Scientific Report No. 102

A BIVARIATIONAL ANALYSIS OF THE ORTHOGONAL MICROSTRIP-SLOTLINE CROSSOVER

by
Alexander A. M. Anger

Electromagnetics Laboratory
Department of Electrical and Computer Engineering
Campus Box 425
University of Colorado
Boulder, Colorado 80309-0425

August 1989

This research was supported by Texas Instruments under a Fellowship at the University of Colorado at Boulder.

A BIVARIATIONAL ANALYSIS OF THE ORTHOGONAL MICROSTRIP-SLOTLINE CROSSOVER

by

Alexander A. M. Anger

B.S. in Physics, University of Denver, 1983

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

1989

Anger, Alexander A. M. (M.S., Electrical Engineering)

A Bivariational Analysis of the Orthogonal Microstrip-Slotline Crossover

Thesis directed by Professor E. F. Kuester

An electromagnetic analysis of the microstrip-slotline crossover is presented. The scattering matrix is calculated for the four-port network that results when an infinitely long microstrip transmission line is coupled at ninety degrees with an infinitely long slot transmission line.

The solution relies on a variational, integral equation technique which has been called a bivariational technique. For the crossover problem, this technique is implemented by approximating the electric field (magnetic current) in the slotline aperture and the electric current in the microstrip. From these approximate magnetic and electric current boundary conditions, the fields in the entire space are computed. Expressions for the scattering parameters are then obtained which must be converted to a variational form to reduce the errors in the approximations to second order. The resulting integrals are solved numerically, and scattering parameters for a sample geometry are calculated.

Additionally, it is shown how the four-port matrix may be reduced to a lower dimension when either of the ports of the transmission lines are terminated. This allows synthesizing the crossover model with models for the open-circuit microstrip line and short-circuit slotline to obtain a model for the microstrip-slotline transition.

DEDICATION

Für Mama

ACKNOWLEDGEMENTS

Graduate school was not easy. I couldn't have finished it without the support of many people as well as the wonderful freedom which was available to me during that time. That precious freedom, although not always used in the most judicious manner, was priceless. I hope it will someday be accessible to all of us. I wish to thank Texas Instruments for sponsoring my fellowship and the following people for their help:

Professor E. F. Kuester for his steadfast persistence and tolerance

Professor D. C. Chang for his generous financial support

Marie Kindgren for her gentle admonition to "Get It Done"

David P. Wilson for being a model of responsibility

My fellow graduate students for various stepping-stones

And my many other friends in Boulder, a place of much love

CONTENTS

FIGURES

FIGURE

TABLES

TABLE

CHAPTER 1

INTRODUCTION

Many new transmission line structures have been developed over the last half century for the efficient transfer of microwave energy. In the 1920's and 30's these consisted primarily of hollow tubes (circular and rectangular waveguides) and coaxial lines. However, after World War II the developing field of passive and active semiconductor devices (pin diodes, bipolar junction transistors, field effect transistors, and so on) fostered the invention of transmission line structures which could be used to couple the microwaves in and out of these devices. These newer types of transmission lines include microstrips, slotlines, finlines, and numerous variations thereof.

Two of the more common microwave transmission lines are microstrip lines and slotlines. Diagrams representing these transmission line structures are shown in Figures 1.1 and 1.2. These lines have complementary properties, so that both are sometimes used in circuit designs. Microstrip lines are ideal for connecting devices, such as diodes, in series, while slotlines are ideal for connecting devices in parallel. On a given substrate, microstrip lines have lower impedances than slotlines of the same width. One other important characteristic, that makes both types of transmission lines desirable to work with, is the ability to fabricate transitions from microstrip or slotline to other types of transmission lines or waveguides. For example,

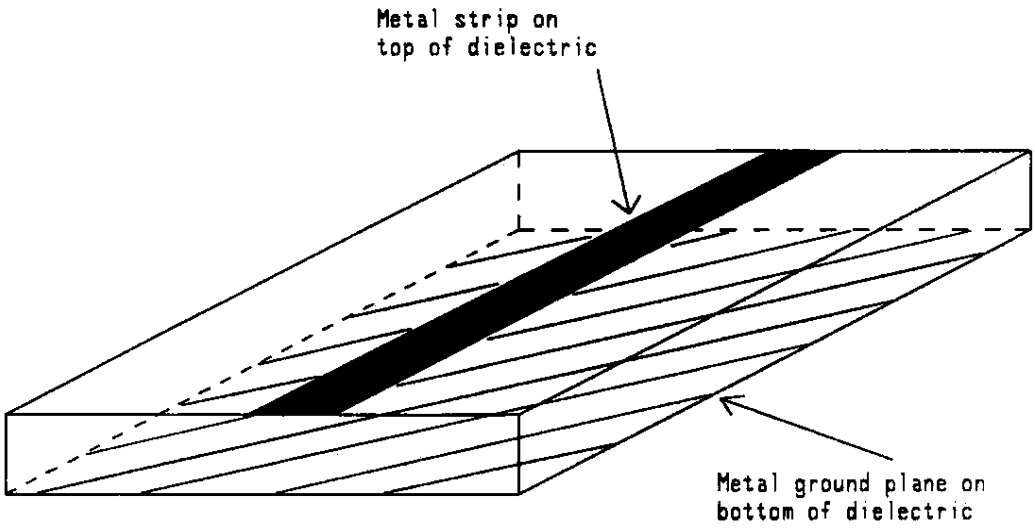


Figure 1.1: Microstrip transmission line

transitions to coaxial or coplanar lines are readily fabricated using microstrip lines, while transitions to finlines or rectangular waveguides are easier to fabricate using slotlines.

Microstrip and slotlines have been studied for over twenty years, and many papers and books have been written about them [1]–[8]. They are of continuing interest to many researchers in the field of electromagnetics. This is due to the mathematical complexity which arises when the laws of classical electrodynamics are applied to structures more complicated than spheres or cylinders in homogeneous, isotropic surroundings. Even when highly idealized, the microstrip geometry cannot be solved analytically, and only approximate results are known for current distributions and propagation characteristics. Nevertheless, these results and models are generally good enough for real-world applications. This is not the case for many of the current models for transmission line or waveguide discontinuities or transitions.

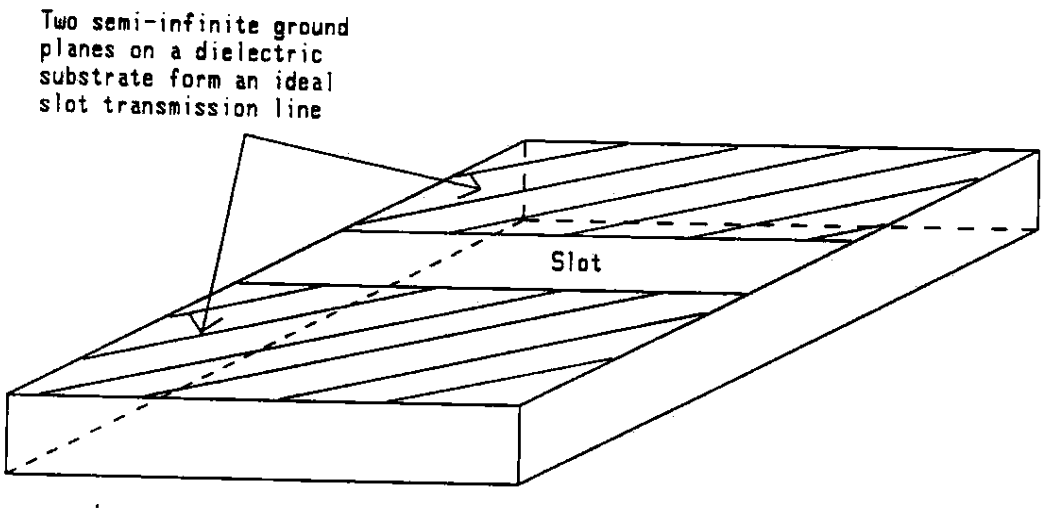


Figure 1.2: Slot transmission line

One of the problems in electromagnetics research which is of interest to microwave engineers involves the coupling which results when a microstrip is crossed at ninety degrees with a slotline (Figure 1.3). One may view this structure as a microstrip line which has a discontinuity in the ground plane or as a slotline which has been covered with a metal strip. This type of geometry, which is referred to as a microstrip-slotline junction or crossover, gives rise to exceptionally strong interaction between the dominant modes of the two respective lines. What this means physically is that an electromagnetic wave traveling along either the microstrip or the slotline, upon reaching such a junction, will be strongly scattered. Some of the wave will be reflected, some transmitted, some coupled into the other line, and some will exit the physical structure entirely in the form of radiation. The exact amounts that constitute "some" and the phase shifts that are incurred on the scattered waves are the crux of this electromagnetics problem.

The above description has assumed that the microstrip line and the slotline are infinitely long and that the universe consists of nothing

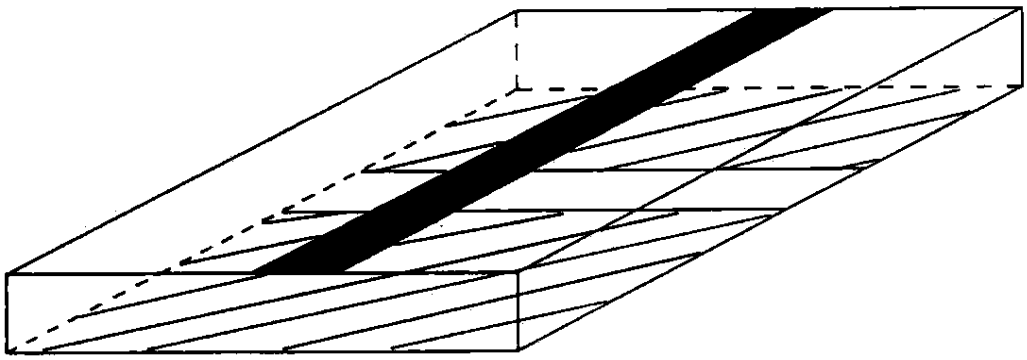


Figure 1.3: Orthogonal microstrip-slotline crossover

other than a dielectric slab to support these. Additional assumptions will be pointed out in Chapter 2.

Some of the reasons for performing this research are discussed next. As was mentioned before, a designer may be interested in using both a microstrip line and a slotline in various parts of his/her circuit. In order to maintain the "connection", the microwave signal needs to be passed from one of these lines to the other. The most common way in which this is currently accomplished is by open-circuiting the microstrip line about a quarter wavelength (microstrip guide wavelength) away from the crossover and short-circuiting the slotline about a quarter wavelength (slot guide wavelength) away from the same. This gives rise to what is known as the microstrip-slotline transition [8], [12], [13], [14]. A top view of this transition is shown in Figure 1.4. The transition is very effective even over a large bandwidth (VSWR $<$ 2, 1.9 – 4.9 GHz, see [13]). The bandwidth can be increased further by fanning the quarter-wave stubs as indicated in Figure 1.5.

The above stub lengths were given as "about a quarter wavelength"

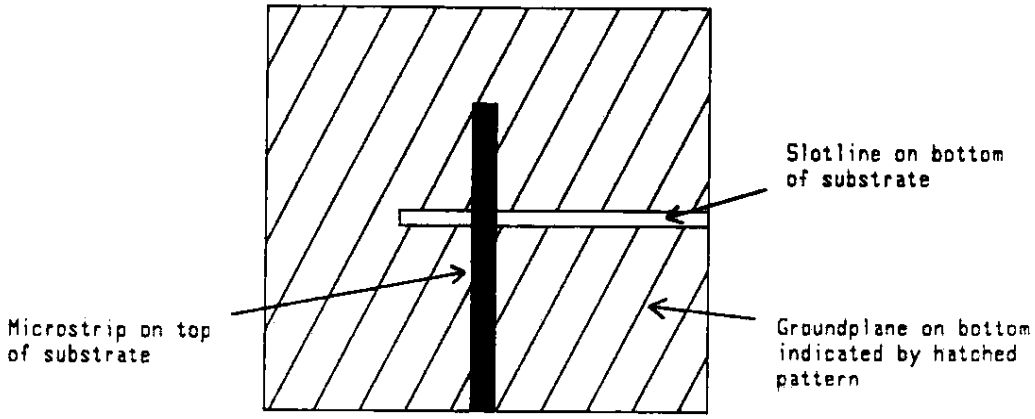


Figure 1.4: Microstrip-slotline transition

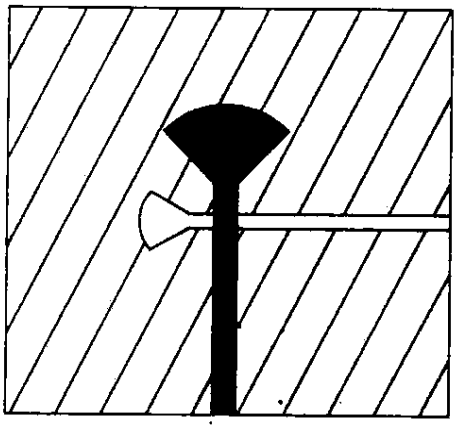


Figure 1.5: Extended bandwidth microstrip-slotline transition

because the exact lengths depend upon the reactive effects of the open microstrip and the shorted slotline. Additionally, the crossover itself contains reactances which must be taken into consideration when designing such a transition for a specific passband. The actual stub lengths should be chosen such that the combined phase shift on the center frequency signal due to the crossover and stub reactances equals that of an ideal quarter-wave stub.

The second major reason for studying the microstrip-slotline crossover is for designing certain types of antenna arrays. An array of slot radiators can be driven using microstrip feed lines, and similarly an array of microstrip radiators can be driven with slotline feeds [9], [10], [11]. A characterization of the crossover is essential if these arrays are to be designed accurately for field patterns and directivity.

Solving any one of the above structures directly using Maxwell's equations would be extremely difficult. Instead one can parse the real structure of interest into various smaller, conceptual blocks which are more amenable to analysis. Solutions for these blocks, which can be thought of as auxiliary geometries, can then be combined to obtain the sought-after solution. This approach is valid if the electromagnetic coupling between the blocks is small compared to the electromagnetic coupling occurring inside each block. Whether this assumption is valid is usually determined ex post facto, that is, after experimental verification of the final model.

Thus, to obtain a prediction of the performance of a microstrip-slotline transition, one first separates the problem into five independent blocks or problems — the uniform microstrip line, the open-circuit microstrip, the

uniform slotline, the short-circuit slotline, and the crossover. For the uniform lines, one needs to find the propagation constants. For the open-circuit microstrip, one needs the fringing capacitance and for the short-circuit slotline, the effective inductance. The crossover solution can be represented by a four-dimensional scattering matrix. These five solutions can then be combined to obtain a solution for the original microstrip-slotline transition. The accuracy of this model depends on whether the electrodynamic response of the individual building blocks is largely unaltered when they are brought into physical proximity.

This thesis consists of finding the 4×4 scattering matrix for the orthogonal microstrip-slotline crossover. At low frequencies, the amount of energy lost due to radiation may be neglected, and the crossover can be represented by a four-port network. Low frequency means that the guide wavelengths appearing in the structure are on the order of six or more times the widths of the microstrip and slotline. The four ports are defined in Chapter 2.

The first published research about the microstrip-slotline transition was described by D. Chambers, et al. in [12]. The paper contains both theoretical and experimental results. These results were expanded on by J. Knorr in [13] where he presents additional measured data. A rigorous analysis of the transition was recently presented by H. Yang and N. Alexópoulos [15] using the method of moments. They also extended their analysis to include microstrip fed slot radiators. The microstrip-slotline crossover was analyzed with the transverse resonance technique by T. Uwano, et al. in [16]. Of this published material, Uwano's paper contains the only results which

could be directly compared to results obtained in this thesis (see Chapter 6).

This thesis consists of six chapters and four appendixes. The detailed geometry of the crossover, assumptions, and symbol definitions are given in Chapter 2. Basic definitions of the field quantities, voltages, currents, and related quantities are also discussed there. Chapter 3 contains solutions for the electromagnetic fields in the crossover geometry. The fields are found using a spectral-domain technique. Derivations for the five independent scattering parameters are given in Chapter 4. The parameters are solved in terms of approximate scattered microstrip currents and slot fields. The solution's accuracy depends on the bivariational nature of the derived functionals [19].

The scattering parameters are found in terms of complicated inner products that take the form of doubly-infinite integrals laden with singularities. The detailed expressions for these integrals are given in Chapter 5, which also discusses the difficulties in obtaining numerical answers to the integrals. Chapter 6 is used to show sample results for the scattering parameters and to compare them to previously published data. It also discusses how to incorporate this model with others to find solutions for more practical geometries.

Appendix A is used to derive the fundamental mode fields of the microstrip line and its characteristic impedance and norm. The same is done for the slotline in Appendix B. Appendix C delves into the intricacies of finding numerical results for the six integrals of Chapter 5. The FORTRAN computer programs that compute the integrals are listed in Appendix D.

CHAPTER 2

THE ORTHOGONAL CROSSOVER

This chapter describes the problem geometry and the assumptions that were made in the analysis. It covers the basic definitions of the fields, voltages, scattering parameters and other conventions. Symmetry is applied to reduce the scattering matrix to one involving five unknowns instead of the original sixteen. Conditions on these five parameters are derived by assuming a small amount of radiation loss. These conditions are used to check the accuracy of the final computations. Also included is a section which lists the primary symbols found in this thesis.

2.1 Problem Formulation

In order to make the microstrip-slotline crossover problem amenable to theoretical analysis, a number of assumptions had to be made. Two kinds of assumptions were made in this analysis — geometrical and electrical.

The geometrical assumptions consist of the following. The microstrip line and the slotline are infinitely long. The conductors on either side of the dielectric substrate have zero thickness.

The structure is defined with respect to a rectangular coordinate system (Figure 2.1). The dielectric slab is of infinite extent in the xy-plane and has thickness d. It is located between $z = 0$ and $z = d$. The microstrip

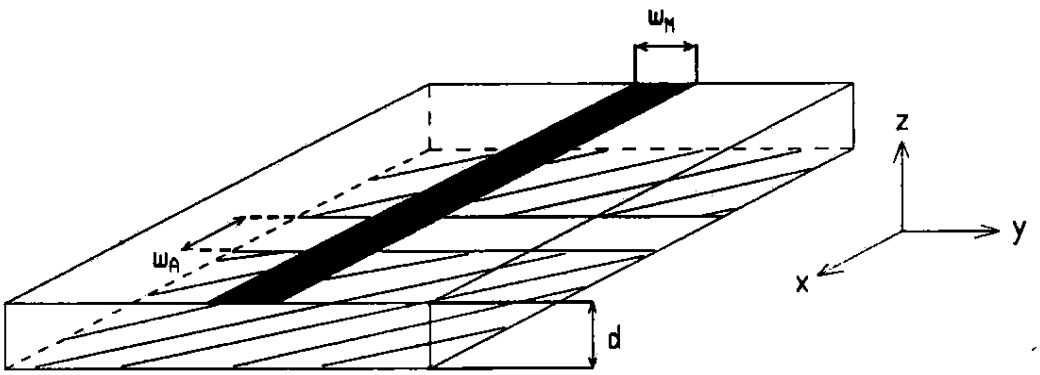


Figure 2.1: Microstrip-Slotline Crossover Geometry

line runs along the x-axis and has width w_M centered so that $-w_M/2 \leq y \leq w_M/2$. It is located on top of the dielectric slab at $z = d$. The slotline runs along the y-axis and has width w_A centered so that $-w_A/2 \leq x \leq w_A/2$. The slot is etched into the bottom of the ground plane at $z = 0$. Strictly speaking, this leaves behind two, semi-infinite ground planes. Taking the latter point of view eliminates the confusion that sometimes arises when the slotline is defined. A slotline by itself has no "ground" plane as such.

The electrical assumptions are as follows. Air has the electric permittivity, ϵ_0, and magnetic permeability, μ_0, of free space. The dielectric substrate has a relative electric permittivity ϵ_r and a relative magnetic permeability μ_r. It is isotropic, homogeneous, and linear. The dielectric losses are vanishingly small but not zero. Consequently, the dielectric constant will be complex. This is an important point which will arise later in the analysis and which will be referred to as the limiting absorption principle.

The limiting absorption principle assures that the guided waves decay to zero at infinity.

The conductivity of the infinitesimal layers of metal is infinite. The thickness of the dielectric substrate and the widths of the microstrip and slotline should be electrically small (i.e., less than a quarter guide wavelength), although this is not required for the general derivation presented in Chapter 4. Only the detailed calculations in the later chapters make use of this hypothesis.

The geometry of the microstrip and slotline must be chosen such that all higher-order modes are cutoff. The propagation characteristics of microstrips and slotlines are presumed known. The accuracy of this analysis is thus directly related to the accuracy of the propagation constants (β_M, β_A) or effective dielectric constants (ϵ_M, ϵ_A) which are used as input. Data on propagation characteristics for certain geometries and dielectrics have been compiled by Hoffmann [3]. Additionally, the current distribution for the fundamental microstrip mode and the electric field distribution for the fundamental slotline mode are being approximated to first order. The variational nature of the formulation will reduce this first-order error to a second-order error.

2.2 Field, Voltage, and Current Definitions

All electromagnetic quantities are represented in phasor notation with the usual factor $e^{i\omega t}$ where $i = \sqrt{-1}$, ω is a wave's angular frequency (in radians), and t represents time. The electric field is then given by

$$\vec{\mathbf{E}}(x, y, z, t) = \Re\left[\vec{E}(x, y, z)e^{i\omega t}\right] \tag{2.1}$$

where $\Re$ extracts the real part of the argument. The phasor $\vec{\boldsymbol{E}}(x,y,z)$ describes the spatial variation of the electric field and has scalar components E_x, E_y, and E_z. Similar expressions hold for the magnetic field, $\vec{\mathbf{H}}(x,y,z,t)$, the electric current density, $\vec{\mathbf{J}}(x,y,z,t)$, and the magnetic current density, $\vec{\mathbf{M}}(x,y,z,t)$. Their corresponding phasor quantities are represented by $\vec{\boldsymbol{H}}(x,y,z)$, $\vec{\boldsymbol{J}}(x,y,z)$, and $\vec{\boldsymbol{M}}(x,y,z)$. For propagating waves (or modes) along a guided structure, say the z-direction, the electric field can be written in terms of a transverse field distribution

$$\vec{\boldsymbol{E}}(x,y,z) = \vec{\boldsymbol{\mathcal{E}}}(x,y)e^{-i\beta z} \tag{2.2}$$

where β corresponds to the guide propagation constant. The minus sign yields a wave with phase velocity along the positive z-axis. The transverse electric field distribution, $\vec{\boldsymbol{\mathcal{E}}}(x,y)$, has scalar components $\mathcal{E}_x(x,y)$ in the x-direction and similar terms along the y- and z-axes. The transverse distributions for magnetic field and electric and magnetic currents are represented by $\vec{\boldsymbol{\mathcal{H}}}(x,y)$, $\vec{\boldsymbol{\mathcal{J}}}(x,y)$, and $\vec{\boldsymbol{\mathcal{M}}}(x,y)$, respectively.

In accordance with these definitions, we write the dominant mode fields for the microstrip and the slotline as a superposition of an incident and a reflected wave. Along the uniform, semi-infinite sections of the microstrip, the electric field becomes

$$\vec{\boldsymbol{E}}_M = \begin{cases} \sqrt{2Z_M}\left[a_1\vec{\boldsymbol{\mathcal{E}}}_M^{+}(y,z)e^{-i\beta_M x} + b_1\vec{\boldsymbol{\mathcal{E}}}_M^{-}(y,z)e^{i\beta_M x}\right] & (x<0) \\ \sqrt{2Z_M}\left[a_2\vec{\boldsymbol{\mathcal{E}}}_M^{-}(y,z)e^{i\beta_M x} + b_2\vec{\boldsymbol{\mathcal{E}}}_M^{+}(y,z)e^{-i\beta_M x}\right] & (x>0) \end{cases} \tag{2.3}$$

The a_i's and b_i's are the wave amplitudes of the incident and scattered

modes, while the superscripts + and − are used to differentiate the transverse field distributions of the forward and backward traveling modes, respectively. The electric field on the semi-infinite sections of the slotline is given by

$$\vec{E}_A = \begin{cases} \sqrt{2Z_A}\left[a_3\vec{\mathcal{E}}_A^{+}(x,z)e^{-i\beta_A y} + b_3\vec{\mathcal{E}}_A^{-}(x,z)e^{i\beta_A y}\right] & (y<0) \\ \sqrt{2Z_A}\left[a_4\vec{\mathcal{E}}_A^{-}(x,z)e^{i\beta_A y} + b_4\vec{\mathcal{E}}_A^{+}(x,z)e^{-i\beta_A y}\right] & (y>0) \end{cases} \tag{2.4}$$

Equations similar to (2.3) and (2.4) hold for $\vec{H}_M$ and $\vec{H}_A$.

From Maxwell's equations for propagating modes, a simple relationship between the transverse field distributions of forward and backward-traveling modes can be obtained [17]. Using the subscript t for the transverse components of $\vec{\mathcal{E}}$ and l for the longitudinal component, one finds that

$$\left.\begin{aligned} \vec{\mathcal{E}}_t^{-} &= \vec{\mathcal{E}}_t^{+} \\ \mathcal{E}_l^{-} &= -\mathcal{E}_l^{+} \\ \vec{\mathcal{H}}_t^{-} &= -\vec{\mathcal{H}}_t^{+} \\ \mathcal{H}_l^{-} &= \mathcal{H}_l^{+} \end{aligned}\right\} \tag{2.5}$$

The wave amplitudes a_i and b_i are indexed by four port numbers. The ports are referenced to the origin of the problem geometry as a limit, with port 1 defined as the plane at $x = 0^-$, port 2 at $x = 0^+$, port 3 at $y = 0^-$, and port 4 at $y = 0^+$.

The wave amplitudes can formally be related to transmission line parameters. These parameters consist of voltage and current waves and a characteristic impedance. The following conventions will be adopted. A $+x$-traveling microstrip mode has current, I_M, flowing in the $+x$ direction.

The strip is at positive voltage, V_M, with respect to the ground plane. A $+y$-traveling slotline mode has the half-plane at $x > 0$ carrying current, I_A, in the $+y$ direction. This half-plane is at positive voltage, V_A, with respect to the other one ($x < 0$). The characteristic impedances , Z_M and Z_A, are obtained from the corresponding norms in Appendixes A and B.

We can now define the wave amplitudes for the microstrip fundamental mode fields found in (2.3):

$$\left.\begin{aligned} a_1 &= \frac{1}{2\sqrt{2Z_M}}[V_M(0^-) + Z_M I_M(0^-)] \\ b_1 &= \frac{1}{2\sqrt{2Z_M}}[V_M(0^-) - Z_M I_M(0^-)] \\ a_2 &= \frac{1}{2\sqrt{2Z_M}}[V_M(0^+) - Z_M I_M(0^+)] \\ b_2 &= \frac{1}{2\sqrt{2Z_M}}[V_M(0^+) + Z_M I_M(0^+)] \end{aligned}\right\} \tag{2.6}$$

For the slotline fundamental mode field of (2.4), the wave amplitudes are defined similarly:

$$\left.\begin{aligned} a_3 &= \frac{1}{2\sqrt{2Z_A}}[V_A(0^-) + Z_A I_A(0^-)] \\ b_3 &= \frac{1}{2\sqrt{2Z_A}}[V_A(0^-) - Z_A I_A(0^-)] \\ a_4 &= \frac{1}{2\sqrt{2Z_A}}[V_A(0^+) - Z_A I_A(0^+)] \\ b_4 &= \frac{1}{2\sqrt{2Z_A}}[V_A(0^+) + Z_A I_A(0^+)] \end{aligned}\right\} \tag{2.7}$$

The scattering parameters can now be defined in terms of these wave amplitudes. With a wave incident at port j,

$$S_{ij} = \frac{b_i}{a_j} \qquad \text{for all } i \tag{2.8}$$

The scattering matrix consists of sixteen parameters, but by symmetry it can be reduced to five independent parameters. The scattering matrix becomes

$$[S] = \begin{bmatrix} \Gamma_M & T_M & C & C \\ T_M & \Gamma_M & -C & -C \\ C & -C & \Gamma_A & T_A \\ C & -C & T_A & \Gamma_A \end{bmatrix} \tag{2.9}$$

where Γ_M and Γ_A are the reflection coefficients for the microstrip and slotline, T_M and T_A are the transmission coefficients, and C is the coupling coefficient from one line to the other.

