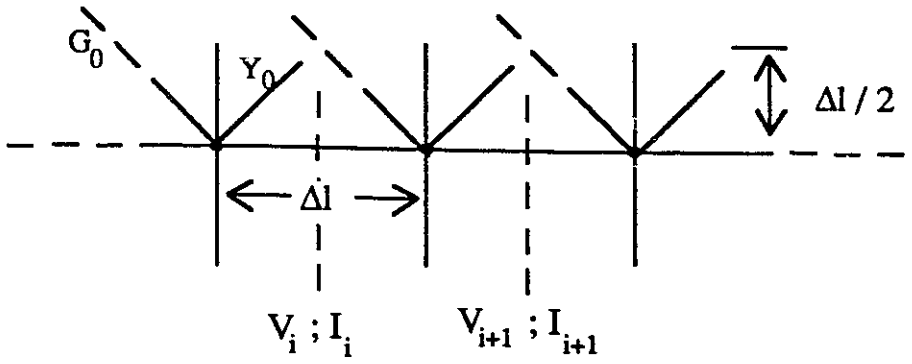


Figure 4.7: One-dimensional TLM.

4.3 Realistic Diffusion Problems to be Solved

As the regions analysed by the TLM method are discretized into interconnecting transmission lines, problems might occur if one isn't careful. Depending on the frequencies of operation and the angle of propagation, the mesh may present a cutoff network to the fields. To calculate the range of frequencies for which the network is valid, the transmission line matrix is analysed for a plane wave travelling in the axial direction (this gives the worst case situation for the network) within a homogeneous region. Since the transmission lines are in a homogeneous region and the impinging fields are plane waves, the analysis of the propagating wave may be conducted via a one-dimensional TLM. In this case the wave travels along a transmission line which is periodically loaded with open circuited stubs of length $\Delta l/2$ as shown in figure 4.7. Now relating V_i & I_i to V_{i+1} & I_{i+1} in figure 4.4, one finds the following [37]

$$\begin{vmatrix} V_i \\ I_i \end{vmatrix} = \begin{vmatrix} \cosh\gamma\frac{\Delta l}{2} & \sinh\gamma\frac{\Delta l}{2} \\ \sinh\gamma\frac{\Delta l}{2} & \cosh\gamma\frac{\Delta l}{2} \end{vmatrix} \times \begin{vmatrix} 1 & 0 \\ (2+Y_0)\tanh\gamma\frac{\Delta l}{2} & 1 \end{vmatrix} \times \begin{vmatrix} \cosh\gamma\frac{\Delta l}{2} & \sinh\gamma\frac{\Delta l}{2} \\ \sinh\gamma\frac{\Delta l}{2} & \cosh\gamma\frac{\Delta l}{2} \end{vmatrix} \times \begin{vmatrix} V_{i+1} \\ I_{i+1} \end{vmatrix} \tag{4.25}$$

A wave propagating in a periodic structure such as this one, with a propagating constant $\gamma_n = \alpha_n + j\beta_n$, would then have the following relationship

$$\begin{vmatrix} V_i \\ I_i \end{vmatrix} = \begin{vmatrix} e^{\gamma_n \Delta l} & 0 \\ 0 & e^{\gamma_n \Delta l} \end{vmatrix} \times \begin{vmatrix} V_{i+1} \\ I_{i+1} \end{vmatrix}. \tag{4.26}$$

The solution to equations (4.25) and (4.26) gives

$$(2 + \frac{Y_0}{2}) \cosh \gamma \Delta l - (1 + \frac{Y_0}{2}) = \cosh \gamma_n \Delta l. \tag{4.27}$$

Replacing γ by $\alpha + j\beta$ and γ_n by $\alpha_n + j\beta_n$ in (4.27) and equating the real and imaginary parts respectively, the following relations are obtained

$$(2 + \frac{Y_0}{2}) \cos \beta \Delta l \cosh \alpha \Delta l - (1 + \frac{Y_0}{2}) = \cos \beta_n \Delta l \cosh \alpha_n \Delta l \tag{4.28}$$

$$(2 + \frac{Y_0}{2}) \sin \beta \Delta l \sinh \alpha \Delta l = \sin \beta_n \Delta l \sinh \alpha_n \Delta l \tag{4.29}$$

in which (4.28) corresponds to the real part and (4.29) to the imaginary. Assuming that both $\alpha \Delta l$ and $\alpha_n \Delta l$ are much smaller than one ($\alpha \Delta l$ & $\alpha_n \Delta l << 1$), equations (4.28) and (4.29) may be rewritten as

$$\frac{\beta}{\beta_n} = \frac{\pi(\frac{\Delta l}{\lambda})}{\sin^{-1}\{\sqrt{2(1 + \frac{Y_0}{4})} \sin(\pi \frac{\Delta l}{\lambda})\}} \tag{4.30}$$

$$\frac{\alpha}{\alpha_n} = \frac{\sqrt{1 - 2(1 + \frac{Y_0}{4}) \sin^2(\pi \frac{\Delta l}{\lambda})}}{\sqrt{2(1 + \frac{Y_0}{4})} \cos(\pi \frac{\Delta l}{\lambda})} \tag{4.31}$$

where $\alpha = \frac{G_0}{4\Delta l(1 + \frac{Y_0}{4})}$. From equations (4.30) and (4.31), the first network cutoff frequency is then found to be

$$\left(\frac{\Delta l}{\lambda}\right)_{cutoff} = \frac{1}{\pi} \sin^{-1}\left(\frac{1}{\sqrt{2(1 + \frac{Y_0}{4})}}\right). \tag{4.32}$$

At relatively low frequencies, well below cutoff, the propagation constants α_n and β_n of the network are relatively constant, thus equation (4.30) and (4.31) reduce to

$$\gamma_n = \sqrt{2(1 + \frac{Y_0}{4})}\, \gamma. \tag{4.33}$$

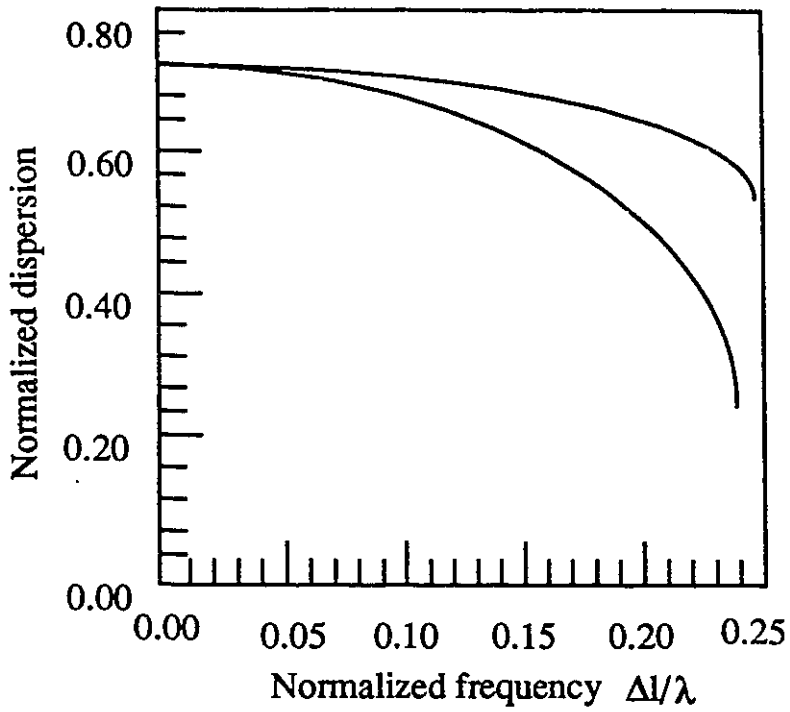


Figure 4.8: Dispersion vs frequency of TLM network.

As demonstrated, when the wavelength of propagation approaches the transmission line mesh size, the equivalence between the network constants and the constant of a continuous medium collapses (see figure 4.8). Equation (4.33) implies that, for non-dielectric medium, the network wavelength λ_n actually represents an effective wavelength of $\sqrt{2}\lambda$ and, similarly, the network conductivity σ_n represents effectively a conductivity of $\sqrt{2}\sigma$.

The introduction of conducting materials brings another limiting factor as the relationship between the attenuation constant of the network and that of the medium, given in equation (4.31), is only valid when

$$\alpha\Delta l \ll 1. \tag{4.34}$$

Replacing α by $G_0/\left(4\Delta l(1+\frac{Y_0}{4})\right)$ and G_0 by $\sigma\Delta l\sqrt{\mu_0/\epsilon_0}$ in equation (4.34),

$$\alpha\Delta l \cong \frac{\sigma\Delta l \cdot 120\pi}{4.(1+\frac{Y_0}{4})} \leq 0.1 \tag{4.35}$$

we obtain, in a non-dielectric medium ($Y_0 = 0$),

$$\Delta l \leq (300\pi\sigma)^{-1}. \tag{4.36}$$

Combining the limiting factor (4.36) with the fact that the relationship between the propagation constant of the network and of the medium, equation (4.30), is valid for non-dielectric medium when (see figure 4.8)

$$\frac{\Delta l}{\lambda} \leq 0.07, \tag{4.37}$$

one may determine the ranges in which TLM will effectively analyse a diffusion problem.

Now with the upper limit defined in (4.37), we may determine f_{max}, the maximum frequency, for a given conductivity, σ, for which TLM may be effectively used. Thus with the following two limits

$$\Delta l_{max} = (300\pi\sigma)^{-1} \tag{4.38}$$

$$\lambda_{min} = \frac{\Delta l_{max}}{0.07} \tag{4.39}$$

f_{max} is obtained

$$f_{max} = \frac{c_0}{\lambda_{min}} = 21\pi\sigma c_0. \tag{4.40}$$

In Table 4.1, the maximum step size is given as a function of the conductivity, σ. As may be observed, the limiting factor for the step size is the conductivity as opposed to the wavelength when considering a non-dielectric material.

At first glance it might seem contradictory that as the conductivity increases, the maximum frequency also increases. But this is an illusion since as the conductivity increases, the maximum step size decreases at the same rate as the maximum frequency increases (see equation (4.39)). When relating the conductivity, the maximum step size and frequency to the skin depth, δ (given in equation (3.25)), we see that

$$\Delta l_{max} \simeq \frac{\delta_{min}}{3.4} = \frac{1}{3.4\sqrt{\pi f_{max}\mu\sigma}} \tag{4.41}$$

Therefore in the limiting case there are approximately four steps per skin depth.

Table 4.1: Calculating the maximum step size as a function of σ.

Conductivity (S/m)	Maximum step[1] size $\Delta l(\mu m)$	Maximum frequency[2] f_{max}(GHz)	Skin depth in (μm) for σ & f_{max}	Typical frequencies- # iterations/cycle[3]	
				1GHz	10GHz
10	106.1	197.9	357.8	2830	283
20	53.0	395.8	178.9	5660	566
50	21.1	989.9	71.5	14151	1415
100	10.6	1979.0	35.8	28300	2830
200	5.3	3958.0	17.9	56600	5660
500	2.1	⋮	⋮	⋮	⋮
1E03	1.1	⋮	⋮	⋮	⋮
1E05	10.6E-03	⋮	⋮	⋮	⋮
1E08	10.6E-06	⋮	⋮	⋮	⋮

1 for $Y_0 = 0$
2 the maximum frequency (f_{max}) is determined from equation (4.40) once Δl is found from (4.38)
3 # of iterations = $\frac{c_0}{\Delta l \times f}$

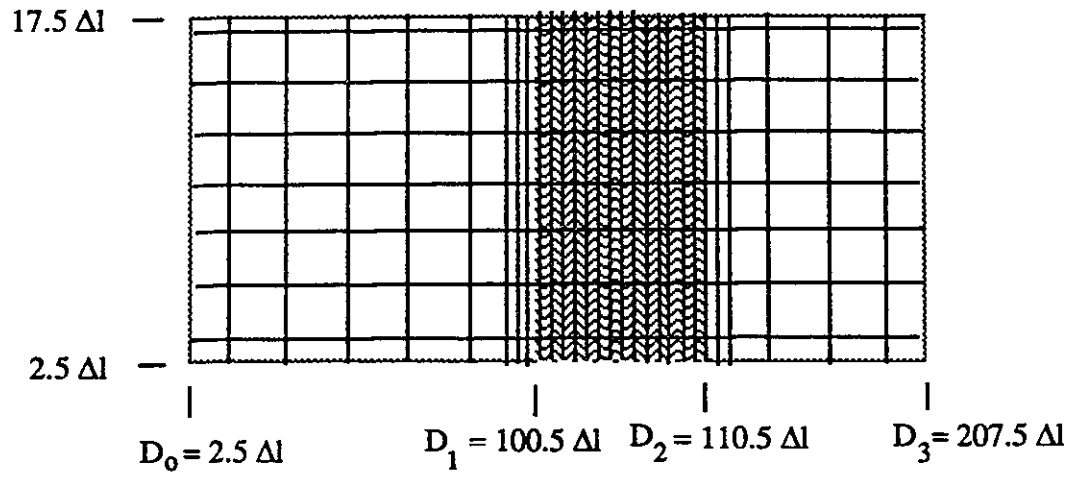


Figure 4.9: TLM division of the one-dimensional problem.

4.4 Validation

The validation of the TLM method, used to analyse diffusion problems, is conducted in the frequency domain for it yields more information. Since TLM is a time domain numerical technique, the result of the analysis will be converted to the frequency domain via the Fourier transform. The effect that both the total number of iterations and the mesh size, Δl, have on the numerical results of TLM are compared to the analytical solution derived in section 2.1.2.

The one-dimensional test problem (see figure 2.1) is discretized into interconnecting transmission lines as shown in figure 4.9. The total region analysed is approximately 210 Δl long by 20 Δl high. Using variable mesh size the region is divided into steps of Δl between 93 Δl and 117 Δl, the rest is divided into steps of 5 Δl. As a consequence, for an impulse to travel from $x = D_3$ to $x = D_0$ and back, 400 iterations are needed.

4.4.1 Number of Iterations Needed for Steady-State

The number of iterations required to reach a steady-state for a problem depends on the size of both the step and the region under investigation as a function of the

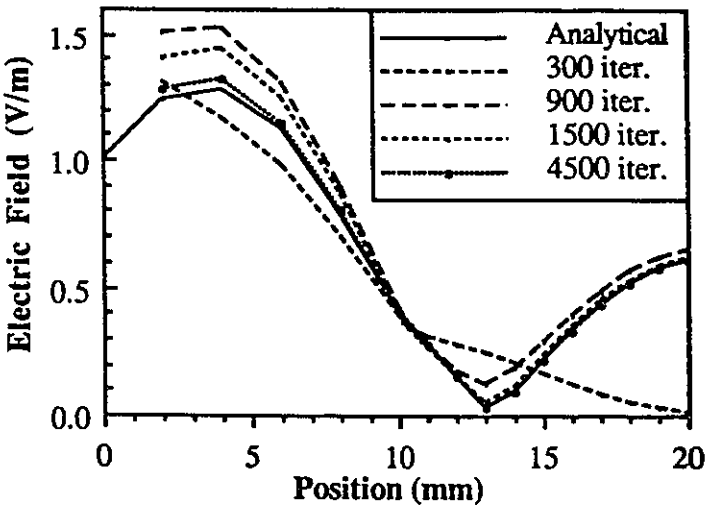


Figure 4.10: TLM vs analytical solution as a function of iteration.

frequencies of interest. To show what effect the number of iterations considered has on the steady-state results, a one-dimensional diffusion problem was solved for various number of iterations and compared to the analytical solution as shown in figure 4.10. The problem geometry and parameters are

$$D_0 = 0.25mm \quad ; \quad D_1 = 10.05mm \quad ; \quad D_2 = 11.05mm \quad ; \quad D_3 = 20.75mm$$

$$f \;=\; 10GHz \quad ; \quad \sigma = 10 \quad ; \quad \Delta l = 100\mu m$$

$$\#\, of\ iterations\ for\ 1\ cycle = \frac{c_0}{f\Delta l} = 300$$

It is seen that the results obtained with 300 iterations is far from being accurate, for two reasons:

- only one full cycle was considered which isn't enough for a steady-state solution.
- also it takes 400 iterations for one impulse to travel to D_3 and back

Although one cycle isn't enough, TLM converges very rapidly; with only three cycles considered, already TLM results are close to the signature, and with 15 cycles, the results are in good agreement.

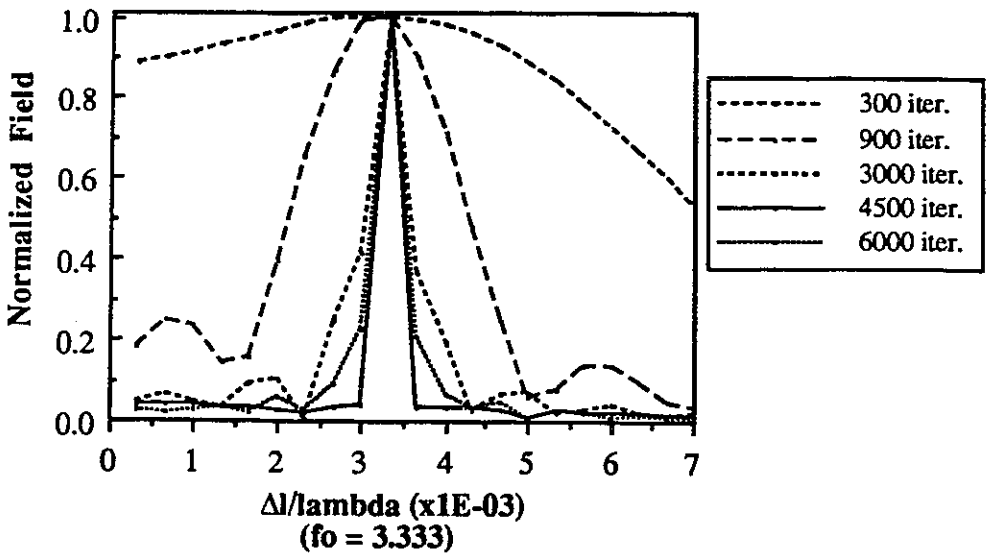


Figure 4.11: Frequency plot of TLM vs analytical solution as a function of iteration.

The implications of the number of iterations considered can also be analysed by plotting the Fourier transform of the signal, at a certain position, for various iterations. In figure 4.11 the convergence to the frequency of interest as a function of the number of iterations is very well demonstrated. It should be noted that there is an optimum number of iterations since, as one increases this number beyond a certain point, the errors from the imperfect terminations become more and more apparent. For this particular situation, fifteen cycles seem to be an optimum.

4.4.2 Effect of Varying the Length of the Step Size

In section 4.3 it was shown that the determining factor for the step size is mainly the conductivity and not the wavelength in a non-dielectric region. The step size Δl will now be increased up to a breaking point such as to verify the presented rule of thumb relating the conductivity to the step size

$$\alpha \Delta l \leq 0.1 \quad \underline{or} \quad \Delta l_{max} = (300\pi\sigma)^{-1}. \tag{4.42}$$

In figures 4.12(a) & (b), the analytical solution is compared to TLM's for various step sizes. In each case the geometry and parameters are as follow:

$$D_0 = 2.5\Delta l \;\; ; \;\; D_1 = 100.5\Delta l \;\; ; \;\; D_2 = 110.5\Delta l \;\; ; \;\; D_3 = 207.5\Delta l$$

$$f = 10GHz \;\; ; \;\; \sigma = 10 \;\; ; \;\; \#\, of\ iterations = 15 \times [\#\, of\ iter./cycle]$$

As the parameter $\alpha\Delta l$ increases from 0.05 to 0.1, there is a very good correlation between TLM and the analytical solution. The TLM results are still in good order for $\alpha\Delta l$ near 0.2, but as $\alpha\Delta l$ increases beyond 0.3, the TLM method cannot follow the field variations anymore. This breakdown phenomenon is well observed with $\alpha\Delta l$ equal to 0.5.

4.5 Discussion

In section 4.3, some rules of thumb regarding modeling of diffusion problems via the TLM method were derived through a one-dimensional analysis. These rules revealed to be very useful in understanding the limits imposed by TLM and in knowing, prior to any analysis, the projected accuracy of the model.

The validation of the TLM method was carried out in the frequency domain because it yields more useful information. The convergence of the technique was demonstrated by increasing the number of iterations considered. It was seen with the comparison to the analytical solution and also with the fourier transform that the convergence is fairly rapid (about 15 cycles were needed to reach steady state) and that there is an optimum number of iterations. The effect of the cutoff of the network is well demonstrated in section 4.4.2 by varying the length of the step size.

The TLM method may now be applied to more complex diffusion problems as its range of operation has been determined.

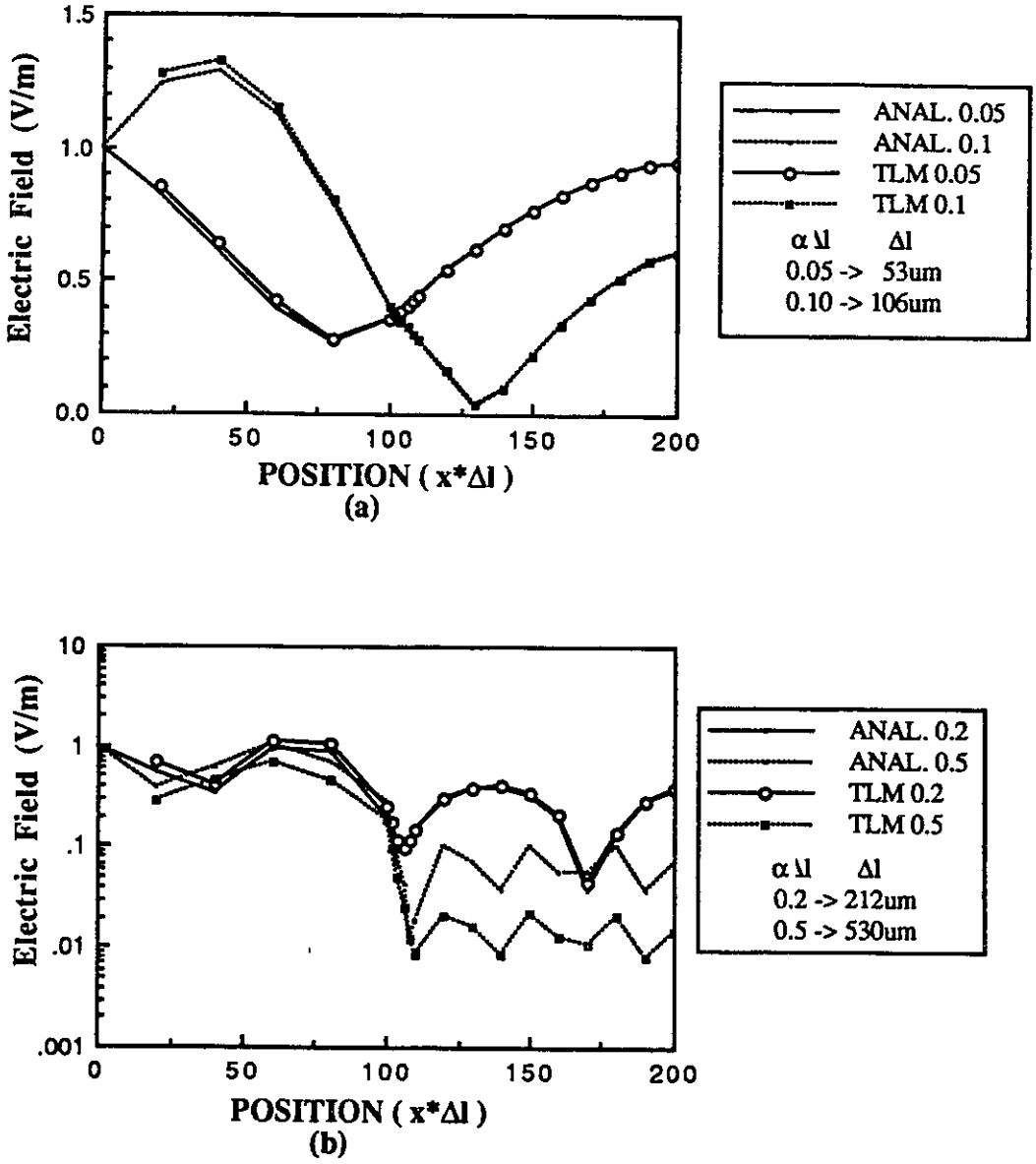


Figure 4.12: Analytical solution vs TLM as a function of Δl.

Chapter 5

Application of FEM to General Problems

5.1 Introduction

In chapter 2, an analytical solution was derived for the diffusion problem in both time and frequency domain. The availability of this theoretical formulation has given us the opportunity to validate both numerical techniques, the FEM and TLM, in solving diffusion problems. It also provided us with means to determine the range of operation of the techniques for following the field penetration into a conducting material.

Having full confidence in the FEM and the TLM method for solving the one-dimensional diffusion problems, we can apply both numerical techniques to more general and realistic problems. The actual enclosures used as shields, for damaging fields, are comprised of conducting materials as well as apertures. Since the shielded equipment has to communicate with the outside world, apertures (or holes) are needed for the link and also they are needed to dissipate the heat generated by the electrical equipment. Now to obtain an analytical solution of field penetration through apertures would require a very complicated formulation, if one exists. Thus to validate

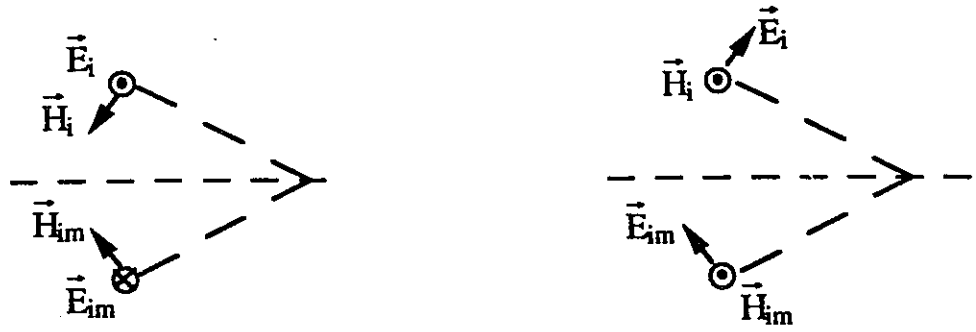
the presented techniques, the same problem will be solved by these two completely different numerical methods.

There are two distinct situations to analyse when talking about the shielding effectiveness of enclosures. In one case, the shield is needed to prevent the damaging field from penetrating while in the other, the shield is intended to confine the field within the enclosure. The first situation is refered to as the immunity case, in which impinging fields are to be reflected and attenuated by the enclosures such as to minimize the effect of the ambient fields on the electronic equipment (see figure 1.1(a)). In the second situation, the emission case, fields generated by the electronic equipment are constrained within the enclosure as much as possible to prevent the emitted fields from corrupting nearby susceptible apparatus (see figure 1.1(b)).

In the following, assumptions are introduced to simplify the problem which in turn reduces the CPU time required for the calculation of the solution; they are by no means an indication of the restrictions imposed by the methods. The most important assumption made in this analysis is as follows: "*the length of the enclosure is assumed to be much longer than its width or height*". This enables a reduction of a three-dimensional problem into a two-dimensional one thereby alleviating the complexity of the situation without sacrificing too much the generality of the analysis. Although only two-dimensional problems are considered here, three-dimensional studies are feasible with FEM [38] and TLM [39] as these numerical methods have been extended to include that type of investigation. Another assumption made in this thesis is with respect to the ground plane: "*the conductivity of the ground plane, over which lies the shield, is assumed to be infinite*". Thus no loss is accounted for within the ground.

5.2 Immunity Case

The analysis of the immunity of a system, assured by an enclosure, is crucial as the system's performance may deteriorate due to the ambient fields. To properly



(a) TE mode (b) TM mode

Figure 5.1: Mirror image concept for (a)TE and (b) TM modes of propagating plane waves.

characterize the immunity via a CAD tool, it has to solve problems as realistic as possible. Therefore both the material's conductivity and the aperture geometries have to be included as parameters along with the geometry of the enclosure.

The enclosure, in the immunity analysis, is lying over a perfectly conducting ground plane ($\sigma \equiv \infty$), which may be taken into consideration by making use of the mirror image concept. Figure 5.1 shows the effect of the ground plane on the region's fields when it is replaced by the image concept. The field at a certain point is given as the superposition of the impinging field and its image. Thus for the TE case, the electric field is set equal to zero everywhere on the ground plane at any time t; whereas for the TM case, the magnetic field is set to be twice that of the field directly above the ground plane ($y = 0^+$) which is dependent on the time t and its position. For the TM case, the ground plane may also be replaced by a homogeneous Neumann boundary since the magnetic field has no variations with respect to the normal of the ground plane near the plane [24]. This will facilitate somewhat the code. In this type of analysis, the ambient field is assumed to impinge uniformly on two sides of the enclosure. With this assumption, a Dirichlet boundary may be set at a certain distance from the enclosure (see figure 5.2) and also at a distance far enough over the enclosure, the field may be assumed invariant with respect to the normal $\left(\frac{\partial \vec{H}}{\partial \hat{n}} = 0\right)$.

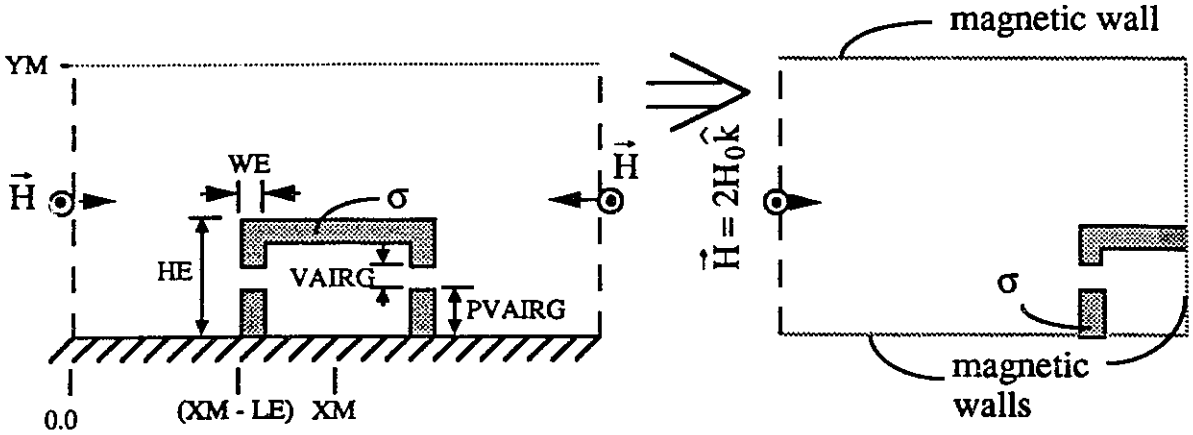


Figure 5.2: Immunity problem under investigation.

5.2.1 Mathematical Formulation

In the previous two chapters, we have solved the field penetration of an incident TE field. Here we shall predict the shielding effectiveness of enclosures to TM impinging fields thus showing the flexibility of both techniques. The mathematical formulation for solving TM fields with FEM and TLM is very similar to the one derived in chapters 3 and 4 respectively, since the wave equations are the same:

$$\nabla^2\vec{e}(t) - \mu\sigma\frac{\partial}{\partial t}\vec{e}(t) - \mu\epsilon\frac{\partial^2}{\partial t^2}\vec{e}(t) = 0 \tag{5.1}$$

$$\nabla^2\vec{h}(t) - \mu\sigma\frac{\partial}{\partial t}\vec{h}(t) - \mu\epsilon\frac{\partial^2}{\partial t^2}\vec{h}(t) = 0. \tag{5.2}$$

The formulation in the FEM is exactly the same as the one derived in section 3.2 as it was given in a very general form, one only needs to replace the general unknown $\vec{g}(t)$ by $\vec{h}(t)$. The variation is introduced by the boundary conditions.

Now in the TLM method, the mapping of the field quantities to the voltages and currents in the transmission lines is obtained from the following sets of equations for the TM case [37],

$$-\frac{\partial}{\partial x}(I_{x_1} - I_{x_3}) - \frac{\partial}{\partial z}(I_{z_2} - I_{z_4}) = 2C(1 + Y_0/4)\frac{\partial V_y}{\partial t} \tag{5.3}$$

$$-\frac{\partial V_y}{\partial x} = L\frac{\partial}{\partial t}(I_{x_1} - I_{x_3}) + \frac{G_0 C c_0}{\Delta l}(I_{x_1} - I_{x_3}) \tag{5.4}$$

$$-\frac{\partial V_y}{\partial z} = L\frac{\partial}{\partial t}(I_{z_2} - I_{z_4}) + \frac{G_0 C c_0}{\Delta l}(I_{z_2} - I_{z_4}) \tag{5.5}$$

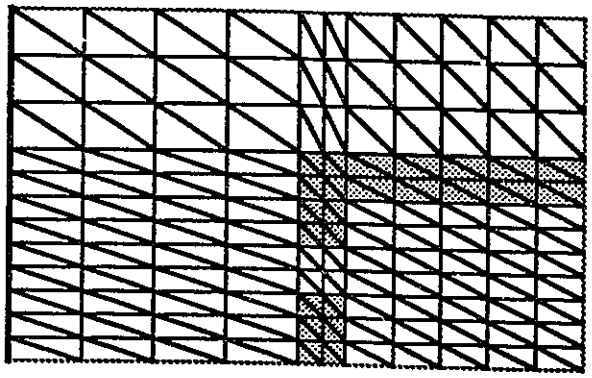


Figure 5.3: Finite element division (immunity).

for the transmission lines and

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t} \tag{5.6}$$

$$\frac{\partial H_y}{\partial z} = -\epsilon \frac{\partial E_x}{\partial t} - \sigma E_x \tag{5.7}$$

$$\frac{\partial H_y}{\partial x} = \epsilon \frac{\partial E_z}{\partial t} + \sigma E_z \tag{5.8}$$

for the field quantities. The mapping then becomes

$$H_y \equiv V_y \;\; ; \;\; E_z \equiv -(I_{x_1} - I_{x_3}) \;\; ; \;\; E_x \equiv (I_{z_2} - I_{z_4})$$

$$\mu_0 \equiv 2C \;\; ; \;\; \mu_r \equiv 1 + Y_0/4 \;\; ; \;\; \epsilon_0 \equiv L \;\; ; \;\; \sigma \equiv \frac{G_0 C c_0}{\Delta l} \equiv \frac{G_0}{\Delta l}\sqrt{\frac{\epsilon_0}{\mu_0}}$$

which is reciprocal to the TE case (see chapter 4). In this particular mapping, a voltage travelling in the transmission lines will represent a propagating magnetic field. Also, materials with a relative permeability different from 1 may be considered by introducing $\Delta l/2$ open shunt stubs at the proper nodes.

The discretization of the region into finite elements and transmission lines is given in figures 5.3 and 5.4 respectively along with the appropriate boundary conditions.

