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.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

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

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.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

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

POWER HANDLING CAPABILITIES OF E-PLANE METAL-INSERT FILTERS

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
August, 1988

ISBN 0-315-46742-8

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Pour Maman

Acknowledgements.

I would like to thank my supervisor, Dr. Michel M. Ney, for his encouragement and guidance throughout this research project. I would also like to thank Dr. George I. Costache for his invaluable help, and Dr. R. Vahldieck for providing comparative data using the mode matching method.

I would also thank Dr. Angelo Mingarelli for discussions on functional analysis, and several of my fellow students for fruitful discussions.

I would like to express my appreciation to the Natural Sciences and Engineering Research Council and the University of Ottawa for providing the financial support for this work.

I thank my parents and Marion for their patience and comprehension.

Abstract

The electromagnetic field distribution in a class of E-plane metal insert filters, operating in the Ka-band, is evaluated using a novel approach based on the finite element method (FEM). Artificial boundary conditions are inserted directly in a functional, enabling a solution of the scalar wave equation by the FEM. The scattering parameters, electromagnetic fields and the current distribution on the thin metal inserts are determined. Finally, the power dissipated in the metallic septa, which is suspected to be the major limitation for the maximum CW power that these type of filters can sustain, and the maximum field intensity, for an estimate of the maximum peak power that these structures can handle are determined.

The scattering parameters of the filter and the field distributions at the front end of the input septum are compared with results from mode matching technique and the transmission-line matrix (TLM) methods. Finally, the current distributions in the septa are compared with those computed with TLM. It is found that the results obtained by the various methods agree fairly well.

Notes:

All underlined symbols represent phasor quantities.

$\underline{a}^*$ = complex conjugate of $\underline{a}$.

$\mu_o = 4\pi \times 10^{-7}$ H/m

$\epsilon_o = 8.85419 \times 10^{-12}$ F/m.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction.

Millimeter-wave filters have drawn particular attention over the past few years. Their utilization at the output stage of satellite transponders or terrestrial stations is at the stage of development. As a result, due to the relatively high power level which will exist in the transmission medium, questions about the capability of such medium to sustain such levels of power should be raised, as very little data is presently available. The limitations of the maximum power that metal insert filters can sustain, follows from two major factors:

1. breakdown phenomena that may occur in the air or dielectric.

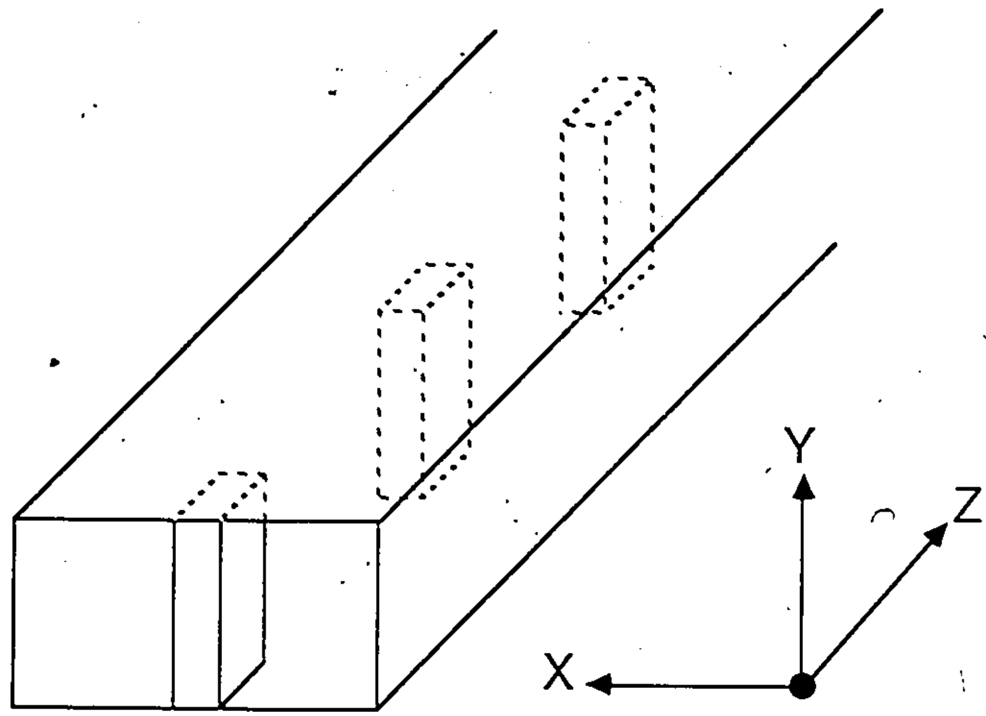
2. over-heating of the conducting strips due to skin effect at high frequency.

Generally, electric field breakdown does not occur simultaneously with over-heating phenomena. Either one may occur first, depending on the type of signals that is being transmitted. For instance, pulsed signals which are used for radar systems have relatively low average power and phenomena such as heating of conductors are most likely to occur after breakdown phenomena since the peak power, i.e. the field strength peak value, is very high in this case. On the other hand, for CW signals which are used in telecommunication systems, one faces the opposite situation in which the dissipations in the conducting strip, which tends to increase with the frequency because of the skin effect, are the most significant.

The structure of typical metallic insert filters and the definition of all relevant geometrical parameters is shown in fig. 1.1. It consists of single or twin ladder-type metallic inserts centered in the E-plane of a rectangular waveguide. The metallic structures can be obtained by etching technique for thicknesses smaller than 0.2 mm and by filing or stamping techniques for larger values. Etching has its advantage over other methods by the fact that it allows a mass production at very low cost. On the other hand, the relatively low thickness values obtainable with this technique render the structure fragile against mechanical stress and over-heating. The dielectrically-supported filters offer an added advantage of being manufactured on the same substrate as a possible neighboring finline circuit, thus easing circuit interconnection. As an example, a millimeter-wave receiver can incorporate, on the same dielectric substrate, a pre-filtering metal-insert filter for the front-end, as well as the rest of the standard finline-technology receiver.

In order to determine the power limitation, due to either breakdown or over-heating, one has to determine the electromagnetic field in the structure. Among the various techniques proposed for the design and optimization of such structures [17]-[25], the most popular is the mode matching technique [17]-[23] which can account for conductor thickness and groove effects in the case of finlines. Unfortunately, the filter characteristics obtained by these techniques are determined by the cascade connections of discontinuities characterized by their S-scattering parameters. Thus, the information about the internal field distribution, which is relevant to the above mentioned application, is lost. An alternative would be to de-embed the different sections and retrieve the fields. Unfortunately, the presence of higher order modes leads to some numerical instabilities.

The approach presented in this thesis uses the finite element method (FEM). Its flexibility allows one to account for the thickness of the septum, which is essential to a)



b)

c)

d)

e)

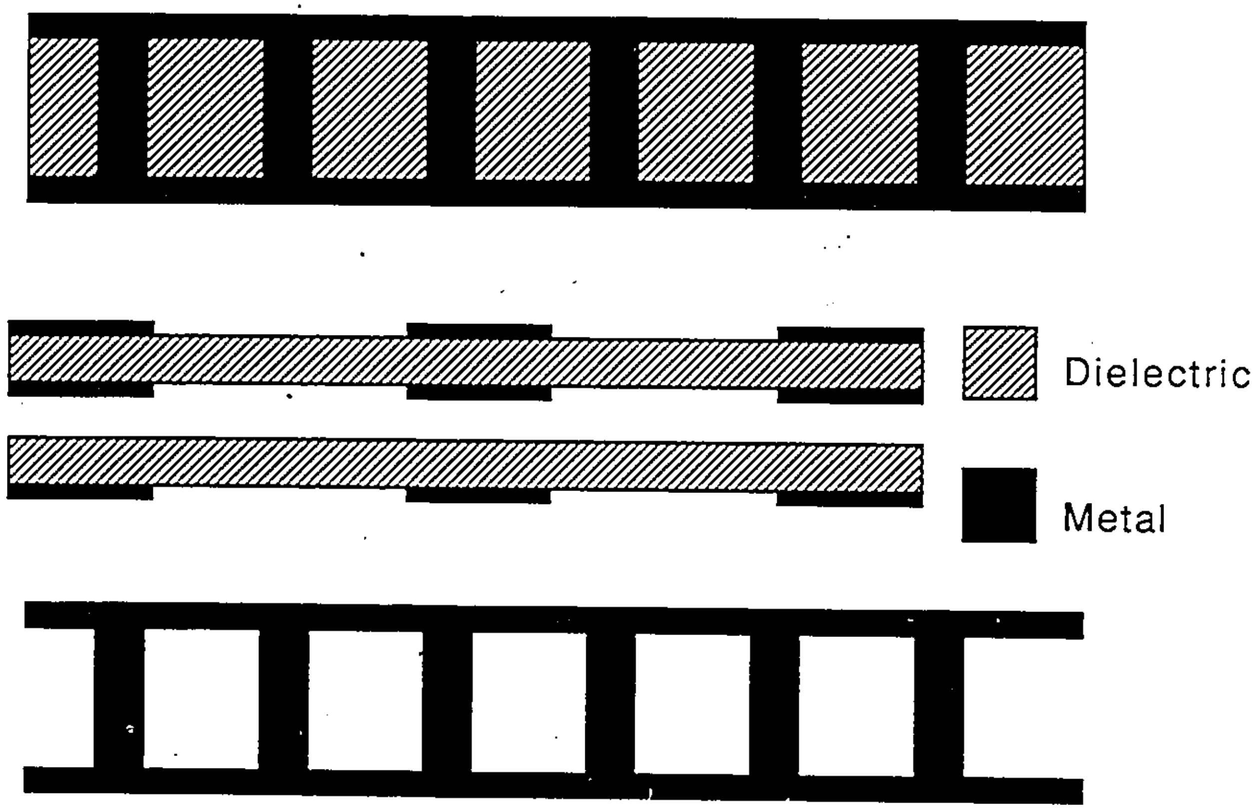


Figure 1.1: E-plane insert filters: a) inside view. b) Top view of dielectrically supported septum using c) bilateral finline technology d) unilateral finline technology. e) Fully metallic metal insert.

correctly model the structure. In addition, the new formulation presented here includes mixed boundary conditions in the functional so that the geometry which is boundless can be treated as a bounded structure. A reader, unfamiliar with FEM, may wish to consult Yue's Masters thesis [9], which offers an excellent discussion on FEM applied to a simpler static field case.

1.2 Literature survey.

To the author's knowledge, the first occurrence of this specific class of filters to appear in any publication was in 1972 by Konishi (see reference by Konishi in 1974 [24]). Subsequent publications are due mostly to Vahldieck et al. [17]-[21] and Itoh et al. [22,23] Both groups used a form of mode-matching method to analyze the filters. The reported insertion losses for such filters is typically 0.5 dB at 35 GHz [17]-[21]. Out-of-band insertion loss can easily reach 50 dB for a frequency 6% out of pass-band [18]. Spurious pass-bands appear at higher frequencies beyond the dominant mode operation of the waveguide [20,21]. Vahldieck and Hoefer [21] recently proposed a variation on the basic insert filter which consists in placing two parallel sets of inserts in an oversized waveguide, yielding increased stop-band attenuation, and improving pass-band operation. The more recent work on E-plane filter design is due mostly to Vahldieck et al. [19]-[21].

Both groups (Vahldieck et al. and Itoh et al.) developed a CAD (computer assisted design) package that can optimize a class of E-plane filters according to specifications such as bandwidth, out-of-band rejection, etc. Both approaches required the computation of the generalized scattering matrix (noted S) at each discontinuity encountered. Such a matrix differs from the regular 2 by 2 S matrix in that the latter gives information on the fundamental propagating mode only, while the generalized S matrix accounts for all mode interactions at a junction. This is necessary, since this type of filter is essentially a series of closely coupled cavities, all of which are coupled together only through higher order evanescent modes. Indeed, septa split the waveguide enclosure in two waveguides, neither of which can support a propagating mode in the frequency band of interest. The inclusion of the effect of an increasing number of modes in the generalized scattering matrices makes the field matching more accurate, at each discontinuity, but creates a larger system to solve.

A note of interest is the fact that, while a large number of modes is needed to match fields at the junctions, a much smaller amount are needed to account for the coupling between cavities. This is quite normal, since very high order modes nearly vanish before reaching the other end of a cavity, precluding interactions between discontinuities. The overall scattering parameters of the complete filter are obtained by combining individual generalized scattering matrices and retaining the terms pertaining to the fundamental mode for the resulting generalized S matrix.

1.3 Motivation.

This work has been made in response to a growing power need at millimeter-wave frequencies. As higher power levels are used in Earth and satellite transponders (it is foreseen that CW power up to 100 W will be needed at 94 GHz), a method of analysis is needed to predict the limitations of filter structures placed in the output section of such transmitters.

The method presented herein is not meant as a substitute method of analysis for the mode-matching technique (MM), but rather as a mean of analyzing a developed and working filter for parameters, such as field and current distribution, not yielded by MM. A suitable method has been developed to accurately predict the pulsed power limitations, for use with radar or any high-power pulsed communication system. Some preliminary results were obtained for use in calculating maximum CW power limitations, and some useful guidelines have been given.

1.4 Method.

The purpose of this work is to investigate the power-handling limits of E-plane metallic insert filters. The field distribution must be known everywhere in order to predict the peak incident power which will cause breakdown in the dielectric (in most cases, air). Further, the tangential H-field must be known everywhere on the surface of the septa to find their current distribution. This, in turn, can yield the power dissipation distribution on their surface by use of the small perturbations technique, which simply consists in assuming that the fields at the surface of a conductor with low resistivity (the real case) are the same as those for a perfect conductor (the theoretical model used herein).

Throughout this thesis, most field simulations will be done in the middle of the pass-band, as the filter is expected to receive most incident power in that narrow band. The computed reflection coefficient will be used to compare accuracy between all numerical methods investigated herein. The transmission coefficient will not be used for any comparisons, as small changes in reflection in the pass-band (where the reflection is very small) will cause negligible changes in the transmission coefficient.

Candidates for this problem are any method that will not analyze the fields as a set of modes, but as a total inseparable entity (in fact, only the total field exists: modes are a mathematical artifact). Of interest, then, are the finite element method (FEM), transmission-line matrix (TLM) method, and finite differences (FD) method. The latter will not be discussed in this thesis, but would probably yield results comparable to the first two. In fact, one may, up to a certain point, consider TLM to be a variation on FDTD (Finite Differences, Time-Domain). The first method (FEM) will be discussed in great detail. The TLM method will be used to compare results and computational efficiency.

Chapter 2

FEM basic theory.

2.1 Introduction

The FEM can be introduced by presenting some basics in calculus of variations, since FEM is itself very closely related to that subject. A skeletal understanding of this branch of mathematics is necessary in order to fully appreciate the contents of further chapters, although no deep notions are required. The implementation of a method to find an approximate solution of a variational problem (VP) is presented, before presenting the FEM, which is also used to solve a VP approximately.

2.2 Calculus of variations

2.2.1 Historical background.

One can introduce the calculus of variations by giving an example of a typical problem that was attacked by mathematicians as early as in ancient Greece: *What geometrical figure of a given perimeter will enclose the largest possible area, the domain of the solution being the set of all closed curves?* The circle was suggested as the solution, partly by intuition, experiments and geometry. But no formal proofs were proposed until the 19th century. In 1696, John Bernouilli(1667-1748) published a challenge to the mathematical world: the brachystochrone problem [6]. This classical problem is illustrated in fig. (2.1),

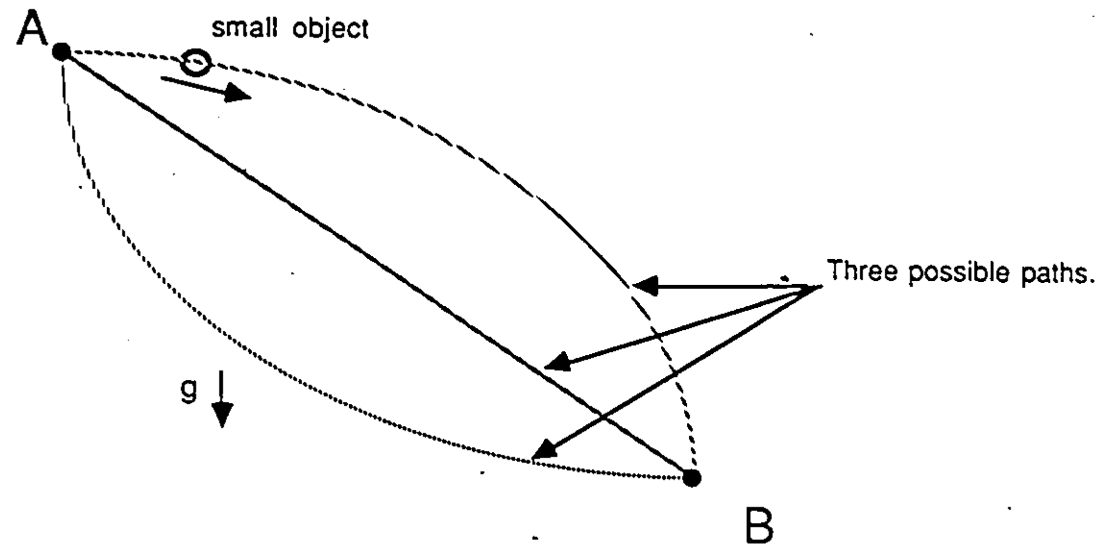


Figure 2.1: Illustration of the brachystochrone problem.

and is described as follows. *Suppose that a small object is dropped from point A and be forced to follow given frictionless paths down to a point B, and given that the only force acting on the object is gravity g, what path will the mass take in order to reach the point B with the minimum possible time?* Leonhard Euler (1707-1783), in 1744, first called this branch of mathematics "calculus of variations", after publishing some extensive work on the subject. He showed that the curve that will optimize an integral functional will satisfy the so-called associated Euler-Lagrange equation, which is sometimes a lot easier to solve than the original problem.

In general, the calculus of variations evolved partly by the work of theoretical mathematics, and partly by the work of physicists. Indeed, some very basic and powerful physics principles are described by the use of a variational principle. One example is Hamilton's principle of minimal action which states that the trajectory of a system of particles is the one which minimizes the "action integral"

$$J[y] = \int_{t_0}^{t_1} (T(y) - U(y))\, dt$$

where T = kinetic energy

U = potential energy

$y(t)$ = trajectory of particle system.

2.2.2 Basic notions.

To summarize, we can use a simple analogy: In ordinary calculus, we find a maximum on a curve $f(x)$ by first finding the points where the first derivative is zero. The first derivative is defined as

$$\lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} = \frac{df}{dx}$$

The extremas of a continuous function must lie on those points. In variational calculus, to find the curve that renders a certain functional J extreme, we must find the curves (there might be more than one) that make the first variation δJ zero, which is defined by

$$\lim_{\epsilon \to 0} \frac{J[y + \epsilon\eta] - J[y]}{\epsilon} = \delta J$$

where η is an arbitrary function in the class of admissible functions for that problem. One can note the similarity between the two limits.

2.3 The Rayleigh-Ritz method

The Rayleigh-Ritz method approximates the extremum of a functional $J[y]$ by replacing y by an approximate expansion of the type

$$y \cong \sum_{i=1}^{n} a_i \phi_i$$

where ϕ_i are linearly independent basis functions. $J[y]$, then, becomes a function of a_i, and the extremas can be found by simply setting the partial derivative of J with respect to the variational parameters a_i to zero. Thus, a variational calculus problem is reduced to a regular calculus problem. This method is guaranteed to converge for increasing n if the set of all ϕ_i is complete in the set containing the class of admissible functions for that particular problem [6]. In other words, every curve y in the set of admissible functions must be expressible as

$$y = \sum_{i=1}^{\infty} a_i \phi_i.$$

A well known example is the set

$$\mathcal{F} = \{1, \cos(\omega_o t), \sin(\omega_o t), \cos(2\omega_o t), \sin(2\omega_o t), \ldots\}$$

is complete in the set of all periodic functions with period $2\pi/\omega_o$ that satisfy the Dirichlet conditions.

2.4 The finite element method.

The FEM is a variation on the Rayleigh-Ritz method, in that the functional for which we wish to find the extremum is broken into a sum of individual terms. Mathematically,

$$\int_{\Omega} f(y)\, dy = \sum_{i=1}^{n} \int_{\Omega_i} f(y)\, dy$$

where $\bigcup_{i=1}^{n}(\Omega_i) = \Omega$

$\Omega_i \bigcap \Omega_j = \emptyset, \quad i \neq j$

The advantage, here, over the Rayleigh-Ritz method, is that we do not need to expand y into a sum of generally complicated functions. Indeed, if the set of all ϕ_i used in the Rayleigh-Ritz method were used to represent a highly irregular field distribution, then the expansion functions would have to be of a complicated analytical nature. This would ultimately imply analytical and computational difficulties. The expansion functions can be kept to their most simplistic form. For example,

$$y_i \cong a_i + b_i x$$

can be used as a simple expansion function in one-dimensional problems. The following can be used for two-dimensional problems:

$$f(x, y) \cong a_i + b_i x + c_i y$$

Thus, instead of solving one large and complicated problem, we simply have to solve a set of small simple problems, one in each subdomain Ω_i. This can be easily implemented and automated in a digital computer[3,4].

Several papers on the use of variational techniques to solve electromagnetic problems are mentioned in the bibliography [1,2],[27]-[32]. Several of these are relatively easy to understand, for a reader not familiar with variational methods [1,2]. An excellent book on mathematical physics [5] introduces variational techniques in physical problems in a very clear manner.

Chapter 3

Theory

3.1 Introduction

This chapter presents the basic electromagnetic theory relating to the geometry of the filter, and the procedure to obtain the functional associated with the field equations inside the structure. This functional will be used to obtain a variational solution to the field equations.

The simplest fully-metallic insert filter, which consists of 2 metallic inserts (fig. 1.1), is analyzed. Once the method has been proved accurate for a 2-septum filter, it can easily be extended with the same procedure presented herein to N-septum filters.

3.2 Geometry of the problem

The incident wave is a TE_{10} fundamental mode type. Because all obstacles are homogeneous in the y-direction, all higher-order modes locally generated by the obstacles will be of the type TE_{no}. All modes having $n > 1$ are evanescent. The fact that there are no TE_{om} or TE_{mn} modes generated considerably reduces the computational and analytical complexity of the problem.

An artificial boundary (since there is no discontinuity at this air-to-air interface) representing the input of the filter is located at $z = Z_1$ and is referred to as C_1. For ease of

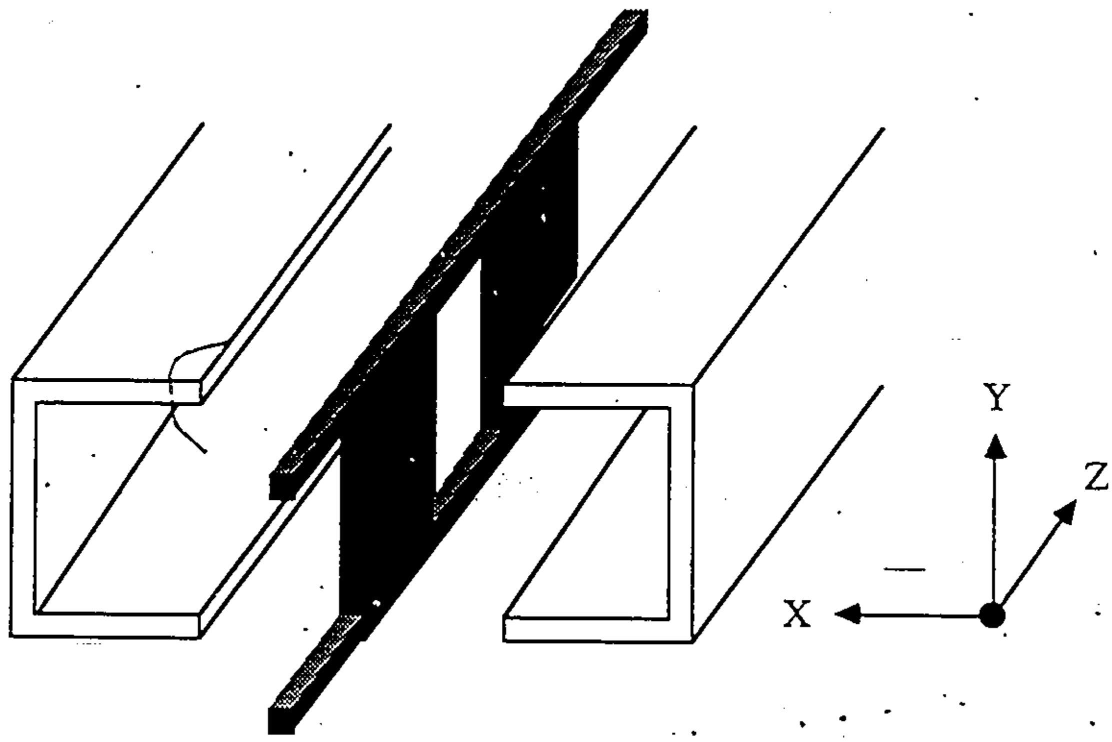


Figure 3.1: Blow-up view of simple 2-insert filter.

computation, Z_1 is chosen to be zero. The artificial boundary (again, not really a discontinuity) at the output of the filter is at $z = Z_2$, and is referred to as C_2. The output port can be assumed to be perfectly matched. This is a fairly reasonable assumption since, in most applications, the output will be a reasonably well matched antenna or an amplifier input port.

3.3 Boundary conditions formulation

The E-field in this structure has only y-components: $\vec{\underline{E}} = (0, \underline{E}_y, 0)$. This is because only TE_{no} modes exist in the filter. The boundaries C_1 and C_2 are chosen such that the higher-order modes have virtually vanished there. Since all electromagnetic quantities can be derived from the y-component of the E-field, then $\underline{E}_y$ is chosen as a scalar potential function $\underline{\phi}$. Figure (1.2) shows the value of $\underline{E}_y$ at different locations to the left of C_1. For clarity, the incident and reflected waves are shown separated. The total $\underline{E}_y$ field at the observation line $z = Z_0$ is the sum of incident and reflected fields at that line:

$$\underline{E}_y(x, Z_0) = \underline{E}_o \sin(k_x x)\,(e^{-jk_z Z_0} + \underline{R}e^{jk_z Z_0})$$

where $\underline{R}$ = reflection coefficient referred to $z = Z_1$

$k_x = \pi/a$

k_o = free-space wave number = $2\pi f/c$

$k_z = \sqrt{k_o^2 - k_x^2}$

$\underline{E}_o$ = phasor amplitude of incident wave at $z = Z_1 = 0$ (fig 1.3).

Z_1 is chosen as zero, for simplicity. Thus, $\underline{R}$ becomes the reflection coefficient referred to $z = 0$. Then the potential $\underline{\phi}$ at any point inside the guide to the left of C_1 becomes

$$\underline{\phi}(x, z) = \underline{E}_o \sin(k_x x)\,(e^{-jk_z z} + \underline{R}e^{jk_z z}) \tag{3.1}$$

where $\underline{R}$ = reflection coefficient referred to $z = 0$.

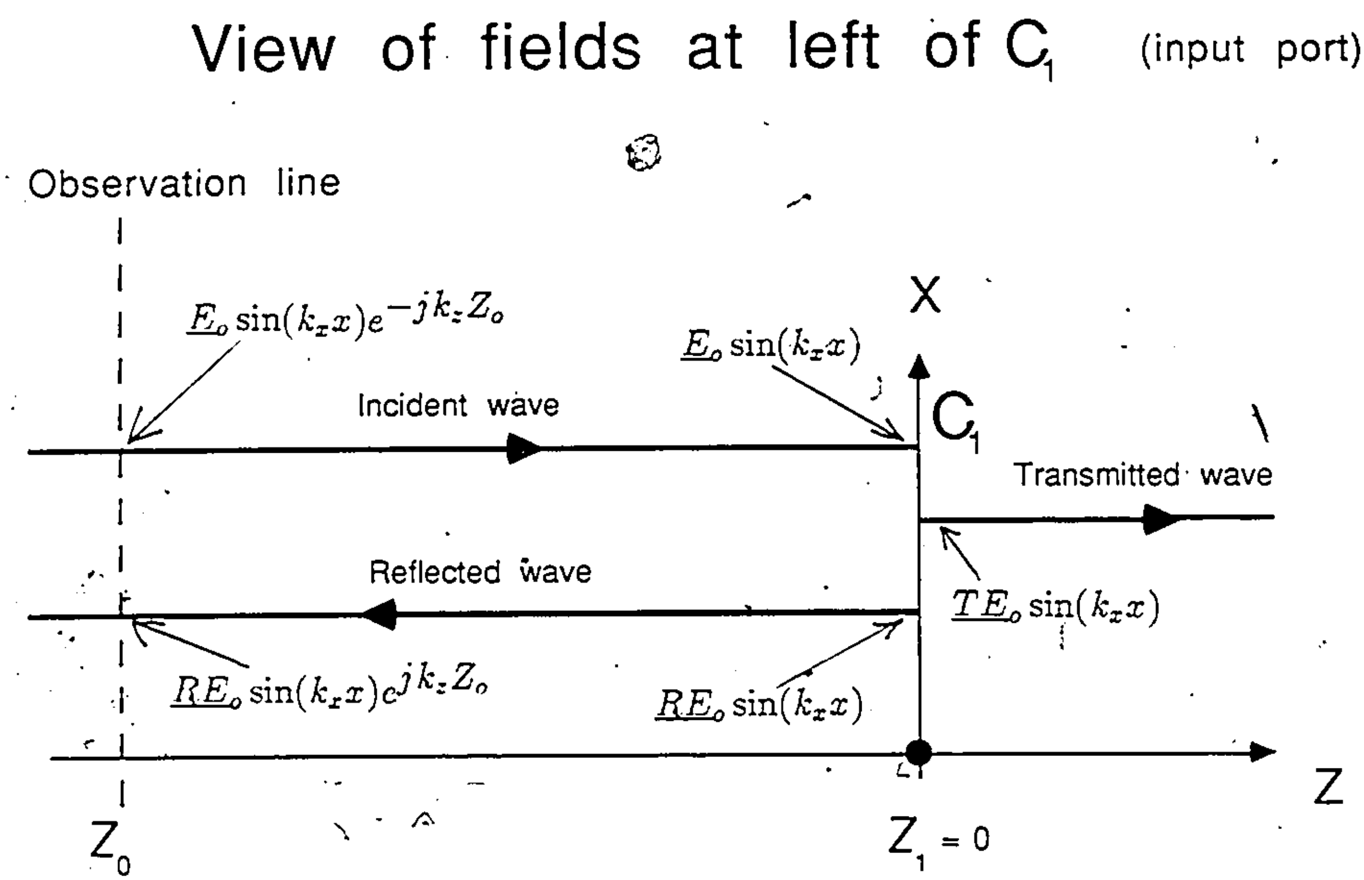


Figure 3.2: Conceptual view of waves present at input port.