For electrically small line widths, the radiation losses will be negligible, and one can expect the scattering matrix to be unitary. Using this, yields the following relations

$$\left.\begin{aligned} |\Gamma_M + T_M|^2 &= 1 \\ |\Gamma_A - T_A|^2 &= 1 \\ \tfrac{1}{2}\left[1 - |\Gamma_M|^2 - |T_M|^2\right] &= |C|^2 \\ \tfrac{1}{2}\left[1 - |\Gamma_A|^2 - |T_A|^2\right] &= |C|^2 \\ C^*\left(\Gamma_M - T_M\right) + C\left(\Gamma_A^* - T_A^*\right) &= 0 \end{aligned}\right\} \tag{2.10}$$

The first two equations are automatically satisfied by the bivariational analysis as will be shown in Chapter 4. The remaining equations will be used in Chapter 6 to verify the internal consistency of the results.

2.3 List of Symbols

Following is a list of the primary symbols used in this thesis. It covers symbols from all chapters and appendixes. Symbols which are missing from this list are generally defined in the same section in which they are used.

The abbreviation FT means Fourier transform. A vector quantity is denoted with a small arrow ($\vec{}$) over a bold letter. This may appear redundant, but some of the bold letters do not stand out noticeably by themselves. Dyads are denoted with a double bar ($\bar{\bar{}}$). Quantities in the FT domain are denoted with a circumflex ($\hat{}$). Complex conjugation is indicated with an asterisk (*).

The subscript A is used for quantities which are associated with the slotline. "A" refers to the slot which may be interpreted as an aperture in the ground plane. "S" is already used for surfaces. The subscripts M and G are used for quantities which are associated with the microstrip or ground plane, respectively. The subscripts p and t refer to the planar or transverse directions. A planar direction is one tangential to the ground plane. A transverse direction is one perpendicular to the propagation axis of a transmission line. The superscripts + and − distinguish forward and backward-traveling modes, respectively. The subscripts + and − distinguish the two sides of a surface. The surfaces appearing in this analysis are all perpendicular to some axis. The + side is the one slightly more positive with respect to this axis than the − side.

Electric and magnetic currents and fields usually have two indexes. The superscript refers to the physical state with which the quantity is associated. The subscript relates the quantity to a geometrical structure, such as the microstrip or slotline. It also indicates the direction of components of a vector. Thus, E^f_{Ax} is the x-directed component of the slotline electric field of some state f.

Symbol	Description
a_i	Incident wave amplitude at port i
$\vec{a}_x, \vec{a}_y, \vec{a}_z$	Unit vectors in rectangular coordinates x, y, z
α	FT variable corresponding to x-transformation
b_i	Reflected wave amplitude at port i
β_A	Propagation constant of slotline
β_M	Propagation constant of microstrip line
c	Speed of light in free space
C	Coupling coefficient of crossover
$\mathrm{Ci}(x)$	Cosine integral
χ	Polar FT variable ($\chi^2 = n_M^2 + \lambda^2$)
$\vec{\chi}$	FT vector ($\vec{\chi} = n_M \vec{a}_x + \lambda \vec{a}_y$)
$\vec{\chi}_\perp$	FT vector ($\vec{\chi}_\perp = \lambda \vec{a}_x - n_M \vec{a}_y$)
d	Thickness of dielectric substrate
e	Natural base ($e \simeq 2.718281828$)
$\vec{E}$	Phasor electric field
$\vec{\mathcal{E}}$	Transverse electric field distribution of a propagating mode
ϵ_A	Effective dielectric constant of slotline
ϵ_M	Effective dielectric constant of microstrip line
ϵ_r	Relative permittivity of the dielectric substrate
η	Impedance of free space ($\eta \simeq 377\ \Omega$)
f	Frequency of electromagnetic sources and fields
γ	Polar FT variable ($\gamma^2 = \alpha^2 + \lambda^2$)
$\vec{\gamma}$	FT vector ($\vec{\gamma} = \alpha \vec{a}_x + \lambda \vec{a}_y$)
$\vec{\gamma}_\perp$	FT vector ($\vec{\gamma}_\perp = \lambda \vec{a}_x - \alpha \vec{a}_y$)

Γ_A	Slotline reflection coefficient of crossover
Γ_M	Microstrip reflection coefficient of crossover
$\vec{H}$	Phasor magnetic field
$\vec{\mathcal{H}}$	Transverse magnetic field distribution of a propagating mode
$H_0^{(2)}(x)$	Zeroth order Hankel function of the second kind
$\mathbf{H}_0(x)$	Zeroth order Struve function
i	Imaginary unity ($i = \sqrt{-1}$)
I	Current or integral (specific definition depends on subscript)
$I_0(x)$	Modified zeroth order Bessel function
$\Im(x)$	Imaginary part of x
$\vec{J}$	Phasor electric current
$\vec{\mathcal{J}}$	Transverse electric current distribution of a propagating mode
$J_0(x)$	Zeroth order Bessel function
$J_1(x)$	First order Bessel function
k_0	Free space propagation constant
$K_0(x)$	Modified zeroth order Neumann function
L, M_1, M_2	Integro-differential operators
$\hat{\hat{L}}, \hat{\hat{M}}_1, \hat{\hat{M}}_2$	Operators L, M_1, and M_2 expressed in FT domain as dyads
λ	FT variable corresponding to y-transformation
$\vec{M}$	Phasor magnetic current
μ_r	Relative permeability of dielectric
n_A	Effective refractive index of slotline
n_M	Effective refractive index of microstrip line
N_A	Slotline norm
N_M	Microstrip norm

ω	Angular frequency ($\omega = 2\pi f$)
ψ	Polar FT variable ($\psi^2 = \alpha^2 + n_A^2$)
$\vec{\psi}$	FT vector ($\vec{\psi} = \alpha\vec{a}_x + n_A\vec{a}_y$)
$\vec{\psi}_\perp$	FT vector ($\vec{\psi}_\perp = n_A\vec{a}_x - \alpha\vec{a}_y$)
π	A number close to 3.14159265358979323846264 3
$\vec{\Pi}_e$	Electric Hertz vector
$\vec{\Pi}_m$	Magnetic Hertz vector
Π_j	z-directed electric or magnetic Hertz vector
R	A functional
$\Re(x)$	Real part of x
$\vec{\rho}$	Plane vector ($\vec{\rho} = x\vec{a}_x + y\vec{a}_y$)
$\mathrm{si}(x)$	Sine integral
S_A	Surface occupied by slotline
S_M	Surface occupied by microstrip line
t	Time — rarely mentioned but always present
T_A	Slotline transmission coefficient of crossover
T_M	Microstrip transmission coefficient of crossover
θ	Polar FT variable ($\tan\theta = \lambda/\alpha$)
V	Voltage (specific definition depends on subscript)
w_A	Width of slotline
w_M	Width of microstrip line
x, y, z	Spatial variables in rectangular coordinate system
$Y_0(x)$	Zeroth order Neumann function
$Y_1(x)$	First order Neumann function
Z_A	Characteristic impedance of slotline

Z_M	Characteristic impedance of microstrip line
$\langle a, b \rangle$	Reaction of a-state fields with b-state sources
(f, g)	Inner product of two elements f and g from a set $\mathcal{S}$
∇_p	Laplacian or Nabla operator in the transverse xy-plane

CHAPTER 3

DERIVATION OF THE ELECTROMAGNETIC FIELDS

This chapter describes how the electromagnetic fields are determined for a dielectric slab with arbitrary electric fields impressed in a surface S_M at $z = d$ and electric currents impressed in a surface S_A at $z = 0$. The derivation is general, so the results apply to arbitrarily shaped surfaces. The impressed sources are used to derive expressions for the electric currents in S_M and electric fields in S_A. These expressions are inverted, and the impressed "sources" are written in terms of the microstrip current and slot electric field.

The fields derived in this manner are required for the computation of the inner products of Chapter 4. The derivation for the microstrip and slotline mode fields also draws heavily on results from this chapter. The fields must be calculated in the spatial Fourier transform domain, also referred to as the spectral domain, since the three-dimensional Helmholtz equation is too difficult to solve directly. By introducing a Fourier transform pair in two dimensions, the Helmholtz equation can be reduced to a simple one-dimensional equation. The spectral domain fields are then found from two scalar Hertz potentials.

3.1 Hertz Potentials

From books on electromagnetic theory, such as Harrington [18], one finds that the solution for electromagnetic fields may be obtained by solving the Helmholtz equation. Rather than solving for the electric or magnetic fields directly, it is easier to make use of the Hertz vectors [22], $\vec{\Pi}_e$ and $\vec{\Pi}_m$. For planar problems such as the microstrip-slotline crossover, it is sufficient to know only two scalar components of the Hertz vectors, one electric and one magnetic, in order to derive the complete electromagnetic fields [20]. These scalar components will be called Hertz potentials. They correspond to the z-directed components of the Hertz vectors and are denoted Π_e and Π_m.

The two Hertz potentials must satisfy the scalar Helmholtz equation

$$\nabla^2 \Pi_j + k^2 \Pi_j = 0 \tag{3.1}$$

where $j = e$ or m and k is the wavenumber of the appropriate dielectric ($k = k_0 = \omega\sqrt{\epsilon_0 \mu_0}$ in air and $k = k_0\sqrt{\epsilon_r \mu_r}$ in dielectric).

Given the Hertz potentials, the electromagnetic fields can be found by using [20]

$$\left.\begin{aligned} E_z &= \left(\frac{\partial^2}{\partial z^2} + k^2\right)\Pi_e \\ \vec{E}_p &= \frac{\partial}{\partial z}(\nabla_p \Pi_e) + i\omega\mu \vec{a}_z \times \nabla_p \Pi_m \\ H_z &= \left(\frac{\partial^2}{\partial z^2} + k^2\right)\Pi_m \\ \vec{H}_p &= \frac{\partial}{\partial z}(\nabla_p \Pi_m) - i\omega\epsilon \vec{a}_z \times \nabla_p \Pi_e \end{aligned}\right\} \tag{3.2}$$

where the subscript p refers to components lying in the x, y-plane. The planar Laplacian operator is defined as

$$\nabla_p = \vec{a}_x \frac{\partial}{\partial x} + \vec{a}_y \frac{\partial}{\partial y} \tag{3.3}$$

3.2 Spatial Fourier Transform

To obtain a solution to the wave equation (3.1), a Fourier transform procedure will be used to reduce the three-dimensional partial differential equation to an ordinary differential equation. Due to the planar nature of the geometry, the transform will be applied to the x- and y-dimensions. This will yield an ordinary differential equation in z which has a straightforward solution.

A spectral domain function, $\hat{f}(\alpha, \lambda)$, is obtained from the corresponding spatial domain function, $f(x, y)$, using the Fourier transform

$$\hat{f}(\alpha, \lambda) = \frac{k_0^2}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, e^{+ik_0 \vec{\gamma} \cdot \vec{\rho}}\, dx\, dy \tag{3.4}$$

where k_0 is the wavenumber of free space,

$$\vec{\gamma} = \alpha \vec{a}_x + \lambda \vec{a}_y \tag{3.5}$$

and

$$\vec{\rho} = x \vec{a}_x + y \vec{a}_y \tag{3.6}$$

A spatial domain function, f, is obtained from the corresponding transformed function, $\hat{f}$, using the inverse Fourier transform

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(\alpha, \lambda)\, e^{-ik_0 \vec{\gamma} \cdot \vec{\rho}}\, d\alpha\, d\lambda \tag{3.7}$$

The transform applies to either scalar or vector functions. It is easily shown from the definitions (3.4) and (3.7) that operating on a spatial quantity with ∇_p corresponds to multiplying its Fourier transform by the vector $-ik_0\vec{\gamma}$. Symbolically, this may be written as

$$\nabla_p f(x,y) \Longleftrightarrow -ik_0\vec{\gamma}\hat{f}(\alpha,\lambda) \tag{3.8}$$

Similarly, the operation $\vec{a}_z \times \nabla_p$ corresponds to multiplication by $ik_0\vec{\gamma}_\perp$ in the transform domain. The vector $\vec{\gamma}_\perp$ is defined as

$$\vec{\gamma}_\perp = -\vec{a}_z \times \vec{\gamma} = \lambda\vec{a}_x - \alpha\vec{a}_y \tag{3.9}$$

3.3 Solution for Electromagnetic Fields

Using (3.4) in conjunction with (3.1) yields the transformed Helmholtz equation

$$\left[\frac{d^2}{dz^2} - (\alpha^2+\lambda^2)\,k_0^2 + k^2\right]\hat{\Pi}_j = 0 \tag{3.10}$$

Keeping in mind that we are interested in physically realistic solutions, only terms decaying as $|z| \to \infty$ are retained. These may be written as

$$\hat{\Pi}_j(\alpha,\lambda,z) = \begin{cases} \mathcal{A}_j e^{-k_0 u_0 (z-d)} & (z \geq d) \\ \mathcal{B}_j \sinh(k_0 u_1 z) + \mathcal{C}_j \cosh(k_0 u_1 z) & (0 \leq z \leq d) \\ \mathcal{D}_j e^{k_0 u_0 z} & (z \leq 0) \end{cases} \tag{3.11}$$

The variables u_0 and u_1 appear frequently and are defined by

$$u_0 = \sqrt{\gamma^2 - 1} \tag{3.12}$$

and

$$u_1 = \sqrt{\gamma^2 - \epsilon_r\mu_r} \tag{3.13}$$

with

$$\gamma^2 = \alpha^2 + \lambda^2 \tag{3.14}$$

The branch for u_0 is chosen such that $\Re(u_0) \geq 0$ and $\Im(u_0) > 0$ if $\Re(u_0) = 0$. The eight amplitude coefficients $\mathcal{A}_{e,m}$, $\mathcal{B}_{e,m}$, $\mathcal{C}_{e,m}$, and $\mathcal{D}_{e,m}$ need to be determined from the boundary conditions. The electric and magnetic fields lying in the xy-plane may be written in terms of the unknown coefficients by combining (3.11) with (3.2) and making use of (3.8). For $z \geq d$,

$$\hat{\tilde{\boldsymbol{E}}}_p = ik_0 \left[k_0 u_0 \vec{\gamma} \mathcal{A}_e + i\mu_0 \omega \vec{\gamma}_\perp \mathcal{A}_m\right] e^{-k_0 u_0 (z-d)} \tag{3.15}$$

$$\hat{\tilde{\boldsymbol{H}}}_p = ik_0 \left[k_0 u_0 \vec{\gamma} \mathcal{A}_m - i\epsilon_0 \omega \vec{\gamma}_\perp \mathcal{A}_e\right] e^{-k_0 u_0 (z-d)} \tag{3.16}$$

For $0 \leq z \leq d$,

$$\begin{aligned} \hat{\tilde{\boldsymbol{E}}}_p \;=\; & -ik_0 \Big\{ k_0 u_1 \vec{\gamma} \left[\mathcal{B}_e \cosh(k_0 u_1 z) + \mathcal{C}_e \sinh(k_0 u_1 z)\right] - \\ & i\mu_r \mu_0 \omega \vec{\gamma}_\perp \left[\mathcal{B}_m \sinh(k_0 u_1 z) + \mathcal{C}_m \cosh(k_0 u_1 z)\right] \Big\} \end{aligned} \tag{3.17}$$

$$\begin{aligned} \hat{\tilde{\boldsymbol{H}}}_p \;=\; & -ik_0 \Big\{ k_0 u_1 \vec{\gamma} \left[\mathcal{B}_m \cosh(k_0 u_1 z) + \mathcal{C}_m \sinh(k_0 u_1 z)\right] + \\ & i\epsilon_r \epsilon_0 \omega \vec{\gamma}_\perp \left[\mathcal{B}_e \sinh(k_0 u_1 z) + \mathcal{C}_e \cosh(k_0 u_1 z)\right] \Big\} \end{aligned} \tag{3.18}$$

For $z \leq 0$,

$$\hat{\tilde{\boldsymbol{E}}}_p = -ik_0 \left[k_0 u_0 \vec{\gamma} \mathcal{D}_e - i\mu_0 \omega \vec{\gamma}_\perp \mathcal{D}_m\right] e^{k_0 u_0 z} \tag{3.19}$$

$$\hat{\tilde{\boldsymbol{H}}}_p = -ik_0 \left[k_0 u_0 \vec{\gamma} \mathcal{D}_m + i\epsilon_0 \omega \vec{\gamma}_\perp \mathcal{D}_e\right] e^{k_0 u_0 z} \tag{3.20}$$

The boundary conditions are now applied to find the unknown coefficients. Only the planar components of the fields will be needed. Electric

field continuity at z=d yields the following two relations

$$\mathcal{A}_e = -\frac{u_1}{u_0}\left[\mathcal{B}_e \cosh(k_0 d u_1) + \mathcal{C}_e \sinh(k_0 d u_1)\right] \tag{3.21}$$

and

$$\mathcal{A}_m = \mu_r\left[\mathcal{B}_m \sinh(k_0 d u_1) + \mathcal{C}_m \cosh(k_0 d u_1)\right] \tag{3.22}$$

These follow from (3.15) and (3.17) and the identity

$$\vec{\gamma} \cdot \vec{\gamma}_\perp = 0 \tag{3.23}$$

The latter equation is used throughout this derivation. Continuity of the electric field at $z = 0$ yields

$$\mathcal{D}_e = \frac{u_1}{u_0}\mathcal{B}_e \tag{3.24}$$

and

$$\mathcal{D}_m = \mu_r \mathcal{C}_m \tag{3.25}$$

Next, the electric field is matched to an equivalent electric field source distribution in the microstrip surface, $\hat{\tilde{\vec{E}}}_{S_M}$. $\hat{\tilde{\vec{E}}}_{S_M}$ is obtained in Chapter 4 when the excitation geometry is decomposed into various physical states. Its origin will become clearer then. Using (3.17) in conjunction with (3.23) results in

$$\vec{\gamma} \cdot \hat{\tilde{\vec{E}}}_{S_M} = -ik_0^2\gamma^2 u_1\left[\mathcal{B}_e \cosh(k_0 d u_1) + \mathcal{C}_e \sinh(k_0 d u_1)\right] \tag{3.26}$$

and

$$\vec{\gamma}_\perp \cdot \hat{\tilde{\vec{E}}}_{S_M} = -\mu_r \mu_0 \omega k_0 \gamma^2\left[\mathcal{B}_m \sinh(k_0 d u_1) + \mathcal{C}_m \cosh(k_0 d u_1)\right] \tag{3.27}$$

The final boundary condition is due to an equivalent current distribution, $\hat{\tilde{\vec{J}}}_{S_A}$, in the slotline aperture causing a discontinuity in the planar magnetic

field, $\hat{\vec{H}}_p$. Its origin will also become clearer in Chapter 4. It is related to the tangential magnetic field just on either side of the ground plane by

$$\hat{\vec{J}}_{S_A} = \vec{a}_z \times \left(\hat{\vec{H}}_{p+} - \hat{\vec{H}}_{p-} \right) \Big|_{z=0} \tag{3.28}$$

Using this with (3.18) and (3.20) yields the pair of equations

$$\vec{\gamma} \cdot \hat{\vec{J}}_{S_A} = \epsilon_0 \omega k_0 \gamma^2 \left(\epsilon_r \mathcal{C}_e - \frac{u_1}{u_0} \mathcal{B}_e \right) \tag{3.29}$$

and

$$\vec{\gamma}_\perp \cdot \hat{\vec{J}}_{S_A} = i k_0^2 \gamma^2 \left(u_1 \mathcal{B}_m - \mu_r u_0 \mathcal{C}_m \right) \tag{3.30}$$

Solving (3.26)–(3.30) for $\mathcal{B}_{e,m}$ and $\mathcal{C}_{e,m}$ requires some algebra which will be left to the reader. With $\eta = \sqrt{\mu_0/\epsilon_0}$, the solutions for four of the coefficients are

$$\mathcal{B}_e = \frac{u_0}{k_0^2 \gamma^2 u_1 D_e} \left[i \epsilon_r \vec{\gamma} \cdot \hat{\vec{E}}_{S_M} - \eta u_1 \sinh(k_0 d u_1) \vec{\gamma} \cdot \hat{\vec{J}}_{S_A} \right] \tag{3.31}$$

$$\mathcal{C}_e = \frac{1}{k_0^2 \gamma^2 D_e} \left[i \vec{\gamma} \cdot \hat{\vec{E}}_{S_M} + \eta u_0 \cosh(k_0 d u_1) \vec{\gamma} \cdot \hat{\vec{J}}_{S_A} \right] \tag{3.32}$$

$$\mathcal{B}_m = \frac{-1}{k_0^2 \gamma^2 D_m} \left[\frac{u_0}{\eta} \vec{\gamma}_\perp \cdot \hat{\vec{E}}_{S_M} + i \cosh(k_0 d u_1) \vec{\gamma}_\perp \cdot \hat{\vec{J}}_{S_A} \right] \tag{3.33}$$

$$\mathcal{C}_m = \frac{-1}{k_0^2 \gamma^2 D_m} \left[\frac{u_1}{\mu_r \eta} \vec{\gamma}_\perp \cdot \hat{\vec{E}}_{S_M} - i \sinh(k_0 d u_1) \vec{\gamma}_\perp \cdot \hat{\vec{J}}_{S_A} \right] \tag{3.34}$$

while the other four coefficients are readily found from (3.21), (3.22), (3.24), and (3.25). Two new variables have been introduced above which will appear frequently throughout the remaining chapters

$$D_e = \epsilon_r u_0 \cosh(k_0 d u_1) + u_1 \sinh(k_0 d u_1) \tag{3.35}$$

and

$$D_m = \mu_r u_0 \sinh(k_0 d u_1) + u_1 \cosh(k_0 d u_1) \tag{3.36}$$

Now that the coefficients have been determined, one may find the fields and currents for the microstrip-slotline geometry given arbitrary source distributions in S_M and S_A.

3.4 Microstrip Current and Slot Electric Field

Two quantities which will be of particular interest later on are the microstrip current and the slot electric field expressed in terms of the equivalent sources, $\hat{\tilde{\vec{E}}}_{S_M}$ and $\hat{\tilde{\vec{J}}}_{S_A}$. A slight reminder on terminology should be stated. The microstrip current is of course the current in S_M as opposed to the ground plane current of the microstrip mode. The slot electric field is the electric field in S_A. The slotline electric field is more general and refers to the electric field in all of space due to the slotline mode.

The microstrip current is found from

$$\hat{\tilde{\vec{J}}}_M = \vec{a}_z \times \left(\hat{\tilde{\vec{H}}}_{p+} - \hat{\tilde{\vec{H}}}_{p-} \right) \Big|_{z=d} \tag{3.37}$$

By using (3.21), (3.22), and (3.31)–(3.34) in (3.16) and (3.18), the microstrip current can be written in two parts as

$$\begin{aligned} \vec{\gamma} \cdot \hat{\tilde{\vec{J}}}_M &= \frac{-1}{D_e} \left\{ \frac{i}{\eta} \left[\left(\frac{\epsilon_r^2 u_0}{u_1} + \frac{u_1}{u_0} \right) \sinh(k_0 d u_1) + 2\epsilon_r \cosh(k_0 d u_1) \right] \times \right. \\ &\quad \left. \times \vec{\gamma} \cdot \hat{\tilde{\vec{E}}}_{S_M} + \epsilon_r u_0 \vec{\gamma} \cdot \hat{\tilde{\vec{J}}}_{S_A} \right\} \end{aligned} \tag{3.38}$$

$$\vec{\gamma}_\perp \cdot \hat{\tilde{\vec{J}}}_M = \frac{1}{D_m} \left\{ \frac{i}{\eta} \left[\left(\mu_r u_0^2 + \frac{u_1^2}{\mu_r} \right) \sinh(k_0 d u_1) + 2 u_0 u_1 \cosh(k_0 d u_1) \right] \times \right.$$

$$\times \vec{\gamma}_\perp \cdot \hat{\bar{\bar{E}}}_{S_M} - u_1 \vec{\gamma}_\perp \cdot \hat{\bar{J}}_{S_A} \Big\} \tag{3.39}$$

The slot field is found in a similar manner from (3.17)

$$\vec{\gamma} \cdot \hat{\bar{E}}_A = \frac{1}{D_e} \left[\epsilon_r u_0 \vec{\gamma} \cdot \hat{\bar{E}}_{S_M} + i\eta u_0 u_1 \sinh(k_0 d u_1) \vec{\gamma} \cdot \hat{\bar{J}}_{S_A} \right] \tag{3.40}$$

$$\vec{\gamma}_\perp \cdot \hat{\bar{E}}_A = \frac{1}{D_m} \left[u_1 \vec{\gamma}_\perp \cdot \hat{\bar{E}}_{S_M} - i\mu_r \eta \sinh(k_0 d u_1) \vec{\gamma}_\perp \cdot \hat{\bar{J}}_{S_A} \right] \tag{3.41}$$

It will be useful to rewrite (3.38)–(3.41) by solving for $\hat{\bar{\bar{E}}}_{S_M}$ and $\hat{\bar{J}}_{S_A}$. The results are shown in the next section.

3.5 Operator Formalism for Fields and Currents

We define the following FT operators

$$\left.\begin{aligned} \hat{\bar{\bar{L}}} &= \frac{1}{\gamma^2} \left(\frac{\epsilon_r u_0}{D_e} \vec{\gamma}\vec{\gamma} + \frac{u_1}{D_m} \vec{\gamma}_\perp \vec{\gamma}_\perp \right) \cdot \\ \hat{\bar{\bar{M}}}_1 &= \frac{i}{\eta \gamma^2} \left(\frac{\mathcal{F}}{D_e} \vec{\gamma}\vec{\gamma} + \frac{\mathcal{G}}{D_m} \vec{\gamma}_\perp \vec{\gamma}_\perp \right) \cdot \\ \hat{\bar{\bar{M}}}_2 &= \frac{i\eta \sinh(k_0 d u_1)}{\gamma^2} \left(\frac{u_0 u_1}{D_e} \vec{\gamma}\vec{\gamma} - \frac{\mu_r}{D_m} \vec{\gamma}_\perp \vec{\gamma}_\perp \right) \cdot \end{aligned}\right\} \tag{3.42}$$

where

$$\mathcal{F} = \frac{\epsilon_r^2 u_0^2 - D_e^2}{u_0 u_1 \sinh(k_0 d u_1)} \tag{3.43}$$

and

$$\mathcal{G} = \frac{D_m^2 - u_1^2}{\mu_r \sinh(k_0 d u_1)} \tag{3.44}$$

The inverted equations corresponding to (3.38)–(3.41) can be written in operator notation as

$$\left.\begin{aligned} \hat{\vec{\vec{E}}}_{S_M} &= \hat{\vec{\vec{L}}}\hat{\vec{\vec{E}}}_A + \hat{\vec{\vec{M}}}_2\hat{\vec{J}}_M \\ \hat{\vec{J}}_{S_A} &= \hat{\vec{\vec{M}}}_1\hat{\vec{\vec{E}}}_A - \hat{\vec{\vec{L}}}\hat{\vec{J}}_M \end{aligned}\right\} \tag{3.45}$$

The above equations correspond to integro-differential operator equations when transformed back into the space domain. They may be interpreted as a solution for the electric field, $\hat{\vec{\vec{E}}}_{S_M}$, in S_M and the electric current, $\hat{\vec{J}}_{S_A}$, in S_A given a predefined microstrip current, $\hat{\vec{J}}_M$, and slotline electric field, $\hat{\vec{\vec{E}}}_A$.

A few properties of the operators (3.42) need to be stated before continuing. First we define the following inner products between two vector functions, $\vec{f}$ and $\vec{g}$. When $\vec{f}$ or $\vec{g} = 0$ outside of S_A

$$(\vec{f}, \vec{g})_{S_A} = \int_{S_A} \vec{f} \cdot \vec{g}\, dS \tag{3.46}$$

that is, the inner product is simply the dot product of the two functions integrated over the slotline surface. When $\vec{f}$ or $\vec{g} = 0$ outside of S_M

$$(\vec{f}, \vec{g})_{S_M} = \int_{S_M} \vec{f} \cdot \vec{g}\, dS \tag{3.47}$$

In this case, the dot product is integrated over the microstrip surface.

Using the FT definitions of Section 3.2 and the symmetric operators (3.42), it is easy to show the following properties. The symbols L, M_1, and M_2 are the space domain operations corresponding to $\hat{\vec{\vec{L}}}$, $\hat{\vec{\vec{M}}}_1$, and $\hat{\vec{\vec{M}}}_2$. When $\vec{f}$ or $\vec{g} = 0$ outside of S_A

$$(M_1\vec{f}, \vec{g})_{S_A} = (\vec{f}, M_1\vec{g})_{S_A} \tag{3.48}$$

When $\vec{f}$ or $\vec{g} = 0$ outside of S_M

$$(M_2\vec{f}, \vec{g})_{S_M} = (\vec{f}, M_2\vec{g})_{S_M} \tag{3.49}$$

When $\vec{f} = 0$ outside of S_A and $\vec{g} = 0$ outside of S_M

$$(L\vec{f}, \vec{g})_{S_M} = (\vec{f}, L\vec{g})_{S_A} \tag{3.50}$$

These operator properties will be used frequently when converting the functionals of Chapter 4 to a variational form.