5.2.2 Frequency Domain Comparison Between FEM and TLM

In the TLM formulation some rules of thumb regarding the possible diffusion problems to analyse were presented. In Table 4.1, the maximum step size is depicted for

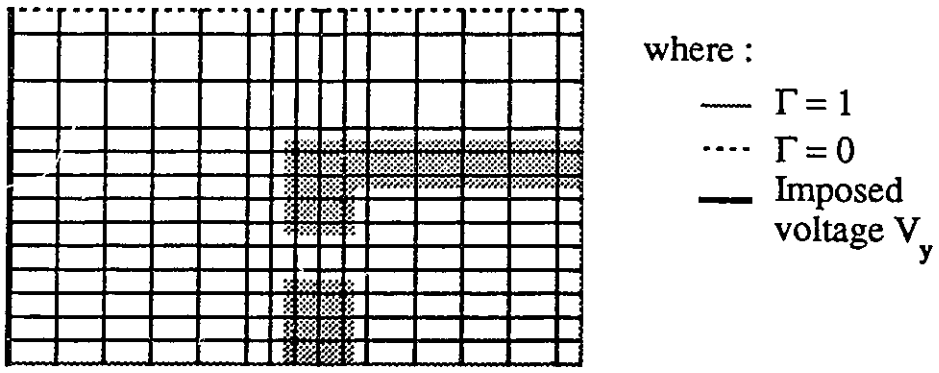


Figure 5.4: TLM division (immunity).

various conductivity and calculated also are the maximum frequency of analysis and its associated skin depth as well as some typical frequencies of analysis with the corresponding number of iterations they require to complete one cycle. It is then seen that in order to investigate diffusion problems in a realistic manner, the best combination of conductivity, wall thickness, and frequency is a shield with a wall thickness in the order of millimeters with conductivity of around $10S/m$ on which a field is impinging at the frequency greater than $1GHz$.

With these considerations in mind, the immunity problem was solved via the two techniques, in the frequency domain, for the following parameters (see figure 5.2)

$$\begin{aligned} LE = 11.2mm \quad &; \quad WE = 2.0mm \quad ; \quad HE = 19.75mm \\ XM = 21.75mm \quad &; \quad YM = 49.75mm \\ PVAIRG = 9.75mm \quad &; \quad VAIRG = 5mm \\ f = 10.0GHz \quad &; \quad \sigma = 10.0 \end{aligned}$$

$$TLM \left\{ \begin{array}{c} \#ofiterations = 14 \times [\#ofiter./cycle] \\ \Delta l = 100\mu m \end{array} \right\}.$$

The comparison between the FEM and TLM is presented in figure 5.5 where the magnitude of the magnetic field is given at $y = 12.25mm$ (through the middle of the aperture) for x varying from $1mm$ to $21.75mm$. As may be observed, there is a very good agreement between both methods.

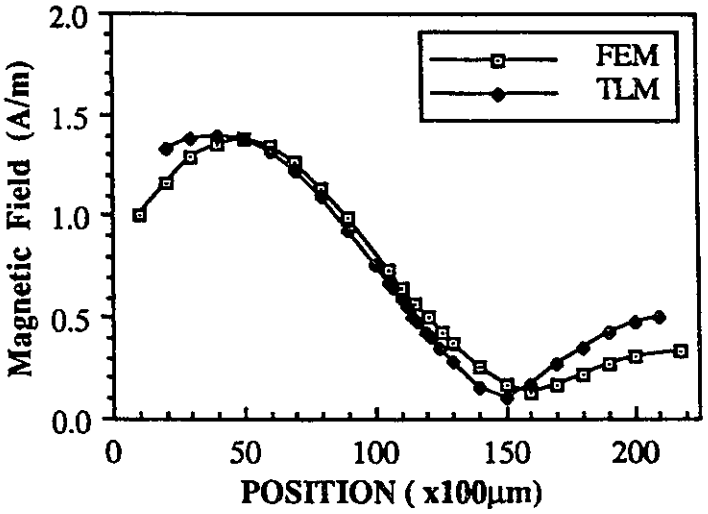


Figure 5.5: Frequency domain comparison between FEM and TLM, immunity case.

5.2.3 Time Domain Comparison Between FEM and TLM

To ensure that the TLM method be able to follow the field variations through the lossy material, we will limit the "maximum frequency component" of the EMP type impinging field to $50GHz$. The period of a $50GHz$ signal is $T|_{50GHz} = 20psec$, therefore by setting the time, $t\prime$, at which the peak of the EMP occurs to $t\prime = T|_{50GHz}$, we make certain that most of the energy will be below $15GHz$ (about one quarter of $50GHz$).

The value of α and β are determined by taking the first derivative of equation (2.11) with respect to time and setting it equal to zero; from this the time at which the peak occurs may be determined

$$t\prime = \frac{1}{(\beta - \alpha)} ln\left(\frac{\beta}{\alpha}\right) = T|_{50GHz}. \tag{5.9}$$

Thus α and β are obtained by setting one to a certain value and then determining the corresponding value of the other. In this analysis the value of α and β were chosen as

$$\alpha = 10^9 \ \ \& \ \ \beta = 3 \times 10^{11}.$$

When using these values, the exact peak time $t\prime$ is calculated to be

$$t\prime = 19.08psec.$$

The minimum number of iterations needed to properly characterize the situation is then determined. Ideally one would like to investigate the effect of the pulse until it dies out, but this isn't feasible. Therefore we shall analyse the problem from the initial time to the time at which the amplitude of the field is equal to or lower than one tenth of the peak pulse value, this will give a final time of

$$t_{fin} = 120t\prime. \tag{5.10}$$

Also since the time step of TLM, Δt, is equal to the smallest step size, Δl, divided by the speed of light, c_0, it is then directly related to the conductivity (see equations (4.38) - (4.40)). Therefore to obtain a practical number of iterations to complete the analysis, the conductivity of the material will be chosen to be equal to or lower than $5S/m$ (see Table 4.1). By choosing a conductivity of $5S/m$, the maximum step size (for TLM), Δl_{max}, then becomes approximately $200\mu m$, and this in turn gives us a maximum time step of $0.67psec$. Therefore to follow the pulse up to $t = 120t\prime$ we need:

$$\frac{120 \times 19.08psec}{0.67psec} \simeq 3400\ iterations.$$

With this in mind, the time domain immunity analysis is solved for the following parameters (refer to figure 5.2)

$$\begin{array}{ccccc} LE = 22.4mm & ; & WE = 4.0mm & ; & HE = 39.75mm \\ XM = 43.5mm & ; & YM = 99.5mm & ; & \alpha = 10^9 \\ PVAIRG = 19.5mm & ; & VAIRG = 10mm & ; & \beta = 3 \times 10^{11} \end{array}$$

$$TLM\{\#ofiterations = 3400; \Delta l = 200\mu m\}.$$

In figures 5.6&5.7, the comparison between FEM and TLM is presented for the magnitude of the magnetic field as a function of time taken at two positions inside the enclosure, $(x = 25mm, y = 25mm)$ and $(x = 38mm, y = 25mm)$ respectively.

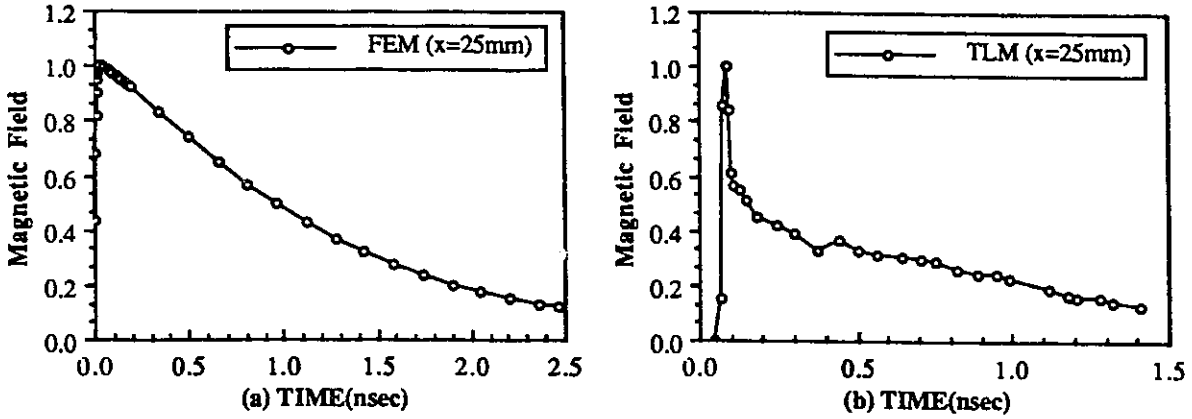


Figure 5.6: Time domain comparison between FEM and TLM for $x = y = 25mm$, immunity case.

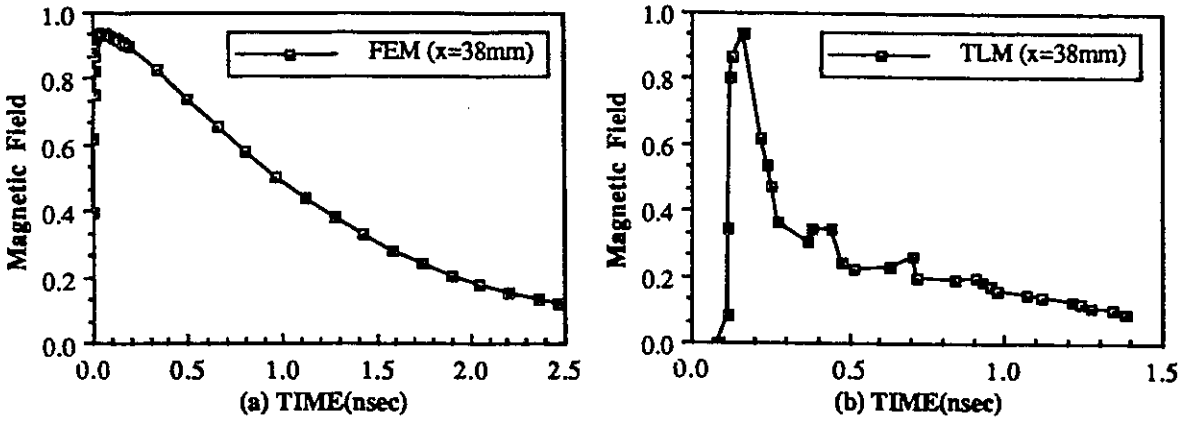


Figure 5.7: Time domain comparison between FEM and TLM for $x = 38mm$; $y = 25mm$, immunity case.

When comparing the results of both techniques one should keep in mind that in TLM, the frequency components lower than $1GHz$ $\left(f = \frac{c_0}{\Delta l * \#iterations}\right)$ do not contribute to the results since not enough iterations are considered for these frequencies. The effect that TLM has on the time response is observed in figures 5.6&5.7, since the lower frequency components do not figure in the results (too much iterations would be needed to include them), it lowers the overall field penetration.

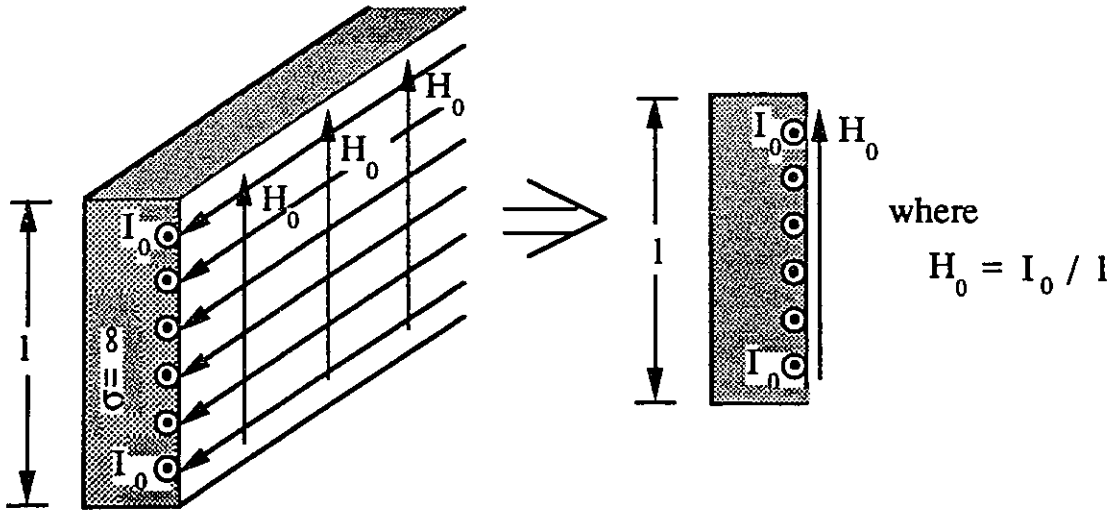


Figure 5.8: Constant magnetic field on a back plane.

5.3 Emission Case

Electromagnetic interference (EMI) caused by radiating electronic instruments is potentially harmful to nearby equipment. It is then of great importance to predict the shielding effectiveness of an enclosure used for confining fields.

In this section, the effect of the radiation from an electronic equipment surrounded by an enclosure with apertures is analysed via FEM and TLM. The comparison of both numerical techniques will validate the results of the analysis.

As the radiations from electronic devices is due to the fields associated with the currents flowing in a conductor or a back plane, for example, the analysis will be carried out through the magnetic field, $\vec{H}$. Now to be able to relate the conditions in both numerical techniques, the source current will be assumed to flow on the ground plane of a microstrip. The current will also be assumed to be uniformly distributed on the plane thus creating a constant magnetic field very close to the ground plane (see figure 5.8) [41]. To reduce memory requirements as well as the processing time, we have solved for two back planes in one enclosure so as to obtain a symmetry with respect to one axis as shown in figure 5.9. This will reduce the region to be analysed by one half and the processing time by at least one fourth. Furthermore, the thickness

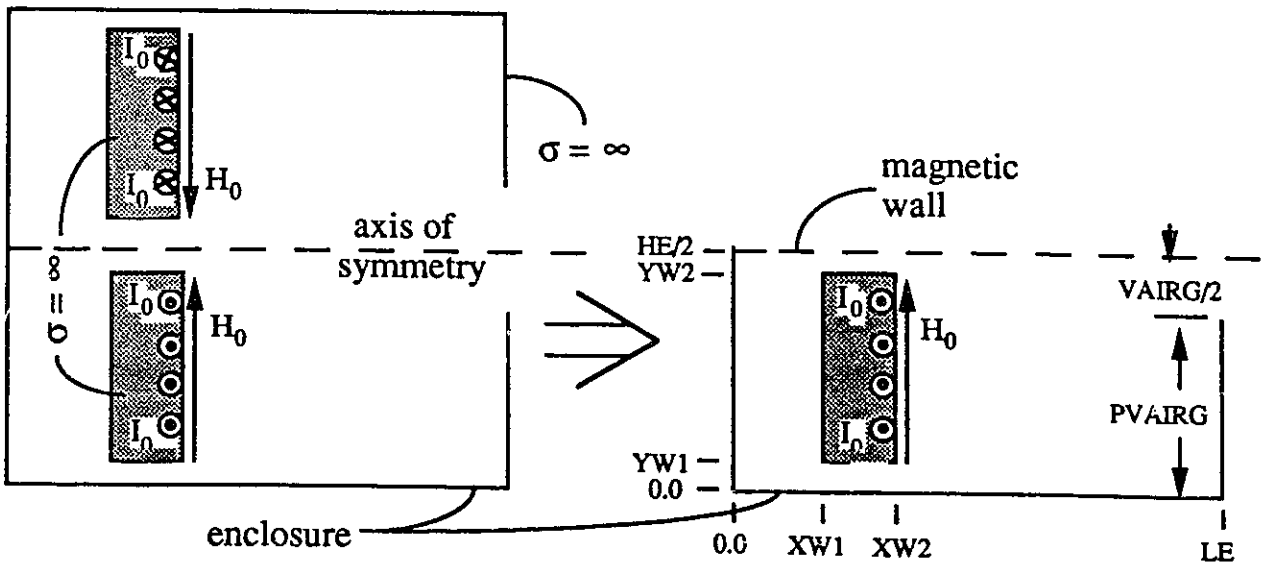


Figure 5.9: Emission problem under investigation.

and material of the enclosure's wall will be chosen in such a way that the diffusion process may be neglected, i.e. the skin depth is chosen to be much smaller than the wall thickness of the enclosure. The diffusion process is neglected in this section for two reasons:

- First to reduce the region to analyse for TLM
- Second, and more importantly, to demonstrate the versatility of both techniques in solving various situations.

5.3.1 Mathematical Formulation

TLM Formulation

The TLM formulation is the same as given in chapter 4. However, since we are interested in the magnetic fields rather than the electric fields, the boundary conditions associated with the source field will be imposed differently. For the shunt network

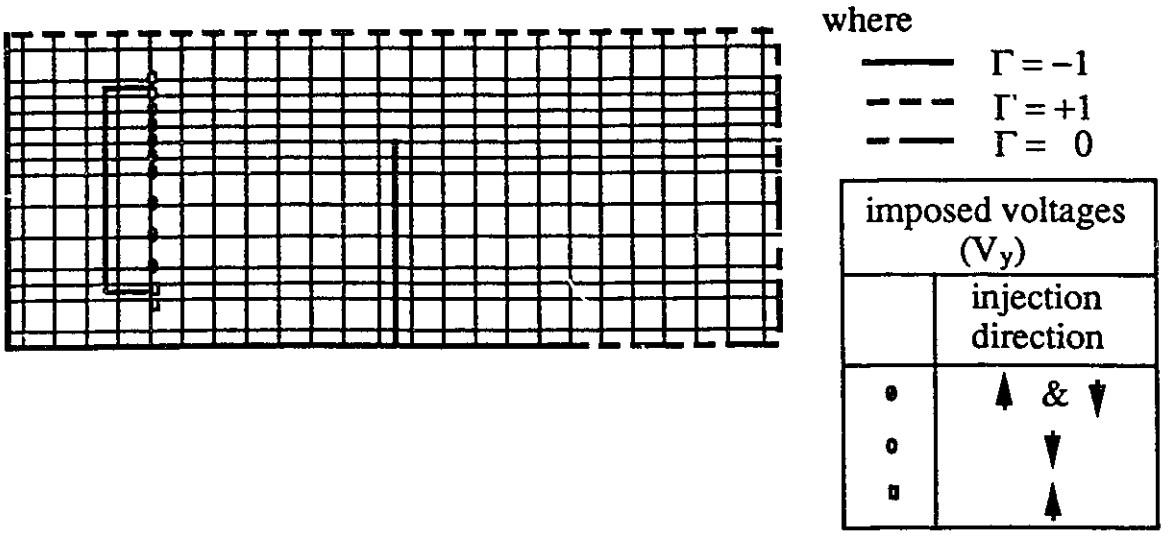


Figure 5.10: TLM mesh division for the emission study.

given in chapter 4, the fields are calculated as follows [42] (refer to figure 4.4):

$$ {}_kE_y \equiv {}_kV_y = \frac{2}{Y}\left(\sum_{m=1}^{4} {}_kV_m^i + {}_kV_5^i Y_0\right) \tag{5.11} $$

$$ -{}_kH_x \equiv {}_kI_z = ({}_kV_2^i - {}_kV_4^i)/Z_0 \tag{5.12} $$

$$ {}_kH_z \equiv {}_kI_z = ({}_kV_1^i - {}_kV_3^i)/Z_0 \tag{5.13} $$

where $Y = 4 + Y_0$ and $Z_0 = \sqrt{L/C} = \sqrt{\mu_0/\epsilon_0}$ is the characteristic impedance of the lines. Therefore to create a constant magnetic field in the x-direction for example, one could set the incident voltages V_1 & V_3 at time $k * \Delta t$ to the same value, thus setting ${}_kH_z = 0$ and ${}_kH_x \neq 0$. This indirect approach is a very effective way of controlling the magnetic field in order to present a Dirichlet boundary condition in the TE formulation.

The magnetic walls are introduced to the incident voltage V_y by a reflection coefficient of $\Gamma = 1$ and the perfectly conducting walls by $\Gamma = -1$ [42]. As this is an unbounded problem, the transmission lines in the outside region may be terminated by loads matched to the characteristic impedance of air as seen by the lines, thus $\Gamma = 0$. The emission problem as given in figure 5.9 is represented in TLM, with the boundary conditions, as shown in figure 5.10.

FEM Formulation

This type of analysis will generate two magnetic field components, the horizontal and the vertical. Solving with FEM directly for the magnetic field, $\vec{h}$, would then require the solution of a system of equations that is proportional to twice the total number of nodes as there are two field components per node, i.e. two unknowns per node. This increases both the memory requirements and the CPU time. But solving for the magnetic vector potential, $\vec{a}$, one may reduce the number of unknowns by one half since there is only one component to solve for each node [43].

The magnetic vector potential, $\vec{a}$, is related to the magnetic flux density, $\vec{b}$, by

$$\vec{b}(t) = \nabla \times \vec{a}(t) \tag{5.14}$$

since the divergence of $\vec{b}$ is always zero and the divergence of the curl of any vector is also zero. Substituting equation 5.14 into Faraday's law,

$$\nabla \times \vec{e}(t) = -\frac{\partial}{\partial t}\vec{b}(t) = -\nabla \times \left(\frac{\partial}{\partial t}\vec{a}(t)\right), \tag{5.15}$$

the following equivalence is obtained

$$\vec{e}(t) = -\frac{\partial}{\partial t}\vec{a}(t) - \nabla\phi \tag{5.16}$$

since the curl of the gradient of any scalar is always zero. Now to determine the characteristic equation associated with the vector and scalar potentials we replace the electric field and the flux density in the following equation

$$\nabla \times \vec{b}(t) = \mu\sigma\vec{e}(t) + \mu\epsilon\frac{\partial}{\partial t}\vec{e} \tag{5.17}$$

by equations (5.16) and (5.15) respectively, which gives

$$\nabla \times (\nabla \times \vec{a}(t)) = (\mu\sigma + \mu\epsilon\frac{\partial}{\partial t})(-\frac{\partial}{\partial t}\vec{a}(t) - \nabla\phi) \tag{5.18}$$

$$\nabla^2\vec{a}(t) - \mu\epsilon\frac{\partial^2}{\partial t^2}\vec{a}(t) - \mu\sigma\frac{\partial}{\partial t}\vec{a}(t) = \nabla \cdot (\nabla\vec{a}(t) + \mu\sigma\phi + \mu\epsilon\frac{\partial}{\partial t}\phi). \tag{5.19}$$

Equation (5.19) is very complicated as such, but by chocsing the divergence of $\vec{a}$ appropriately, it may be simplified considerably. A popular choice for the divergence of $\vec{a}$ is the so-called Lorentz condition

$$\nabla\vec{a}(t) = -\mu\epsilon\frac{\partial}{\partial t}\phi. \tag{5.20}$$

But by choosing this, we are still left with a term on the right hand side of equation (5.19), which complicates the analysis. Thus an appropriate choice, in our case, would be the following [43],

$$\nabla\vec{a}(t) = -\mu\sigma\phi - \mu\epsilon\frac{\partial}{\partial t}\phi \tag{5.21}$$

which reduces equation (5.19) to the wave equation

$$\nabla^2\vec{a}(t) - \mu\sigma\frac{\partial}{\partial t}\vec{a}(t) - \mu\epsilon\frac{\partial^2}{\partial t^2}\vec{a}(t) = 0. \tag{5.22}$$

By using this characteristic equation, only the boundary conditions need to be modified and not the core of the FEM formulation as presented in chapter 3.

The magnetic vector potentials, as defined in (5.14), is normal to the magnetic field, i.e. parallel to any flow of current. Since the current in the back plane is assumed to flow in the z-direction only, the magnetic vector potential will then have but one component directed in the z-axis. The relationship between these two magnetic "quantities" is as follows

$$\vec{h}(t) = \frac{1}{\mu}\nabla \times (a_z(t)\hat{z}) = \frac{1}{\mu}\left[\hat{x}\frac{\partial}{\partial y}a_z(t) - \hat{y}\frac{\partial}{\partial x}a_z(t)\right] \tag{5.23}$$

for a one-component formulation of $\vec{a}$.

The source field $h_0(t)$, a Dirichlet boundary, is then mapped to an inhomogeneous Neumann boundary in the magnetic vector formulation, given as

$$\frac{\partial\vec{a}(t)}{\partial\hat{n}} = -\mu_0 h_0(t)\hat{z} \tag{5.24}$$

where the normal $\hat{n}$ is pointing outside of the region. This boundary condition is incorporated in the final system of equations through the surface integral matrix, U,

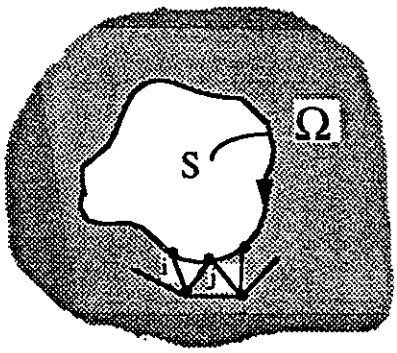


Figure 5.11: Inhomogeneous Neumann boundary.

given in equation (3.24). The shape function used to approximate the field on the surface is a contour type of function with the value a_i on node i and a_j on j [44]

$$\vec{a}(s,t) = \hat{z}\left(a_i(t)\frac{s_j - s}{s_j - s_i} + a_j(t)\frac{s_i - s}{s_i - s_j}\right) \tag{5.25}$$

in which s is a contour variable and s_i & s_j are the node positions of the segment l (see figure 5.11). The surface integral matrix, U, is then obtained as follows

$$\begin{aligned} U_{ij} &= \frac{\partial}{\partial a_i(t)}\int_{s_i}^{s_j}\left[2\cdot\left(a_i(t)\frac{s_j - s}{s_j - s_i} + a_j(t)\frac{s_i - s}{s_i - s_j}\right)\hat{z}\cdot(-\mu_0 h_0(t)\hat{z})\right]ds \\ &= -2\mu_0 h_0(t)\int_{s_i}^{s_j}\left(\frac{s_j - s}{s_j - s_i}\right)ds \\ &= -2\mu_0 h_0(t)\frac{(s_j - s_i)}{2} \end{aligned} \tag{5.26}$$

and similarly

$$U_{ji} = 2\mu_0 h_0(t)\frac{(s_i - s_j)}{2}. \tag{5.27}$$

Now since the region analysed is unbounded, a special type of element will be needed to somehow bound the region. One such type of elements are the infinite elements first introduced by Ungless [45] as reciprocal decay elements. But since Pissanetzky [46] has brought forward a simple and effective infinite element formulation, it will be chosen for this analysis.

The infinite elements, seen in figure 5.12, have two nodes on the boundary (i&j) and two nodes at infinity. To make the analysis more attractive and simpler, a new

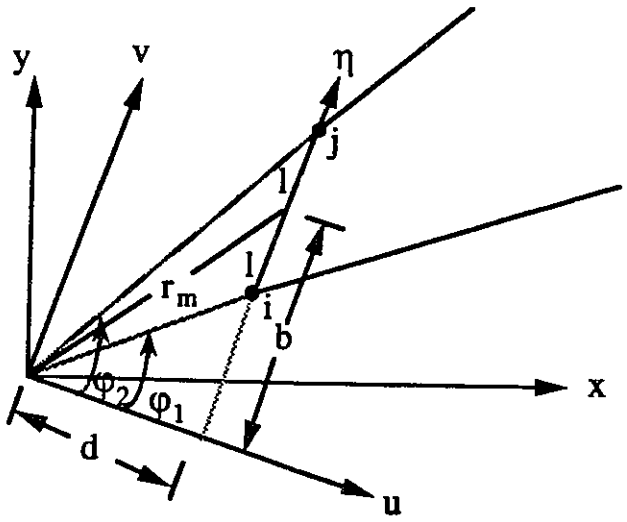


Figure 5.12: The infinite element.

coordinate system is chosen for each element. This rectangular coordinate system (u, v) has the v-axis parallel to the segment in question $(i - j)$. The angle ϕ is measured counter-clockwise from the u-axis and the length of the segment $i - j$ is $2l$.

The shape function in this case will vary linearly along the segment $i - j$ and will have the value of a_i and a_j on nodes i and j respectively

$$\vec{a}_{i-j}(t) = \hat{z}\frac{1}{2}\left[(1-\eta)a_i(t) + (1+\eta)a_j(t)\right] \tag{5.28}$$

where η is a coordinate that varies linearly along $i - j$ given as

$$\eta = \frac{d}{l}tan\phi - \frac{b}{l}. \tag{5.29}$$

The coordinate η will then have the value of -1 when $a = a_i$, and of 1 when $a = a_j$.

The shape function should also decay as a function of r, the distance from the origin, towards infinity. Then, in terms of polar coordinates (r, ϕ), the shape function is given as

$$\vec{a}(r, \phi, t) = \hat{z}\frac{f(r)}{2f(d/cos\phi)}\left[(1-\eta)a_i(t) + (1+\eta)a_j(t)\right] \tag{5.30}$$

in which $d/cos\phi$ is the distance from the origin to a particular point on the segment (when $r = d/cos\phi$, equation (5.30) becomes equivalent to (5.28)).

When considering the case of reciprocal decay, $f(r) = 1/r^n$ with $n \geq 1$, the shape functions become

$$\vec{a}(r,\phi,t) = \hat{z}\frac{d^n}{2r^n cos^n\phi}\left[(1-\eta)a_i(t) + (1+\eta)a_j(t)\right] \tag{5.31}$$

or in terms of the rectangular coordinate system u, v

$$\vec{a}(u,v,t) = \hat{z}\frac{d^n}{2u^n}\left[\left(1+\frac{b}{l}-\frac{dv}{lu}\right)a_i(t) + \left(1-\frac{b}{l}+\frac{dv}{lu}\right)a_j(t)\right]. \tag{5.32}$$

The contribution of the infinite elements to the final system of equations given in (3.24) is with respect to the matrices S and T. The contribution to the S matrix is [46]

$$\begin{aligned} S_{ii} &= \frac{1}{6nd}\left[(4n^2+2n+1)l - 6nb + \frac{3r_m^2}{l}\right] \\ S_{ij} = S_{ji} &= \frac{1}{6nd}\left[(2n^2-2n-1)l - \frac{3r_m^2}{l}\right] \\ S_{jj} &= \frac{1}{6nd}\left[(4n^2+2n+1)l + 6nb + \frac{3r_m^2}{l}\right] \end{aligned} \tag{5.33}$$

and to the T matrix (see Appendix B),

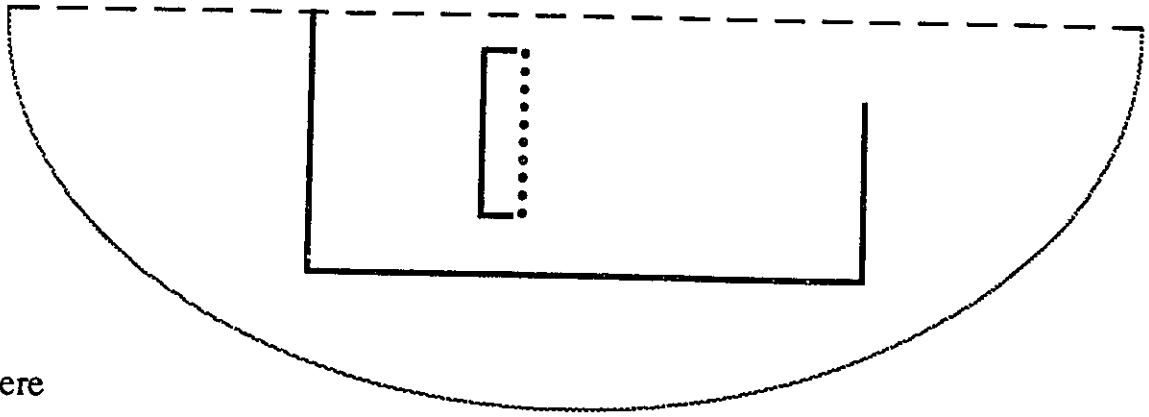
$$\begin{aligned} T_{ii} &= T_{jj} = \frac{2dl}{(3n-3)} \\ T_{ij} &= T_{ji} = \frac{dl}{(3n-3)} \end{aligned} \tag{5.34}$$

where $n \geq 2$.

The magnetic walls are introduced in the FEM by homogeneous Neumann boundary conditions ($\frac{\partial\vec{a}(t)}{\partial\hat{n}} = 0$) and the perfectly conducting walls by a Dirichlet boundary ($\vec{a}(t) = 0$). Therefore the emission problem as given in figure 5.9 is represented in FEM, with the boundary conditions, as shown in figure 5.13

5.3.2 Frequency Domain Comparison Between FEM and TLM

In the emission analysis, the walls of the enclosure are set to a high conductivity such that the skin depth is much lower than the wall thickness. This is readily obtained



where

Dirichlet boundary {a(t) = 0}

Homogeneous Neumann boundary {∂a(t)/∂n = 0}

Inhomogeneous Neumann boundary {∂a(t)/∂n = -μh$_o$(t)}

Infinite elements

Figure 5.13: FEM boundary condition for the emission study.

by choosing an enclosure with a wall thickness of let's say 0.5mm and a conductivity of $0.6 \times 10^8 S/m$ (as for copper material) and with an operating frequency of 1GHz. With these choices, the skin depth is 243 times smaller than the wall thickness, thus a perfectly conducting wall may be assumed.

For such analysis, the limiting factors are again imposed by TLM but this time are determined by the cutoff frequency of the network and the number of iterations needed to reach steady-state. For a non-dielectric region, equation (4.37) determines the frequencies that are below the cutoff point. The precision of the TLM mesh size will be set to 1mm, for which the cutoff frequency is 21GHz. Therefore by choosing the frequency of analysis to be 1GHz we ensure a well posed TLM model.

Refering to figure 5.9, the parameters of the problems are listed below

$$\begin{aligned} LE = 0.2m \quad &; \quad HE/2 = 0.1m \\ XW1 = 0.05m \quad &; \quad XW2 = 0.055m \\ YW1 = 0.05m \quad &; \quad YW2 = 0.087m \\ PVAIRG = 0.075m \quad &; \quad VAIRG/2 = 0.025m \\ f = 1.0GHz & \end{aligned}$$

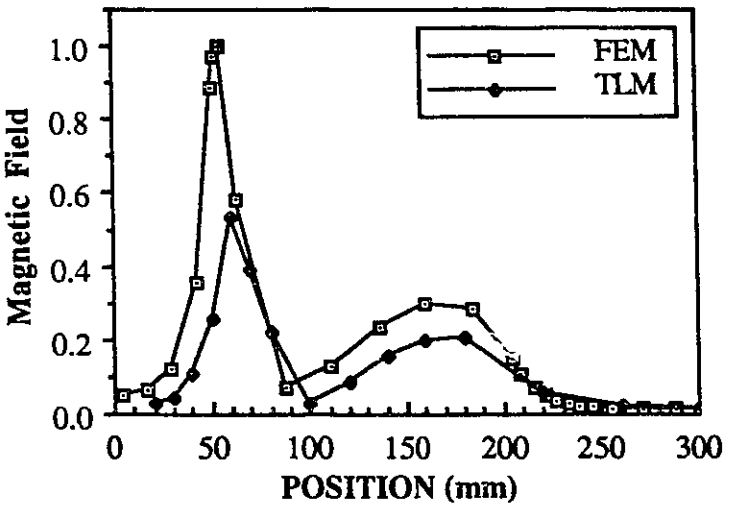


Figure 5.14: Frequency domain comparison between FEM and TLM, emission case.

$$TLM \left\{ \begin{array}{c} \#ofiterations = 15 \times [\#ofiter./cycle] \\ \Delta l = 1.0mm \end{array} \right\}.$$

In figure 5.14, the results of the analysis by both numerical techniques are presented for the magnitude of the magnetic fields as a function of the x-position at $y = 0.1m$ (in the middle of the aperture). Notice how close they compare. This comparison is extremely important as it shows very well the adaptability of both CAD tools in modeling complex boundary conditions such as a field created by a current source.