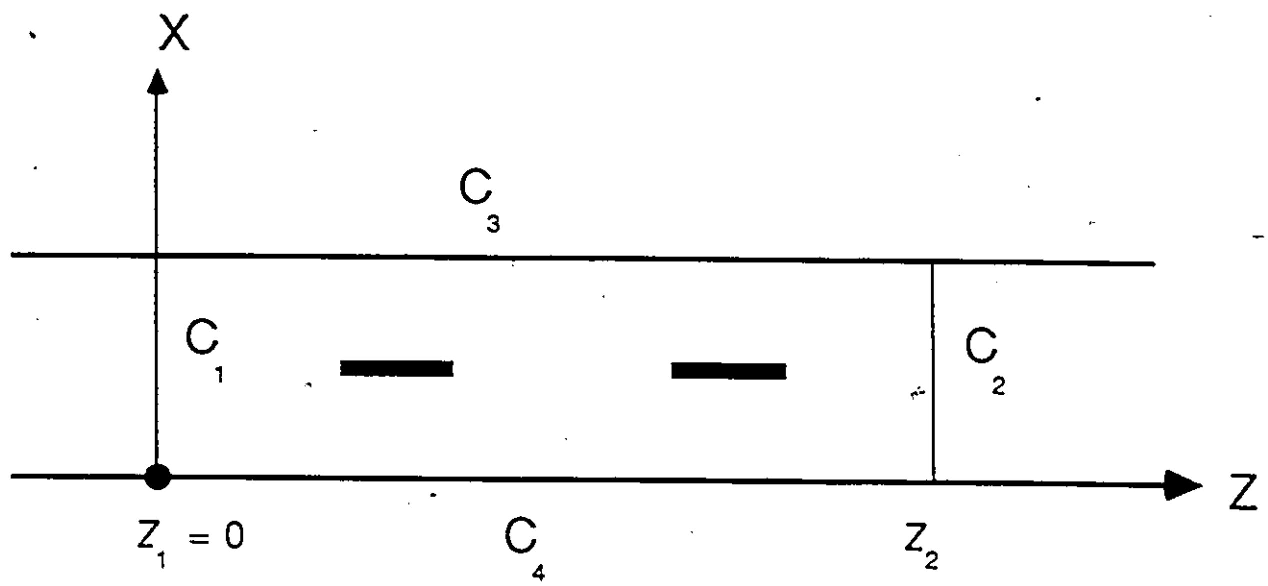


Figure 3.3: Top view of filter with axes.

Similarly, the potential $\underline{\phi}$ at any point in the waveguide to the right of boundary C_2 is

$$\underline{\phi}(x, z) = \underline{E}_o \sin(k_x x) \underline{T} \, e^{j k_z z} \tag{3.2}$$

where $\underline{T}$ = transmission coefficient referred to $z = 0$.

In order to find the boundary conditions of $\underline{\phi}$ at both interfaces C_1 and C_2, the continuity of the tangential E-field and H-field must be applied. We know from equations (3.1) and (3.2) the potential outside of the region bounded by C_1 and C_2. The distribution inside is not known. The continuity of the tangential component of the E-field across C_1 requires that

$$\underline{\phi}(x, Z_1) = \underline{E}_o \sin(k_x x) \left(e^{-j k_z Z_1} + \underline{R} e^{j k_z Z_1}\right) . \tag{3.3}$$

It is emphasized that the walls C_1 and C_2 are placed at such a distance from the discontinuities that only the fundamental propagating mode exist at the boundaries.

The continuity of the tangential component of the H-field requires that

$$\underline{H}_x = \frac{-j}{\omega\mu_0}\frac{\partial \underline{E}_y}{\partial z} = \frac{-j}{\omega\mu_0}\frac{\partial \underline{\phi}}{\partial z} = \frac{-j}{\omega\mu_0}\underline{E}_o \sin(k_x x) j k_z (-e^{-jk_z Z_1} + \underline{R}e^{jk_z Z_1}) \qquad (3.4)$$

The reason for choosing $Z_1 = 0$ should be obvious by now. All terms involving an exponential with Z_1 at the exponent become equal to one. This simplifies analysis. Combining (1.3) and (1.4) and eliminating $\underline{R}$ yields

$$\frac{\partial \underline{\phi}}{\partial z} - jk_z\underline{\phi} = -2jk_z\underline{E}_o \sin(k_x x). \qquad (3.5)$$

The boundary condition (BC) on C_1 then is

$$\frac{\partial \underline{\phi}}{\partial n} + jk_z\underline{\phi} = \underline{A} \qquad (3.6)$$

where $\frac{\partial}{\partial n}$ is a directional derivative operator with respect to $\vec{n} = (0, 0, -1)$ (outward normal unit vector), and $\underline{A} = 2jk_z\underline{E}_o \sin(k_x x)$. Equation (1.6) is an inhomogeneous Cauchy boundary condition (also known as inhomogeneous Robin boundary condition).

Similarly, using (1.2) and matching the tangential components of E and H fields on boundary C_2 yields

$$\frac{\partial \underline{\phi}}{\partial z} = -jk_z\underline{\phi}. \qquad (3.7)$$

Therefore, the boundary condition for C_2 is

$$\frac{\partial \underline{\phi}}{\partial n} + jk_z\underline{\phi} = 0 \qquad (3.8)$$

where $\frac{\partial}{\partial n}$ is a directional derivative operator with respect to $\vec{n} = (0, 0, 1)$ (outward normal unit vector). Equation (1.8) is an homogeneous Cauchy boundary condition.

The other two boundary conditions are fairly simple. Since $\underline{\phi}$ is 0 on the waveguide walls, then

$$\underline{\phi} = 0 \qquad (3.9)$$

on C_3 and C_4. Condition (1.9) is an homogeneous Dirichlet boundary condition.

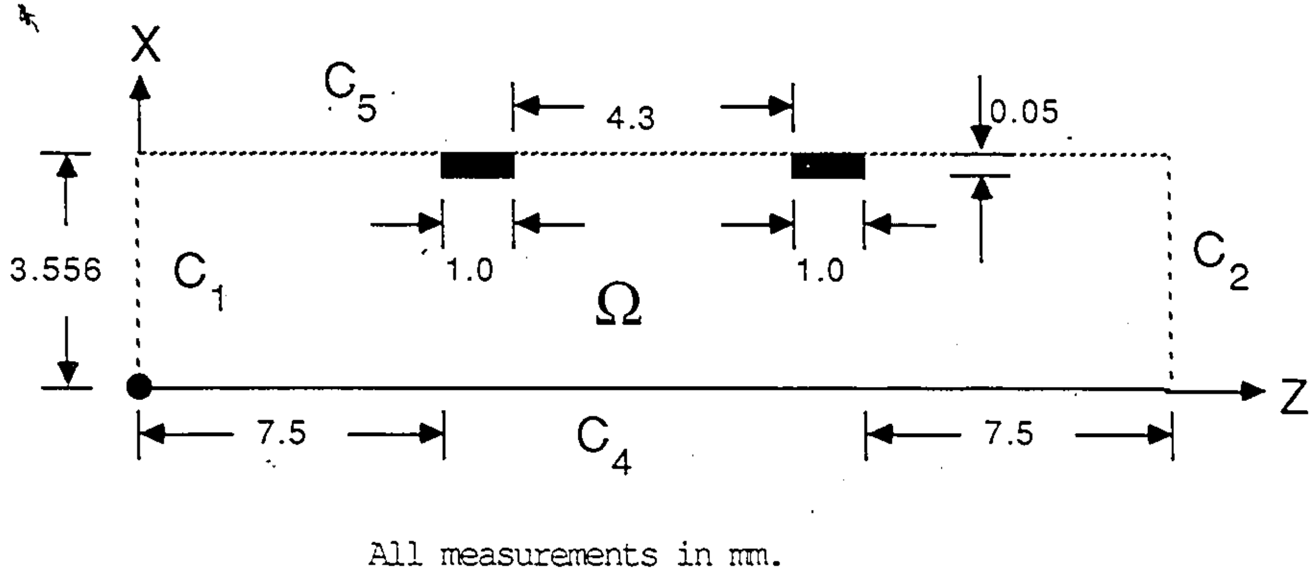


Figure 3.4: Definition of boundaries.

The structure is symmetric with respect to its median C_5 (fig. 1.4), and since the excitation is also symmetric about its median (TE_{10} mode), the field $\underline{\phi}$ will also have a symmetrical distribution about C_5, one can take advantage of this by introducing a new boundary at C_5 (a magnetic wall) and solving only half the problem. This reduces both the computer storage and CPU time. The new boundary condition, valid on C_5 except in regions intercepted by a septum, is

$$\frac{\partial \underline{\phi}}{\partial x} = 0. \tag{3.10}$$

or equivalently

$$\frac{\partial \underline{\phi}}{\partial n} = 0 \tag{3.11}$$

where $\frac{\partial}{\partial n}$ is a directional derivative operator with respect to $\vec{n} = (1, 0, 0)$ (outward normal unit vector). A direct consequence of this is that even-order modes are not present in the filter structure, except possibly in regions of the waveguide separated by a septum. In the present case, this represents the areas enclosed between $z = 7.5$ mm to 8.5 mm and between $z = 12.8$ mm to 13.8 mm. But, by using the property of continuity of tangential fields across air-to-air boundaries inside the filter, even order modes are seen not to exist in these regions, either. The condition (1.11) is an homogeneous Newmann boundary condition. Note that the boundary of Ω in fig. 1.4 is defined as

$$\Gamma = C_1 \cup C_2 \cup C_4 \cup C_5. \tag{3.12}$$

and has a counter-clockwise orientation.

3.4 Variational method formulation

To find the field inside the filter, one must solve the boundary value problem

$$L\underline{\phi} = v, \text{ in } \Omega \tag{3.13}$$

$$B\underline{\phi} = h, \text{ on } \Gamma$$

where $L = -\nabla^2 - k_o^2$

$v = 0$

$B = \partial/\partial n + jk_z$, on C_1 and C_2

$\partial/\partial n$, on C_5

1 , on C_4

$h = \underline{A}$, on C_1

0 , on all other boundaries.

It is understood that ∇^2 is the 2-D Laplacian operator. Equation 1.13 is, in fact, the wave equation, or Helmholtz equation, with the above boundary conditions. The operator notation is used according to the general method presented in [5] which finds a variational formulation for any operator.

The domain of L (noted D_L) is specified as the subset of $\mathcal{L}_2^{(c)}(\Omega)$ which contains all twice-differentiable functions. [1] The set $\mathcal{L}_2^{(c)}(\Omega)$ is chosen because only functions having finite average power are considered. The functions must also be twice differentiable in the region Ω to satisfy the operator L.

From [5], a functional to be made stationary about the solution of (1.13) is

$$F(\underline{u}) = < L\underline{u}, \underline{u} > - < \underline{u}, v > - < v, \underline{u} > \tag{3.14}$$

where $< \cdot, \cdot >$ is a complex inner product,

$\underline{u} = \underline{\phi}$ when (1.14) is stationary

Ω is the region of definition of L

and L is a symmetric operator.

The inner product is defined by:

$$< x, y > = < y, x >^*$$

[1] $\mathcal{L}_2^{(c)}(\Omega)$ is the set of Lebesgue integrable complex functions x such that $\int_\Omega | x(\Omega) |^2 \, d\Omega < \infty$.

$$< \alpha x, y > = \alpha < x, y >$$

$$< x_1 + x_2, y > = < x_1, y > + < x_2, y >$$

$$< x, x > \geq 0, \quad and \quad < x, x > = 0 \ \ iff \ \ x \equiv 0$$

where α is a complex constant, and x and y are defined on an open set Ω (x^* denotes the complex conjugate of x). The inner product chosen is:

$$< u, v > \; = \; \iint_\Omega uv^* \, d\Omega$$

The reader can easily verify that the above satisfies the definition of an inner product.

An operator L is symmetric if:

$$< u, Lv > \; = \; < Lu, v > .$$

for every element $u, v \in D_L$.

Unfortunately, L, as specified in (1.13) is not symmetric, and therefore (1.14) is not directly applicable. To show that L is not symmetric, consider:

$$\begin{aligned}
< Lu, v > \quad &= \quad \iint_\Omega (-v^* \nabla^2 u - k_o^2 uv^*) \, d\Omega \\
&= \iint_\Omega (\nabla u \cdot \nabla v^* - k_o^2 uv^*) \, d\Omega - \oint_\Gamma v^* \frac{\partial u}{\partial n} \, d\Gamma \\
&= \iint_\Omega (\nabla u \cdot \nabla v^* - k_o^2 uv^*) \, d\Omega - \int_{C_1} v^*(\underline{A} - jk_z u) \, d\Gamma - \int_{C_2} v^*(-jk_z u) \, d\Gamma \\
&\quad - \int_{C_4} 0 \cdot \frac{\partial u}{\partial n} \, d\Gamma - \int_{C_5} v^* \cdot 0 \, d\Gamma
\end{aligned}$$

and then:

$$\begin{aligned}
< u, Lv > \quad &= \quad < Lv, u >^* \; = \; [\iint_\Omega (\nabla u^* \cdot \nabla v - k_o^2 u^* v) \, d\Omega - \oint_\Gamma u^* \frac{\partial v}{\partial n} \, d\Gamma \,]^* \\
&= \iint_\Omega (\nabla u \cdot \nabla v^* - k_o^2 uv^*) \, d\Omega \\
&\quad + [-\int_{C_1} u^*(\underline{A} - jk_z v) \, d\Gamma - \int_{C_2} u^*(-jk_z v) \, d\Gamma - \int_{C_4} 0 \cdot \frac{\partial v}{\partial n} \, d\Gamma \\
&\quad - \int_{C_5} u^* \cdot 0 \, d\Gamma \,]^* \\
&= \iint_\Omega (\nabla u \cdot \nabla v^* - k_o^2 uv^*) \, d\Omega - \int_{C_1} u(\underline{A}^* + jk_z v^*) \, d\Gamma - \int_{C_2} jk_z uv^* \, d\Gamma
\end{aligned}$$

Thus, is general, $< Lu, v > \neq < u, Lv >$. (Note that the Green's identity

$$\iint_\Omega (\nabla u \cdot \nabla v)\, d\Omega = -\iint_\Omega (v\nabla^2 u)\, d\Omega + \oint_\Gamma v\frac{\partial u}{\partial n}\, d\Gamma$$

has been used in the previous demonstration). However, a functional to be made stationary about the solution of (1.13), using its adjoint [2], is

$$F(\underline{u}, \underline{u}_a) = < L\underline{u}, \underline{u}_a > - < v, \underline{u}_a > - < \underline{u}, v_a > \tag{3.15}$$

where

$$\begin{aligned} L_a\underline{u}_a &= v_a, \text{ in } \Omega \\ B_a\underline{u}_a &= h_a, \text{ on } \Gamma \end{aligned} \tag{3.16}$$

is the adjoint problem of (1.13). The difficulty is that the adjoint problem of (1.13) cannot be found, in general, for any inhomogeneous Dirichlet, Newmann, or Cauchy boundary condition.

In order to show this, let $K = -\nabla^2 - k_o^2$ in Ω, and let $Gf = \frac{\partial f}{\partial n} + Af = B$ be the associated boundary operator on Γ, where $A, B \in C$. This represents a general complex inhomogeneous Cauchy boundary condition. Then:

$$< Kf, g > = \iint_\Omega (\nabla f \cdot \nabla g^* - k_o^2 f g^*)\, d\Omega - \oint_\Gamma g^*(B - Af)\, d\Gamma \tag{3.17}$$

Choosing $K = K_a$ leads to:

$$\begin{aligned} < f, K_a g > &= < K_a g, f >^* \\ &= \iint_\Omega (\nabla f \cdot \nabla g^* - k_o^2 f g^*)\, d\Omega - [\oint_\Gamma f^* \frac{\partial g}{\partial n}\, d\Gamma\,]^* \end{aligned} \tag{3.18}$$

Now, choosing the adjoint boundary condition G_a as:

$$G_a f = \frac{\partial f}{\partial n} + Cf = D$$

[2]The adjoint of L noted L_a is such that $< Lf, g > = < f, L_a g >$. A self-adjoint operator L is such that $L = L_a$ and $D_L = D_{L_a}$. Thus a self-adjoint operator is symmetric.

(another boundary condition of the same type as G) yields:

$$
\begin{aligned}
< Kf, g > - < f, K_a g > &= -\oint_\Gamma g^*(B - Af)\, d\Gamma + [\oint_\Gamma f^*(D - Cg)\, d\Gamma\,]^* \\
&= \oint_\Gamma (fD^* - g^*B)\, d\Gamma + \oint_\Gamma fg^*(A - C^*)\, d\Gamma
\end{aligned}
$$

This is to say that the adjoint problem can be found only if $D = B = 0$ and if $A = C^*$.

A similar analysis reveals that L_a cannot be found for inhomogeneous Dirichlet and Newmann BC.

Now, define

$$\underline{w} = \underline{\phi} - \underline{s} \tag{3.19}$$

where $\underline{s} \in D_L$ is a fixed function and satisfies the BC's (not necessarily the solution). Since L is a linear operator, and since both $\underline{\phi}$ and $\underline{s}$ are in the domain of L, then $\underline{w}$ must also be in the domain of L. Thus, a new equivalent problem with homogeneous BC's can be written as

$$
\begin{aligned}
L\underline{w} &= -L\underline{s}\ , \text{ in } \Omega \\
B\underline{w} &= 0\ , \text{ on } \Gamma
\end{aligned} \tag{3.20}
$$

and its adjoint operator L_a can be found. One can now write the adjoint problem of (1.20) as

$$
\begin{aligned}
L_a\underline{w}^* &= -L_a\underline{s}^*\ , \text{ in } \Omega \\
B_a\underline{w}^* &= 0\ , \text{ on } \Gamma
\end{aligned} \tag{3.21}
$$

where $(\cdot)^*$ denotes the complex conjugate,

$L_a = L$,

$B_a = \partial/\partial n\ - jk_z$, on C_1 and C_2,

$\quad\quad \partial/\partial n$, on C_5,.

$\quad\quad 1$, on C_4,

or, symbolically, $B_a = B^*$.

The domain of L_a (noted D_{L_a}) is again the subset of $\mathcal{L}_2^{(c)}(\Omega)$ which contains all twice differentiable functions. Then $D_L = D_{L_a}$. Inserting (1.20) and (1.21) into (1.15) yields

$$
\begin{aligned}
F(\underline{u}) &= < L\underline{w}, \underline{w}^* > + < \underline{w}, L_a\underline{s}^* > + < L\underline{s}, \underline{s}^* > \\
&= < L\underline{u}, \underline{u}^* > - < L\underline{u}, \underline{s}^* > + < \underline{u}, L_a\underline{s}^* > + f(\underline{s}).
\end{aligned} \tag{3.22}
$$

This functional is stationary about the solution $\underline{u} = \underline{\phi}$. The term $f(\underline{s})$ represents all terms which are function of $\underline{s}$ only, and is therefore a constant. Evaluation of the inner products in (1.22), using Green's theorem, yields

$$
\begin{aligned}
F(\underline{u}) &= \iint_\Omega (\nabla\underline{u} \cdot \nabla\underline{u} - k_o^2\underline{u}^2)\, d\Omega + \int_{C_1 \cup C_2} jk_z\underline{u}^2\, d\Gamma - \int_{C_1} \underline{Au}\, d\Gamma \\
&+ f(\underline{s}) - \oint_\Gamma \left(\underline{s}\frac{\partial \underline{u}}{\partial n} - \underline{u}\frac{\partial \underline{s}}{\partial n}\right) d\Gamma
\end{aligned} \tag{3.23}
$$

One must remember that the integration on the contour Γ must be done counter-clockwise [7]. The last member of the right of (1.23) is 0 everywhere except on C_1. Its value is then

$$
\oint_\Gamma (\underline{s}\frac{\partial \underline{u}}{\partial n} - \underline{u}\frac{\partial \underline{s}}{\partial n})\, d\Gamma = \int_{C_1} (s\underline{A} - \underline{uA})\, d\Gamma = -\int_{C_1} \underline{uA}\, d\Gamma + g(\underline{s}). \tag{3.24}
$$

where $g(\underline{s})$ is a function of $\underline{s}$ only and is a constant. Inserting (1.24) into (1.23) yields the final functional to be made stationary:

$$
F(\underline{u}) = \iint_\Omega (\nabla\underline{u} \cdot \nabla\underline{u} - k_o^2\underline{u}^2)\, d\Omega + \int_{C_1 \cup C_2} jk_z\underline{u}^2\, d\Gamma - 2\int_{C_1} \underline{Au}\, d\Gamma. \tag{3.25}
$$

Note that $f(\underline{s})$ (from 1.23) and $g(\underline{s})$ (from 1.24) can be removed from the functional, since two functionals that differ by only a constant are equivalent [6].

Chapter 4

Solution by FEM.

4.1 introduction

This chapter presents the basic notions necessary to implement the method into a computer program. The emphasis is made on solving the continuous problem, mathematically formulated in terms of differential calculus in the preceding chapter, using discretization of the potential field $\underline{\phi}$. Enough information is included so that a computer program can easily be constructed to solve the actual problem, that of a filter with two metal inserts, using only the material given herein. Such a computer program is listed in the appendix.

4.2 Decomposing the functional.

In order to solve (3.25), using the FEM, one must break the region Ω into a finite number of connected polygons. The most widely used polygon is the triangle [4], because of its geometrical simplicity and the analytical simplicity which it implies. The boundary Γ must also be broken into elements, namely connected segments.

The functional (3.25) can be decomposed into 5 distinctive parts, each of which can be analyzed separately:

$$F(\underline{u}) = F_1 + F_2 + F_3 + F_4 + F_5 \tag{4.1}$$

$$F_1 = \iint_{\Omega} (\nabla\underline{u} \cdot \nabla\underline{u})\, d\Omega$$

$$F_2 = -\iint_{\Omega} k_x^2 \underline{u}^2 \, d\Omega$$

$$F_3 = jk_z \int_{C_1} \underline{u}^2 \, d\Gamma$$

$$F_4 = -2 \int_{C_1} \underline{Au} \, d\Gamma$$

$$F_5 = jk_z \int_{C_2} \underline{u}^2 \, d\Gamma$$

4.3 Boundary linear elements.

We first wish to approximate the field function $\underline{u}$ on a segment of Γ (noted $\underline{u}_\Gamma$) by an N^{th} degree polynomial. We need to sample $\underline{u}_\Gamma$ at $N+1$ different locations, so that the resulting polynomial is uniquely determined. For simplicity, and for compatibility with the algorithm proposed by Silvester [3] used herein, all sampled nodes are equidistant, as in fig. (4.1). The resulting discrete approximation to the function $\underline{u}_\Gamma$ is the column vector $\underline{U}_\Gamma$, described in this section. Since on C_1 and C_2 the Z-coordinate is fixed, then the approximation polynomial should only be a function of X-coordinates. Therefore

$$\underline{u}_\Gamma(x) \cong Dh^T \tag{4.2}$$

where $h = [x^0, \ldots, x^N]$

$D = [D_0, \ldots, D_N]$, D_i being unknown complex coefficients

and superscript T represents matrix transposition.

The function $\underline{u}_\Gamma$ at the discrete node i is

$$\underline{u}_\Gamma(x_i) \cong D\nu_i^T = \underline{U}_{\Gamma i} \tag{4.3}$$

where $i \in \{0, \ldots, N\}$

$\nu_i = [x_i^0, \ldots, x_i^N]$

$\underline{U}_{\Gamma i}$ = discrete potential at node i

and $\underline{U}_\Gamma = [\underline{U}_{\Gamma 1}, \ldots, \underline{U}_{\Gamma(N+1)}]$.

Therefore

$$\underline{U}_\Gamma^T = XD^T \tag{4.4}$$

where

$$X = \begin{bmatrix} \nu_0 \\ \vdots \\ \nu_N \end{bmatrix}.$$

Also, we can write

$$\underline{u}_\Gamma(x) \cong hD^T = \beta \underline{U}_\Gamma^T = \beta X D^T \tag{4.5}$$

where $\beta = [\beta_0(x), \ldots, \beta_N(x)]$. β_i are $\mathrm{N^{th}}$ degree interpolation polynomials (or shape functions) defined as

$$\beta_i(x) = \sum_{j=0}^{N} \beta_{ij}\, x^j. \tag{4.6}$$

with property

$$\beta_j(x_i) = \delta_{ij} = \begin{cases} 1 & , \quad j = i \\ 0 & , \quad j \neq i. \end{cases}$$

δ_{ij} is known as Kronecker's delta. From (4.5) we identify β as

$$\beta = hX^{-1} \tag{4.7}$$

which finally yields

$$\beta_{ij} = X^{-1}_{(i+1)(j+1)} \tag{4.8}$$

where X^{-1}_{ij} is the element in matrix X^{-1} at row i and column j. Thus the distribution of $\underline{u}_\Gamma$ is approximated, on each segment, by a piecewise $\mathrm{N^{th}}$ degree curve-fitting process. One method to evaluate X^{-1} requires the computation of the determinant of X, which is referred to as the Vandermonde determinant [8] and to which some special theorems apply.

In order to adapt this method to Silvester's algorithm [3,4], all nodes must be equidistant on any segment. These segments are made to coincide with the segment of the

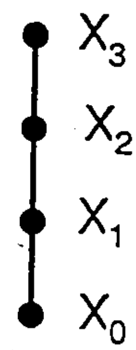


Figure 4.1: Example of discretized segment of Γ for third degree approximation polynomial.

corresponding triangle in Ω that is touching the boundary Γ. The degree of approximation used on the boundary for the β_i shape functions is made the same as the degree of approximation used for the shape functions in Ω. Use of different degree for β_i and for α_i has been found to yield reasonably accurate results, but not quite as good as when both functions are of the same degree. Slight lack of continuity between fields modeled on the surface of a bordering triangle and fields modeled on a boundary element explains this discrepancy. But for sufficiently smooth fields near the boundary, one can use different degrees of approximations on Γ and Ω and, yet, obtain sufficiently continuous fields on the segments of the boundary and on the side of the corresponding triangles in Ω. All simulations reported in graphs herein were made using 3rd degree β_i's and α_i's.

Inserting (4.5) into (4.1) yields

$$\begin{aligned} F_3 &\cong jk_z\underline{U}_\Gamma\left(\int_{C_1} \beta^T\beta \, d\Gamma\right)\underline{U}_\Gamma^T \qquad (4.9) \\ F_4 &\cong -2\int_{C_1} \underline{A}\beta\underline{U}_\Gamma^T \, d\Gamma \\ F_5 &\cong jk_z\underline{U}_\Gamma\left(\int_{C_2} \beta^T\beta \, d\Gamma\right)\underline{U}_\Gamma^T. \end{aligned}$$

It is worth pointing out that (4.9) is only a function of x-coordinates (β) and the unknown function at the discrete nodes ($\underline{U}_\Gamma$). All integrals involved in (4.9) are analytically easy to evaluate. Moreover, the first and third integral need only be evaluated once, as explained later.

The variational parameter being the unknown discrete value of the field at each node, it is required, for stationarity [6], that

$$\frac{\partial F(\underline{U})}{\partial \underline{U}^T} = [0] \tag{4.10}$$

where [0] is a null column vector, and $\underline{U}$ is $\underline{U}_\Omega$ in Ω, and $\underline{U}_\Gamma$ on Γ. References for matrix differentiation, for which some readers may not be familiar with, can be found in [10, pages44-48]. The derivatives of F_3, F_4 and F_5 with respect to $\underline{U}_\Gamma^T$ are

$$\frac{\partial F_3}{\partial \underline{U}_\Gamma^T} = 2jk_z\left(\int_{C_1} (\beta^T\beta)\, d\Gamma\right)\underline{U}_\Gamma^T \tag{4.11}$$

$$\frac{\partial F_4}{\partial \underline{U}_\Gamma^T} = -2\int_{C_1} \underline{A}\beta^T\, d\Gamma$$

$$\frac{\partial F_5}{\partial \underline{U}_\Gamma^T} = 2jk_z\left(\int_{C_2} (\beta^T\beta)\, d\Gamma\right)\underline{U}_\Gamma^T.$$

This is valid for every segment on C_1 and C_2.

Both integrals involving $(\beta^T\beta)$ can be quickly evaluated, keeping in mind that all nodes on all segments of C_1 and C_2 are equidistant. Integrating a single term (row i, column j) of the preceding matrix over a segment yields, using (4.8),

$$\int_{x_0}^{x_N} (\beta^T\beta)_{ij}\, dx = \sum_{n=1}^{N+1}\sum_{m=1}^{N+1} X_{ni}^{-1} X_{mj}^{-1} \frac{x_N^{n+m-1} - x_0^{n+m-1}}{n+m-1} \tag{4.12}$$

It can be shown that the integration of the complete matrix system yields

$$\int_{x_0}^{x_N} (\beta^T\beta)\, d\Gamma = B_N$$

x_N is the last node on the segment, while x_0 is the first one (following the direction of integration). Remember that an integration on C_1 or C_2 is only part of the integration over the contour Γ, *which must be performed in a counter-clockwise direction.* Integration on C_1 is thus performed from highest value of x-coordinate to lowest value (from top to bottom, in our case), and conversely for C_2. Specifically, for a segment on C_1, x_0 is the x-coordinate of the segment of the segment having the highest value. Matrix B_N is a square symmetrical matrix of size $N+1$. Further, all of its diagonals are symmetrical. The following gives these B matrices for N ranging from 1 to 4.

$$B_1 = \frac{(x_N - x_0)}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$B_2 = \frac{(x_N - x_0)}{30}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}$$

$$B_3 = \frac{(x_N - x_0)}{1680}\begin{bmatrix} 128 & 99 & -36 & 19 \\ 99 & 648 & -81 & -36 \\ -36 & -81 & 648 & 99 \\ 19 & -36 & 99 & 128 \end{bmatrix}$$

$$B_4 = \frac{(x_N - x_0)}{2835}\begin{bmatrix} 146 & 148 & -87 & 28 & -14 \\ 148 & 896 & -192 & 128 & 28 \\ -87 & -192 & 936 & -192 & -87 \\ 28 & 128 & -192 & 896 & 148 \\ -14 & 28 & -87 & 148 & 146 \end{bmatrix}$$

Thus, only the coordinates of the extremities of each segment are used to compute the contributions of F_3 and F_5. Note that [4] describes an even more general procedure for computing the B matrices using the so-called simplex coordinate system.