3.6 A Bivariational Functional

We will now derive a bivariational functional involving the quantities from the operator equation (3.45). Let's say we needed to accurately compute the following inner product even though we only had an approximate estimate for the functions $\vec{E}_A^a$ and $\vec{J}_M^a$.

$$(\vec{\Phi}^a, \vec{F}^b) = (\vec{E}_A^a, \vec{J}_{S_A}^b)_{S_A} - (\vec{J}_M^a, \vec{E}_{S_M}^b)_{S_M} \tag{3.51}$$

This inner product is a more general inner product than the ones defined in Section 3.5. The two vectors $\vec{\Phi}$ and $\vec{F}$ are defined by

$$\vec{\Phi}^a = \begin{bmatrix} \vec{E}_A^a \\ -\vec{J}_M^a \end{bmatrix} \tag{3.52}$$

and

$$\vec{F}^b = \begin{bmatrix} \vec{J}_{S_A}^b \\ \vec{E}_{S_M}^b \end{bmatrix} \tag{3.53}$$

The superscripts a and b distinguish different states (a term that refers to having the same problem geometry but with different sources). The concept of states is explained in more detail in Chapter 4. Thus, $\vec{E}_A^a$ and $\vec{E}_A^b$ refer to two entirely different electric field distributions in the slotline aperture. This hints at the origin of the term bivariational. It relates to the fact that

the inner product (3.51), for which a variational form is sought, consists of a reaction integral between fields from one state with sources from a second state.

With the matrix notation used in (3.52) and (3.53), the operator equation (3.45) may be rewritten as

$$\mathcal{L}\vec{\Phi} = \vec{F} \tag{3.54}$$

where

$$\mathcal{L} = \begin{bmatrix} M_1 & L \\ L & -M_2 \end{bmatrix} \tag{3.55}$$

Since the derivation of (3.45) was for arbitrary source distributions $\vec{J}_{S_A}$ and $\vec{E}_{S_M}$, (3.54) holds for any state of the dielectric slab geometry. This implies that $\mathcal{L}\vec{\Phi}^a = \vec{F}^a$ and $\mathcal{L}\vec{\Phi}^b = \vec{F}^b$. The operator $\mathcal{L}$ is symmetric as can readily be shown by using the properties stated in (3.48)–(3.50)

$$(\mathcal{L}\vec{\Phi}^a, \vec{\Phi}^b) = (\vec{\Phi}^a, \mathcal{L}\vec{\Phi}^b) \tag{3.56}$$

Now, if we approximate our vectors $\vec{\Phi}^a$ and $\vec{\Phi}^b$ by $\vec{\Psi}^a = \vec{\Phi}^a + \delta\vec{\Phi}^a$ and $\vec{\Psi}^b = \vec{\Phi}^b + \delta\vec{\Phi}^b$, where $\delta\vec{\Phi}^a$ and $\delta\vec{\Phi}^b$ are the errors in our approximation, then the functional

$$R_{12} = (\vec{\Psi}^a, \vec{F}^b) + (\vec{\Psi}^b, \vec{F}^a) - (\vec{\Psi}^a, \mathcal{L}\vec{\Psi}^b) \tag{3.57}$$

is easily shown to be bivariational. This means that R_{12} approximates the inner product (3.51) to second-order error, or after using the symmetry property of $\mathcal{L}$

$$R_{12} = \begin{cases} (\vec{\Phi}^a, \vec{F}^b) - (\delta\vec{\Phi}^a, \mathcal{L}\,\delta\vec{\Phi}^b) \\ (\vec{\Phi}^b, \vec{F}^a) - (\delta\vec{\Phi}^a, \mathcal{L}\,\delta\vec{\Phi}^b) \end{cases} \tag{3.58}$$

The functional (3.58) reduces to the following, variational form when only one state of the problem geometry is being considered

$$R = 2(\vec{\Psi}, \vec{F}) - (\vec{\Psi}, \mathcal{L}\vec{\Psi}) \approx (\vec{\Phi}, \vec{F}) \tag{3.59}$$

The functional R will be used to calculate the microstrip and slotline reflection coefficients while R_{12} will be used for the transmission and coupling coefficients.

CHAPTER 4

FINDING THE SCATTERED WAVES

In this chapter the derivation of the scattering parameters for the microstrip-slotline crossover is shown. The general approach consists of launching the fundamental mode on either the microstrip line or the slotline. This is done by making use of the equivalence principle. The scattered mode is then "picked" out from the total scattered field distribution using either the same or a second set of sources. The resulting inner products are rewritten in terms of quantities which can be approximated, that is the microstrip current and the slotline electric field. Heavy use is made of the Lorentz reciprocity theorem. Finally, the inner products are converted into a variational form with respect to the microstrip current and the slotline field.

4.1 Microstrip Reflection Coefficient

Before deriving the scattering parameters for the crossover, it will be helpful to explain the concept of "states". A state is defined as a combination of a physical geometry and a set of sources. The use of states allows one to split a problem geometry into various pieces and to recombine them by the principle of superposition. The best way to explain states is by using them.

In order to compute scattering parameters, one needs to excite

the geometry in a suitable manner to generate the electromagnetic waves. The sources should be chosen such that they produce only the fundamental mode on the launching transmission line. The discontinuities, if any, will cause scattering of the waves in all directions. A scattering parameter, however, requires that one sample only the energy which is contained in the scattered, fundamental mode. In this problem, this sampling is performed with similar sources which launch the fundamental modes. Heavy use is made of the completeness and orthogonality of modes to accomplish this sampling.

4.1.1 Microstrip Mode Sources To derive an equation for the microstrip reflection coefficient, Γ_M, one first needs to launch the incident wave. In this and the following sections, we will make use of equivalent electric and magnetic current sources, chosen such that they launch only the dominant mode on the transmission line of interest. The first set of sources are designated as $\vec{J}_M^{\,1}$ and $\vec{M}_M^{\,1}$ and are sometimes called microstrip mode sources. They are distributed on a semi-infinite, transverse plane. This plane is called S_1 and is located at $x = -w_A/2$ as indicated in Figure 4.1. The sources may be thought of as equivalent sources corresponding to the transverse field distribution of the fundamental microstrip mode traveling in the positive x-direction, or using (2.2)

$$\left.\begin{aligned} \vec{J}_M^{\,1} &= \vec{a}_x \times \vec{\mathcal{H}}_M^{+}\, e^{i\beta_M w_A/2} \\ \vec{M}_M^{\,1} &= -\vec{a}_x \times \vec{\mathcal{E}}_M^{+}\, e^{i\beta_M w_A/2} \end{aligned}\right\} \tag{4.1}$$

The $+$ superscript is used for forward propagating waves, $-$ for backward propagating waves. The sources have been normalized such that the fields

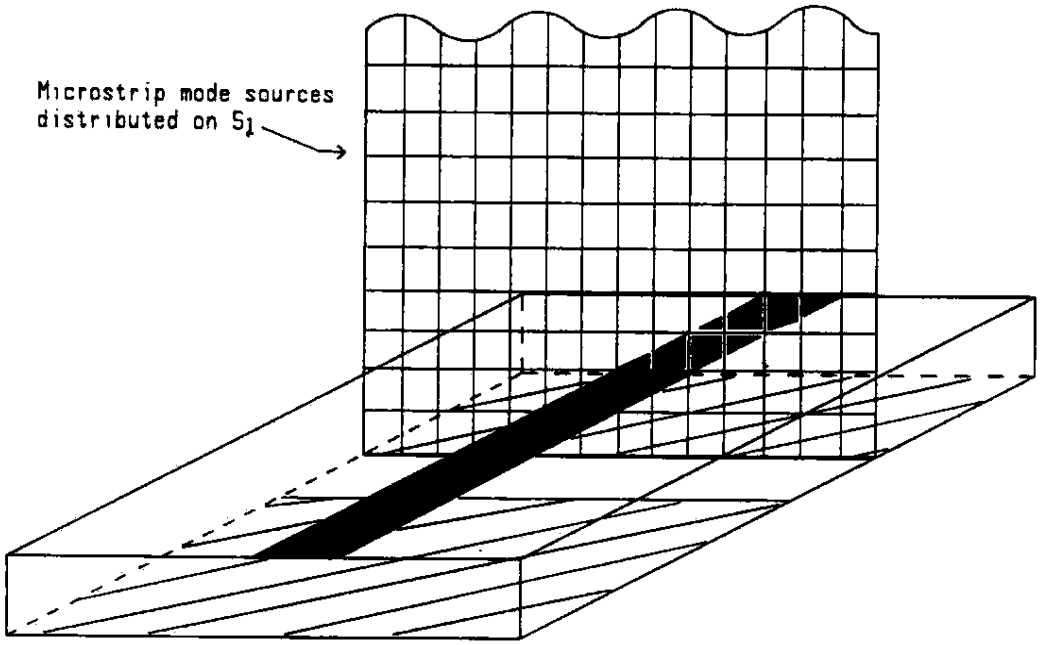


Figure 4.1: The t_1-state

have unity amplitude. When computing the scattering parameters from (2.8), the constant factors $\sqrt{2Z_M}$ or $\sqrt{2Z_A}$ from (2.3) and (2.4) will cancel each other for four of the five parameters. For the coupling factor, the situation is a bit more complicated, but the resolution of this issue is deferred until Section 4.5.

The sources launch a wave on the microstrip, but the wave is immediately scattered from the junction. Due to the completeness property of the mode spectrum, the total electromagnetic fields may be written as a sum of the fundamental microstrip mode plus higher order terms

$$\left.\begin{aligned} \vec{E}^{\,t_1} &= \Gamma_M \vec{\mathcal{E}}_M^{-} e^{-i\beta_M w_A/2} + \cdots \\ \vec{H}^{\,t_1} &= \Gamma_M \vec{\mathcal{H}}_M^{-} e^{-i\beta_M w_A/2} + \cdots \end{aligned}\right\} \tag{4.2}$$

Here is an example of a "state", in this case the t_1-state. The subscript 1

is used to distinguish states with similar geometries, but with different excitations. The t_1-state refers to the microstrip-slotline crossover geometry under the influence of sources $\vec{J}_M^1$ and $\vec{M}_M^1$ as shown in Figure 4.1. Additional states will be defined shortly.

One can readily write down an expression for Γ_M by making use of mode orthogonality [17] in conjunction with (2.5), (4.1) and (4.2)

$$\Gamma_M = -\frac{1}{2N_M} \int_{S_1} \left(\vec{J}_M^1 \cdot \vec{E}^{t_1} - \vec{M}_M^1 \cdot \vec{H}^{t_1} \right) dS \tag{4.3}$$

The norm appearing in (4.3) as N_M is defined by the following integral

$$N_M = \int_S \vec{a}_x \cdot \left(\vec{\mathcal{E}}_M^+ \times \vec{\mathcal{H}}_M^+ \right) dS \tag{4.4}$$

where S is any surface transverse to the x-axis. The derivation for the norms of the microstrip and slotline are given in Appendixes A and B. The normalization condition is that the total current on the microstrip equals 1 A.

The problem with (4.3) is that we cannot compute anything from it directly since we do not know the total fields. This is why the problem must be split into various states and auxiliary geometries must be considered. By separating the problem and applying the Lorentz reciprocity theorem [17], the expression (4.3) can be converted to a computationally accessible form.

4.1.2 Reactions, States, and Lorentz Reciprocity The integrations found in (4.3) are performed over the region of space occupied by sources of one state and involve the fields of either the same or a different state. The latter is what makes the Lorentz reciprocity theorem so useful. It allows one to mix results of simpler problems to generate solutions to more

complicated problems. The reaction integral of the type found in (4.3) can be expressed in shorter form as in

$$\int_{S_1} \left(\vec{J}_M^1 \cdot \vec{E}^{t_1} - \vec{M}_M^1 \cdot \vec{H}^{t_1} \right) dS = \langle t_1, f_1 \rangle \tag{4.5}$$

The first term between the angle brackets points to the state from which the electromagnetic fields are obtained. The second term points to the state from which the sources are obtained. Another representation of (4.5) can be obtained by recalling the inner product notation (3.46) and (3.47). The notation can readily be generalized to represent an integral over the surface S_1.

$$\langle t_1, f_1 \rangle = (\vec{J}_M^1, \vec{E}^{t_1})_{S_1} - (\vec{M}_M^1, \vec{H}^{t_1})_{S_1} \tag{4.6}$$

The reaction and inner product notation will be used frequently.

With these preliminary details out of the way, one may now break up the t_1-state into a superposition of the two states d_1 and f_1 as shown in Figures 4.2 and 4.3. To understand this superposition, one must first apply the equivalence principle to the t_1-state and transform it into a slightly modified form.

We replace the microstrip by wrapping it in an infinitesimal surface inside of which we place the equivalent current source, $\vec{J}_M^{d_1}$. This equivalent current source corresponds exactly to the incident and scattered currents which were generated by $\vec{J}_M^1$ and $\vec{M}_M^1$, but using the equivalent current allows us to take away the microstrip conductor. Thus, the equivalent current is an impressed current while the original current was an induced current, but since they are practically identical, the same symbol will be used for either.

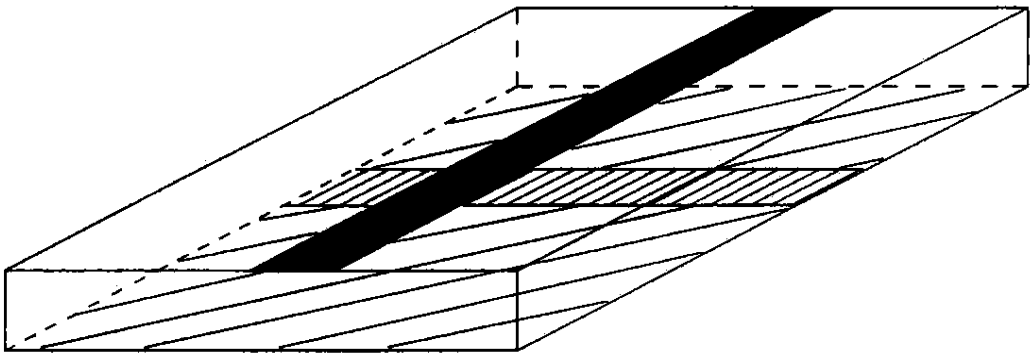


Figure 4.2: The d_1-state

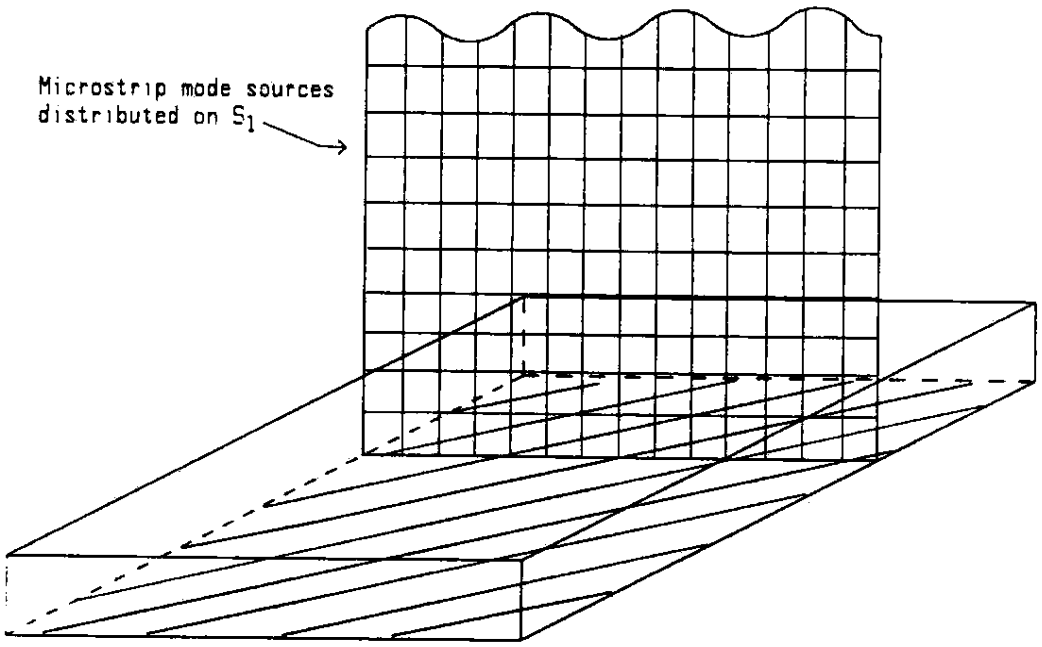


Figure 4.3: The f_1-state

In a similar manner, we remove the slot from the ground plane and replace it by the equivalent magnetic current sources $\vec{M}_{A+}^{d_1}$ and $\vec{M}_{A-}^{d_1}$ defined by

$$\left.\begin{aligned} \vec{M}_{A+}^{d_1} &= -\vec{a}_z \times \vec{E}_A^{d_1} \\ \vec{M}_{A-}^{d_1} &= \vec{a}_z \times \vec{E}_A^{d_1} \end{aligned}\right\} \tag{4.7}$$

The subscript $+$ refers to the $z > 0$ side of S_A and $-$ to the $z < 0$ side. The aperture can now be replaced with a conductor leaving behind a solid ground plane once again. The modified t_1-state may be viewed as the geometry of a grounded dielectric slab excited by the original sources, $\vec{J}_M^1$ and $\vec{M}_M^1$, and by the equivalent sources, $\vec{J}_M^{d_1}$, $\vec{M}_{A+}^{d_1}$ and $\vec{M}_{A-}^{d_1}$.

The splitting of the t_1-state, via $t_1 = d_1 + f_1$, should be clearer now. The d_1-state corresponds to the equivalent microstrip and slotline sources radiating into the grounded dielectric slab environment. It generates the fields $\vec{E}^{d_1}$ and $\vec{H}^{d_1}$. The f_1-state corresponds to the original sources radiating into the grounded dielectric slab environment. It generates the fields $\vec{E}^{f_1}$ and $\vec{H}^{f_1}$. Superposing the two states and applying the equivalence principle in reverse gives us back the t_1-state, i.e., the original sources radiating into the microstrip-slotline geometry.

The reaction (4.6) may now be broken into the following secondary reactions

$$\langle t_1, f_1 \rangle = \langle d_1, t_1 \rangle + \langle f_1, t_1 \rangle \tag{4.8}$$

which can be further manipulated using the Lorentz reciprocity theorem. The Lorentz reciprocity theorem may be stated as

$$\langle a, b \rangle = \langle b, a \rangle \tag{4.9}$$

that is, the reaction of a-state fields with b-state sources equals the reaction of b-state fields with a-state sources. Applying this theorem to the first term in (4.8) yields

$$\langle d_1, f_1 \rangle = \langle f_1, d_1 \rangle \tag{4.10}$$

which may be expanded as in (4.6)

$$\langle f_1, d_1 \rangle = (\vec{J}_M^{\,d_1}, \vec{E}^{\,f_1})_{S_M} - (\vec{M}_{A+}^{\,d_1}, \vec{H}_{+}^{\,f_1})_{S_A} - (\vec{M}_{A-}^{\,d_1}, \vec{H}_{-}^{\,f_1})_{S_A} \tag{4.11}$$

Using (4.7) and the following definition for the ground plane current of the f_1-state,

$$\vec{J}_G^{\,f_1} = \vec{a}_z \times \left(\vec{H}_{p+}^{\,f_1} - \vec{H}_{p-}^{\,f_1} \right)\Big|_{z=0} \tag{4.12}$$

allows us to rewrite (4.11) as

$$\langle f_1, d_1 \rangle = (\vec{J}_M^{\,d_1}, \vec{E}^{\,f_1})_{S_M} - (\vec{E}_A^{\,d_1}, \vec{J}_G^{\,f_1})_{S_A} \tag{4.13}$$

This reaction is in a more useful form than the original since we can approximate the microstrip current and slotline field. However, we have no easy way of obtaining $\vec{E}^{\,f_1}$ or $\vec{J}_G^{\,f_1}$. To substitute other known quantities for the f_1-state field and current, we first need to revisit the operator solution of Chapter 3. By applying this solution to the related problem of a uniform microstrip excited by $\vec{J}_M^{\,1}$ and $\vec{M}_M^{\,1}$, we can convert $\vec{E}^{\,f_1}$ and $\vec{J}_G^{\,f_1}$ to dominant mode quantities of the microstrip.

4.1.3 Operator Equations For Related States Recall the operator solution (3.45). Since the derivation was for arbitrary source distributions $\hat{\vec{E}}_{S_M}$ and $\hat{\vec{J}}_{S_A}$, the results can readily be applied to the d_1-state. Furthermore, we will write the equation using the spatial domain quantities.

$$\left.\begin{aligned} \vec{E}_p^{\,d_1} &= L\vec{E}_A^{\,d_1} + M_2\vec{J}_M^{\,d_1} \qquad (z=d) \\ \vec{J}_G^{\,d_1} &= M_1\vec{E}_A^{\,d_1} - L\vec{J}_M^{\,d_1} \qquad (z=0) \end{aligned}\right\} \tag{4.14}$$

$\vec{E}_p^{\,d_1}$ is the tangential electric field that would be produced by $\vec{E}_A^{\,d_1}$ and $\vec{J}_M^{\,d_1}$ radiating alone into the grounded dielectric slab environment, and $\vec{J}_G^{\,d_1}$ is the ground plane current which would be induced. We can gain additional insight into the origins of $\vec{E}_p^{\,d_1}$ and $\vec{J}_G^{\,d_1}$ by considering the original geometry. In the t_1-state geometry, the tangential electric field must be zero in S_M and the electric current must be zero in S_A. This is not the case for the f_1 and d_1 states since they consist of the grounded dielectric slab geometry. Since $t_1 = f_1 + d_1$, the following must hold in order to satisfy the boundary conditions of the t_1-state

$$\left.\begin{aligned} \vec{E}_p^{\,d_1} + \vec{E}_p^{\,f_1} &= 0 \qquad (\text{in } S_M) \\ \vec{J}_G^{\,d_1} + \vec{J}_G^{\,f_1} &= 0 \qquad (\text{in } S_A) \end{aligned}\right\} \tag{4.15}$$

This allows us to write $\vec{E}_p^{\,d_1}$ and $\vec{J}_G^{\,d_1}$ in terms of the f_1-state field and current. But how can we get additional information on the f_1-state? To address this question we next look at the uniform microstrip geometry.

Let $\vec{J}_M^{\,1}$ and $\vec{M}_M^{\,1}$ radiate in the presence of an infinite, uniform microstrip (Figure 4.4). We will refer to this as the u_1-state. By mode orthogonality [17] and (4.1), the sources produce the dominant microstrip mode fields

$$\vec{E}^{\,u_1} = \vec{E}_M^{\,+} = \begin{cases} 0 & (x < -w_A/2) \\ \vec{\mathcal{E}}_M^{\,+}(y,z)e^{-i\beta_M x} & (x > -w_A/2) \end{cases} \tag{4.16}$$

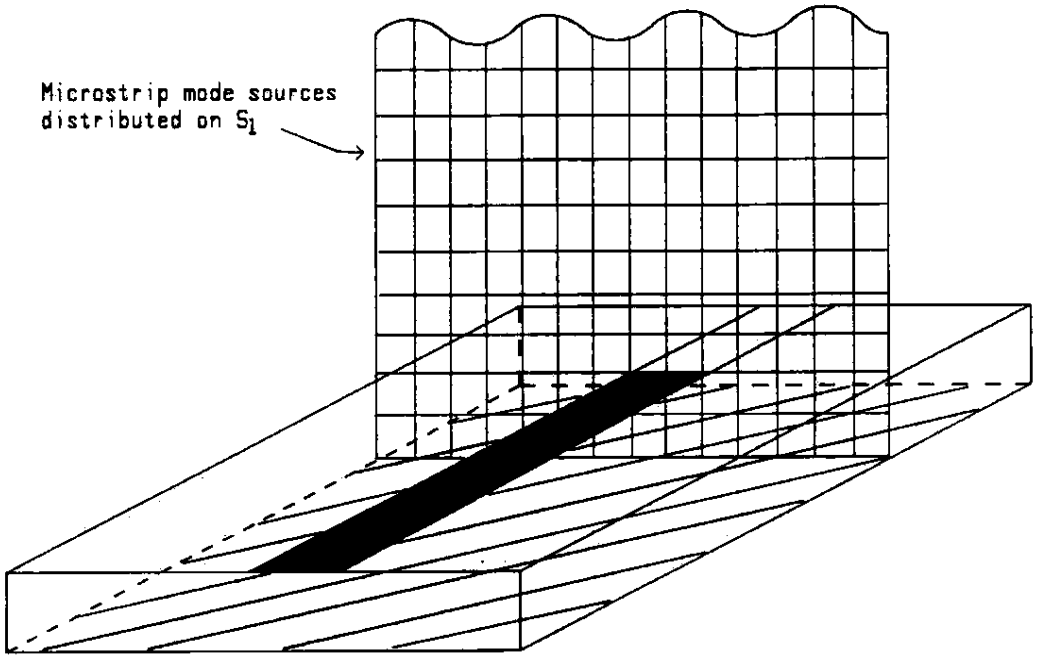


Figure 4.4: The u_1-state

and a similar term for $\vec{H}^{u_1}$. The microstrip current has been visually indicated in Figure 4.4 by the black shading.

The u_1-state is now partitioned into two states. First, the microstrip is replaced by an equivalent current source, $\vec{J}_M^{i_1}$. This leaves the alternate representation of the u_1-state as the microstrip mode sources and the equivalent microstrip current radiating into the grounded dielectric slab environment. By superposition, this version of the u_1-state is readily decomposed into the f_1-state from before and into a new state, i_1 (Figure 4.5). The i_1-state refers to the equivalent current source, $\vec{J}_M^{i_1}$, radiating into the grounded dielectric slab environment.

The operator equations applicable to the i_1-state may now be obtained by realizing that it has no equivalent magnetic currents. From (3.45), we have

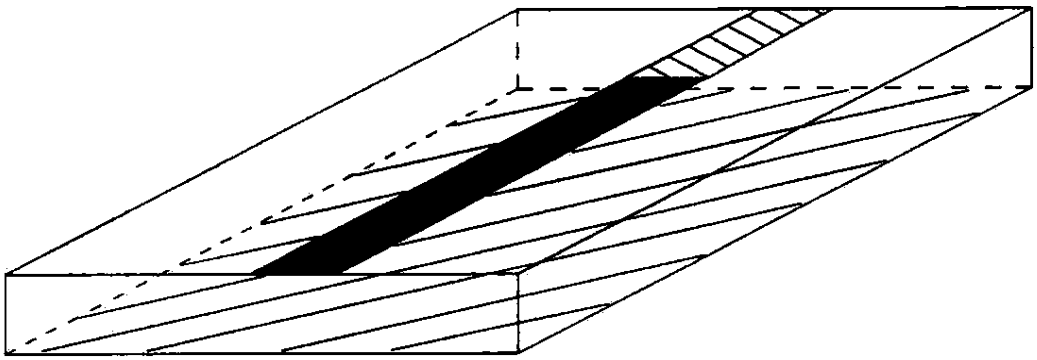


Figure 4.5: The i_1-state

$$\left.\begin{array}{rcccll} M_2\vec{J}_M^{i_1} & = & \vec{E}_p^{i_1} & = & \vec{E}_p^{u_1} - \vec{E}_p^{f_1} & (z=d) \\ -L\vec{J}_M^{i_1} & = & \vec{J}_G^{i_1} & = & \vec{J}_G^{u_1} - \vec{J}_G^{f_1} & (z=0) \end{array}\right\} \tag{4.17}$$

$\vec{J}_G^{u_1}$ is just the ground plane current of the fundamental microstrip mode, that is

$$\vec{J}_G^{u_1} = \vec{J}_G^{+} = \vec{a}_z \times \vec{H}_{M+}^{+}\Big|_{z=0} \tag{4.18}$$

The last result is due to the fact that there are no fields below the ground plane, i.e., $\vec{H}_{M-}^{+} = 0$. The second set of equalities in (4.17) follow from the state superposition $u_1 = i_1 + f_1$. We now have the ability to eliminate the f_1-state quantities by rewriting them in terms of fundamental microstrip mode quantities. Since $\vec{E}_p^{u_1} = 0$ in S_M,

$$\left.\begin{array}{rcll} \vec{E}_p^{f_1} & = & -M_2\vec{J}_M^{i_1} & (\text{in } S_M) \\ \vec{J}_G^{f_1} & = & L\vec{J}_M^{i_1} + \vec{J}_G^{+} & (\text{in } S_A) \end{array}\right\} \tag{4.19}$$

Introducing the definitions,

$$\left.\begin{aligned} \vec{E}^{1}_{S_M} &= M_2 \vec{J}^{i_1}_{M} \\ \vec{J}^{1}_{S_A} &= -L\vec{J}^{i_1}_{M} - \vec{J}^{+}_{G} \end{aligned}\right\} \tag{4.20}$$

allows us to write (4.13) in the form

$$\langle f_1, d_1 \rangle = (\vec{E}^{d_1}_{A}, \vec{J}^{1}_{S_A})_{S_A} - (\vec{J}^{d_1}_{M}, \vec{E}^{1}_{S_M})_{S_M} \tag{4.21}$$

This is exactly the kind of inner product found in (3.51). Thus, $\langle f_1, d_1 \rangle$ can be variationally approximated by using the functional R from (3.59). This means that the first term of (4.8) is now in a computable form.