5.3.3 Time Domain Comparison Between FEM and TLM

The time domain analysis for the emission case is also conducted for an EMP type signal. The reason for choosing such a time signature is mainly for the simplicity of implementation in both tools. The analogy to a real situation could be as follows : an EMP field inducing a current, with a similar time signature, in a power line, finds its way into an improperly filtered enclosure on the back plane of a microstrip, thus radiating and affecting nearby equipment that were initially well filtered.

In this analysis, the current source will have the same time signature as the EMP

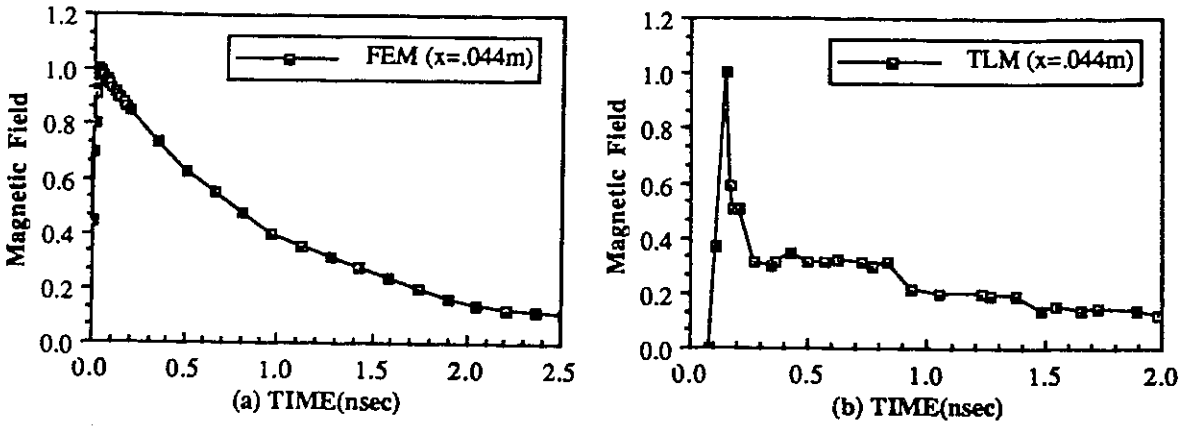


Figure 5.15: Time domain comparison between FEM and TLM, emission case.

type field from section 5.2.3. This current source will therefore generate a magnetic field of the same shape. The problem to investigate in the time domain emission analysis has the following parameters (see figure 5.9)

$$\begin{aligned} LE = 0.04m \quad &; \quad HE/2 = 0.02m \\ XW1 = 0.01m \quad &; \quad XW2 = 0.011m \\ YW1 = 0.01m \quad &; \quad YW2 = 0.0174m \\ PVAIRG = 0.015m \quad &; \quad VAIRG/2 = 0.005m \\ \alpha = 10^9 \quad &; \quad \beta = 3 \times 10^{11} \end{aligned}$$

$$TLM\{\#ofiterations = 5000; \Delta l = 200\mu m\}.$$

The results of this analysis are presented in figure 5.15 in which the magnetic field's magnitude is plotted as a function of time at the position ($x = 0.044m$, $y = 0.019m$). In the TLM results the frequency components lower than 500 GHz do not figure in the results since, as explained in section 5.2.3, not enough iterations are considered for these frequencies which has the effect of reducing the overall field radiations.

5.4 Discussion

The comparison between these two completely different numerical techniques has greatly increased our confidence in the CAD tool for predicting the shielding effectiveness of enclosures.

The near perfect agreement in the frequency immunity analysis has shown us the ability of both techniques to predict the shielding effectiveness of enclosures with apertures. In the emission frequency analysis, the comparisons have demonstrated the adaptability of both methods to various situations.

In the time domain analysis some insights, into the TLM method, were brought forward. Mainly that the analysis of wide frequency band signals is not feasible because the effects of the lower frequencies are not included in this analysis initially. Nevertheless, in the initial and latter stage of the analysis, both techniques compare fairly well.

The analysis of the two types of situations, namely the immunity and the emission cases, has shown us the versatility of both methods.

Chapter 6

Time-Domain Analysis of Complex Shielding Situations with FEM

6.1 Introduction

In the previous chapter, both numerical techniques were proven to be very effective in predicting the shielding effectiveness of enclosures with apertures. Their comparison has greatly increased our confidence in solving such problems via CAD tools; even the emission analysis which has a much more complicated formulation yields a fairly good comparison between these two completely different numerical techniques. The tools at this point may be applied confidently to much more complex situations. Although TLM could also be used to analyse the problems presented in this section, only FEM will be used as it is more versatile for these types of analyses.

The analysis is conducted in the time domain since studies, in this domain, are rarely performed. The reason why this type of analysis is infrequent is the lack of measurement scheme to carry out such a complex investigation (especially for fast

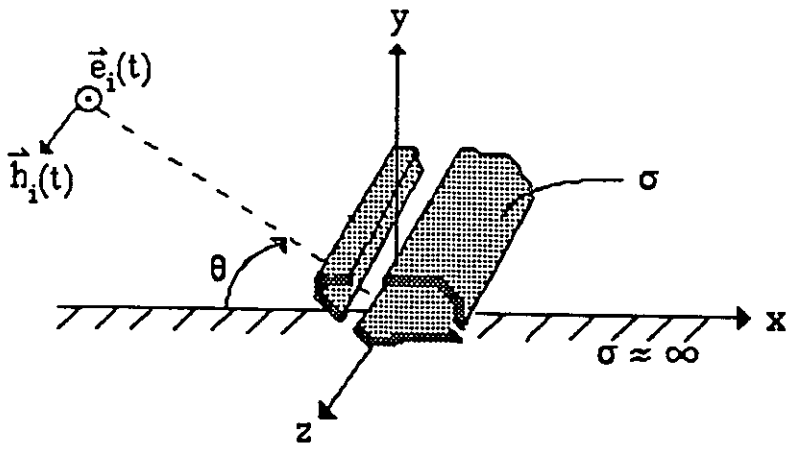


Figure 6.1: General immunity problem to investigate.

transients). The difficulty in measuring/predicting the field variations and penetrations in the time domain may be overcome by the introduction of CAD tools such as the ones presented here. With these tools, more possibilities exist now for the exploration of this new field.

As it was previously stated in chapter 5, "the length of the enclosure is assumed to be much longer than its width or height", thus making it possible to reduce a three-dimensional problem into a two-dimensional one. The complexity of the problem is then diminished without sacrificing too much the generality of the analysis.

6.2 Immunity Analysis

The problem of interest consists of predicting the time-domain shielding performance of any arbitrarily shaped enclosure with apertures. The enclosure, lying above a ground plane of infinite conductivity, is subjected to a time varying plane wave impinging at a certain angle θ, as shown in figure 6.1.

The main interest of this analysis is to investigate the shielding performance of enclosures illuminated by a high-altitude nuclear electromagnetic pulse (HNEMP), therefore the study is conducted in the far field region for which the impinging fields are plane waves [20]. The fields can then be decomposed into TE and TM propagating

modes and analysed separately, thus reducing the number of field components from three to one, thereby reducing the system of equations to be solved by a factor of three. Both modes are treated in the FEM code presented here (see Appendix C).

6.2.1 Considerations

The fact that the enclosure is lying above a perfect ground plane ($\sigma = \infty$) may be taken into consideration by making use of the mirror image concept, as explained in section 5.2.

The use of a numerical technique such as the FEM requires that the problem be enclosed within a bounded region. In this analysis the field is known at a distance far from the enclosure therefore the region under investigation may then be enclosed by an imaginary Dirichlet boundary at a certain distance from the enclosure. The field values at this boundary are calculated by superimposing the impinging fields to the ground reflected fields. This imaginary boundary is chosen to be a half sphere of radius r equal to ten times the enclosure's longest dimension [1]. To obtain the value of the field on the half sphere as a function of the impinging field, location and time, one needs to first calculate the distance from the real and image sources to any point on the half sphere as shown in figure 6.2.

In figure 6.2, $d_i(\phi)$ and $d_{im}(\phi)$, are respectively the distance from the real and image sources to a certain position on the sphere of radius r and R_s is the distance of the source with respect to the enclosure. Since the impinging fields are plane waves, we may set R_s to any value, thus R_s is chosen to be equal to r. The distances are given in the following

$$d_i(\phi) = r(1 - cos(\theta - \phi)) \tag{6.1}$$

$$d_{im}(\phi) = r(1 - cos(\theta + \phi)). \tag{6.2}$$

Now from equations (6.1) and (6.2), the field value at any position on the half sphere,

[1] The reason for specifying such a large radius is to allow the scattered fields on the enclosure to be negligeable (although the option for including the scattered field is provided in the program)

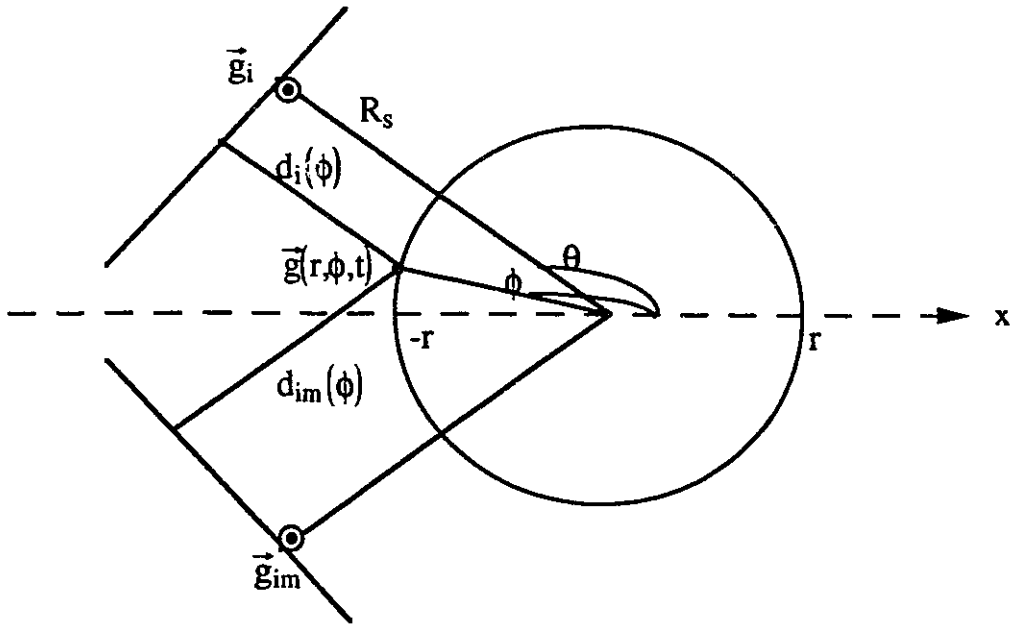


Figure 6.2: Distance from the real and image source to a boundary point.

when neglecting the scattered fields on the enclosure , is calculated as follows:

$$\begin{aligned}\vec{g}(r,\phi,t) &= \vec{g}_i\left(t-\frac{d_i(\phi)}{c_0}\right)+\vec{g}_{im}\left(t-\frac{d_{im}(\phi)}{c_0}\right)\\ &= \vec{g}_i\left(t-\frac{d_i(\phi)}{c_0}\right)+\Gamma\vec{g}_i\left(t-\frac{d_{im}(\phi)}{c_0}\right)\end{aligned} \tag{6.3}$$

where $\Gamma=-1$ for the TE case and $\Gamma=+1$ for the TM case.

The enclosure analysed is presented in figure 6.3 where R_s is the distance to the source, θ, the angle of incidence, r, the radius of the half spere boundary and $HE, LE, WE, PVAIRG, VAIRG$ correspond to the geometry of the enclosure. The division into triangular finite elements is similar to that of figure 6.4 only with a more refined mesh.

6.2.2 Results

The penetration of an NEMP incident field through an enclosure with apertures, similar to that shown in figure 6.3 is investigated. The geometry and field parameters

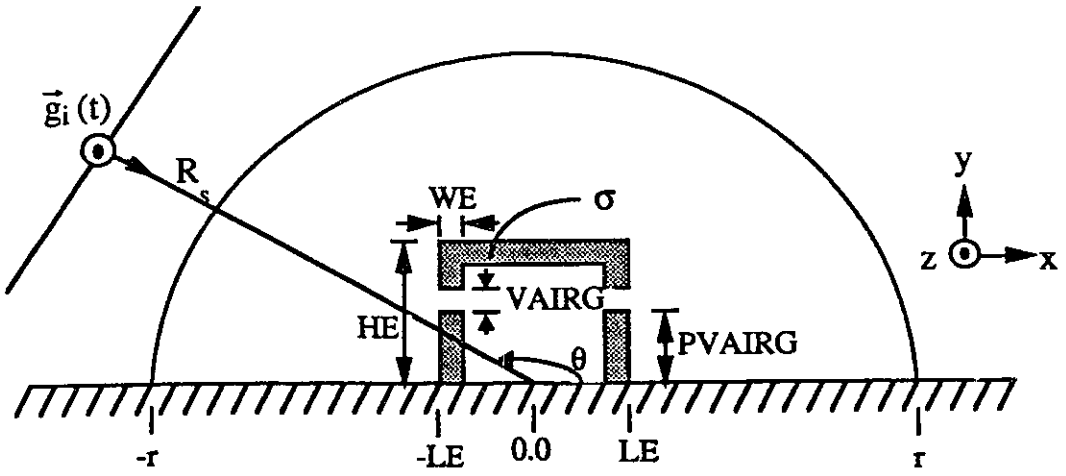


Figure 6.3: The analysed enclosure (immunity).

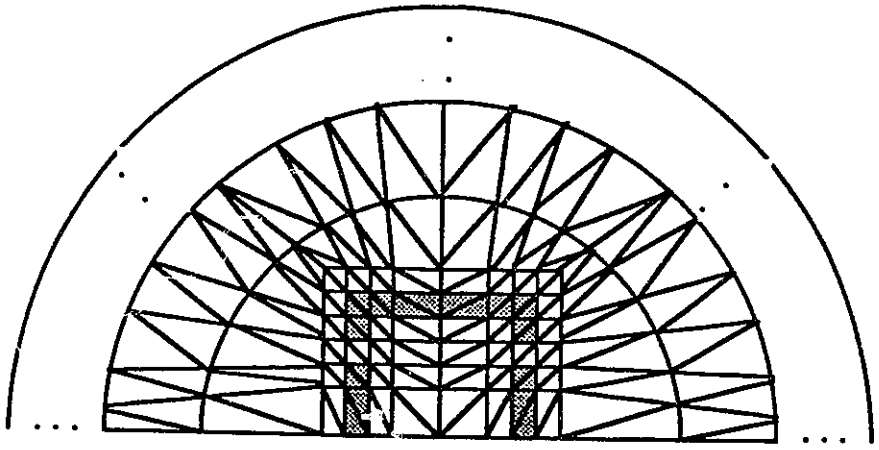
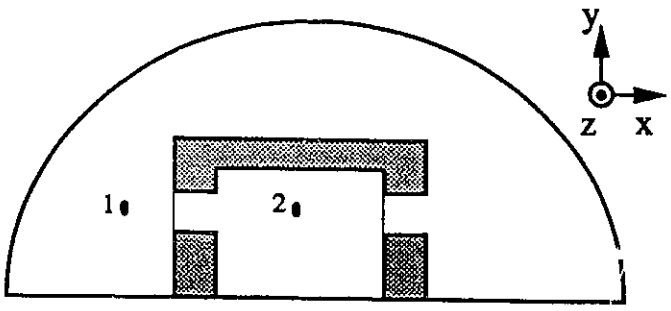


Figure 6.4: Division into FEs of region (immunity).



Position number	x (cm)	y (cm)
1	-11.25	8.5
2	0.0	8.5

Figure 6.5: Position of the field calculation.

of the problem are given as :

$$LE = 0.1m \qquad HE = 0.1m \qquad WE = 0.1mm \quad PVAIRG = 0.08m$$
$$VAIRG = 0.01m \quad R_s = r = 1.0m \qquad \theta = 160°$$

$$E_0 = 5.25 \times 10^4 V/m \quad H_0 = E_0/\eta_0 = 1.39 \times 10^2 A/m$$
$$\alpha = 4.0 \times 10^6 \qquad\qquad \beta = 4.76 \times 10^8$$

where E_0 and H_0 correspond respectively to the electric and magnetic field constants as in equation (2.11) and η_0 is the free space characteristic impedance ($\eta_0 = \sqrt{\frac{\mu_0}{\epsilon_0}} \simeq 377\Omega$).

The results of this analysis are plotted in figures 6.6 to 6.9 for which the numbers 1 & 2 determine the position of the field calculation points (see figure 6.5).

In figures 6.6 and 6.8 the electric and magnetic field penetration are plotted as a function of time. Notice here the time delay and the flatter response of the predicted magnetic field in the middle of the enclosure. The effect of the ground plane on the electric field signature is shown in figure 6.6, the waveform reveals to be rather complex due to the ground plane reflections.

The shielding effectiveness (SE) of enclosures is generally defined in the frequency domain as the ratio of the measured or calculated field without the enclosure to the field with the enclosure [47].

$$SE(\omega) = 20 \log_{10} \left\{ \frac{|\vec{G}_{no\ enclosure}(\omega)|}{|\vec{G}_{with\ enclosure}(\omega)|} \right\} \tag{6.4}$$

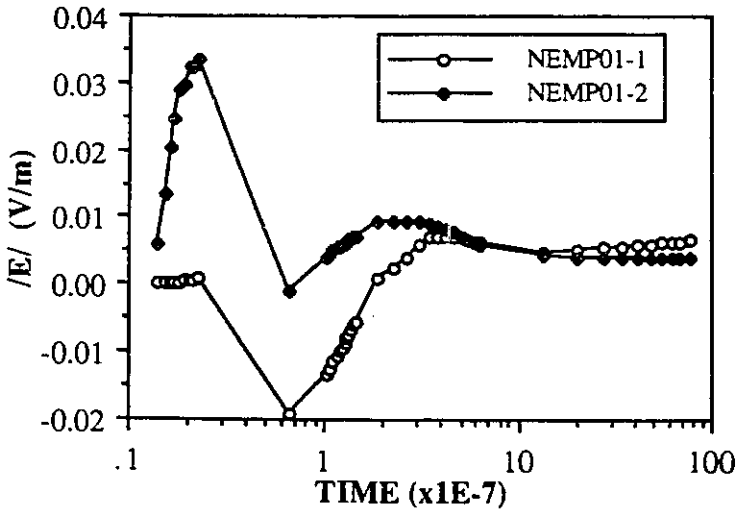


Figure 6.6: Electric field penetration at positions 1 & 2.

where $\vec{G}$ is the field in question. In this analysis we define the time domain shielding effectiveness as

$$SE(t) = 20 \log_{10} \left\{ \frac{|\vec{g}_{no\ enclosure}(t)|}{|\vec{g}_{with\ enclosure}(t)|} \right\} \tag{6.5}$$

which is also given in decibels, dBs. Thus the time domain shielding effectiveness of the enclosure presented here, is plotted in figures 6.7 and 6.9 for the position #2 inside the enclosure for both TE and TM cases, respectively. It is seen that initially the SE of the enclosure is very high since the higher frequency components of the impinging field are greatly attenuated by the enclosure, but as time progress, the lower frequency components predominate thus reducing the overall shielding effectiveness.

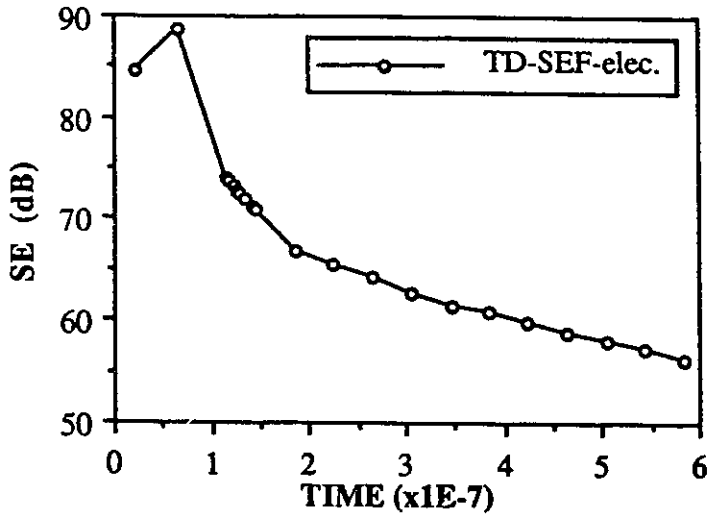


Figure 6.7: Time domain electric SE at position 2.

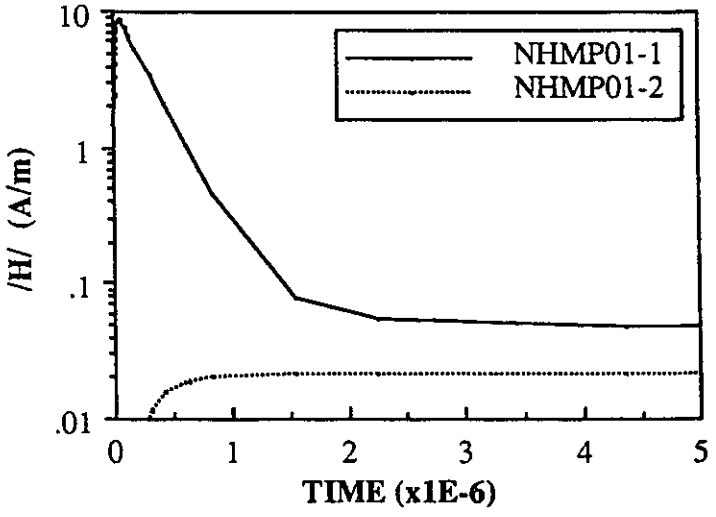


Figure 6.8: Magnetic field penetration at positions 1 & 2.

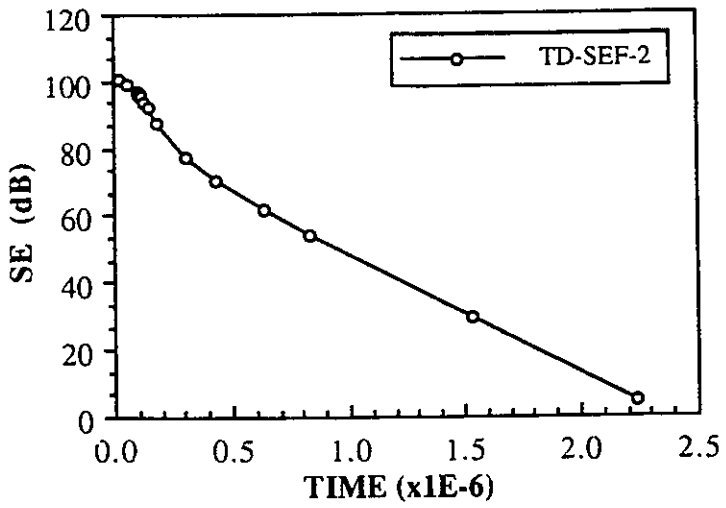


Figure 6.9: Time domain magnetic SE at position 2.

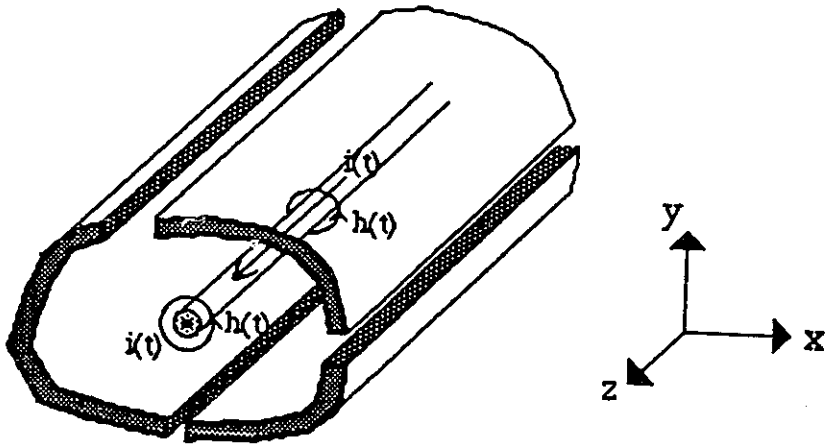


Figure 6.10: General emission problem under investigation.

6.3 Emission Analysis

This section treats the time-domain shielding effectiveness of an enclosure in which a radiating conductor is located. Contrary to the immunity case, we are interested in keeping as much energy as possible confined within the enclosure and its walls. The conductor, situated inside an arbitrarily shaped enclosure, is carrying a time varying current from which an associated magnetic field is generated (see figure 6.10).

The interest here lies in the analysis of the shielding performance of an enclosure in which a current flowing in a conductor, has a time signature similar to a digital pulse

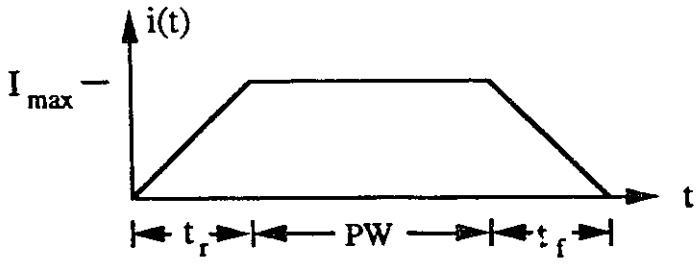


Figure 6.11: Current time signature.

signal. This signal will be assumed to have a rise and fall time varying linearly and a constant magnitude throughout the pulse width (see figure 6.11). The associated magnetic field will have the same shape as the current and a magnitude that is governed by the following equation [41]

$$h(t) = \frac{i(t)}{2\pi a} \tag{6.6}$$

where a is the radius of the conductor. Equation 6.6 is valid at a close proximity of the conductor. Again, to reduce the size of the final system of equations to be solved, the analysis is conducted using the magnetic vector potential, $\vec{a}(t)$, see section 5.3.1 for more details.

6.3.1 Considerations

The comparison, in section 5.3, between FEM and TLM for the emission case was carried out by neglecting the diffusion process and also by imposing symmetry to the problem. This was done to limit the processing time required by the TLM method, especially in the time domain analysis. Here neither a low skin depth nor a symmetry shall be imposed. We will then be at liberty to place the conductor anywhere within the enclosure since the finite element division code (see Appendix D) is written for general problems.

The region of analysis is unbounded, thus there is a need for infinite elements to bound such a region. The derivation of these infinite elements is found in section 5.3.1 as well as in Appendix B.

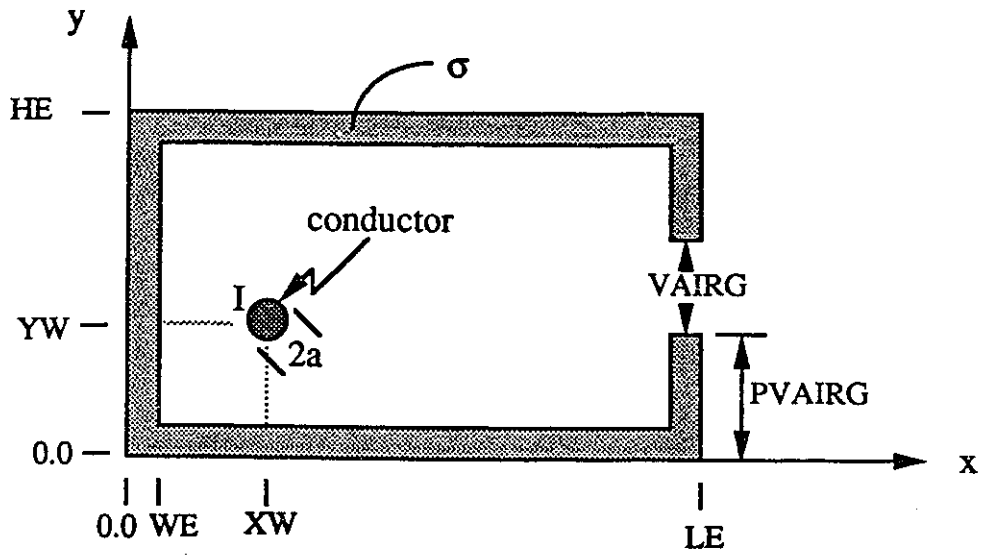


Figure 6.12: The analysed enclosure (emission).

The emission problem analysed is shown in figure 6.12 where LE, HE, WE, $PVAIRG$, $VAIRG$ correspond to the geometry of the enclosure, XW and YW are the position of the conducting wire and σ is the conductivity of the walls.

6.3.2 Results

The emission problem under investigation has a wire conducting a current from a digital circuit that is inside an enclosure with an aperture, see figure 6.12. Two analysis were conducted in which the conductor was positioned differently. In one analysis the conductor is far from the aperture ($XW = 0.03m, YW = 0.03m$) and in the other it is set closer ($XW = 0.07m, YW = 0.07m$). The geometry and current parameters are given as follows

$$\begin{array}{cccc} LE = 0.1m & HE = 0.1m & WE = 0.1mm & 2a = 1.0mm \\ PVAIRG = 0.045m & VAIRG = 0.01m & \sigma = 0.6x10^8 S/m & \\ I_{max} = 1.0mA & t_r = 1.0ns & PW = 10.0ns & t_f = 1.0ns \end{array}$$

The effect of the current source position on the radiated energy is well observed in figure 6.14 where the time domain magnetic field is presented, in both cases, at

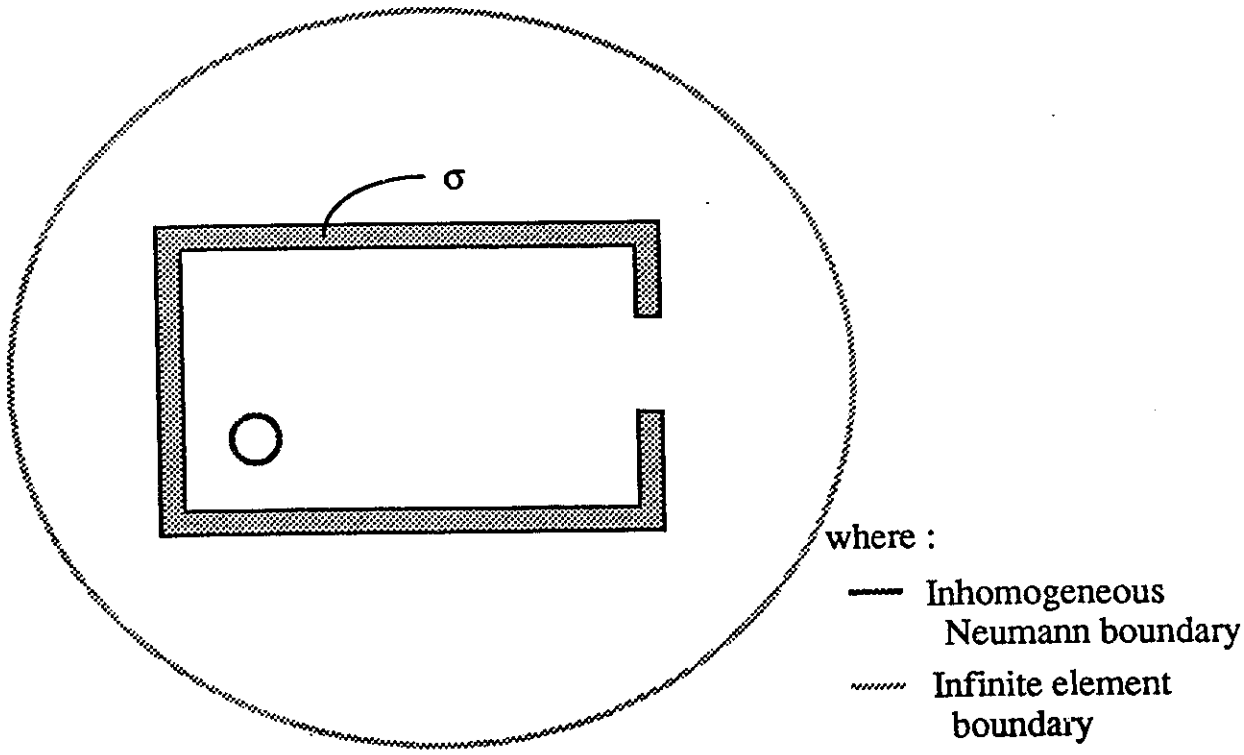


Figure 6.13: FEM boundary conditions (emission).

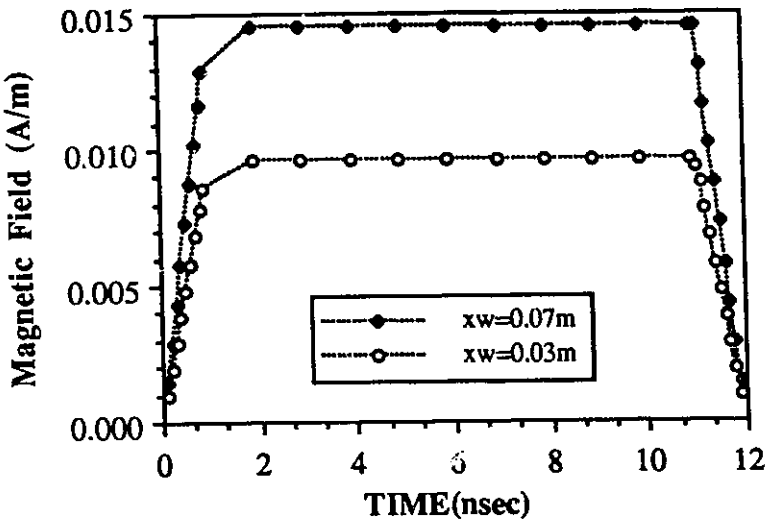


Figure 6.14: Magnetic field radiation for different positions of source.

a position just outside the enclosure's aperture ($x = 0.139m, y = 0.055m$). Notice that the radiated field almost doubles when the current source is neer the aperture as compared to the furthest one.

Chapter 7

Conclusion

The object of this whole analysis was to develop a CAD tool capable of solving shielding effectiveness problems both in time and frequency domains. To achieve this task we have chosen the finite element method as the foundation for the tool. This numerical technique was proven over the years to be a very efficient and versatile method.