We can now describe each sub-region i on Γ by a system

$$A_i \underline{U}_{\Gamma i} = B_i \qquad (4.13)$$

using (4.10) and (4.11). On C_1, (4.10) becomes (4.13),

where $A_i = 2jk_z(\int_{C_1}(\beta^T\beta)\, d\Gamma)$

$\underline{U}_{\Gamma i} = \underline{U}_{\Gamma T}$

$B_i = 2\int_{C_1} \underline{A}\beta^T\, d\Gamma$

On C_2, (4.10) becomes (4.13),

where $A_i = 2jk_z(\int_{C_2}(\beta^T\beta)\,d\Gamma)$

$\underline{U}_{\Gamma i} = \underline{U}_{\Gamma T}$

$B_i = [0]$

These sub-systems will be assembled into the global system of equations, as described in the last section of this chapter. Note that all A_i matrices are symmetric.

4.4 Internal triangular elements.

We now turn to the modeling of the interior region Ω. Let us refer to fig. 4.2 for a typical example of a triangular division in Ω. The nodes are numbered in the order indicated by [4]. The example shown is for a 3^{rd} degree approximation polynomial of the function $\underline{u}$ in the region of the triangle. Note that there are M nodes per triangle (see below), and $N+1$ nodes on every side of a triangle. All nodes are equally spaced, as in the 1-D case. For the general 2-D case, the approximation of $\underline{u}$ is given by

$$\underline{u}_\Omega \cong \alpha \underline{U}_\Omega^T \tag{4.14}$$

where $\underline{u}_\Omega$ = function $\underline{u}$ inside Ω

$\underline{U}_\Omega$ = discrete version of $\underline{u}_\Omega$

$= [\underline{U}_{\Omega 1}, \ldots, \underline{U}_{\Omega M}]$

$\underline{U}_{\Omega i}$ = potential at node i

$\alpha \;= [\alpha_0(x,z), \ldots, \alpha_M(x,z)]$

$M = (N+1)(N+2)/2$, in general

and $\alpha_i(x,z)$ are the 2-D shape functions.

The shape functions α_i have the same property as their 1-D counterparts, i.e.:

$$\alpha_i(x_j, z_j) = \delta_{ij}$$

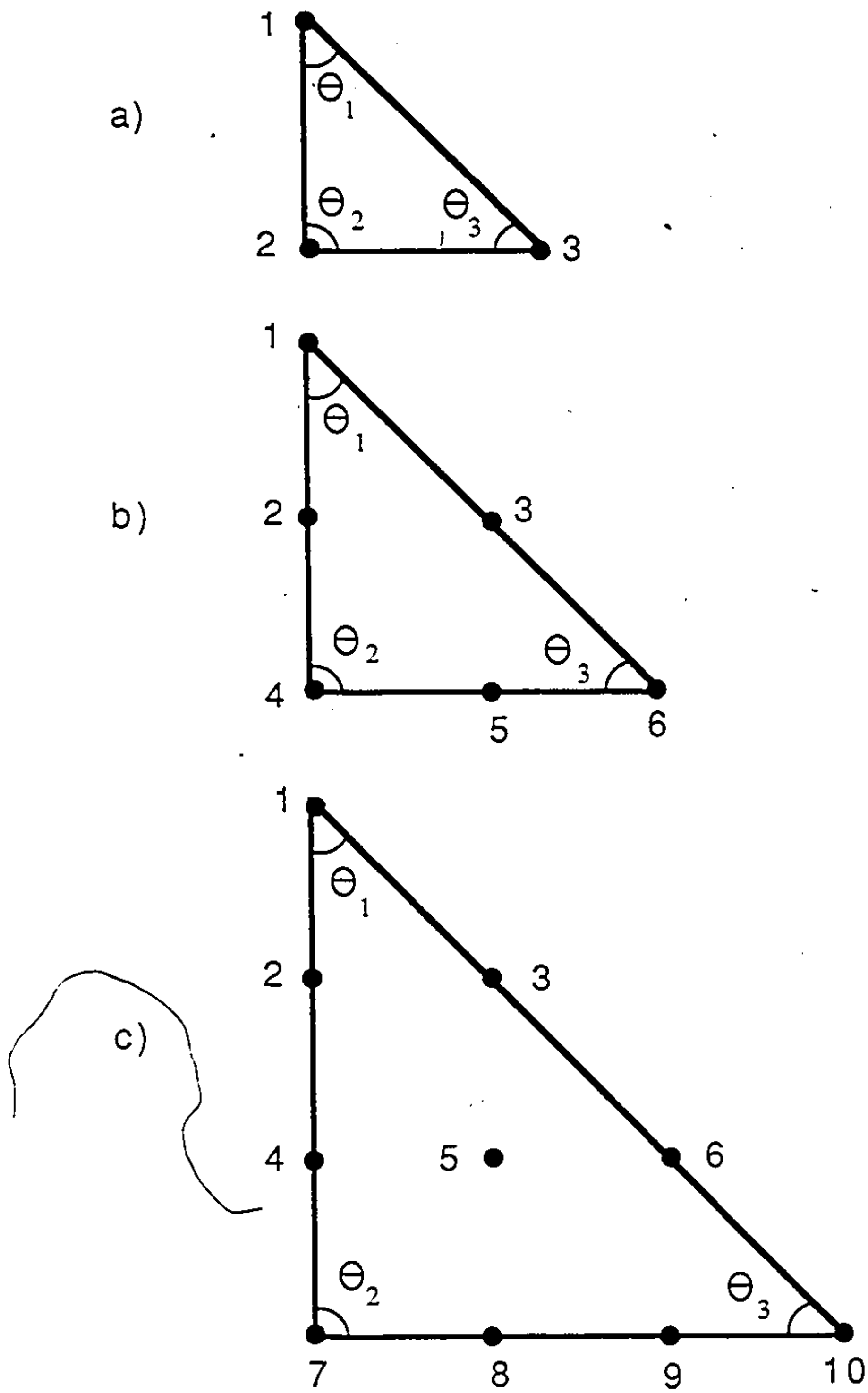


Figure 4.2: Example of triangular element in region Ω, for a) first b)second c)third degree polynomial approximation.

The shape functions are, in general, polynomials in J variables and degree N, where J is the number of dimensions of the region to be modeled (in our case, $J = 2$). Examples for $\alpha_i(x, z)$ are

$$\alpha_i(x,z) = \alpha_{i1} + \alpha_{i2}x + \alpha_{i3}z$$

for $N = 1$,

$$\alpha_i(x,z) = \alpha_{i1} + \alpha_{i2}x + \alpha_{i3}z + \alpha_{i4}x^2 + \alpha_{i5}xz + \alpha_{i6}z^2$$

for $N = 2$, and

$$\begin{aligned} \alpha_i(x,z) = \; & \alpha_{i1} + \alpha_{i2}x + \alpha_{i3}z + \alpha_{i4}x^2 + \alpha_{i5}xz + \alpha_{i6}z^2 \\ & \alpha_{i7}x^3 + \alpha_{i8}x^2z + \alpha_{i9}xz^2 + \alpha_{i10}z^3 \end{aligned} \tag{4.15}$$

for $N = 3$. The terms F_1 and F_2 of (4.1) are approximated by

$$\begin{aligned} F_1 &= \underline{U}_\Omega\left(\iint_\Omega (\nabla\alpha^T \cdot \nabla\alpha)\, d\Omega\right)\underline{U}_\Omega^T \\ F_2 &= -k_o^2\underline{U}_\Omega\left(\iint_\Omega \alpha^T\alpha\, d\Omega\right)\underline{U}_\Omega^T \end{aligned} \tag{4.16}$$

where $\underline{U}_\Omega = [\underline{U}_{\Omega 1}, \ldots, \underline{U}_{\Omega M}]$

$\underline{U}_{\Omega i}$ = discrete potential at node i.

Again, for stationarity, it is required that

$$\frac{\partial F(\underline{U}_\Omega)}{\partial \underline{U}_\Omega^T} = [0]$$

The derivatives of F_1 and F_2 with respect to $\underline{U}_\Omega^T$ are

$$\begin{aligned} \frac{\partial F_1}{\partial \underline{U}_\Omega^T} &= 2\iint_\Omega (\nabla\alpha^T \cdot \nabla\alpha)\, d\Omega\, \underline{U}_\Omega^T = 2S\underline{U}_\Omega^T \\ \frac{\partial F_2}{\partial \underline{U}_\Omega^T} &= -2k_o^2\iint_\Omega (\alpha^T\alpha)\, d\Omega\, \underline{U}_\Omega^T = -2k_o^2T\underline{U}_\Omega^T \end{aligned} \tag{4.17}$$

Actual computation of T and S becomes rather lengthy and involved for $N > 1$. Rather than attempting to compute these matrices, the author used results published in [4]. A practical table is given in the above reference which uses a different procedure than the one presented herein to compute the S and T matrices, for any N. These matrices are given below, for $N = 1$ (which is very often used with the FEM) and $N = 3$ (presently used).

For $N = 1$:

$$T = \frac{\Delta}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$S = \sum_{i=1}^{3} Q_i \cot(\theta_i)$$

where

$$Q_1 = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$Q_2 = \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Q_3 = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and θ_i is the angle at vertex i, as seen in figure 4.2. Δ is the area of the triangle, and can be evaluated as:

$$\Delta = \text{absolute value of } \frac{1}{2}\ det \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{bmatrix}$$

where x_i and z_i are the coordinates of vertex i, corresponding to angles θ_i.

For $N = 3$, the T and S matrices are as follows.

$$T = \frac{\Delta}{6720}\begin{bmatrix} 76 & 18 & 18 & 0 & 36 & 0 & 11 & 27 & 27 & 11 \\ 18 & 540 & 270 & -189 & 162 & -135 & 0 & -135 & -54 & 27 \\ 18 & 270 & 540 & -135 & 162 & -189 & 27 & -54 & -135 & 0 \\ 0 & -189 & -135 & 540 & 162 & -54 & 18 & 270 & -135 & 27 \\ 36 & 162 & 162 & 162 & 1944 & 162 & 36 & 162 & 162 & 36 \\ 0 & -135 & -189 & -54 & 162 & 540 & 27 & -135 & 270 & 18 \\ 11 & 0 & 27 & 18 & 36 & 27 & 76 & 18 & 0 & 11 \\ 27 & -135 & -54 & 270 & 162 & -135 & 18 & 540 & -189 & 0 \\ 27 & -54 & -135 & -135 & 162 & 270 & 0 & -189 & 540 & 18 \\ 11 & 27 & 0 & 27 & 36 & 18 & 11 & 0 & 18 & 76 \end{bmatrix}$$

$$S = \sum_{i=1}^{3} Q_i \cot(\theta_i)$$

where

$$Q_1 = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 135 & -135 & -27 & 0 & 27 & 3 & 0 & 0 & -3 \\ 0 & -135 & 135 & 27 & 0 & -27 & -3 & 0 & 0 & 3 \\ 0 & -27 & 27 & 135 & -162 & 27 & 3 & 0 & 0 & -3 \\ 0 & 0 & 0 & -162 & 324 & -162 & 0 & 0 & 0 & 0 \\ 0 & 27 & -27 & 27 & -162 & 135 & -3 & 0 & 0 & 3 \\ 0 & 3 & -3 & 3 & 0 & -3 & 34 & -54 & 27 & -7 \\ 0 & 0 & 0 & 0 & 0 & 0 & -54 & 135 & -108 & 27 \\ 0 & 0 & 0 & 0 & 0 & 0 & 27 & -108 & 135 & -54 \\ 0 & -3 & 3 & -3 & 0 & 3 & -7 & 27 & -54 & 34 \end{bmatrix}$$

$$Q_2 = \frac{1}{2}\begin{bmatrix} 34 & 3 & -54 & 3 & 0 & 27 & 0 & -3 & -3 & -7 \\ 3 & 135 & 0 & -27 & -162 & 0 & 0 & 27 & 27 & -3 \\ -54 & 0 & 135 & 0 & 0 & -108 & 0 & 0 & 0 & 27 \\ 3 & -27 & 0 & 135 & 0 & 0 & 0 & -135 & 27 & -3 \\ 0 & -162 & 0 & 0 & 324 & 0 & 0 & 0 & -162 & 0 \\ 27 & 0 & -108 & 0 & 0 & 135 & 0 & 0 & 0 & -54 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 27 & 0 & -135 & 0 & 0 & 0 & 135 & -27 & 3 \\ -3 & 27 & 0 & 27 & -162 & 0 & 0 & -27 & 135 & 3 \\ -7 & -3 & 27 & -3 & 0 & -54 & 0 & 3 & 3 & 34 \end{bmatrix}$$

$$Q_3 = \frac{1}{2}\begin{bmatrix} 34 & -54 & 3 & 27 & 0 & 3 & -7 & -3 & -3 & 0 \\ -54 & 135 & 0 & -108 & 0 & 0 & 27 & 0 & 0 & 0 \\ 3 & 0 & 135 & 0 & -162 & -27 & -3 & 27 & 27 & 0 \\ 27 & -108 & 0 & 135 & 0 & 0 & -54 & 0 & 0 & 0 \\ 0 & 0 & -162 & 0 & 324 & 0 & 0 & -162 & 0 & 0 \\ 3 & 0 & -27 & 0 & 0 & 135 & -3 & 27 & -135 & 0 \\ -7 & 27 & -3 & -54 & 0 & -3 & 34 & 3 & 3 & 0 \\ -3 & 0 & 27 & 0 & -162 & 27 & 3 & 135 & -27 & 0 \\ -3 & 0 & 27 & 0 & 0 & -135 & 3 & -27 & 135 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The matrix T of (4.17) is, similarly to matrix B in the previous section, a simple function of the coordinates of the 3 vertices of the triangle multiplied by a constant matrix.

Figure 4.3 shows the mesh modeling the filter, divided into triangular regions, as used in the FEM program included in the appendix.

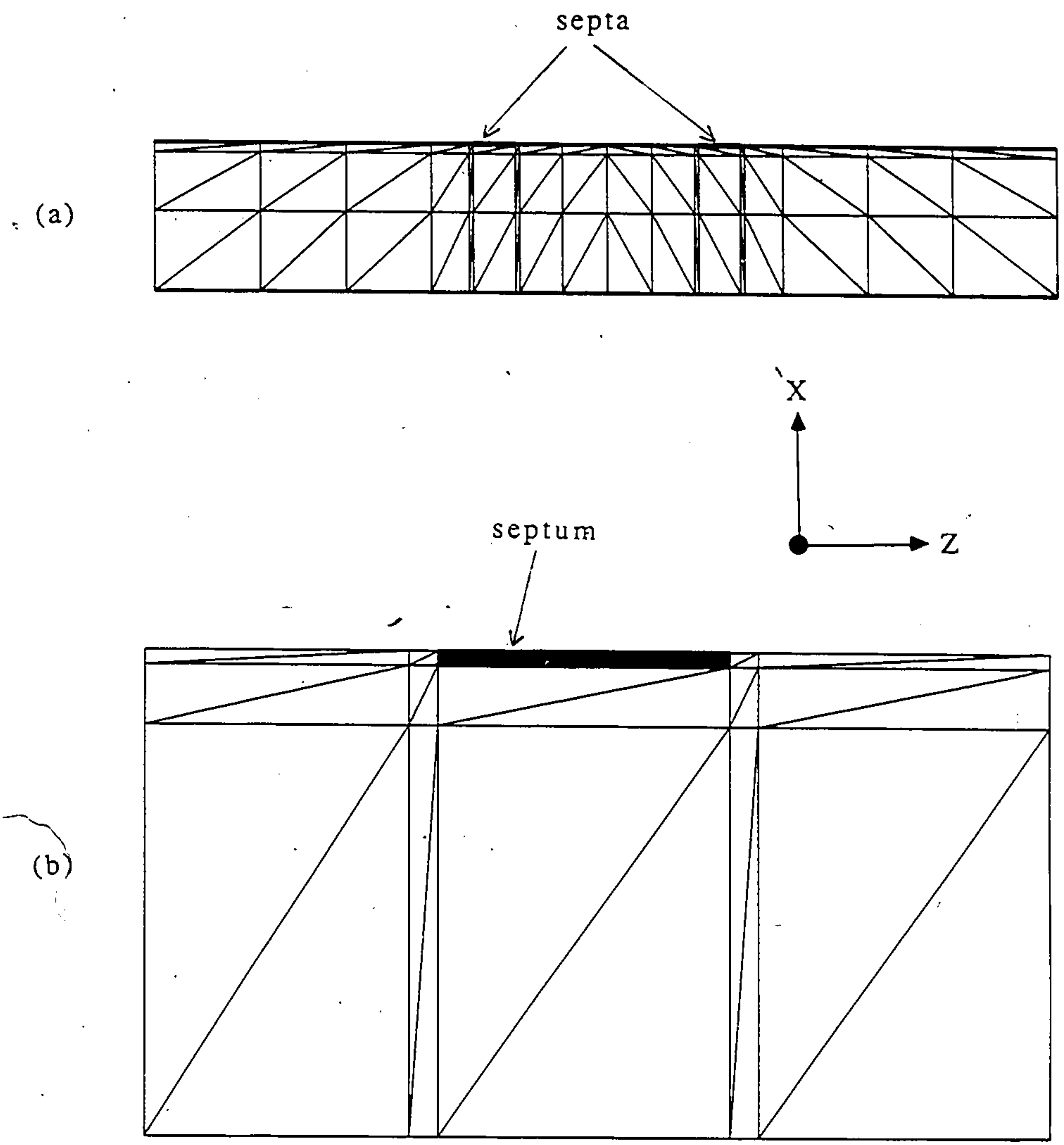


Figure 4.3: (a)Model of filter used by FEM program in appendix, with (b) close-up view near septum.

4.5 Final system of equations.

Assembling the final system is quite systematic. A mathematical way of connecting all elements together is described in [4]. The author presents here an algorithm, rather than a mathematical formula, to connect all elements in a simple and easily understandable way.

Each triangular region Ω_i has M nodes. A system $A_i\underline{U}_{\Omega i} = B_i$ has been described for each of these sub-regions. Note that a node on the side of a triangle is usually shared with an adjacent triangle. Hence, there are much less than $M \times K$ free nodes in the structure, where K is the total number of triangles.

The following describes the procedure used to reduce a system of size M to one of size $M-1$, when one of the nodes in a sub-region is of Dirichlet type (the node no longer corresponds to an unknown of the system of equations). An example fully illustrates the complete procedure. Let $M = 3$. Then the sub-system $A_i\underline{U} = B_i$ can be written as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Now let node 2 be of Dirichlet type: $u_2 = c_2 = constant$ (Note that all Dirichlet nodes in the model presently studied are forced to a null value. See equation (3.9)). The system is now reduced to:

$$\begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} = \begin{bmatrix} b_1 - a_{12}c_2 \\ b_3 - a_{32}c_2 \end{bmatrix}$$

Note that the square matrix is still symmetric. The process may be repeated if there are any other Dirichlet nodes in this sub-region. In the limit, if all nodes are of Dirichlet type in any sub-region, then the resulting system of equations will be of size 0. Although this sounds trivial, it is nevertheless useful in a computer algorithm, where the automatic (or manual) meshing algorithm may subdivide a region that is completely inside a conductor. Since a conductor will force the electric field to zero, then all nodes inside a conductor are of Dirichlet type (at least for the problem studied here), and none of these nodes contribute to the final system of equations.

The final system is assembled sequentially. Start by a system $A\underline{U} = B$, where A, $\underline{U}$ and B contain all zeros. The size of this system is equal to the number of free nodes in the modeled structure, including all nodes in Ω and on Γ. B and $\underline{U}$ are column vectors, while A is a square matrix. Now, all A_i, $\underline{U}_{\Omega i}$ and B_i are sequentially added to the system $A\underline{U} = B$ as follows. Let node j of the sub-region Ω_i correspond to node k of the global structure. Then:

$$A(k,m) \leftarrow A(k,m) + A_i(j,m), \forall \text{ possible } m$$
$$B(k) \leftarrow B(k) + B_i(j)$$
$$\underline{U}(k) \leftarrow \underline{U}(k) + \underline{U}_{\Omega i}(j)$$

where $M(k,m)$ = element at row k, column m of matrix M

$V(k)$ = k^{th} element of column vector V.

Similarly, the same procedure is applied for all nodes j in all sub-regions Γ_i (on the boundary) corresponding to node k of the global structure. And note that since all A_i's are symmetric matrices, then A is also a symmetric matrix. This is a useful observation since some software packages are available to solve such systems, using time-saving and core-saving algorithms.

The final matrix system

$$A\underline{U} = B \tag{4.18}$$

to be solved can be visualized as follows. Matrix A contains the geometrical properties of the system, only. That is to say, it describes the natural response of the system. Vector B contains all forcing functions, that is the terms resulting from F_4 (input wave) and from the Dirichlet nodes (forced boundaries). Note that A is unique for a given geometry, regardless of all forcing functions.

Chapter 5

Comparison with the TLM method.

5.1 Introduction.

This section is introduced as an attempt to compare the results obtained from the FEM with those obtained from another method since no such information is available anywhere in the literature. The TLM method is used more to confirm the accuracy of results from FEM than to try to substitute it, since the weak spatial resolution problems of this method (in its basic form) make it a less ideal candidate than the FEM for filters with thin inserts.

5.2 Introduction to the TLM method.

An excellent overview of the TLM method is given in [11]. Several papers are available to the interested reader, including the first papers introducing the TLM method by Johns and Beurle [12]-[14]. The TLM method simulates the propagation of waves in space in the time domain. It is therefore an extremely general approach for solving Maxwell's equations.

The basic idea of the 2-D TLM method is that 2-D space is modeled by a mesh of intersecting open 2-wire transmission lines parallel to Cartesian coordinates. The distance between consecutive nodes along a Cartesian axis is Δd. Voltage and current equations

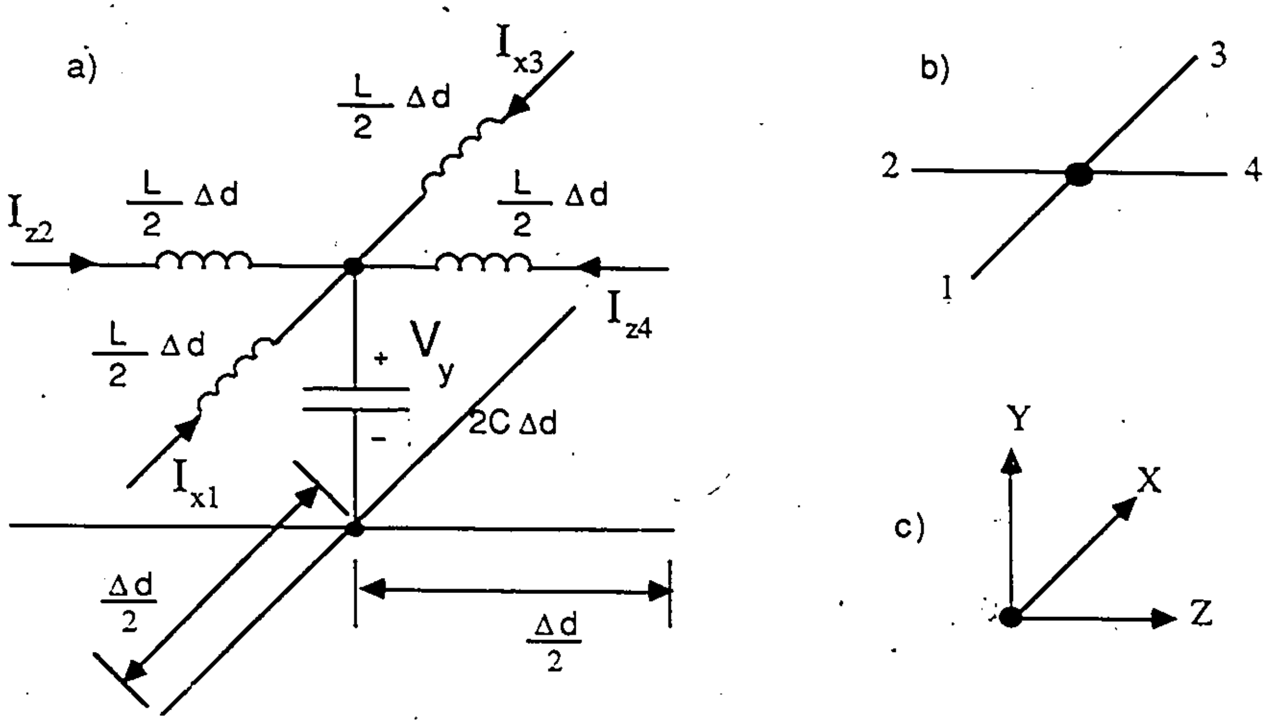


Figure 5.1: Schematic equivalent of a TLM shunt node: a) diagram. b)standard directions. c) axis.

on the lines will be shown to be of the same form as Maxwell's equations. Thus, by identifying terms properly, one can get a one-to-one equivalence between lumped-circuit parameters of the mesh and electromagnetic quantities in space. Voltage impulse functions are launched in the mesh and scatter according to the same principle of wave dispersion described by Huygens. Thus, waves are accurately modeled by impulses travelling through the mesh of transmission lines since they behave as a wave-front in space [14].

5.2.1 Full-wave analysis, and TLM network equations.

Figure 5.1 shows the equivalent lumped-element equivalence of the shunt interconnection of two open transmission lines used in the TLM grid. Letting the constitutive parameters of the lines be that of free space ($\epsilon_r = \mu_r = 1$), the velocity of propagation in the lines becomes $1/\sqrt{LC} = 1/\sqrt{\epsilon_o\mu_o} = c$. These 2-conductor lines support a TEM mode. The voltages and currents are thus uniquely defined, and *for infinitesimal spatial increments* are given by

$$\frac{\partial V_y}{\partial z} = -L\frac{\partial I_z}{\partial t} \tag{5.1}$$
$$\frac{\partial V_y}{\partial x} = -L\frac{\partial I_x}{\partial t}$$
$$-\frac{\partial I_z}{\partial z} - \frac{\partial I_x}{\partial x} = 2C\frac{\partial V_y}{\partial t}$$

where $I_x = I_{x1} - I_{x3}$ and $I_z = I_{z2} - I_{z4}$. A full-wave analysis of the fields inside the filter yields, directly from both Maxwell's curl equations:

$$\frac{\partial E_y}{\partial z} = \mu\frac{\partial H_x}{\partial t} \tag{5.2}$$
$$\frac{\partial E_y}{\partial x} = -\mu\frac{\partial H_z}{\partial t}$$
$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \epsilon\frac{\partial E_y}{\partial t}$$

Thus, there is a one-to-one relationship between all relevant quantities in the TLM grid analysis and the full-wave analysis, as follows:

$$\mu \equiv L \tag{5.3}$$
$$\epsilon \equiv 2C$$
$$E_y \equiv V_y$$
$$H_z \equiv I_x$$
$$H_x \equiv -I_z$$

We see that $\epsilon = 2C$, implying that the structure simulated by the TLM network is actually a structure having $\epsilon_r = 2$. Therefore, the dimensions must be reduced by a factor of $\sqrt{2}$, thus simulating a real structure having $\epsilon_r = 2$. Equivalently, one can keep the actual dimensions and reduce the operating frequency by a factor of $\sqrt{2}$.

From [12] and [14], the following relations describe the simulation of a wave scattering in a TLM mesh:

$$\begin{bmatrix} V_1(z,x) \\ V_2(z,x) \\ V_3(z,x) \\ V_4(z,x) \end{bmatrix}_{k+1}^{r} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} V_3(z,x) \\ V_4(z,x) \\ V_1(z,x) \\ V_2(z,x) \end{bmatrix}_{k}^{i} \tag{5.4}$$

$$\begin{bmatrix} V_1(z,x) \\ V_2(z,x) \\ V_3(z,x) \\ V_4(z,x) \end{bmatrix}_{k+1}^{i} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1(z,x+1) \\ V_2(z+1,x) \\ V_3(z,x-1) \\ V_4(z-1,x) \end{bmatrix}_{k+1}^{r}$$

where superscript i and r respectively refer to incident and reflected voltages on each line, (z,x) refers to the coordinate of a particular node, V_i is the voltage (either incident or reflected) at branch i (see Fig. 5.1) of a node, and the subscript refers to the iteration number. Thus it is only necessary to store the incident voltages for each node and each of their 4 branches at every iteration. A practical FORTRAN implementation requires two 3-dimensional arrays of size 4 by X by Z to store all incident voltages in all 4 directions, for given iterations k and $k+1$. Note that the TLM method is completely iterative, and requires no matrix inversion, making it very stable. Increasing the number of iterations simply models the wave evolving in time.

It is important to note that we are simulating *voltage* impulses, so that currents are obtained by dividing the voltages by the impedance of the lines, which is $Z = \sqrt{L/C} = Z_o$, the intrinsic impedance of vacuum. Another note of importance is that the current thus obtained is equivalent to the H-field in the corresponding structure having constitutive parameter $\epsilon_r = 2$. Thus, the true H-field value is obtained by dividing the currents in the TLM mesh by $\sqrt{2}$.

5.2.2 Boundary models.

Boundaries in the TLM network are obtained by examining the equivalences (5.3). The boundary condition at a metal wall is given by (3.9) as $E_y = 0$. From (5.3) this translates into$V_y = 0$ for the TLM network. In other words, the line must be short-circuited. Alternatively, a magnetic wall becomes an open-circuit, in the TLM model. Note that all such discontinuities must be placed midway between two adjacent nodes, in order to conserve synchronism [12]. For an absorbing wall, consider a TE_{10} wave traveling in the positive z-direction. The wave impedance is found to be [33]:

$$\frac{E_y}{-H_x} = Z_o \frac{\lambda_g}{\lambda} = \frac{\omega \mu_o}{\beta_g} \tag{5.5}$$

where Z_o = free-space intrinsic impedance ($\cong 377\Omega$)

λ_g = guided wavelength

λ = free-space wavelength

β_g = propagation factor in the guide.

Remembering the $\sqrt{2}$ scaling factor relating the real and TLM structures, the terminating resistor for the TLM network is, from the equivalence (5.3),

$$R = \frac{V_y}{I_z} = \frac{\omega \mu_o}{\beta_g \sqrt{2}} \tag{5.6}$$

where ω is the actual frequency. Note that although the wave is completely absorbed by the wall, there will still be local reflections, due to the fact that the terminating resistor R does not match the impedance of the lines given by $Z = Z_o$. Also, one must keep in mind that this analysis is only valid if only the fundamental mode of propagation exists at the boundary. This implies that all discontinuities must be located at a reasonable distance (a minimum of one wavelength) from the absorbing walls, where all higher order modes vanish.

Thus, magnetic wall C_5 is modeled by open-circuited lines, electric wall C_4 as well as all sides of each septum are modeled by shorted lines, and absorbing walls C_1 and C_2 are modeled by terminating resistors of value R (5.6). Note that R must be calculated for every simulation at a different frequency.