The second term of (4.8) may be rewritten in a similar manner. Since the source plane represents a discontinuity in the fields, the reaction will need to be taken as a limit. The surface integrals will be taken with quantities just on the negative side of S_1 $(x < -w_A/2)$. This side will be denoted S_1- and the other side of S_1 will be denoted S_1+ $(x > -w_A/2)$. The reaction $\langle f_1, f_1 \rangle$ may be written as

$$\langle f_1, f_1 \rangle = \int_{S_1} \left(\vec{J}^{1}_{M} \cdot \vec{E}^{f_1}_{S_1-} - \vec{M}^{1}_{M} \cdot \vec{H}^{f_1}_{S_1-} \right) dS \tag{4.22}$$

Using (4.1) and remembering that $\vec{J}^{1}_{M} = \vec{a}_x \times (\vec{H}^{f_1}_{S_1+} - \vec{H}^{f_1}_{S_1-})$ and $\vec{M}^{1}_{M} = -\vec{a}_x \times (\vec{E}^{f_1}_{S_1+} - \vec{E}^{f_1}_{S_1-})$, yields

$$\langle f_1, f_1 \rangle = \int_{S_1} \left[\vec{J}^{1}_{M} \cdot \left(\vec{E}^{f_1}_{S_1+} - \vec{E}^{+}_{M} \right) - \vec{M}^{1}_{M} \cdot \left(\vec{H}^{f_1}_{S_1+} - \vec{H}^{+}_{M} \right) \right] dS \tag{4.23}$$

But since $\vec{E}^{+}_{M} = \vec{E}^{u_1}_{M}$, $\vec{H}^{+}_{M} = \vec{H}^{u_1}_{M}$, and $u_1 = i_1 + f_1$, (4.23) is just equal to the reaction $-\langle i_1, f_1 \rangle$. Using Lorentz reciprocity, (4.9), gives the new relationship

$$\langle f_1, f_1 \rangle = -\langle f_1, i_1 \rangle = -(\vec{E}^{f_1}_{p}, \vec{J}^{i_1}_{M})_{S_M} \tag{4.24}$$

Using the definition (4.20) and (4.19), this can be written as

$$\langle f_1, f_1 \rangle = (\vec{E}^{1}_{S_M}, \vec{J}^{i_1}_{M})_{S_M} \tag{4.25}$$

Finally, combining the results (4.25) and (4.21) with (4.8), (4.5), and (4.3), allows us to write Γ_M as

$$-2N_M\Gamma_M = (\vec{E}^{d_1}_{A}, \vec{J}^{1}_{S_A})_{S_A} - (\vec{J}^{d_1}_{M} - \vec{J}^{i_1}_{M}, \vec{E}^{1}_{S_M})_{S_M} \tag{4.26}$$

The expression (4.26) is converted to a variational form using (3.59) and (4.20)

$$\begin{aligned} -2N_M\Gamma_M &= (M_2[\vec{J}^{d_1}_{M} - \vec{J}^{i_1}_{M}], \vec{J}^{d_1}_{M} - \vec{J}^{i_1}_{M})_{S_M} - (M_1\vec{E}^{d_1}_{A}, \vec{E}^{d_1}_{A})_{S_A} + \\ &\quad 2(\vec{E}^{d_1}_{A}, L[\vec{J}^{d_1}_{M} - \vec{J}^{i_1}_{M}] - \vec{J}^{+}_{G})_{S_A} \end{aligned} \tag{4.27}$$

Using s_1 to refer to scattered fields and currents, this becomes

$$\begin{aligned} -2N_M\Gamma_M &= (M_2\vec{J}^{s_1}_{M}, \vec{J}^{s_1}_{M})_{S_M} - (M_1\vec{E}^{s_1}_{A}, \vec{E}^{s_1}_{A})_{S_A} + \\ &\quad 2(\vec{E}^{s_1}_{A}, L\vec{J}^{s_1}_{M} - \vec{J}^{+}_{G})_{S_A} \end{aligned} \tag{4.28}$$

One last step remains before we can compute the required inner products and the reflection coefficient. The scattered fields and currents have unknown amplitude coefficients which we must still eliminate. This is done by applying the Ritz-Galerkin procedure. It entails finding the stationary value of Γ_M with respect to the two unknown amplitudes.

4.1.4 Ritz-Galerkin Procedure Let the peak amplitude of the scattered slotline electric field be C_{A_1} and that of the scattered microstrip current C_{M_1}. We can then write (4.28) in terms of the following new variables

$$-2N_M\Gamma_M = C^2_{M_1} I_{MM} + 2C_{A_1}C_{M_1} I_{AM} - 2C_{A_1} I_{AG} - C^2_{A_1} I_{AA} \tag{4.29}$$

The modified inner products are defined as

$$\left.\begin{aligned} I_{MM} &= (M_2\vec{J}_M^{s_1}, \vec{J}_M^{s_1})_{S_M}/C_{M_1}^2 \\ I_{AA} &= (M_1\vec{E}_A^{s_1}, \vec{E}_A^{s_1})_{S_A}/C_{A_1}^2 \\ I_{AM} &= (L\vec{J}_M^{s_1}, \vec{E}_A^{s_1})_{S_A}/(C_{A_1}C_{M_1}) \\ I_{AG} &= (\vec{E}_A^{s_1}, \vec{J}_G^{+})_{S_A}/C_{A_1} \end{aligned}\right\} \tag{4.30}$$

To find the stationary values of Γ_M, we first differentiate Γ_M with respect to C_{A_1} and C_{M_1} and set the result equal to zero. One obtains

$$\left.\begin{aligned} -2N_M\frac{\partial\Gamma_M}{\partial C_{M_1}} &= 2C_{M_1}I_{MM} + 2C_{A_1}I_{AM} &&= 0 \\ -2N_M\frac{\partial\Gamma_M}{\partial C_{A_1}} &= 2C_{M_1}I_{AM} - 2I_{AG} - 2C_{A_1}I_{AA} &&= 0 \end{aligned}\right\} \tag{4.31}$$

These are now solved for C_{A_1} and C_{M_1}. The solution is

$$\left.\begin{aligned} C_{A_1} &= \frac{-I_{MM}I_{AG}}{(I_{AM}^2 + I_{MM}I_{AA})} \\ C_{M_1} &= \frac{I_{AM}I_{AG}}{(I_{AM}^2 + I_{MM}I_{AA})} \end{aligned}\right\} \tag{4.32}$$

With C_{A_1} and C_{M_1} determined, Γ_M may be written in its final form

$$\Gamma_M = \frac{-I_{MM}I_{AG}^2}{2N_M(I_{AM}^2 + I_{MM}I_{AA})} \tag{4.33}$$

This is a rather innocent looking expression, but it still requires a lot of numerical work before it will yield answers in a reasonable amount of time.

4.2 Microstrip Transmission Coefficient

The microstrip transmission coefficient is derived in a similar manner as the microstrip reflection coefficient. However in this case, we will need

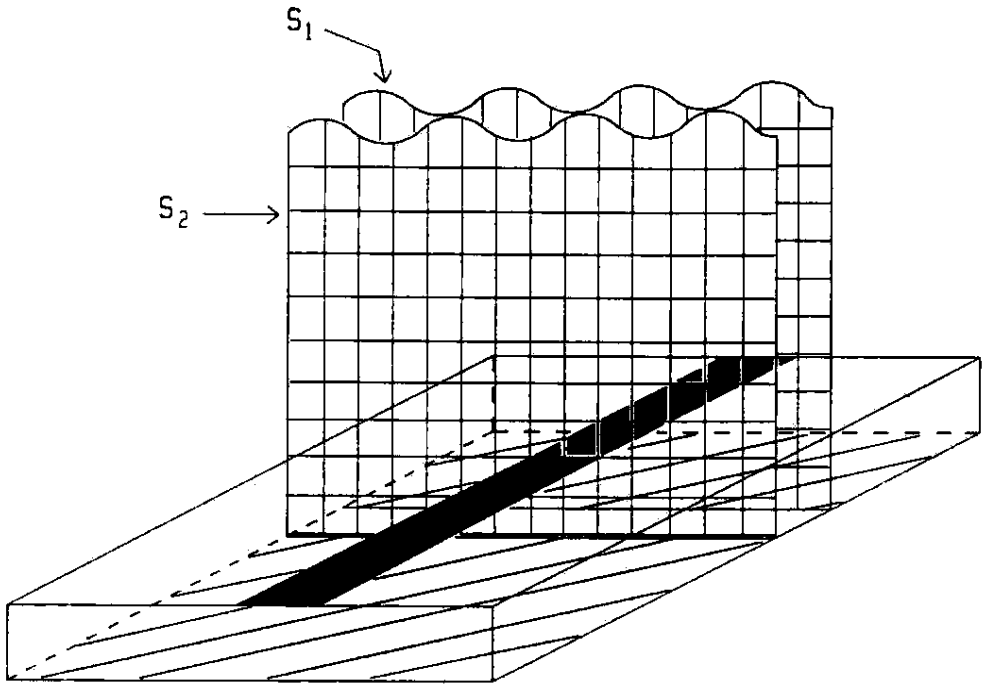


Figure 4.6: Superposition of the t_1 and t_2 states

to work with two sets of microstrip mode sources simultaneously. One will launch the electromagnetic fields. The other one, located across the junction, will pick out the transmitted microstrip mode. Figure 4.6 shows the two surfaces on which these sources are located. As in Section 4.1, sources $\vec{J}_M^1$ and $\vec{M}_M^1$ launch a forward-traveling dominant microstrip mode. The second set of sources, $\vec{J}_M^2$ and $\vec{M}_M^2$ which are distributed on S_2, launch a backward-traveling dominant microstrip mode. S_2 is located at $x = w_A/2$. The sources are defined as in (4.1)

$$\left.\begin{aligned} \vec{J}_M^2 &= -\vec{a}_x \times \vec{\mathcal{H}}_M^- e^{i\beta_M w_A/2} \\ \vec{M}_M^2 &= \vec{a}_x \times \vec{\mathcal{E}}_M^- e^{i\beta_M w_A/2} \end{aligned}\right\} \tag{4.34}$$

These sources can also be written in terms of the transverse field distributions of the forward-traveling mode by using (2.5) and by realizing that the longitudinal components do not appear in the result due to the cross product with $\vec{a}_x$

$$\left.\begin{aligned} \vec{J}_M^{\,2} &= \vec{a}_x \times \vec{\mathcal{H}}_M^{+}\, e^{i\beta_M w_A/2} \\ \vec{M}_M^{\,2} &= \vec{a}_x \times \vec{\mathcal{E}}_M^{+}\, e^{i\beta_M w_A/2} \end{aligned}\right\} \tag{4.35}$$

As before, the total electromagnetic fields may be written as a sum of the fundamental microstrip mode plus higher-order terms, except that this time the t_1 fields are expanded in terms of the forward-traveling mode spectrum

$$\left.\begin{aligned} \vec{E}^{\,t_1} &= T_M \vec{\mathcal{E}}_M^{+}\, e^{-i\beta_M w_A/2} + \cdots \\ \vec{H}^{\,t_1} &= T_M \vec{\mathcal{H}}_M^{+}\, e^{-i\beta_M w_A/2} + \cdots \end{aligned}\right\} \tag{4.36}$$

The transmitted microstrip mode can be picked out from the total scattered field by mode orthogonality. But this time we must use the second set of sources as the probes. Associated with $\vec{J}_M^{\,2}$ and $\vec{M}_M^{\,2}$ will be five new states t_2, d_2, f_2, u_2, and i_2. These are exactly analogous to the five states from Section 4.1. The difference lies with the specific values of the sources and the fields, but the geometries and the types of sources used to excite them remain the same. Thus, the derivation for T_M relies heavily on equations already derived.

Using (4.35) and (4.36), we can pick out T_M by forming the reaction $\langle t_1, f_2 \rangle$

$$T_M = -\frac{1}{2N_M} \int_{S_2} \left(\vec{J}_M^{\,2} \cdot \vec{E}^{\,t_1} - \vec{M}_M^{\,2} \cdot \vec{H}^{\,t_1} \right) dS \tag{4.37}$$

We will need to separate $\langle t_1, f_2 \rangle$ into secondary reactions as was done in (4.8). (4.37) is split analogous to (4.8)

$$\langle t_1, f_2 \rangle = \langle d_1, f_2 \rangle + \langle f_1, f_2 \rangle \tag{4.38}$$

Using Lorentz reciprocity (4.9) and equations analogous to (4.11) and (4.12)

for the f_2-state, yields

$$\langle d_1, f_2 \rangle = \langle f_2, d_1 \rangle = (\vec{J}_M^{\,d_1}, \vec{E}^{\,f_2})_{S_M} - (\vec{E}_A^{\,d_1}, \vec{J}_G^{\,f_2})_{S_A} \tag{4.39}$$

The f_2-state quantities can be eliminated by using the appropriate version of (4.19)

$$\left.\begin{aligned} \vec{E}_p^{\,f_2} &= -M_2 \vec{J}_M^{\,i_2} \qquad (\text{in } S_M) \\ \vec{J}_G^{\,f_2} &= L\vec{J}_M^{\,i_2} + \vec{J}_G^{\,u_2} \qquad (\text{in } S_A) \end{aligned}\right\} \tag{4.40}$$

The ground plane current $\vec{J}_G^{\,u_2}$ is given by

$$\vec{J}_G^{\,u_2} = \vec{J}_G^{\,-} = \vec{a}_z \times \vec{H}_{M+}^{-}\Big|_{z=0} \tag{4.41}$$

since $\vec{H}_{M-}^{-} = 0$. Introducing the variables,

$$\left.\begin{aligned} \vec{E}_{S_M}^{\,2} &= M_2 \vec{J}_M^{\,i_2} \\ \vec{J}_{S_A}^{\,2} &= -L\vec{J}_M^{\,i_2} - \vec{J}_G^{\,-} \end{aligned}\right\} \tag{4.42}$$

allows us to write (4.39) in the form

$$\langle d_1, f_2 \rangle = (\vec{E}_A^{\,d_2}, \vec{J}_{S_A}^{\,2})_{S_A} - (\vec{J}_M^{\,d_2}, \vec{E}_{S_M}^{\,2})_{S_M} \tag{4.43}$$

This is again the kind of inner product found in (3.51). Thus, $\langle d_1, f_2 \rangle$ can be approximated by using the functional R_{12} from (3.57).

The second term of (4.38) is rewritten as follows. The reaction $\langle f_1, f_2 \rangle$ does not have to be stated as a limit since the f_1-state fields are continuous across S_2. The reaction is defined as

$$\langle f_1, f_2 \rangle = \int_{S_2} \left(\vec{J}_M^{\,2} \cdot \vec{E}_{S_2}^{\,f_1} - \vec{M}_M^{\,2} \cdot \vec{H}_{S_2}^{\,f_1} \right) dS \tag{4.44}$$

Adding the term $2\vec{a}_x \cdot \vec{E}_M^{+} \times \vec{H}_M^{+}$ and using (4.35) and (4.17) yields

$$\langle f_1, f_2 \rangle = -(\vec{E}_p^{\,f_2}, \vec{J}_M^{\,i_1})_{S_M} - 2N_M \tag{4.45}$$

Using the definition (4.42) and (4.40), this can further be written as

$$\langle f_1, f_2 \rangle = (\vec{E}^{\,2}_{S_M}, \vec{J}^{\,i_1}_M)_{S_M} - 2N_M \tag{4.46}$$

After combining (4.37), (4.43) and (4.46)

$$-2N_M T_M = (\vec{E}^{\,d_1}_A, \vec{J}^{\,2}_{S_A})_{S_A} - (\vec{J}^{\,d_1}_M - \vec{J}^{\,i_1}_M, \vec{E}^{\,2}_{S_M})_{S_M} - 2N_M \tag{4.47}$$

Now we convert this expression to its bivariational form. Using (3.57) and (4.42),

$$\begin{aligned} 2N_M(1-T_M) \;=\; & (M_2[\vec{J}^{\,d_2}_M - \vec{J}^{\,i_2}_M], \vec{J}^{\,d_1}_M - \vec{J}^{\,i_1}_M)_{S_M} - (M_1\vec{E}^{\,d_2}_A, \vec{E}^{\,d_1}_A)_{S_A} + \\ & (\vec{E}^{\,d_2}_A, L[\vec{J}^{\,d_1}_M - \vec{J}^{\,i_1}_M])_{S_A} - (\vec{E}^{\,d_2}_A, \vec{J}^{\,+}_G)_{S_A} + \\ & (\vec{E}^{\,d_1}_A, L[\vec{J}^{\,d_2}_M - \vec{J}^{\,i_2}_M])_{S_A} - (\vec{E}^{\,d_1}_A, \vec{J}^{\,-}_G)_{S_A} \end{aligned} \tag{4.48}$$

In terms of scattered fields and currents this becomes

$$\begin{aligned} 2N_M(1-T_M) \;=\; & (M_2\vec{J}^{\,s_2}_M, \vec{J}^{\,s_1}_M)_{S_M} - (M_1\vec{E}^{\,s_2}_A, \vec{E}^{\,s_1}_A)_{S_A} + \\ & (\vec{E}^{\,s_2}_A, L\vec{J}^{\,s_1}_M)_{S_A} - (\vec{E}^{\,s_2}_A, \vec{J}^{\,+}_G)_{S_A} + \\ & (\vec{E}^{\,s_1}_A, L\vec{J}^{\,s_2}_M)_{S_A} - (\vec{E}^{\,s_1}_A, \vec{J}^{\,-}_G)_{S_A} \end{aligned} \tag{4.49}$$

We can now apply the Ritz-Galerkin procedure. First, we write (4.49) in the terms of the inner products (4.30). Using C_{A_1}, C_{A_2}, C_{M_1}, and C_{M_2} for the amplitudes of the scattered fields and currents $\vec{E}^{\,s_1}_A$, $\vec{E}^{\,s_2}_A$, $\vec{J}^{\,s_1}_M$, and $\vec{J}^{\,s_2}_M$, respectively, yields

$$\begin{aligned} 2N_M(1-T_M) \;=\; & C_{M_1}C_{M_2}I_{MM} + (C_{A_2}C_{M_1} + C_{A_1}C_{M_2})I_{AM} - \\ & C_{A_1}C_{A_2}I_{AA} + (C_{A_1} - C_{A_2})I_{AG} \end{aligned} \tag{4.50}$$

The origin of the last term is not obvious. It follows after taking into consideration the precise nature of our approximating functions for $\vec{J}_M$ and $\vec{E}_A$.

$\vec{J}_M$ consists of only a longitudinal current component, while $\vec{E}_A$ consists of only a transverse field component. In conjunction with (4.41) and (2.5), it follows that $\vec{J}_G^{+} = -\vec{J}_G^{-}$.

The stationary values of T_M are found by using the Ritz-Galerkin procedure. The solution for the coefficients is similar to (4.32)

$$\left.\begin{aligned} C_{A_1} &= -C_{A_2} = \frac{-I_{MM}I_{AG}}{(I_{AM}^2 + I_{MM}I_{AA})} \\ C_{M_1} &= -C_{M_2} = \frac{I_{AM}I_{AG}}{(I_{AM}^2 + I_{MM}I_{AA})} \end{aligned}\right\} \tag{4.51}$$

Using these in (4.50), we get

$$T_M = 1 + \frac{I_{MM}I_{AG}^2}{2N_M(I_{AM}^2 + I_{MM}I_{AA})} = 1 - \Gamma_M \tag{4.52}$$

It is not clear whether this simple result would have held if we had used more complicated approximating functions for $\vec{J}_M$ and $\vec{E}_A$. Thus, this derivation is only completely general up to (4.49).

4.3 Slotline Reflection Coefficient

The slotline reflection coefficient is found the same way as the microstrip reflection coefficient. A forward-traveling wave is launched on the slotline using the slotline mode sources, $\vec{J}_A^3$ and $\vec{M}_A^3$. They are defined by

$$\left.\begin{aligned} \vec{J}_A^3 &= \vec{a}_y \times \vec{\mathcal{H}}_A^{+}\, e^{i\beta_A w_M/2} \\ \vec{M}_A^3 &= -\vec{a}_y \times \vec{\mathcal{E}}_A^{+}\, e^{i\beta_A w_M/2} \end{aligned}\right\} \tag{4.53}$$

The slotline mode sources are distributed on a surface S_3 which is located at $y = -w_M/2$. In contrast with S_1, this surface extends all the way along

the z-axis due to the open aperture in the ground plane. However, this does not affect any of the earlier derivations. The t_3-state fields on S_3 can be expanded in terms of the slotline modes as

$$\left.\begin{aligned} \vec{E}^{\,t_3} &= \Gamma_A \vec{\mathcal{E}}_A^{\,-} e^{-i\beta_A w_M/2} + \cdots \\ \vec{H}^{\,t_3} &= \Gamma_A \vec{\mathcal{H}}_A^{\,-} e^{-i\beta_A w_M/2} + \cdots \end{aligned}\right\} \tag{4.54}$$

Using these with (2.5) and mode orthogonality, Γ_A can be expressed as

$$\Gamma_A = -\frac{1}{2N_A} \int_{S_3} \left(\vec{J}_A^{\,3} \cdot \vec{E}^{\,t_3} - \vec{M}_A^{\,3} \cdot \vec{H}^{\,t_3} \right) dS \tag{4.55}$$

The norm appearing in (4.55) as N_A is defined by the following integral

$$N_A = \int_S \vec{a}_y \cdot \left(\vec{\mathcal{E}}_A^{\,+} \times \vec{\mathcal{H}}_A^{\,+} \right) dS \tag{4.56}$$

where S is any surface transverse to the y-axis.

The integral in (4.55) is the reaction $\langle t_3, f_3 \rangle$ and is broken into two parts

$$\langle t_3, f_3 \rangle = \langle d_3, f_3 \rangle + \langle f_3, f_3 \rangle \tag{4.57}$$

As in the derivation leading up to (4.13), the reaction

$$\langle d_3, f_3 \rangle = \langle f_3, d_3 \rangle = (\vec{J}_M^{\,d_3}, \vec{E}^{\,f_3})_{S_M} - (\vec{E}_A^{\,d_3}, \vec{J}_G^{\,f_3})_{S_A} \tag{4.58}$$

This reaction can further be modified by introducing the u_3-state, i.e., where the slotline mode sources radiate into the uniform slotline geometry. The u_3-state can be separated into the two related states f_3 and i_3. The i_3-state corresponds to the slotline electric field, $\vec{E}_A^{\,i_3}$, radiating into the grounded dielectric slab environment. The operator equations applicable to this state can be found directly from (3.45). The same argument leading up to (4.19) allows us to write

$$\left.\begin{aligned} \vec{E}_p^{f_3} &= \vec{E}_A^{u_3} - L\vec{E}_A^{i_3} \qquad (\text{in } S_M) \\ \vec{J}_G^{f_3} &= -M_1\vec{E}_A^{i_3} \qquad (\text{in } S_A) \end{aligned}\right\} \tag{4.59}$$

The variable $\vec{E}_A^{u_3}$ is just the electric field in S_M due to the fundamental slotline mode, or

$$\vec{E}_A^{u_3} = \vec{E}_A^{+} \tag{4.60}$$

Again we introduce variables as in (4.20)

$$\left.\begin{aligned} \vec{E}_{S_M}^{3} &= L\vec{E}_A^{i_3} - \vec{E}_A^{+} \\ \vec{J}_{S_A}^{3} &= M_1\vec{E}_A^{i_3} \end{aligned}\right\} \tag{4.61}$$

which allows us to rewrite (4.58) in the standard form

$$\langle d_3, f_3\rangle = (\vec{E}_A^{d_3}, \vec{J}_{S_A}^{3})_{S_A} - (\vec{J}_M^{d_3}, \vec{E}_{S_M}^{3})_{S_M} \tag{4.62}$$

The second term of (4.57) may be rewritten in a similar manner. Since the source plane represents a discontinuity in the fields, the reaction will need to be taken as a limit. The surface integrals will be taken with quantities just on the negative side of S_3 $(y < -w_M/2)$. This side will be denoted S_3- and the other side of S_3 will be denoted S_3+ $(y > -w_M/2)$. Now the reaction $\langle f_3, f_3\rangle$ may be written as

$$\langle f_3, f_3\rangle = \int_{S_3} \left(\vec{J}_A^{3} \cdot \vec{E}_{S_3-}^{f_3} - \vec{M}_A^{3} \cdot \vec{H}_{S_3-}^{f_3}\right) dS \tag{4.63}$$

Using (4.53) and remembering that $\vec{J}_A^{3} = \vec{a}_y \times (\vec{H}_{S_3+}^{f_3} - \vec{H}_{S_3-}^{f_3})$ and $\vec{M}_A^{3} = -\vec{a}_y \times (\vec{E}_{S_3+}^{f_3} - \vec{E}_{S_3-}^{f_3})$, yields

$$\langle f_3, f_3\rangle = \int_{S_3} \left[\vec{J}_A^{3} \cdot \left(\vec{E}_{S_3+}^{f_3} - \vec{E}_A^{+}\right) - \vec{M}_A^{3} \cdot \left(\vec{H}_{S_3+}^{f_3} - \vec{H}_A^{+}\right)\right] dS \tag{4.64}$$

But since $u_3 = i_3 + f_3$, $\vec{E}_A^+ = \vec{E}_A^{u_1}$, and $\vec{H}_A^+ = \vec{H}_A^{u_1}$, (4.64) is just the reaction $-\langle i_3, f_3 \rangle$. Lorentz reciprocity gives the new relationship

$$\langle f_3, f_3 \rangle = -\langle f_3, i_3 \rangle \tag{4.65}$$

The last reaction is readily converted from

$$\langle f_3, i_3 \rangle = -\int_{S_A} \left(\vec{M}_{S_A+}^{i_3} \cdot \vec{H}_{S_A+}^{f_3} + \vec{M}_{S_A-}^{i_3} \cdot \vec{H}_{S_A-}^{f_3} \right) dS \tag{4.66}$$

to

$$\langle f_3, i_3 \rangle = -\int_{S_A} \vec{E}_A^{i_3} \cdot \vec{J}_G^{f_3} \, dS = -(\vec{E}_A^{i_3}, \vec{J}_G^{f_3})_{S_A} \tag{4.67}$$

This follows from the definitions of the currents in terms of the fields analogous to (4.7). Thus, (4.65) becomes

$$\langle f_3, f_3 \rangle = (\vec{E}_A^{i_3}, \vec{J}_G^{f_3})_{S_A} \tag{4.68}$$

Using the definition (4.61) and (4.59), this can be written as

$$\langle f_3, f_3 \rangle = -(\vec{E}_A^{i_3}, \vec{J}_{S_A}^3)_{S_A} \tag{4.69}$$

Combining (4.69) with (4.62), (4.57), and (4.55), results in

$$-2N_A\Gamma_A = (\vec{E}_A^{d_3} - \vec{E}_A^{i_3}, \vec{J}_{S_A}^3)_{S_A} - (\vec{J}_M^{d_3}, \vec{E}_{S_M}^3)_{S_M} \tag{4.70}$$

To convert this to a variational form, we use the functional R from (3.59) and the definitions (4.61). This results in

$$\begin{aligned} -2N_A\Gamma_A \;=\; & (M_2\vec{J}_M^{d_3}, \vec{J}_M^{d_3})_{S_M} - (M_1[\vec{E}_A^{d_3} - \vec{E}_A^{i_3}], \vec{E}_A^{d_3} - \vec{E}_A^{i_3})_{S_A} + \\ & 2([\vec{E}_A^{d_3} - \vec{E}_A^{i_3}], L\vec{J}_M^{d_3})_{S_A} + 2(\vec{J}_M^{d_3}, \vec{E}_A^+)_{S_M} \end{aligned} \tag{4.71}$$

or in terms of scattered fields and currents

$$\begin{aligned} -2N_A\Gamma_A \;=\; & (M_2\vec{J}_M^{s_3}, \vec{J}_M^{s_3})_{S_M} - (M_1\vec{E}_A^{s_3}, \vec{E}_A^{s_3})_{S_A} + \\ & 2(\vec{E}_A^{s_3}, L\vec{J}_M^{s_3})_{S_A} + 2(\vec{J}_M^{s_3}, \vec{E}_A^+)_{S_M} \end{aligned} \tag{4.72}$$

Using similar amplitude coefficients for the scattered fields and currents that we introduced in the previous sections, allows us to write (4.72) as

$$-2N_A\Gamma_A = C_{M_3}^2 I_{MM} + 2C_{A_3}C_{M_3}I_{AM} - 2C_{M_3}I_{MI} - C_{A_3}^2 I_{AA} \tag{4.73}$$

We have introduced the new modified inner product

$$I_{MI} \;=\; (\vec{J}_M^{\,s_3}, \vec{E}_A^{\,+})_{S_M}/C_{M_3} \tag{4.74}$$

Applying the Ritz-Galerkin procedure to the result (4.73), yields the final version for Γ_A:

$$\Gamma_A = \frac{I_{AA}I_{MI}^2}{2N_A(I_{AM}^2 + I_{MM}I_{AA})}. \tag{4.75}$$

4.4 Slotline Transmission Coefficient

To pick out the slotline transmission coefficient, we introduce a fourth set of mode sources, $\vec{J}_A^{\,4}$ and $\vec{M}_A^{\,4}$. They are defined by

$$\left.\begin{aligned} \vec{J}_A^{\,4} &= -\vec{a}_y \times \vec{\mathcal{H}}_A^{-}\, e^{i\beta_A w_M/2} \\ \vec{M}_A^{\,4} &= \quad \vec{a}_y \times \vec{\mathcal{E}}_A^{-}\, e^{i\beta_A w_M/2} \end{aligned}\right\} \tag{4.76}$$

The t_3-state fields on S_4 can be expanded in terms of the forward-traveling slotline modes as

$$\left.\begin{aligned} \vec{E}^{\,t_3} &= T_A \vec{\mathcal{E}}_A^{\,+}\, e^{-i\beta_A w_M/2} + \cdots \\ \vec{H}^{\,t_3} &= T_A \vec{\mathcal{H}}_A^{\,+}\, e^{-i\beta_A w_M/2} + \cdots \end{aligned}\right\} \tag{4.77}$$

From mode orthogonality,

$$T_A = -\frac{1}{2N_A}\int_{S_4} \left(\vec{J}_A^{\,4}\cdot\vec{E}^{\,t_3} - \vec{M}_A^{\,4}\cdot\vec{H}^{\,t_3}\right) dS \tag{4.78}$$

The analysis proceeds as in Section 4.2. Two intermediate results will be stated explicitly, the analogues to (4.59) and (4.61). They are

$$\left.\begin{aligned} \vec{E}_p^{\,f_4} &= \vec{E}_A^{\,u_4} - L\vec{E}_A^{\,i_4} \qquad (\text{in } S_M) \\ \vec{J}_G^{\,f_4} &= -M_1\vec{E}_A^{\,i_4} \qquad (\text{in } S_A) \end{aligned}\right\} \tag{4.79}$$

and

$$\left.\begin{aligned} \vec{E}_{S_M}^{\,4} &= L\vec{E}_A^{\,i_4} - \vec{E}_A^{\,u_4} \\ \vec{J}_{S_A}^{\,4} &= M_1\vec{E}_A^{\,i_4} \end{aligned}\right\} \tag{4.80}$$

We note that $\vec{E}_A^{\,u_4} = \vec{E}_A^{\,-}$. The analogue to (4.47) is

$$-2N_AT_A = (\vec{E}_A^{\,d_3} - \vec{E}_A^{\,i_3}, \vec{J}_{S_A}^{\,4})_{S_A} - (\vec{J}_M^{\,d_3}, \vec{E}_{S_M}^{\,4})_{S_M} - 2N_A \tag{4.81}$$

With (3.57), (4.81), and (4.80), this turns into the bivariational result

$$\begin{aligned} 2N_A(1-T_A) &= (M_2\vec{J}_M^{\,s_4}, \vec{J}_M^{\,s_3})_{S_M} - (M_1\vec{E}_A^{\,s_4}, \vec{E}_A^{\,s_3})_{S_A} + \\ &\quad (\vec{E}_A^{\,s_4}, L\vec{J}_M^{\,s_3})_{S_A} + (\vec{J}_M^{\,s_4}, \vec{E}_A^{\,+})_{S_M} + \\ &\quad (\vec{E}_A^{\,s_3}, L\vec{J}_M^{\,s_4})_{S_A} + (\vec{J}_M^{\,s_3}, \vec{E}_A^{\,-})_{S_M} \end{aligned} \tag{4.82}$$

The superscript s refers to scattered currents and fields.