The CAD tool's performance was analysed in a methodic manner, starting from very simple problems and working our way up to complex situations. The derivation of an analytical solution as a means of validation, in the initial phase, proved to be an excellent choice as it provided a solid means by which the operating range of the numerical techniques may be determined. With the availability of this analytical formulation, rules of thumb regarding the diffusion process could be easily verified thus increasing our confidence in the numerical methods.

The introduction of a second tool gave us the opportunity to validate the finite element code for slightly more involved analysis which included apertures (needed to communicate with the outside world and to allow the dissipation of the heat generated by the electronic device) in the shield.

This comparison was very instructive for both techniques as it provided further information on their operating range. Moreover, it was seen that the time-domain

analysis through TLM is not feasible for wide frequency band signals because this analysis does not take into consideration the effect of the lower frequencies.

With full confidence in the CAD tool for predicting shielding effectiveness of enclosures within known limits, the tool was then applied to more realistic situations.

The main contributions of this thesis are summarized below:

- development of a CAD tool to predict shielding effectiveness of enclosures
- analytical formulation of both frequency and time diffusion problems
- devising new applications of the FEM
- devising new applications of the TLM method
- deriving rules of thumb for both FEM and TLM regarding diffusion problems
- time-domain analysis of the diffusion problem with FEM (to the authors' knowledge, no other analysis of such problems has been reported)
- better understanding of diffusion processes

Appendix A

Time Domain Analytical Formulation

The general solution for the time domain analysis is given in equations (2.14) and (2.15) and repeated here for convenience

$$\vec{E}_2(x,s) = \hat{z}[A\cos(\sqrt{-s\mu\sigma}\,x) + B\sin(\sqrt{-s\mu\sigma}\,x)] \quad \text{(A.1)}$$

$$\vec{E}_3(x,s) = \hat{z}C\cosh(\sqrt{s^2\mu_0\epsilon_0}\,x) \quad \text{(A.2)}$$

where A, B, and C are determined by the boundary conditions. The derivation leads to the following

$$A = L(s)(\cosh\phi\cos\gamma - \delta\sinh\phi\sin\gamma)/R(s) \quad \text{(A.3)}$$

$$B = L(s)(\cosh\phi\sin\gamma + \delta\sinh\phi\cos\gamma)/R(s) \quad \text{(A.4)}$$

$$C = L(s)/R(s) \quad \text{(A.5)}$$

where

$$R(s) = [\cosh\phi\cos(\gamma-\theta) - \delta\sinh\phi\sin(\gamma-\theta)]$$

$$L(s) = \mathcal{L}\{E_0(e^{-\alpha t} - e^{-\beta t})\} = \frac{E_0(\beta-\alpha)}{(s+\alpha)(s+\beta)}.$$

Then (A.1) and (A.2) may be rewritten as:

$$\vec{E}_2(x,s) = \hat{z}P_2(s)/Q(s) \tag{A.6}$$

$$\vec{E}_3(x,s) = \hat{z}P_3(s)/Q(s) \tag{A.7}$$

in which

$$\begin{aligned} P_2(s) &= E_0(\beta-\alpha)\{(\cosh\phi\cos\gamma - \delta\sinh\phi\sin\gamma)\cos(\sqrt{-s\mu\sigma}\,x) + \\ &\quad (\cosh\phi\sin\gamma + \delta\sinh\phi\cos\gamma)\sin(\sqrt{-s\mu\sigma}\,x)\} \\ P_3(s) &= E_0(\beta-\alpha)\cosh(\sqrt{s^2\mu_0\epsilon_0}\,x) \\ Q(s) &= (s+\alpha)(s+\beta)R(s) \end{aligned}$$

The time domain response can be obtained by transforming (A.6) and (A.7) with Heaviside's formula

$$e(t) = \sum_{k=1}^{\infty} \frac{P(s_k)}{\dot{Q}(s_k)} e^{s_k t} \tag{A.8}$$

with s_k being the poles of $Q(s)$ are given as:

- $s_1 = -\alpha$
- $s_2 = -\beta$
- s_n, the roots of the following equation
  $\tanh(\sqrt{s_n^2\mu_0\epsilon_0}\,D_1) = \eta_0\sqrt{\frac{\sigma}{-s_n\mu}}\,\cot(\sqrt{-s_n\mu\sigma}\,(D_1 - D_2))$

and $\dot{Q}(s)$ the derivative of $Q(s)$ with respect to s. The derivative of $Q(s)$ is given as

$$\dot{Q}(s) = \{(2s+\alpha+\beta)R(s) + (s+\alpha)(s+\beta)\dot{R}(s)\} \tag{A.9}$$

where

$$\begin{aligned} \dot{R}(s) &= [\sinh\phi\frac{\partial\phi}{\partial s}\cos(\gamma-\theta) - \cosh\phi\sin(\gamma-\theta)\frac{\partial(\gamma-\theta)}{\partial s} - \\ &\quad \frac{\partial\delta}{\partial s}\sinh\phi\sin(\gamma-\theta) - \delta\cosh\phi\frac{\partial\phi}{\partial s}\sin(\gamma-\theta) - \\ &\quad \delta\sinh\phi\cos(\gamma-\theta)\frac{\partial(\gamma-\theta)}{\partial s}] \end{aligned}$$

$$\begin{aligned}
\frac{\partial \phi}{\partial s} &= \sqrt{\mu_0 \epsilon_0}\, D_1 \\
\frac{\partial \delta}{\partial s} &= \frac{-1}{2\eta_0}\sqrt{\frac{\mu}{-s\sigma}} \\
\frac{\partial(\gamma - \theta)}{\partial s} &= -\sqrt{\frac{\mu\sigma}{-s}}\,\frac{(D_1 - D_2)}{2}
\end{aligned}$$

Now by introducing the poles of $Q(s)$ in equation (A.8), the time domain conterpart of (A.6) and (A.7) are obtained

$$\vec{e}_{2/3}(x,t) = \hat{z}\left[\frac{P_{2/3}(s_1)}{\dot{Q}(s_1)}\, e^{s_1 t} + \frac{P_{2/3}(s_2)}{\dot{Q}(s_2)}\, e^{s_2 t} + \sum_{n=1}^{\infty} \frac{P_{2/3}(s_n)}{\dot{Q}(s_n)}\, e^{s_n t}\right] \qquad \text{(A.10)}$$

Appendix B

Infinite Element Contribution to the System of Equations

The contribution to the T matrix given in the general equation of (3.24) is calculated as follows. For the finite elements located on the open boundary (infinite boundary), an additional contribution must be included in the final system of equations. On this boundary, the field may be approximated by the following linear shape function, as described in section 5.3.1,

$$\begin{aligned} \vec{a}(u,v,t) &= \hat{z}\frac{d^n}{2u^n}\left[\left(1+\frac{b}{l}-\frac{dv}{lu}\right)a_i(t)+\left(1-\frac{b}{l}+\frac{dv}{lu}\right)a_j(t)\right] \\ &= \hat{z}\left[N_i a_i(t)+N_j a_j(t)\right] \end{aligned} \tag{B.1}$$

Now replacing equation (B.1) into the integral related to the T matrix we get the following

$$T\prime = \int_{\Omega_l}\left[N_i^2 a_i^2(t)+N_j^2 a_j^2(t)+2N_i N_j a_i(t)a_j(t)\right]d\Omega_l \tag{B.2}$$

where

$$N_i^2 = \frac{d^{2n}}{4n^{2n}}\left(1+\frac{b^2}{l^2}+\frac{d^2v^2}{l^2u^2}+\frac{2b}{l}-\frac{2dv}{lu}-\frac{2bdv}{l^2u}\right)$$

$$N_j^2 = \frac{d^{2n}}{4n^{2n}}\left(1+\frac{b^2}{l^2}+\frac{d^2v^2}{l^2u^2}-\frac{2b}{l}+\frac{2dv}{lu}-\frac{2bdv}{l^2u}\right)$$

$$2N_iN_j = \frac{d^{2n}}{2n^{2n}}\left(1 - \frac{b^2}{l^2} - \frac{d^2v^2}{l^2u^2} + \frac{2bdv}{l^2u}\right)$$

Taking the first variation of (B.2) with respect to the unknown and setting it equal to zero

$$T_{ii} = \frac{\partial}{\partial a_i(t)}T\prime = 2\int_d^\infty \int_{u\left(\frac{b-l}{d}\right)}^{u\left(\frac{b+l}{d}\right)} N_i^2 dv\, du \tag{B.3}$$

$$T_{jj} = \frac{\partial}{\partial a_j(t)}T\prime = 2\int_d^\infty \int_{u\left(\frac{b-l}{d}\right)}^{u\left(\frac{b+l}{d}\right)} N_j^2 dv du \tag{B.4}$$

$$T_{ij} = T_{ji} = \frac{\partial}{\partial a_{i-j}(t)}T\prime = 2\int_d^\infty \int_{u\left(\frac{b-l}{d}\right)}^{u\left(\frac{b+l}{d}\right)} N_iN_j dv du \tag{B.5}$$