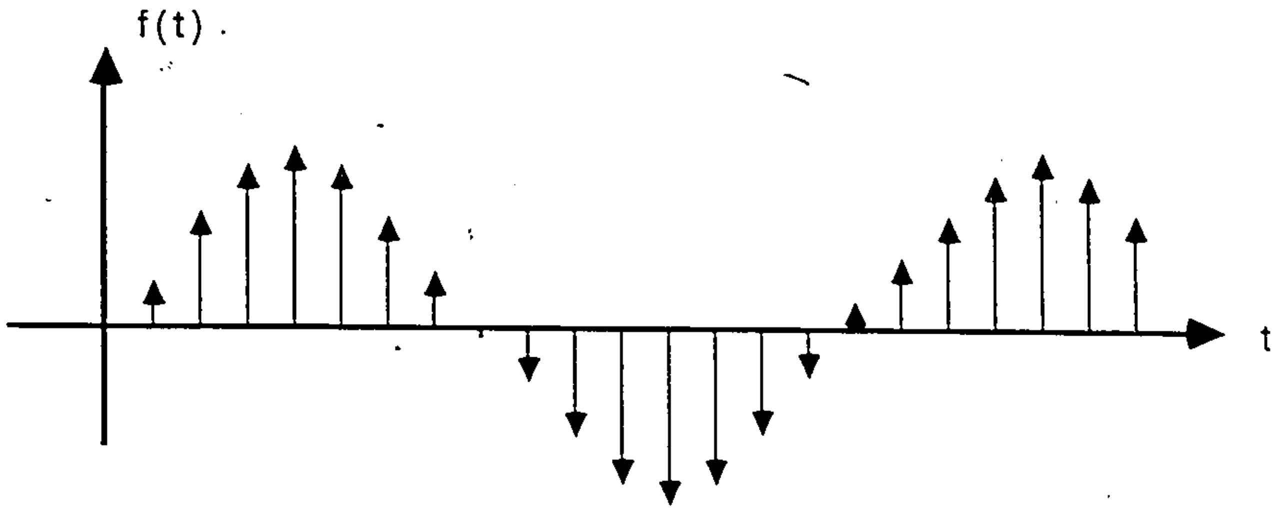


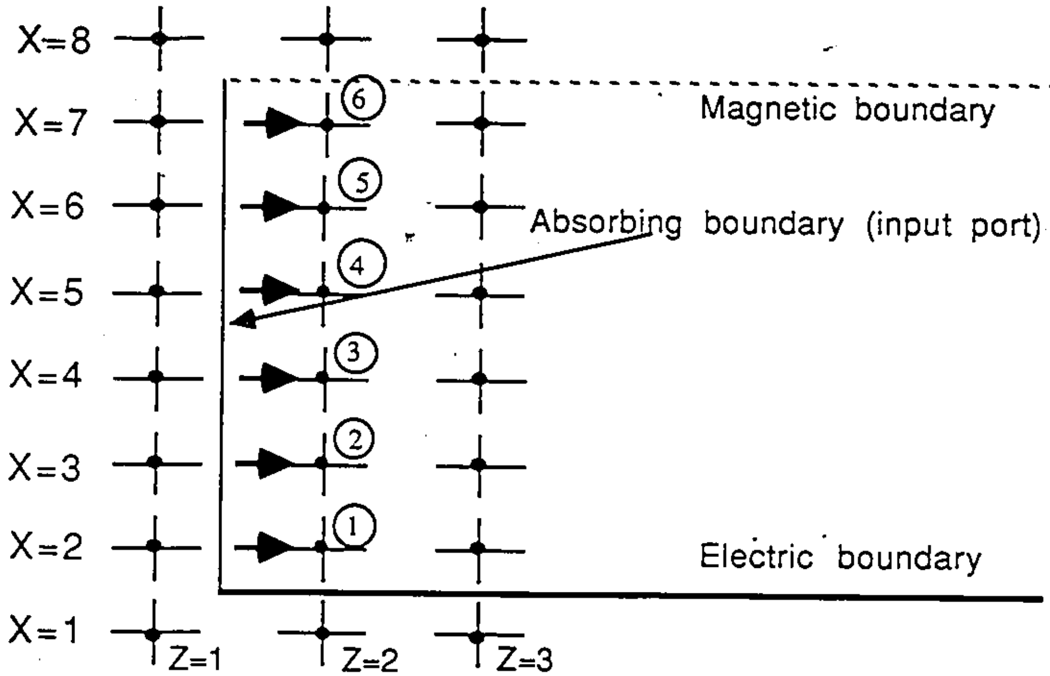
Figure 5.2: Simulation of sinusoidal wave with impulses, as used in TLM method.

5.3 Simulation of continuous waves with TLM.

Since the TLM algorithm simulates scattering of an impulse through discretized space, a sinusoidal signal is modeled as a series of weighted impulses, as shown in figure 5.2. Each impulse travels at the speed of light through the lines, as is the case with all TEM air-filled lines. The TLM method is an iterative algorithm, where each iteration represents a step Δt in time, making one iteration correspond to the time taken for an impulse to reach a neighboring node. Thus, $\Delta t = \Delta d/c$, where c is the speed of light. Since all nodes are equidistant, all impulses are synchronized together.

One common use of the TLM method is to find an approximation of the frequency response of a structure. To do so, one has to inject a single unit impulse at an arbitrary input node of the mesh and let it propagate. The propagation from one node to another is discrete and represents one iteration in the TLM method. One can then compute the discrete Fourier transform of the impulse stream sampled at an arbitrary node in the mesh to obtain the frequency spectrum of the TLM model, which approximates the real structure. This method yields very accurate results for finding cutoff frequencies of arbitrary cross-section waveguides.

Figure 5.3: Excitation of waveguide with continuous wave, for TLM.



A modification on the above, used to compute the field distribution inside the insert filters, is to inject a continuous set of impulses at the input boundary of the filter, representing a discrete sinusoidal excitation, as shown in Fig. 5.2. All nodes at the input boundary C_1 are excited as such, to correctly represent an incident TE_{10}-mode wave. The impulses must be weighted according to the fundamental mode E-field distribution across the cross-section of the guide. The following example illustrates this procedure.

Figure (5.3) represents the TLM model of a half-waveguide structure similar to the one used for the metal insert filter. The waveguide half-width ($a/2$) is $6\Delta d$. The nodes numbered 1 to 6 receive the incident excitation at their branch number 2 (see Fig. 5.1) at every iteration. The incident TE_{10} mode has field distribution

$$E_y(x,z,t) = E_o \sin(\frac{\pi x}{a}) \sin(\omega t)$$

where $z = 0$ on nodes 1 to 6, without loss of generality. Since the TLM algorithm works in discrete time and space, the previous expression becomes

$$E_y(x,z,t) = E_o \sin(\pi \frac{(2N-1)}{24}) \sin(\omega k \Delta t)$$

where k = iteration number

N = node number, on input boundary (as shown in Fig. 5.3).

E_o is the input wave amplitude, $\sin(\omega k \Delta t)$ is the term representing the time variation, as shown in figure 5.2, and $\sin(\pi \frac{(2N-1)}{24})$ is the weight factor added to each node to effectively simulate a TE_{10} wave at the input of the waveguide.

Choosing a proper time step Δt is discussed in a later section. It is not done arbitrarily, since it is directly tied to the choice of node spacing Δd.

5.4 Sources of errors.

Several sources of errors have been reported by authors [11]. These all play a role in limiting the accuracy of the TLM method. These will all be discussed, with emphasis on the most relevant error sources, for the present case, which are the velocity error and the coarseness error.

The author has first examined another possible source of error: truncation of real numbers due to limited computer resolution. The analysis being too analytically complex to perform, the author simply compared outputs from 2 different versions of the final program, one using simple precision arithmetic (4-byte real numbers), the other using double precision arithmetic. Since no differences were observed down to 4 digits of accuracy, this source of error was deemed insignificant. This ultimately helped to build a faster program with lower memory requirements, as simple precision arithmetic is slightly faster and uses half the memory as double precision. This also implies that the TLM algorithm is extremely stable. It does not suffer from ill-conditioning of systems of linear equations encountered in most other methods, including the FEM.

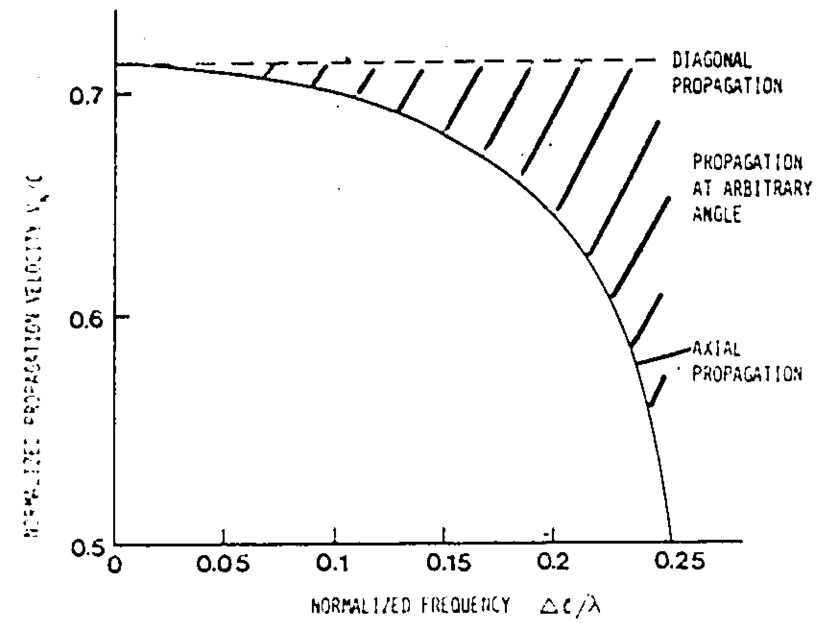


Figure 5.4: Dispersion of the velocity of waves in 2-D TLM network (after Hoefer [11]).

One type of error, the truncation of impulse response error [11], does not come into consideration as we do not Fourier-transform the time-response of the system, as will be discussed later.

The velocity error was first reported in [12]. This is due to the fact that the propagation constant in the mesh depends on the direction of propagation relative to the axis of orientation of the branches. The propagation velocity in a 45 degree direction is the same for all frequencies (no dispersion), while for a direction parallel to one of the main axes it is a function of frequency. Figure 5.4 shows this dispersion characteristic. In the limit, as the parameter $\Delta d/\lambda$ tends to zero, the propagation velocity tends to the same value as in the 45 degree case. Hence, as a rule of thumb, the parameter $\Delta d/\lambda$ must be kept smaller than 0.05. Using the center frequency of the filter (32.5 GHz) with the minimum Δd possible, namely the half-thickness of a septum (50 μm), yields $\Delta d/\lambda = 0.0054$. This is clearly much lower than the highest allowable value, hence no practical velocity error is expected.

Coarseness error is due to a lack of spatial resolution in regions where the modeled field is highly nonuniform, as near edges. The reported cures include refining the mesh considerably (which in our case would require a prohibitive amount of memory and time), or using a variable mesh size [15], which would involve programming implementation difficulty. Another method is to compute results using different coarseness, and then extrapolate for $\Delta d \rightarrow 0$ [11]. Again, the results obtained are extremely close to those obtained via FEM and mode matching. Coarseness error is therefore neglected.

Hence, very little errors are expected by using the TLM method. Of course, lack of spatial resolution of the current is expected on the front (narrow) face of each septum.

5.5 Implementation of the TLM method.

The TLM grid used for the present problem is a regularly spaced grid. Thus the grid density throughout the structure depends directly on the spatial resolution of the smallest object that we wish to model. This, of course, implies terrific computer storage waste in the regions where not much resolution is needed. The same region is modeled with the TLM as with the FEM method. That is only one half of the structure is modeled (Fig. 5.5).

The smallest resolution needed in this case is the septa half-thickness, $50 \mu m$. One must remember that all dimensions must be an integer multiple of this minimum spatial resolution Δd which is the smallest distance between consecutive nodes. This implies that the structure can no longer be exactly modeled, as with FEM. For instance, the waveguide half-width cannot be 3.556 mm, but 3.55 mm (the closest integer multiple of $50 \mu m$). Since the septa are "reasonably" far from the walls (most of the disturbance happens in the middle of the guide), we expect that this small change in dimensions does not affect the final field distribution. This will be confirmed later by the results.

The TLM method will directly yield the time-domain field distribution. In order to obtain the amplitude of the field distribution, one must observe all nodes of interest

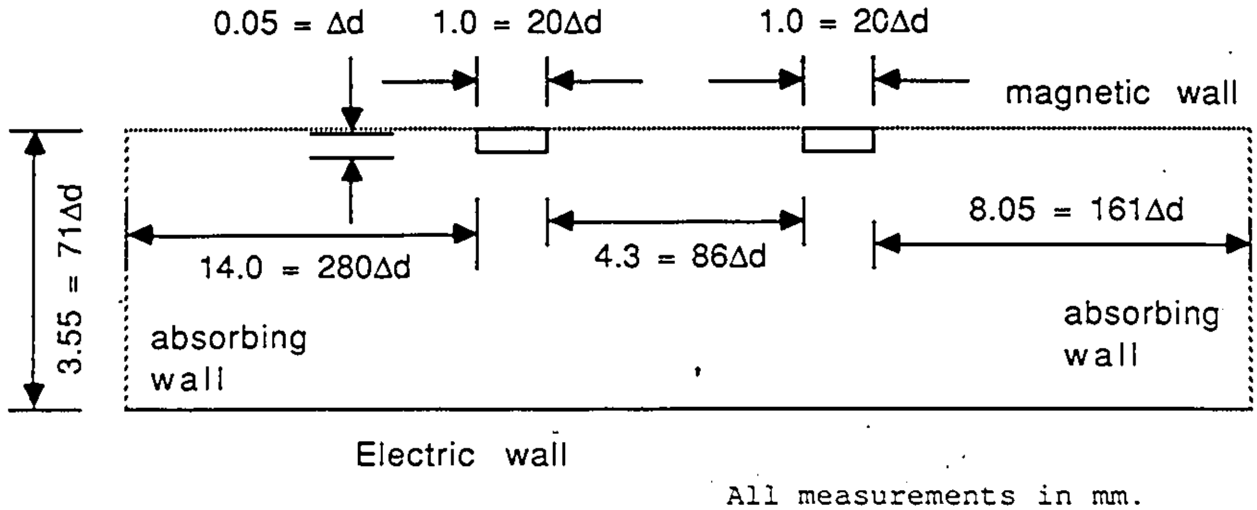


Figure 5.5: View of the filter simulation for use with the TLM method.

for a duration of at least one half-cycle of the incident wave, and find their maximum instantaneous field. A disadvantage (for other applications, since we are not concerned with phase response) is obvious: here, the phase information at each node is lost. Having obtained this, we can directly compare results between TLM and FEM. On the other hand, the reflection coefficient is not directly calculated from the field at the boundary (as with FEM), but from a more fundamental and physically related method, that of the slotted transmission line. In fact, the TLM method being a time-domain method that yields the field at any point and at any time, we may insert a "software detector diode" in a line oriented in the longitudinal direction of the median of the guide. Then, the peak amplitude is found everywhere on that line, and, thus, a typical VSWR pattern for the input of the filter can be obtained. From the position of a minimum, we may easily get the phase of the reflection coefficient (referred to the first septum), and from the ratio of the maximum to the minimum, the amplitude of the reflection coefficient. One must remember that the maximum and minimum must be located in a region where only the fundamental mode exists. Consequently, the boundary at the front of the filter is further located than for FEM which requires only that the absorbing walls not be put in contact with higher-order modes.

Chapter 6

Results.

6.1 Introduction.

In previous chapters, the FEM and TLM method were described, and the former was discussed in detail. Here, the final results obtained from the FEM and the TLM method will be shown, and some difficulties encountered while deriving them discussed. All graphs showing field and current distribution in the filter are normalized to 1W of incident power. Most graphs show, simultaneously, results obtained from FEM and TLM. Some graphs also include results from mode matching (MM).

6.2 Results from FEM.

6.2.1 System size, and CPU time.

The FEM yields a system of N by N linear equations (chap. 4), N being the number of free (unknown) node field values. The system is always sparse [4], so that special techniques may eventually be implemented in order to solve the system in a minimal amount of time, or to minimize storage requirements. Indeed, an entry indexed A_{ij} of the final symmetric matrix is non-zero only if nodes i and j are from the same element, or if they are from adjacent triangles. Since most triangles in a large mesh are not adjacent, then it follows that the matrix A must be sparse. The matrix reduction technique used is Gauss-Jordan elimination, using pivots, available as a standard LINPACK program from Argonne National Lab (node address NETLIB@ANL-MCS, on ARPANET network). This subroutine takes advantage of the fact that the system is symmetric by storing only the upper triangular part of the matrix A, thus reducing memory requirements by a half. IEEE double precision (8-byte REAL*8, or 16-byte COMPLEX*16) representation is used everywhere in the FEM program, included as an appendix. Preliminary simulations by the author revealed that single precision was insufficient. The total number of COMPLEX unknowns is 628. This represents a total minimum storage of 3.17 Mbytes for the linear system alone, not including the program or other related variables. The shape functions used are of the 3rd degree. The gridding algorithm is fairly simple. First, the x and z coordinates of the vertices of rectangles in a rectangular grid is introduced. Then, the program generates vertical and horizontal lines passing by all (x,z) coordinates previously entered. Finally, all resulting rectangles are automatically cut with a diagonal line, forming a grid of triangles. A total of 144 triangles were hence generated and used for all computations. The region in the front of the septa is particularly dense in triangles, so that the highly irregular field near those discontinuities can be correctly modeled. The x and z coordinates used are included in the computer program listing.

The CPU time, using a micro-VAX (VAXstation-2000), required to solve the field distribution in the 2-septum filter was typically in the order of 25 minutes. It was observed that 1 minute of that time was spent setting up the matrix system, while the rest of the time was used to solve the system of linear equations using LINPACK double precision COMPLEX Gaussian elimination routines. The same program was run on the University of Toronto Cray XM-P supercomputer. The total CPU time used was 2.2 seconds. This represents a drastic improvement of nearly 3 orders of magnitude. Future larger simulations (for 3 septa and above) will be made possible by the use of such calculating machines.

An interesting note comes from [26], which states that the accuracy obtained using the FEM will be increased if the triangles describing the region Ω have one side which follows the constant values of the field to be computed. Although the field values are complex, [26] claims that it is sufficient, in most cases, to have one side of the triangles follow the curves of constant modulus, even though the real and imaginary part may be radically different. This would require a rather complicated automatic mesh generating program, or that all coordinates of all triangles be interactively generated.

6.2.2 Plotting FEM results: some considerations.

Accurately plotting the field values at arbitrary points between the nodes may not be trivial. There are two cases that we can consider, the first being relatively easy to solve. If we wish to plot the field along the edges of any triangle that is aligned with one of the Cartesian axes, a simple 3rd degree curve-fitting process must be made with the field values at the 4 nodes on the edge. This process is known to produce a unique polynomial. Since the field on the edges has also been modeled by a 3rd degree polynomial, then this polynomial must represent exactly the solution obtained by FEM. If the line on which we want to plot the values of the field does not follow the edge of a triangle, then we must find the 2-dimensional 3rd degree polynomial which models the field everywhere inside the triangle (equation 4.15). This process is analytically similar to the preceding method, with the added difficulty that we must do a 2-D curve-fitting process. Note that $\underline{u}(x,z)$ is defined everywhere within a triangle as:

$$\begin{aligned}\underline{u}(x,z) = a_1 + a_2x + a_3z + a_4x^2 + a_5xz + a_6z^2 \\ +a_7x^3 + a_8x^2z + a_9xz^2 + a_{10}z^3\end{aligned} \qquad (6.1)$$

where a_i are complex coefficients. Since the value of $\underline{u}$ and the coordinates of all 10 nodes are known, the coefficients a_1 to a_{10} can be identified by solving a 10 by 10 linear system of equations. But beware! Double precision arithmetic must absolutely be used, as the matrix is Vandermonde-type, and is usually ill-conditioned. The author has verified that a typical estimate of the condition number of such a matrix system is in the vicinity of 10^{15}!

Finding an accurate representation of the derivative of the field values is also important, since $\underline{H}_x$ and $\underline{H}_z$ are found by differentiating the E-field, using Maxwell's curl equation:

$$\nabla \times \vec{\underline{E}} = -j\omega\mu\vec{\underline{H}}.$$

$$\underline{H}_x = \frac{-j}{\omega\mu}\frac{\partial \underline{E}_y}{\partial z}$$

$$\underline{H}_z = \frac{j}{\omega\mu}\frac{\partial \underline{E}_y}{\partial x}$$

Using the same method, as previously discussed to obtain the 2-D approximation polynomial, an analytical derivative is easily done with respect to either Cartesian coordinates yielding components of the H field at any interpolated point. Note that, for reasons explained in the next subsection, the continuity of the $\underline{H}$ field from one triangle to the next is not guaranteed, while the $\underline{E}_y$ field is always continuous.

6.2.3 E and H field.

The factor $\underline{E}_o$, discussed in chapters 3 and 4, which represents the amplitude of the incident E-field, is normalized so that the incident power is 1 W. This value is used in all analysis discussed throughout this thesis. It is given by [33]

$$\underline{E}_o = \sqrt{\frac{4\omega\mu_o}{k_z ab}} \tag{6.2}$$

where a and b are the waveguide width and height, respectively, and k_z is the propagation factor described in chapter 3. Changing the phase of the complex quantity $\underline{E}_o$ would simply yield a corresponding phase shift on all output quantities, but would not affect the amplitude. The output from the FEM program, for which the algorithm was described in chapter 4, directly yields the value of the electric field component $\underline{E}_y$ at all nodes.

Figures 6.1 and 6.2 show a longitudinal view of the amplitude of $\underline{E}_y$ in the median of the waveguide, while figures 6.3 and 6.4 show a lateral view of the same field at the front of the first septum. The maximum $\underline{E}_y$-field amplitude occurs along the median of the waveguide. Several longitudinal views of the field along different x-coordinates reveal that the peak occurs on the median of the guide. Thus, only views at this coordinate are reported. From figure 6.1, the peak amplitude of $\underline{E}_y$, in the middle of pass-band, is in the vicinity of 32.1 KV/m, while the peak amplitude of the field before the filter is near 8.8 KV/m. This implies that a filter can only sustain a maximum input power of

$$P_{max-filter} = (\frac{8.8}{32.1})^2 P_{max-no-filter}$$

Thus, the filter structure can sustain 13.3 times less power than the empty waveguide before air breakdown occurs (assuming matched output conditions, as stated in section 3.2). The filter analyzed uses WR-28 housing (7.112 mm by 3.556 mm). The peak power handling of such an empty guide is in the vicinity of 60 kW [34]. Peak power handling of the filter is thus reduced to approx. 4.5 kW. Graphs of longitudinal E-field distribution are shown for 2 different frequencies: in pass-band (32.5665 GHz), and out of pass-band (30.0 GHz). Note the typical VSWR pattern appearing at the input port of the out-of-band simulations. As expected, the peak E-field for out-of-band behavior occurs at the input port, before the filter, and is due to a near-perfect standing wave. The maximum field amplitude is then twice the peak value of the incident wave.

Figures 6.5, 6.6, 6.9 and 6.10 show the $\underline{H}_x$ field in longitudinal and lateral cuts, at band-pass and out of band-pass frequencies. Note that the magnetic field is not continuous, from one finite element to another. The continuity of the E-field across a finite element interface is guaranteed, analytically and numerically, whereas it is only guaranteed analytically for the H-field. A condition of continuity of the derivative of the E-field from one triangular region to the next is not imposed, nor does it appear naturally, for the numerical solution, as the E-field's expansion functions do not correspond exactly to the true solution. Such discontinuities in the H-field occur, for example, in Fig. 6.9. One must always remember that any numerical method applied to a problem is an *approximation* to the original problem, and is, in fact, a different problem that merely resembles the original. As shown by figures 6.7 and 6.8, $\underline{H}_z$ is zero everywhere on the median because of the magnetic wall C_5. From these graphs, one can observe the high amplitude of the magnetic field at the surface of the septa, yielding large surface currents. These will be discussed later.

6.2.4 Reflection coefficient.

The reflection coefficient is computed by using equation (3.1). Isolating $\underline{R}$ yields the reflection coefficient anywhere to the left of C_1:

$$\underline{R} = \left(\frac{\underline{\phi}}{\underline{E}_o \sin(k_x x)} - e^{-jk_z z}\right)e^{-jk_z z} \tag{6.3}$$

On boundary C_1 (z=0), the reflection coefficient is computed as

$$\underline{R} = \left(\frac{\underline{\phi}}{\underline{E}_o \sin(k_x x)} - 1\right) \tag{6.4}$$

Thus, a crude test to see if the simulation is correct is to compute the reflection coefficient on every point on the boundary C_1, that is for $0 \leq x \leq a/2$. It is seen from figure 6.11 that the computed reflection coefficient is fairly stable, except sometimes near the edge of the waveguide, and at frequencies where the reflection coefficient is very low. This can be explained by examining the following:

$$\frac{\partial \underline{R}}{\partial \underline{\phi}} = \frac{1}{\underline{E}_o \sin(k_x x)}$$

$$\rightarrow \quad \frac{\Delta \underline{R}}{\underline{R}} \cong \frac{\Delta \underline{\phi}}{\underline{R}\,\underline{E}_o \sin(k_x x)}$$

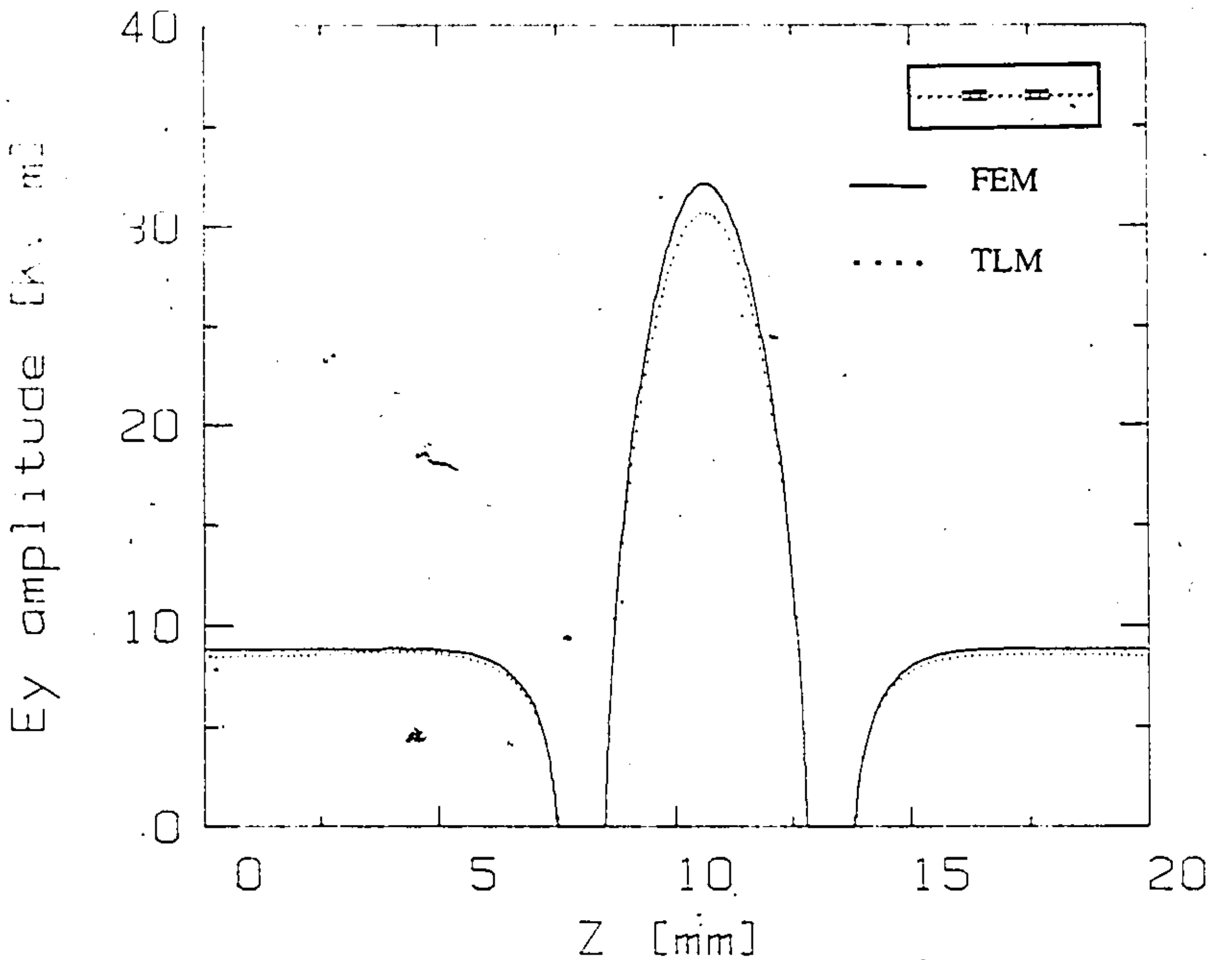


Figure 6.1: Longitudinal view of amplitude of $\underline{E}_y$, in pass-band; ... = TLM (32.5 GHz), continuous = FEM (32.5665 GHz).