Once again, we need to make use of the fact that our approximating functions consist of longitudinal currents and transverse electric fields. In combination with (2.5), this allows us to equate the two inner products involving the incident slotline field. Of course, the amplitude coefficients, C_{M_3} and C_{M_4}, are not equal and need to be divided out.

$$(\vec{J}_M^{\,s_3}, \vec{E}_A^{\,-})_{S_M}/C_{M_3} = (\vec{J}_M^{\,s_4}, \vec{E}_A^{\,+})_{S_M}/C_{M_4} \tag{4.83}$$

With (4.30) and (4.74), equation (4.82) can be written as

$$\begin{aligned} 2N_A(1-T_A) &= C_{M_3}C_{M_4}I_{MM} + (C_{A_4}C_{M_3} + C_{A_3}C_{M_4})I_{AM} - \\ &\quad C_{A_3}C_{A_4}I_{AA} + (C_{M_3} - C_{M_4})I_{MI} \end{aligned} \tag{4.84}$$

We apply the Ritz-Galerkin procedure. The solution for the stationary values of T_M is similar to (4.51)

$$\left.\begin{aligned} C_{A_3} &= C_{A_4} = \frac{-I_{AM}I_{MI}}{(I_{AM}^2 + I_{MM}I_{AA})} \\ C_{M_3} &= C_{M_4} = \frac{-I_{AA}I_{MI}}{(I_{AM}^2 + I_{MM}I_{AA})} \end{aligned}\right\} \tag{4.85}$$

Using these results in (4.84), yields the final result for T_A

$$T_A = 1 + \frac{I_{AA}I_{MI}^2}{2N_A(I_{AM}^2 + I_{MM}I_{AA})} = 1 + \Gamma_A \tag{4.86}$$

4.5 Coupling Coefficient

The coupling coefficient is found by superposing the t_1 and t_4 states as indicated in Figure 4.7. The analysis could be done either with the t_1 or the t_4-state sources acting as the probes. The result will be the same. One of the issues which needs to be considered here has to do with the intersection of the two source planes along the line $(-w_A/2,\, w_M/2,\, z)$. In this case, Lorentz reciprocity and the equivalence principle are no longer valid, and we have to use a limiting argument. We will let the source planes recede to infinity along the x and y axes. The intersection point thus moves away from the junction. Since the microstrip and slotline mode fields decay along the transverse direction, the equivalent mode sources will also be zero at infinity. Hence they will not interfere with each other.

The total fields due to the t_1 sources may be written in terms of the backward-traveling slotline modes:

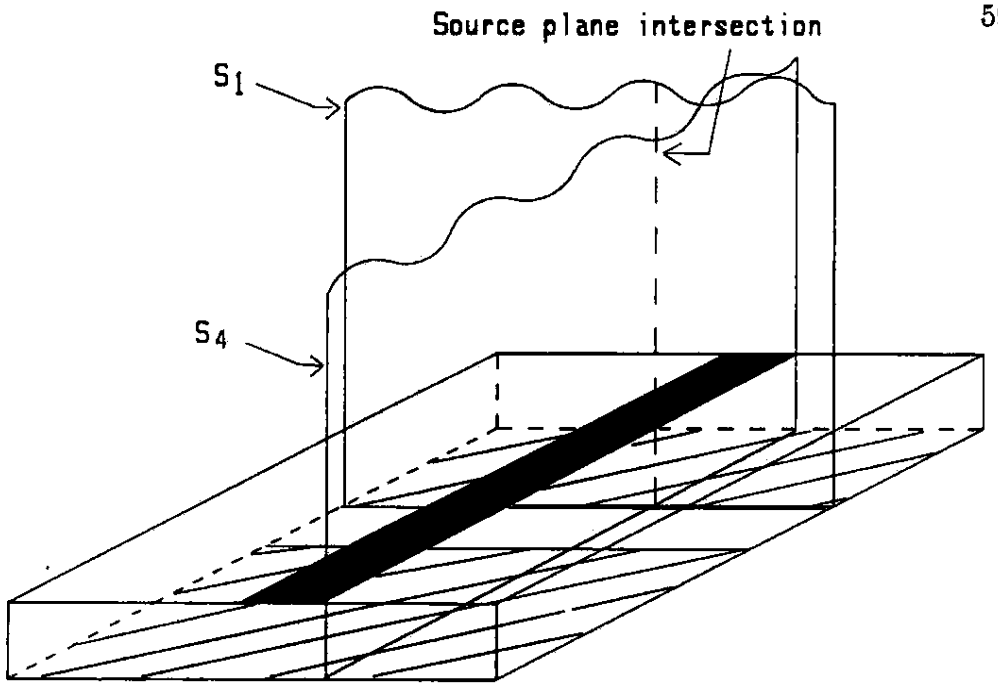


Figure 4.7: Superposition of the t_1 and t_4 states

$$\left.\begin{aligned} \vec{E}^{t_1} &= C'\vec{\mathcal{E}}_A^- e^{-i\beta_A y} + \cdots \\ \vec{H}^{t_1} &= C'\vec{\mathcal{H}}_A^- e^{-i\beta_A y} + \cdots \end{aligned}\right\} \tag{4.87}$$

The amplitude coefficient C' does not correspond to the desired coupling coefficient directly because the incident mode fields are normalized differently for the microstrip than for the slotline. In Appendixes A and B, we made use of the normalization condition that the microstrip carries 1 A of current and the slotline voltage is 1 V. This means that the incident microstrip mode carries total power

$$P_i = \frac{Z_M}{2} \tag{4.88}$$

Since C' represents the scattered slot voltage, the scattered power is

$$P_s = \frac{C'^2}{2Z_A} \tag{4.89}$$

Thus, the coupling coefficient of the scattering matrix is actually given by

$$C = \sqrt{\frac{P_s}{P_t}} = \frac{C'}{\sqrt{Z_A Z_M}} \tag{4.90}$$

With this bit of background, the total fields may be expressed as

$$\left.\begin{aligned} \vec{E}^{\,t_1} &= \frac{C}{\sqrt{Z_A Z_M}} \vec{\mathcal{E}}_A^{\,-} e^{-i\beta_A y} + \cdots \\ \vec{H}^{\,t_1} &= \frac{C}{\sqrt{Z_A Z_M}} \vec{\mathcal{H}}_A^{\,-} e^{-i\beta_A y} + \cdots \end{aligned}\right\} \tag{4.91}$$

The coupling coefficient can be picked out from the total fields with the reaction $\langle t_1, f_4 \rangle$. We also use the fact that $N_A = 1/Z_A$ to derive

$$C = -\sqrt{\frac{Z_A}{4Z_M}} \int_{S_3} \left(\vec{J}_A^{\,4} \cdot \vec{E}^{\,t_1} - \vec{M}_A^{\,4} \cdot \vec{H}^{\,t_1} \right) dS \tag{4.92}$$

We split the reaction into two parts:

$$\langle t_1, f_4 \rangle = \langle d_1, f_4 \rangle + \langle f_1, f_4 \rangle \tag{4.93}$$

Proceeding as we did to obtain (4.43), we find

$$\langle d_1, f_4 \rangle = (\vec{E}_A^{\,d_1}, \vec{J}_{S_A}^{\,4})_{S_A} - (\vec{J}_M^{\,d_1}, \vec{E}_{S_M}^{\,4})_{S_M} \tag{4.94}$$

The reaction $\langle f_1, f_4 \rangle$ can be related to $\langle i_1, f_4 \rangle$ by subtracting the term

$$\int_{S_4} \vec{a}_y \cdot \left(\vec{\mathcal{E}}_M^{\,+} \times \vec{\mathcal{H}}_A^{\,-} - \vec{\mathcal{E}}_A^{\,-} \times \vec{\mathcal{H}}_M^{\,+} \right) dS$$

from $\langle f_1, f_4 \rangle$. This term vanishes when we let the source planes recede to infinity. After applying Lorentz reciprocity, we obtain

$$\langle f_1, f_4 \rangle = \langle f_4, i_1 \rangle = -(\vec{E}_{S_M}^{\,4}, \vec{J}_M^{\,i_1})_{S_M} \tag{4.95}$$

Combining this with (4.94), (4.93), and (4.92) results in

$$-2C\sqrt{Z_M/Z_A} = (\vec{E}_A^{\,d_1}, \vec{J}_{S_A}^{\,4})_{S_A} - (\vec{J}_M^{\,d_1} - \vec{J}_M^{\,i_1}, \vec{E}_{S_M}^{\,4})_{S_M} \tag{4.96}$$

We apply (3.57), (4.20), and (4.80) to convert this to its bivariational form

$$\begin{aligned} -2\mathcal{C}\sqrt{Z_M/Z_A} \;=\; & (M_2\vec{J}_M^{s_1}, \vec{J}_M^{s_4})_{S_M} - (M_1\vec{E}_A^{s_1}, \vec{E}_A^{s_4})_{S_A} + \\ & (\vec{E}_A^{s_4}, L\vec{J}_M^{s_1})_{S_A} + (\vec{E}_A^{s_1}, L\vec{J}_M^{s_4})_{S_A} - \\ & (\vec{E}_A^{s_4}, \vec{J}_G^{+})_{S_A} - (\vec{E}_A^{i_4}, \vec{J}_G^{+})_{S_A} + \\ & (\vec{J}_M^{s_1}, \vec{E}_A^{-})_{S_M} \end{aligned} \tag{4.97}$$

We can shorten this by using the definitions (4.30) and (4.74) and the amplitude coefficients of the scattered fields and currents to

$$\begin{aligned} -2\mathcal{C}\sqrt{Z_M/Z_A} \;=\; & C_{M_1}C_{M_4}I_{MM} + (C_{A_4}C_{M_1} + C_{A_1}C_{M_4})I_{AM} - \\ & C_{A_1}C_{A_4}I_{AA} - C_{A_4}I_{AG} - I'_{AG} + C_{M_1}I_{MI} \end{aligned} \tag{4.98}$$

The modified inner product I'_{AG} is given by

$$I'_{AG} = (\vec{E}_A^{i_4}, \vec{J}_G^{+})_{S_A} = (\vec{E}_A^{-}, \vec{J}_G^{+})_{S_A} \tag{4.99}$$

After applying the Ritz-Galerkin procedure to (4.98), one obtains the following solutions for the amplitude coefficients

$$\left.\begin{aligned} C_{A_1} &= \frac{-I_{MM}I_{AG}}{I_{AM}^2 + I_{MM}I_{AA}} \\ C_{M_1} &= \frac{I_{MM}I_{AG}}{I_{AM}^2 + I_{MM}I_{AA}} \\ C_{A_4} &= \frac{-I_{AM}I_{MI}}{I_{AM}^2 + I_{MM}I_{AA}} \\ C_{M_4} &= \frac{I_{AA}I_{MI}}{I_{AM}^2 + I_{MM}I_{AA}} \end{aligned}\right\} \tag{4.100}$$

Using these in (4.98), the final solution for $\mathcal{C}$ is obtained:

$$\mathcal{C} = \sqrt{\frac{Z_A}{4Z_M}}\left(I'_{AG} - \frac{I_{AM}I_{AG}I_{MI}}{I_{AM}^2 + I_{MM}I_{AA}}\right) \tag{4.101}$$

CHAPTER 5

EXPLICIT FORM OF THE INNER PRODUCTS

In Chapter 4, formulas for the five independent scattering parameters for the microstrip-slotline crossover were derived in terms of six inner products. These inner products need to be evaluated in order to obtain useful answers for the desired S-matrix. This chapter describes the approach used to evaluate the inner products and the resulting two-dimensional integrals. It will then discuss the difficulties encountered in evaluating the so-called Sommerfeld integrals.

5.1 The Inner Products in Detail

We will start by recalling the inner products defined in Chapter 4:

$$\left.\begin{aligned} I_{MM} &= (M_2\vec{J}_M^{s_1}, \vec{J}_M^{s_1})_{S_M}/C_{M_1}^2 \\ I_{AA} &= (M_1\vec{E}_A^{s_1}, \vec{E}_A^{s_1})_{S_A}/C_{A_1}^2 \\ I_{AM} &= (L\vec{J}_M^{s_1}, \vec{E}_A^{s_1})_{S_A}/(C_{A_1}C_{M_1}) \\ I_{MI} &= (\vec{J}_M^{s_3}, \vec{E}_A^{+})_{S_M}/C_{M_3} \\ I_{AG} &= (\vec{E}_A^{s_1}, \vec{J}_G^{+})_{S_A}/C_{A_1} \\ I'_{AG} &= (\vec{E}_A^{i_4}, \vec{J}_G^{+})_{S_A} \end{aligned}\right\} \qquad (5.1)$$

We cannot evaluate these in the spatial domain directly, but have to convert them to spectral domain integrals instead. This is done in the following fashion. Since by definition, one of the elements in the inner product is zero

outside of the surface S, we can readily extend the limits of integration to infinity. Using notation similar to (3.46) and our inverse Fourier transform relation (3.7), we can write the general result

$$(\vec{f},\vec{g})_S = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{\vec{f}} \cdot \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \vec{g}\, dx\, dy\right] e^{-ik_0\vec{\gamma}\cdot\vec{\rho}}\, d\alpha\, d\lambda \tag{5.2}$$

By regrouping the terms in the integrands and applying our Fourier transform definition (3.4), the inner product is readily converted to a doubly infinite integral over the spectral variables α and λ:

$$\begin{aligned}(\vec{f},\vec{g})_S &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{\vec{f}} \cdot \left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \vec{g} e^{-ik_0\vec{\gamma}\cdot\vec{\rho}}\, dx\, dy\right] d\alpha\, d\lambda \\ &= \frac{4\pi^2}{k_0^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{\vec{f}}(\vec{\gamma})\cdot\hat{\vec{g}}(-\vec{\gamma})\, d\alpha\, d\lambda \end{aligned} \tag{5.3}$$

From experience we know that three of the integrals which will be obtained are easier to evaluate by converting them into polar form. This is done with the following transformations

$$\left.\begin{aligned} \alpha &= \gamma\cos\theta \\ \lambda &= \gamma\sin\theta \\ d\alpha\, d\lambda &= \gamma\, d\gamma\, d\theta \end{aligned}\right\} \tag{5.4}$$

so that (5.3) becomes

$$(\vec{f},\vec{g})_S = \frac{4\pi^2}{k_0^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \gamma\, \hat{\vec{f}}(\vec{\gamma})\cdot\hat{\vec{g}}(-\vec{\gamma})\, d\gamma\, d\theta \tag{5.5}$$

We are now ready to apply this result to the inner products in (5.1). We begin with I_{MM}. The currents we transform are scattered currents only, so that using (3.4), the Fourier transform of the propagation factor becomes

$$\frac{k_0}{2\pi}\int_{-\infty}^{\infty} e^{ik_0(\alpha x - n_M|x|)}\, dx = \frac{in_M}{\pi(\alpha^2 - n_M^2)} \tag{5.6}$$

We made use of the limiting absorption principle to achieve a convergent result at the infinite limits (i.e., let $n_M = n'_M - in''_M$ and take the limit $n''_M \to 0$). Recalling our definition (2.2), the FT of the scattered current can now be written as

$$\hat{\vec{J}}_M^{s_1}(\vec{\gamma}) = \frac{in_M C_{M_1}}{\pi(\alpha^2 - n_M^2)} \hat{\vec{\mathcal{J}}}_M^{s_1}(\lambda) \tag{5.7}$$

We will use the same trial functions for the scattered currents as we did for the incident current. This allows us to use (A.28) to write the scattered current as

$$\hat{\vec{J}}_M^{s_1}(\vec{\gamma}) = \frac{in_M k_0 C_{M_1}}{2\pi^2(\alpha^2 - n_M^2)} J_0(k_0 w_M \lambda/2)\vec{a}_x \tag{5.8}$$

We now use the symmetry properties of the current distribution from (A.23) and our operator definition (3.42) in (5.5) to write our first inner product:

$$\begin{aligned} I_{MM} &= -\frac{4i\eta n_M^2}{\pi^2} \int_0^{\pi/2} \int_0^{\infty} \gamma \sinh(k_0 d u_1) \left[\frac{J_0(\bar{\lambda})}{\gamma^2 \cos^2\theta - n_M^2}\right]^2 \\ &\quad \times \left(\frac{u_0 u_1}{D_e}\cos^2\theta - \frac{\mu_r}{D_m}\sin^2\theta\right) d\gamma\, d\theta \end{aligned} \tag{5.9}$$

We have used the fact that $J_0(x)$ is even to reduce the θ-integration to one quarter of the transform space. Also, the variable

$$\bar{\lambda} = \frac{k_0 w_M \lambda}{2} = \frac{k_0 w_M \gamma \sin\theta}{2} \tag{5.10}$$

The integral I_{AA} is found in a similar manner. The trial function for the scattered electric field is the same as that for the incident slot field. The FT of the propagation factor is found analogous to (5.6)

$$\frac{k_0}{2\pi} \int_{-\infty}^{\infty} e^{ik_0(\lambda y - n_A|y|)}\, dy = \frac{in_A}{\pi(\lambda^2 - n_A^2)} \tag{5.11}$$

We will use the same trial functions for the scattered slot fields as we did for the incident fields. This allows us to use (B.31) to write the scattered slot field as

$$\tilde{\vec{E}}_A^{s_1}(\vec{\gamma}) = \frac{in_A k_0 C_{A_1}}{2\pi^2(\lambda^2 - n_A^2)} J_0(k_0 w_A \alpha/2)\vec{a}_x \tag{5.12}$$

so that in conjunction with (B.26) and (3.42), we obtain

$$\begin{aligned} I_{AA} &= -\frac{4in_A^2}{\eta\pi^2}\int_0^{\pi/2}\int_0^{\infty} \frac{\gamma}{\sinh(k_0 d u_1)}\left[\frac{J_0(\bar{\alpha})}{\gamma^2\sin^2\theta - n_A^2}\right]^2 \\ &\quad \times \left(\frac{\epsilon_r^2 u_0^2 - D_e^2}{u_0 u_1 D_e}\cos^2\theta + \frac{D_m^2 - u_1^2}{\mu_r D_m}\sin^2\theta\right) d\gamma\, d\theta \end{aligned} \tag{5.13}$$

We have introduced the new variable

$$\bar{\alpha} = \frac{k_0 w_A \alpha}{2} = \frac{k_0 w_A \gamma\cos\theta}{2} \tag{5.14}$$

Again the trial function for the scattered electric field is the same as that for the incident slot field. Using the quantities derived above and (3.42), the third inner product is readily obtained as

$$\begin{aligned} I_{AM} &= -\frac{4n_A n_M}{\pi^2}\int_0^{\pi/2}\int_0^{\infty} \frac{\gamma J_0(\bar{\alpha})J_0(\bar{\lambda})}{(\gamma^2\sin^2\theta - n_A^2)(\gamma^2\cos^2\theta - n_M^2)} \\ &\quad \times \left(\frac{\epsilon_r u_0}{D_e}\cos^2\theta + \frac{u_1}{D_m}\sin^2\theta\right) d\gamma\, d\theta \end{aligned} \tag{5.15}$$

To find the fourth inner product, I_{MI}, we need to make use of some additional results from Appendixes A and B. Because we have a longitudinal microstrip current for our trial function, we will only need the x-component of $\vec{E}_A^+$. To find the Fourier transform, we make use of the fact that the electric field is zero for $y < -w_M/2$. Using our previous results (B.17), (B.31),

and (B.5), allows us to write the following expression for the transformed electric field in S_M

$$\begin{aligned}\hat{E}_{Ax}^{+} &= \frac{ik_0 J_0(\bar{\alpha})}{4\pi^2(\lambda - n_A)} e^{-ik_0 w_M(\lambda - n_A)/2} \\ &\quad \times \left[\cosh(k_0 d s_1) - \frac{\sinh(k_0 d s_1)}{\alpha^2 + n_A^2}\left(\frac{\alpha^2 R_e}{S_e} + \frac{n_A^2 R_m}{S_m}\right)\right] \qquad (5.16)\end{aligned}$$

Thus,

$$\begin{aligned}\hat{E}_{Ax}^{+}(-\vec{\gamma}) &= \frac{-ik_0 J_0(\bar{\alpha})}{4\pi^2(\lambda + n_A)} e^{ik_0 w_M(\lambda + n_A)/2} \\ &\quad \times \left[\cosh(k_0 d s_1) - \frac{\sinh(k_0 d s_1)}{\alpha^2 + n_A^2}\left(\frac{\alpha^2 R_e}{S_e} + \frac{n_A^2 R_m}{S_m}\right)\right] \qquad (5.17)\end{aligned}$$

We combine this with (5.1), (5.3), and (5.8) to obtain our fourth inner product

$$\begin{aligned}I_{MI} &= \frac{n_M}{2\pi^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{J_0(\bar{\alpha}) J_0(\bar{\lambda})}{(\alpha^2 - n_M^2)(\lambda + n_A)} e^{ik_0 w_M(\lambda + n_A)/2} \\ &\quad \times \left[\cosh(k_0 d s_1) - \frac{\sinh(k_0 d s_1)}{\alpha^2 + n_A^2}\left(\frac{\alpha^2 R_e}{S_e} + \frac{n_A^2 R_m}{S_m}\right)\right] d\alpha\, d\lambda \qquad (5.18)\end{aligned}$$

We did not apply the polar transformation (5.4) here and on the next two inner products because these are easier to solve using the original, rectangular coordinates.

The next inner product I_{AG} requires that we find the ground plane current of the microstrip mode, $\vec{J}_G^{+}$. In order to make use of the boundary condition (4.18), we need to know the longitudinal component of the magnetic field, H_{Mx}^{+}, which was not derived in Appendix A. There we were only

interested in the transverse magnetic field components. H_{Mx}^{+} is derived in the same way as H_{My}^{+}. The result is

$$\hat{H}_{Mx}^{+} = \frac{1}{\chi^2}\left[\frac{\epsilon_r n_M t_0}{T_e}\vec{\chi}\cdot\hat{\vec{J}}_M + \frac{\lambda t_1}{T_m}\vec{\chi}_\perp\cdot\hat{\vec{J}}_M\right]\cosh(k_0 t_1 z) \tag{5.19}$$

Using this and (A.17), allows us to evaluate the FT of the transverse ground plane current distribution as

$$\hat{\vec{J}}_G^{+} = -\frac{1}{\chi^2}\left(\frac{\epsilon_r t_0 \vec{\chi}}{T_e}\vec{\chi}\cdot\hat{\vec{J}}_M + \frac{t_1\vec{\chi}_\perp}{T_m}\vec{\chi}_\perp\cdot\hat{\vec{J}}_M\right)\cosh(k_0 t_1 z) \tag{5.20}$$

By realizing that the ground plane current is zero for $x < -w_A/2$, the FT of the propagation factor becomes

$$\frac{k_0}{2\pi}\int_{-w_A/2}^{\infty} e^{ik_0(\alpha-n_M)x}\,dx = \frac{ie^{-ik_0 w_A(\alpha-n_M)/2}}{2\pi(\alpha-n_M)} \tag{5.21}$$

We now use this with our previous results (A.3)–(A.5) and (A.28) to write the required x-component of the current as

$$\hat{J}_{Gx}^{+} = \frac{-ik_0 J_0(\bar{\lambda})e^{-ik_0 w_A(\alpha-n_M)/2}}{4\pi^2(\lambda^2+n_M^2)(\alpha-n_M)}\left(\frac{\epsilon_r n_M^2 t_0}{T_e}+\frac{\lambda^2 t_1}{T_m}\right) \tag{5.22}$$

Thus,

$$\hat{J}_{Gx}^{+}(-\vec{\gamma}) = \frac{ik_0 J_0(\bar{\lambda})e^{ik_0 w_A(\alpha+n_M)/2}}{4\pi^2(\lambda^2+n_M^2)(\alpha+n_M)}\left(\frac{\epsilon_r n_M^2 t_0}{T_e}+\frac{\lambda^2 t_1}{T_m}\right) \tag{5.23}$$

Combining this with our trial function for the scattered slot field (5.12) and (5.3), yields the fifth inner product

$$I_{AG} = -\frac{n_A}{2\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{J_0(\bar{\alpha})J_0(\bar{\lambda})\,e^{ik_0w_A(\alpha+n_M)/2}}{(\alpha+n_M)(\lambda^2+n_M^2)(\lambda^2-n_A^2)} \times\left(\frac{\epsilon_r n_M^2 t_0}{T_e}+\frac{\lambda^2 t_1}{T_m}\right)d\alpha\,d\lambda \tag{5.24}$$

The last inner product, I'_{AG} is very similar to I_{AG}, except that we now use the incident slot field of the i_4-state. Remembering that $\hat{\tilde{E}}_A^{i_4} = 0$ for $y > w_M/2$, the transform of the propagation factor is obtained as

$$\frac{k_0}{2\pi}\int_{-\infty}^{w_M/2} e^{ik_0(\lambda+n_A)y}\,dy = \frac{-ie^{ik_0w_M(\lambda+n_A)/2}}{2\pi(\lambda+n_A)} \tag{5.25}$$

Combining this with (B.31), the transform of the slot electric field is obtained

$$\hat{\tilde{E}}_A^{i_4} = \frac{-ik_0J_0(\bar{\alpha})}{4\pi^2(\lambda+n_A)}e^{ik_0w_M(\lambda+n_A)/2} \tag{5.26}$$

Using (5.3) and (5.23), we obtain our final inner product

$$I'_{AG} = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{J_0(\bar{\alpha})e^{ik_0w_A(\alpha+n_M)/2}\,J_0(\bar{\lambda})e^{ik_0w_M(\lambda+n_A)/2}}{(\alpha+n_M)(\lambda^2+n_M^2)(\lambda+n_A)} \times\left(\frac{\epsilon_r n_M^2 t_0}{T_e}+\frac{\lambda^2 t_1}{T_m}\right)d\alpha\,d\lambda \tag{5.27}$$

The integrals which we have obtained in the manner described above are sometimes referred to as Sommerfeld integrals. They make up a class of integrals which have been studied for some time and for which computer aided solutions have been obtained. However, they are by no means easy to compute.