Solving the integrals of (B.3) to (B.5) gives

$$T_{ii} = T_{jj} = \frac{2dl}{(3n-3)} \tag{B.6}$$

$$T_{ij} = T_{ji} = \frac{dl}{(3n-3)} \tag{B.7}$$

Appendix C

FEM Computer Program for Immunity Problems in Time Domain

```
      PROGRAM SETDAE                                                    SEM00010
C                                                                       SEM00020
C                                                                       SEM00030
C  >=================================================================<  SEM00040
C  >                                                                  <  SEM00050
C  >                  PROGRAM SEMPTDDA                                <  SEM00060
C  >                  ----------------                                <  SEM00070
C  >                                                                  <  SEM00080
C  > ***DATE WRITTEN   10/08/89 (DD/MM/YY)                           <  SEM00090
C  > ***AUTHOR     GRAVELLE, LUC B.                                  <  SEM00100
C  >               ELECTRICAL ENGINEERING DEPARTMENT                 <  SEM00110
C  >               UNIVERSITY OF OTTAWA, CANADA                      <  SEM00120
C  > ***PURPOSE    PREDICT THE SHEILDING EFFECTIVENESS OF A TWO-     <  SEM00130
C  >               DIMENSIONAL ENCLOSURE WITH APERTURES IN THE       <  SEM00140
C  >               TIME-DOMAIN.                                      <  SEM00150
C  >                                                                 <  SEM00160
C  >=================================================================<  SEM00170
C  >                                                                  <  SEM00180
C  >  THIS FORTRAN CODE IS A FINITE ELEMENT METHOD (FEM) BASED        <  SEM00190
C  >  TECHNIQUE USED TO PREDICT THE SHIELDING PERFORMANCE OF          <  SEM00200
C  >  ANY ARBITRARELY SHAPED TWO-DIMENSIONAL ENCLOSURE WITH           <  SEM00210
C  >  APERTURES (SINCE THE DIVISION INTO TRIANGLES IS                 <  SEM00220
C  >  AUTOMATED, SOME RESTRICTIONS ARE IMPOSED ON THE SHAPE           <  SEM00230
C  >  OF THE ENCLOSURE).                                             <  SEM00240
C  >  THE SOLUTION OF AN EMP TYPE FIELD IMPINGING ON THE ENCLOSURE    <  SEM00250
C  >  AT AN ARBITRARY ANGLE IS OBTAINED DIRECTLY IN THE TIME DOMAIN   <  SEM00260
C  >  VIA A DIFFERENTIAL/ALGEBRAIC SOLVER, THE "DDASSL".  THIS CODE   <  SEM00270
C  >  IS BASED ON THE BACKWARD DIFFERENTIATION FORMULAS OF ORDERS ONE <  SEM00280
C  >  TO FIVE (THE ORDER FIVE IS USED IN THIS PROGRAM).  IT WAS       <  SEM00290
C  >  WRITTEN BY LINDA R. PETZOLD FROM LAWRENCE LIVERMORE NATIONAL.   <  SEM00300
C  >                                                                  <  SEM00310
C  >     --> FOR MORE INFORMATION ON THE PROGRAM "DDASSL" SEE ITS     <  SEM00320
C  >         LISTING                                                  <  SEM00330
C  >                                                                  <  SEM00340
C  >     --> PLEASE READ THE FILE NAMED "READ.ME" BEFORE STARTING     <  SEM00350
C  >                                                                  <  SEM00360
C  >=================================================================<  SEM00370
C  >                                                                  <  SEM00380
C  >                  DESCRIPTION OF PARAMETERS                       <  SEM00390
C  >                  -------------------------                       <  SEM00400
C  >  NUNK,NTRI      : THE MAXIMUM # OF UNKNOWNS & TRIANGLES          <  SEM00410
C  >  MLMX,MUMX      : THE MAXIMUM HALF-BANDWITHS ML(LOWER) & MU(UPPER) <  SEM00420
C  >                                                                  <  SEM00430
C  >  NVAIRG         : THE NUMBER (#) OF APERTURES PER VERTICLE WALLS  <  SEM00440
C  >  PVAIRG,VAIRG   : THE POSITION & LENGTH OF THE APERTURES         <  SEM00450
C  >  HE,LE,WE       : HEIGHT, LENGTH & WALL THICKNESS OF ENCLOSURE    <  SEM00460
C  >  SIG,MIURW      : CONDUCTIVITY & RELATIVE PERMEABILITY OF THE WALL<  SEM00470
C  >  EO,ALPHA,BETA  : PEAK VALUE & TIME CHARACTERISTICS OF THE FIELD  <  SEM00480
C  >  RS,TETAS       : DISTANCE & ANGLE OF INCIDENCE OF SOURCE         <  SEM00490
C  >  NREG           : # OF REGIONS TO ANALYZE                         <  SEM00500
C  >  X1,Y1,X2,Y2    : COORDINATES OF THE REGIONS TO BE ANALYZED       <  SEM00510
C  >  NINT,TINT      : # OF INTEGRATION TIMES & THE TIMES              <  SEM00520
C  >  TM             : TYPE OF ANALYSIS (TM OR TE)                     <  SEM00530
C  >  SCAT,IOUT      : SCATTERING & TIME PARAMETERS (PARAM.)           <  SEM00540
                                                                        SEM00550
```

```
C  >
C  >  X,Y             : COORDINATES OF THE TRIANGLE VERTICES                    < SEM00560
C  >  SIGMA,MIUR      : CONDUCTIVITY & RELATIVE PERMEABILITY OF TRIANGLE        < SEM00570
C  >  NV,DIRP         : # ASSOCIATED TO EACH NODES & BOUNDARY CONDITION         < SEM00580
C  >                                                                            < SEM00590
C  >  A,AP            : THE VALUE & DERIVATIVE OF THE FIELD AT EACH NODE        < SEM00600
C  >  W,RWORK,IWORK   : WORKING SPACE REQUIRED BY "DDASSL"                      < SEM00610
C  >  R,S,SE          : MATRICES RELATED TO THE SYSTEM OF EQUATIONS             < SEM00620
C  >  Q,TQ            : REALTED TO P. SILVESTER'S SHAPE FUNCTIONS               < SEM00630
C  >  INFO            : INFORMATIONS REQUIRED BY "DDASSL"                       < SEM00640
C  >  PIMN1 - PIMX3   : CRITICAL ANGLES WHEN CONSIDERING SCATTERED FIELD       < SEM00650
C  >  VMIUO,CO        : PERMEABILITY & SPEED OF LIGHT IN AIR                    < SEM00660
C  >                                                                            < SEM00670
C  >                                                                            < SEM00680
C  >============================================================================< SEM00690
C                                                                               < SEM00700
C                                                                               SEM00710
C                                                                               SEM00720
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                            SEM00730
      IMPLICIT  REAL*8(A-H,O-Z)                                                 SEM00740
C  >--------- VALUES FROM SUBROUTINE GETINF ---------------------------------<  SEM00750
      REAL*8    PVAIRG(5),VAIRG(5),HE,LE,WE                                     SEM00760
      REAL*8    SIG,MIURW,EO,ALPHA,BETA,TETAS                                   SEM00770
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)                            SEM00780
      INTEGER   NVAIRG,TM,SCAT,NREG,IOUT,NINT                                   SEM00790
C  >--------- VALUES FROM TRIANG SUBROUTINE ----------------------------------<  SEM00800
      REAL*8    X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS                    SEM00810
      INTEGER   NV(NTRI,3)                                                      SEM00820
      LOGICAL*4 DIRP(NUNK)                                                      SEM00830
C  >--------- VALUES FOR PROGRAM SEMPTDDA ------------------------------------<  SEM00840
      REAL*8    A(NUNK),AP(NUNK)                                                SEM00850
      REAL*8    RWORK(40+9*NUNK+(2*MLMX+MUMX+1)*NUNK)                           SEM00860
C     REAL*8    RWORK(40+9*NUNK+NUNK*NUNK)                                      SEM00870
      REAL*8    R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1),SE(3,3),T(200)          SEM00880
      INTEGER   Q(4,15,15,3),TQ(4,15,15),INFO(15),IWORK(20+NUNK)                SEM00890
C                                                                               SEM00900
C  >============================================================================< SEM00910
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                          < SEM00920
C  >               SUBROUTINES AND FUNCTIONS                                    < SEM00930
C  >============================================================================< SEM00940
C                                                                               < SEM00950
      COMMON    /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,           SEM00960
     &           BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT           SEM00970
      COMMON    /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU DIRP       SEM00980
      COMMON    /PHI/PIMN1,PIMMN1,PIMMX1,PIMN2,PIMMN2,PIMMX2,                   SEM00990
     &           PIMN3,PIMX3                                                    SEM01000
      COMMON    /OUT/A                                                          SEM01010
      COMMON    /RAS/R,S,NEQ/SIMPL/Q,TQ                                         SEM01020
C                                                                               SEM01030
      EXTERNAL RES,JAC                                                          SEM01040
C                                                                               SEM01050
      P(A1,B1,A2,B2,A3,B3) = A1*B2-A2*B1+A2*B3-A3*B2+A3*B1-A1*B3                SEM01060
C                                                                               SEM01070
C                                                                               SEM01080
C  >============================================================================< SEM01090
                                                                                  SEM01100
```

```
C  >              HERE ARE THE OPENING STATEMENTS                          < SEM01110
C  >-------------------------------------------------------------------------< SEM01120
C                                                                             SEM01130
      OPEN(UNIT=7,FILE='NEMP01.INPUT',STATUS='OLD')                           SEM01140
      OPEN(UNIT=9,FILE='NEMP01.SEMPTDDA',STATUS='NEW')                        SEM01150
C                                                                             SEM01160
      PI    = DACOS(-1.0D0)                                                   SEM01170
      VMIUO = 4.0D-07*PI                                                      SEM01180
      CO    = 3.0D08                                                          SEM01190
C                                                                             SEM01200
CC >-------------------------------------------------------------------------< SEM01210
CC >     THE FOLLOWING CALL GETS THE NECESSARY INFORMATION                    < SEM01220
CC >              CONCERNING THE GEOMETRY & THE FIELD                         < SEM01230
CC >-------------------------------------------------------------------------< SEM01240
C                                                                             SEM01250
      CALL GETINF                                                             SEM01260
C                                                                             SEM01270
CC >-------------------------------------------------------------------------< SEM01280
CC >  THE FOLLOWING CALL PRINTS THE INFORMATION ON THE OUTPUT FILE            < SEM01290
CC >-------------------------------------------------------------------------< SEM01300
C                                                                             SEM01310
      CALL PRTINF                                                             SEM01320
C                                                                             SEM01330
CC >-------------------------------------------------------------------------< SEM01340
CC >     THIS CALL TO "TRIANG" WILL TRIANGULARIZE THE REGION OF               < SEM01350
CC >        INTEREST AND DETERMINE THE DIRICHLET BOUNDARIES                   < SEM01360
CC >-------------------------------------------------------------------------< SEM01370
C                                                                             SEM01380
      CALL TRIANG                                                             SEM01390
      NN = NOPTL - NDIR                                                       SEM01400
C                                                                             SEM01410
CC >-------------------------------------------------------------------------< SEM01420
CC >     THIS CALL TO "PHICAL" WILL CALCULATE THE CRITICAL                    < SEM01430
CC >        ANGLES OF SCATTERING ON THE ENCLOSURE (IF REQUIRED)               < SEM01440
CC >-------------------------------------------------------------------------< SEM01450
C                                                                             SEM01460
      IF (SCAT .EQ. 1) CALL PHICAL                                            SEM01470
C                                                                             SEM01480
CC >-------------------------------------------------------------------------< SEM01490
CC >     NEXT WE DETERMINE THE TIME STEPS AND THE FINAL TIME                  < SEM01500
CC >        FOR THE ANALYSIS OF THIS PARTICULAR PROBLEM                       < SEM01510
CC >-------------------------------------------------------------------------< SEM01520
      SMIU  = SIG*VMIUO*MIURW                                                 SEM01530
      TSTA  = (RS - 0.990D0*RB)/CO                                            SEM01540
      TPEAK = DLOG(BETA/ALPHA)/(BETA - ALPHA)                                 SEM01550
      TDIF  = SMIU*WE**2                                                      SEM01560
      TPIN  = TSTA + TPEAK + TDIF/(2.0D0**(DLOG10(ALPHA) - 4))                SEM01570
      EPEAK = DEXP(-ALPHA*TPEAK) - DEXP(-BETA*TPEAK)                          SEM01580
      DO 100 I = 2 , 10000                                                    SEM01590
         IF (-BETA*TPEAK*REAL(I) .LE. -120.D0) THEN                           SEM01600
           IF (DEXP(-ALPHA*TPEAK*REAL(I))  .LT. 0.10D0*EPEAK) GO TO 200       SEM01610
         ELSE                                                                 SEM01620
           IF ((DEXP(-ALPHA*TPEAK*REAL(I)) - DEXP(-BETA*TPEAK*REAL(I)))       SEM01630
     &       .LT. 0.10D0*EPEAK) GO TO 200                                     SEM01640
                                                                              SEM01650
```

```
        ENDIF
100   CONTINUE                                                          SEM01660
200   CONTINUE                                                          SEM01670
      TFIN  = REAL(I)*TPEAK                                             SEM01680
      N1    = I                                                         SEM01690
      TSTEP = TPEAK/10.0D0                                              SEM01700
      IF (N1 .GT. 15) THEN                                              SEM01710
        N1 = 15                                                         SEM01720
        TSTEP2 = TFIN/15.0D0                                            SEM01730
      ELSE                                                              SEM01740
        TSTEP2 = TPEAK                                                  SEM01750
      ENDIF                                                             SEM01760
      TSTEP3 = TDIF                                                     SEM01770
      T(1)   = TSTA                                                     SEM01780
C                                                                       SEM01790
C                                                                       SEM01800
C                                                                       SEM01810
C  >-------------------------------------------------------------------< SEM01820
C  >     THEN THE DISCRETE TIMES ARE CALCULATED                        < SEM01830
C  >-------------------------------------------------------------------< SEM01840
C                                                                       SEM01850
      K = 0                                                             SEM01860
      DO 300 I = 2 , 20 + N1                                            SEM01870
        IF (I .LE. 10) THEN                                             SEM01880
          T(I+K) = T(I+K-1) + TSTEP                                     SEM01890
        ELSE IF (I .LE. 10 + N1) THEN                                   SEM01900
          T(I+K) = T(I+K-1) + TSTEP2                                    SEM01910
        ELSE                                                            SEM01920
          T(I+K) = T(I+K-1) + TSTEP3                                    SEM01930
        ENDIF                                                           SEM01940
        IF (T(I+K-1)  .LT. TPIN  .AND.  T(I+K) .GE. TPIN) THEN          SEM01950
          STEPX  = (T(I+K) - T(I+K-1))/10.0D0                           SEM01960
          T(I+K) = T(I+K-1) + STEPX                                     SEM01970
          DO 400 K = 1 , 10                                             SEM01980
              T(I+K) = T(I+K-1) + STEPX                                 SEM01990
400       CONTINUE                                                      SEM02000
          K = K - 1                                                     SEM02010
        ENDIF                                                           SEM02020
300   CONTINUE                                                          SEM02030
C                                                                       SEM02040
C  >-------------------------------------------------------------------< SEM02050
C  >     NEXT THE TIME REQUIRED BY THE USER ARE MERGED                 < SEM02060
C  >        WITH THE CALCULATED TIMES (IF THERE ARE ANY!!!)            < SEM02070
C  >-------------------------------------------------------------------< SEM02080
C                                                                       SEM02090
      IF (NINT .NE. 0) THEN                                             SEM02100
        DO 500 I = 1 , NINT                                             SEM02110
          TINT(I) = TINT(I) + TSTA                                      SEM02120
500     CONTINUE                                                        SEM02130
        K = 1                                                           SEM02140
        DO 600 I = 2 , 30 + N1 + NINT                                   SEM02150
          IF (T(I) .GT. TINT(K)  .AND.  T(I-1) .LT. TINT(K)) THEN       SEM02160
            DO 700 J = 30 + N1 + K , I , -1                             SEM02170
              T(J) = T(J-1)                                             SEM02180
700         CONTINUE                                                    SEM02190
                                                                        SEM02200
```

```
            T(I) = TINT(K)                                                    SEM02210
            K    = K + 1                                                      SEM02220
          ENDIF                                                               SEM02230
600     CONTINUE                                                              SEM02240
        DO 800 I = 30 + N1 + K , 30 + N1 + NINT                               SEM02250
          T(I) = TINT(K)                                                      SEM02260
          K    = K + 1                                                        SEM02270
800     CONTINUE                                                              SEM02280
      ENDIF                                                                   SEM02290
C                                                                             SEM02300
C  >-------------------------------------------------------------------<     SEM02310
C  >     THIS CALL TO "SIMPLX" DEFINES THE SHAPE FUNCTIONS AS          <     SEM02320
C  >       GIVEN IN P. SILVESTER'S BOOK ON THE FEM                     <     SEM02330
C  >-------------------------------------------------------------------<     SEM02340
C                                                                             SEM02350
      CALL SIMPLX(1)                                                          SEM02360
C                                                                             SEM02370
C  >-------------------------------------------------------------------<     SEM02380
C  >     WE CAN NOW SET-UP THE "S" AND "R" MATRICES AT THE             <     SEM02390
C  >               INITIAL TIME "T(1)"                                 <     SEM02400
C  >-------------------------------------------------------------------<     SEM02410
C                                                                             SEM02420
      DO 900 I = 1 , NUNK                                                     SEM02430
        DO 900 J = 1 , MLMX+MUMX+1                                            SEM02440
          R(I,J) = 0.0D0                                                      SEM02450
          S(I,J) = 0.0D0                                                      SEM02460
900   CONTINUE                                                                SEM02470
      DO 1000 ITR = 1 , NOTR                                                  SEM02480
        VM = VMIUO*MIUR(ITR)                                                  SEM02490
        I1 = NV(ITR,1)                                                        SEM02500
        I2 = NV(ITR,2)                                                        SEM02510
        I3 = NV(ITR,3)                                                        SEM02520
C       WRITE(9,111) ITR,I1,I2,I3,SIGMA(ITR),MIUR(ITR)                        SEM02530
        DET = P(X(I1),Y(I1),X(I2),Y(I2),X(I3),Y(I3))                          SEM02540
        IF (DET .EQ. 0.0D0) GO TO 8888                                        SEM02550
        AR = ABS(DET)/2.0D0                                                   SEM02560
        R12 = (X(I1) - X(I2))*(X(I1) - X(I2))                                 SEM02570
     &        + (Y(I1) - Y(I2))*(Y(I1) - Y(I2))                               SEM02580
        R13 = (X(I1) - X(I3))*(X(I1) - X(I3))                                 SEM02590
     &        + (Y(I1) - Y(I3))*(Y(I1) - Y(I3))                               SEM02600
        R23 = (X(I2) - X(I3))*(X(I2) - X(I3))                                 SEM02610
     &        + (Y(I2) - Y(I3))*(Y(I2) - Y(I3))                               SEM02620
        COSA1 = (R12 + R13 - R23)/(2.D0*DSQRT(R12*R13))                       SEM02630
        IF (COSA1 .GE. 1.0D0) WRITE(9,'(5HCOSA1,2X,F5.3)') COSA1              SEM02640
        COSA2 = (R12 + R23 - R13)/(2.D0*DSQRT(R12*R23))                       SEM02650
        IF (COSA2 .GE. 1.0D0) WRITE(9,'(5HCOSA2,2X,F5.3)') COSA2              SEM02660
        COSA3 = (R13 + R23 - R12)/(2.D0*DSQRT(R13*R23))                       SEM02670
        IF (COSA3 .GE. 1.0D0) WRITE(9,'(5HCOSA3,2X,F5.3)') COSA3              SEM02680
        COTA1 = COSA1/DSQRT(1.0D0 - COSA1*COSA1)                              SEM02690
        COTA2 = COSA2/DSQRT(1.0D0 - COSA2*COSA2)                              SEM02700
        COTA3 = COSA3/DSQRT(1.0D0 - COSA3*COSA3)                              SEM02710
        DO 1100 I = 1 , 3                                                     SEM02720
          DO 1100 J = 1 , 3                                                   SEM02730
            SE(I,J) = (COTA1*REAL(Q(1,I,J,1)) + COTA2*                        SEM02740
     &              REAL(Q(1,I,J,2)) + COTA3*REAL(Q(1,I,J,3)))/2.0D0          SEM02750
```

```
1100     CONTINUE                                                          SEM02760
         DO 1200 I = 1 , 3                                                 SEM02770
           L = NV(ITR,I)                                                   SEM02780
           IF(DIRP(L)) GO TO 1200                                          SEM02790
           DO 1300 K = 1 , 3                                               SEM02800
             LC = NV(ITR,K)                                                SEM02810
             IF (.NOT.DIRP(LC)) THEN                                       SEM02820
               LC      = LC - L + ML + 1                                   SEM02830
               S(L,LC) = S(L,LC) + SE(I,K)                                 SEM02840
               R(L,LC) = R(L,LC) + VM*SIGMA(ITR)*TQ(1,I,K)*AR/12.0D0       SEM02850
             ENDIF                                                         SEM02860
1300       CONTINUE                                                        SEM02870
1200     CONTINUE                                                          SEM02880
1000   CONTINUE                                                            SEM02890
C                                                                          SEM02900
C  >-------------------------------------------------------------------<  SEM02910
C  >     THE "S" & "R" MATRICES NEED TO BE SHIFTED UP BY NDIR          <  SEM02920
C  >        AND ALSO SIDEWAYS                                          <  SEM02930
C  >-------------------------------------------------------------------<  SEM02940
C                                                                          SEM02950
       DO 1400 I = 1 , NOPTL                                               SEM02960
         IF (DIRP(I)) GO TO 1400                                           SEM02970
         DO 1500 K = 1 , ML + MU + 1                                       SEM02980
           IF (K .EQ. 1) K1 = 0                                            SEM02990
1800       CONTINUE                                                        SEM03000
           L = K + I - ML - 1 + K1                                         SEM03010
           IF (L .LT. 1  .OR.  K .EQ. ML+1) GO TO 1500                     SEM03020
           IF (L .GT. NOPTL) GO TO 1400                                    SEM03030
           IF (DIRP(L)) THEN                                               SEM03040
             IF (K .EQ. 1) THEN                                            SEM03050
               R(I,K) = 0.0D0                                              SEM03060
               S(I,K) = 0.0D0                                              SEM03070
             ELSE IF (K .EQ. ML+MU+1) THEN                                 SEM03080
               R(I,K) = 0.0D0                                              SEM03090
               S(I,K) = 0.0D0                                              SEM03100
             ELSE IF (K .LT. ML+1) THEN                                    SEM03110
               DO 1600 J = K , 2 , -1                                      SEM03120
                 R(I,J) = R(I,J-1)                                         SEM03130
                 S(I,J) = S(I,J-1)                                         SEM03140
1600           CONTINUE                                                    SEM03150
               R(I,1) = 0.0D0                                              SEM03160
               S(I,1) = 0.0D0                                              SEM03170
             ELSE IF(K .GT. ML+1) THEN                                     SEM03180
               DO 1700 J = K , ML+MU                                       SEM03190
                 R(I,J) = R(I,J+1)                                         SEM03200
                 S(I,J) = S(I,J+1)                                         SEM03210
1700           CONTINUE                                                    SEM03220
               R(I,ML+MU+1) = 0.0D0                                        SEM03230
               S(I,ML+MU+1) = 0.0D0                                        SEM03240
               K1 = K1 + 1                                                 SEM03250
               GO TO 1800                                                  SEM03260
             ENDIF                                                         SEM03270
           ENDIF                                                           SEM03280
1500     CONTINUE                                                          SEM03290
1400   CONTINUE                                                            SEM03300
```

```
      M1 = 0                                                              SEM03310
      DO 1900 I = 1 , NOPTL                                               SEM03320
        IF (.NOT. DIRP(I)) THEN                                           SEM03330
          DO 2000 K = 1 , ML + MU + 1                                     SEM03340
            R(I-M1,K) = R(I,K)                                            SEM03350
            S(I-M1,K) = S(I,K)                                            SEM03360
2000      CONTINUE                                                        SEM03370
        ELSE                                                              SEM03380
          M1 = M1 + 1                                                     SEM03390
        ENDIF                                                             SEM03400
1900  CONTINUE                                                            SEM03410
C                                                                         SEM03420
C  >-----------------------------------------------------------------<    SEM03430
C  >     SET-UP THE INITIAL VALUES FOR THE CALL TO "DDASSL"           <    SEM03440
C  >-----------------------------------------------------------------<    SEM03450
C                                                                         SEM03460
      DO 2100 I = 1 , NN                                                  SEM03470
        A(I)  = 0.0D+00                                                   SEM03480
        AP(I) = 0.0D+00                                                   SEM03490
2100  CONTINUE                                                            SEM03500
      DO 2200 I = 1 , 15                                                  SEM03510
        INFO(I)  = 0                                                      SEM03520
2200  CONTINUE                                                            SEM03530
      INFO(5)  = 1                                                        SEM03540
      INFO(6)  = 1                                                        SEM03550
      IWORK(1) = ML                                                       SEM03560
      IWORK(2) = MU                                                       SEM03570
      NEQ  = NN                                                           SEM03580
      RTOL = 0.001D0                                                      SEM03590
      ATOL = 0.001D0*EO                                                   SEM03600
      LRW  = 40 + 9*NUNK + (2*MLMX+MUMX+1)*NUNK                           SEM03610
C     LRW  = 40 + 9*NUNK + NUNK*NUNK                                      SEM03620
      LIW  = 20 + NUNK                                                    SEM03630
      IPAR = NEQ                                                          SEM03640
C                                                                         SEM03650
C  >-----------------------------------------------------------------<    SEM03660
C  >     SOLVE WITH "DDASSL", THE SUBROUTINE WRITTEN BY               <    SEM03670
C  >         LINDA R. PETZOLD                                         <    SEM03680
C  >-----------------------------------------------------------------<    SEM03690
C                                                                         SEM03700
      DO 2300 I = 2 , 30 + N1 + NINT                                      SEM03710
        CALL DDASSL(RES,NEQ,T(I-1),A,AP,T(I),INFO,RTOL,ATOL,IDID,RWORK,   SEM03720
     &              LRW,IWORK,LIW,RPAR,IPAR,JAC)                          SEM03730
        IF (IDID .LT. 0) THEN                                             SEM03740
          GO TO 2500                                                      SEM03750
        ELSE                                                              SEM03760
          IF (IOUT .EQ. 0) THEN                                           SEM03770
            CALL OUTP(T(I))                                               SEM03780
          ELSE                                                            SEM03790
            DO 2400 J = 1 , NINT                                          SEM03800
              IF (T(I) .GE. 0.999D0*TINT(J)  .AND.  T(I) .LE. 1.001D0*    SEM03810
     &            TINT(J)) CALL OUTP(T(I))                              SEM03820
2400        CONTINUE                                                      SEM03830
          ENDIF                                                           SEM03840
        ENDIF                                                             SEM03850
```

```
 2300  CONTINUE                                                          SEM03860
       STOP                                                              SEM03870
 2500  CONTINUE                                                          SEM03880
       WRITE(6,333) IDID                                                 SEM03890
       WRITE(9,333) IDID                                                 SEM03900
       STOP                                                              SEM03910
 8888  CONTINUE                                                          SEM03920
       WRITE(9,'(5X,23HDETERMINANT OF TRIANGLE,I4,10H = 0.0!!!!)' ) ITR  SEM03930
       WRITE(6,'(5X,23HDETERMINANT OF TRIANGLE,I4,10H = 0.0!!!!)' ) ITR  SEM03940
       STOP                                                              SEM03950
 5     FORMAT(8(1X,E8.2))                                                SEM03960
 111   FORMAT(1X,I3,3(1X,I4),7X,E14.7,4X,E10.4)                          SEM03970
 222   FORMAT(//,5X,'HERE ARE SOME CONTROLLING VARIABLES',/)             SEM03980
 333   FORMAT(//,5X,'AN ERROR HAS OCCURED IN DDASSL, IDID = ',I4)        SEM03990
       END                                                               SEM04000
C                                                                        SEM04010
C                                                                        SEM04020
C  >-----------------------------------------------------------------<  SEM04030
C  >                                                                 <  SEM04040
C  >                  SUBROUTINE OUTP                                <  SEM04050
C  >                                                                 <  SEM04060
C  >       THIS SUBROUTINE PRINTS OUT THE RESULTS                    <  SEM04070
C  >                                                                 <  SEM04080
C  >-----------------------------------------------------------------<  SEM04090
C                                                                        SEM04100
C                                                                        SEM04110
      SUBROUTINE OUTP(TIME)                                              SEM04120
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                     SEM04130
      IMPLICIT  REAL*8(A-H,O-Z)                                          SEM04140
C  >-------- VALUES FROM SUBROUTINE GETINF ------------------------<     SEM04150
      REAL*8    PVAIRG(5),VAIRG(5),HE,LE,WE                              SEM04160
      REAL*8    SIG,MIURW,EO,ALPHA,BETA,TETAS                            SEM04170
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)                     SEM04180
      INTEGER   NVAIRG,TM,SCAT,NREG,IOUT,NINT                            SEM04190
C  >-------- VALUES FROM TRIANG SUBROUTINE ------------------------<     SEM04200
      REAL*8    X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS             SEM04210
      INTEGER   NV(NTRI,3)                                               SEM04220
      LOGICAL*4 DIRP(NUNK)                                               SEM04230
C  >-------- VALUES FROM PROGRAM SEMPTDDA -------------------------<     SEM04240
      REAL*8    TIME,A(NUNK),EFLD                                        SEM04250
C                                                                        SEM04260
C  >-----------------------------------------------------------------<  SEM04270
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE               <  SEM04280
C  >               SUBROUTINES AND FUNCTIONS                         <  SEM04290
C  >-----------------------------------------------------------------<  SEM04300
C                                                                        SEM04310
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,     SEM04320
     &          BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT     SEM04330
      COMMON   /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU,DIRPSEM04340
      COMMON   /OUT/A                                                    SEM04350
C                                                                        SEM04360
      IF (TM .EQ. 0) THEN                                                SEM04370
        WRITE(9,111) TIME                                                SEM04380
        WRITE(9,222)                                                     SEM04390
                                                                         SEM04400
```

```
      ELSE                                                                  SEM04410
        WRITE(9,333) TIME                                                   SEM04420
        WRITE(9,444)                                                        SEM04430
      ENDIF                                                                 SEM04440
      N1 = 1                                                                SEM04450
      DO 100 I = 1 , NOPTL                                                  SEM04460
        IF (NREG .EQ. 0) THEN                                               SEM04470
C                                                                           SEM04480
C >-------------------------------------------------------------------<     SEM04490
C >     THE USER WANTS TO PRINT OUT THE WHOLE REGION                  <     SEM04500
C >-------------------------------------------------------------------<     SEM04510
C                                                                           SEM04520
         IF (DIRP(I)) THEN                                                  SEM04530
            IF (Y(I) .EQ. 0.0D0  .AND.  TM .EQ. 0) THEN                     SEM04540
              WRITE(9,555) X(I),Y(I),0.0D0                                  SEM04550
            ELSE                                                            SEM04560
              WRITE(9,555) X(I),Y(I),EFLD(TIME,X(I),Y(I),0)                 SEM04570
            ENDIF                                                           SEM04580
         ELSE                                                               SEM04590
           WRITE(9,555) X(I),Y(I),A(N1)                                     SEM04600
           N1 = N1 + 1                                                      SEM04610
         ENDIF                                                              SEM04620
       ELSE                                                                 SEM04630
C                                                                           SEM04640
C >-------------------------------------------------------------------<     SEM04650
C >     THE USER WANTS TO PRINT OUT ONLY PARTS OF THE REGION          <     SEM04660
C >-------------------------------------------------------------------<     SEM04670
C                                                                           SEM04680
         DO 200 J = 1 , NREG                                                SEM04690
           IF ((X(I) .GE. X1(J)  .AND.  X(I) .LE. X2(J))  .AND.             SEM04700
     &         (Y(I) .GE. Y1(J)  .AND.  Y(I) .LE. Y2(J)))  THEN           SEM04710
             IF (DIRP(I)) THEN                                              SEM04720
               IF (Y(I) .EQ. 0.0D0  .AND.  TM .EQ. 0) THEN                  SEM04730
                 WRITE(9,555) X(I),Y(I),0.0D0                               SEM04740
               ELSE                                                         SEM04750
                 WRITE(9,555) X(I),Y(I),EFLD(TIME,X(I),Y(I),0)              SEM04760
               ENDIF                                                        SEM04770
             ELSE                                                           SEM04780
               WRITE(9,555) X(I),Y(I),A(N1)                                 SEM04790
               N1 = N1 + 1                                                  SEM04800
             ENDIF                                                          SEM04810
200        CONTINUE                                                         SEM04820
       ENDIF                                                                SEM04830
100   CONTINUE                                                              SEM04840
111   FORMAT(////,10X,'/',30('-'),'/',/,10X,'/',30X,'/',/,10X,              SEM04850
     & '/   POSITION AND ASSOCIATED    /',                                 SEM04860
     &/,10X,'/',5X,'ELECTRIC FIELD (V/m)','                                 SEM04870
     & ,5X,'/',/,10X,'/',6X,'AT T = ',E10.4,7X,'/',/,10X,'/',30X,'/',/,     SEM04880
     & 10X,'/',30('-'),'/',//)                                              SEM04890
222   FORMAT(6X,'X',11X,'Y',15X,'|E|',/)                                    SEM04900
333   FORMAT(////,10X,'/',30('-'),'/',/,10X,'/',30X,'/',/,10X,              SEM04910
     & '/   POSITION AND ASSOCIATED    /',                                 SEM04920
     &/,10X,'/',5X,'MAGNETIC FIELD (A/m)','                                 SEM04930
     & ,5X,'/',/,10X,'/',6X,'AT T = ',E10.4,7X,'/',/,10X,'/',30X,'/',/,     SEM04940
                                                                            SEM04950
```

```
     & 10X,'/',30('-'),'/',//)                                          SEM04960
444   FORMAT(6X,'X',11X,'Y',15X,'|H|',/)                                SEM04970
555   FORMAT(2E12.5,E18.7)                                              SEM04980
      RETURN                                                            SEM04990
      END                                                               SEM05000
C                                                                       SEM05010
C                                                                       SEM05020
C  >-------------------------------------------------------------<      SEM05030
C  >                                                             <      SEM05040
C  >                  SUBROUTINE RES                             <      SEM05050
C  >                                                             <      SEM05060
C  >       THIS SUBROUTINE DEFINES THE DIFFERENTIAL/ALGEBRAIC    <      SEM05070
C  >          SYSTEM OF EQUATIONS TO BE SOLVED :                 <      SEM05080
C  >                 DELTA = G(T,A,AP)                           <      SEM05090
C  >                                                             <      SEM05100
C  >-------------------------------------------------------------<      SEM05110
C                                                                       SEM05120
C                                                                       SEM05130
      SUBROUTINE RES (TIME,A,AP,DELTA,IRES,RPAR,IPAR)                   SEM05140
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                    SEM05150
      IMPLICIT  REAL*8(A-H,O-Z)                                         SEM05160
C  >-------- VALUES FROM SUBROUTINE GETINF ---------------------<      SEM05170
      REAL*8    PVAIRG(5),VAIRG(5),HE,LE,WE                             SEM05180
      REAL*8    SIG,MIURW,EO,ALPHA,BETA,TETAS                           SEM05190
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)                    SEM05200
      INTEGER   NVAIRG,TM,SCAT,NREG,IOUT,NINT                           SEM05210
C  >-------- VALUES FROM TRIANG SUBROUTINE ---------------------<      SEM05220
      REAL*8    X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS            SEM05230
      INTEGER   NV(NTRI,3)                                              SEM05240
      LOGICAL*4 DIRP(NUNK)                                              SEM05250
C  >-------- VALUES FROM PROGRAM SEMPTDDA ----------------------<      SEM05260
      REAL*8   R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1)                  SEM05270
      INTEGER  Q(4,15,15,3),TQ(4,15,15)                                 SEM05280
C  >-------- VALUES FOR SUBROUTINE RES -------------------------<      SEM05290
      REAL*8   TIME,A(NUNK),AP(NUNK),B(NUNK),DELTA(IPAR),SE(3,3),EFLD   SEM05300
C                                                                       SEM05310
C  >-------------------------------------------------------------<      SEM05320
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE           <      SEM05330
C  >              SUBROUTINES AND FUNCTIONS                      <      SEM05340
C  >-------------------------------------------------------------<      SEM05350
C                                                                       SEM05360
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,    SEM05370
     &          BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT    SEM05380
      COMMON   /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU,DIRPSEM05390
      COMMON   /RAS/R,S,NEQ/SIMPL/Q,TQ                                  SEM05400
C                                                                       SEM05410
      P(A1,B1,A2,B2,A3,B3) = A1*B2-A2*B1+A2*B3-A3*B2+A3*B1-A1*B3        SEM05420
C                                                                       SEM05430
      PI    = DACOS(-1.0D0)                                             SEM05440
      VMIUO = 4.0D-07*PI                                                SEM05450
      CO    = 3.0D08                                                    SEM05460
      DO 100 I = 1 , NUNK                                               SEM05470
        B(I)    = 0.0D0                                                 SEM05480
100   CONTINUE                                                          SEM05490
                                                                        SEM05500
```

```
C                                                                       SEM05510
C  >-----------------------------------------------------------------<  SEM05520
C  >     THE "B" VECTOR (THE SOURCE VECTOR) NEED TO BE SET-UP        <  SEM05530
C  >           AT "TIME"                                             <  SEM05540
C  >-----------------------------------------------------------------<  SEM05550
C                                                                       SEM05560
      DO 200 ITR = 1 , NOTR                                             SEM05570
        VM = VMIUO*MIUR(ITR)                                            SEM05580
        I1 = NV(ITR,1)                                                  SEM05590
        I2 = NV(ITR,2)                                                  SEM05600
        I3 = NV(ITR,3)                                                  SEM05610
        DET = P(X(I1),Y(I1),X(I2),Y(I2),X(I3),Y(I3))                    SEM05620
        IF (DET .EQ. 0.0D0) GO TO 9999                                  SEM05630
        AR = ABS(DET)/2.0D0                                             SEM05640
        R12 = (X(I1) - X(I2))*(X(I1) - X(I2))                           SEM05650
     &       + (Y(I1) - Y(I2))*(Y(I1) - Y(I2))                          SEM05660
        R13 = (X(I1) - X(I3))*(X(I1) - X(I3))                           SEM05670
     &       + (Y(I1) - Y(I3))*(Y(I1) - Y(I3))                          SEM05680
        R23 = (X(I2) - X(I3))*(X(I2) - X(I3))                           SEM05690
     &       + (Y(I2) - Y(I3))*(Y(I2) - Y(I3))                          SEM05700
        COSA1 = (R12 + R13 - R23)/(2.D0*DSQRT(R12*R13))                 SEM05710
        IF (COSA1 .GE. 1.0D0) WRITE(9,'(5HCOSA1,2X,F5.3)') COSA1        SEM05720
        COSA2 = (R12 + R23 - R13)/(2.D0*DSQRT(R12*R23))                 SEM05730
        IF (COSA2 .GE. 1.0D0) WRITE(9,'(5HCOSA2,2X,F5.3)') COSA2        SEM05740
        COSA3 = (R13 + R23 - R12)/(2.D0*DSQRT(R13*R23))                 SEM05750
        IF (COSA3 .GE. 1.0D0) WRITE(9,'(5HCOSA3,2X,F5.3)') COSA3        SEM05760
        COTA1 = COSA1/DSQRT(1.0D0 - COSA1*COSA1)                        SEM05770
        COTA2 = COSA2/DSQRT(1.0D0 - COSA2*COSA2)                        SEM05780
        COTA3 = COSA3/DSQRT(1.0D0 - COSA3*COSA3)                        SEM05790
        DO 300 I = 1 , 3                                                SEM05800
          DO 300 J = 1 , 3                                              SEM05810
            SE(I,J) = (COTA1*REAL(Q(1,I,J,1)) + REAL(COTA2*Q(1,I,J,2)) +SEM05820
     &                COTA3*REAL(Q(1,I,J,3)))/2.0D0                     SEM05830
300     CONTINUE                                                        SEM05840
        DO 400 I = 1 , 3                                                SEM05850
          L = NV(ITR,I)                                                 SEM05860
          IF(DIRP(L)) GO TO 400                                         SEM05870
          DO 500 K = 1 , 3                                              SEM05880
            LC = NV(ITR,K)                                              SEM05890
            IF (DIRP(LC)) THEN                                          SEM05900
               IF ((Y(LC) .NE. 0.0D0  .AND.  TM .EQ. 0)  .OR.           SEM05910
     &              TM .EQ. 1) THEN                                     SEM05920
             B(L) = B(L) - EFLD(TIME,X(LC),Y(LC),0)*SE(I,K) -           SEM05930
     &                    EFLD(TIME,X(LC),Y(LC),1)*                    SEM05940
     &                    VM*SIGMA(ITR)*TQ(1,I,K)*AR/12.0D0            SEM05950
            ENDIF                                                       SEM05960
          ENDIF                                                         SEM05970
500       CONTINUE                                                      SEM05980
400     CONTINUE                                                        SEM05990
200   CONTINUE                                                          SEM06000
C                                                                       SEM06010
C  >-----------------------------------------------------------------<  SEM06020
C  >     THE "B" VECTOR NEED TO BE SHIFTED UP BY NDIR                <  SEM06030