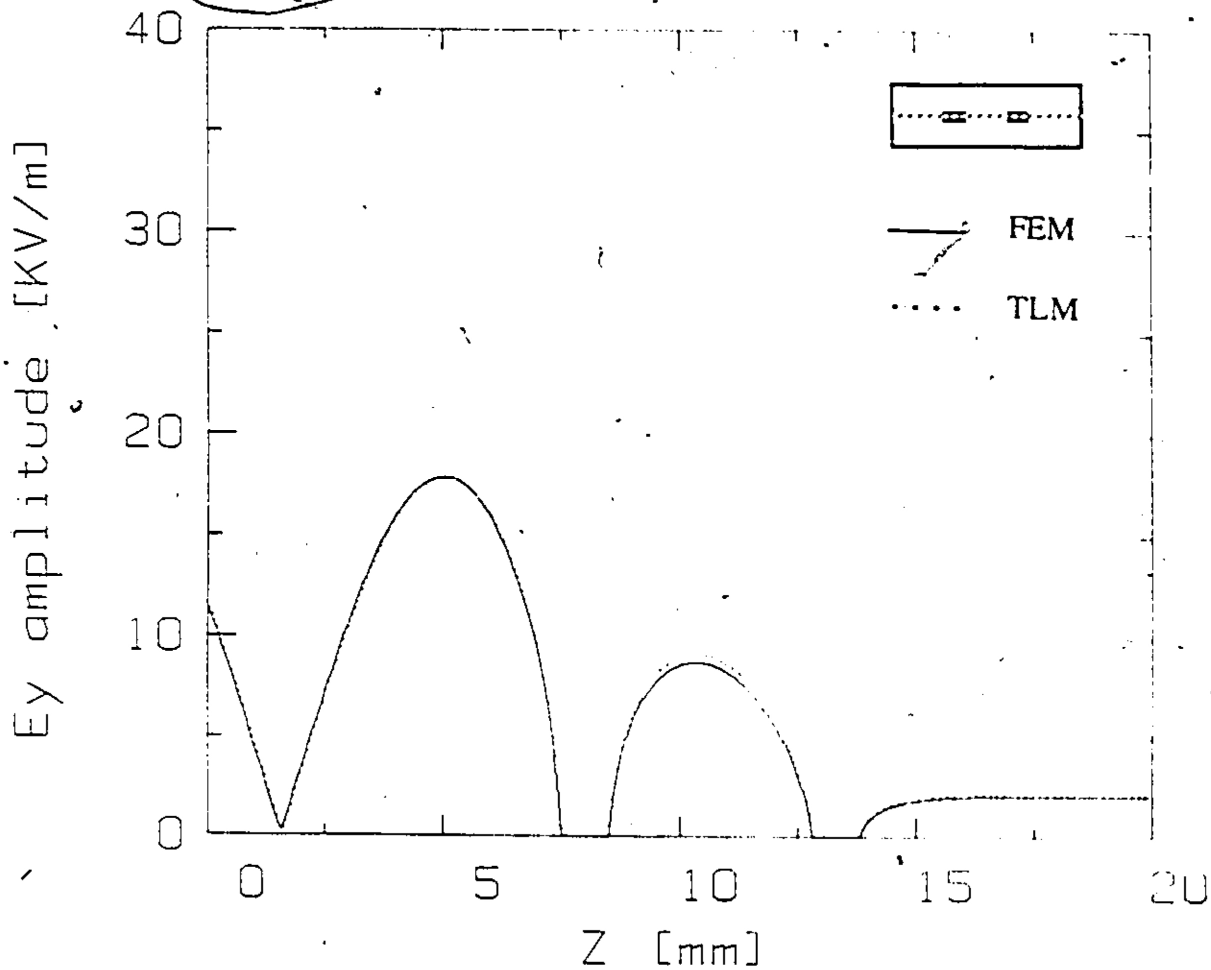


Figure 6.2: Longitudinal view of amplitude of $\underline{E}_y$, out of pass-band (30 GHz): ... = TLM, continuous = FEM.

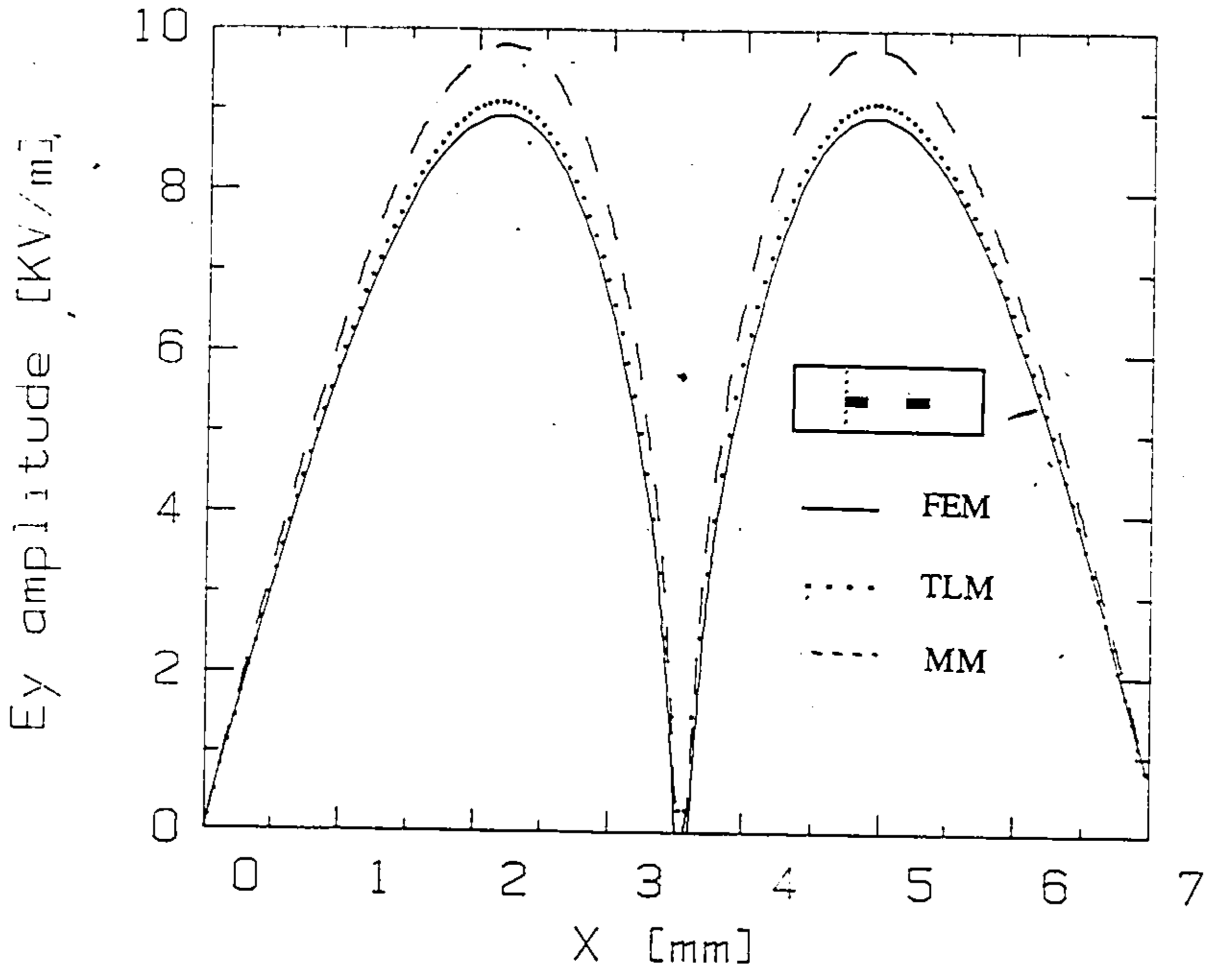


Figure 6.4: Lateral view of amplitude of $\underline{E}_y$, out of pass-band (30 GHz): ... = TLM, continuous = FEM, dash = MM.

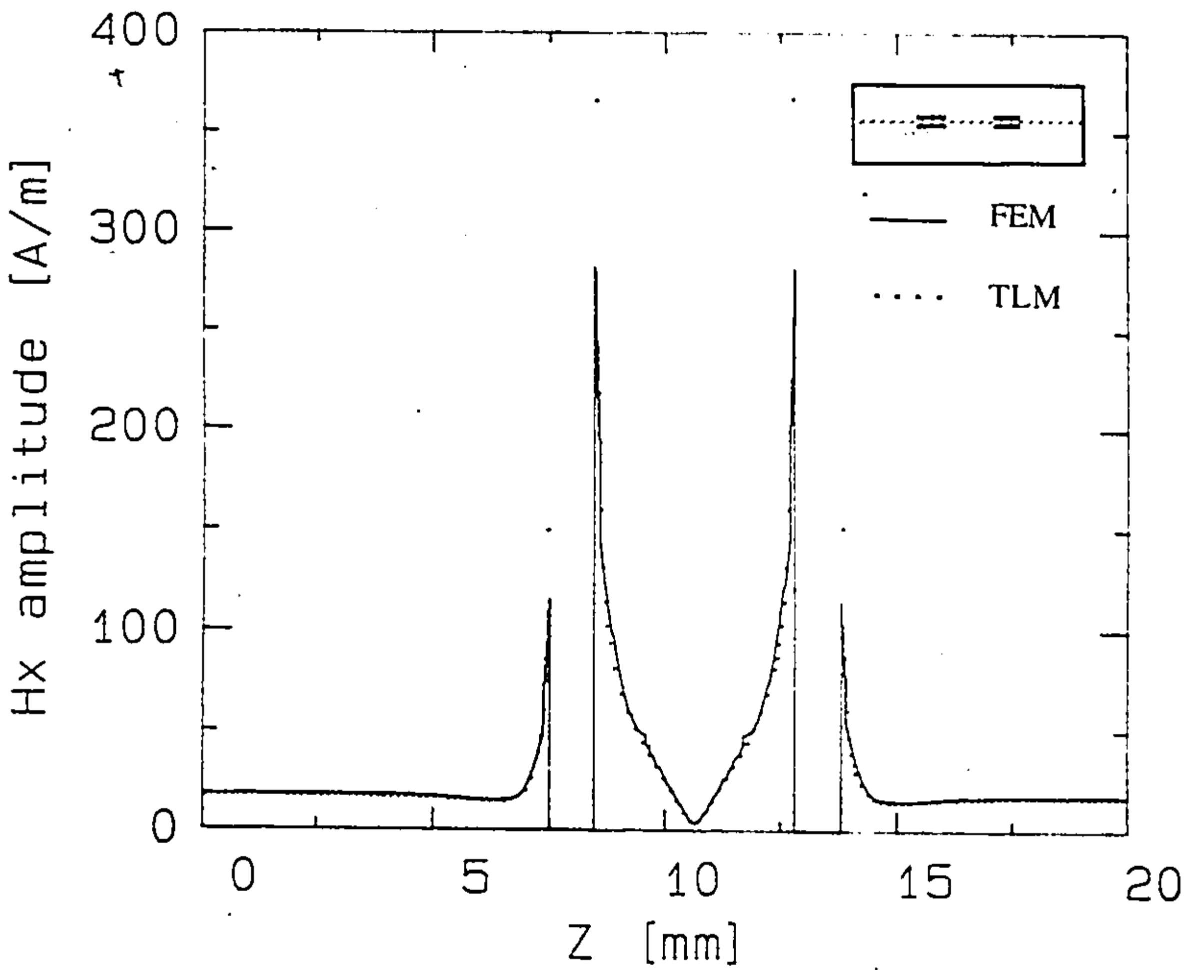


Figure 6.5: Longitudinal view of amplitude of $\underline{H_x}$, in band-pass: ... = TLM (32.5 GHz), continuous = FEM (32.5665 GHz).

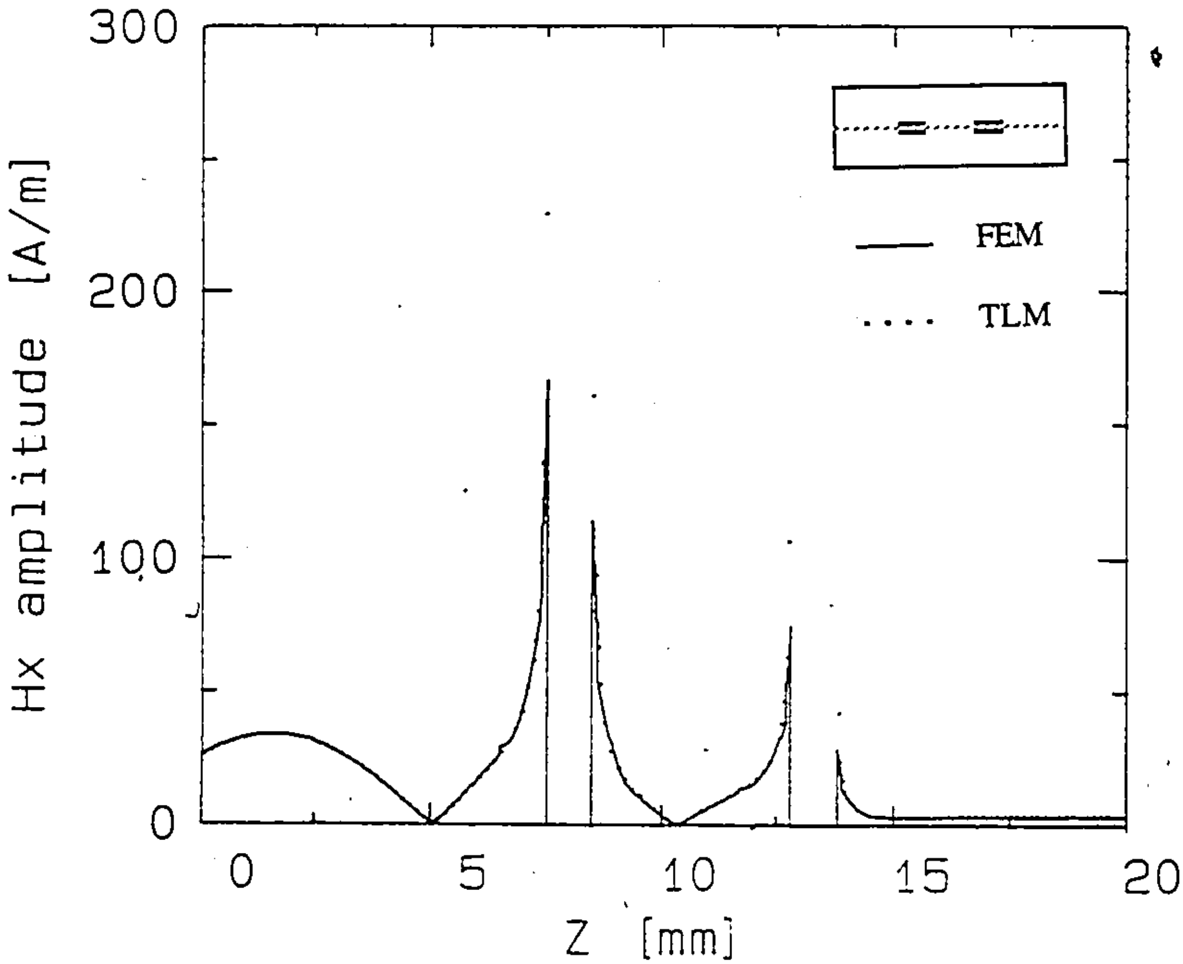


Figure 6.6: Longitudinal view of amplitude of $\underline{H_x}$, out of band-pass (30 GHz): ... = TLM, continuous = FEM.

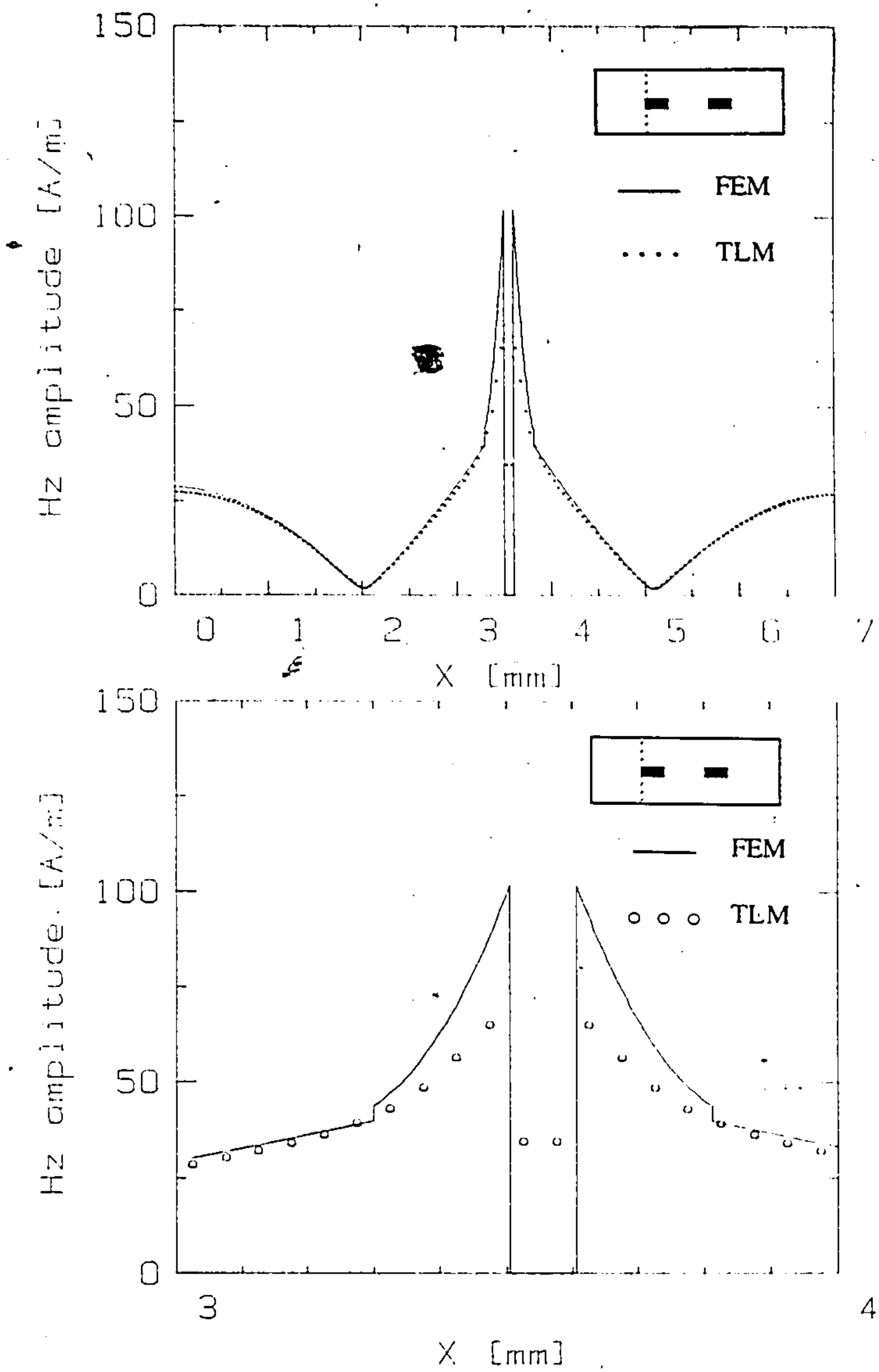


Figure 6.7: Lateral view of amplitude of $\underline{H}_z$, in pass-band: ... = TLM (32.5 GHz), continuous = FEM (32.5665 GHz). Top: complete view. Bottom: close-up of septum region.

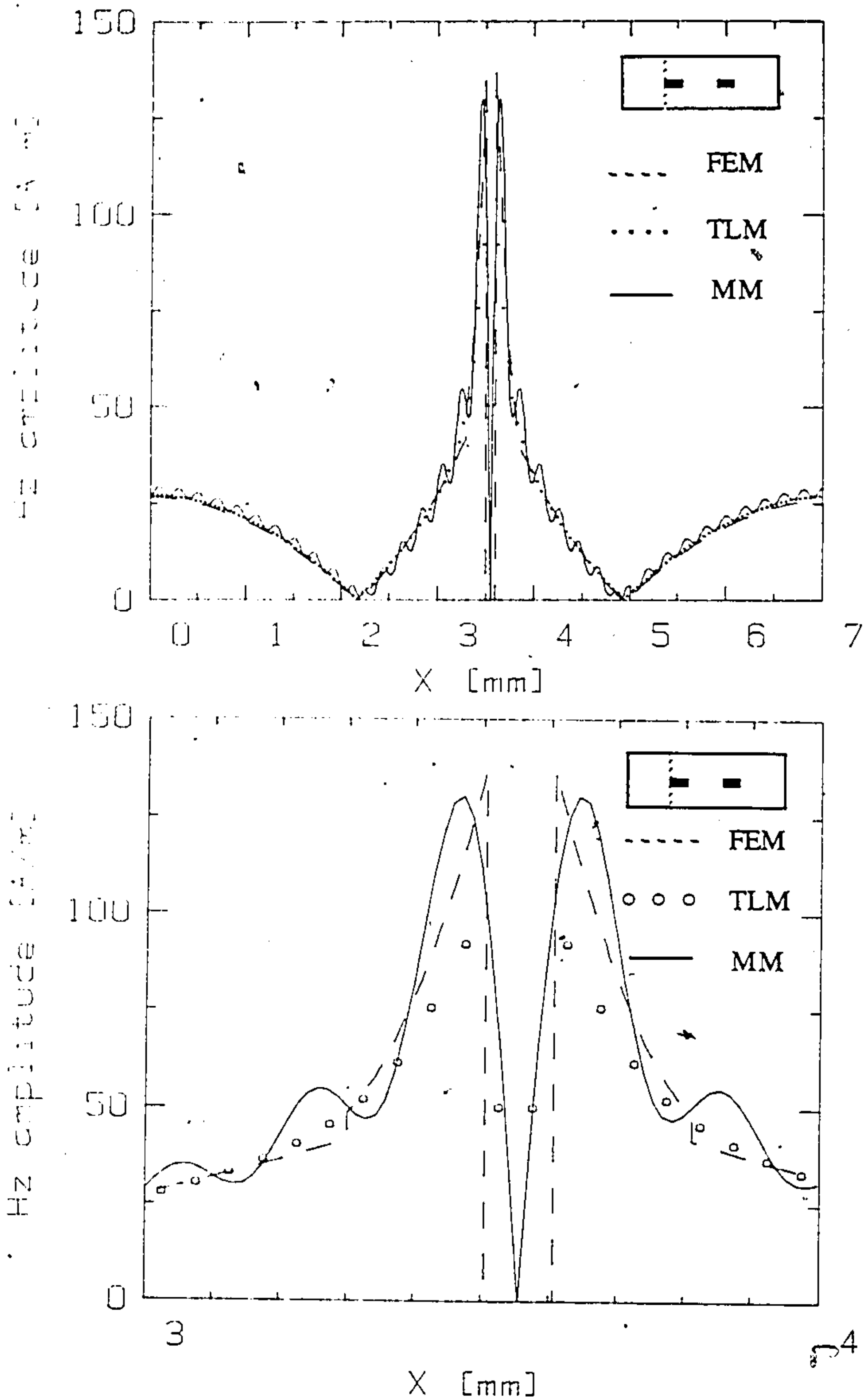


Figure 6.8: Lateral view of amplitude of $\underline{H_z}$, out of pass-band (30 GHz): ... = TLM, dash = FEM, continuous = MM. Top: complete view. Bottom: close-up of septum region.

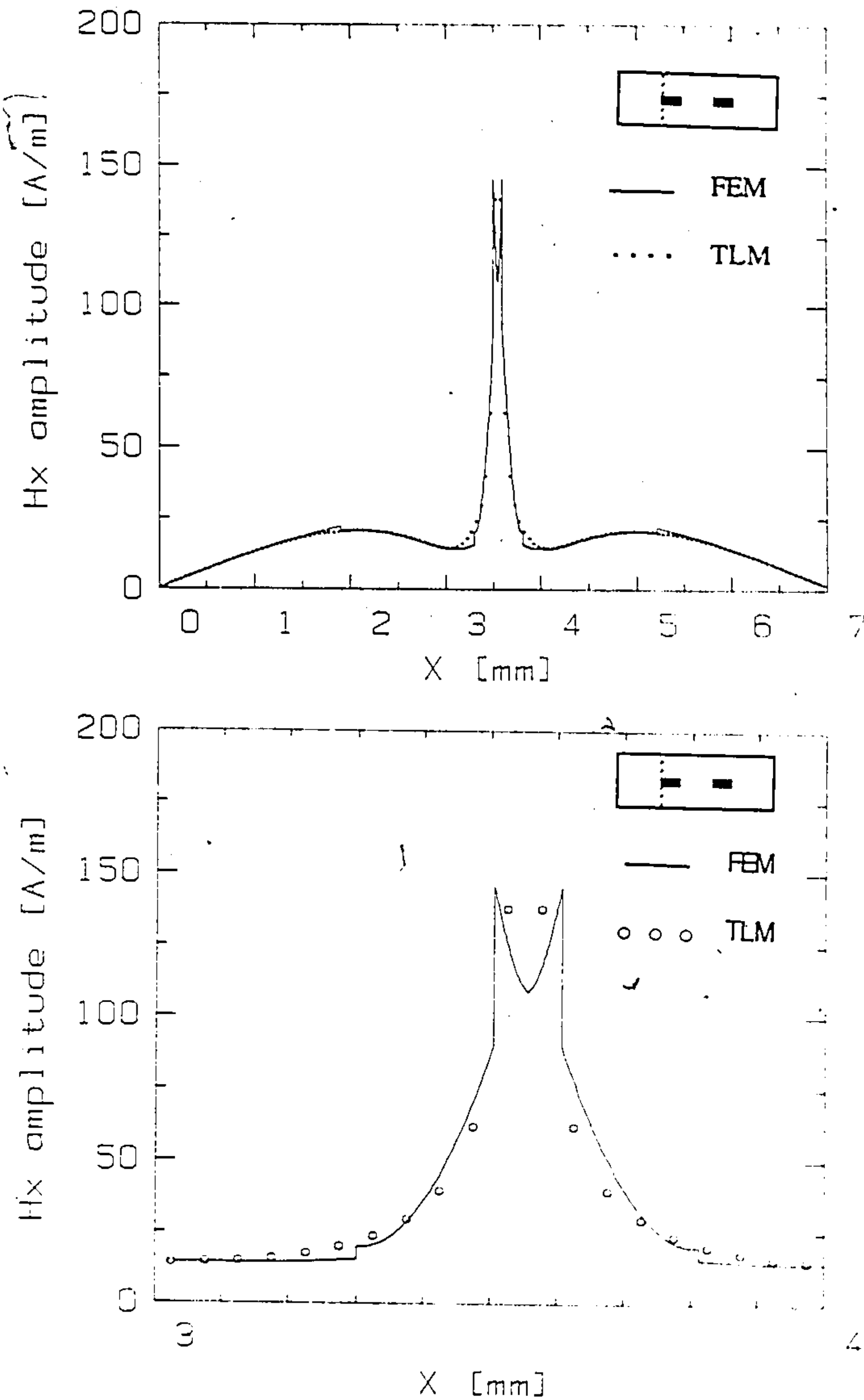


Figure 6.9: Lateral view of amplitude of $\underline{H_x}$, in pass-band: ... = TLM (32.5 GHz), continuous = FEM (32.5665 GHz). Top: complete view. Bottom: close-up of septum region.

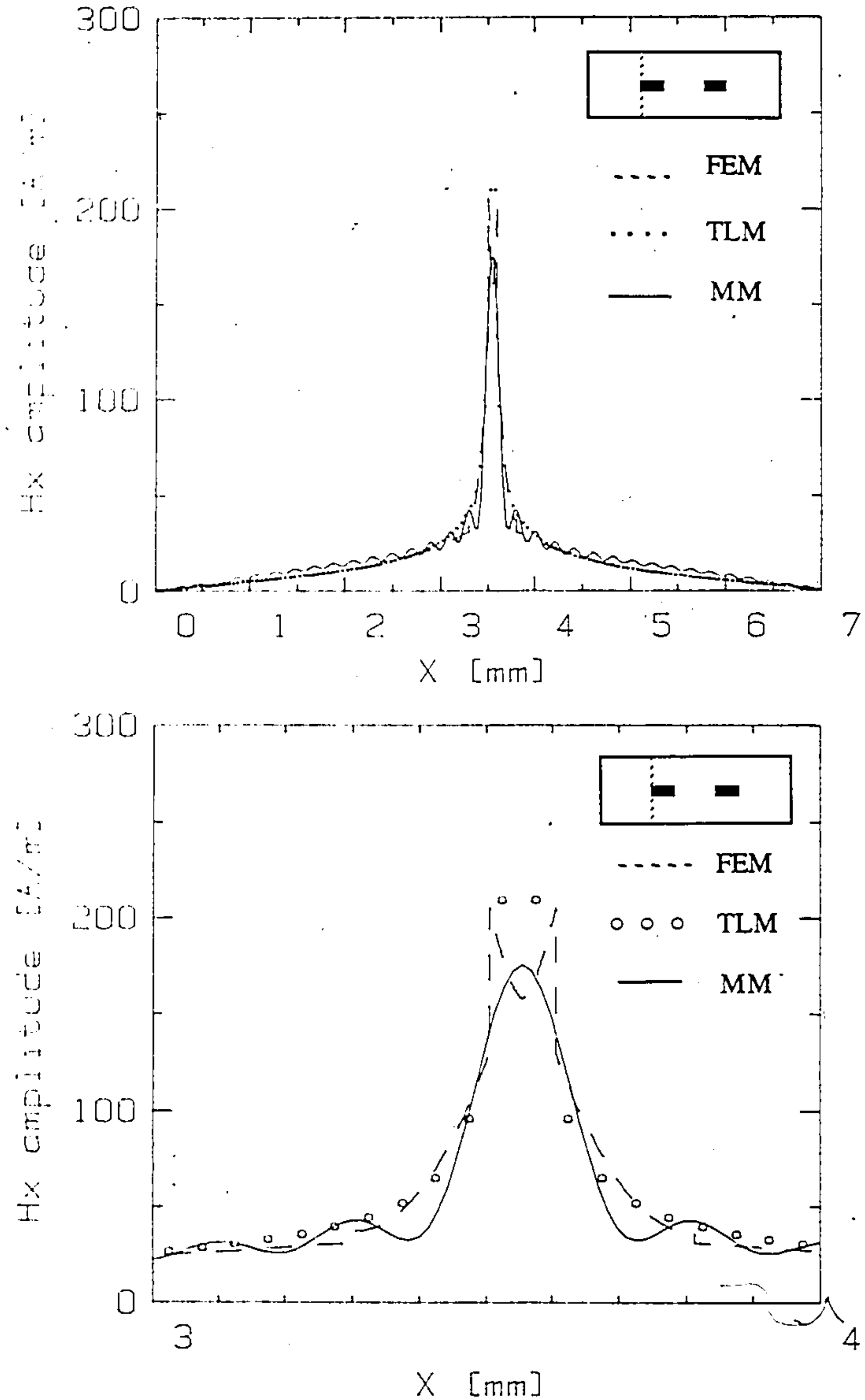


Figure 6.10: Lateral view of amplitude of $\underline{H}_x$, out of pass-band (30 GHz): ... = TLM, continuous = FEM, dash = MM. Top: complete view. Bottom: close-up of septum region.