5.2 Problems in Evaluating the Sommerfeld Integrals

The Sommerfeld integrals derived in the previous section present a number of challenges before they can be coded for numerical evaluation. It

turns out that this part of the analysis actually took most of the time, i.e., the analytical manipulation of the integrals and the resulting programming effort. This also seems to be one of the major drawbacks of the integral equation technique described herein. Every time a more complicated trial function is used for the microstrip currents or slot fields, the bulk of the results have to be derived again. The same is true, of course, when the geometry is to be altered.

Not all six of the integrals required the same amount of analytical effort. In fact, I'_{AG} can be found in closed form. The integrals I_{MI} and I_{AG} reduce to single integrals which converge rapidly. The integral I_{AM} is also rapidly convergent after some manipulation. Its major difficulty lies in the two simple poles which overlap when $n_A/\sin\theta = n_M/\cos\theta$.

The remaining two integrals I_{MM} and I_{AA} required major analytical maneuvers before they could be computed accurately (3 or 4 significant digits) and in a reasonable amount of time (on the order of a few minutes on a computer workstation such as an HP 370). These integrals represent a "self-action" of the fundamental modes with themselves. This results in some interesting behavior of the integrands. They contain a double pole which needs to be extracted. They are oscillatory because of the Bessel function. They contain a simple pole due to the function D_e in the denominator of one term. They are slowly converging in γ, decaying only algebraically. One of them, I_{AA}, has a square root singularity in γ. This, in combination with the fact that often the terms that were derived in the process of making the Sommerfeld integrals manageable, themselves gave rise to interesting numerical problems, is why this part of the thesis took the most time.

Since the numerical work is of a purely mathematical nature, it has been deferred to Appendix C. It should also be noted that in the process of coding the results and debugging the programs, a number of algorithms were included which were based on empirical observation of the behavior of the integrals. For example, one of the terms which was added to the integrand of I_{AA} and subtracted in closed form from the result was used to speed up convergence of the γ-integration. However, it also turns out to have a square-root singularity when θ approaches 0. This singularity only occurs upon performing the γ-integration, which had to be done numerically. Thus, the singularity could not be treated analytically, but had to accounted for by empirically observing the functional behavior of the integrand after the γ-integration had been performed.

The same kind of empirical observation was required for determining the optimal integration parameters, i.e., the number of sections and points used for the integrations. The trade-off in this business is always that of speed vs. accuracy, but it is usually impossible to predict ahead of time what the parameters should be. In some sense, this numerical work can almost be considered a branch of experimentation and not theory, a rather interesting twist.

CHAPTER 6

RESULTS AND CONCLUSIONS

With the integrals coded, we are now ready to show some representative results for the S-matrix of the microstrip-slotline crossover. The results are compared to those published by Uwano, et al [16] and to a simple transformer model used by Chambers, et al [12] and by Knorr [13]. A few curves are shown which can be used in conjunction with results for an open microstrip and a shorted slotline to derive a solution for a microstrip-slotline transition. How this is done is described. This chapter also discusses potential future work in order to increase the accuracy of the results.

6.1 Some Comparisons

In Chapter 2, expressions were derived which hold for low-loss crossovers and which can be used to verify the numerical convergence of the scattering parameters. We define three convergence parameters based on the last three equations in (2.10)

$$\left.\begin{aligned} U_1 &= 2|\mathcal{C}|^2 + |\Gamma_M|^2 + |T_M|^2 \\ U_2 &= 2|\mathcal{C}|^2 + |\Gamma_A|^2 + |T_A|^2 \\ U_3 &= \mathcal{C}^*\left(\Gamma_M - T_M\right) + \mathcal{C}\left(\Gamma_A^* - T_A^*\right) \end{aligned}\right\} \tag{6.1}$$

The convergence parameters U_1 and U_2 should equal 1 and U_3 should equal 0. However, even if these conditions are satisfied, it does not mean that the scattering parameters are necessarily correct to within the same precision.

Table 6.1. Comparison of this analysis with transformer model and published data ($d = w_M = w_A = 0.6\,\text{mm}$, $\epsilon_r = 9.6$, $\mu_r = 1.0$, $\epsilon_M = 6.55$, $\epsilon_A = 2.48$, and $f = 2\,\text{GHz}$)

S-parameter	Bivariational	Uwano, et al	Transformer
$\Re(\Gamma_M)$	0.296	0.30	0.296
$\Im(\Gamma_M)$	0.019	-0.019	0
$\Re(T_M)$	0.704	0.71	0.704
$\Im(T_M)$	-0.019	0.046	0
$\Re(\Gamma_A)$	-0.291	-0.30	-0.296
$\Im(\Gamma_A)$	0.018	0.024	0
$\Re(T_A)$	0.709	0.71	0.704
$\Im(T_A)$	0.018	-0.022	0
$\Re(C)$	-0.452	-0.46	0.457
$\Im(C)$	-0.001	0.011	0
$U_1 = 0.993$, $U_2 = 0.996$, $U_3 = (-0.00445, -0.00182)$			

For example, the microstrip reflection coefficient should have a negative phase angle because the slot in the ground plane represents a capacitive reactance. In this analysis, we found that the phase angle actually crosses over the axis and is slightly positive for low frequencies. This can be seen in Table 6.1. The phase angle of 3.7 degrees corresponds to at least a 2% error with respect to 180 degrees, but probably not more than 4% (if Uwano's results are any indication of the actual phase angle). The convergence parameters for this case were $U_1 = .993$, $U_2 = .996$, $\Re(U_3) = -0.00445$, and $\Im(U_3) = -0.00182$ which are all within 1% of the ideal. Thus, the accuracy of the S-parameters cannot be inferred from the accuracy of the convergence parameters.

Tables 6.1, 6.2, and 6.3 show comparative results from this analysis with those published by Uwano, et al [16] and with the simple transformer model used by Knorr [13]. The real parts match very well, but the imaginary

Table 6.2: Same as Table 6.1 except $\epsilon_M = 6.60$, $\epsilon_A = 2.77$, and $f = 4\,\mathrm{GHz}$

S-parameter	Bivariational	Uwano, et al	Transformer
$\Re(\Gamma_M)$	0.323	0.32	0.324
$\Im(\Gamma_M)$	0.004	-0.060	0
$\Re(T_M)$	0.677	0.68	0.676
$\Im(T_M)$	-0.004	0.073	0
$\Re(\Gamma_A)$	-0.315	-0.32	-0.324
$\Im(\Gamma_A)$	0.025	0.044	0
$\Re(T_A)$	0.685	0.68	0.676
$\Im(T_A)$	0.025	-0.051	0
$\Re(\mathcal{C})$	-0.460	-0.47	0.468
$\Im(\mathcal{C})$	-0.027	0.025	0
$U_1 = 0.987$, $U_2 = 0.994$, $U_3 = (-0.00952, -0.000024)$			

parts are substantially different. We believe that the imaginary part of T_A from Uwano [16] was plotted with the wrong sign since it violates the relationship $T_A = 1+\Gamma_A$. The signs for the coupling factors are also reversed, but this is due to Uwano's polarity definitions. Their coupling factor is just $-\mathcal{C}$. Even with that, the amount by which the imaginary parts increase are still quite different. This analysis predicts a greater change in the imaginary part of the coupling coefficient; Uwano's predicts a greater change in the imaginary parts of the other S-parameters.

The expressions for the scattering parameters of the transformer model can readily be derived. We found that the transformer model results match ours and Uwano's the best when the transformer turns ratio is set equal to 1. The S-parameters in terms of the line impedances are

Table 6.3: Same as Table 6.1 except $\epsilon_M = 6.67$, $\epsilon_A = 2.97$, and $f = 6\,\mathrm{GHz}$

S-parameter	Bivariational	Uwano, et al	Transformer
$\Re(\Gamma_M)$	0.342	0.33	0.343
$\Im(\Gamma_M)$	-0.010	-0.097	0
$\Re(T_M)$	0.658	0.65	0.657
$\Im(T_M)$	0.010	0.098	0
$\Re(\Gamma_A)$	-0.330	-0.33	-0.343
$\Im(\Gamma_A)$	0.028	0.067	0
$\Re(T_A)$	0.670	0.65	0.657
$\Im(T_A)$	0.028	-0.075	0
$\Re(C)$	-0.463	-0.49	0.475
$\Im(C)$	-0.052	0.032	0
$U_1 = 0.984$, $U_2 = 0.993$, $U_3 = (-0.00126, -0.00169)$			

$$\left.\begin{aligned} \Gamma_M &= \frac{Z_A}{4Z_M + Z_A} \\ T_M &= \frac{4Z_M}{4Z_M + Z_A} \\ \Gamma_A &= \frac{-Z_A}{4Z_M + Z_A} \\ T_A &= \frac{4Z_M}{4Z_M + Z_A} \\ C &= \frac{2\sqrt{Z_A Z_M}}{4Z_M + Z_A} \end{aligned}\right\} \tag{6.2}$$

These were used with the impedances derived in Appendixes A and B to generate the data in the tables.

The following figures show representative S-parameters for two cases of microstrip lines. The first set, Figures 6.1–6.5, was derived for a $50\,\Omega$ microstrip line and the second set, Figures 6.6–6.10, for a $75\,\Omega$ microstrip line. The $50\,\Omega$ microstrip has a width of $w_M = 0.25\,\mathrm{mm}$ and the $75\,\Omega$ microstrip has $w_M = 0.093\,\mathrm{mm}$. The impedances are the quasi-static impedances and were obtained from Gupta [5]. The substrate height of

$d = 0.254\,\text{mm}$ corresponds to a 10 mil substrate. The frequencies were swept from $f = 18\,\text{GHz}$ to $f = 28\,\text{GHz}$. The four heavy traces on each plot are for $w_A = 0.1d$, $0.3d$, $0.5d$, and $0.7d$, respectively. This is noted on the plots. There are also two light traces on each of the magnitude plots. These are the results from the transformer model for $w_A = 0.1d$ and $w_A = 0.7d$.

The transformer model matches the magnitudes of the S-parameters reasonably well. However, it does not predict the imaginary parts of the S-parameters. It also predicts that the reflection and transmission coefficients for the microstrip and slotline have the same magnitude. This is not confirmed by our bivariational analysis. In fact, the discrepancies between the magnitudes of the S-parameters is greatest for Γ_A reaching a maximum of 18% for $w_A = 0.1d$ in Figure 6.8.

There is some interesting behavior when the slotline is very narrow. This can be seen in both the magnitude and phase plots. The magnitude of the S-parameters starts drifting in the opposite direction for the narrow slots. This has an impact on the phase response and causes the curves to cross as the frequency increases. If one looks at the actual percent differences in the numbers involved, one finds that the largest phase difference, which is in Figure 6.3, is on the order of 1%. For the data in Figure 6.8, the convergence parameters were found to be $U_1 = 1.01$, $U_2 = 1.005$, $\Re(U_3) = 0.0016$, and $\Im(U_3) = 0.0078$. The integration parameters were varied in a number of ways, and the results were always within 1%. Thus, the behavior seen in the figures is very likely a real effect due to the complicated interactions of the fields in the junction region.

The code of Appendix D can be used to generate a database of

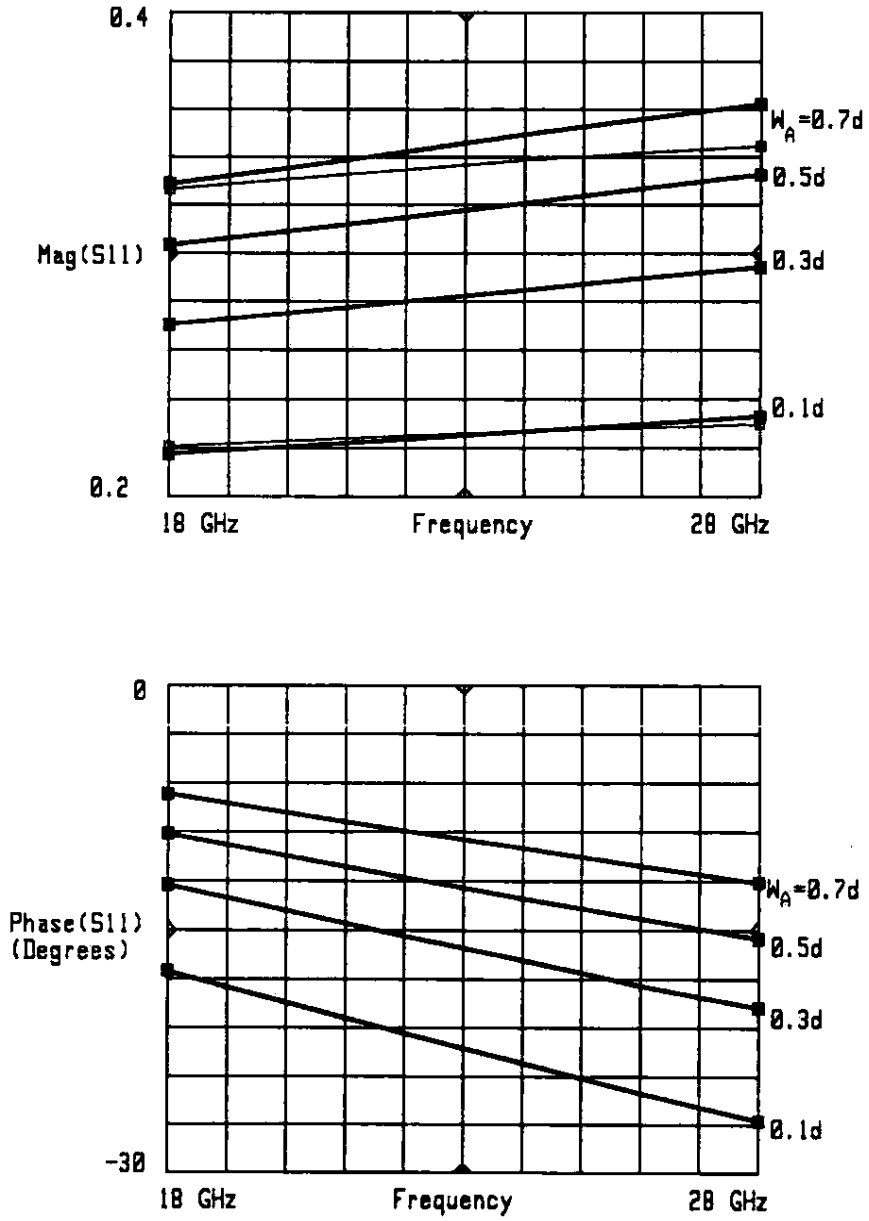


Figure 6.1: Microstrip reflection coefficient, Γ_M, for a $50\,\Omega$ microstrip

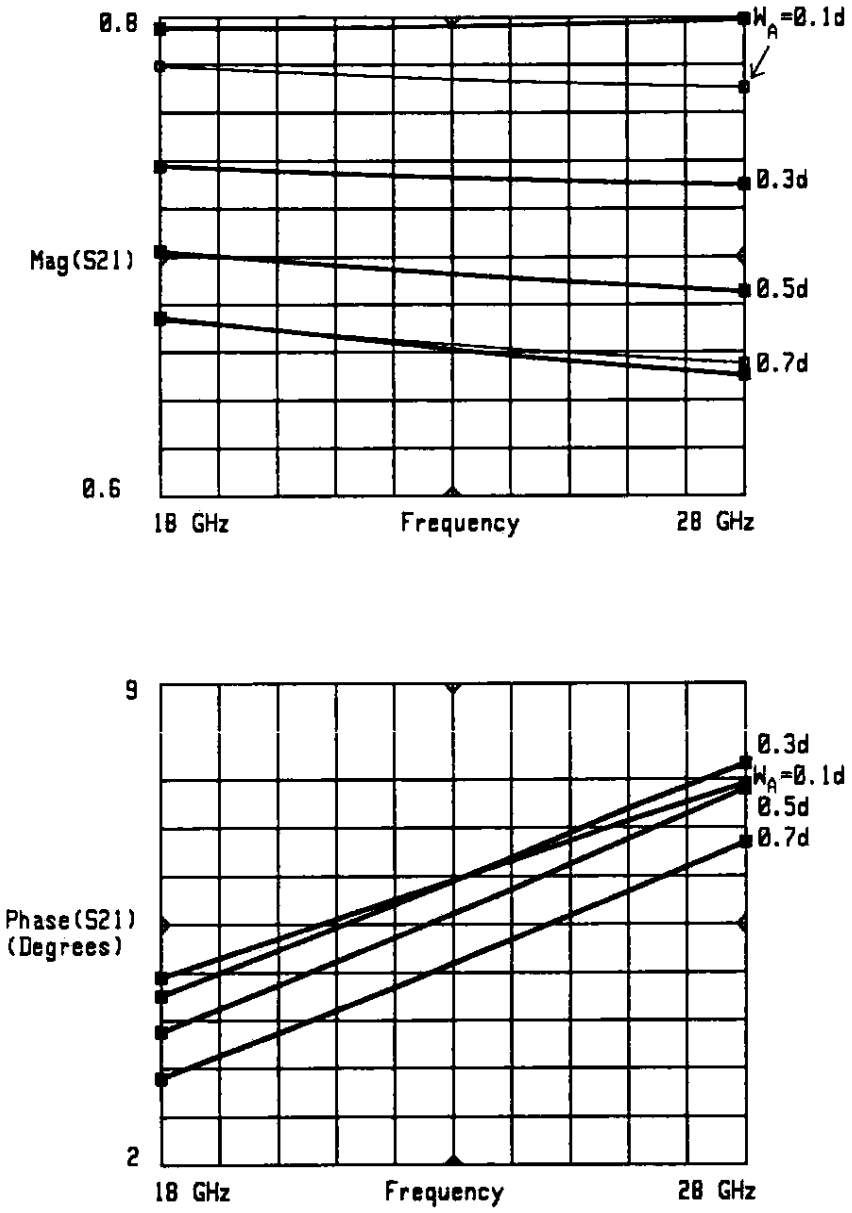


Figure 6.2: Microstrip transmission coefficient, T_M, for a $50\,\Omega$ microstrip

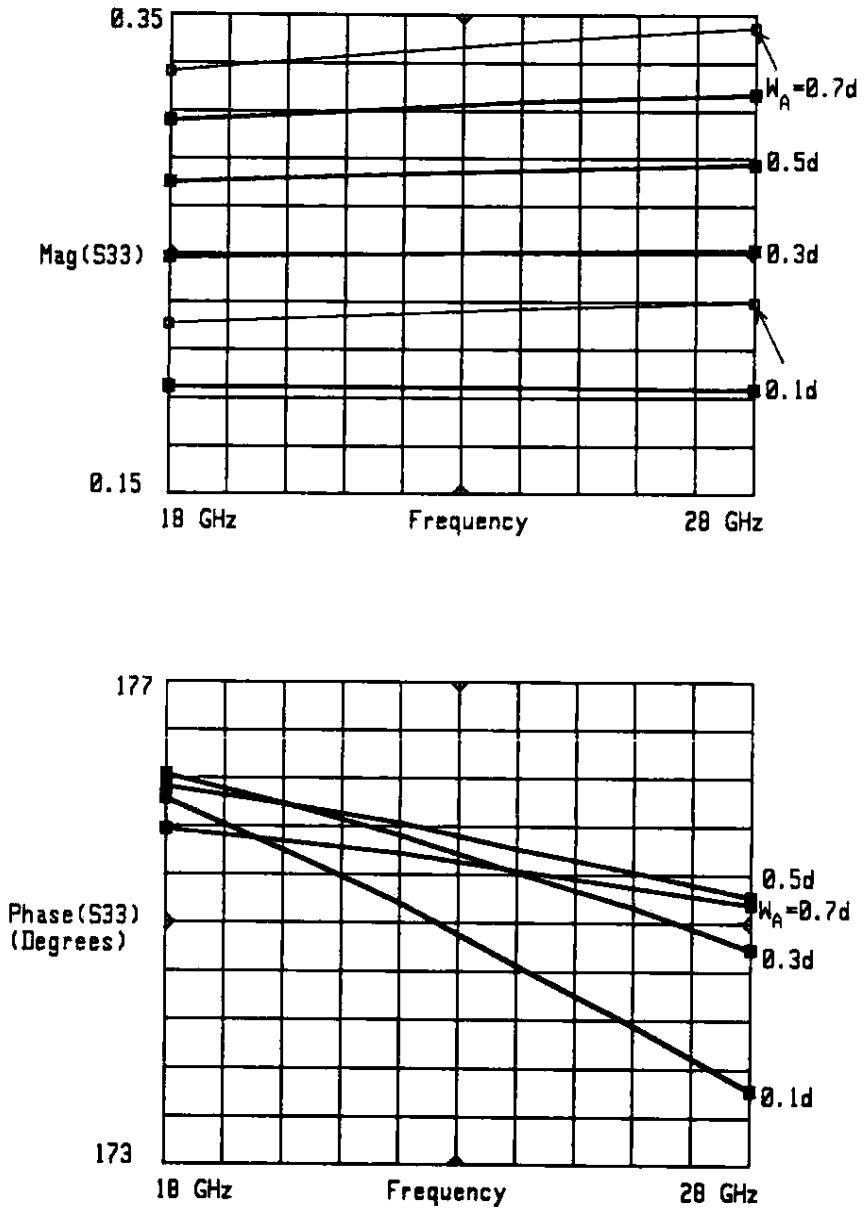


Figure 6.3: Slotline reflection coefficient, Γ_A, for a 50 Ω microstrip

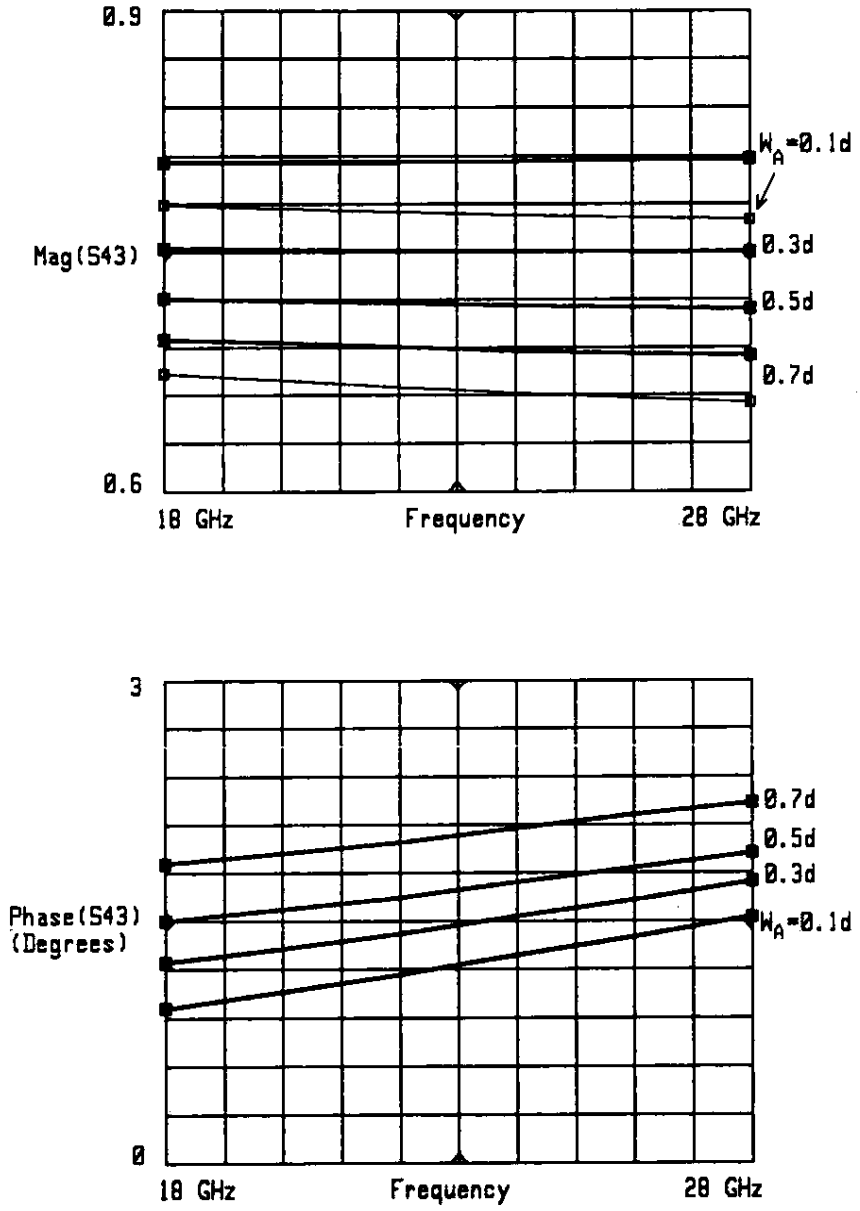


Figure 6.4: Slotline transmission coefficient, T_A, for a $50\,\Omega$ microstrip