C  >-----------------------------------------------------------------<  SEM06040
C                                                                       SEM06050
```

```
      N1 = 0                                                              SEM06060
      DO 600 I = 1 , NOPTL                                                SEM06070
        IF (.NOT. DIRP(I)) THEN                                           SEM06080
          B(I-N1) = B(I)                                                  SEM06090
        ELSE                                                              SEM06100
          N1 = N1 + 1                                                     SEM06110
        ENDIF                                                             SEM06120
600   CONTINUE                                                            SEM06130
C                                                                         SEM06140
C  >-------------------------------------------------------------------<  SEM06150
C  >     WE THEN CALCULATE  DELTA = G(TIME,A,AP)                       <  SEM06160
C  >-------------------------------------------------------------------<  SEM06170
C                                                                         SEM06180
      DO 800 I = 1 , NEQ                                                  SEM06190
        SUM = 0.0D0                                                       SEM06200
        DO 900 J = 1 , NEQ                                                SEM06210
          K = J - I + ML + 1                                              SEM06220
          IF (K .GT. 0  .AND.  K .LE. ML+MU+1) THEN                       SEM06230
            SUM = SUM + R(I,K)*AP(J) + S(I,K)*A(J)                        SEM06240
          ENDIF                                                           SEM06250
900     CONTINUE                                                          SEM06260
        DELTA(I) = SUM - B(I)                                             SEM06270
800   CONTINUE                                                            SEM06280
      RETURN                                                              SEM06290
9999  CONTINUE                                                            SEM06300
      WRITE(9,*) 'ITR = ',ITR                                             SEM06310
      WRITE(9,*) 'ITR = ',ITR                                             SEM06320
      STOP                                                                SEM06330
      END                                                                 SEM06340
C                                                                         SEM06350
C                                                                         SEM06360
C  >-------------------------------------------------------------------<  SEM06370
C  >                                                                  <  SEM06380
C  >                  SUBROUTINE JAC                                  <  SEM06390
C  >                  ==========                                      <  SEM06400
C  >                                                                  <  SEM06410
C  >       THIS SUBROUTINE DEFINES THE JACOBIAN, THE MATRIX OF        <  SEM06420
C  >          PARTIAL DERIVATIVES :                                   <  SEM06430
C  >                    PD = DG/DA + CJ*DG/DAP                        <  SEM06440
C  >-------------------------------------------------------------------<  SEM06450
C                                                                         SEM06460
C                                                                         SEM06470
      SUBROUTINE JAC (T,A,AP,PD,CJ,RPAR,IPAR)                             SEM06480
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                      SEM06490
      IMPLICIT  REAL*8(A-H,O-Z)                                           SEM06500
C  >--------- VALUES FROM TRIANG SUBROUTINE ---------------------------<  SEM06510
      REAL*8    X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS            SEM06520
      INTEGER   NV(NTRI,3)                                                SEM06530
      LOGICAL*4 DIRP(NUNK)                                                SEM06540
C  >--------- VALUES FOR SUBROUTINE JAC -------------------------------<  SEM06550
      REAL*8    A(NUNK),AP(NUNK),PD(2*ML+MU+1,IPAR)                       SEM06560
      REAL*8    R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1),T,CJ              SEM06570
C                                                                         SEM06580
C  >-------------------------------------------------------------------<  SEM06590
                                                                          SEM06600
```

```
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                  < SEM06610
C  >               SUBROUTINES AND FUNCTIONS                           < SEM06620
C  >-------------------------------------------------------------------< SEM06630
C                                                                        SEM06640
      COMMON   /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU,DIRPSEM06650
      COMMON   /RAS/R,S,NEQ                                              SEM06660
C                                                                        SEM06670
C                                                                        SEM06680
C  >-------------------------------------------------------------------< SEM06690
C  >     AS THE "S" AND "R" MATRICES ARE INDEPENDENT OF TIME,           < SEM06700
C  >        THE JACOBIAN IS ALSO TIME INDEPENDENT                       < SEM06710
C  >-------------------------------------------------------------------< SEM06720
C                                                                        SEM06730
      DO 100 I = 1 , NEQ                                                 SEM06740
        DO 100 J = 1 , NEQ                                               SEM06750
          K = J - I + ML + 1                                             SEM06760
          IF (K .GT. 0  .AND.  K .LE. ML+MU+1) THEN                      SEM06770
            IROW = I - J + ML + MU + 1                                   SEM06780
            PD(IROW,J) = S(I,K) + CJ*R(I,K)                              SEM06790
          ENDIF                                                          SEM06800
100   CONTINUE                                                           SEM06810
      RETURN                                                             SEM06820
      END                                                                SEM06830
C                                                                        SEM06840
C                                                                        SEM06850
C  >-------------------------------------------------------------------< SEM06860
C  >                                                                    < SEM06870
C  >                    FUNCTION EFLD                                   < SEM06880
C  >                                                                    < SEM06890
C  >                                                                    < SEM06900
C  >       THIS FUNCTION CALCULATES THE FIELD, AS A FUNCTION            < SEM06910
C  >         TIME, ON THE SPHERICAL BOUNDARY                            < SEM06920
C  >                                                                    < SEM06930
C  >-------------------------------------------------------------------< SEM06940
C                                                                        SEM06950
          FUNCTION EFLD(TIME,XA,YA,IDER)                                 SEM06960
C                                                                        SEM06970
      IMPLICIT  REAL*8(A-H,O-Z)                                          SEM06980
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                     SEM06990
C  >--------- VALUES FROM SUBROUTINE GETINF ------------------------------< SEM07000
      REAL*8     PVAIRG(5),VAIRG(5),HE,LE,WE                             SEM07010
      REAL*8     SIG,MIURW,EO,ALPHA,BETA,TETAS                           SEM07020
      REAL*8     X1(10),Y1(10),X2(10),Y2(10),TINT(50)                    SEM07030
      INTEGER    NVAIRG,TM,SCAT,NREG,IOUT,NINT                           SEM07040
C  >--------- VALUES FROM TRIANG SUBROUTINE ------------------------------< SEM07050
      REAL*8     X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS            SEM07060
      INTEGER    NV(NTRI,3)                                              SEM07070
      LOGICAL*4 DIRP(NUNK)                                               SEM07080
C  >--------- VALUES FOR EFLD FUNCTION -----------------------------------< SEM07090
      REAL*8    TIME,XA,YA,PHI,HP,FACT(5),RF(5),TD(5),EFLD               SEM07100
C                                                                        SEM07110
C  >-------------------------------------------------------------------< SEM07120
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                  < SEM07130
C  >               SUBROUTINES AND FUNCTIONS                            < SEM07140
                                                                         < SEM07150
```

```
C >-------------------------------------------------------------< SEM07160
C                                                                  SEM07170
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,  SEM07180
     &          BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT  SEM07190
      COMMON   /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU,DIRPSEM07200
     &          PIMN3,PIMX3                                          SEM07210
C                                                                  SEM07220
      CO  = 3.0D08                                                 SEM07230
      PHI = DATAN2(YA,XA)                                          SEM07240
      DO 100 I = 1 , 5                                             SEM07250
        FACT(I) = 0.0D0                                            SEM07260
100   CONTINUE                                                     SEM07270
      IF (TM .EQ. 0) THEN                                          SEM07280
        RF(1) = 1.0D0                                              SEM07290
        RF(2) = 0.0D0                                              SEM07300
        RF(3) = -1.0D0                                             SEM07310
        RF(4) = 0.0D0                                              SEM07320
        RF(5) = -1.0D0                                             SEM07330
      ELSE                                                         SEM07340
        RF(1) = 1.0D0                                              SEM07350
        RF(2) = 1.0D0                                              SEM07360
        RF(3) = 1.0D0                                              SEM07370
        RF(4) = 1.0D0                                              SEM07380
        RF(5) = 1.0D0                                              SEM07390
      ENDIF                                                        SEM07400
      IF (SCAT .EQ. 0) THEN                                        SEM07410
        I1      = 2                                                SEM07420
        TD(1)   = (RS-RB*DCOS(TETAS - PHI))/CO                     SEM07430
        FACT(1) = 1.0D0                                            SEM07440
        TD(2)   = (RS-RB*DCOS(TETAS + PHI))/CO                     SEM07450
        FACT(2) = 1.0D0                                            SEM07460
      ELSE                                                         SEM07470
        I1 = 5                                                     SEM07480
        IF (PHI .GE. PIMN1) THEN                                   SEM07490
          TD(1)   = (RS-RB*DCOS(TETAS - PHI))/CO                   SEM07500
          FACT(1) = 1.0D0                                          SEM07510
        ENDIF                                                      SEM07520
        IF (PHI .GE. PIMMN1  .OR.  PHI .LE. PIMMX1) THEN           SEM07530
          TD(2)   = (RS-RB*DCOS(TETAS + PHI))/CO                   SEM07540
          FACT(2) = 1.0D0                                          SEM07550
        ENDIF                                                      SEM07560
        IF (PHI .GE. PIMN2)  THEN                                  SEM07570
          HI      = DSQRT(LE**2 + (-(LE + RB*DCOS(PHI))*DTAN(TETAS) +  SEM07580
     &              RB*DSIN(PHI))**2)                            SEM07590
          TD(3)   = ((LE + RB*DCOS(PHI))/DCOS(TETAS) + RS -        SEM07600
     &              HI*DCOS(TETAS - PI/2.0D0 - DASIN(LE/HI)))/CO   SEM07610
          FACT(3) = 1.0D0                                          SEM07620
        ENDIF                                                      SEM07630
        IF (PHI .GE. PIMMN2  .AND.  PHI .LE. PIMMX2)  THEN         SEM07640
          HIM     = DSQRT(LE**2 + ((LE + RB*DCOS(PHI))*DTAN(TETAS) +   SEM07650
     &              RB*DSIN(PHI))**2)                            SEM07660
          TD(4)   = ((LE + RB*DCOS(PHI))/DCOS(TETAS) + RS -        SEM07670
     &              HIM*DCOS(PI - TETAS - DACOS(LE/HIM)))/CO       SEM07680
          FACT(4) = 1.0D0                                          SEM07690
                                                                   SEM07700
```

```
          ENDIF                                                          SEM07710
          IF (PHI .GE. PIMMN3  .AND.  PHI .LE. PIMMX3)  THEN             SEM07720
            HP      = DSQRT(HE**2 + ((RB*DSIN(PHI) - HE)*DCOTAN(TETAS) + SEM07730
     &                RB*DSIN(PHI))**2)                                  SEM07740
            TD(5)   = ((RB*DSIN(PHI) - HE)/DSIN(TETAS) + RS -            SEM07750
     &                HP*DCOS(PI - TETAS - DASIN(HE/HP)))/CO             SEM07760
            FACT(5) = 1.0D0                                              SEM07770
          ENDIF                                                          SEM07780
        ENDIF                                                            SEM07790
C                                                                        SEM07800
        SUM = 0.0D0                                                      SEM07810
        DO 400 I = 1 , I1                                                SEM07820
          IF ((TIME - TD(I)) .GT. 0.0D0  .AND.  FACT(I) .NE. 0.0D0) THEN SEM07830
            IF (IDER .EQ. 0) THEN                                        SEM07840
              IF (-BETA*(TIME-TD(I)) .LE. -100.0D0) THEN                 SEM07850
                SUM = SUM + RF(I)*DEXP(-ALPHA*(TIME - TD(I)))            SEM07860
              ELSE                                                       SEM07870
                SUM = SUM + RF(I)*(DEXP(-ALPHA*(TIME - TD(I))) -         SEM07880
     &                DEXP(-BETA*(TIME-TD(I))))                          SEM07890
              ENDIF                                                      SEM07900
            ELSEIF (IDER .EQ. 1) THEN                                    SEM07910
              IF (-BETA*(TIME-TD(I)) .LE. -100.0D0) THEN                 SEM07920
                SUM = SUM + RF(I)*(-ALPHA*DEXP(-ALPHA*(TIME - TD(I))))   SEM07930
              ELSE                                                       SEM07940
                SUM = SUM + RF(I)*(-ALPHA*DEXP(-ALPHA*(TIME - TD(I))) +  SEM07950
     &                BETA*DEXP(-BETA*(TIME-TD(I))))                     SEM07960
              ENDIF                                                      SEM07970
            ENDIF                                                        SEM07980
          ENDIF                                                          SEM07990
400     CONTINUE                                                         SEM08000
        EFLD = EO*SUM                                                    SEM08010
        RETURN                                                           SEM08020
        END                                                              SEM08030
C                                                                        SEM08040
C                                                                        SEM08050
C  >-------------------------------------------------------------------< SEM08060
C  >                                                                   < SEM08070
C  >                  SUBROUTINE PHICAL                                < SEM08080
C  >                                                                   < SEM08090
C  >    THIS SUBROUTINE CALCULATES THE CRITICAL SCATTERING ANGLES      < SEM08100
C  >                                                                   < SEM08110
C  >-------------------------------------------------------------------< SEM08120
C                                                                        SEM08130
C                                                                        SEM08140
      SUBROUTINE PHICAL                                                  SEM08150
C                                                                        SEM08160
      IMPLICIT  REAL*8(A-H,O-Z)                                          SEM08170
      PARAMETER (NUNK=680,NTRI=1280,MLMX=90,MUMX=90)                     SEM08180
C  >-------- VALUES FROM SUBROUTINE GETINF ---------------------------< SEM08190
      REAL*8    PVAIRG(5),VAIRG(5),HE,LE,WE                            SEM08200
      REAL*8    SIG,MIURW,EO,ALPHA,BETA,TETAS                          SEM08210
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)                   SEM08220
      INTEGER   NVAIRG,TM,SCAT,NREG,IOUT,NINT                          SEM08230
C  >-------- VALUES FROM TRIANG SUBROUTINE ---------------------------< SEM08240
                                                                         SEM08250
```

```
      REAL*8    X(NUNK),Y(NUNK),SIGMA(NTRI),MIUR(NTRI),RB,RS                  SEM08260
      INTEGER   NV(NTRI,3)                                                    SEM08270
      LOGICAL*4 DIRP(NUNK)                                                    SEM08280
C                                                                             SEM08290
C     >-------------------------------------------------------------<         SEM08300
C     >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE           <         SEM08310
C     >              SUBROUTINES AND FUNCTIONS                      <         SEM08320
C     >-------------------------------------------------------------<         SEM08330
C                                                                             SEM08340
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,          SEM08350
     &          BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT          SEM08360
      COMMON   /TRIAN/X,Y,SIGMA,MIUR,RB,RS,NOTR,NOPTL,NDIR,NV,ML,MU,DIRP     SEM08370
      COMMON   /PHI/PIMN1,PIMMN1,PIMMX1,PIMN2,PIMMN2,PIMMX2,                  SEM08380
     &          PIMN3,PIMX3                                                   SEM08390
C                                                                             SEM08400
      PI  = DACOS(-1.0D0)                                                     SEM08410
      LH2 = DSQRT(LE**2 + HE**2)                                              SEM08420
C                                                                             SEM08430
      PIMN1  = TETAS - PI + DASIN((LH2*DSIN(DATAN(LE/HE) + TETAS -            SEM08440
     &          PI/2.0D0))/RB)                                                SEM08450
      PIMMN1 = PI - TETAS - DASIN(((LE - HE*DCOTAN(TETAS))*DSIN(TETAS))/      SEM08460
     &          RB)                                                           SEM08470
      PIMMX1 = PI - TETAS + DASIN(((LE - HE*DCOTAN(TETAS))*DSIN(TETAS))/      SEM08480
     &          RB)                                                           SEM08490
      PIMN2  = -TETAS - DASIN((LH2*DSIN(DATAN(LE/HE) + TETAS -                SEM08500
     &          PI/2.0D0))/RB)                                                SEM08510
      PIMMN2 = TETAS - DASIN((LH2*DSIN(DATAN(LE/HE) - TETAS +                 SEM08520
     &          1.50D0*PI))/RB)                                               SEM08530
      PIMMX2 = TETAS + DASIN((LH2*DSIN(DATAN(LE/HE) + TETAS -                 SEM08540
     &          PI/2.0D0))/RB)                                                SEM08550
      PIMMN3 = PI - TETAS + DASIN((LH2*DSIN(DATAN(LE/HE) - TETAS +            SEM08560
     &          1.50D0*PI))/RB)                                               SEM08570
      PIMMX3 = PI - TETAS + DASIN((LH2*DSIN(DATAN(HE/LE) - TETAS +            SEM08580
     &          PI))/RB)                                                      SEM08590
C                                                                             SEM08600
      RETURN                                                                  SEM08610
      END                                                                     SEM08620
```

```
      SUBROUTINE GETINF
C
C  >-----------------------------------------------------------------<
C  >     THIS SUBROUTINE READS THE REQUIRED INFORMATION EITHER FROM   <
C  >      THE TERMINAL OR AN INPUT FILE                               <
C  >-----------------------------------------------------------------<
C
      IMPLICIT  REAL*8(A-H,O-Z)
      REAL*8    PVAIRG(5),VAIRG(5),HE,LE,WE
      REAL*8    SIG,MIURW,EO,ALPHA,BETA,TETAS
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)
      INTEGER   NVAIRG,TM,SCAT,NREG,IOUT,NINT
      CHARACTER*1 FILE,CHAR*1
C
C  >-----------------------------------------------------------------<
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE MAIN           <
C  >              PROGRAM                                             <
C  >-----------------------------------------------------------------<
C
      COMMON    /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,EO,TINT,ALPHA,
     &           BETA,TETAS,X1,Y1,X2,Y2,NVAIRG,TM,SCAT,NREG,IOUT,NINT
C
      PI     = DACOS(-1.0D0)
      NVAIRG = 0
C
C  >-----------------------------------------------------------------<
C  >     THE INFORMATION IS READ FROM THE TERMINAL IF THE ANSWER IS   <
C  >      NO OTHERWISE IT IS FOUND IN AN EXISTING FILE                <
C  >-----------------------------------------------------------------<
C
      WRITE(6,*) 'IS THERE AN EXISTING INPUT FILE ? (Y/N)'
      READ(5,'(A1)') FILE
      IF ( FILE .EQ. 'N'  .OR.  FILE .EQ. 'N' ) THEN
         IUNIT=5
      ELSE
         IUNIT=7
      ENDIF

      WRITE(6,10)
      WRITE(6,*) '
      WRITE(6,*) '                                                    '
      WRITE(6,*) '                                                    '
      WRITE(6,*) '          ENTER THE ANGLE OF INCIDENCE, TETAS,      '
      WRITE(6,*) '           IN DEGREES                               '
      WRITE(6,*) '                                                    '
      WRITE(6,*) '                       /                            '
      WRITE(6,*) '                      /                             '
      WRITE(6,*) '                     /                              '
      WRITE(6,*) '                   /\                               '
      WRITE(6,*) '                  /   \                             '
      WRITE(6,*) '                 /     \                            '
      WRITE(6,*) '                /       \                           '
      WRITE(6,*) '               /         \                          '
      WRITE(6,*) '              /           \<--\                     '
      WRITE(6,*) '             /              \   \ TETAS             '
```

```
      WRITE(6,*) '                                 \    \                     '
      WRITE(6,*) '    ____________________________________________________      '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                       TETAS ? (in degrees)                   '
      READ(IUNIT,*) TETAS
C
      TETAS = TETAS/180.D0*PI
C
      WRITE(6,10)
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '          FOR WHAT MODE DO YOU WHICH TO ANALYSE               '
      WRITE(6,*) '           THE FIELD PENETRATION (TE = 0 ; TM = 1)            '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '     TM                  TE                                   '
      WRITE(6,*) ' **********          **********          |------------|       '
      WRITE(6,*) '   H                   E                 |   TE = 0   |       '
      WRITE(6,*) '    O---> E             O---> H          |   TM = 1   |       '
      WRITE(6,*) '                                         |------------|       '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                          TM ?                                '
      READ(IUNIT,*) TM
C
      WRITE(6,10)
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      IF (TM .EQ. 1) THEN
       WRITE(6,*) '        ENTER THE PEAK MAGNETIC FIELD STRENGTH,              '
       WRITE(6,*) '         IN AMPERES PER METERS (A/m)                         '
      ELSE
       WRITE(6,*) '        ENTER THE PEAK ELECTRIC FIELD STRENGTH,              '
       WRITE(6,*) '         IN VOLTS PER METERS (V/m)                           '
      ENDIF
      WRITE(6,*) '           THEN ALPHA & BETA, THE CHARACTERISTIC OF           '
      WRITE(6,*) '           EMP TYPE IMPINGING PLANE WAVE                      '
      WRITE(6,*) '                                                              '
      IF (TM .EQ. 1) THEN
       WRITE(6,*) '     h(t) = H0 * [exp(-ALPHA * t) - exp(-BETA * t)]          '
      ELSE
       WRITE(6,*) '     e(t) = E0 * [exp(-ALPHA * t) - exp(-BETA * t)]          '
      ENDIF
      WRITE(6,*) '             ?              ?                    ?          '
      WRITE(6,*) '                                                              '
      WRITE(6,*) '                                                              '
      IF (TM .EQ. 1) THEN
       WRITE(6,*) '                H0 (A/m), ALPHA , BETA ?                     '
      ELSE
       WRITE(6,*) '                E0 (V/m), ALPHA , BETA ?                     '
      ENDIF
      READ(IUNIT,*) E0,ALPHA,BETA
C
```

```
      WRITE(6,10)                                                       GET01110
      WRITE(6,*) '                                                     ' GET01120
      WRITE(6,*) '                                                     ' GET01130
      WRITE(6,*) '       ENTER THE "OUTSIDE" HORIZONTAL (LE), THEN     ' GET01140
      WRITE(6,*) '        VERTICAL (HE), DIMENSIONS OF THE ENCLOSURE ;' GET01150
      WRITE(6,*) '        AS WELL AS THE WIDTH OF THE WALL (WE).       ' GET01160
      WRITE(6,*) '                                                     ' GET01170
      WRITE(6,*) '                                                     ' GET01180
      WRITE(6,*) '                                                     ' GET01190
      WRITE(6,*) '                                                     ' GET01200
      WRITE(6,*) '                                                     ' GET01210
      WRITE(6,*) '                                                     ' GET01220
      WRITE(6,*) '                                                     ' GET01230
      WRITE(6,*) '                                          HE         ' GET01240
      WRITE(6,*) '                                                     ' GET01250
      WRITE(6,*) '                                                     ' GET01260
      WRITE(6,*) '                                                     ' GET01270
      WRITE(6,*) '                                                     ' GET01280
      WRITE(6,*) '    ---->| WE |<----                                 ' GET01290
      WRITE(6,*) '                                                     ' GET01300
      WRITE(6,*) '                                                     ' GET01310
      WRITE(6,*) '                        |<---- LE ---->|             ' GET01320
      WRITE(6,*) '                                                     ' GET01330
      WRITE(6,*) '                                                     ' GET01340
      WRITE(6,*) '                  LE , HE , WE ? (in meters)         ' GET01350
      READ(IUNIT,*) LE,HE,WE                                            GET01360
      WRITE(6,10)                                                       GET01370
      WRITE(6,*) '                                                     ' GET01380
      WRITE(6,*) '                                                     ' GET01390
      WRITE(6,*) '       IS THERE ANY APERTURES ON THE VERTICAL'        GET01400
      WRITE(6,*) '        WALLS OF THE SHIELD ? (Y/N)'                  GET01410
      READ(IUNIT,'(A1)') CHAR                                           GET01420
      I1 = 0                                                            GET01430
      IF(CHAR .EQ. 'Y'  .OR.  CHAR .EQ. 'y' ) THEN                      GET01440
       WRITE(6,*) '                                                   'GET01450
       WRITE(6,*) '                                                   'GET01460
       WRITE(6,*) '       ENTER THE NUMBER OF APERTURES ON ONE WALL    'GET01470
       WRITE(6,*) '        (THE 2 VERTICAL WALLS SHOULD BE SYMMETRICAL)'GET01480
100    WRITE(6,*) '                                                   'GET01490
       READ(IUNIT,*) NVAIRG          NVAIRG ?                         'GET01500
       IF (NVAIRG .GT. 3) THEN                                          GET01510
        WRITE(6,*) ' ',                                                 GET01520
        WRITE(6,*) '                                                   'GET01530
        WRITE(6,*) '       THE NUMBER OF MAXIMUM APERTURES IS THREE (3)'GET01540
        IF (I1 .GT. 3) STOP  PLEASE ENTER OVER THE NUMBER OF APERTURES 'GET01550
        I1 = I1 + 1                                                     GET01560
        GO TO 100                                                       GET01570
       ENDIF                                                            GET01580
C                                                                       GET01590
       IF (NVAIRG .EQ. 0) GO TO 200                                     GET01600
C                                                                       GET01610
       WRITE(6,10)                                                      GET01620
       WRITE(6,*) '                                                    'GET01630
       WRITE(6,*) '       ENTER THE Y-COORDINATE OF THE AIRGAPS, THEN  'GET01640
       WRITE(6,*) '        THE LENGTH OF THE ASSOCIATED AIRGAP         'GET01650
```

```
      WRITE(6,*) '                                                          'GET01660
      WRITE(6,*) '                                                          'GET01670
      WRITE(6,*) '                                                          'GET01680
      WRITE(6,*) '                                                          'GET01690
      WRITE(6,*) '                                                          'GET01700
      WRITE(6,*) '  POSITION                                    LENGTH OF   'GET01710
      WRITE(6,*) '  OF AIRGAP                                    AIRGAP   'GET01720
      WRITE(6,*) '   [PVAIRG(2)]                               [VAIRG(2)]  'GET01730
      WRITE(6,*) '                                                          'GET01740
      WRITE(6,*) '                                                          'GET01750
      WRITE(6,*) ' [PVAIRG(1)]                                  [VAIRG(1)]  'GET01760
      WRITE(6,*) '                                                          'GET01770
      WRITE(6,*) '                                                          'GET01780
      WRITE(6,*) '                                                          'GET01790
      WRITE(6,*) '                                                          'GET01800
      IF (NVAIRG .GE. 1) THEN                                               'GET01810
       WRITE(6,*) '       ENTER THE Y-COORDINATE & GAP OF THE APERTURE       GET01820
       IF (NVAIRG .GT. 1) THEN                                             'GET01830
       WRITE(6,*) '         CLOSEST TO THE GROUND PLANE                      GET01840
       ENDIF                                                              'GET01850
       WRITE(6,*) '                                                        'GET01860
       WRITE(6,*) '              PVAIRG(1) , VAIRG(1) ? (in meters)        'GET01870
       READ(IUNIT,*) PVAIRG(1),VAIRG(1)                                   'GET01880
      ENDIF                                                                 GET01890
      IF (NVAIRG .GE. 2) THEN                                               GET01900
       WRITE(6,*) '                                                        'GET01910
       IF (NVAIRG .EQ. 2) THEN                                            'GET01920
       WRITE(6,*) '     ENTER THE Y-COORDINATE & GAP OF TOP APERTURE        GET01930
       ELSE                                                               'GET01940
       WRITE(6,*) '     ENTER THE Y-COORDINATE & GAP OF MIDDLE APERTURE    'GET01950
       ENDIF                                                              'GET01960
       WRITE(6,*) '                                                        'GET01970
       WRITE(6,*) '              PVAIRG(2) , VAIRG(2) ? (in meters)        'GET01980
       READ(IUNIT,*) PVAIRG(2),VAIRG(2)                                   'GET01990
      ENDIF                                                                 GET02000
      IF (NVAIRG .EQ. 3) THEN                                               GET02010
       WRITE(6,*) '                                                        'GET02020
       WRITE(6,*) '     ENTER THE Y-COORDINATE & GAP OF TOP APERTURE       'GET02030
       WRITE(6,*) '                                                        'GET02040
       WRITE(6,*) '              PVAIRG(3) , VAIRG(3) ? (in meters)        'GET02050
       READ(IUNIT,*) PVAIRG(3),VAIRG(3)                                   'GET02060
      ENDIF                                                                 GET02070
      ENDIF                                                                 GET02080
C                                                                           GET02090
200   CONTINUE                                                              GET02100
C                                                                           GET02110
      WRITE(6,10)                                                           GET02120
      WRITE(6,*) '       ENTER THE CONDUCTIVITY AND RELATIVE                GET02130
      WRITE(6,*) '        PERMEABILITY OF THE WALL                        ' GET02140
      WRITE(6,*) '                                                        ' GET02150
      WRITE(6,*) '                                                        ' GET02160
      WRITE(6,*) '                    SIGMA , MIUR ?                      ' GET02170
      WRITE(6,*) '                                                        ' GET02180
      WRITE(6,*) '                                                        ' GET02190
      WRITE(6,*) '                                                        ' GET02200
```

```
      WRITE(6,*) '
      WRITE(6,*) '            |--|          |--|                         '
      WRITE(6,*) '            |  |          |  |                         '
      WRITE(6,*) '            |  |          |  |                         '
      WRITE(6,*) '   _________|__|__________|__|_________                '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '              SIGMA (S/m), MIUR ?                      '
      READ(IUNIT,*) SIG,MIURW
C
      WRITE(6,10)
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '     SHALL WE INCLUDE THE FIELD SCATTERED ON           '
      WRITE(6,*) '      THE ENCLOSURE WHEN CALCULATING THE FIELD         '
      WRITE(6,*) '      ON THE CIRCULAR BOUNDARY:                        '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '       - CALCULATE THE FIELD AS IF THERE IS            '
      WRITE(6,*) '          NO ENCLOSURE (SCAT = 0)                      '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                    \                                  '
      WRITE(6,*) '                     \                                 '
      WRITE(6,*) '         \            \                                '
      WRITE(6,*) '          \           .#  .                            '
      WRITE(6,*) '           \      .   /       .                        '
      WRITE(6,*) '            \ .      /           .                     '
      WRITE(6,*) '          .  \      /               .                  '
      WRITE(6,*) '        .     \    /                   .               '
      WRITE(6,*) '      .        \  /                       .            '
      WRITE(6,*) '   ___.________\/_________________________.____        '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '      - INCLUDE THE ENCLOSURE WHEN CALCULATING         '
      WRITE(6,*) '         THE FIELD (SCAT = 1)                          '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '         \                                             '
      WRITE(6,*) '          \    \      .   .                            '
      WRITE(6,*) '           \    \  .          .                        '
      WRITE(6,*) '            \   .#                .                    '
      WRITE(6,*) '             \   \|-------|          .                 '
      WRITE(6,*) '          .   \   >||-----||            .              '
      WRITE(6,*) '        .      \ / ||     ||               .           '
      WRITE(6,*) '   ___.________\/_||_____||________________.____       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '              SCAT ?                                   '
      READ(IUNIT,*) SCAT
C
      NREG = 0
      WRITE(6,10)
      WRITE(6,*) '                                                       '
      WRITE(6,*) '                                                       '
      WRITE(6,*) '       NOW LET"S TALK OUTPUT !!!!                      '
```

```
      WRITE(6,*) '                                                          ' GET02760
      WRITE(6,*) '       FIRST WOULD YOU LIKE TO ANALYZE THE WHOLE          ' GET02770
      WRITE(6,*) '        REGION (Y/N)?                                     ' GET02780
      WRITE(6,*) '                                                          ' GET02790
      READ(IUNIT,'(A1)') CHAR                                                 GET02800
      IF (CHAR .EQ. 'Y'  .OR.  CHAR .EQ.'y') GO TO 400                        GET02810
      WRITE(6,10)                                                             GET02820
      WRITE(6,*) '                                                          ' GET02830
      WRITE(6,*) '                                                          ' GET02840
      WRITE(6,*) '       HOW MANY REGIONS THEN WOULD YOU LIKE TO            ' GET02850
      WRITE(6,*) '        ANALYZE (PLEASE DO ENTER LARGE ENOUGH             ' GET02860
      WRITE(6,*) '                 RECTANGULAR REGIONS AS THE               ' GET02870
      I1 = 0                       TRIANGULARIZATION IS AUTOMATED)          ' GET02880
500   WRITE(6,*) '                                                            GET02890
      WRITE(6,*) '                # OF REGIONS ?                           ' GET02900
      READ(IUNIT,*) NREG                                                    ' GET02910
      IF (NREG .GT. 10) THEN                                                  GET02920
       WRITE(6,*) '                                                           GET02930
       WRITE(6,*) '       THE NUMBER OF MAXIMUM REGIONS  IS TEN (10)       ' GET02940
       WRITE(6,*) '        PLEASE ENTER OVER THE NUMBER OF REGIONS          ' GET02950
       IF (I1 .GT. 3) STOP                                                  ' GET02960
       I1 = I1 + 1                                                            GET02970
       GO TO 500                                                              GET02980
      ENDIF                                                                   GET02990
C                                                                             GET03000
      WRITE(6,*) '                                                            GET03010
      WRITE(6,*) '                                                          ' GET03020
      WRITE(6,*) '                                    * (X2,Y2)     Y       ' GET03030
      WRITE(6,*) '                                                          ' GET03040
      WRITE(6,*) '                                                          ' GET03050
      WRITE(6,*) '                                                          ' GET03060
      WRITE(6,*) '    (X1,Y1) *                                      -> X   ' GET03070
      DO 600 I = 1 , NREG                                                   ' GET03080
       WRITE(6,*) '                                                           GET03090
       WRITE(6,'(20H        FOR REGION #,I2,15H PLEASE ENTER :)') I         'GET03100
       WRITE(6,*) '                X1 , Y1 ? (in meters)                    GET03110
       READ(IUNIT,*) X1(I),Y1(I)                                            'GET03120
       WRITE(6,*) '                                                           GET03130
       READ(IUNIT,*) X2(I),Y2(I)        X2 , Y2 ? (in meters)               'GET03140
600   CONTINUE                                                                GET03150
400   CONTINUE                                                                GET03160
      WRITE(6,10)                                                             GET03170
      WRITE(6,*) '                                                            GET03180
      WRITE(6,*) '                                                          ' GET03190
      WRITE(6,*) '       AS A FINAL QUESTION, DO YOU PREFER TO SET           ' GET03200
      WRITE(6,*) '        THE TIMES FOR AT WHICH THE FIELDS ARE TO          ' GET03210
      WRITE(6,*) '        BE DETERMINED (OUT = 1), OR USE THE DEFAULT       ' GET03220
      WRITE(6,*) '        VALUES CALCULATED INTERNALLY (OUT = 0)?           ' GET03230
      WRITE(6,*) '          (IF YOU CHOOSE OUT = 1 SOME CRITICAL            ' GET03240
      WRITE(6,*) '           TIME VALUES SHALL BE GIVEN TO YOU)             ' GET03250
      WRITE(6,*) '                                                          ' GET03260
      WRITE(6,*) '                OUT ?                                     ' GET03270
      READ(IUNIT,*) IOUT                                                    ' GET03280
      IF (IOUT .EQ. 1) THEN                                                   GET03290
                                                                              GET03300
```

```
      WRITE(6,10)                                                       GET03310
      WRITE(6,*) '                                                      'GET03320
      WRITE(6,*) '                                                      'GET03330
      WRITE(6,*) ' f(t)                                                 'GET03340
      WRITE(6,*) '                                                      'GET03350
      WRITE(6,*) '                                                      'GET03360
      WRITE(6,*) '                                                      'GET03370
      WRITE(6,*) '                                                      'GET03380
      WRITE(6,*) '                                                      'GET03390
      WRITE(6,*) '                                          > t         'GET03400
      WRITE(6,*) '                                                      'GET03410
      WRITE(6,*) '            tpeak              tfin                   'GET03420
      WRITE(6,*) '                                                      'GET03430
      WRITE(6,*) '          WHERE  f(tfin) = 0.1 * f(tpeak)             'GET03440
      WRITE(6,*) '                                                      'GET03450
C                                                                       GET03460
      TPEAK = DLOG(BETA/ALPHA)/(BETA-ALPHA)                             GET03470
      EPEAK = DEXP(-ALPHA*TPEAK) - DEXP(-BETA*TPEAK)                    GET03480
      DO 700 I = 2 , 10000                                              GET03490
        IF (-BETA*TPEAK*REAL(I) .LE. -120.D0) THEN                      GET03500
          IF (DEXP(-ALPHA*TPEAK*REAL(I))  .LT. 0.10D0*EPEAK) GO TO 800  GET03510
        ELSE                                                            GET03520
          IF ((DEXP(-ALPHA*TPEAK*REAL(I)) - DEXP(-BETA*TPEAK*REAL(I))) GET03530
     &       .LT. 0.10D0*EPEAK) GO TO 800                              GET03540
        ENDIF                                                           GET03550
700   CONTINUE                                                          GET03560
800   CONTINUE                                                          GET03570
      TFIN  = REAL(I)*TPEAK                                             GET03580
      TDIFF = 4.0D-07*PI*MIURW*SIG*WE**2                                GET03590
      TFINA = TFIN + TDIFF                                              GET03600
      WRITE(6,*) '                                                      'GET03610
      WRITE(6,*) '          HERE ARE THE CRITICAL TIME VALUES :         'GET03620
      WRITE(6,*) '                                                      'GET03630
      WRITE(6,*) '                                                      'GET03640
      WRITE(6,'(7X,8Htpeak = ,E10.4,5X,7Htfin = ,E10.4)') TPEAK,TFIN    GET03650
      WRITE(6,'(7X,8Htdiff = ,E10.4,5X,9Htfinal = ,E10.4)') TDIFF,TFINAGET03660
      WRITE(6,*) '                                                      'GET03670
      WRITE(6,*) '          tdiff IS THE REQUIRED TIME FOR DIFFUSION    'GET03680
      WRITE(6,*) '          AND tfinal IS THE OVERALL FINAL INTEGRATION'GET03690
      WRITE(6,*) '          TIME CALCULATED AS tfinal = tfin + tdiff    'GET03700
      WRITE(6,*) '                                                      'GET03710
      WRITE(6,*) '         PLEASE ENTER THE NUMBER OF SPECIFIC TIMES    'GET03720
      WRITE(6,*) '          FOR WHICH YOU ARE INTERESTED FOR THE        'GET03730
      WRITE(6,*) '          INVESTIGATION                               'GET03740
900   WRITE(6,*) '                                                      'GET03750
      WRITE(6,*) '              # OF INTEGRATION TIME ?                 'GET03760
      READ(IUNIT,*) NINT                                                GET03770
      IF (NINT .GT. 50) THEN                                            GET03780
       WRITE(6,*) '                                                     'GET03790
       WRITE(6,*) '        THE MAXIMUM # OF INT. TIMES IS FIFTY (50)   'GET03800
       WRITE(6,*) '         PLEASE ENTER OVER THE NUMBER OF INT. TIME   'GET03810
       IF (I1 .GT. 3) STOP                                              GET03820
       I1 = I1 + 1                                                      GET03830
       GO TO 900                                                        GET03840
      ENDIF                                                             GET03850
```

```
C                                                                       GET03860
      WRITE(6,*) '                                                     'GET03870
      WRITE(6,*) '        IF THE INVESTIGATION IS CARRIED OUT THROUGH  'GET03880
      WRITE(6,*) '         CONSTANT TIME STEPS ENTER 1, 0 OTHERWISE    'GET03890
      WRITE(6,*) '                                                     'GET03900
      WRITE(6,*) '                    1 or 0 ?                         'GET03910
      READ(IUNIT,*) NSTE                                               GET03920
      IF (NSTE .EQ. 0) THEN                                            GET03930
       DO 1000 I = 1 , NINT                                            GET03940
        WRITE(6,*) '                                                   'GET03950
        WRITE(6,'(40H        PLEASE ENTER INTEGRATION TIME # ,I2)') I   GET03960
        WRITE(6,*) '                  (in seconds)                     'GET03970
        WRITE(6,*) '                                                   'GET03980
        READ(IUNIT,*) TINT(I)                                          GET03990
        WRITE(6,*) '                                                   'GET04000
1000    CONTINUE                                                       GET04010
       ELSE                                                            GET04020
        WRITE(6,*) '                                                   ' GET04030
        WRITE(6,*) '       PLEASE ENTER THE INTEGRATION TIME STEP      ' GET04040
        WRITE(6,*) '                  (in seconds)                     ' GET04050
        WRITE(6,*) '                                                   ' GET04060
        READ(IUNIT,*) TINTS                                            GET04070
        WRITE(6,*) '                                                   ' GET04080
        TINT(1) = TINTS                                                GET04090
        DO 1100 I = 2 , NINT                                           GET04100
          TINT(I) = TINT(I-1) + TINTS                                  GET04110
1100    CONTINUE                                                       GET04120
       ENDIF                                                           GET04130
      ENDIF                                                            GET04140
C                                                                      GET04150
C                                                                      GET04160
C  >-------------------------------------------------------------------< GET04170
C  >              RETURN TO THE MAIN PROGRAM                           < GET04180
C  >-------------------------------------------------------------------< GET04190
C                                                                      GET04200
10    FORMAT('1')                                                      GET04210
      RETURN                                                           GET04220
      END                                                              GET04230
```