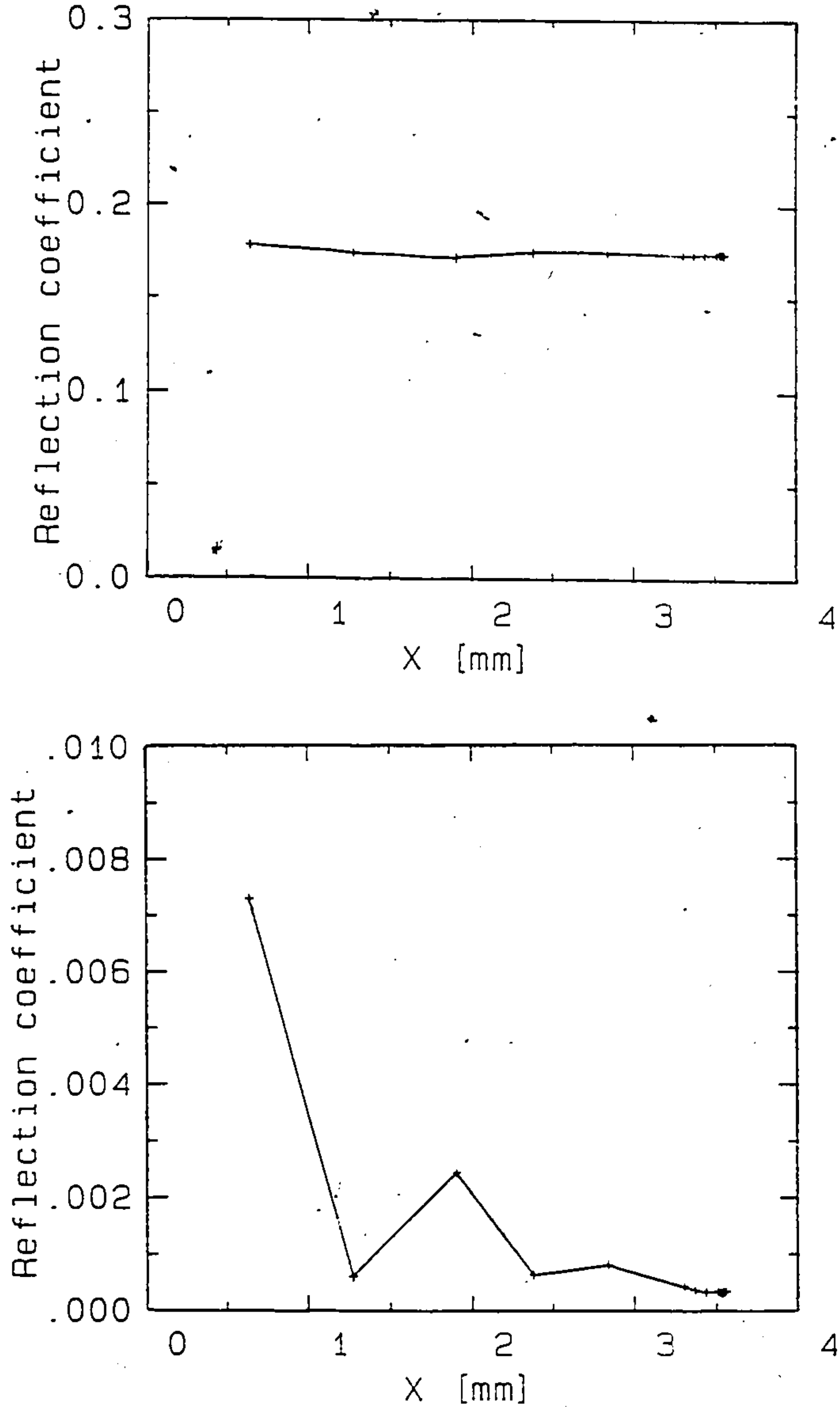


Figure 6.11: Reflection coefficient as a function of lateral coordinate X (from FEM). Top: 32.42 GHz. Botom: 32.5665 GHz.

Consider $\Delta\underline{\phi}$ to be the computational error. We can assume this error not to vary very widely within the range of frequencies of interest. Then $\Delta\underline{R}/\underline{R}$, the relative computational error in the reflected coefficient, becomes more important as $\underline{R} \rightarrow 0$, or as $\underline{E}_o \sin(k_x x) \rightarrow 0$ (near the waveguide wall). Thus, to minimize errors, all reflection coefficients are computed at x=3.556 mm on C_1 (along the median of the guide).

Figure 6.12 compares reflection coefficients obtained by FEM, MM and TLM, as a function of frequency. The middle of pass-band obtained by FEM is seen to be 0.8% higher than results obtained by MM. This error is relatively small, but might be considered excessively large for the design of precision filters. But it is stated again that FEM is not meant, by the author, as a means of replacing existing CAD packages, such as EPLANFIL2, but as a method to analyze field distribution in a filter with given geometry. The field distribution at the pass-band frequency obtained by FEM is assumed to correspond to the field distribution at the pass-band obtained by MM, which was found to agree very closely with measurements [18].

6.2.5 Current on septa.

The surface current density on all metallic structures of the filter is given by:

$$\underline{\vec{J}_s} = \vec{n} \times \underline{\vec{H}} \qquad (6.5)$$

where $\vec{n}$ is a unit outward normal to the metal surface, and $\underline{\vec{H}}$ is the H-field at the surface of the conductor. As all electromagnetic field quantities in the problem investigated are constant with respect to y, current densities are also constant along that direction. Using E_y-field data obtained from the FEM program, the H-field is computed as discussed in section 6.2.2. The current distribution, obtained from 6.5, is shown in Fig. 6.13 and Fig. 6.14, for pass-band and out of pass-band frequencies, respectively. Note the increase in current density near the edges of the septum.

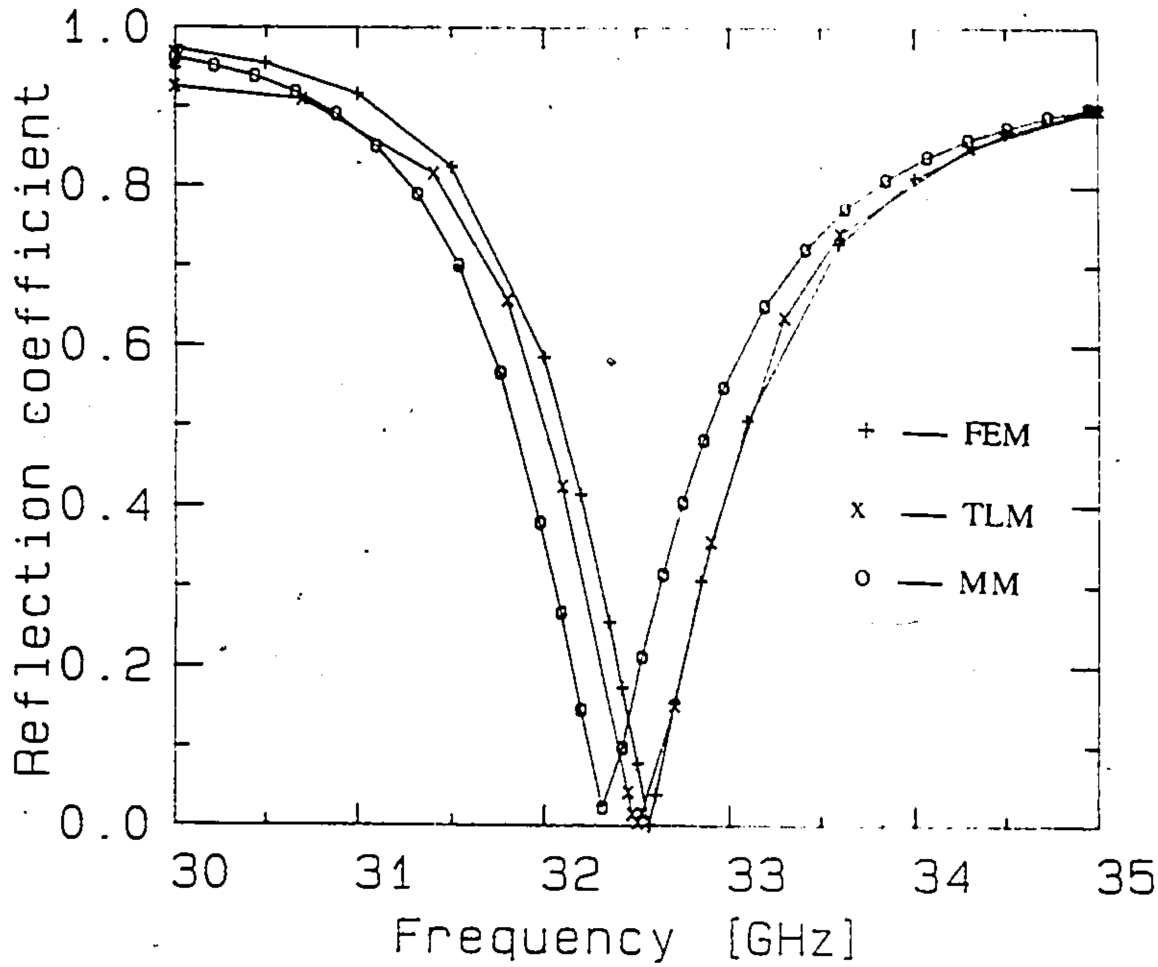


Figure 6.12: Reflection coefficient as a function of frequency: x = TLM, + = FEM, o = MM.

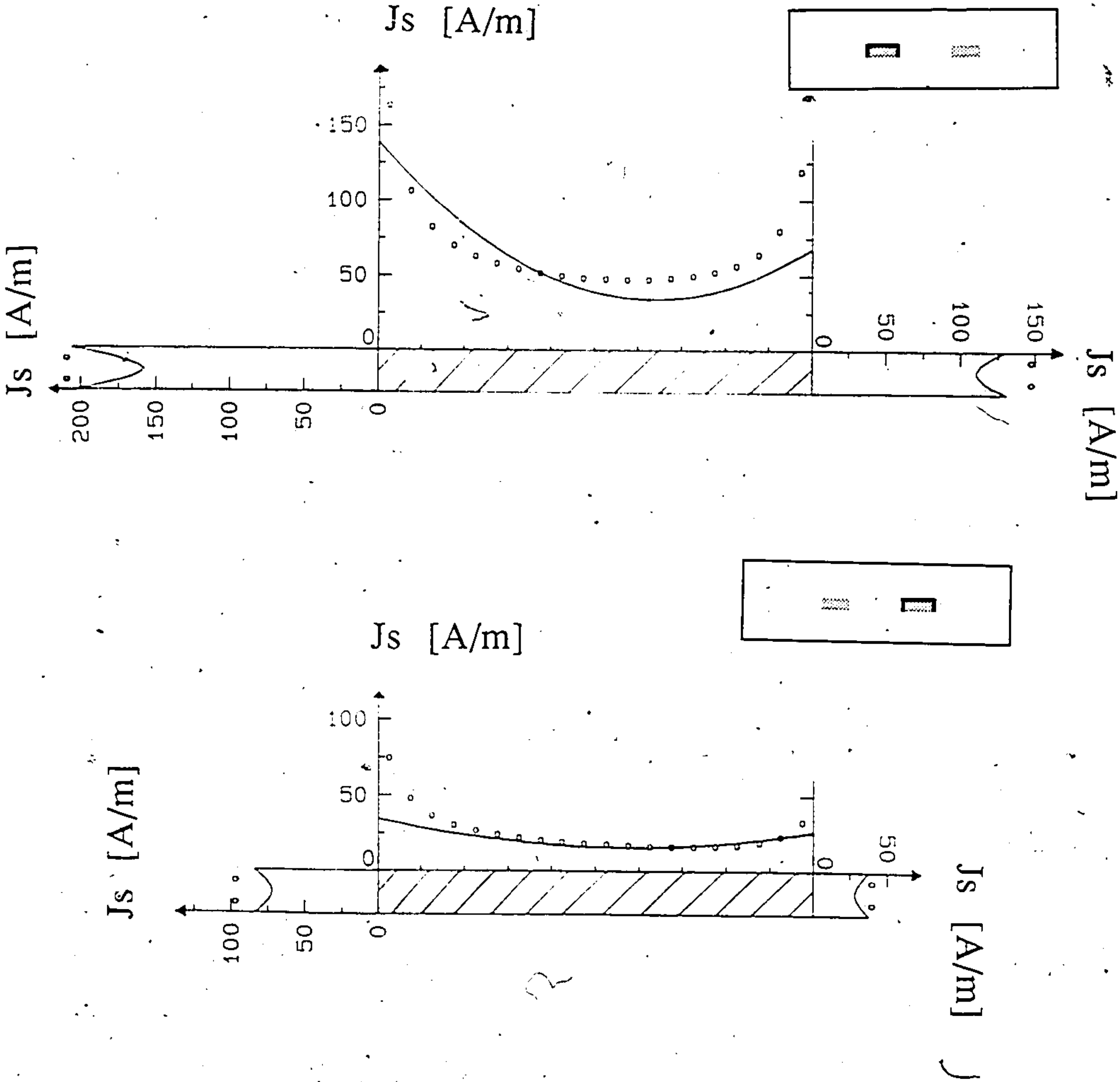


Figure 6.14: Current distribution out of pass-band (30 GHz). Top: input septum. Bottom: output septum (o = TLM, continuous = FEM).

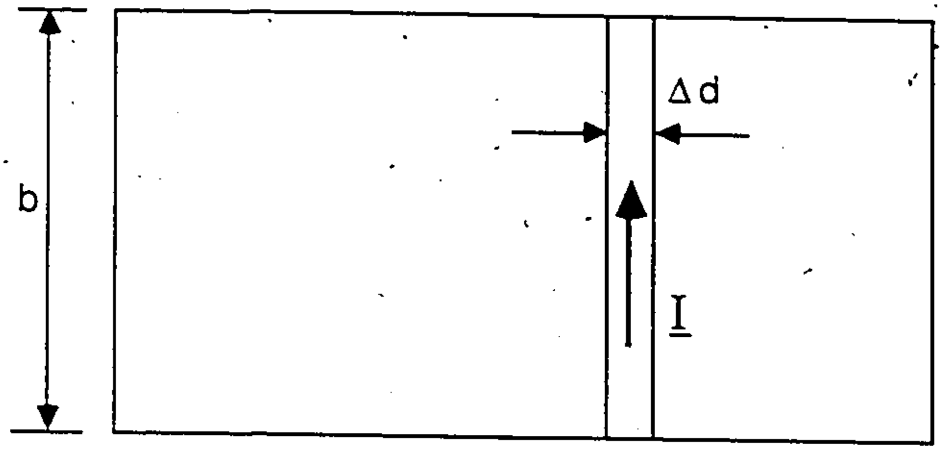


Figure 6.15: View of current strip on side of septum (section 6.2.6).

6.2.6 Evaluation of power dissipated on a septum.

Consider the current on all faces of the septums to be approximated by a series of strips of width Δd, one of which is shown in Fig. 6.15. The power dissipated on that strip is

$$\Delta P_i = \frac{1}{2} \mid \underline{I} \mid^2 R_i = \frac{1}{2} \mid \underline{J}_{s_i} \mid^2 R_i(\Delta d)^2 = \frac{1}{2} \mid \vec{n} \times \underline{\vec{H}} \mid^2 R_i(\Delta d)^2 \tag{6.6}$$

where $\vec{n}$ is a unit normal vector, directed out of the septum, $\underline{\vec{H}}$ is the magnetic field at the surface of the septum, and R_i is the resistance of the strip of length b shown in figure 6.15. From Ohm's law,

$$R_i = \frac{l_i}{A_i \sigma}$$

where R_i = resistance of strip i

l_i = length of strip $i = b = 3.556$mm

A_i = cross-section area through which the current flows

σ = conductivity of material.

A is found to be

$$A = \delta_s \Delta d = \Delta d \sqrt{\frac{2}{\omega \mu \sigma}}$$

where δ_s is the skin depth. The total power dissipated on the septum is:

$$P = \oint_L \frac{|\vec{n} \times \vec{H}|^2}{2} \frac{b}{\sigma \delta_s} dL \tag{6.7}$$

where L is the 2-D contour of the septum.

Using the data given in part in figures 6.13 and 6.14, the total power dissipated on each septum as a function of frequency is given in Fig. 6.16. The maximum power dissipated by any single septum is 2.4 mW (-26.2 dbW), at 32.2 GHz (below pass-band), by the input septum. In the middle of pass-band, exactly the same power is dissipated by each septum (approximately 2.0 mW, or -27.0 dbW). This is expected, since the field distribution in the middle of pass-band is nearly perfectly symmetrically distributed, with respect to the center of the filter.

The maximum admissible CW power for an empty WR-28 guide is rated at approx. 250 W [34], implying that dissipation on the input septum will not exceed 600 mW, while the output septum will dissipate a maximum of 490 mW. This is a relatively small power, except for paper-thin metallic septa. But at maximum CW input power, the inner waveguide walls will become relatively hot, which will further limit the maximum power dissipation from the septa. They will not melt, but will deform through thermal expansion, causing degradation in performances. Reduction of skin depth is also an important factor, as power dissipation increases as a function of the square root of the frequency. As frequency is increased toward 94 GHz and above, anomalous skin effect increases this power by as much as 1.5 times [34]. The exact CW power limit has yet to be determined.

6.3 Results from TLM.

It should be pointed out that, while the system to be solved by the TLM method was relatively large (568 by 73 nodes, see Fig. 5.5), single precision IEEE REAL format (4

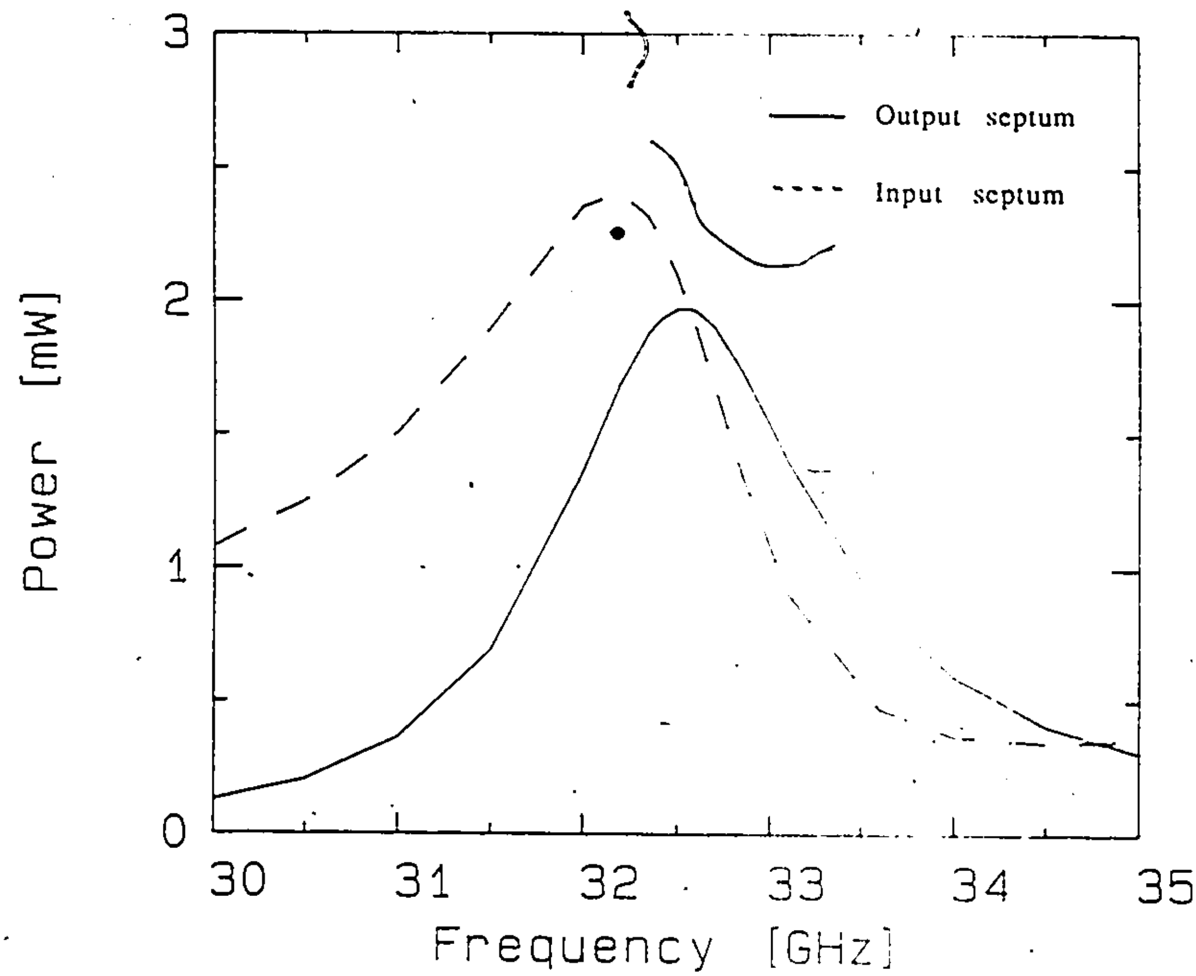


Figure 6.16: Total power dissipated on each septum as a function of frequency. (continuous = output septum, dash = input septum)

bytes) calculation was sufficient. Indeed, the use of double precision IEEE REAL format (8 bytes) did not change the results within 4 decimal digits of accuracy. This is due to the inherently good numerical stability of the TLM method, which is a purely physical and iterative model that does not require any matrix inversions.

Two programming schemes were tried out. The first, used in Y. C. Shih's M.A.Sc. thesis [16], implements a clever scheme in which only one matrix of node currents is used (see section 5.2.1). The matrix is updated at every iteration without using an intermediate matrix. Since TLM matrices tend to be large, this is a notable improvement over the author's attempt, which required the use of 2 matrices, one for iteration k (the old iteration), and another for iteration $k+1$ (the new iteration). But the elementary operations on each node could not be readily transformed into a matrix equation, as was presented in section 5.2.1. This is a pitfall if the program is to be executed on a vectorizing supercomputer, as will be discussed shortly. The author's method is much simpler to program, as the entries into the new matrix (iteration $k+1$) is strictly a matrix function of the values of the old matrix (iteration k). Shih's algorithm turned out to be nearly twice as efficient, both in terms of CPU time and memory storage, on a non-vectorizing computer (such as the VAX 750, and the microVAX).

In order to reach the steady-state, 5000 iterations were first used. After this, 150 iterations were needed to determine the peak amplitude for all nodes of interest. Figure 6.17 shows the peak E_y distribution thus obtained for different frequencies, at $x = {}^{a}/_{2}$, along the longitudinal section of the filter, using the TLM method. Note that the VSWR pattern at the input side of the filter never vanishes, even in the middle of the pass-band (near 32.5 GHz). Computations using 10000 iterations at the same frequency show a drastic decrease in input VSWR level, as illustrated by the same figure. Thus, the initial number of iterations has an impact on the computed field distribution (and thus on the determination of the reflection coefficient), mostly in the region of the input port. On

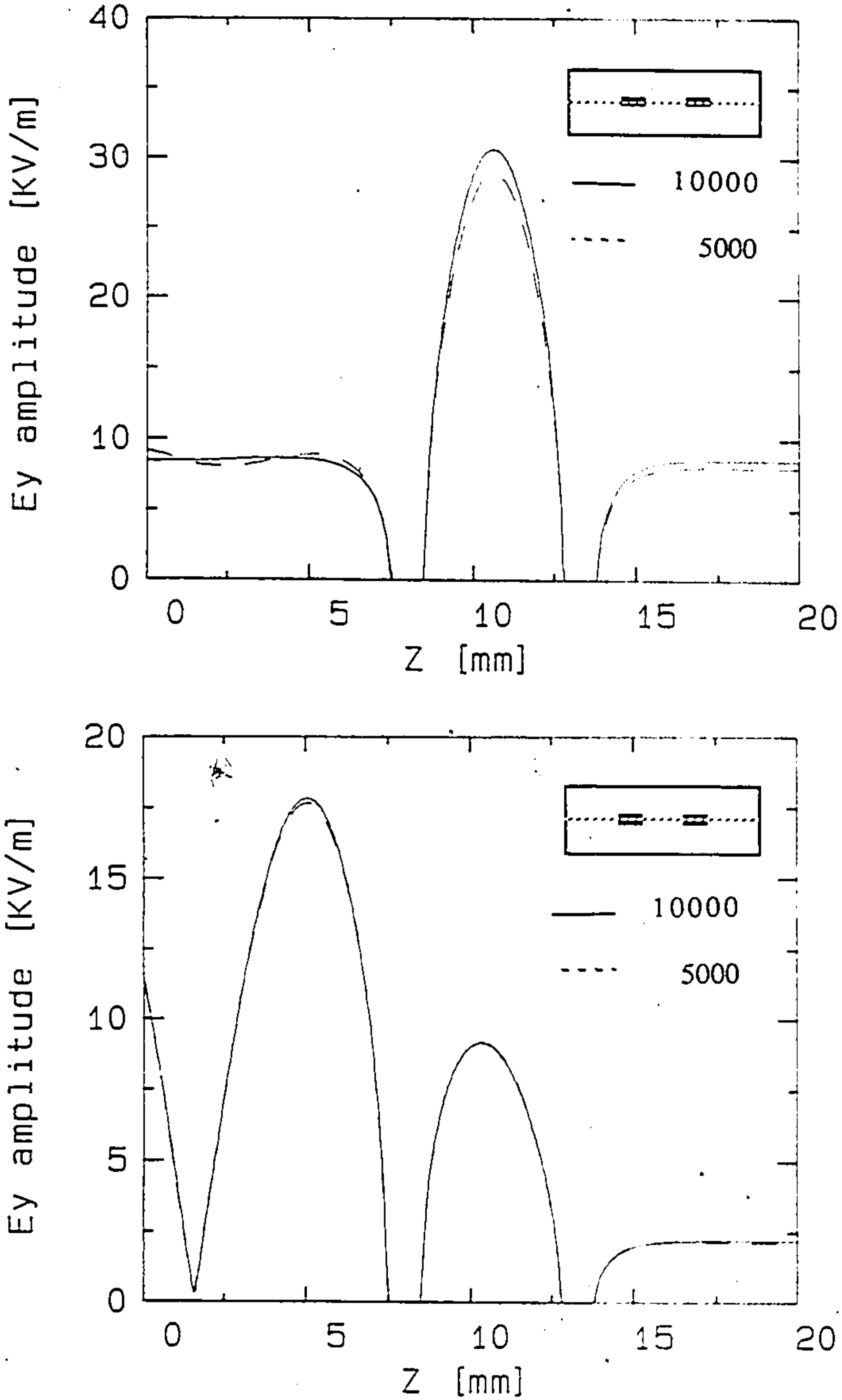
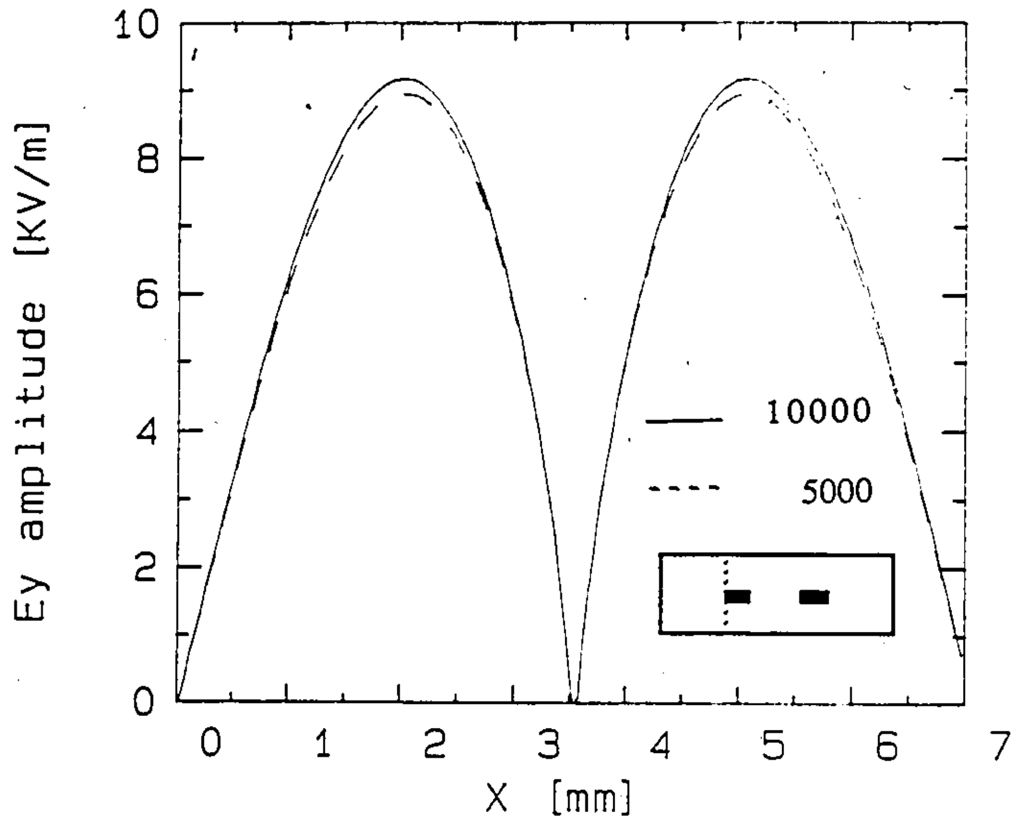


Figure 6.17: Longitudinal Ey-field distribution using TLM: continuous = 10000 iterations, dash = 5000 iterations. Top: pass-band (32.5 GHz). Bottom: out of pass-band (30.0 GHz).



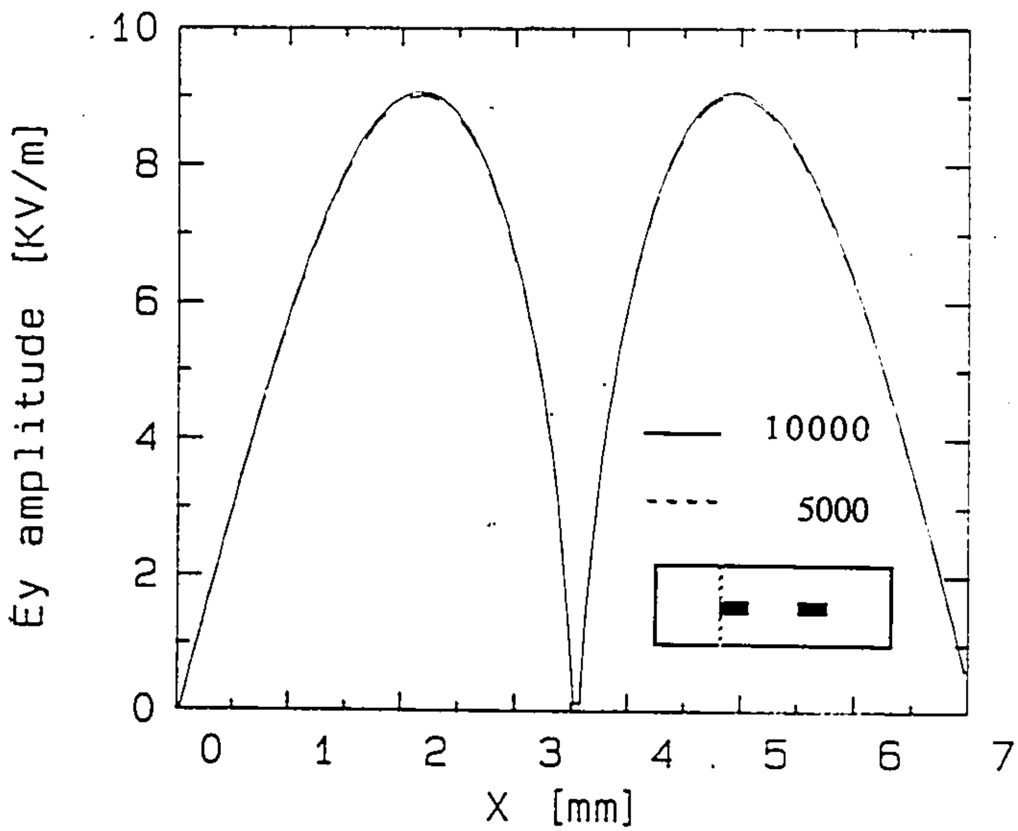


Figure 6.18: Lateral Ey-field distribution using TLM: continuous = 10000 iterations, dash = 5000 iterations. Top: pass-band (32.5 GHz). Bottom: out of pass-band (30.0 GHz).

the other hand, residual transient response has little effect at frequencies out of pass-band where the VSWR value is very high. This is illustrated by Fig. 6.19, which shows the reflection coefficient as a function of frequency, obtained with 5000 or 10000 initial iterations.