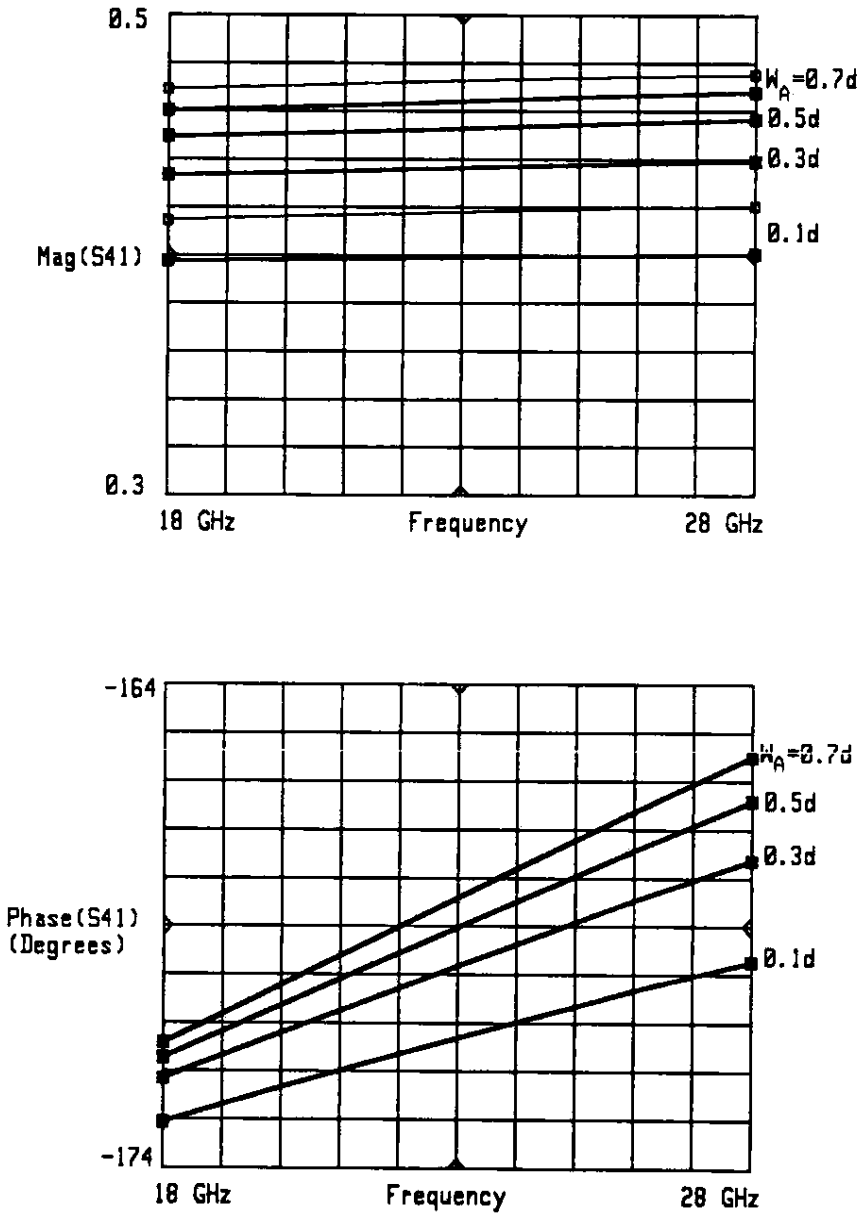


Figure 6.5: Coupling coefficient, $\mathcal{C}$, for a $50\,\Omega$ microstrip

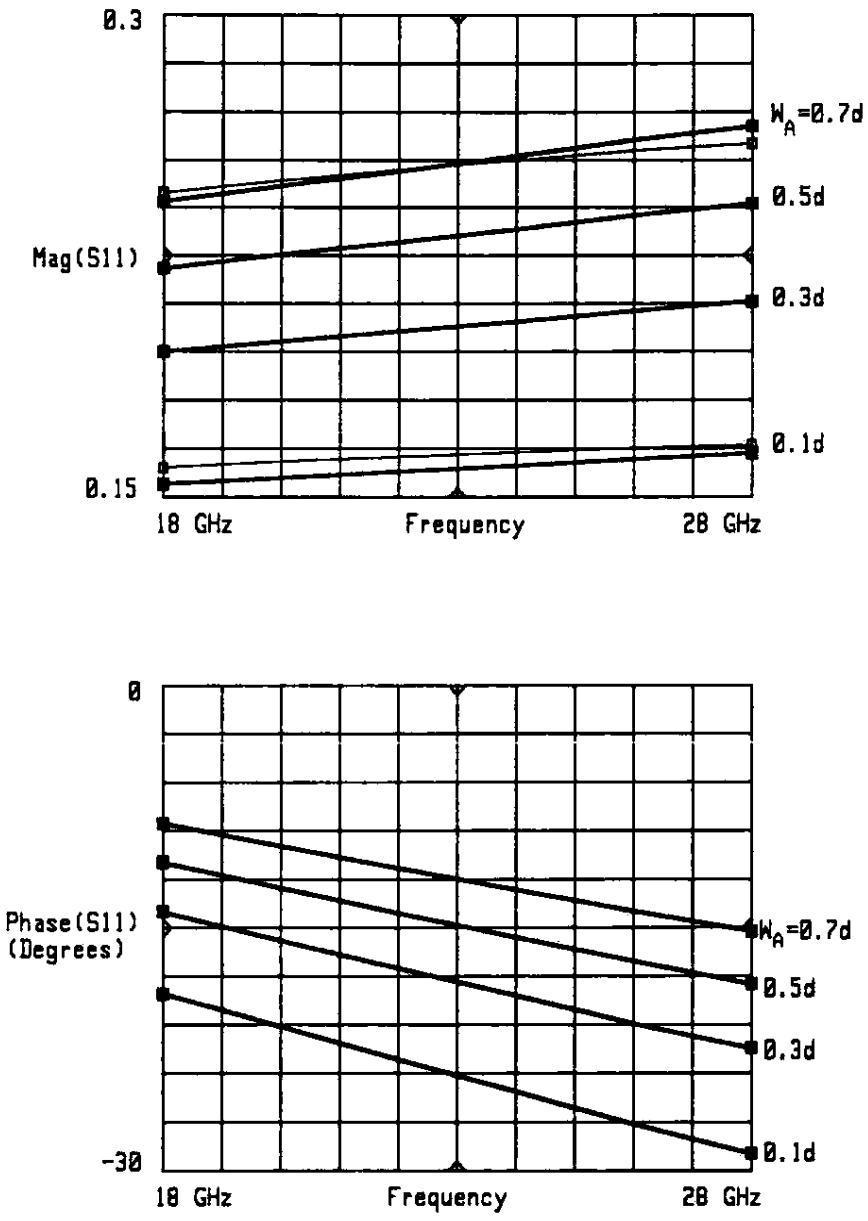


Figure 6.6: Microstrip reflection coefficient, Γ_M, for a 75 Ω microstrip

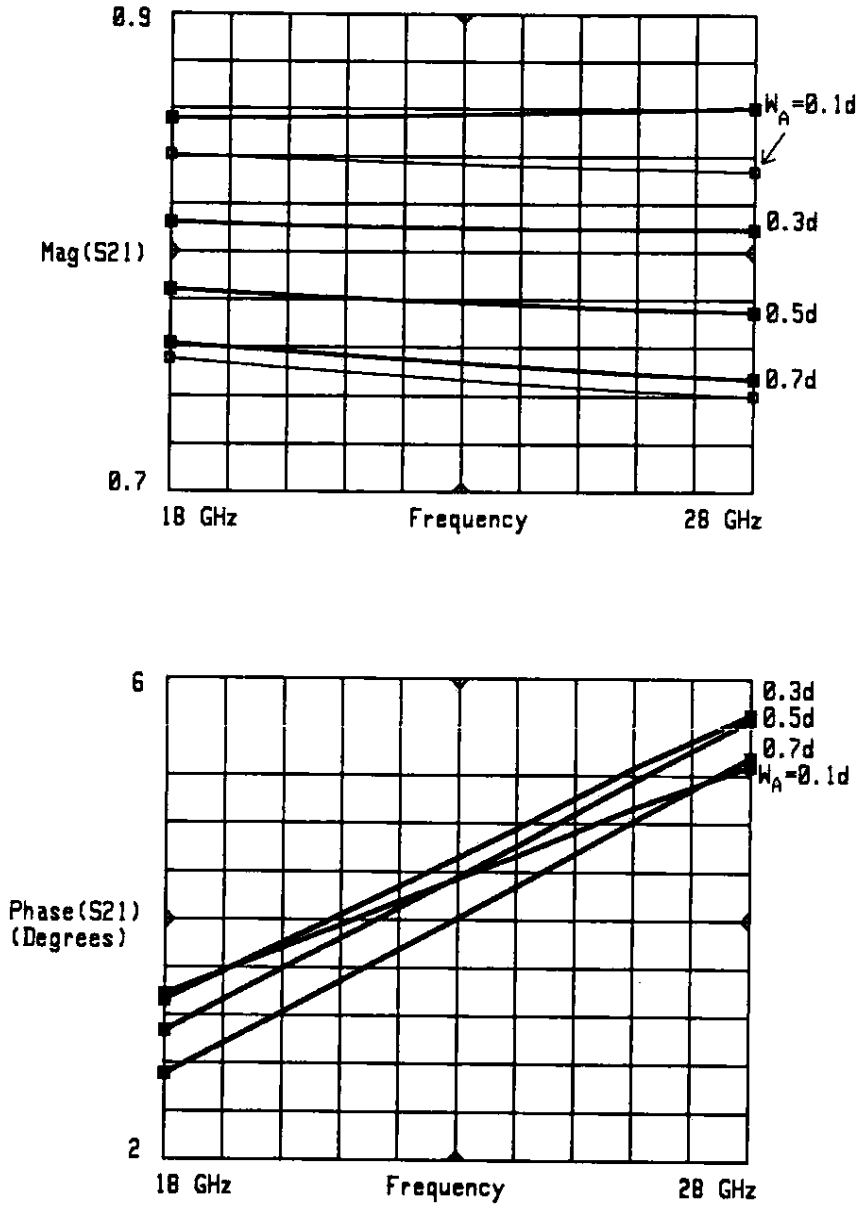


Figure 6.7: Microstrip transmission coefficient, T_M, for a 75 Ω microstrip

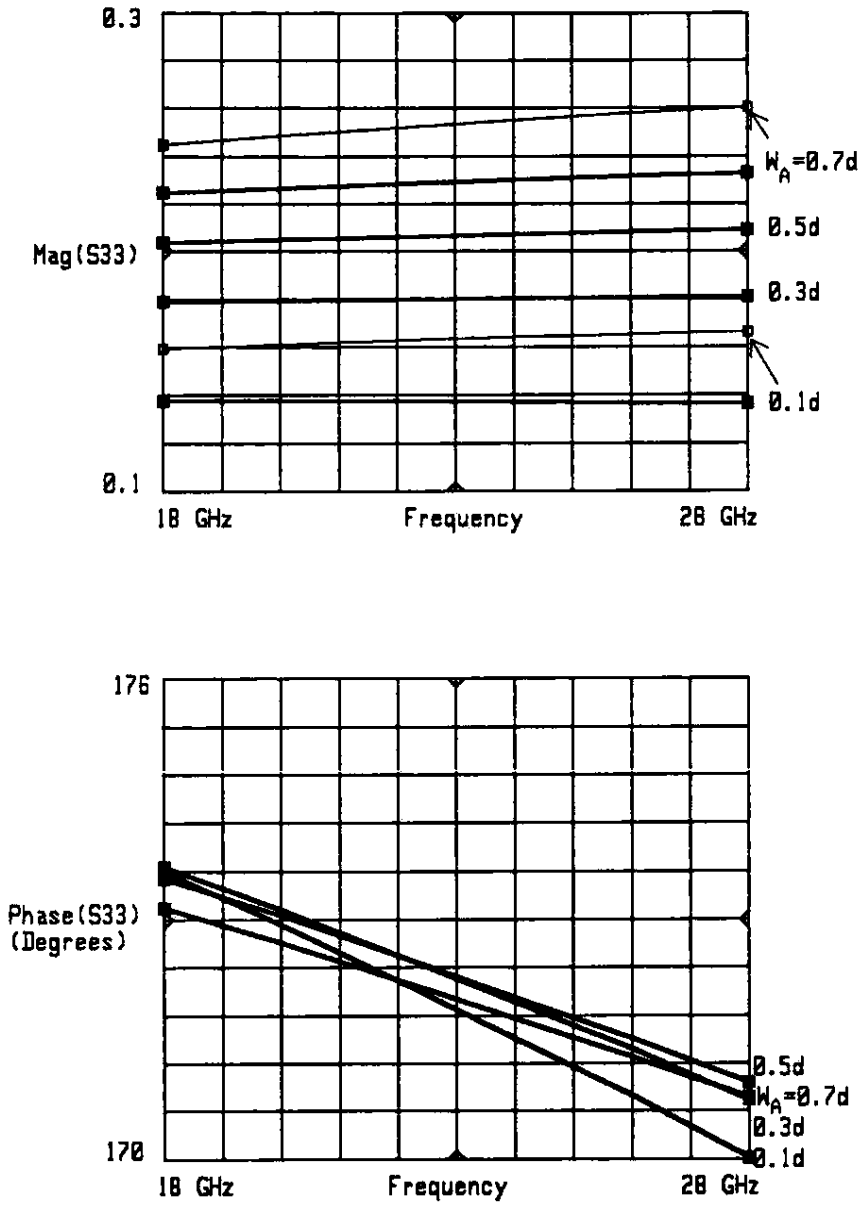


Figure 6.8: Slotline reflection coefficient, Γ_A, for a $75\,\Omega$ microstrip

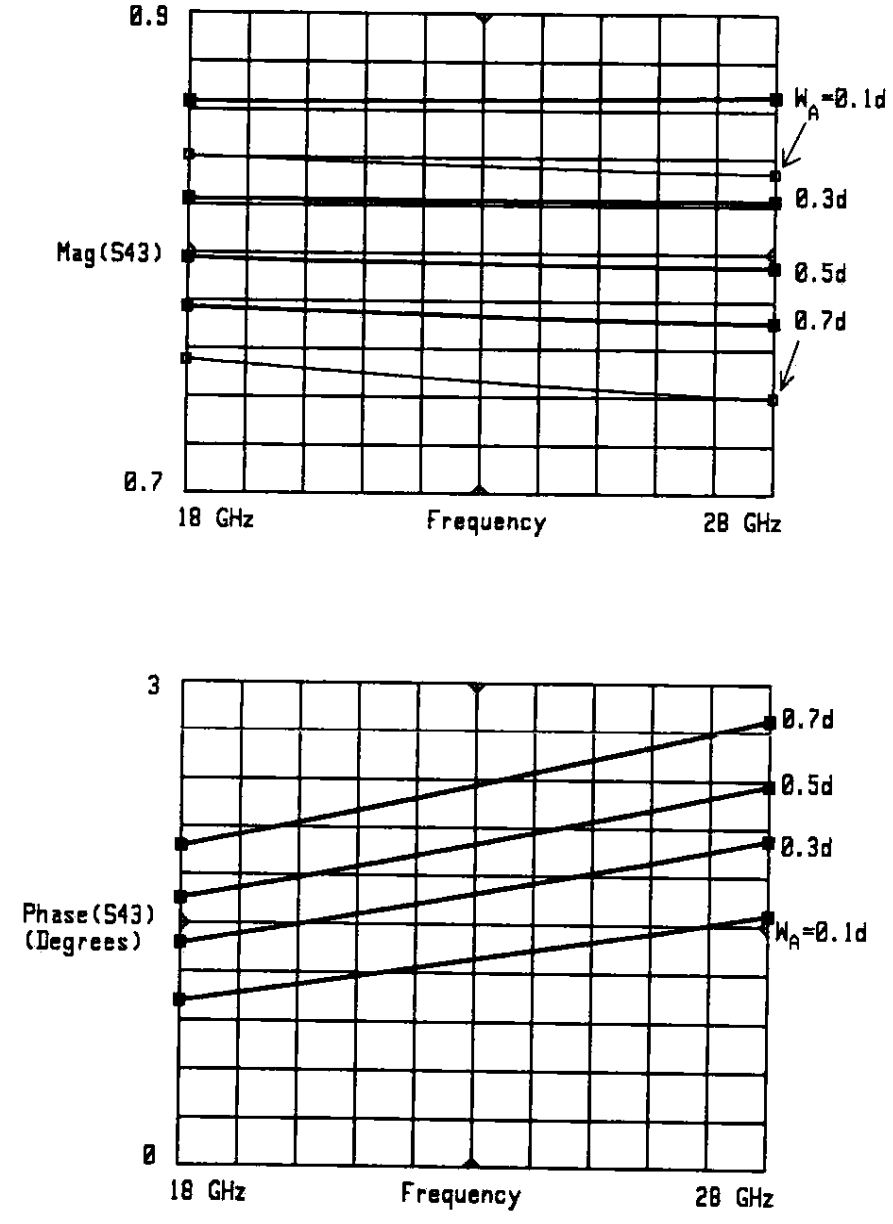


Figure 6.9: Slotline transmission coefficient, T_A, for a 75 Ω microstrip

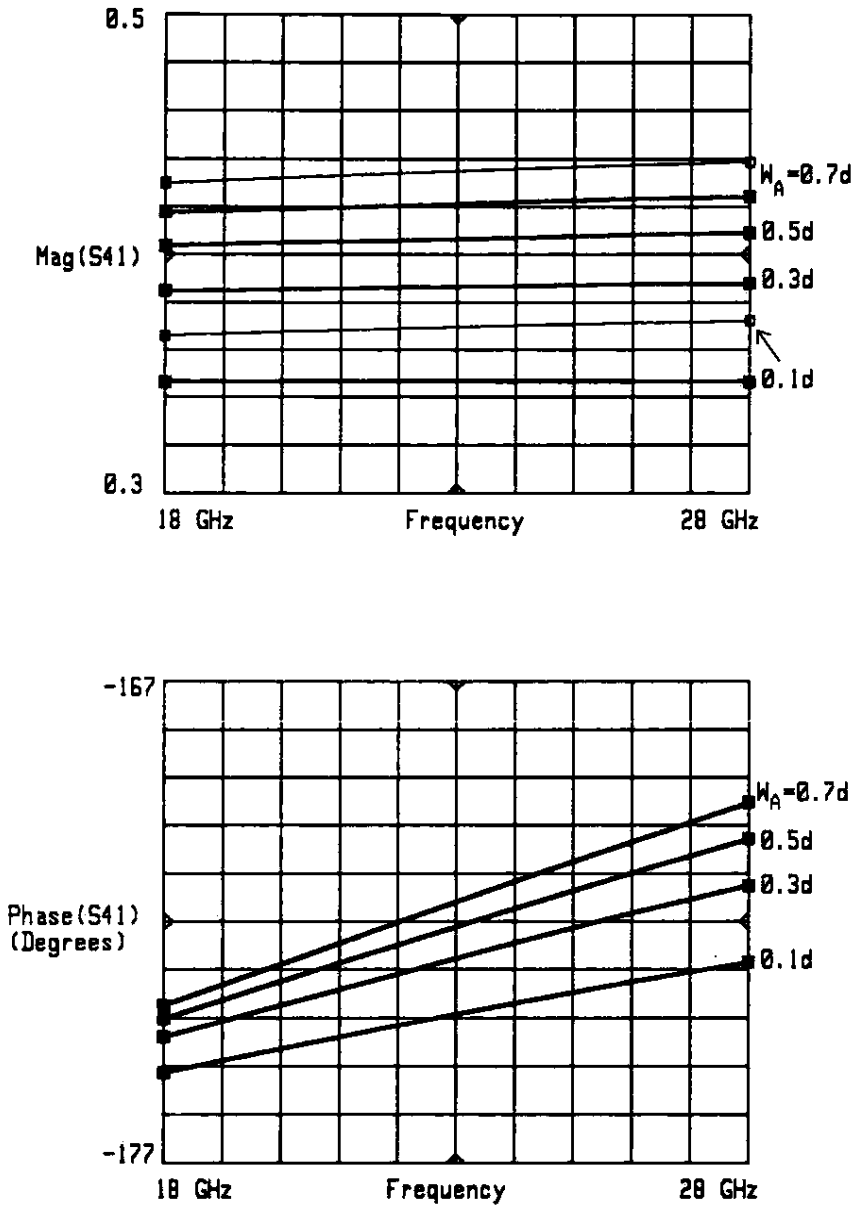


Figure 6.10: Coupling coefficient, C, for a $75\,\Omega$ microstrip

S-parameters for the microstrip-slotline crossover. This database can then be incorporated with models of the open microstrip line and shorted slotline to obtain a reasonably accurate model for the microstrip-slotline transition. It can be used to determine the necessary stub lengths of the microstrip and slotline to minimize the return loss of the transition.

The S-matrix for the crossover can readily be reduced to a two-port S-matrix once the terminations for two of the ports are known. From the definition of scattering parameters the following result can be derived when terminating one port of a four-port network

$$S_{ij}^3 = \left[S_{ij}^4 + \frac{S_{i4}^4 S_{4j}^4}{1/\Gamma_4 - S_{44}^4} \right] \tag{6.3}$$

Here the superscript refers to the number of ports of the resulting network and Γ_4 is the reflection coefficient of terminating impedance on port 4. If port 3 is now terminated in a load with reflection coefficient Γ_3, then the S-matrix of the two-port network is given in terms of the three-port scattering parameters as

$$S_{ij}^2 = \left[S_{ij}^3 + \frac{S_{i3}^3 S_{3j}^3}{1/\Gamma_3 - S_{33}^3} \right] \tag{6.4}$$

When using these equations, one should remember that the terminating impedances must be calculated with the reference planes at the origin of the crossover.

6.2 Future Improvements

The accuracy of the above model is rather astonishing when one considers that no account was taken of the actual distributions of the scattered currents and fields around the junction. We assumed the microstrip

current to be the same as that of the fundamental microstrip mode and the slotline electric field to be the same as that of the fundamental slotline mode. Intuitively, one would expect the real current and field distributions around the junction to be far more intricate, containing all kinds of terms due to the higher-order modes which are invariably generated by the discontinuity. This leads to a natural extension of the present solution technique.

In order to improve the accuracy of the model derived above, one would first have to come up with valid first-order approximations for the actual current and field distributions around the junction. The Fourier transforms of these could then be used in the Sommerfeld integrals instead of our Bessel functions. This would of course entail that the numerical work be repeated.

One useful approach might be to expand the real current and field distributions in terms of well characterized basis functions. One could then solve the integrals in a general form and use the amplitude coefficients of the basis functions as unknowns. This might allow one to generalize this technique to more complex geometries without having to repeat the analytical work every time a subtle change is introduced.

Another way to improve the accuracy of our model would be to accurately determine the propagation characteristics of slotlines and even microstrips (there are still significant discrepancies between various published results). This means developing efficient code to calculate the effective dielectric constants for the fundamental mode of slotlines and microstrips and incorporating it into the code at the end of Appendix D.

In conclusion, we believe that this bivariational analysis offers an

accurate model (5% or better) for the microstrip-slotline crossover for practical line dimensions. This model is especially useful at higher frequencies where the reactive effects of the crossover are expected to become more pronounced. By combining our model with models for broadband microstrip and slotline stubs, engineers should now be able to design microstrip-slotline transitions without having to go through the cumbersome procedure of measuring sample circuits. This has been necessary in the past in order to determine the reference plane positions for the terminating quarter-wave stubs. In other words, engineers did not have an accurate understanding of the reactive effects of the microstrip-slotline crossover, of the open-circuited microstrip line and of the short-circuited slotline. This model helps to provide such an understanding.

BIBLIOGRAPHY

[1] K. C. Gupta, Ramesh Garg, and I. J. Bahl, **Microstrip Lines and Slotlines**. Norwood, MA: Artech House, 1979.

[2] T. C. Edwards, **Foundations for Microstrip Circuit Design**. Chichester, Great Britain: John Wiley & Sons, 1981.

[3] Reinmut K. Hoffmann, **Handbook of Microwave Integrated Circuits**. Norwood, MA: Artech House, 1987, pp. 164, 381.

[4] Inder Bahl and Prakash Bhartia, **Microwave Solid State Circuit Design**. New York: John Wiley & Sons, 1988.

[5] K. C. Gupta, Ramesh Garg, and Rakesh Chadha, **Computer-Aided Design of Microwave Circuits**. Dedham, MA: Artech House, 1981, pp. 60–67.

[6] Rolf H. Jansen, "High-Speed Computation of Single and Coupled Microstrip Parameters Including Dispersion, High-Order Modes, Loss and Finite Strip Thickness", **IEEE Trans. MTT–26**, Feb. 1978, pp. 75–82.

[7] Seymour B. Cohn, "Slot-line Field Components", **IEEE Trans. MTT–20**, Feb. 1972, pp. 172–174.

[8] E. A. Mariani, C. P. Heinzman, J. P. Agrios, and S. B. Cohn, "Slot Line Characteristics", **IEEE Trans. MTT–17**, Dec. 1969, pp. 1091–1096.

[9] B. N. Das and K. K. Joshi, "Impedance of a Radiating Slot in the Ground Plane of a Microstripline", **IEEE Trans. AP–30**, Sept. 1982, pp. 922–926.

[10] D. M. Pozar, N. K. Das, B. N. Das, and K. K. Joshi, "Comments on 'Impedance of a Radiating Slot in the Ground Plane of a Microstripline'," **IEEE Trans. AP–34**, July 1986, pp. 958–959.

[11] David M. Pozar, "A Reciprocity Method of Analysis for Printed Slot and Slot-Coupled Microstrip Antennas", **IEEE Trans. AP–34**, Dec.

1986, pp. 1439–1446.

[12] D. Chambers, S. B. Cohn, E. G. Cristal, and L. Young, "Microwave Active Network Synthesis", Stanford Res. Inst., Menlo Park, CA, Contract DAAB07-70-C-004, SRI Project 8245, Semiannual Report, June 1970.

[13] Jeffrey B. Knorr, "Slot-Line Transitions", **IEEE Trans. MTT–22**, May 1974, pp. 548–554.

[14] A. Podcameni and M. L. Coimbra, "Slotline-Microstrip Transition on Iso/Anisotropic Substrate: A More Accurate Design", **Electronics Letters**, Vol. 16, Sept. 1980, pp. 388–389.

[15] H. Y. Yang and N. G. Alexópoulos, "A Dynamic Model for Microstrip-Slotline Transition and Related Structures", **IEEE Trans. MTT–36**, Feb. 1988, pp. 286–293.

[16] Tomoki Uwano, Tatsuo Itoh, and Roberto Sorrentino, "Characterization of Microstrip-to-Slotline Transition Discontinuities by Transverse Resonance Analysis", **Alta Frequenza**, Vol. 57, June 1988, pp. 183–191.

[17] E. F. Kuester and D. C. Chang, **Theory of Waveguides and Transmission Lines**. ECE511 Course Notes, University of Colorado, Unpublished, 1985, Chapters 5–6.

[18] R. F. Harrington, **Time-Harmonic Electromagnetic Fields**. New York: McGraw-Hill, 1961, pp. 116-120.

[19] E. F. Kuester and D. C. Chang, "A Unified Approach to the Derivation of Bivariational Principles in Acoustics and Electromagnetics", Scientific Report No. 80, Electromagnetics Lab., Dept. of Electrical and Computer Engineering, Univ. of Colorado, Boulder, July 1985, pp. 4–18, 82.

[20] E. F. Kuester, **Elecromagnetic Boundary Problems**. ECE514 Course Notes, Dept. of Electrical and Computer Engineering, University of Colorado, Unpublished, Fall, 1986, pp. 1-19–1-21.

[21] Carlton H. Walter, **Traveling Wave Antennas**. New York: McGraw-Hill Book Company, 1965, pp. 299–302.

[22] Julius A. Stratton, **Electromagnetic Theory**. New York: McGraw-Hill Book Company, 1941, pp. 28–32.

[23] I. S. Gradshteyn and I. M. Ryzhik, **Table of Integrals, Series, and Products**. Orlando, FL: Academic Press, 1980.

[24] Milton Abramowitz and Irene A. Stegun, **Handbook of Mathematical Functions**. New York: Dover Publications, 1972.

[25] George E. Forsythe, **Computer Methods for Mathematical Computations**. Englewood Cliffs, NJ: Prentice Hall, Inc., 1977, pp. 101–105.

[26] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, **Numerical Recipes**. Cambridge, England: Cambridge University Press, 1986, pp. 277–282.

APPENDIX A

CALCULATION OF THE MICROSTRIP NORM

In Chapter 4 we found that in order to calculate the scattering coefficients for the microstrip-slotline crossover, we need to know the norm of the fundamental microstrip mode. This appendix describes how to find the electromagnetic fields of this mode by using a good approximation for the microstrip current. From these fields, the norm can be found by performing an integration of the Poynting vector over a transverse surface intersecting the microstrip. The norm can also be used to calculate the characteristic impedance of the microstrip transmission line.

For this analysis, the microstrip mode fields were normalized by setting the total current on the microstrip equal to 1 A. The characteristic impedance is calculated based on a power-current definition. The characteristic impedance also provides a useful check on the spectral domain derivations since published data for it already exists. Some numerical results are summarized at the end of the appendix.

A.1 Microstrip Mode Fields in the Spectral Domain

The derivation for the microstrip mode fields proceeds very similar to the more general derivation found in Chapter 4. In this case, we have only a uniform microstrip geometry. We assume that the fundamental microstrip mode has been excited in some fashion. The equivalence principle allows us

to convert the problem into a grounded dielectric slab geometry which has the microstrip current impressed as an equivalent source. This allows us to find the mode fields in terms of the microstrip current.