Appendix D

FEM Computer Program for Emission Problems in Time Domain

```
      PROGRAM SMTDWR                                                    SMT00010
C                                                                       SMT00020
C                                                                       SMT00030
C  >=================================================================<  SMT00040
C  >                                                                 <  SMT00050
C  >                    PROGRAM SMTDWIRE                             <  SMT00060
C  >                    ----------------                             <  SMT00070
C  >                                                                 <  SMT00080
C  > ***DATE WRITTEN   10/08/89 (DD/MM/YY)                          <  SMT00090
C  > ***AUTHOR    GRAVELLE, LUC B.                                  <  SMT00100
C  >              ELECTRICAL ENGINEERING DEPARTMENT                 <  SMT00110
C  >              UNIVERSITY OF OTTAWA, CANADA                      <  SMT00120
C  > ***PURPOSE   PREDICT THE SHEILDING EFFECTIVENESS OF A TWO-     <  SMT00130
C  >              DIMENSIONAL ENCLOSURE WITH APERTURES IN THE       <  SMT00140
C  >              TIME-DOMAIN FOR RADIATION ANALYSIS.               <  SMT00150
C  >                                                                 <  SMT00160
C  >=================================================================<  SMT00170
C  >                                                                 <  SMT00180
C  >  THIS FORTRAN CODE IS A FINITE ELEMENT METHOD (FEM) BASED       <  SMT00190
C  >  TECHNIQUE USED TO PREDICT THE SHIELDING PERFORMANCE OF         <  SMT00200
C  >  ANY ARBITRARELY SHAPED TWO-DIMENSIONAL ENCLOSURE WITH         <  SMT00210
C  >  APERTURES (SINCE THE DIVISION INTO TRIANGLES IS               <  SMT00220
C  >  AUTOMATED, SOME RESTRICTIONS ARE IMPOSED ON THE SHAPE         <  SMT00230
C  >  OF THE ENCLOSURE).                                           <  SMT00240
C  >  THE SOLUTION OF THE RADIATED FIELDS GENERATED BY A CURRENT    <  SMT00250
C  >  SOURCE INSIDE THE ENCLOSURE IS PREDICTED IN THE TIME DOMAIN   <  SMT00260
C  >  VIA A DIFFERENTIAL/ALGEBRAIC SOLVER, THE "DDASSL".  THIS CODE <  SMT00270
C  >  IS BASED ON THE BACKWARD DIFFERENTIATION FORMULAS OF ORDERS ONE<  SMT00280
C  >  TO FIVE (THE ORDER FIVE IS USED IN THIS PROGRAM).  IT WAS     <  SMT00290
C  >  WRITTEN BY LINDA R. PETZOLD FROM LAWRENCE LIVERMORE NATIONAL. <  SMT00300
C  >                                                                 <  SMT00310
C  >     --> FOR MORE INFORMATION ON THE PROGRAM "DDASSL" SEE ITS    <  SMT00320
C  >           LISTING                                               <  SMT00330
C  >                                                                 <  SMT00340
C  >     --> PLEASE READ THE FILE NAMED "READ.ME" BEFORE STARTING    <  SMT00350
C  >                                                                 <  SMT00360
C  >=================================================================<  SMT00370
C  >                                                                 <  SMT00380
C  >                  DESCRIPTION OF PARAMETERS                      <  SMT00390
C  >                  -------------------------                      <  SMT00400
C  >  NUNK,NTRI     : THE MAXIMUM # OF UNKNOWNS & TRIANGLES          <  SMT00410
C  >  MLMX,MUMX     : THE MAXIMUM HALF-BANDWITHS ML(LOWER) & MU(UPPER)<  SMT00420
C  >                                                                 <  SMT00430
C  >                                                                 <  SMT00440
C  >  NVAIRG        : THE NUMBER (#) OF APERTURES PER VERTICLE WALLS <  SMT00450
C  >  PVAIRG,VAIRG  : THE POSITION & LENGTH OF THE APERTURES         <  SMT00460
C  >  HE,LE,WE      : HEIGHT, LENGTH & WALL THICKNESS OF ENCLOSURE   <  SMT00470
C  >  SIG,MIURW     : CONDUCTIVITY & RELATIVE PERMEABILITY OF THE WALL< SMT00480
C  >  HO,TR,PW,TF   : PEAK VALUE & TIME CHARACTERISTICS OF THE CURRENT< SMT00490
C  >  ALPHA,BETA    : TIME CHARACTERISTICS OF THE CURRENT FOR EMP ANAL< SMT00500
C  >  XW,YW         : CENTER LOCATION OF THE CONDUCTOR               <  SMT00510
C  >  NREG          : # OF REGIONS TO ANALYZE                        <  SMT00520
C  >  XR1 - YR2     : COORDINATES OF THE REGIONS TO BE ANALYZED      <  SMT00530
C  >  NINT,TINT     : # OF INTEGRATION TIMES & THE TIMES             <  SMT00540
C  >  TM            : TYPE OF ANALYSIS (TM OR TE)                    <  SMT00550
```

```
C  >  IOUT          : TIME PARAMETER (PARAM.)                                  < SMT00560
C  >                                                                          < SMT00570
C  >  X,Y           : COORDINATES OF THE TRIANGLE VERTICES                     < SMT00580
C  >  SIGMA,MIUR    : CONDUCTIVITY & RELATIVE PERMEABILITY OF TRIANGLE         < SMT00590
C  >  NV,DIRP       : # ASSOCIATED TO EACH NODES & BOUNDARY CONDITION          < SMT00600
C  >                                                                          < SMT00610
C  >  A,AP          : THE VALUE & DERIVATIVE OF THE FIELD AT EACH NODE         < SMT00620
C  >  W,RWORK,IWORK : WORKING SPACE REQUIRED BY "DDASSL"                      < SMT00630
C  >  R,S,SE        : MATRICES RELATED TO THE SYSTEM OF EQUATIONS              < SMT00640
C  >  Q,TQ          : REALTED TO P. SILVESTER'S SHAPE FUNCTIONS                < SMT00650
C  >  INFO          : INFORMATIONS REQUIRED BY "DDASSL"                        < SMT00660
C  >  VMIUO,CO      : PERMEABILITY & SPEED OF LIGHT IN AIR                     < SMT00670
C  >                                                                          < SMT00680
C  >                                                                          < SMT00690
C  >==========================================================================< SMT00700
C                                                                             SMT00710
C                                                                             SMT00720
C                                                                             SMT00730
      PARAMETER (NUNK=780,NTRI=1500,MLMX=300,MUMX=300)                        SMT00740
      IMPLICIT  REAL*8(A-H,O-Z)                                               SMT00750
C  >-------- VALUES FROM SUBROUTINE GETINF ------------------------------------< SMT00760
      REAL*8    PVAIRG,VAIRG,HE,LE,WE,XW,YW                                   SMT00770
      REAL*8    SIG,MIURW,HO,ALPHA,BETA                                       SMT00780
      REAL*8    XR1(10),YR1(10),XR2(10),YR2(10),TINT(50)                      SMT00790
      INTEGER   IBND,TM,NREG,IOUT,NINT                                        SMT00800
C  >-------- VALUES FROM TRIANG SUBROUTINE -------------------------------------< SMT00810
      REAL*8    X(NUNK),Y(NUNK),POT(NUNK),SIGMA(NTRI),MIUR(NTRI)              SMT00820
      INTEGER   NV(NTRI,3),NC(100,2),NI(100,2)                                SMT00830
      LOGICAL*4 DIRP(NUNK)                                                    SMT00840
C  >-------- VALUES FOR PROGRAM SEMPTDDA ---------------------------------------< SMT00850
      REAL*8    A(NUNK),AP(NUNK)                                              SMT00860
      REAL*8    RWORK(40+9*NUNK+(2*MLMX+MUMX+1)*NUNK)                         SMT00870
      REAL*8    R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1),SE(3,3),T(200)        SMT00880
      INTEGER   Q(4,15,15,3),TQ(4,15,15),INFO(15),IWORK(20+NUNK)              SMT00890
C                                                                             SMT00900
C  >==========================================================================< SMT00910
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                        < SMT00920
C  >               SUBROUTINES AND FUNCTIONS                                  < SMT00930
C  >==========================================================================< SMT00940
C                                                                             SMT00950
      COMMON    /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,HO,TINT,ALPHA,         SMT00960
     &BETA,XR1,YR1,XR2,YR2,XW,YW,AW,TR,PW,TF,                                 SMT00970
     &IBND,TM,NREG,IOUT,NINT,IFIELD,ITIME                                     SMT00980
      COMMON    /TRIWR/SIGMA,MIUR,                                            SMT00990
     &X,Y,POT,NOTR,NOPTL,NDIR,NV,NS,NC,NIE,NI,ML,MU,DIRP                      SMT01000
      COMMON    /OUT/A                                                        SMT01010
      COMMON    /RAS/R,S,NEQ/SIMPL/Q,TQ                                       SMT01020
C                                                                             SMT01030
      EXTERNAL RES,JAC                                                        SMT01040
C                                                                             SMT01050
      P(A1,B1,A2,B2,A3,B3) = A1*B2-A2*B1+A2*B3-A3*B2+A3*B1-A1*B3              SMT01060
C                                                                             SMT01070
      PI    = DACOS(-1.0D0)                                                   SMT01080
      VMIUO = 4.0D-07*PI                                                      SMT01090
      CO    = 3.0D08                                                          SMT01100
```

```
C                                                                       SMT01110
C  >-------------------------------------------------------------------< SMT01120
C  >     THE FOLLOWING CALL GETS THE NECESSARY INFORMATION             < SMT01130
C  >               CONCERNING THE GEOMETRY & THE FIELD                 < SMT01140
C  >-------------------------------------------------------------------< SMT01150
C                                                                       SMT01160
      CALL GETWTD                                                       SMT01170
C                                                                       SMT01180
C  >-------------------------------------------------------------------< SMT01190
C  >   THE FOLLOWING CALL PRINTS THE INFORMATION ON THE OUTPUT FILE    < SMT01200
C  >-------------------------------------------------------------------< SMT01210
C                                                                       SMT01220
      CALL PRTINF                                                       SMT01230
C                                                                       SMT01240
C  >-------------------------------------------------------------------< SMT01250
C  >     THIS CALL TO "TRIANG" WILL TRIANGULARIZE THE REGION OF        < SMT01260
C  >        INTEREST AND DETERMINE THE DIRICHLET BOUNDARIES            < SMT01270
C  >-------------------------------------------------------------------< SMT01280
C                                                                       SMT01290
      IF (IFIELD .EQ. 0) THEN                                           SMT01300
        CALL TRWRTD                                                     SMT01310
      ELSE                                                              SMT01320
        CALL TRIBPT                                                     SMT01330
      ENDIF                                                             SMT01340
      NN = NOPTL - NDIR                                                 SMT01350
C                                                                       SMT01360
C  >-------------------------------------------------------------------< SMT01370
C  >     NEXT WE DETERMINE THE TIME STEPS AND THE FINAL TIME           < SMT01380
C  >        FOR THE ANALYSIS OF THIS PARTICULAR PROBLEM                < SMT01390
C  >-------------------------------------------------------------------< SMT01400
C                                                                       SMT01410
      IF (IFIELD .EQ. 1) THEN                                           SMT01420
        SMIU  = SIG*VMIUO*MIURW                                         SMT01430
        TSTA  = 0.0D0                                                   SMT01440
        TPEAK = DLOG(BETA/ALPHA)/(BETA - ALPHA)                         SMT01450
        TDIF  = SMIU*WE**2                                              SMT01460
        TPIN  = TSTA + TPEAK + TDIF/(2.0D0**(DLOG10(ALPHA) - 4))        SMT01470
        EPEAK = DEXP(-ALPHA*TPEAK) - DEXP(-BETA*TPEAK)                  SMT01480
        DO 100 I = 2 , 10000                                            SMT01490
           IF (-BETA*TPEAK*REAL(I) .LE. -120.D0) THEN                   SMT01500
             IF (DEXP(-ALPHA*TPEAK*REAL(I))  .LT. 0.10D0*EPEAK)         SMT01510
     &       GO TO 200                                                  SMT01520
           ELSE                                                         SMT01530
             IF ((DEXP(-ALPHA*TPEAK*REAL(I)) -                          SMT01540
     &         DEXP(-BETA*TPEAK*REAL(I))) .LT. 0.10D0*EPEAK) GO TO 200  SMT01550
           ENDIF                                                        SMT01560
100     CONTINUE                                                        SMT01570
200     CONTINUE                                                        SMT01580
        TFIN  = REAL(I)*TPEAK                                           SMT01590
        N1    = I                                                       SMT01600
        TSTEP = TPEAK/10.0D0                                            SMT01610
        IF (N1 .GT. 15) THEN                                            SMT01620
          N1 = 15                                                       SMT01630
          TSTEP2 = TFIN/15.0D0                                          SMT01640
        ELSE                                                            SMT01650
```

```
          TSTEP2 = TPEAK                                                SMT01660
        ENDIF                                                           SMT01670
        TSTEP3 = TDIF                                                   SMT01680
        IF (TDIF .EQ. 0.0D0) TSTEP3 = TSTEP2                            SMT01690
        T(1)   = TSTA                                                   SMT01700
      ELSE                                                              SMT01710
        TSTEP = TR/10.0D0                                               SMT01720
        TSTEP2 = PW/10.0D0                                              SMT01730
        TSTEP3 = TF/10.0D0                                              SMT01740
        TSTEP4 = (TR + PW + TF)/2.0D0                                   SMT01750
      ENDIF                                                             SMT01760
C                                                                       SMT01770
C                                                                       SMT01780
C                                                                       SMT01790
C  >-------------------------------------------------------------------< SMT01800
C  >     THEN THE DISCRETE TIMES ARE CALCULATED                        < SMT01810
C  >-------------------------------------------------------------------< SMT01820
C                                                                       SMT01830
      IF (IFIELD .EQ. 1) THEN                                           SMT01840
        K = 0                                                           SMT01850
        DO 300 I = 2 , 20 + N1                                          SMT01860
          IF (I .LE. 10) THEN                                           SMT01870
            T(I+K) = T(I+K-1) + TSTEP                                   SMT01880
          ELSE IF (I .LE. 10 + N1) THEN                                 SMT01890
            T(I+K) = T(I+K-1) + TSTEP2                                  SMT01900
          ELSE                                                          SMT01910
            T(I+K) = T(I+K-1) + TSTEP3                                  SMT01920
          ENDIF                                                         SMT01930
          IF (T(I+K-1)  .LT. TPIN  .AND.  T(I+K) .GE. TPIN) THEN        SMT01940
            STEPX  = (T(I+K) - T(I+K-1))/10.0D0                         SMT01950
            T(I+K) = T(I+K-1) + STEPX                                   SMT01960
            DO 400 K = 1 , 10                                           SMT01970
                 T(I+K) = T(I+K-1) + STEPX                              SMT01980
400         CONTINUE                                                    SMT01990
            K = K - 1                                                   SMT02000
          ENDIF                                                         SMT02010
300     CONTINUE                                                        SMT02020
      ELSE                                                              SMT02030
        T(1) = 0.0D0                                                    SMT02040
        DO 350 I = 2 , 50                                               SMT02050
          IF (I .LE. 10) THEN                                           SMT02060
            T(I+K) = T(I+K-1) + TSTEP                                   SMT02070
          ELSE IF (I .LE. 20) THEN                                      SMT02080
            T(I+K) = T(I+K-1) + TSTEP2                                  SMT02090
          ELSE IF (I .LE. 40) THEN                                      SMT02100
            T(I+K) = T(I+K-1) + TSTEP3                                  SMT02110
          ELSE                                                          SMT02120
            T(I+K) = T(I+K-1) + TSTEP4                                  SMT02130
          ENDIF                                                         SMT02140
350     CONTINUE                                                        SMT02150
      ENDIF                                                             SMT02160
C                                                                       SMT02170
C  >-------------------------------------------------------------------< SMT02180
C  >     NEXT THE TIME REQUIRED BY THE USER ARE MERGED                 < SMT02190
C  >        WITH THE CALCULATED TIMES (IF THERE ARE ANY!!!)            < SMT02200
```

```
C  >-------------------------------------------------------------< SMT02210
C                                                                  SMT02220
      IF (NINT .NE. 0) THEN                                        SMT02230
        DO 500 I = 1 , NINT                                        SMT02240
          TINT(I) = TINT(I) + TSTA                                 SMT02250
500     CONTINUE                                                   SMT02260
        K = 1                                                      SMT02270
        DO 600 I = 2 , 30 + N1 + NINT                              SMT02280
          IF (T(I) .GT. TINT(K)  .AND.  T(I-1) .LT. TINT(K)) THEN  SMT02290
            DO 700 J = 30 + N1 + K , I , -1                        SMT02300
              T(J) = T(J-1)                                        SMT02310
700         CONTINUE                                               SMT02320
            T(I) = TINT(K)                                         SMT02330
            K    = K + 1                                           SMT02340
          ENDIF                                                    SMT02350
600     CONTINUE                                                   SMT02360
        DO 800 I = 30 + N1 + K , 30 + N1 + NINT                    SMT02370
          T(I) = TINT(K)                                           SMT02380
          K    = K + 1                                             SMT02390
800     CONTINUE                                                   SMT02400
      ENDIF                                                        SMT02410
C                                                                  SMT02420
C  >-------------------------------------------------------------< SMT02430
C  >     THIS CALL TO "SIMPLX" DEFINES THE SHAPE FUNCTIONS AS      < SMT02440
C  >        GIVEN IN P. SILVESTER'S BOOK ON THE FEM                < SMT02450
C  >-------------------------------------------------------------< SMT02460
C                                                                  SMT02470
      CALL SIMPLX(1)                                               SMT02480
C                                                                  SMT02490
C  >-------------------------------------------------------------< SMT02500
C  >     WE CAN NOW SET-UP THE "S" AND "R" MATRICES AT THE        < SMT02510
C  >              INITIAL TIME "T(1)"                             < SMT02520
C  >-------------------------------------------------------------< SMT02530
C                                                                  SMT02540
      DO 900 I = 1 , NUNK                                          SMT02550
        DO 900 J = 1 , MLMX+MUMX+1                                 SMT02560
          R(I,J) = 0.0D0                                           SMT02570
          S(I,J) = 0.0D0                                           SMT02580
900   CONTINUE                                                     SMT02590
      DO 1000 ITR = 1 , NOTR                                       SMT02600
        VM = MIUR(ITR)*VMIUO                                       SMT02610
        I1 = NV(ITR,1)                                             SMT02620
        I2 = NV(ITR,2)                                             SMT02630
        I3 = NV(ITR,3)                                             SMT02640
C       WRITE(9,111) ITR,I1,I2,I3,SIGMA(ITR),MIUR(ITR)             SMT02650
        DET = P(X(I1),Y(I1),X(I2),Y(I2),X(I3),Y(I3))               SMT02660
        IF (DET .EQ. 0.0D0) GO TO 8888                             SMT02670
        AR = ABS(DET)/2.0D0                                        SMT02680
        R12 = (X(I1) - X(I2))*(X(I1) - X(I2))                      SMT02690
     &        + (Y(I1) - Y(I2))*(Y(I1) - Y(I2))                    SMT02700
        R13 = (X(I1) - X(I3))*(X(I1) - X(I3))                      SMT02710
     &        + (Y(I1) - Y(I3))*(Y(I1) - Y(I3))                    SMT02720
        R23 = (X(I2) - X(I3))*(X(I2) - X(I3))                      SMT02730
     &        + (Y(I2) - Y(I3))*(Y(I2) - Y(I3))                    SMT02740
        COSA1 = (R12 + R13 - R23)/(2.D0*DSQRT(R12*R13))            SMT02750
```

```
      IF (COSA1 .GE. 1.0D0) WRITE(9,'(5HCOSA1,2X,F5.3)') COSA1          SMT02760
      COSA2 = (R12 + R23 - R13)/(2.D0*DSQRT(R12*R23))                   SMT02770
      IF (COSA2 .GE. 1.0D0) WRITE(9,'(5HCOSA2,2X,F5.3)') COSA2          SMT02780
      COSA3 = (R13 + R23 - R12)/(2.D0*DSQRT(R13*R23))                   SMT02790
      IF (COSA3 .GE. 1.0D0) WRITE(9,'(5HCOSA3,2X,F5.3)') COSA3          SMT02800
      COTA1 = COSA1/DSQRT(1.0D0 - COSA1*COSA1)                          SMT02810
      COTA2 = COSA2/DSQRT(1.0D0 - COSA2*COSA2)                          SMT02820
      COTA3 = COSA3/DSQRT(1.0D0 - COSA3*COSA3)                          SMT02830
      DO 1100 I = 1 , 3                                                 SMT02840
        DO 1100 J = 1 , 3                                               SMT02850
          SE(I,J) = (COTA1*REAL(Q(1,I,J,1)) + COTA2*                    SMT02860
     &          REAL(Q(1,I,J,2)) + COTA3*REAL(Q(1,I,J,3)))/2.0D0        SMT02870
1100    CONTINUE                                                        SMT02880
      DO 1200 I = 1 , 3                                                 SMT02890
        L = NV(ITR,I)                                                   SMT02900
        IF(DIRP(L)) GO TO 1200                                          SMT02910
        DO 1300 K = 1 , 3                                               SMT02920
          LC = NV(ITR,K)                                                SMT02930
          IF (.NOT.DIRP(LC)) THEN                                       SMT02940
            LC      = LC - L + ML + 1                                   SMT02950
            S(L,LC) = S(L,LC) + SE(I,K)                                 SMT02960
            IF (SIG .NE. 0.0D0) THEN                                    SMT02970
              R(L,LC) = R(L,LC) + VM*SIGMA(ITR)*TQ(1,I,K)*AR/12.0D0     SMT02980
            ENDIF                                                       SMT02990
          ENDIF                                                         SMT03000
1300    CONTINUE                                                        SMT03010
1200  CONTINUE                                                          SMT03020
1000  CONTINUE                                                          SMT03030
C                                                                       SMT03040
C  >-------------------------------------------------------------<      SMT03050
C  >     THE CONTRIBUTION OF THE INFINITE ELEMENT BOUNDARY         <     SMT03060
C  >         IS ADDED TO THE "R" AND "S" MATRICES                  <     SMT03070
C  >-------------------------------------------------------------<      SMT03080
C                                                                       SMT03090
      N = 1                                                             SMT03100
      DO 1350 I = 1 , NIE                                               SMT03110
        L1   = NI(I,1)                                                  SMT03120
        L2   = NI(I,2)                                                  SMT03130
        LI   = DSQRT((X(L2) - X(L1))**2 + (Y(L2) - Y(L1))**2)/2.0D0     SMT03140
        XMI  = (X(L1) + X(L2))/2.0D0                                    SMT03150
        YMI  = (Y(L1) + Y(L2))/2.0D0                                    SMT03160
        RMI  = DSQRT(XMI**2 + YMI**2)                                   SMT03170
        TETA = DATAN2(YMI,XMI)                                          SMT03180
        PHIE = DATAN2(Y(L2) - Y(L1) , X(L2) - X(L1)) - PI/2.0D0         SMT03190
        D    = RMI*DCOS(TETA - PHIE)                                    SMT03200
        B    = RMI*DSIN(TETA - PHIE)                                    SMT03210
        LC   = ML + 1                                                   SMT03220
        LC21 = L1 - L2 + ML + 1                                         SMT03230
        LC12 = L2 - L1 + ML + 1                                         SMT03240
        S(L1,LC) = S(L1,LC) + 1.0D0/(6.0D0*N*D)*((4.0D0*N**2 + 2.0D0*N  SMT03250
     &             + 1)*LI - 6.0D0*N*B + 3.0D0*RMI**2/LI)               SMT03260
        S(L2,LC) = S(L2,LC) + 1.0D0/(6.0D0*N*D)*((4.0D0*N**2 + 2.0D0*N  SMT03270
     &             + 1)*LI + 6.0D0*N*B + 3.0D0*RMI**2/LI)               SMT03280
        S(L1,LC12) = S(L1,LC12) + 1.0D0/(6.0D0*N*D)*((2.0D0*N**2        SMT03290
     &             -2.0D0*N - 1)*LI - 3.0D0*RMI**2/LI)                  SMT03300
```

```
        S(L2,LC21) = S(L2,LC21) + 1.0D0/(6.0D0*N*D)*((2.0D0*N**2          SMT03310
     &              -2.0D0*N - 1)*LI - 3.0D0*RMI**2/LI)                   SMT03320
1350  CONTINUE                                                           SMT03330
C                                                                        SMT03340
C  >-------------------------------------------------------------------< SMT03350
C  >     THE "S" & "R" MATRICES NEED TO BE SHIFTED UP BY NDIR          < SMT03360
C  >        AND ALSO SIDEWAYS                                          < SMT03370
C  >-------------------------------------------------------------------< SMT03380
C                                                                        SMT03390
      DO 1400 I = 1 , NOPTL                                              SMT03400
        IF (DIRP(I)) GO TO 1400                                          SMT03410
        DO 1500 K = 1 , ML + MU + 1                                      SMT03420
          IF (K .EQ. 1) K1 = 0                                           SMT03430
1800      CONTINUE                                                       SMT03440
          L = K + I - ML - 1 + K1                                        SMT03450
          IF (L .LT. 1  .OR.  K .EQ. ML+1) GO TO 1500                    SMT03460
          IF (L .GT. NOPTL) GO TO 1400                                   SMT03470
          IF (DIRP(L)) THEN                                              SMT03480
            IF (K .EQ. 1) THEN                                           SMT03490
              R(I,K) = 0.0D0                                             SMT03500
              S(I,K) = 0.0D0                                             SMT03510
            ELSE IF (K .EQ. ML+MU+1) THEN                                SMT03520
              R(I,K) = 0.0D0                                             SMT03530
              S(I,K) = 0.0D0                                             SMT03540
            ELSE IF (K .LT. ML+1) THEN                                   SMT03550
              DO 1600 J = K , 2 , -1                                     SMT03560
                R(I,J) = R(I,J-1)                                        SMT03570
                S(I,J) = S(I,J-1)                                        SMT03580
1600          CONTINUE                                                   SMT03590
              R(I,1) = 0.0D0                                             SMT03600
              S(I,1) = 0.0D0                                             SMT03610
            ELSE IF(K .GT. ML+1) THEN                                    SMT03620
              DO 1700 J = K , ML+MU                                      SMT03630
                R(I,J) = R(I,J+1)                                        SMT03640
                S(I,J) = S(I,J+1)                                        SMT03650
1700          CONTINUE                                                   SMT03660
              R(I,ML+MU+1) = 0.0D0                                       SMT03670
              S(I,ML+MU+1) = 0.0D0                                       SMT03680
              K1 = K1 + 1                                                SMT03690
              GO TO 1800                                                 SMT03700
            ENDIF                                                        SMT03710
          ENDIF                                                          SMT03720
1500    CONTINUE                                                         SMT03730
1400  CONTINUE                                                           SMT03740
      M1 = 0                                                             SMT03750
      DO 1900 I = 1 , NOPTL                                              SMT03760
        IF (.NOT. DIRP(I)) THEN                                          SMT03770
          DO 2000 K = 1 , ML + MU + 1                                    SMT03780
            R(I-M1,K) = R(I,K)                                           SMT03790
            S(I-M1,K) = S(I,K)                                           SMT03800
2000      CONTINUE                                                       SMT03810
        ELSE                                                             SMT03820
          M1 = M1 + 1                                                    SMT03830
        ENDIF                                                            SMT03840
1900  CONTINUE                                                           SMT03850
```

```
C                                                                       SMT03860
C  >-----------------------------------------------------------------<  SMT03870
C  >     SET-UP THE INITIAL VALUES FOR THE CALL TO "DDASSL"           <  SMT03880
C  >-----------------------------------------------------------------<  SMT03890
C                                                                       SMT03900
      DO 2100 I = 1 , NN                                                SMT03910
        A(I)  = 0.0D+00                                                 SMT03920
        AP(I) = 0.0D+00                                                 SMT03930
2100  CONTINUE                                                          SMT03940
      DO 2200 I = 1 , 15                                                SMT03950
        INFO(I)  = 0                                                    SMT03960
2200  CONTINUE                                                          SMT03970
      INFO(5)  = 1                                                      SMT03980
      INFO(6)  = 1                                                      SMT03990
      IWORK(1) = ML                                                     SMT04000
      IWORK(2) = MU                                                     SMT04010
      NEQ  = NN                                                         SMT04020
      RTOL = 0.001D0                                                    SMT04030
      ATOL = 0.001D0*H0                                                 SMT04040
      LRW  = 40 + 9*NUNK + (2*MLMX+MUMX+1)*NUNK                         SMT04050
      LIW  = 20 + NUNK                                                  SMT04060
      IPAR = NEQ                                                        SMT04070
      WRITE(9,*) NEQ                                                    SMT04080
C                                                                       SMT04090
C  >-----------------------------------------------------------------<  SMT04100
C  >     SOLVE WITH "DDASSL", THE SUBROUTINE WRITTEN BY               <  SMT04110
C  >         LINDA R. PETZOLD                                         <  SMT04120
C  >-----------------------------------------------------------------<  SMT04130
C                                                                       SMT04140
      I2 = 50 + NINT                                                    SMT04150
      IF (IFIELD .EQ. 1) I2 = 30 + N1 + NINT                            SMT04160
      DO 2300 I = 2 , 30 + N1 + NINT                                    SMT04170
        CALL DDASSL(RES,NEQ,T(I-1),A,AP,T(I),INFO,RTOL,ATOL,IDID,RWORK, SMT04180
     &              LRW,IWORK,LIW,RPAR,IPAR,JAC)                        SMT04190
        IF (IDID .LT. 0) THEN                                           SMT04200
          GO TO 2500                                                    SMT04210
        ELSE                                                            SMT04220
          IF (ITIME .EQ. 0) THEN                                        SMT04230
            CALL OUTP(T(I))                                             SMT04240
          ELSE                                                          SMT04250
            DO 2400 J = 1 , NINT                                        SMT04260
              IF (T(I) .GE. 0.999D0*TINT(J)  .AND.  T(I) .LE. 1.001D0*  SMT04270
     &            TINT(J)) CALL OUTP(T(I))                            SMT04280
2400        CONTINUE                                                    SMT04290
          ENDIF                                                         SMT04300
        ENDIF                                                           SMT04310
2300  CONTINUE                                                          SMT04320
      STOP                                                              SMT04330
2500  CONTINUE                                                          SMT04340
      WRITE(6,333) IDID                                                 SMT04350
      WRITE(9,333) IDID                                                 SMT04360
      STOP                                                              SMT04370
8888  CONTINUE                                                          SMT04380
      WRITE(9,'(5X,23HDETERMINANT OF TRIANGLE,I4,10H = 0.0!!!!)' ) ITR  SMT04390
      WRITE(6,'(5X,23HDETERMINANT OF TRIANGLE,I4,10H = 0.0!!!!)' ) ITR  SMT04400
```

```
      STOP                                                                SMT04410
5     FORMAT(8(1X,E8.2))                                                  SMT04420
111   FORMAT(1X,I3,3(1X,I4),7X,E14.7,4X,E10.4)                            SMT04430
222   FORMAT(//,5X,'HERE ARE SOME CONTROLLING VARIABLES',/)               SMT04440
333   FORMAT(//,5X,'AN ERROR HAS OCCURED IN DDASSL, IDID = ',I4)          SMT04450
      END                                                                 SMT04460
C                                                                         SMT04470
C                                                                         SMT04480
C  >--------------------------------------------------------------------< SMT04490
C  >                                                                    < SMT04500
C  >                  SUBROUTINE OUTP                                   < SMT04510
C  >                  ---------------                                   < SMT04520
C  >                                                                    < SMT04530
C  >       THIS SUBROUTINE PRINTS OUT THE RESULTS                       < SMT04540
C  >                                                                    < SMT04550
C  >--------------------------------------------------------------------< SMT04560
C                                                                         SMT04570
C                                                                         SMT04580
      SUBROUTINE OUTP(TIME)                                               SMT04590
      PARAMETER (NUNK=780,NTRI=1500,MLMX=300,MUMX=300)                    SMT04600
      IMPLICIT  REAL*8(A-H,O-Z)                                           SMT04610
C  >-------- VALUES FROM SUBROUTINE GETINF ---------------------------< SMT04620
      REAL*8    PVAIRG,VAIRG,HE,LE,WE,XW,YW                             SMT04630
      REAL*8    SIG,MIURW,HO,ALPHA,BETA                                 SMT04640
      REAL*8    XR1(10),YR1(10),XR2(10),YR2(10),TINT(50)                SMT04650
      INTEGER   IBND,TM,NREG,IOUT,NINT                                  SMT04660
C  >-------- VALUES FROM TRIANG SUBROUTINE ---------------------------< SMT04670
      REAL*8    X(NUNK),Y(NUNK),POT(NUNK),SIGMA(NTRI),MIUR(NTRI)        SMT04680
      INTEGER   NV(NTRI,3),NC(100,2),NI(100,2)                          SMT04690
      LOGICAL*4 DIRP(NUNK)                                              SMT04700
C  >-------- VALUES FROM PROGRAM SEMPTDDA ----------------------------< SMT04710
      REAL*8    TIME,A(NUNK),EFLD,B(NUNK)                               SMT04720
      REAL*8    XH(NTRI),YH(NTRI),HX(NTRI),HY(NTRI),H(NTRI),HDB         SMT04730
C                                                                         SMT04740
C  >--------------------------------------------------------------------< SMT04750
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                  < SMT04760
C  >               SUBROUTINES AND FUNCTIONS                            < SMT04770
C  >--------------------------------------------------------------------< SMT04780
C                                                                         SMT04790
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,HO,TINT,ALPHA,     SMT04800
     &BETA,XR1,YR1,XR2,YR2,XW,YW,AW,TR,PW,TF,                           SMT04810
     &IBND,TM,NREG,IOUT,NINT,IFIELD,ITIME                               SMT04820
      COMMON   /TRIWR/SIGMA,MIUR,                                       SMT04830
     &X,Y,POT,NOTR,NOPTL,NDIR,NV,NS,NC,NIE,NI,ML,MU,DIRP                SMT04840
      COMMON   /OUT/A                                                   SMT04850
C                                                                         SMT04860
      P(A1,B1,A2,B2,A3,B3) = A1*B2-A2*B1+A2*B3-A3*B2+A3*B1-A1*B3        SMT04870
C                                                                         SMT04880
      WRITE(9,111) TIME                                                 SMT04890
      WRITE(9,222)                                                      SMT04900
      N1 = 1                                                            SMT04910
      DO 100 I = 1 , NOPTL                                              SMT04920
        IF (NREG .EQ. 0) THEN                                           SMT04930
C                                                                         SMT04940
C  >--------------------------------------------------------------------< SMT04950
```

```
C  >     THE USER WANTS TO PRINT OUT THE WHOLE REGION                    < SMT04960
C  >-------------------------------------------------------------------< SMT04970
C                                                                         SMT04980
         IF (DIRP(I)) THEN                                                SMT04990
           WRITE(9,333) X(I),Y(I),POT(I)                                  SMT05000
         ELSE                                                             SMT05010
           WRITE(9,333) X(I),Y(I),A(N1)                                   SMT05020
           N1 = N1 + 1                                                    SMT05030
         ENDIF                                                            SMT05040
       ELSE                                                               SMT05050
C                                                                         SMT05060
C  >-------------------------------------------------------------------< SMT05070
C  >     THE USER WANTS TO PRINT OUT ONLY PARTS OF THE REGION           < SMT05080
C  >-------------------------------------------------------------------< SMT05090
C                                                                         SMT05100
         DO 200 J = 1 , NREG                                              SMT05110
           IF ((X(I) .GE. XR1(J)  .AND.  X(I) .LE. XR2(J))  .AND.         SMT05120
     &         (Y(I) .GE. YR1(J)  .AND.  Y(I) .LE. YR2(J)))  THEN       SMT05130
             IF (DIRP(I)) THEN                                            SMT05140
               WRITE(9,333) X(I),Y(I),POT(I)                              SMT05150
             ELSE                                                         SMT05160
               WRITE(9,333) X(I),Y(I),A(N1)                               SMT05170
               N1 = N1 + 1                                                SMT05180
             ENDIF                                                        SMT05190
           ENDIF                                                          SMT05200
200      CONTINUE                                                         SMT05210
       ENDIF                                                              SMT05220
100    CONTINUE                                                           SMT05230
C                                                                         SMT05240
C                                                                         SMT05250
C  >-------------------------------------------------------------------< SMT05260
C  >          NEXT WE'LL CALCULATE THE MAGNETIC FIELD                   < SMT05270
C  >            WITHIN EACH TRIANGLES                                   < SMT05280
C  >-------------------------------------------------------------------< SMT05290
C                                                                         SMT05300
       N1 = NOPTL - NDIR                                                  SMT05310
       DO 300 I = NOPTL , 1 , -1                                          SMT05320
         IF (DIRP(I)) THEN                                                SMT05330
           B(I) = POT(I)                                                  SMT05340
         ELSE                                                             SMT05350
           B(I) = A(N1)                                                   SMT05360
           N1   = N1 - 1                                                  SMT05370
         ENDIF                                                            SMT05380
300    CONTINUE                                                           SMT05390
       DO 400 ITR = 1 , NOTR                                              SMT05400
         I1  = NV(ITR,1)                                                  SMT05410
         I2  = NV(ITR,2)                                                  SMT05420
         I3  = NV(ITR,3)                                                  SMT05430
         DET = P(X(I1),Y(I1),X(I2),Y(I2),X(I3),Y(I3))                     SMT05440
         DADX = ((Y(I2) - Y(I3))*B(I1) + (Y(I3) - Y(I1))*B(I2) +          SMT05450
     &           (Y(I1) - Y(I2))*B(I3))/DET                             SMT05460
         DADY = ((X(I3) - X(I2))*B(I1) + (X(I1) - X(I3))*B(I2) +          SMT05470
     &           (X(I2) - X(I1))*B(I3))/DET                             SMT05480
         HX(ITR) = DADY                                                   SMT05490
         HY(ITR) = -DADX                                                  SMT05500
```