The TLM method is seen to suffer from the same ailment as all other iterative techniques: it is sometimes impossible (or very difficult) to predict how many iterations are needed to obtain a required accuracy. One possibility is to stop iterating when there is no significant change in the magnitude of the output quantity for a given number of iterations. This scheme is only valid for steady-state analysis, as used in this thesis.

Another problem encountered is the limited spatial resolution. Detailed information of the field at the input of the first septum is impossible, with the proposed minimum spatial resolution of $\Delta d = 50\mu m$. The only information obtained is a rough average of the current on the front side of the septum. No idea as to the shape of the current distribution can be gained, whereas the FEM can clearly display it. In order to obtain enough resolution and, therefore, see the irregularities of the current distribution as clearly as with the FEM, we would have to use Δd at least three times as small. This implies three times as much computer storage, and 9 times as much CPU time. The actual problem takes as much as 4 hours (for frequencies out of pass-band) and 8 hours in pass-band (because of increased number of iterations needed in pass-band). Thus this method becomes much less efficient than the FEM, for the computation of current distributions. However, if one wishes to compute the reflection coefficient and E_y field distribution only, the simplicity of the method makes it more attractive than the FEM. Indeed, a working TLM program can be developed and debugged in as little as 2 days (or even less for experienced programmers). In order to increase the efficiency of the TLM method, one would have to use an irregular mesh proposed by Saguet and Pic [15],which can be refined as much as needed near discontinuities, yet be kept coarse far from them. The authors have reported a reduction

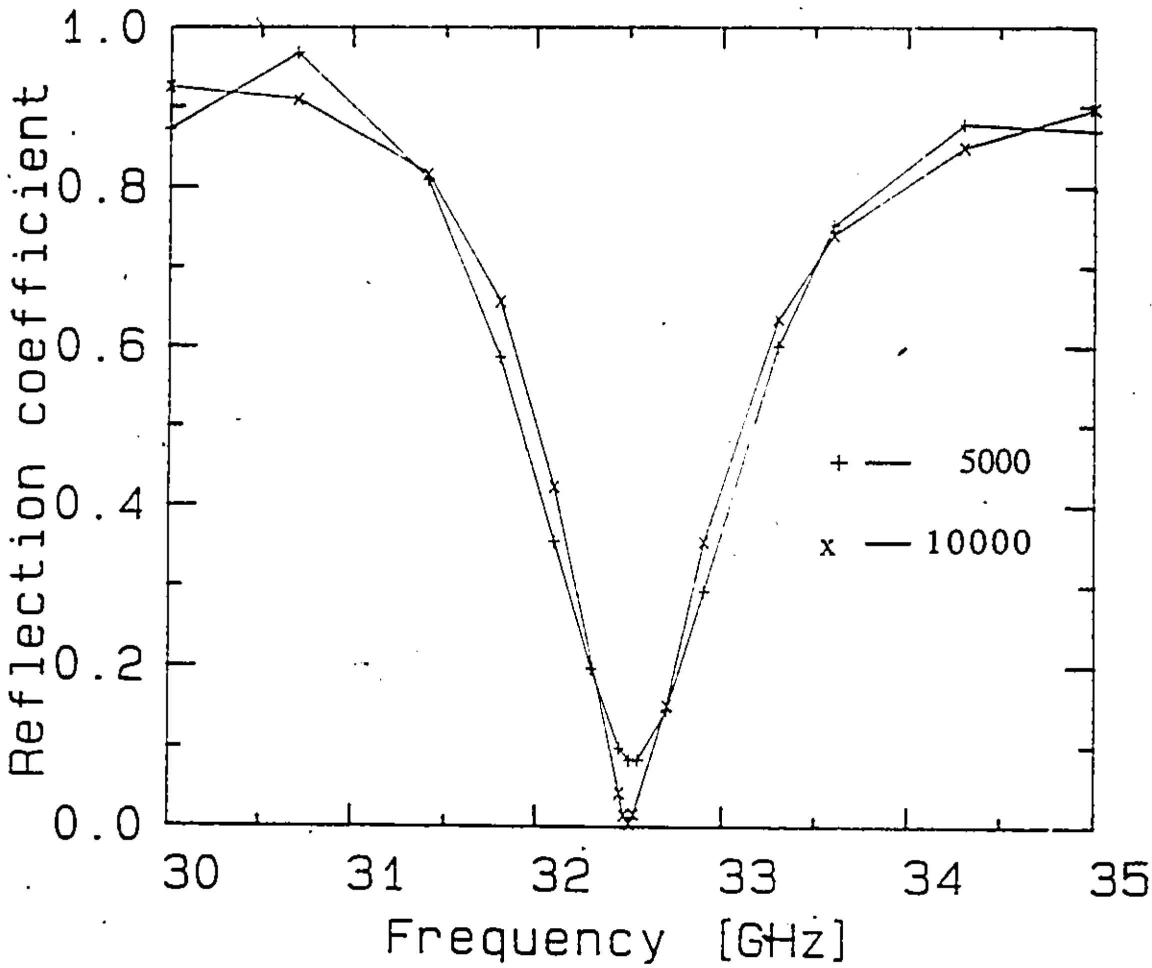


Figure 6.19: Reflection coefficient as function of frequency using TLM: + = 5000 iterations, x = 10000 iterations.

of processing time by approximately 4 times.

Of modern consideration is the availability of inexpensive CPU time on vectorizing supercomputers. A system of linear equations, such as that encountered in the FEM, can be solved using a CRAY. Indeed, those machines are adapted to solve matrix equation problems by vector processing, resulting in a reduction of several orders of magnitude in CPU time. Algorithms expressible as matrix operations are well suited for implementation on a vector computer. This is where the author's TLM implementation may have great advantages over [16]. Indeed, as shown in (5.4), all iterative steps can be represented as matrix operations. No such form could be found in Shih's approach.

Figures 6.3 and 6.4 show the E-field distribution at the input of the first septum. The field does not exactly vanish on the region of the septum because of a fundamental property of the TLM method: the nodes are not modeled on a boundary, as in the FEM, but at a distance $\Delta d/2$ from all boundaries. Although second degree interpolation was used to interpolate the value of the field at the metallic boundary using the field values at the nearest 3 points, the computed field value is not exactly 0, due to the strong variation of the fields near this obstacle.

Figure 6.9 and 6.10 show $\underline{H}_x$-field distribution as a function of X, a lateral view of the field, at the input to the first septum. The results obtained from TLM and FEM are nearly identical, although there is a distinct lack of resolution around the septum region which causes the current distribution on the lateral side of the septum to be difficult to obtain with any accuracy. One may argue that the contribution of the lateral sides of the septum to the total power dissipation by the septa is relatively small, and that lack of resolution is irrelevant if the value of interest is total power dissipation, not local current distribution and local power dissipation.

6.4 Conclusion.

The E-field and H-field distribution inside the filter were determined using FEM and TLM. From these, the peak power handling was directly calculated, and the current distribution on each septum was found, yielding the total power dissipation on each of these, as a function of frequency. The maximum peak power handling capability of the filter was well defined, but the maximum CW handling is still an open problem.

The FEM has had two distinct advantages over the TLM method, in the present thesis. The first technique is much faster, and offers arbitrary resolution of any part of the problem's geometry. This last advantage may be diminished by the use of Saguet's adaptive-mesh algorithm. The main advantage of the basic TLM method over the FEM method, in the present case, is algorithmic simplicity. The FEM would seem like the method to be used for future work on larger filter structures, as the TLM model would require an increasingly large number of initial iterations for increasing filter size, as well as an increase in time per iteration. Note that both methods yield very similar results: one is not more accurate than the other. Other methods, such as the method of moments, FDTD and various other old and new methods could be studied to solve this problem. An exhaustive study of all possibilities would be prohibitive.

Chapter 7

Conclusion

A description of the FEM method was given, as applied to the particular class of E-plane metal-insert filters described herein. The reflection coefficient, computed both by FEM and TLM, was shown to agree closely with results from the mode matching (MM) technique, which has been shown by measurements to be accurate. Since results obtained by MM (reflection coefficient as a function of frequency, and field values at the input of the input septum) agreed closely with those obtained by FEM and TLM, it was assumed that all other field values thus obtained, but not given by MM, were also correct. This includes distribution of E- and H-field inside the filters, as well as the current distribution on the septum's surface. The 2-insert filter is shown to sustain 13 times less power than a corresponding empty waveguide before dielectric (air) breakdown.

The determination of maximum CW power that an insert filter can handle involves several disciplines, including a thermodynamic analysis for the temperature distribution, the determination of the mechanical deformation of the septa under thermal stress, as well as a field distribution analysis, as presented in previous chapters. Furthermore, mechanical deformations necessitate the use of more complicated and time consuming 3-D analysis.

While the peak power limitation is set by the maximum field before an arc occurs, the limit on CW power is not very clearly defined. Changes, while the structure is heating up, occur gradually, and do not cause an abrupt catastrophic failure of the device, as in the previous case. The first phenomena that is expected to occur is the deformation of the septa due to thermal expansion. If the process continues, the septa might even reach the melting point (for "extremely high" power levels). After the septum is deformed, the characteristics of the filter mat change. For instance, the center frequency can be shifted, causing high reflections, and possibly severe damage to the source. What, then is the CW limit criterion? A few suggestions are: septa not exceeding a given critical temperature; band-rejection characteristics not falling below a determined transmission loss; bandwidth and center frequency not deviating by more than a given tolerance; or a combination of the above.

The use of supercomputers will make feasible the analysis of multi-septa filters using the FEM. The final goal of these studies is to obtain empirical formulas giving the maximum peak and CW power limitations within a given tolerance, given the geometric parameters of the filter. The maximum CW power capability of the insert filters remains an open problem.

References

[1] A. Wexler, "Computation of Electromagnetic Fields", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-17, no. 8, pp. 413-439, August 1969.

[2] G. Jeng and A. Wexler, "Self-Adjoint Variational Formulation of Problems Having Non-Self-Adjoint Operators", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-26, no. 2, pp. 91-94, February 1978.

[3] P. Silvester, "High-Order Polynomial Triangular Finite Elements for Potential Problems", *Int. J. Eng Sci.*, vol. 7, pp. 849-861. Permagon Press 1969.

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

[5] I. Stakgold, *Boundary Value Problems of Mathematical Physics, volume 2*, New York: The MacMillan Co., 1968.

[6] I. M. Gelfand and S. V. Fomin, *Calculus of variations*, Englewood Cliffs, N.J.: Prentice-Hall, 1963.

[7] R. Weinstock, *Calculus of Variations, with applications to physics and engineering*, New York, N. Y.: McGraw-Hill, 1952.

[8] P. Leroux, *Algebre Lineaire, une approche matricielle*, p. 289, Quebec: Modulo Editeur, 1983.

[9] W. Yue, *Peak Power-Handling Capacity of Finline Structures Evaluated by Numerical and Experimental Methods*, Ottawa, Ont.: M.A.Sc. thesis, 1986.

[10] W. H. Beyer, *CRC Standard Mathematical Tables, 27^{th} edition*, p. 45, Boca Raton, Florida: CRC Press, 1985.

[11] W. J. R. Hoefer, "The Transmission-Line Matrix Method - Theory and Applications", *IEEE Trans. Microwave Theory and Techniques*; vol. MTT-33, no. 10, pp. 882-893, October 1985.

[12] P. B. Johns and R. L. Beurle, "Numerical Solution of 2-Dimensional Scattering Problems Using a Transmission-Line Matrix", *Proc. IEE*, vol. 118, no. 9, pp. 1203-1208, September 1971.

[13] P. B. Johns, "Application of the Transmission-Line Matrix Method to Homogeneous Waveguides of Arbitrary Cross-Section", *Proc. IEE*, vol. 119, no. 8, pp. 1086-1091, August 1972.

[14] P. B. Johns, "The Solution of Inhomogeneous Waveguide Problems Using a Transmission-Line Matrix", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-22, no. 3, pp. 209-215, March 1974.

[15] P. Saguet and E. Pic, "Le Maillage Rectangulaire et le Changement de Maille Dans la Methode TLM en Deux Dimensions", *Electronics Letters*, 2^{nd} April, 1981, vol. 17, no. 7.

[16] Y. C. Shih, "The Analysis of Fin Lines Using Transmission Line Matrix and Transverse Resonance Methods", *M.A.Sc. thesis*, University of Ottawa, Ottawa, 1980.

[17] F. Arndt, J. Bornemann, D. Grauerholtz, R. Vahldieck, "Theory and Design of Low-Insertion Loss Fin-Line Filters", *IEEE Trans. Microwave Theory and Techniques*,

vol. MTT-30, no. 2, pp. 155-163, February 1982.

[18] R. Vahldieck, J. Bornemann, F. Arndt, D. Grauerholz, "Optimized Waveguide E-plane Metal Insert Filters for Millimeter-Wave Applications", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-31, no. 1, pp. 65-69, January 1983.

[19] R. Vahldieck, J. Bornemann, F. Arndt, D. Grauerholz, "W-Band Low-Insertion-Loss E-Plane Filter", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-32, no. 1, pp. 133-135, January 1984.

[20] F. Arndt, J. Bornemann, R. Vahldieck, D. Grauerholz, "E-Plane Integrated Circuit filters with Improved Stopband Attenuation", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-32, no. 10, pp. 1391-1394, October 1984.

[21] R. Vahldieck, W. J. R. Hoefer, "Finline and Metal Insert Filters with Improved Passband Separation and Increased Stopband Attenuation", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-33, no. 12, pp. 1333-1339, December 1985.

[22] Y.-C. Shih, T. Itoh, L. Q. Bui, "Computed-Aided Design of Millimeter-Wave E-Plane Filters", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-31, no. 2, pp. 135-142, February 1983.

[23] Y.-C. Shih, T. Itoh, "E-Plane Filters with Finite-Thickness Septa", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-31, no. 12, pp. 1009-1013, December 1983.

[24] Y. Konishi, K. Uenakada, "The Design of a Bandpass Filter with Inductive Strip - Planar Circuit Mounted in Waveguide", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-22, no. 10, pp. 869-873, October 1974.

[25] Y. Tajima, Y. Sawayama, "Design and Analysis of a Waveguide-Sandwich Microwave Filter", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-22, no. 9, pp. 839-841, September 1974.

[26] C. P. Riley, J. Simkin, C. W. Trowbridge, "Accuracy of Results From Finite Element Models", *Workshop on Electromagnetic Field Computation*, Proceedings, vol. 1, October 1986, Schenectady, N.Y.

[27] N. Anderson, A. M. Arthurs, "A variational principle for Maxwell's equations", *Int. J. Electronics*, 1978, vol. 45, no. 3, pp. 333-334.

[28] N. Anderson, A. M. Arthurs, "Complementary variational principles for Maxwell's equations", *Int. J. Electronics*, 1979, vol. 47, no. 3, pp. 229-236.

[29] N. Anderson, A. M. Arthurs, "Variational principles for Maxwell's equations *II*, *Int. J. Electronics*, 1981, vol. 51, no. 1, pp. 71-77.

[30] M. Koshiba, K. Hayata, M. Suzuki, "Finite-Element Formulation in Terms of the Electric-Field Vector for Electromagnetic Waveguide Problem", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-33, no. 10, pp. 900-905, October 1985.

[31] C. H. Chen, C.-D. Lien, "The Variational Principle for Non-Self-Adjoint Electromagnetic Problems", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-28, no. 8, pp. 878-886, August 1985.

[32] Z. J. Csendes, P. Silvester, "Numerical Solution of Dielectric Loaded Waveguides: I-Finite-Element Analysis", *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-18, no. 12, pp. 1124-1131, December 1970.

[33] R. E. Collins, "Foundations for microwave engineering", New York: McGraw Hill Book Company, 1966.

[34] P. Bhartia, I. J. Bahl, "Millimeter wave engineering and applications", New York: John Wiley and Sons, 1984.

Bibliography

I. Stakgold, *Green's Functions and Boundary Value Problems*, New York: John Wiley and Sons, 1979.

E. Kreyszig, *Introductory Functional Analysis*, New York: John Wiley and Sons, 1978.

P. R. Garabedian, *Partial Differential Equations*, New York: John Wiley and Sons, 1964.

Appendix.