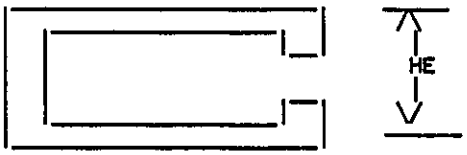
```
        H(ITR)  = DSQRT(DADX**2 + DADY**2)                                SMT05510
        XH(ITR) = (X(I1) + X(I2) + X(I3))/3.0D0                           SMT05520
        YH(ITR) = (Y(I1) + Y(I2) + Y(I3))/3.0D0                           SMT05530
400   CONTINUE                                                            SMT05540
      WRITE(9,444)                                                        SMT05550
      DO 500 I = 1 , NOTR                                                 SMT05560
        IF (NREG .EQ. 0) THEN                                             SMT05570
C                                                                         SMT05580
C  >-------------------------------------------------------------------<  SMT05590
C  >     THE USER WANTS TO PRINT OUT THE WHOLE REGION                   <  SMT05600
C  >-------------------------------------------------------------------<  SMT05610
C                                                                         SMT05620
          WRITE(9,555) XH(I),YH(I),HX(I),HY(I),H(I)                       SMT05630
        ELSE                                                              SMT05640
C                                                                         SMT05650
C  >-------------------------------------------------------------------<  SMT05660
C  >     THE USER WANTS TO PRINT OUT ONLY PARTS OF THE REGION           <  SMT05670
C  >-------------------------------------------------------------------<  SMT05680
C                                                                         SMT05690
          DO 600 J = 1 , NREG                                             SMT05700
            IF ((XH(I) .GE. XR1(J)  .AND.  XH(I) .LE. XR2(J))  .AND.      SMT05710
     &          (YH(I) .GE. YR1(J)  .AND.  YH(I) .LE. YR2(J)))  THEN      SMT05720
              WRITE(9,555) XH(I),YH(I),HX(I),HY(I),H(I)                   SMT05730
            ENDIF                                                         SMT05740
600       CONTINUE                                                        SMT05750
        ENDIF                                                             SMT05760
500   CONTINUE                                                            SMT05770
111   FORMAT(////,10X,'/',30('-'),'/',/,10X,'/',30X,'/',/,10X,            SMT05780
     & '/   POSITION AND ASSOCIATED    /',                               SMT05790
     &/,10X,'/',6X,'MAGN. VECTOR POT. A'                                  SMT05800
     & ,5X,'/',/,10X,'/',6X,'AT T = ',E10.4,7X,'/',/,10X,'/',30X,'/',/,   SMT05810
     & 10X,'/',30('-'),'/',//)                                            SMT05820
222   FORMAT(6X,'X',11X,'Y',16X,'A',/)                                    SMT05830
333   FORMAT(2E12.5,E18.7)                                                SMT05840
444   FORMAT(//,7X,'X',11X,'Y',10X,'HX',10X,'HY',10X,'H',/)               SMT05850
555   FORMAT(5E12.3)                                                      SMT05860
      RETURN                                                              SMT05870
      END                                                                 SMT05880
C                                                                         SMT05890
C                                                                         SMT05900
C  >-------------------------------------------------------------------<  SMT05910
C  >                                                                   <  SMT05920
C  >                  SUBROUTINE RES                                   <  SMT05930
C  >                  ----------                                       <  SMT05940
C  >                                                                   <  SMT05950
C  >       THIS SUBROUTINE DEFINES THE DIFFERENTIAL/ALGEBRAIC          <  SMT05960
C  >          SYSTEM OF EQUATIONS TO BE SOLVED :                       <  SMT05970
C  >                 DELTA = G(T,A,AP)                                 <  SMT05980
C  >                                                                   <  SMT05990
C  >-------------------------------------------------------------------<  SMT06000
C                                                                         SMT06010
C                                                                         SMT06020
      SUBROUTINE RES (TIME,A,AP,DELTA,IRES,RPAR,IPAR)                     SMT06030
      PARAMETER (NUNK=780,NTRI=1500,MLMX=300,MUMX=300)                    SMT06040
      IMPLICIT  REAL*8(A-H,O-Z)                                           SMT06050
```

```
C  >-------- VALUES FROM SUBROUTINE GETINF ----------------------------< SMT06060
      REAL*8    PVAIRG,VAIRG,HE,LE,WE,XW,YW                            SMT06070
      REAL*8    SIG,MIURW,HO,ALPHA,BETA                                SMT06080
      REAL*8    XR1(10),YR1(10),XR2(10),YR2(10),TINT(50)               SMT06090
      INTEGER   IBND,TM,NREG,IOUT,NINT                                 SMT06100
C  >-------- VALUES FROM TRIANG SUBROUTINE ----------------------------< SMT06110
      REAL*8    X(NUNK),Y(NUNK),POT(NUNK),SIGMA(NTRI),MIUR(NTRI)       SMT06120
      INTEGER   NV(NTRI,3),NC(100,2),NI(100,2)                         SMT06130
      LOGICAL*4 DIRP(NUNK)                                             SMT06140
C  >-------- VALUES FROM PROGRAM SEMPTDDA ----------------------------< SMT06150
      REAL*8   R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1)                 SMT06160
      INTEGER  Q(4,15,15,3),TQ(4,15,15)                                SMT06170
C  >-------- VALUES FOR SUBROUTINE RES ----------------------------< SMT06180
      REAL*8   TIME,A(NUNK),AP(NUNK),B(NUNK),DELTA(IPAR),SE(3,3),EFLD  SMT06190
C                                                                      SMT06200
C  >-------------------------------------------------------------------< SMT06210
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE               < SMT06220
C  >               SUBROUTINES AND FUNCTIONS                         < SMT06230
C  >-------------------------------------------------------------------< SMT06240
C                                                                      SMT06250
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,HO,TINT,ALPHA,   SMT06260
     &BETA,XR1,YR1,XR2,YR2,XW,YW,AW,TR,PW,TF,                          SMT06270
     &IBND,TM,NREG,IOUT,NINT,IFIELD,ITIME                              SMT06280
      COMMON   /TRIWR/SIGMA,MIUR,                                      SMT06290
     &X,Y,POT,NOTR,NOPTL,NDIR,NV,NS,NC,NIE,NI,ML,MU,DIRP               SMT06300
      COMMON   /RAS/R,S,NEQ/SIMPL/Q,TQ                                 SMT06310
C                                                                      SMT06320
      P(A1,B1,A2,B2,A3,B3) = A1*B2-A2*B1+A2*B3-A3*B2+A3*B1-A1*B3       SMT06330
C                                                                      SMT06340
      PI    = DACOS(-1.0D0)                                            SMT06350
      VMIUO = 4.0D-07*PI                                               SMT06360
      CO    = 3.0D08                                                   SMT06370
      DO 100 I = 1 , NUNK                                              SMT06380
        B(I)    = 0.0D0                                                SMT06390
100   CONTINUE                                                         SMT06400
C                                                                      SMT06410
C  >-------------------------------------------------------------------< SMT06420
C  >     THE "B" VECTOR (THE SOURCE VECTOR) NEED TO BE SET-UP        < SMT06430
C  >           AT "TIME"                                             < SMT06440
C  >                                                                 < SMT06450
C  >     THE CONTRIBUTION OF THE INEOMOGENEOUS NEUMANN BOUNDARY      < SMT06460
C  >         IS ADDED TO THE "B" VECTOR                              < SMT06470
C  >-------------------------------------------------------------------< SMT06480
C                                                                      SMT06490
      IF (IFIELD .EQ. 1) THEN                                          SMT06500
        VALUE = HO*(DEXP(-ALPHA*TIME) - DEXP(-BETA*TIME))              SMT06510
      ELSE                                                             SMT06520
        IF (TIME .LE. TR) THEN                                         SMT06530
          VALUE = HO*TIME/TR                                           SMT06540
        ELSE IF (TIME .LE. TR+PW) THEN                                 SMT06550
          VALUE = HO                                                   SMT06560
        ELSE IF (TIME .LE. TR+PW+TF) THEN                              SMT06570
          VALUE = -HO/TF*(TIME - TR - PW) + HO                         SMT06580
        ELSE                                                           SMT06590
          VALUE = 0.0D0                                                SMT06600
```

```
        ENDIF                                                          SMT06610
      ENDIF                                                            SMT06620
      DO 200 I = 1 , NS                                                SMT06630
        VL = DSQRT((X(NC(I,2)) - X(NC(I,1)))**2 +                      SMT06640
     &           (Y(NC(I,2)) - Y(NC(I,1)))**2)                         SMT06650
        DO 300 J = 1 , 2                                               SMT06660
          L     = NC(I,J)                                              SMT06670
          B(L) = B(L) - VL/2.0D0*VALUE                                 SMT06680
300     CONTINUE                                                       SMT06690
200   CONTINUE                                                         SMT06700
C                                                                      SMT06710
C  >-------------------------------------------------------------<     SMT06720
C  >     THE "B" VECTOR NEED TO BE SHIFTED UP BY NDIR              <    SMT06730
C  >-------------------------------------------------------------<     SMT06740
C                                                                      SMT06750
      N1 = 0                                                           SMT06760
      DO 400 I = 1 , NOPTL                                             SMT06770
        IF (.NOT. DIRP(I)) THEN                                        SMT06780
          B(I-N1) = B(I)                                               SMT06790
        ELSE                                                           SMT06800
          N1 = N1 + 1                                                  SMT06810
        ENDIF                                                          SMT06820
400   CONTINUE                                                         SMT06830
C                                                                      SMT06840
C  >-------------------------------------------------------------<     SMT06850
C  >     WE THEN CALCULATE  DELTA = G(TIME,A,AP)                   <    SMT06860
C  >-------------------------------------------------------------<     SMT06870
C                                                                      SMT06880
      DO 500 I = 1 , NEQ                                               SMT06890
        SUM = 0.0D0                                                    SMT06900
        DO 600 J = 1 , NEQ                                             SMT06910
          K = J - I + ML + 1                                           SMT06920
          IF (K .GT. 0  .AND.  K .LE. ML+MU+1) THEN                    SMT06930
            IF (SIG .EQ. 0.0D0) THEN                                   SMT06940
              SUM = SUM + S(I,K)*A(J)                                  SMT06950
          ELSE                                                         SMT06960
              SUM = SUM + R(I,K)*AP(J) + S(I,K)*A(J)                   SMT06970
          ENDIF                                                        SMT06980
          ENDIF                                                        SMT06990
600     CONTINUE                                                       SMT07000
        DELTA(I) = SUM - B(I)                                          SMT07010
500   CONTINUE                                                         SMT07020
      RETURN                                                           SMT07030
9999  CONTINUE                                                         SMT07040
      WRITE(9,*) 'ITR = ',ITR                                          SMT07050
      WRITE(9,*) 'ITR = ',ITR                                          SMT07060
      STOP                                                             SMT07070
      END                                                              SMT07080
C                                                                      SMT07090
C                                                                      SMT07100
C  >-------------------------------------------------------------<     SMT07110
C  >                                                            <     SMT07120
C  >                 SUBROUTINE JAC                             <     SMT07130
C  >                 ______________                             <     SMT07140
C  >                                                            <     SMT07150
```

```
C  >       THIS SUBROUTINE DEFINES THE JACOBIAN, THE MATRIX OF          < SMT07160
C  >          PARTIAL DERIVATIVES :                                    < SMT07170
C  >                        PD = DG/DA + CJ*DG/DAP                     < SMT07180
C  >                                                                   < SMT07190
C  >-------------------------------------------------------------------< SMT07200
C                                                                      SMT07210
C                                                                      SMT07220
      SUBROUTINE JAC (T,A,AP,PD,CJ,RPAR,IPAR)                          SMT07230
      PARAMETER (NUNK=780,NTRI=1500,MLMX=300,MUMX=300)                 SMT07240
      IMPLICIT  REAL*8(A-H,O-Z)                                        SMT07250
C  >-------- VALUES FROM TRIANG SUBROUTINE ------------------------------< SMT07260
      REAL*8    X(NUNK),Y(NUNK),POT(NUNK),SIGMA(NTRI),MIUR(NTRI)       SMT07270
      INTEGER   NV(NTRI,3),NC(100,2),NI(100,2)                         SMT07280
      LOGICAL*4 DIRP(NUNK)                                             SMT07290
C  >-------- VALUES FOR SUBROUTINE JAC ------------------------------< SMT07300
      REAL*8    A(NUNK),AP(NUNK),PD(2*ML+MU+1,IPAR)                    SMT07310
      REAL*8    R(NUNK,MLMX+MUMX+1),S(NUNK,MLMX+MUMX+1),T,CJ           SMT07320
C                                                                      SMT07330
C  >-------------------------------------------------------------------< SMT07340
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE                 < SMT07350
C  >               SUBROUTINES AND FUNCTIONS                           < SMT07360
C  >-------------------------------------------------------------------< SMT07370
C                                                                      SMT07380
      COMMON   /TRIWR/SIGMA,MIUR,                                      SMT07390
     &X,Y,POT,NOTR,NOPTL,NDIR,NV,NS,NC,NIE,NI,ML,MU,DIRP               SMT07400
      COMMON   /RAS/R,S,NEQ                                            SMT07410
C                                                                      SMT07420
C                                                                      SMT07430
C  >-------------------------------------------------------------------< SMT07440
C  >     AS THE "S" AND "R" MATRICES ARE INDEPENDENT OF TIME,          < SMT07450
C  >        THE JACOBIAN IS ALSO TIME INDEPENDENT                      < SMT07460
C  >-------------------------------------------------------------------< SMT07470
C                                                                      SMT07480
      DO 100 I = 1 , NEQ                                               SMT07490
        DO 100 J = 1 , NEQ                                             SMT07500
          K = J - I + ML + 1                                           SMT07510
          IF (K .GT. 0  .AND.  K .LE. ML+MU+1) THEN                    SMT07520
            IROW = I - J + ML + MU + 1                                 SMT07530
            IF (SIG .EQ. 0.0D0) THEN                                   SMT07540
              PD(IROW,J) = S(I,K)                                      SMT07550
            ELSE                                                       SMT07560
              PD(IROW,J) = S(I,K) + CJ*R(I,K)                          SMT07570
            ENDIF                                                      SMT07580
          ENDIF                                                        SMT07590
100   CONTINUE                                                         SMT07600
      RETURN                                                           SMT07610
      END                                                              SMT07620
```

```
                SUBROUTINE GETWTD                                       GET00010
C                                                                       GET00020
C  >-------------------------------------------------------------------< GET00030
C  >     THIS SUBROUTINE READS THE REQUIRED INFORMATION EITHER FROM    < GET00040
C  >      THE TERMINAL OR AN INPUT FILE                                < GET00050
C  >-------------------------------------------------------------------< GET00060
C                                                                       GET00070
      IMPLICIT  REAL*8(A-H,O-Z)                                         GET00080
      REAL*8    PVAIRG,VAIRG,HE,LE,WE,XW,YW                             GET00090
      REAL*8    SIG,MIURW,HO,ALPHA,BETA                                 GET00100
      REAL*8    X1(10),Y1(10),X2(10),Y2(10),TINT(50)                    GET00110
      INTEGER   IBND,TM,NREG,IOUT,NINT                                  GET00120
      CHARACTER*1 FILE,CHAR*1                                           GET00130
C                                                                       GET00140
C  >-------------------------------------------------------------------< GET00150
C  >     THE FOLLOWING IS THE COMMON BLOCK LINK TO THE MAIN            < GET00160
C  >               PROGRAM                                             < GET00170
C  >-------------------------------------------------------------------< GET00180
C                                                                       GET00190
      COMMON   /GETIN/PVAIRG,VAIRG,HE,LE,WE,SIG,MIURW,HO,TINT,ALPHA,    GET00200
     &       BETA,X1,Y1,X2,Y2,XW,YW,A,TR,PW,TF,                         GET00210
     &       IBND,TM,NREG,IOUT,NINT,IFIELD,ITIME                        GET00220
C                                                                       GET00230
      PI     = DACOS(-1.0D0)                                           GET00240
      NVAIRG = 0                                                        GET00250
      SIG    = 0.0D0                                                    GET00260
C                                                                       GET00270
C  >-------------------------------------------------------------------< GET00280
C  >     THE INFORMATION IS READ FROM THE TERMINAL IF THE ANSWER IS    < GET00290
C  >      NO OTHERWISE IT IS FOUND IN AN EXISTING FILE                 < GET00300
C  >-------------------------------------------------------------------< GET00310
C                                                                       GET00320
      WRITE(6,*) 'IS THERE AN EXISTING INPUT FILE ? (Y/N)'              GET00330
      READ(5,'(A1)') FILE                                               GET00340
      IF ( FILE .EQ. 'N'  .OR.  FILE .EQ. 'N' ) THEN                    GET00350
         IUNIT=5                                                        GET00360
      ELSE                                                              GET00370
         IUNIT=7                                                        GET00380
      ENDIF                                                             GET00390
C                                                                       GET00400
C                                                                       GET00410
      WRITE(6,10)                                                       GET00420
      WRITE(6,*) '                                                    ' GET00430
      WRITE(6,*) '                                                    ' GET00440
      WRITE(6,*) '        ENTER THE "OUTSIDE" HORIZONTAL (LE), THEN   ' GET00450
      WRITE(6,*) '         VERTICAL (HE), DIMENSIONS OF THE ENCLOSURE ' GET00460
      WRITE(6,*) '         ALONG WITH THE WALL THICKNESS (WE)         ' GET00470
      WRITE(6,*) '                                                    ' GET00480
      WRITE(6,*) '                                                    ' GET00490
      WRITE(6,*) '                                                    ' GET00500
      WRITE(6,*) '                                                    ' GET00510
      WRITE(6,*) '                                                    ' GET00520
      WRITE(6,*) '                                                    ' GET00530
      WRITE(6,*) '                                                    ' GET00540
      WRITE(6,*) '                                                    ' CET00550
```



```
      WRITE(6,*) '                                                      '
      WRITE(6,*) '            -->| |<--                  |               '
      WRITE(6,*) '               | WE                    |               '
      WRITE(6,*) '               |<--------- LE -------->|               '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '           LE , HE , WE ? (IN METERS)                 '
      READ(IUNIT,*) LE,HE,WE
      WRITE(6,10)
      WRITE(6,*) '                                                      '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '      ENTER THE Y-COORDINATE OF THE AIRGAP, THEN      '
      WRITE(6,*) '       THE LENGTH OF THE AIRGAP                        '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '         |--------------------------|                 '
      WRITE(6,*) '         |                          |                 '
      WRITE(6,*) '         |                LENGTH ---                   '
      WRITE(6,*) '         |             OF AIRGAP__       POSITION     '
      WRITE(6,*) '         |             (VAIRG)  --  | ---  OF AIRGAP'
      WRITE(6,*) '         |                          |     (PVAIRG)    '
      WRITE(6,*) '         |--------------------------|                 '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '            PVAIRG , VAIRG ? (in meters)              '
      READ(IUNIT,*) PVAIRG,VAIRG
      WRITE(6,10)
      WRITE(6,*) '                                                      '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '     ARE WE ANALYSING THE FIELD RADIATED BY A         '
      WRITE(6,*) '      CIRCULAR CONDUCTOR (IFIELD = 0) OR A PRINTED'
      WRITE(6,*) '      CIRCUIT BOARD"S BACK PLANE (IFIELD = 1)         '
      WRITE(6,*) '      OR TWO BOARD"S BACK PLANES (IFIELD = 2)         '
      WRITE(6,*) '                                                      '
      WRITE(6,*) '                   IFIELD ?                           '
      READ(IUNIT,*) IFIELD
      IF (IFIELD .EQ. 0) THEN
       WRITE(6,10)
       WRITE(6,*) '                                                      '
       WRITE(6,*) '                                                      '
       WRITE(6,*) '     ENTER THE POSITION AND THE RADIUS OF THE         '
       WRITE(6,*) '      CONDUCTOR                                       '
       WRITE(6,*) '           |--------------------------|               '
       WRITE(6,*) '           |                          |               '
       WRITE(6,*) '           |      |a|                 |               '
       WRITE(6,*) '           |      . .                                 '
       WRITE(6,*) '     YW - -|- .     .<-- I                            '
       WRITE(6,*) '           |      . .                                 '
       WRITE(6,*) '           |      |                   |               '
       WRITE(6,*) '           |______|___________________|               '
       WRITE(6,*) '                  |                                   '
       WRITE(6,*) '                  XW                                  '
       WRITE(6,*) '                                                      '
       WRITE(6,*) '             XW , YW ? (in meters)                    '
       WRITE(6,*) '             a       ? (in meters)                    '
       READ(IUNIT,*) XW,YW,A
       WRITE(6,10)
```

```
      WRITE(6,*) '                                                        'GET01110
      WRITE(6,*) '       ENTER THE PEAK CURRENT STRENGTH IN AMPERS (A)'GET01120
      WRITE(6,*) '        THEN THE RISE, PULSE WIDTH AND THE FALL    ' GET01130
      WRITE(6,*) '        TIMES                                      ' GET01140
      WRITE(6,*) '                                                   'GET01150
      WRITE(6,*) '          ^                                        'GET01160
      WRITE(6,*) '   IO _   |     .............                      'GET01170
      WRITE(6,*) '          |   .               .                     'GET01180
      WRITE(6,*) '          |  .                 .                    'GET01190
      WRITE(6,*) '          | .                   .                   'GET01200
      WRITE(6,*) '          |_____________________________ t          'GET01210
      WRITE(6,*) '            tr |<-- PW -->|tf|                      'GET01220
      WRITE(6,*) '                                                   'GET01230
      WRITE(6,*) '                                                   'GET01240
      WRITE(6,*) '              IO (A), tr , PW , tf ?               ' GET01250
      READ(IUNIT,*) HO,TR,PW,TF                                        GET01260
      HO = 2.0D-07*HO/A                                                GET01270
      WRITE(6,10)                                                      GET01280
      WRITE(6,*) '                                                   'GET01290
      WRITE(6,*) '                                                   'GET01300
      WRITE(6,*) '       BOUNDARY CONDITIONS :                       'GET01310
      WRITE(6,*) '                                                   'GET01320
      WRITE(6,*) '     IOUT = 0              IOUT = 1                 'GET01330
      WRITE(6,*) '                                                   'GET01340
      WRITE(6,*) '       * — *                 * — *                 'GET01350
      WRITE(6,*) '     *     *               *     *                 'GET01360
      WRITE(6,*) '    |  |—|  | <- A=0     |  |—|  | <-INFINITE     'GET01370
      WRITE(6,*) '     *     *               *     *    ELEMENTS     'GET01380
      WRITE(6,*) '       * — *                 * — *                 'GET01390
      WRITE(6,*) '                                                   'GET01400
      WRITE(6,*) '                                                   'GET01410
      WRITE(6,*) '     IOUT = 2                IOUT = 3              'GET01420
      WRITE(6,*) '                                                   'GET01430
      WRITE(6,*) '    |———————| * <- DA/DN       * — *               'GET01440
      WRITE(6,*) '    |         *   =0         *     *               'GET01450
      WRITE(6,*) '    |         *            |__|—|__| <-INFINITE    'GET01460
      WRITE(6,*) '    |_______| *                    ELEMENTS'GET01470
      WRITE(6,*) '                                  | A=0             'GET01480
      WRITE(6,*) '                                                   'GET01490
      WRITE(6,*) '                IOUT = 4                           'GET01500
      WRITE(6,*) '                                                   'GET01510
      WRITE(6,*) '          |———————|*——————————|                    'GET01520
      WRITE(6,*) '          |         *          | <- INFINITE       'GET01530
      WRITE(6,*) '          |         *          |    ELEMENTS       'GET01540
      WRITE(6,*) '          |_______|_*__________|                    'GET01550
      WRITE(6,*) '                                                   'GET01560
      WRITE(6,*) '                                | A=0              'GET01570
      WRITE(6,*) '                                                   'GET01580
      WRITE(6,*) '       IOUT=0 -> FIELD IS ZERO ON CIRCULAR BOUND.  'GET01590
      WRITE(6,*) '       IOUT=1 -> USE INFINITE ELEMENTS ON C. BOUND.'GET01600
      WRITE(6,*) '       IOUT=2 -> HOMOG. NEUMANN B. C. NEER APERTURE'GET01610
      WRITE(6,*) '       IOUT=3 -> GROUND PLANE + INFINITE ELEMENTS  'GET01620
      WRITE(6,*) '       IOUT=4 -> 2 GROUND PLANES + INFINITE ELEMENTS'GET01630
      WRITE(6,*) '                                                   'GET01640
      WRITE(6,*) '                                                   'GET01650
```

```
      WRITE(6,*) '                        IOUT ?                          'GET01660
      READ(IUNIT,*) IOUT                                                   GET01670
      ELSE                                                                 GET01680
      WRITE(6,10)                                                          GET01690
      WRITE(6,*) '                                                        'GET01700
      WRITE(6,*) '                                                        'GET01710
      WRITE(6,*) '       ENTER THE POSITION AND THE WIDTH OF THE          'GET01720
      WRITE(6,*) '        BACK PLANE                                      'GET01730
      WRITE(6,*) '                                                        'GET01740
      WRITE(6,*) '                                                        'GET01750
      WRITE(6,*) '                                                        'GET01760
      WRITE(6,*) '                                                        'GET01770
      WRITE(6,*) '                                <-- I                   'GET01780
      WRITE(6,*) '       YBP                                              'GET01790
      WRITE(6,*) '                                                        'GET01800
      WRITE(6,*) '                               WIDTH                    'GET01810
      WRITE(6,*) '                               <->                      'GET01820
      WRITE(6,*) '                                                        'GET01830
      WRITE(6,*) '                              XBP                       'GET01840
      WRITE(6,*) '                                                        'GET01850
      WRITE(6,*) '                          XBP , YBP ? (in meters)        'GET01860
      WRITE(6,*) '                          WIDTH     ? (in meters)        'GET01870
      READ(IUNIT,*) XW,YW,A                                                GET01880
      WRITE(6,10)                                                          GET01890
      WRITE(6,*) '                                                        'GET01900
      WRITE(6,*) '                                                        'GET01910
      WRITE(6,*) '                                                        'GET01920
      WRITE(6,*) '       ENTER THE PEAK CURRENT STRENGTH IN AMPERS (A)'GET01930
      WRITE(6,*) '        THEN ALPHA & BETA, THE CHARACTERISTIC OF    ' GET01940
      WRITE(6,*) '        EMP TYPE CONDUCTING CURRENT                 ' GET01950
      WRITE(6,*) '                                                        'GET01960
      WRITE(6,*) '     I(t) = IO * [exp(-ALPHA * t) - exp(-BETA * t)] 'GET01970
      WRITE(6,*) '            ?           ?                 ?         ' GET01980
      WRITE(6,*) '                                                        'GET01990
      WRITE(6,*) '                                                        'GET02000
      WRITE(6,*) '              IO (A), ALPHA , BETA ?                '  GET02010
      READ(IUNIT,*) HO,ALPHA,BETA                                          GET02020
      HO = 4.0D-07*PI*HO/((HE-2.0D0*YW)*A)                                 GET02030
      ENDIF                                                                GET02040
C                                                                          GET02050
      NREG = 0                                                             GET02060
      WRITE(6,10)                                                          GET02070
      WRITE(6,*) '                                                      ' GET02080
      WRITE(6,*) '                                                      ' GET02090
      WRITE(6,*) '       NOW LET"S TALK OUTPUT !!!!                     ' GET02100
      WRITE(6,*) '                                                      ' GET02110
      WRITE(6,*) '       FIRST WOULD YOU LIKE TO ANALYZE THE WHOLE      ' GET02120
      WRITE(6,*) '        REGION (Y/N)?                                 ' GET02130
      WRITE(6,*) '                                                      ' GET02140
      READ(IUNIT,'(A1)') CHAR                                              GET02150
      IF (CHAR .EQ. 'Y'  .OR.  CHAR .EQ.'y') GO TO 400                     GET02160
      WRITE(6,10)                                                          GET02170
      WRITE(6,*) '                                                      ' GET02180
      WRITE(6,*) '                                                      ' GET02190
      WRITE(6,*) '       HOW MANY REGIONS THEN WOULD YOU LIKE TO        ' GET02200
```

```
      WRITE(6,*) '        ANALYZE (PLEASE DO ENTER LARGE ENOUGH       ' GET02210
      WRITE(6,*) '                 RECTANGULAR REGIONS AS THE         ' GET02220
      WRITE(6,*) '                 TRIANGULARIZATION IS AUTOMATED)    ' GET02230
      I1 = 0                                                            GET02240
500   WRITE(6,*) '                                                    ' GET02250
      WRITE(6,*) '              # OF REGIONS ?                        ' GET02260
      READ(IUNIT,*) NREG                                                GET02270
      IF (NREG .GT. 10) THEN                                            GET02280
       WRITE(6,*) '                                                   ' GET02290
       WRITE(6,*) '       THE NUMBER OF MAXIMUM REGIONS  IS TEN (10)  ' GET02300
       WRITE(6,*) '        PLEASE ENTER OVER THE NUMBER OF REGIONS    ' GET02310
       IF (I1 .GT. 3) STOP                                              GET02320
       I1 = I1 + 1                                                      GET02330
       GO TO 500                                                        GET02340
      ENDIF                                                             GET02350
C                                                                       GET02360
      WRITE(6,*) '                                                    ' GET02370
      WRITE(6,*) '                                                    ' GET02380
      WRITE(6,*) '                          * (X2,Y2)                 ' GET02390
      WRITE(6,*) '                                                    ' GET02400
      WRITE(6,*) '                                                    ' GET02410
      WRITE(6,*) '                                                    ' GET02420
      WRITE(6,*) '   (X1,Y1) *                                        ' GET02430
      DO 600 I = 1 , NREG                                               GET02440
       WRITE(6,*) '                                                   'GET02450
       WRITE(6,'(20H        FOR REGION #,I2,15H PLEASE ENTER :)') I    GET02460
       WRITE(6,*) '                X1 , Y1 ? (in meters)              'GET02470
       READ(IUNIT,*) X1(I),Y1(I)                                        GET02480
       WRITE(6,*) '                X2 , Y2 ? (in meters)              'GET02490
       READ(IUNIT,*) X2(I),Y2(I)                                        GET02500
600   CONTINUE                                                          GET02510
400   CONTINUE                                                          GET02520
      WRITE(6,10)                                                       GET02530
      WRITE(6,*) '                                                    ' GET02540
      WRITE(6,*) '                                                    ' GET02550
      WRITE(6,*) '       AS A FINAL QUESTION, DO YOU PREFER TO SET     ' GET02560
      WRITE(6,*) '        THE TIMES FOR AT WHICH THE FIELDS ARE TO    ' GET02570
      WRITE(6,*) '        BE DETERMINED (TIM = 1), OR USE THE DEFAULT'  GET02580
      WRITE(6,*) '        VALUES CALCULATED INTERNALLY (TIM = 0)?     '  GET02590
      WRITE(6,*) '          (IF YOU CHOOSE TIM = 1 SOME CRITICAL      '  GET02600
      WRITE(6,*) '           TIME VALUES SHALL BE GIVEN TO YOU)       ' GET02610
      WRITE(6,*) '                                                    ' GET02620
      WRITE(6,*) '                TIM ?                               ' GET02630
      READ(IUNIT,*) ITIME                                               GET02640
      IF (ITIME .EQ. 1) THEN                                            GET02650
       WRITE(6,10)                                                      GET02660
       WRITE(6,*) '                                                   'GET02670
       WRITE(6,*) '                                                   'GET02680
       WRITE(6,*) ' f(t)                                              'GET02690
       WRITE(6,*) '                                                   'GET02700
       WRITE(6,*) '                                                   'GET02710
       WRITE(6,*) '                                                   'GET02720
       WRITE(6,*) '                                                   'GET02730
       WRITE(6,*) '                                                   'GET02740
       WRITE(6,*) '                                          > t      'GET02750
```

```
      WRITE(6,*) '                |                       |                  'GET02760
      WRITE(6,*) '              tpeak                   tfin                 'GET02770
      WRITE(6,*) '                                                           'GET02780
      WRITE(6,*) '          WHERE  f(tfin) = 0.1 * f(tpeak)                  'GET02790
      WRITE(6,*) '                                                           'GET02800
C                                                                             GET02810
      TPEAK = DLOG(BETA/ALPHA)/(BETA-ALPHA)                                   GET02820
      EPEAK = DEXP(-ALPHA*TPEAK) - DEXP(-BETA*TPEAK)                          GET02830
      DO 700 I = 2 , 10000                                                    GET02840
        IF (-BETA*TPEAK*REAL(I) .LE. -120.D0) THEN                            GET02850
          IF (DEXP(-ALPHA*TPEAK*REAL(I))  .LT. 0.10D0*EPEAK) GO TO 800        GET02860
        ELSE                                                                  GET02870
          IF ((DEXP(-ALPHA*TPEAK*REAL(I)) - DEXP(-BETA*TPEAK*REAL(I)))        GET02880
     &       .LT. 0.10D0*EPEAK) GO TO 800                                     GET02890
        ENDIF                                                                 GET02900
700   CONTINUE                                                                GET02910
800   CONTINUE                                                                GET02920
      TFIN  = REAL(I)*TPEAK                                                   GET02930
      TDIFF = 4.0D-07*PI*MIURW*SIG*WE**2                                      GET02940
      TFINA = TFIN + TDIFF                                                    GET02950
      WRITE(6,*) '                                                           'GET02960
      WRITE(6,*) '          HERE ARE THE CRITICAL TIME VALUES :              'GET02970
      WRITE(6,*) '          ----------------------------------               'GET02980
      WRITE(6,*) '                                                           'GET02990
      WRITE(6,'(7X,8Htpeak = ,E10.4,5X,7Htfin = ,E10.4)') TPEAK,TFIN          GET03000
      WRITE(6,'(7X,8Htdiff = ,E10.4,5X,9Htfinal = ,E10.4)') TDIFF,TFINAGET03010
      WRITE(6,*) '                                                           'GET03020
      WRITE(6,*) '          tdiff IS THE REQUIRED TIME FOR DIFFUSION         'GET03030
      WRITE(6,*) '          AND tfinal IS THE OVERALL FINAL INTEGRATION      'GET03040
      WRITE(6,*) '          TIME CALCULATED AS tfinal = tfin + tdiff         'GET03050
      WRITE(6,*) '                                                           'GET03060
      WRITE(6,*) '         PLEASE ENTER THE NUMBER OF SPECIFIC TIMES         'GET03070
      WRITE(6,*) '          FOR WHICH YOU ARE INTERESTED FOR THE             'GET03080
      WRITE(6,*) '          INVESTIGATION                                    'GET03090
900   WRITE(6,*) '                                                           'GET03100
      WRITE(6,*) '              # OF INTEGRATION TIME ?                      'GET03110
      READ(IUNIT,*) NINT                                                      GET03120
      IF (NINT .GT. 50) THEN                                                  GET03130
       WRITE(6,*) '                                                          'GET03140
       WRITE(6,*) '        THE MAXIMUM # OF INT. TIMES IS FIFTY (50)         'GET03150
       WRITE(6,*) '         PLEASE ENTER OVER THE NUMBER OF INT. TIME         'GET03160
       IF (I1 .GT. 3) STOP                                                    GET03170
       I1 = I1 + 1                                                            GET03180
       GO TO 900                                                              GET03190
      ENDIF                                                                   GET03200
C                                                                             GET03210
      WRITE(6,*) '                                                           'GET03220
      WRITE(6,*) '          IF THE INVESTIGATION IS CARRIED OUT THROUGH       'GET03230
      WRITE(6,*) '          CONSTANT TIME STEPS ENTER 1, 0 OTHERWISE         'GET03240
      WRITE(6,*) '                                                           'GET03250
      WRITE(6,*) '                        1 or 0 ?                           'GET03260
      READ(IUNIT,*) NSTE                                                      GET03270
      IF (NSTE .EQ. 0) THEN                                                   GET03280
       DO 1000 I = 1 , NINT                                                   GET03290
        WRITE(6,*) '                                                         'GET03300
```

```
        WRITE(6,'(40H        PLEASE ENTER INTEGRATION TIME # ,I2)') I  GET03310
        WRITE(6,*) '                    (in seconds)                 'GET03320
        WRITE(6,*) '                                                 'GET03330
        READ(IUNIT,*) TINT(I)                                         GET03340
        WRITE(6,*) '                                                 'GET03350
1000    CONTINUE                                                      GET03360
       ELSE                                                           GET03370
        WRITE(6,*) '                                                 ' GET03380
        WRITE(6,*) '       PLEASE ENTER THE INTEGRATION TIME STEP    ' GET03390
        WRITE(6,*) '                    (in seconds)                 ' GET03400
        WRITE(6,*) '                                                 ' GET03410
        READ(IUNIT,*) TINTS                                           GET03420
        WRITE(6,*) '                                                 ' GET03430
        TINT(1) = TINTS                                               GET03440
        DO 1100 I = 2 , NINT                                          GET03450
          TINT(I) = TINT(I-1) + TINTS                                 GET03460
1100    CONTINUE                                                      GET03470
       ENDIF                                                          GET03480
      ENDIF                                                           GET03490
C                                                                     GET03500
C                                                                     GET03510
C  >-------------------------------------------------------------<    GET03520
C  >              RETURN TO THE MAIN PROGRAM                     <    GET03530
C  >-------------------------------------------------------------<    GET03540
C                                                                     GET03550
10    FORMAT('1')                                                     GET03560
      RETURN                                                          GET03570
      END                                                             GET03580
```

Bibliography

[1] Clough, R.W., "The finite element method in plane stress analysis," Proc. 2nd Conf. Electronic Comput., ASCE, Pittsburg, Pa., 1960.

[2] Winslow, A.M., "Magnetic field calculation in an irregular triangular mesh," Lawrence Radiation Laboratory, Livermore, California, UCRL-7784-T, Rev. 1, 1965.

[3] Costache, G.H., "Calculation of eddy currents and skin effect in nonmagnetic conductors by the finite element method," Rev. Roum. Sci. Techn.-Electrotechn. et Energ., Bucarest, vol. 21, no. 3, pp. 357-363, 1976.

[4] Silvester, P. and C.R.S. Haslam, "Magnetotelluric modelling by the finite element method," Geophysical Prospecting, vol.20, pp. 872-891, 1972.

[5] Brauer, J.R. "Finite elements calculation of eddy currents and skin effects," IEEE Trans. Magn., vol. MAG-18, NO. 2, PP.504- 509, March 1982.

[6] Silvester, P. and M.V.K. Chari, "Finite element solution of saturable magnetic field problems, "IEEE Trans. PAS, vol. PAS- 89, 1642-1651, 1970.

[7] Anderson, O.W., "Laplacian electrostatic field calculations by finite elements with automatic grid generation," IEEE Trans. PAS, vol. PAS-92, 1973.

[8] Chari, M.V.K. and P. Silvester, "Analysis of turbo-alternator magnetic fields by finite elements," IEEE Trans PAS, vol. PAS- 90, PP. 454-464, 1971.

[9] Wexler, A., "Finite-element analysis of inhomogeneous anisotropic reluctance machine rotor," IEEE Trans. PAS, vol. PAS-92, no.1, pp. 145-149, 1973.

[10] Akhtarzad S. and P.B. Johns, "Numerical solution of lossy waveguides: T.L.M. computer program," Electron. Lett., vol. 10, no. 15, pp. 309-311, July 25, 1974.

[11] Akhtarzad S. and P.B. Johns, "Three dimensional transmission-line matrix computer analysis of microstrip resonators," IEEE Trans. Microwave Theory Tech., vol. MTT-23, pp. 990-997, Dec. 1975.

[12] Akhtarzad S. and P.S. Johns, "Solution of Maxwell's equations in three space dimensions and time by the T.L.M. method of analysis, "Proc. Inst. Elec. Eng., vol. 122, no.12, pp. 1344- 1348, Dec. 1975.

[13] Al. Mukhtar D. A. and J.E. Sitch, "Transmission-line matrix method with irregularly graded space, "Proc. Inst. Elec. Eng., vol. 128, no. 6, pp. 299-305, Dec. 1981.

[14] Shih, Y.-C. and W.J.R. Hoefer, "Dominant and second-order mode cutoff frequencies in fin lines calculated with a two-dimensional TLM program," IEEE Trans. Microwave Theory Tech., vol. MTT-28, pp. 1443-1448, Dec. 1980.

[15] Shih, Y.-C., " The analysis of fin lines using transmission line matrix and tranverse resonance methods," M.A.Sc. thesis, Univ. of Ottawa, Canada, 1980

[16] Yoshida, N. I. Fukai and J.Fukuoka, "Transient analysis of two-dimensional Maxwell's equations by Bergeron's method," Trans. IECE Japan, vol. J62B, pp. 511-518, June 1979.

[17] Yoshida, N., I. Fukai and J. Fukuoka, "Application of Bergeron's method to anisotropic medias," Trans. IECE Japan, vol. J64B, pp. 1242-1249, Nov. 1981.

[18] Hoefer, W.J.R., "The transmission-line matrix method — Theory and applications," IEEE Trans. Microwave Theory Tech., vol. MTT-33, no.10, pp. 882-893, Oct. 1985.

[19] Spiegel, M.R., *Theory and Problems of Laplace Transforms*, Schaum's Outline Series, McGraw-Hill, 1965.

[20] Shermann, R., R.A. DeMoss, W.C. Freeman, G.J. Greco, D.G. Larson, L. Levey, and D. S. Wilson, *EMP Engineering and Design Principles*, Bell Laboratories, Whippany, New Jersey, 1975.

[21] Rektorys, K., *Variational Methods in Mathematics, Science and Engineering*, Reidel Publication Co, 1980.

[22] Foggia, A., J.C. Sabonnadiere and P. Silvester,"Finite element solution of saturated travelling magnetic field problems," IEEE Trans. Power Ap. Sys., vol. PAS-94, no. 3, pp.866-871, May/June 1975.

[23] Kreyszig, E., *Advanced Engineering Mathematics*, John Wiley & Sons, New York, 1983.

[24] Silvester, P.P., *Finite Elements for Electical Engineers*, Cambridge University Press, Cambridge, 1983.

[25] Silvester, P.P., "Higher-order polynomial triangle finite elements for potential problems," Int. J. Eng. Science, vol. 7, no.8-F, pp. 849-860, 1969.

[26] Cheng, D.K., *Field and Wave Electromagnetics*, Addison-Wesley, Massachusettes, 1983.

[27] Anton, H., *Elementary Linear Algebra*, John Wiley & Sons, New York, 1981.

[28] Ney, M. and G. Costache, "Penetration des impulsions electromagnetiques dans les blindages cylindriques de section arbitraire," Colloque international en compatibilite electromagnetique, Limoges, juin 1987.

[29] International Mathematical and Statistical Librairies, inc., "IMSL library reference manual," Texas, Houston, 1982.

[30] Petzold, L., "Differential/algebraic equations are not ODE's," SIAM J. Sci. Stat. Comput., vol. 3, no. 3, Sept. 1982.

[31] Petzold, L.R., "A description of DDASSL: a differential/algebraic system solver," Sandia National Laboratories, Livermore, SAND82-8637, Sept. 1982.

[32] Huygens, C., *Traite de la lumiere*, Leiden, 1690.

[33] Johns, P.B. and R.L. Beurle, "Numerical solution of 2-dimensional scattering problems using a transmission-line matrix," Proc. Inst. Elec. Eng., vol. 118, no. 9, pp. 1203-1208, Sept. 1971.

[34] Johns, P.B., "The solution of inhomogeneous waveguide problems using transmission-line matrix," IEEE Trans. Microwave Theory Tech., vol. MTT-22, no. 3, pp. 209-215, March 1974.

[35] Akhtarzad, S. and P.B. Johns, "Generalized elements for t.l.m. of numerical analysis," Proc. Inst. Elec. Eng., vol. 122, no. 12, pp. 1349-1352, Dec. 1975

[36] Saguet, P., E. Pic, "Le maillage rectangulaire et le changement de maille dans la methode TLM en deux dimensions," Electron. Lett., vol. 17, no. 7, pp. 277-279, April 2, 1981.

[37] Saguet, P., "Analyse des milieux guides, la methode MTLM," Ph.D. thesis, Universite Scientifique et Medical - Institut National Polytechnique de Grenoble, Dec. 1985.

[38] Konrad, A., "A direct three-dimensional finite element method for the solution of electromagnetic fields in cavities", IEEE Trans. Magn., vol. MAG-21, no. 6, pp. 2276-2279, November 1985.

[39] Akhtarzad, S. and P.B. Johns, "Solution of G-component electromagnetic fields in three sapce dimensions and time by the T.L.M. method, " Electron. Lett., vol. 10, no. 25/26, pp. 535-537, Dec. 12, 1974.

[40] *FCC Part 15J Rules and Regulations*, 1980.

[41] Shen, L.C. and J.A. Kong, *Applied Electromagnetism*, PWS, Massachusetts, Boston, 1983.

[42] Hoefer, W.J.R., The Transmission Line Matrix (TLM) Method, in T. Itoh: *Numerical Techniques for Microwave and Millimeter Wave Passive Structures*, John Wiley & Sons, New York, 1989.

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

[44] Costache, G.I., "Finite element method applied to skin-effect problems in strip transmission lines", IEEE Trans. Microwave Theory Tech., vol. MTT-35, no. 11, pp. 1009-1013, Nov. 1987.

[45] Ungless, R.F., "An infinite element", M.A. Sc. Thesis, University of British Columbia, 1973.

[46] Pissanetzky, S., "A simple infinite element", Int. J. Comput. Math. Elec. Eng., vol. 3, no. 2, pp. 107-114, 1984.

[47] Wilson, P.F. and M.T.Ma, "A study of techniques for measuring the electromagnetic shielding effectiveness of materials", National Bureau of Standards, Colorado, NBS Technical Note 1095, 1986.