The following computer program was used throughout this thesis for simulations with FEM using 3rd degree approximation polynomials. The subroutines have been built so that they can relatively easily be modified to any degree of approximation.

```
C  Main program for FEM implementation.

C  This FORTRAN program runs on VAX/VMS 4.7
C  Link with LINPACK routines ZSPCO and ZSPSL.

      PROGRAM filter

      CHARACTER*35       nom_fichier
      INTEGER            option
      DOUBLE PRECISION   cote_a, cote_b, frequence

C  This grid is specially made for 3rd degree approximation shape
C   functions.  Only the bottom half of the guide is modeled.

      PARAMETER dimX =          4
      PARAMETER dimZ =         18
      PARAMETER debut_septum1 =  6
      PARAMETER fin_septum1 =    7
      PARAMETER debut_septum2 = 13
      PARAMETER fin_septum2 =   14
      PARAMETER septum_bas =     2
      PARAMETER degre_approximation = 3
      PARAMETER da = degre_approximation
      PARAMETER noptl_tr = (da+1)*(da+2)/2
      PARAMETER noptl=(3*dimX+1)*(3*dimZ+1)
      PARAMETER notr=2*dimX*dimZ
      PARAMETER ndir=3*dimZ+1 +
     &    (3*septum_bas-2)*(3*(fin_septum1-debut_septum1)+1 +
     &     3*(fin_septum2-debut_septum2)+1)
      PARAMETER nn=noptl-ndir
      PARAMETER a_size=(nn*(nn+1))/2

      DOUBLE COMPLEX     pot(noptl), work(nn), a(a_size), b(nn)
      DOUBLE PRECISION   d, ls, x(noptl), x_grid(25),
     &                   z(noptl), z_grid(100)
      INTEGER*2          index(noptl), iv(notr,noptl_tr)
      INTEGER            i, j, pivot(nn), temp1, temp2
      LOGICAL*1          dirichlet(noptl)

C  Metallic septum (half-thickness is 50 microns).

      x_grid ( 1) =   3.556D-3
      x_grid ( 2) =   3.506D-3

C  Air in guide.

      x_grid ( 3) =   3.300D-3
      x_grid ( 4) =   1.900D-3

C  Metallic wall (C4) at bottom of model of guide.

      x_grid ( 5) =   0.000D-3

C  Input port (boundary C1).
```

```
      z_grid ( 1) =   0.000D-3
      d = z_grid(1)
      z_grid ( 2) =   2.500D-3
      z_grid ( 3) =   4.500D-3
      z_grid ( 4) =   6.500D-3
      z_grid ( 5) =   7.400D-3

C  Beginning of first septum (1mm long).

C  debut_septum1 = 6
      z_grid ( 6) =   7.500D-3
      z_grid ( 7) =   8.500D-3
C  fin_septum1 = 7

C  End of first septum.

      z_grid ( 8) =   8.600D-3
      z_grid ( 9) =   9.600D-3
      z_grid (10) =  10.650D-3
      z_grid (11) =  11.700D-3
      z_grid (12) =  12.700D-3

C  Beginning of seconf septum (1mm long).

C  debut_septum2 = 13
      z_grid (13) =  12.800D-3
      z_grid (14) =  13.800D-3
C  fin_septum2 = 14

C  End of second septum.

      z_grid (15) =  13.900D-3
      z_grid (16) =  14.800D-3
      z_grid (17) =  16.800D-3
      z_grid (18) =  18.800D-3
C  Output port (boundary C2).
      z_grid (19) =  21.300D-3
      ls = z_grid(19)


C  Definition of coordinates.

      templ = 3*dimZ + 1
C  Definition of Z-coordinates a vertex of triangles, on first row.
      DO i = 0, dimZ
         z(i*3+1) = z_grid(i+1)
      END DO
C  Definition of interpolated Z-coordinates, on first row.
      DO i = 0, dimZ-1
         z(i*3+2) = (z(i*3+1)*2.0D0+z((i+1)*3+1))/3.0D0
         z(i*3+3) = (z(i*3+1)+2.0D0*z((i+1)*3+1))/3.0D0
      END DO
C  Definition of other Z-coordinates in the mesh.
      DO i = 1, 3*dimX
         DO j = 1, templ
```

```
            z(j+i*templ) = z(j)
         END DO
      END DO
C  Definition of X-coordinates a vertex of triangles, on first column.
      DO i = 1, dimX+1
         x(1+(i-1)*3*templ) = x_grid(i)
      END DO
C  Definition of interpolated X-coordinates, on first column.
      DO i = 1, dimX
         temp2 = (i-1)*3*templ + 1
         x(temp2+templ) = (x(temp2+3*templ)+2.0D0*x(temp2))/3.0D0
         x(temp2+2*templ) = (2.0D0*x(temp2+3*templ)+x(temp2))/3.0D0
      END DO
C  Definition of other X-coordinates in the mesh.
      DO i = 2, templ
         DO j = 0, (3*dimX+1)-1
            x(i+j*templ) = x(1+j*templ)
         END DO
      END DO


C  Initialization of variables "dirichlet" and "pot".

      DO i = 1, noptl
         dirichlet(i) = .FALSE.
         pot(i) = (0.0D0,0.0D0)
      END DO


C  Definition of potentials on Dirichlet-type boundaries.

C  Metallic wall at base of guide.
      DO i = 1+3*dimX*templ, (3*dimX+1)*templ
         dirichlet(i) = .TRUE.
      END DO
C  Metallic septums.
      DO i = 1, 1+3*(septum_bas-1)
         DO j = 1+3*(debut_septum1-1), 1+3*(fin_septum1-1)
            dirichlet((i-1)*templ+j) = .TRUE.
         END DO
         DO j = 1+3*(debut_septum2-1), 1+3*(fin_septum2-1)
            dirichlet((i-1)*templ+j) = .TRUE.
         END DO
      END DO

      WRITE(*,*) 'Size of system to be solved = ',nn
      WRITE(*,*) 'Total number of nodes = ',noptl
      WRITE(*,*) 'Number of Dirichlet-type nodes = ',ndir
      WRITE(*,*) 'Number of triangles = ',notr
      CALL estimer_pages (nn, noptl, notr, a_size, noptl_tr)

C  Definition of some parameters used in this algorithm.
      cote_a = 7.112D-3
      cote_b = cote_a / 2.0D0
      nom_fichier = 'filter.out'
      option = 1
```

```
C  Generate mesh.
      CALL filter_grid (dimX, dimZ, iv, notr, option)

C  The simulation frequency is entered interactively:
      WRITE (*,'(1X,A,$)') 'Frequency:  '
      READ (*,*) frequence

      CALL filter_main (a, b, work, pivot, pot, index, dirichlet, x,
     &                  z, iv, nn, notr, noptl, x_grid, z_grid,
     &                  frequence, cote_a, cote_b, dimX, dimZ, ls, d,
     &                  ndir, nom_fichier, degre_approximation)
      STOP
      END

C***********************************************************************

C  This is the automatic gridding algorithm for 3rd degree shape functions.

C  The grid is symmetrical about the middle of the guide.
C   Hence, "dimZ" must be even.

C  Parameters:
C  "option" : 1 or 2 (see text).

      SUBROUTINE filter_grid (dimX, dimZ, iv, notr, option)

      INTEGER      notr
      INTEGER*2    iv(notr,1)
      INTEGER      degre_grille, dimX, dimZ, i, indice, j, k, m,
     &             option

C  Definition of triangles.
C  "i" = actual column (i[max] is the number of columns per row).
C  "j" = 0: left triangle of the square is being defined.
C        1: right triangle is being defined.
C  "k + 1" = actual row (k[max] is the number of rows minus 1).

C  Note:  the node order is such that node1, node2, node3
C   forms a counter-clockwise set, which is necessary for
C   Silvester's algorithm. (Ref: International journal of
C   Engineering Science, vol 7, pp 849-861, Permagon press, 1969)

C  Triangle node order, for option 1 (left side of median),
C   and option 2 (right side of median).

C       j=0                  j=1
C
C        3  6
C  1 *__*__*__* 10        1 *
C    |       /             /|
C    |  5   /             / |
C  2 *  *  * 9         2 *  * 3
C    |    /              /  |
C    |   /              /   |
C  4 *  * 8         4 *  *  * 6
```

```
C     | /                /    5   |
C     |/               /_________|
C  7  *            7 *__*__*__* 10
C                       8  9

C  Triangle node order, for option 2 (left side of median)
C   and option 1 (right side of median).

C      j=0                j=1
C
C                         3  6
C   1  *             1 *__*__*__* 10
C      |\                \      |
C      | \                \  5  |
C   2  *  * 3          2  *  *  * 9
C      |   \               \    |
C      |    \               \   |
C   4  *  *  * 6            4 *  * 8
C      |  5   \                \ |
C      |___ ___\                \|
C   7  *  *  *  * 10             * 7
C         8  9

      IF (MOD(dimZ,2) .NE. 0) THEN
        WRITE (*,*) 'Error in FILTGRID3:'
        WRITE (*,*) '   Parameter dimZ should be an even number.'
        STOP 'Execution aborted.'
      END IF

C  Left half of guide.
      DO i = 1, dimZ/2
        DO j = 0, 1
          DO k = 0, dimX-1
            IF (option .EQ. 1) THEN
              CALL filter_grid_method1 (dimX, dimZ, i, iv, notr, j, k)
            ELSE
              CALL filter_grid_method2 (dimX, dimZ, i, iv, notr, j, k)
            END IF
          END DO
        END DO
      END DO

C  Right half of guide.
      DO i = dimZ/2+1, dimZ
        DO j = 0, 1
          DO k = 0, dimX-1
            IF (option .EQ. 1) THEN
              CALL filter_grid_method2 (dimX, dimZ, i, iv, notr, j, k)
            ELSE
              CALL filter_grid_method1 (dimX, dimZ, i, iv, notr, j, k)
            END IF
          END DO
        END DO
      END DO

      RETURN
```

```
      END
C**********************************************************************

      SUBROUTINE filter_grid_method1 (dimX, dimZ, i, iv, notr, j, k)

      INTEGER      dimX, dimZ, i, indice, notr, j, k, temp1, temp2
      INTEGER*2    iv(notr,1)

      temp2 = 3*dimZ+1
      temp1 = i*3-2 + k*3*temp2
      indice = i + j*dimZ + k*2*dimZ
      iv(indice,1) = temp1 + 3*j
      iv(indice,2) = temp1 + temp2 + 2*j
      iv(indice,3) = temp1 + 1 + j*(temp2+2)
      iv(indice,4) = temp1 + 2*temp2 + j
      iv(indice,5) = temp1 + temp2 + 1 + j*(temp2+1)
      iv(indice,6) = temp1 + 2 + j*(2*temp2+1)
      iv(indice,7) = temp1 + 3*temp2
      iv(indice,8) = temp1 + 2*temp2 + 1 + j*temp2
      iv(indice,9) = temp1 + temp2 + 2 + 2*j*temp2
      iv(indice,10) = temp1 + 3 + j*3*temp2

      RETURN

      END
C**********************************************************************

      SUBROUTINE filter_grid_method2 (dimX, dimZ, i, iv, notr, j, k)

      INTEGER      dimX, dimZ, i, indice, notr, j, k, temp1, temp2
      INTEGER*2    iv(notr,1)

      temp2 = 3*dimZ+1
      temp1 = i*3-2 + k*3*temp2
      indice = i + j*dimZ + k*2*dimZ
      iv(indice,1) = temp1
      iv(indice,2) = temp1 + temp2 + j
      iv(indice,3) = temp1 + 1 + (1-j)*temp2
      iv(indice,4) = temp1 + 2*temp2 + 2*j
      iv(indice,5) = temp1 + 1 + j + (2-j)*temp2
      iv(indice,6) = temp1 + 2 + (1-j)*2*temp2
      iv(indice,7) = temp1 + 3*j + 3*temp2
      iv(indice,8) = temp1 + 3 + 2*temp2 + (1-j)*(temp2-2)
      iv(indice,9) = temp1 + 3 + temp2 + (1-j)*(2*temp2-1)
      iv(indice,10) = temp1 + 3 + (1-j)*3*temp2

      RETURN

      END
C**********************************************************************

C  This is the main routine of the FEM program.  It contains the majority
```

```
C   of the FEM algorithm.

      SUBROUTINE filter_main (a, b, work, pivot, pot, index,
     &                         dirichlet, x, z, iv, nn, notr,
     &                         noptl, x_grid, z_grid, frequence,
     &                         cote_a, cote_b, dimX, dimZ, ls, d,
     &                         ndir, nom_fichier, degre_grille)

      INTEGER           nn
      DOUBLE COMPLEX    a(1), b(1), pot(1), work(1)
      DOUBLE PRECISION  x(1), z(1)
      INTEGER           dimX, dimZ, ndir, noptl, notr,
     &                  info, pivot(1)
      INTEGER*2         index(1), iv(notr,1)
      LOGICAL*1         dirichlet(1)


      CHARACTER*35      nom_fichier
      DOUBLE COMPLEX    djay, e0
      DOUBLE PRECISION  cote_a, cote_b, d, epsilon_zero, frequence,
     &                  k_carre, k1x, k1z, ls, mu_zero, omega, pi,
     &                  rcond, surface, t1, x_grid(1), z_grid(1)
      INTEGER           degre_grille, i, j, k, i1, i2, i3, job,
     &                  position

      COMMON /constantes/ pi, djay
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface,
     &                    t1, e0

C   Definition of physical and mathematical constants.

      pi = DACOS(-1.0D0)
      epsilon_zero = 8.854187818D-12
      mu_zero = 4.0D-07 * pi
      djay = (0.0D0, 1.0D0)

C   Definition of parameters describing the problem.

      omega         = 2.0D0 * pi * frequence
      k_carre       = omega * omega * mu_zero * epsilon_zero
      k1x           = pi / cote_a
      k1z           = DSQRT(k_carre - k1x**2)
      surface       = cote_a * cote_b
      e0            = DSQRT (4.0D0 * omega * mu_zero /
     &                  (k1z * surface))

      CALL ordonner_noeuds (dirichlet, index, noptl)
      CALL effacer_a_et_b (a, b, nn)
      WRITE(*,*) 'Initialization completed.'

C   Insert contributions of S and T matrix into final system.
      CALL contribution_S_et_T (iv, notr, x, z, dirichlet,
     &                          a, b, k_carre, pot,
     &                          degre_grille, ndir, index)
      WRITE(*,*) 'Contribution from S and T completed.'
```

```
C  Insert contributions from input boundary C1 into final system of equatio
      CALL limite_C1 (dimZ, dimX, ndir, index,
     &                  dirichlet, x, pot, a, b, d, degre_grille)
      WRITE(*,*) 'Contribution from boundary C1 completed.'

C  Insert contributions from output boundary C2 into final system of equati
      CALL limite_C2 (dimZ, dimX, ndir, index,
     &                  dirichlet, x, pot, a, b, degre_grille)
      WRITE(*,*) 'Contribution from boundary C2 completed.'

      WRITE(*,*) 'Final system of equations completed.'
      WRITE(*,*) 'Working hard on solution, now.'

C  Solve system of equations.
C  At this point, any program can be used, including direct
C   techniques (Gaussian reduction), and iterative techniques (conjugate
C   gradient).

C  Factor packed matrix "a" by Gaussian elimination.
C  ZSPCO is a LINPACK program.
      CALL zspco (a, nn, pivot, rcond, work)
      IF (rcond .EQ. 0.0D0) THEN
        WRITE (*,*) 'System is singular.'
        STOP
      ELSE
        rcond = 1.0D0/rcond
        WRITE (*,1001) rcond
1001  FORMAT (' Estimated condition number of matrix A is ',1PE13.5)
      END IF

C  Solve packed system [a][x] = [b].
C  ZSPSL is a LINPACK program.
      CALL zspsl (a, nn, pivot, b)

C  Array "b" now contains the result.  Insert these into "pot", the value
C   of the Ey-field at the discrete nodes.

      DO i = 1, nn
         pot(index(i + ndir)) = b(i)
      END DO

C  Create output file with results.
      CALL sortie (pot, nn, ndir, nom_fichier,
     &             cote_a, cote_b, frequence, d, ls, dimX,
     &             dimZ, rcond, noptl, dirichlet, z, x)

      RETURN
      END

C*********************************************************************

C  This routine estimates the number of pages of memory used by
C   the arrays of the program.
C  A page is 512 bytes.
```

```
      SUBROUTINE estimer_pages (nn, noptl, notr, ia_size, noptl_tr)

      INTEGER   nn, noptl, pages, notr, ia_size, noptl_tr

      pages = 16*(noptl + ia_size + 2*nn)
      pages = pages + 8*(2*noptl + 125)
      pages = pages + 4*(nn)
      pages = pages + 2*(noptl + notr*noptl_tr)
      pages = pages + 1*(noptl)
      WRITE(*,*) 'Number of pages required for arrays = ',
     &            1 + pages / 512
      RETURN
      END

C***********************************************************************

C  This function returns the position of "element" in the list "liste".
C   Note that "element" MUST be in "liste", or else the calling program
C   will crash.

      FUNCTION position (liste, element)

      INTEGER*2  element, liste(1)
      INTEGER    pointeur, position, i

      DO i = 1, 10000
         pointeur = i
         IF (liste(pointeur) .EQ. element) GOTO 2
      END DO
2     CONTINUE
      position = pointeur
      RETURN
      END

C***********************************************************************

C  This function transforms a COMPLEX number Z = (X + jY) into another
C   one W = (R + jTHETA), where R is the modulus of Z, and THETA is
C   the argument of Z, in degrees.

      FUNCTION rect_a_polaire (valeur)

      DOUBLE COMPLEX    valeur, rect_a_polaire, buffc, djay
      DOUBLE PRECISION  buffr(2), pi
      EQUIVALENCE       (buffr, buffc)
      COMMON /constantes/ pi, djay

      IF(valeur .EQ. (0.0D0,0.0D0)) THEN
         rect_a_polaire = (0.0D0,0.0D0)
      ELSE
         buffc = valeur
         rect_a_polaire = DCMPLX(CDABS(valeur),
     &       DATAN2(buffr(2),buffr(1)) * 180.0D0 / pi)
      END IF
      RETURN
      END
```

```
C***********************************************************************

C  Node order is re-arranged so that Dirichlet nodes appear first
C   on the list of nodes.  Order is re-arranged through
C   pointer variable "index".

      SUBROUTINE ordonner_noeuds (dirichlet, index, noptl)

      INTEGER*2  index(1), temporaire
      INTEGER    i, pointeur, noptl
      LOGICAL*1  dirichlet(1)

      DO i = 1, noptl
         index(i) = i
      END DO

      pointeur = 1
      DO i = 1, noptl
         IF (dirichlet(i)) THEN
            temporaire = index(i)
            index(i) = index(pointeur)
            index(pointeur) = temporaire
            pointeur = pointeur + 1
         END IF
      END DO
      RETURN
      END

C***********************************************************************

C  Initialize matrix "a" and vector "b".

      SUBROUTINE effacer_a_et_b (a, b, nn)

      INTEGER          a_size, i, nn
      DOUBLE COMPLEX   a(1), b(1)

      DO i = 1, nn
        b(i) = (0.0D0, 0.0D0)
      END DO
      a_size = (nn*(nn+1))/2
      DO i = 1, a_size
        a(i) = (0.0D0, 0.0D0)
      END DO
      RETURN
      END

C***********************************************************************

C  This routine creates the output file with results.

      SUBROUTINE sortie (pot, nn, ndir, nom_fichier,
     &                   cote_a, cote_b, frequence, d, ls, dimX,
     &                   dimZ, rcond, noptl, dirichlet, z, x)
```

```fortran
      CHARACTER*35        nom_fichier
      DOUBLE COMPLEX      pot(1), rect_a_polaire
      INTEGER             nn, ndir, dimX, dimZ, noptl, i
      LOGICAL*1           dirichlet(1)
      DOUBLE PRECISION    cote_a, cote_b, frequence, d, ls, rcond,
     &                    z(1), x(1)

      OPEN (1, FILE = nom_fichier, STATUS = 'NEW',
     &      CARRIAGECONTROL = 'LIST')

C  Give list of variable parameters.

      WRITE (1,1) cote_a, cote_b, (frequence / 1.0D9), d, ls,
     &            dimX, dimZ, rcond'

C  All Dirichlet nodes have "***" written at the end of the line.
      DO i = 1, noptl
         IF (dirichlet(i)) THEN
            WRITE(1,4) z(i), x(i),
     &         rect_a_polaire(pot(i))
         ELSE
            WRITE(1,3) z(i), x(i),
     &         rect_a_polaire(pot(i))
         END IF
      END DO

      CLOSE (1)
      RETURN

1     FORMAT('Variable parameter list:'//
     &       ' a = waveguide width = ', 3PF8.3, ' mm'/
     &       ' b = waveguide thickness = ', 3PF8.3, ' mm'/
     &       ' frequency = ', 0PF10.4, ' Ghz'/
     &       ' d = ', 3PF8.3,' mm'/' ls = ', 3PF8.3, ' mm'/
     &       ' dimX = ', 0PI3/' dimZ = ', I3//
     &       ' Estimated matrix condition number = ', 1PE10.3//
     &       ' X and Y coordinates are in mm.'//
     &       ' Z          X             reconstructed phi'/)
3     FORMAT(3PF7.3, 1X, F7.3, 1X, 1PD11.4, 1X, 0PF7.2)
4     FORMAT(3PF7.3, 1X, F7.3, 1X, 1PD11.4, 1X, 0PF7.2, '  ***')

      END

C*************************************************************************

C  This function returns the value of the cotangent of the angle
C   formed by the segments 1-3 and 2-3, where i-j indicates
C   the segment between coordinates (zi,xi) and (zj,xj).

      FUNCTION cotan (z1, x1, z2, x2, z3, x3)

      DOUBLE PRECISION  x1, z1, x2, z2, x3, z3, cotan, a2, b2, c2, M2

      a2 = (x2-x3)**2 + (z2-z3)**2
      b2 = (x1-x3)**2 + (z1-z3)**2
      c2 = (x1-x2)**2 + (z1-z2)**2
```

```
      M2 = (b2+c2-a2)**2 / (4.0D0*b2*c2)
      IF (M2 .LT. 1.75D-7) THEN
         cotan = 0.0D0
      ELSE
         cotan = 1.0D0 / DSQRT((1.0D0/M2) - 1.0D0)
      END IF
      RETURN
      END

C*********************************************************************

C  Function used in INTERIEUR

      FUNCTION ordre_S (ptr1, ptr2, n, dimension_S)

      INTEGER   dimension_S, ordre_S
      INTEGER   n, ptr1(3,dimension_S), ptr2(3,dimension_S), i
      DO i = 1, dimension_S
         IF (ptr1(1,i).EQ.ptr2(1,n) .AND.
     &       ptr1(2,i).EQ.ptr2(2,n) .AND.
     &       ptr1(3,i).EQ.ptr2(3,n)) THEN
           ordre_S = i
           RETURN
         END IF
      END DO
      STOP 'Error in "ordre_S".'
      END

C*********************************************************************

C  Indefinite integral of sin(k1x*x)

      FUNCTION x0sin (x)

      DOUBLE COMPLEX     e0
      DOUBLE PRECISION   omega, k_carre, k1x, k1z, surface, t1, x0sin, x
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface, t1, e0

      x0sin = -DCOS(k1x*x)/k1x
      RETURN
      END

C*********************************************************************

C  Indefinite integral of x*sin(k1x*x)

      FUNCTION x1sin (x)

      DOUBLE COMPLEX     e0
      DOUBLE PRECISION   omega, k_carre, k1x, k1z, surface, t1, x1sin, x
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface, t1, e0

      x1sin = (DSIN(k1x*x)/k1x - x*DCOS(k1x*x))/k1x
      RETURN
      END
```

```
C***********************************************************************

C  Indefinite integral of (x**2)*sin(k1x*x)

      FUNCTION x2sin (x)

      DOUBLE COMPLEX    e0
      DOUBLE PRECISION  omega, k_carre, k1x, k1z, surface, t1, x2sin, x
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface, t1, e0

      x2sin = (2.0D0*x*DSIN(k1x*x) + (2.0D0/k1x-x*x*k1x)*
     &         DCOS(k1x*x))/k1x/k1x
      RETURN
      END

C***********************************************************************

C  Indefinite integral of (x**3)*sin(k1x*x)

      FUNCTION x3sin (x)

      DOUBLE COMPLEX    e0
      DOUBLE PRECISION  omega, k_carre, k1x, k1z, surface, t1, x3sin, x
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface, t1, e0

      x3sin = (6.0D0/k1x/k1x-x*x)*x/k1x*DCOS(k1x*x) +
     &   3.0D0/k1x/k1x*DSIN(k1x*x)*(x*x-2.0D0/k1x/k1x)
      RETURN
      END

C***********************************************************************

C  Contributions from triangles, 3rd degree shape functions.

      SUBROUTINE contribution_S_et_T (iv, notr, x, z, dirichlet,
     &                                a, b, k_carre, pot,
     &                                da, ndir, index)

      INTEGER           notr
      DOUBLE COMPLEX    a(1), b(1), pot(1)
      DOUBLE PRECISION  aire_triangle, cotan, det, det3, diviseur_S,
     &                  diviseur_T, k_carre, matrice_S, matrice_T,
     &                  Q1(10,10), Q2(10,10), Q3(10,10), T(10,10),
     &                  x(1), z(1), z1, z2, z3, x1, x2, x3
      INTEGER*2         iv(notr,1), index(1), noeud
      INTEGER           da, degre_approximation, dimension_S, i, i1,
     &                  i2, i3, index1, index2, inx, j, k, itr, l,
     &                  lc, n, ndir, ordre_S, position,
     &                  pointeur_S(3,10), pointeur_S_2(3,10), temp
      LOGICAL*1         dirichlet(1)

C  Data for matrices Q1 and T is input column by column, first from top
C   to bottom, then from left to right.

      DATA Q1 /11*0.0D0, 135.0D0, -135.0D0, -27.0D0, 0.0D0, 27.0D0,
     &  3.0D0, 2*0.0D0, -3.0D0, 0.0D0, -135.0D0, 135.0D0, 27.0D0,
```

```
     &  0.0D0, -27.0D0, -3.0D0, 2*0.0D0, 3.0D0, 0.0D0, -27.0D0, 27.0D0,
     &  135.0D0, -162.0D0, 27.0D0, 3.0D0, 2*0.0D0, -3.0D0,3*0.0D0,
     & -162.0D0, 324.0D0, -162.0D0, 5*0.0D0, 27.0D0, -27.0D0, 27.0D0,
     & -162.0D0, 135.0D0, -3.0D0, 2*0.0D0, 3.0D0, 0.0D0, 3.0D0, -3.0D0,
     & 3.0D0, 0.0D0, -3.0D0, 34.0D0, -54.0D0, 27.0D0, -7.0D0, 6*0.0D0,
     & -54.0D0, 135.0D0, -108.0D0, 27.0D0, 6*0.0D0, 27.0D0, -108.0D0,
     & 135.0D0, -54.0D0, 0.0D0, -3.0D0, 3.0D0, -3.0D0, 0.0D0, 3.0D0,
     & -7.0D0, 27.0D0, -54.0D0, 34.0D0/,
     &     diviseur_S /80.0D0/,
     &     T /76.0D0, 2*18.0D0, 0.0D0, 36.0D0, 0.0D0, 11.0D0, 2*27.0D0,
     & 11.0D0, 18.0D0, 540.0D0, 270.0D0, -189.0D0, 162.0D0, -135.0D0,
     & 0.0D0, -135.0D0, -54.0D0, 27.0D0, 18.0D0, 270.0D0, 540.0D0,
     & -135.0D0, 162.0D0, -189.0D0, 27.0D0, -54.0D0, -135.0D0, 0.0D0,
     & 0.0D0, -189.0D0, -135.0D0, 540.0D0, 162.0D0, -54.0D0, 18.0D0,
     & 270.0D0, -135.0D0, 27.0D0, 36.0D0, 3*162.0D0, 1944.0D0, 162.0D0,
     & 36.0D0, 2*162.0D0, 36.0D0, 0.0D0, -135.0D0, -189.0D0, -54.0D0,
     & 162.0D0, 540.0D0, 27.0D0, -135.0D0, 270.0D0, 18.0D0, 11.0D0,
     & 0.0D0, 27.0D0, 18.0D0, 36.0D0, 27.0D0, 76.0D0, 18.0D0, 0.0D0,
     & 11.0D0, 27.0D0, -135.0D0, -54.0D0, 270.0D0, 162.0D0, -135.0D0,
     & 18.0D0, 540.0D0, -189.0D0, 0.0D0, 27.0D0, -54.0D0, 2*-135.0D0,
     & 162.0D0, 270.0D0, 0.0D0, -189.0D0, 540.0D0, 18.0D0, 11.0D0,
     & 27.0D0, 0.0D0, 27.0D0, 36.0D0, 18.0D0, 11.0D0, 0.0D0, 18.0D0,
     & 76.0D0/,
     &     diviseur_T /6720.0D0/,
     &     degre_approximation /3/,
     &     pointeur_S /3,0,0,2,1,0,2,0,1,1,2,0,1,1,1,
     & 1,0,2,0,3,0,0,2,1,0,1,2,0,0,3/

      IF (da .NE. degre_approximation) THEN
        WRITE (*,*) 'Message from CONTRIBUTION_S_ET_T:'
        WRITE (*,1000)
1000  FORMAT ('   Degree of approximation does not ',
     &        'match calling routine.')
        STOP ' '
      END IF

      dimension_S = (degre_approximation + 1) *
     &              (degre_approximation + 2) / 2

C  Preparation of matrices S and T.

      DO i = 1, dimension_S
        DO j = 1, dimension_S
          T(i,j) = T(i,j) / diviseur_T
          Q1(i,j) = Q1(i,j) / diviseur_S
        END DO
      END DO

      DO i = 1, dimension_S
        DO j = 1, 3
          pointeur_S_2(j,i) = pointeur_S(j,i)
        END DO
      END DO

      k = dimension_S
```

```
      DO i = 1, dimension_S
        temp = pointeur_S_2(1,i)
        pointeur_S_2(1,i) = pointeur_S_2(2,i)
        pointeur_S_2(2,i) = pointeur_S_2(3,i)
        pointeur_S_2(3,i) = temp
      END DO
      DO i = 1, dimension_S
        index1 = ordre_S(pointeur_S, pointeur_S_2, i, k)
        DO j = 1, dimension_S
          index2 = ordre_S(pointeur_S, pointeur_S_2, j, k)
          Q2(i,j) = Q1(index1,index2)
        END DO
      END DO

      DO i = 1, dimension_S
        temp = pointeur_S_2(1,i)
        pointeur_S_2(1,i) = pointeur_S_2(2,i)
        pointeur_S_2(2,i) = pointeur_S_2(3,i)
        pointeur_S_2(3,i) = temp
      END DO
      DO i = 1, dimension_S
        index1 = ordre_S(pointeur_S, pointeur_S_2, i, k)
        DO j = 1, dimension_S
          index2 = ordre_S(pointeur_S, pointeur_S_2, j, k)
          Q3(i,j) = Q1(index1,index2)
        END DO
      END DO

      n = degre_approximation

C   Main loop.

      DO itr = 1, notr

        i1 = iv(itr,1)
        i2 = iv(itr,(n+1)*(n+2)/2-n)
        i3 = iv(itr,(n+1)*(n+2)/2)

        z1 = z(i1)
        z2 = z(i2)
        z3 = z(i3)
        x1 = x(i1)
        x2 = x(i2)
        x3 = x(i3)

        det = det3(1.0D0,z1,x1,1.0D0,z2,x2,1.0D0,z3,x3)

        IF(det .EQ. 0.0D0) THEN
          WRITE(1,5) itr
          WRITE(*,5) itr
          STOP
        END IF

        aire_triangle = DABS(det) / 2.0D0

C   Construction of sub-matrix of dimension dimension_S by dimension_S.
```

```
C  "i" = column number.

        DO i = 1, dimension_S
          noeud = iv(itr,i)
          IF (.NOT. dirichlet(noeud)) THEN
            l = position (index, noeud) - ndir

C  Construction of a row of the sub-matrix.

            DO k = 1, dimension_S
              matrice_T = - k_carre * aire_triangle *
     &             2.0D0 * T(k,i)
              matrice_S = (Q1(k,i)*cotan(x1,z1,x2,z2,x3,z3) +
     &                     Q2(k,i)*cotan(x2,z2,x3,z3,x1,z1) +
     &                     Q3(k,i)*cotan(x3,z3,x1,z1,x2,z2)) * 2.0D0
              noeud = iv(itr,k)
              IF (.NOT. dirichlet(noeud)) THEN
                lc = position(index,noeud) - ndir
                IF (l .LE. lc) THEN
                  inx = l + (lc*(lc-1))/2
                  a(inx) = a(inx) + (matrice_S + matrice_T)
                END IF
              ELSE
                  b(l) = b(l) - pot(noeud) *
     &              (matrice_S + matrice_T)
              END IF
            END DO
          END IF
        END DO

      END DO
      RETURN
5     FORMAT(' Determinant of a triangular element is zero.'/
     &       ' Iteration = ', I3)
      END

C*************************************************************************

C  This routine inserts contributions to matrix "a" and vector "b",
C   from boundary linear elements on C1 (input port).
C  Integration is performed counter-clockwise, therefore,
C   from top to bottom of C1.

      SUBROUTINE limite_C1 (dimZ, dimX, ndir, index,
     &                      dirichlet, x, pot, a, b, d, da)

      DOUBLE COMPLEX      a(1), b(1), const_alpha, const_beta,
     &                    djay, e0, matrice_beta, pot(1)
      DOUBLE PRECISION    beta(4,4), d, det, det3, det4, diviseur_beta,
     &                    factx0, factx1, factx2, factx3,
     &                    k_carre, k1x, k1z, omega, pi, surface, t1,
     &                    x(1), xi, xi2, xi3, xj, xj2, xj3, xm, xm2, xm3,
     &                    xn, xn2, xn3, x0sin, x1sin, x2sin, x3sin
      INTEGER             da, degre_approximation, dimZ, dimX,
     &                    g, i, inx, itr, k, l, lc, ndir, noeud, position
      INTEGER*2           index(1)
```

```
      LOGICAL*1         dirichlet(1)

      DATA   beta/128.0D0, 99.0D0, -36.0D0, 19.0D0, 99.0D0,
     &   648.0D0, -81.0D0, 2*-36.0D0, -81.0D0, 648.0D0, 99.0D0,
     &   19.0D0, -36.0D0, 99.0D0, 128.0D0/,
     &        diviseur_beta/1680.0D0/,
     &        degre_approximation/3/
      COMMON /constantes/ pi, djay
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface,
     &                    t1, e0

      IF (da .NE. degre_approximation) THEN
        WRITE (*,*) 'Message from LIMITE_C1:'
        WRITE (*,1000)
1000  FORMAT ('   Degree of approximation does not ',
     &        'match calling routine.')
        STOP ' '
      END IF

      g = degre_approximation
      DO itr = 1, 1+(g*dimX-g)*(g*dimZ+1), g*(g*dimZ+1)
        xj = x(itr+g*(g*dimZ+1))
        xn = x(itr+(g-1)*(g*dimZ+1))
        xm = x(itr+(g-2)*(g*dimZ+1))
        xi = x(itr)
C  xj is the lowest X, while xi is the highest X.
C  The integral is from top to bottom, on C1.
C  Therefore, the integral is performed counter-clockwise.
        xi2 = xi*xi
        xm2 = xm*xm
        xn2 = xn*xn
        xj2 = xj*xj
        xi3 = xi2*xi
        xm3 = xm2*xm
        xn3 = xn2*xn
        xj3 = xj2*xj
        det = det4(1.0D0,xi,xi2,xi3,1.0D0,xm,xm2,xm3,
     &             1.0D0,xn,xn2,xn3,1.0D0,xj,xj2,xj3)
        const_alpha = djay*k1z*4.0D0*e0*CDEXP(-djay*k1z*d)/det
        const_beta = -djay*2.0D0*k1z*(xj-xi)/diviseur_beta
C  i = 1: contribution of derivative w/r to phi(i)
C  i = 2: contribution of derivative w/r to phi(m)
C  i = 3: contribution of derivative w/r to phi(n)
C  i = 4: contribution of derivative w/r to phi(j)
        DO i = 1, g+1
          noeud = itr + (i-1)*(g*dimZ+1)
          IF (.NOT. dirichlet(noeud)) THEN
            l = position (index, noeud) - ndir

C  Prepare contribution to vector "b".

            IF (i .EQ. 1) THEN
C  factx0 = A, in the following expression:
C  factx1 = B, "
C  factx2 = C, "
C  factx3 = D, "
```

```
C  beta(i) = A + Bx + Cx**2 + Dx**3
              factx0 = det3(xm,xm2,xm3,xn,xn2,xn3,xj,xj2,xj3)
              factx1 = -det3(1.0D0,xm2,xm3,1.0D0,xn2,xn3,1.0D0,xj2,
     &                 xj3)
              factx2 = det3(1.0D0,xm,xm3,1.0D0,xn,xn3,1.0D0,xj,xj3)
              factx3 = -det3(1.0D0,xm,xm2,1.0D0,xn,xn2,1.0D0,xj,xj2)
            ELSE IF (i .EQ. 2) THEN
C  factx0 = E, in the following expression:
C  factx1 = F, "
C  factx2 = G, "
C  factx3 = H, "
C  beta(m) = E + Fx + Gx**2 + Hx**3
              factx0 = -det3(xi,xi2,xi3,xn,xn2,xn3,xj,xj2,xj3)
              factx1 = det3(1.0D0,xi2,xi3,1.0D0,xn2,xn3,1.0D0,xj2,
     &                 xj3)
              factx2 = -det3(1.0D0,xi,xi3,1.0D0,xn,xn3,1.0D0,xj,xj3)
              factx3 = det3(1.0D0,xi,xi2,1.0D0,xn,xn2,1.0D0,xj,xj2)
            ELSE IF (i .EQ. 3) THEN
C  factx0 = K, in the following expression:
C  factx1 = L, "
C  factx2 = M, "
C  factx3 = N, "
C  beta(n) = K + Lx + Mx**2 + Nx**3
              factx0 = det3(xi,xi2,xi3,xm,xm2,xm3,xj,xj2,xj3)
              factx1 = -det3(1.0D0,xi2,xi3,1.0D0,xm2,xm3,1.0D0,xj2,
     &                 xj3)
              factx2 = det3(1.0D0,xi,xi3,1.0D0,xm,xm3,1.0D0,xj,xj3)
              factx3 = -det3(1.0D0,xi,xi2,1.0D0,xm,xm2,1.0D0,xj,xj2)
            ELSE
C  factx0 = P, in the following expression:
C  factx1 = Q, "
C  factx2 = R, "
C  factx3 = V, "
C  beta(j) = P + Qx + Rx**2 + Vx**3
              factx0 = -det3(xi,xi2,xi3,xm,xm2,xm3,xn,xn2,xn3)
              factx1 = det3(1.0D0,xi2,xi3,1.0D0,xm2,xm3,1.0D0,xn2,
     &                  xn3)
              factx2 = -det3(1.0D0,xi,xi3,1.0D0,xm,xm3,1.0D0,xn,xn3)
              factx3 = det3(1.0D0,xi,xi2,1.0D0,xm,xm2,1.0D0,xn,xn2)
            END IF

C  This line inserts contribution to vector "b".

            b(1) = b(1) - const_alpha*(
     &                factx0*(x0sin(xj)-x0sin(xi)) +
     &                factx1*(x1sin(xj)-x1sin(xi)) +
     &                factx2*(x2sin(xj)-x2sin(xi)) +
     &                factx3*(x3sin(xj)-x3sin(xi)))

C  This next block inserts contribution to matrix "a".

C  k = 1: contribution of coefficient of phi(i)
C  k = 2: contribution of coefficient of phi(m)
C  k = 3: contribution of coefficient of phi(n)
C  k = 4: contribution of coefficient of phi(j)
            DO k = 1, g+1
```

```
            matrice_beta = const_beta*beta(i,k)
            noeud = itr + (k-1)*(g*dimZ+1)
            IF (.NOT. dirichlet(noeud)) THEN
              lc = position (index, noeud) - ndir
              IF (l .LE. lc) THEN
                inx = l + (lc*(lc-1))/2
                a(inx) = a(inx) + matrice_beta
              END IF
            ELSE
              b(l) = b(l) - pot(noeud) * matrice_beta
            END IF
          END DO
        END IF
      END DO
    END DO

    RETURN
    END

C*************************************************************************

C  This routine inserts contributions to matrix "a" and vector "b",
C   from boundary linear elements on C2 (input port).
C  Integration is performed counter-clockwise, therefore,
C   from bottom to top of C2.

      SUBROUTINE limite_C2 (dimZ, dimX, ndir, index,
     &                      dirichlet, x, pot, a, b, da)

      DOUBLE COMPLEX      a(1), b(1), const_beta, djay, e0,
     &                    matrice_beta, pot(1)
      DOUBLE PRECISION    beta(4,4), det2, diviseur_beta, k_carre,
     &                    k1z, k1x, omega, pi, surface, t1, xi, xj, x(1)
      INTEGER             da, degre_approximation,  dimZ, dimX, g, itr,
     &                    i, inx, k, l, lc, ndir, noeud, position
      INTEGER*2           index(1)
      LOGICAL*1           dirichlet(1)

      DATA   beta/128.0D0, 99.0D0, -36.0D0, 19.0D0, 99.0D0,
     &   648.0D0, -81.0D0, 2*-36.0D0, -81.0D0, 648.0D0, 99.0D0,
     &   19.0D0, -36.0D0, 99.0D0, 128.0D0/,
     &       diviseur_beta/1680.0D0/,
     &       degre_approximation/3/
      COMMON /constantes/ pi, djay
      COMMON /parametres/ omega, k_carre, k1x, k1z, surface,
     &                    t1, e0

      IF (da .NE. degre_approximation) THEN
        WRITE (*,*) 'Message from LIMITE_C2:'
        WRITE (*,1000)
1000  FORMAT (' Degree of approximation does not ',
     &        'match calling routine.')
        STOP ' '
      END IF
      g = degre_approximation
```

```
      DO itr = (g*dimX+1-g)*(g*dimZ+1), g*dimZ+1, -g*(g*dimZ+1)
C  xi is the lowest X, while xj is the highest X.
C  The integral is from bottom to top, on C2.
C  Therefore, the integral is performed counter-clockwise.
        xj = x(itr)
        xi = x(itr+g*(g*dimZ+1))
        const_beta = djay*2.0D0*k1z*(xj-xi)/diviseur_beta
C  i = 1: contribution of derivative w/r to phi(i)
C  i = 2: contribution of derivative w/r to phi(m)
C  i = 3: contribution of derivative w/r to phi(n)
C  i = 4: contribution of derivative w/r to phi(j)
        DO i = 1, g+1
          noeud = itr + (g+1-i)*(g*dimZ+1)
          IF (.NOT. dirichlet(noeud)) THEN
            l = position (index, noeud) - ndir
C  k = 1: contribution of coefficient of phi(i)
C  k = 2: contribution of coefficient of phi(m)
C  k = 3: contribution of coefficient of phi(n)
C  k = 4: contribution of coefficient of phi(j)
            DO k = 1, g+1
              matrice_beta = const_beta*beta(i,k)
              noeud = itr + (g+1-k)*(g*dimZ+1)
              IF (.NOT. dirichlet(noeud)) THEN
                lc = position (index, noeud) - ndir
                IF (l .LE. lc) THEN
                  inx = l + (lc*(lc-1))/2
                  a(inx) = a(inx) + matrice_beta
                END IF
              ELSE
                b(l) = b(l) - pot(noeud)*matrice_beta
              END IF
            END DO
          END IF
        END DO
      END DO

      RETURN
      END

C**********************************************************************

C  Determinant of a 3 by 3 matrix.

      FUNCTION det3(a11,a12,a13,a21,a22,a23,a31,a32,a33)

      IMPLICIT DOUBLE PRECISION   (A-Z)
      DOUBLE PRECISION   det3

      det3 = a11*(a22*a33-a32*a23)
     &       - a21*(a12*a33-a32*a13)
     &       + a31*(a12*a23-a22*a13)

      RETURN
      END

C**********************************************************************
```

```
C  Determinant of a 4 by 4 matrix.

C  Uses DET3.

      FUNCTION det4(a11,a12,a13,a14,a21,a22,a23,a24,a31,a32,
     &               a33,a34,a41,a42,a43,a44)

      IMPLICIT DOUBLE PRECISION  (A-Z)
      DOUBLE PRECISION   det4

      det4 = a11*det3(a22,a23,a24,a32,a33,a34,a42,a43,a44)
     &     - a21*det3(a12,a13,a14,a32,a33,a34,a42,a43,a44)
     &     + a31*det3(a12,a13,a14,a22,a23,a24,a42,a43,a44)
     &     - a41*det3(a12,a13,a14,a22,a23,a24,a32,a33,a34)

      RETURN
      END
```