Since the fields and currents of the microstrip fundamental mode behave as $e^{-i\beta_M x} = e^{-in_M k_0 x}$, where $n_M = \sqrt{\epsilon_M}$, the Helmholtz equation only needs to be transformed in the y-dimension. Equation (3.1) becomes

$$\left[\frac{d^2}{dz^2} - \left(n_M^2 + \lambda^2\right) k_0^2 + k^2\right] \hat{\Pi}_j = 0 \tag{A.1}$$

But this is just like (3.10), except that α is replaced by n_M. This means that the results (3.11) for the Hertz potentials are still valid. However, there are no fields below the ground plane, so only two regions of space need to be considered.

$$\hat{\Pi}_j(n_M, \lambda, z) = \begin{cases} \mathcal{A}_j e^{-k_0 t_0 (z-d)} & (z \geq d) \\ \mathcal{B}_j \sinh(k_0 t_1 z) + \mathcal{C}_j \cosh(k_0 t_1 z) & (0 \leq z \leq d) \\ 0 & (z \leq 0) \end{cases} \tag{A.2}$$

The FT variables $\vec{\gamma}$, $\vec{\gamma}_\perp$, γ^2, u_0, u_1, D_e, and D_m from Chapter 3 are now mapped to the analogous variables

$$\vec{\chi} = n_M \vec{a}_x + \lambda \vec{a}_y \tag{A.3}$$

$$\vec{\chi}_\perp = \lambda \vec{a}_x - n_M \vec{a}_y \tag{A.4}$$

$$\chi^2 = \lambda^2 + n_M^2 \tag{A.5}$$

$$t_0 = \sqrt{\chi^2 - 1} \tag{A.6}$$

$$t_1 = \sqrt{\chi^2 - \epsilon_r \mu_r} \tag{A.7}$$

$$T_e = \epsilon_r t_0 \cosh(k_0 d t_1) + t_1 \sinh(k_0 d t_1) \tag{A.8}$$

and

$$T_m = \mu_r t_0 \sinh(k_0 d t_1) + t_1 \cosh(k_0 d t_1) \tag{A.9}$$

The unknown amplitude coefficients in (A.2) are found by applying the boundary conditions. The tangential electric field has to be zero at $z = 0$ and continuous across $z = d$. The tangential magnetic field is discontinuous across $z = d$ due to the microstrip current $\vec{J}_M$. Using these boundary conditions and solving the resulting equations for the coefficients yields

$$\begin{aligned}
\mathcal{A}_e &= \eta t_1 \sinh(k_0 d t_1)\, \vec{\chi} \cdot \hat{\vec{J}}_M / (k_0^2 \chi^2 T_e) \\
\mathcal{A}_m &= i\mu_r \sinh(k_0 d t_1)\, \vec{\chi}_\perp \cdot \hat{\vec{J}}_M / (k_0^2 \chi^2 T_m) \\
\mathcal{B}_e &= 0 \\
\mathcal{B}_m &= i\, \vec{\chi}_\perp \cdot \hat{\vec{J}}_M / (k_0^2 \chi^2 T_m) \\
\mathcal{C}_e &= -\eta t_0 \vec{\chi} \cdot \hat{\vec{J}}_M / (k_0^2 \chi^2 T_e) \\
\mathcal{C}_m &= 0
\end{aligned} \tag{A.10}$$

We can now obtain the fields from (3.2) and (A.2). We will also make use of the definitions (A.3)–(A.9). Only the components which will actually be required have been derived here. The y and z-components of the electric field for $z \geq d$ are

$$\hat{E}^+_{My} = \frac{i\eta \sinh(k_0 d t_1)}{\chi^2} \left[\frac{\lambda t_0 t_1}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M + \frac{\mu_r n_M}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M \right] e^{-k_0 t_0 (z-d)} \tag{A.11}$$

$$\hat{E}^+_{Mz} = \frac{\eta t_1 \sinh(k_0 d t_1)}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M e^{-k_0 t_0 (z-d)} \tag{A.12}$$

The magnetic field components for $z \geq d$ are

$$\hat{H}^+_{My} = \frac{-\sinh(k_0 d t_1)}{\chi^2} \left[\frac{n_M t_1}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M + \frac{\mu_r \lambda t_0}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M \right] e^{-k_0 t_0 (z-d)} \tag{A.13}$$

$$\hat{H}^+_{Mz} = \frac{i\mu_r \sinh(k_0 dt_1)}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M e^{-k_0 t_0 (z-d)} \tag{A.14}$$

The electric field components for $0 \leq z \leq d$ are

$$\hat{E}^+_{My} = \frac{i\eta}{\chi^2} \left[\frac{\lambda t_0 t_1}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M + \frac{\mu_r n_M}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M \right] \sinh(k_0 t_1 z) \tag{A.15}$$

$$\hat{E}^+_{Mz} = -\frac{\eta t_0}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M \cosh(k_0 t_1 z) \tag{A.16}$$

The magnetic field components for $0 \leq z \leq d$ are

$$\hat{H}^+_{My} = \frac{1}{\chi^2} \left[\frac{\epsilon_r n_M t_0}{T_e} \vec{\chi} \cdot \hat{\vec{J}}_M + \frac{\lambda t_1}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M \right] \cosh(k_0 t_1 z) \tag{A.17}$$

$$\hat{H}^+_{Mz} = \frac{i}{T_m} \vec{\chi}_\perp \cdot \hat{\vec{J}}_M \sinh(k_0 t_1 z) \tag{A.18}$$

We can now compute the required norm.

A.2 Microstrip Norm as a Spectral Domain Integral

To calculate the norm, we first rewrite (4.4) as

$$N_M = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\mathcal{E}^+_{My} \mathcal{H}^+_{Mz} - \mathcal{E}^+_{Mz} \mathcal{H}^+_{My} \right] dy\, dz \tag{A.19}$$

With our inverse Fourier transform relation (3.7), this becomes

$$N_M = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\hat{\mathcal{E}}^+_{My} \hat{\mathcal{H}}^+_{Mz} - \hat{\mathcal{E}}^+_{Mz} \hat{\mathcal{H}}^+_{My} \right] e^{-ik_0 y(\lambda+\lambda')} dy\, dz\, d\lambda\, d\lambda' \tag{A.20}$$

One of the definitions of the delta function is

$$\int_{-\infty}^{\infty} e^{-ik_0 y(\lambda+\lambda')} dy = \frac{2\pi}{k_0} \delta(\lambda + \lambda') \tag{A.21}$$

This allows us to reduce (A.20) to

$$N_M = \frac{2\pi}{k_0} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\hat{\mathcal{E}}^+_{My}(\lambda)\hat{\mathcal{H}}^+_{Mz}(-\lambda) - \hat{\mathcal{E}}^+_{Mz}(-\lambda)\hat{\mathcal{H}}^+_{My}(\lambda)\right] dz\, d\lambda \qquad \text{(A.22)}$$

The z-integrations are straightforward to perform. The λ-integration needs to be performed numerically. We can simplify the last integral further by realizing that the only terms in the field solutions which are not even in λ are $\vec{\chi}\cdot\hat{\vec{\mathcal{J}}}_M$ and $\vec{\chi}_\perp\cdot\hat{\vec{\mathcal{J}}}_M$. But we know that for the fundamental microstrip mode, the longitudinal current component is even about x and the transverse component is odd. This results in the simple symmetry properties

$$\left.\begin{aligned} \vec{\chi}(-\lambda)\cdot\hat{\vec{\mathcal{J}}}_M(-\lambda) &= \vec{\chi}(\lambda)\cdot\hat{\vec{\mathcal{J}}}_M(\lambda) \\ \vec{\chi}_\perp(-\lambda)\cdot\hat{\vec{\mathcal{J}}}_M(-\lambda) &= -\vec{\chi}_\perp(\lambda)\cdot\hat{\vec{\mathcal{J}}}_M(\lambda) \end{aligned}\right\} \qquad \text{(A.23)}$$

Using these symmetry properties and our field solutions in (A.22), allows us to write the following general solution for the microstrip norm

$$\begin{aligned} N_M = \frac{2\pi\eta}{k_0}\int_{-\infty}^{\infty}\frac{1}{\chi^2}\Bigg\{ & \left(\frac{\vec{\chi}_\perp\cdot\hat{\vec{\mathcal{J}}}_M}{T_m}\left(\frac{\lambda t_0 t_1}{T_e}\vec{\chi}\cdot\hat{\vec{\mathcal{J}}}_M + \frac{\mu_r n_M}{T_m}\vec{\chi}_\perp\cdot\hat{\vec{\mathcal{J}}}_M\right)\right. \\ & \left.\times\left[\frac{\sinh(2k_0dt_1)-2k_0dt_1}{4k_0t_1} + \frac{\mu_r\sinh^2(k_0dt_1)}{2k_0t_0}\right]\right) + \frac{\vec{\chi}\cdot\hat{\vec{\mathcal{J}}}_M}{T_e} \\ & \times\left(\frac{n_M\vec{\chi}\cdot\hat{\vec{\mathcal{J}}}_M}{T_e}\left\{\epsilon_r t_0^2\left[\frac{\sinh(2k_0dt_1)+2k_0dt_1}{4k_0t_1}\right] + \frac{t_1^2\sinh^2(k_0dt_1)}{2k_0t_0}\right\}\right. \\ & \left.\left. + \frac{\lambda t_0 t_1\vec{\chi}_\perp\cdot\hat{\vec{\mathcal{J}}}_M}{T_m}\left[\frac{\sinh(2k_0dt_1)+2k_0dt_1}{4k_0t_1} + \frac{\mu_r\sinh^2(k_0dt_1)}{2k_0t_0}\right]\right)\right\} d\lambda \end{aligned} \qquad \text{(A.24)}$$

A.2.1 Microstrip Current Trial Function We now have to make some approximation for the microstrip current, $\vec{J}_M$, since no closed

form solution is available for it. This approximation will be called our trial function. We assume only longitudinal currents exist on the strip. The following transverse distribution is known to provide an accurate estimate for the longitudinal microstrip current

$$\vec{\mathcal{J}}_M \approx \mathcal{J}_{Mx}(y)\,\vec{a}_x = \frac{A_M \vec{a}_x}{\sqrt{(w_M/2)^2 - y^2}} \tag{A.25}$$

where A_M is the amplitude coefficient of $\mathcal{J}_{Mx}$. The total microstrip current is found by integrating this function across the microstrip width. The following result is obtained

$$I_M = \int_{-w_M/2}^{w_M/2} \frac{A_M}{\sqrt{(w_M/2)^2 - y^2}}\, dy = \pi A_M \tag{A.26}$$

We also need to convert the current density (A.25) to its spectral domain equivalent. Applying our Fourier transform (3.4) and using the substitution $y = (w_M/2)\sin\xi$, yields the transformed current

$$\hat{\mathcal{J}}_{Mx}(\lambda) = \frac{k_0 A_M}{2\pi} \int_{-\pi/2}^{\pi/2} \cos\left(\frac{k_0 w_M}{2}\lambda \sin\xi\right) d\xi \tag{A.27}$$

But this is just the defining integral for the zeroth order Bessel function of the first kind, $J_0(k_0 w_M \lambda/2)$. In conjunction with (A.26) and our normalization condition that the current equals 1 A, the transformed current becomes

$$\hat{\mathcal{J}}_{Mx}(\lambda) = \frac{k_0}{2\pi} J_0(k_0 w_M \lambda/2) \tag{A.28}$$

A.2.2 Simplified Norm Integral Now that we have a trial function for the microstrip current, the norm integral (A.24) reduces to

$$
\begin{aligned}
N_M \;=\; & \frac{\eta\beta_M}{\pi}\int_0^\infty \left[\frac{J_0(k_0 w_M \lambda/2)}{\chi}\right]^2 \left\{\frac{\lambda^2}{T_m}\left(\frac{t_0 t_1}{T_e}+\frac{\mu_r}{T_m}\right)\right. \\
& \times \left[\frac{\sinh(2k_0dt_1)-2k_0dt_1}{4k_0t_1}+\frac{\mu_r\sinh^2(k_0dt_1)}{2k_0t_0}\right] \\
& +\frac{1}{T_e}\left(\frac{n_M^2}{T_e}\left\{\epsilon_r t_0^2\left[\frac{\sinh(2k_0dt_1)+2k_0dt_1}{4k_0t_1}\right]+\frac{t_1^2\sinh^2(k_0dt_1)}{2k_0t_0}\right\}\right. \\
& \left.\left.+\frac{\lambda^2 t_0 t_1}{T_m}\left[\frac{\sinh(2k_0dt_1)+2k_0dt_1}{4k_0t_1}+\frac{\mu_r\sinh^2(k_0dt_1)}{2k_0t_0}\right]\right)\right\} d\lambda
\end{aligned}
\tag{A.29}
$$

This result was obtained by combining (A.24), (A.28), and (A.3)–(A.5).

The above integral was evaluated using the FORTRAN program `ZM` which is listed in Appendix D. With the normalization of the microstrip current to 1 A, the norm is equal to the characteristic impedance, or $N_A = Z_A$.

The results of the above derivation were verified by comparing the characteristic impedance values with published data from Jansen [6]. The parameters used in the computation of Z_M were $d = 0.64\,\text{mm}$, $\epsilon_r = 9.9$, and $\mu_r = 1.0$. The effective dielectric constants, ϵ_M, were obtained from [3]. The impedances match quite well as can be seen in Table A.2.2. The largest difference, -7.6%, is for wide microstrips at high frequencies. This is to be expected since our trial functions do not take into account the more complex current distributions in that case.

Table A.1: Microstrip characteristic impedance

	$f = 3\,\text{GHz}$				$f = 6\,\text{GHz}$			
w_M	ϵ_M	$Z_M\ (\Omega)$		Δ	ϵ_M	$Z_M\ (\Omega)$		Δ
(mm)		Eq. A.29	Jansen	%		Eq. A.29	Jansen	%
.04	5.71	117.00	113.94	2.7	5.73	116.96	114.30	2.3
.08	6.00	101.27	98.73	2.6	6.06	101.47	100.04	1.4
.18	6.23	80.99	79.72	1.6	6.32	81.24	80.79	0.6
.40	6.48	60.65	60.24	0.7	6.58	60.77	61.07	-0.5
.60	6.68	50.66	50.73	-0.1	6.79	50.73	51.21	-0.9
1.00	7.00	38.75	39.09	-0.9	7.13	38.79	39.56	-1.9
1.40	7.28	31.72	31.84	-0.4	7.43	31.77	32.32	-1.7
2.00	7.59	25.18	25.31	-0.5	7.77	25.25	25.90	-2.5
3.00	7.97	19.06	19.13	-0.4	8.19	19.18	19.60	-2.1
	$f = 9\,\text{GHz}$				$f = 12\,\text{GHz}$			
w_M	ϵ_M	$Z_M\ (\Omega)$		Δ	ϵ_M	$Z_M\ (\Omega)$		Δ
(mm)		Eq. A.29	Jansen	%		Eq. A.29	Jansen	%
.04	5.76	117.20	115.37	1.6	5.82	117.99	116.55	1.2
.08	6.13	101.90	101.11	0.8	6.24	102.86	102.53	0.3
.18	6.41	81.63	81.98	-0.4	6.52	82.26	83.17	-1.1
.40	6.68	61.00	62.14	-1.8	6.80	61.50	63.21	-2.7
.60	6.91	50.95	52.28	-2.5	7.05	51.40	53.35	-3.7
1.00	7.29	39.03	40.51	-3.7	7.46	39.43	41.47	-4.9
1.40	7.60	32.03	33.27	-3.7	7.78	32.35	34.22	-5.5
2.00	7.96	25.47	26.73	-4.7	8.16	25.81	27.56	-6.3
3.00	8.38	19.37	20.44	-5.2	8.58	19.65	21.27	-7.6

APPENDIX B

CALCULATION OF THE SLOTLINE NORM

In Chapter 4 we found that in order to calculate the scattering coefficients for the microstrip-slotline crossover, we need to know the norm of the fundamental slotline mode. This appendix describes how to find the electromagnetic fields of this mode by using a good approximation for the slot electric field. From these fields, the norm can be found by performing an integration of the Poynting vector over a transverse surface intersecting the slotline. The norm can also be used to calculate the characteristic impedance of the slotline transmission line.

For this analysis, the slotline mode fields were normalized by setting the total electric field on the slotline equal to 1 V. The characteristic impedance is calculated based on a power-voltage definition. Some numerical results are summarized at the end of the appendix.

B.1 Slotline Mode Fields in the Spectral Domain

The derivation for the slotline mode fields proceeds very similar to the more general derivation found in Chapter 4. In this case, we have only a uniform slotline geometry. We assume that the fundamental slotline mode has been excited in some fashion. The equivalence principle allows us to convert the problem into a grounded dielectric slab geometry which has the slot electric field impressed as an equivalent magnetic current source. This

allows us to find the mode fields in terms of the slot electric field.

Since the fields and currents of the slotline fundamental mode behave as $e^{-i\beta_A y} = e^{-in_A k_0 y}$, where $n_A = \sqrt{\epsilon_A}$, the Helmholtz equation only needs to be transformed in the x-dimension. Equation (3.1) becomes

$$\left[\frac{d^2}{dz^2} - \left(\alpha^2 + n_A^2\right) k_0^2 + k^2\right] \hat{\Pi}_j = 0 \tag{B.1}$$

But this is just like (3.10), except that α is replaced by n_A. This means that the results (3.11) for the Hertz potentials are still valid. With the variable change $\alpha \rightarrow n_A$, the Hertz potentials are

$$\hat{\Pi}_j(\alpha, n_A, z) = \begin{cases} \mathcal{A}_j e^{-k_0 s_0 (z-d)} & (z \geq d) \\ \mathcal{B}_j \sinh(k_0 s_1 z) + \mathcal{C}_j \cosh(k_0 s_1 z) & (0 \leq z \leq d) \\ \mathcal{D}_j e^{k_0 s_0 z} & (z \leq 0) \end{cases} \tag{B.2}$$

The FT variables $\vec{\gamma}$, $\vec{\gamma}_\perp$, γ^2, u_0, u_1, D_e, and D_m from Chapter 3 are now mapped to the analogous variables

$$\vec{\psi} = \alpha \vec{a}_x + n_A \vec{a}_y \tag{B.3}$$

$$\vec{\psi}_\perp = n_A \vec{a}_x - \alpha \vec{a}_y \tag{B.4}$$

$$\psi^2 = \alpha^2 + n_A^2 \tag{B.5}$$

$$s_0 = \sqrt{\psi^2 - 1} \tag{B.6}$$

$$s_1 = \sqrt{\psi^2 - \epsilon_r \mu_r} \tag{B.7}$$

$$S_e = \epsilon_r s_0 \cosh(k_0 d s_1) + s_1 \sinh(k_0 d s_1) \tag{B.8}$$

$$S_m = \mu_r s_0 \sinh(k_0 d s_1) + s_1 \cosh(k_0 d s_1) \tag{B.9}$$

In addition, we introduce two new FT variables

$$R_e = \epsilon_r s_0 \sinh(k_0 d s_1) + s_1 \cosh(k_0 d s_1) \tag{B.10}$$

$$R_m = s_1 \sinh(k_0 d s_1) + \mu_r s_0 \cosh(k_0 d s_1) \tag{B.11}$$

The unknown amplitude coefficients in equation (B.2) are found by applying the boundary conditions. The tangential electric field has to be continuous across $z = 0$ and equal to the slot electric field $\vec{E}_A$. It also has to be continuous across $z = d$. The tangential magnetic field is continuous across $z = d$ since there are no electric currents there. Using these boundary conditions and solving the resulting equations for the coefficients yields

$$\begin{aligned}
\mathcal{A}_e &= -i\vec{\psi}\cdot\hat{\vec{E}}_A[R_e \sinh(k_0 d s_1)/S_e - \cosh(k_0 d s_1)]/(k_0^2\psi^2 s_0) \\
\mathcal{A}_m &= \vec{\psi}_\perp\cdot\hat{\vec{E}}_A[R_m \sinh(k_0 d s_1)/S_m - \cosh(k_0 d s_1)]/(\eta k_0^2\psi^2) \\
\mathcal{B}_e &= i\vec{\psi}\cdot\hat{\vec{E}}_A/(k_0^2\psi^2 s_1) \\
\mathcal{B}_m &= R_m\vec{\psi}_\perp\cdot\hat{\vec{E}}_A/(\mu_r\eta k_0^2\psi^2 S_m) \\
\mathcal{C}_e &= -iR_e\vec{\psi}\cdot\hat{\vec{E}}_A/(k_0^2\psi^2 s_1 S_e) \\
\mathcal{C}_m &= -\vec{\psi}_\perp\cdot\hat{\vec{E}}_A/(\mu_r\eta k_0^2\psi^2) \\
\mathcal{D}_e &= i\vec{\psi}\cdot\hat{\vec{E}}_A/(k_0^2\psi^2 s_0) \\
\mathcal{D}_m &= -\vec{\psi}_\perp\cdot\hat{\vec{E}}_A/(\eta k_0^2\psi^2)
\end{aligned} \tag{B.12}$$

We can now obtain the fields from (3.2) and (B.2). We will also make use of the definitions (B.3)–(B.11). Only the components which will actually be required have been derived here. The x and z-components of the electric field for $z \geq d$ are

$$\hat{E}^+_{Ax} = \frac{-e^{-k_0 s_0(z-d)}}{\psi^2}\left\{\alpha\vec{\psi}\cdot\hat{\vec{E}}_A\left[\frac{R_e}{S_e}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right] +\right.$$

$$n_A \vec{\psi}_\perp \cdot \hat{\bar{\bar{E}}}_A \left[\frac{R_m}{S_m}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right]\Bigg\} \tag{B.13}$$

$$\hat{E}^+_{Az} = \frac{i}{s_0}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A\left[\frac{R_e}{S_e}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right] e^{-k_0 s_0 (z-d)} \tag{B.14}$$

The magnetic field components for $z \geq d$ are

$$\hat{H}^+_{Ax} = \frac{i e^{-k_0 s_0 (z-d)}}{\eta \psi^2}\Bigg\{\frac{n_A}{s_0}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A\left[\frac{R_e}{S_e}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right] + \alpha s_0 \vec{\psi}_\perp \cdot \hat{\bar{\bar{E}}}_A\left[\frac{R_m}{S_m}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right]\Bigg\} \tag{B.15}$$

$$\hat{H}^+_{Az} = \frac{1}{\eta}\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\left[\frac{R_m}{S_m}\sinh(k_0 d s_1) - \cosh(k_0 d s_1)\right] e^{-k_0 s_0 (z-d)} \tag{B.16}$$

The electric field components for $0 \leq z \leq d$ are

$$\hat{E}^+_{Ax} = \frac{1}{\psi^2}\Bigg[\left(\alpha\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A + n_A\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\right)\cosh(k_0 s_1 z) - \left(\frac{\alpha R_e}{S_e}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A + \frac{n_A R_m}{S_m}\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\right)\sinh(k_0 s_1 z)\Bigg] \tag{B.17}$$

$$\hat{E}^+_{Az} = \frac{i}{s_1}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A\left[\sinh(k_0 s_1 z) - \frac{R_e}{S_e}\cosh(k_0 s_1 z)\right] \tag{B.18}$$

The magnetic field components for $0 \leq z \leq d$ are

$$\hat{H}^+_{Ax} = \frac{i}{\eta\psi^2}\Bigg[\left(\frac{\epsilon_r n_A}{s_1}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A + \frac{\alpha s_1}{\mu_r}\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\right)\sinh(k_0 s_1 z) - \left(\frac{\epsilon_r n_A R_e}{s_1 S_e}\vec{\psi}\cdot\hat{\bar{\bar{E}}}_A + \frac{\alpha s_1 R_m}{\mu_r S_m}\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\right)\cosh(k_0 s_1 z)\Bigg] \tag{B.19}$$

$$\hat{H}^+_{Az} = \frac{1}{\mu_r\eta}\vec{\psi}_\perp\cdot\hat{\bar{\bar{E}}}_A\left[\frac{R_m}{S_m}\sinh(k_0 s_1 z) - \cosh(k_0 s_1 z)\right] \tag{B.20}$$

The electric field components for $z \leq 0 \leq$ are

$$\hat{E}^+_{Ax} = \frac{1}{\psi^2}\left(\alpha\,\vec{\psi}\cdot\hat{\vec{E}}_A + n_A\,\vec{\psi}_\perp\cdot\hat{\vec{E}}_A\right)e^{k_0 s_0 z} \tag{B.21}$$

$$\hat{E}^+_{Az} = \frac{i}{s_0}\,\vec{\psi}\cdot\hat{\vec{E}}_A e^{k_0 s_0 z} \tag{B.22}$$

The magnetic field components for $z \leq 0$ are

$$\hat{H}^+_{Ax} = \frac{1}{\eta\psi^2}\left(\frac{n_A}{s_0}\,\vec{\psi}\cdot\hat{\vec{E}}_A + \alpha s_0\,\vec{\psi}_\perp\cdot\hat{\vec{E}}_A\right)e^{k_0 s_0 z} \tag{B.23}$$

$$\hat{H}^+_{Az} = \frac{-1}{\eta}\,\vec{\psi}_\perp\cdot\hat{\vec{E}}_A e^{k_0 s_0 z} \tag{B.24}$$

We can now compute the required norm.

B.2 Slotline Norm as a Spectral Domain Integral

To calculate the slotline norm, we repeat the first few steps of Section A.2 leading up to (A.22). The norm is found from

$$N_A = \frac{2\pi}{k_0}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\hat{\mathcal{E}}^+_{Ax}(\alpha)\hat{\mathcal{H}}^+_{Az}(-\alpha) - \hat{\mathcal{E}}^+_{Az}(-\alpha)\hat{\mathcal{H}}^+_{Ax}(\alpha)\right]\,dz\,d\alpha \tag{B.25}$$

The z-integrations are straightforward to perform. The α-integration needs to be performed numerically. We can simplify the last integral further by realizing that the only terms in the field solutions which are not even in α are $\vec{\psi}\cdot\hat{\vec{E}}_A$ and $\vec{\psi}_\perp\cdot\hat{\vec{E}}_A$. But we know that for the fundamental slotline mode, the longitudinal electric field component is odd about y and the transverse component is even. This results in the simple symmetry properties

$$\left.\begin{aligned}\vec{\psi}(-\alpha)\cdot\hat{\vec{\mathcal{E}}}_A(-\alpha) &= -\vec{\psi}(\alpha)\cdot\hat{\vec{\mathcal{E}}}_A(\alpha)\\ \vec{\psi}_\perp(-\alpha)\cdot\hat{\vec{\mathcal{E}}}_A(-\alpha) &= \vec{\psi}_\perp(\alpha)\cdot\hat{\vec{\mathcal{E}}}_A(\alpha)\end{aligned}\right\} \tag{B.26}$$

Using these symmetry properties and our field solutions in (B.25), results in the following general solution for the slotline norm

$$
\begin{aligned}
N_A \;=\; & \frac{2\pi}{\eta k_0}\int_{-\infty}^{\infty}\frac{1}{\psi^2}\Bigg\{\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A\Bigg[\frac{1}{2k_0s_0}\left(\frac{n_A}{s_0^2}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\alpha\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\\
&+\left(\frac{\epsilon_r n_A}{s_1^2}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{\alpha}{\mu_r}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\left[\frac{\sinh(2k_0ds_1)-2k_0ds_1}{4k_0s_1}\right]\\
&+\frac{R_e}{S_e}\left(\frac{\epsilon_r n_A R_e}{s_1^2S_e}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{\alpha R_m}{\mu_r S_m}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\\
&\times\left[\frac{\sinh(2k_0ds_1)+2k_0ds_1}{4k_0s_1}\right]+\frac{1-\cosh(2k_0ds_1)}{4k_0s_1}\\
&\times\left[\frac{2\epsilon_r n_A R_e}{s_1^2S_e}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{\alpha}{\mu_r}\left(\frac{R_e}{S_e}+\frac{R_m}{S_m}\right)\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right]\\
&+\frac{\epsilon_r}{2k_0S_e}\left(\frac{\epsilon_r n_A}{s_0S_e}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{\alpha s_1}{S_m}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\Bigg]\\
&+\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\Bigg[\frac{1}{2k_0s_0}\left(\alpha\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+n_A\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)+\frac{R_m}{\mu_r S_m}\\
&\times\left(\frac{\alpha R_e}{S_e}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{n_A R_m}{S_m}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\left[\frac{\sinh(2k_0ds_1)-2k_0ds_1}{4k_0s_1}\right]\\
&+\frac{1}{\mu_r}\left(\alpha\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+n_A\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)+\left[\frac{\sinh(2k_0ds_1)+2k_0ds_1}{4k_0s_1}\right]\\
&+\frac{1-\cosh(2k_0ds_1)}{4\mu_r k_0s_1}\left[\alpha\left(\frac{R_e}{S_e}+\frac{R_m}{S_m}\right)\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{2n_A R_m}{S_m}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right]\\
&+\frac{s_1}{2k_0S_m}\left(\frac{\epsilon_r\alpha}{S_e}\vec{\psi}\cdot\hat{\vec{\mathcal{E}}}_A+\frac{n_A s_1}{s_0S_m}\vec{\psi}_\perp\cdot\hat{\vec{\mathcal{E}}}_A\right)\Bigg]\Bigg\}\,d\alpha \qquad \text{(B.27)}
\end{aligned}
$$

B.2.1 Slot Electric Field Trial Function We now have to make some approximation for the slot electric field, $\vec{E}_A$, since no closed form solution is available for it. We assume only a transverse electric field exists across the slot. The following distribution is known to provide an accurate estimate for the transverse slot electric field

$$\vec{\mathcal{E}}_A \approx \mathcal{E}_{Ax}(x)\,\vec{a}_x = \frac{A_A \vec{a}_x}{\sqrt{(w_A/2)^2 - x^2}} \tag{B.28}$$

where A_A is the amplitude coefficient of $\mathcal{E}_{Ax}$. The slot voltage is found by integrating $\mathcal{E}_{Ax}$ across the slot width. The following result is obtained

$$V_A = \int_{-w_A/2}^{w_A/2} \frac{A_A}{\sqrt{(w_A/2)^2 - x^2}}\,dx = \pi A_A \tag{B.29}$$

We also need to convert the electric field (B.28) to its spectral domain equivalent. Applying our Fourier transform (3.4) and using the substitution $x = (w_A/2)\sin\xi$, yields the transformed electric field

$$\hat{\mathcal{E}}_{Ax}(\alpha) = \frac{k_0 A_A}{2\pi} \int_{-\pi/2}^{\pi/2} \cos\left(\frac{k_0 w_A}{2}\alpha\,\sin\xi\right)\,d\xi \tag{B.30}$$

But this is just the defining integral for the zeroth order Bessel function of the first kind, $J_0(k_0 w_A \alpha/2)$. In conjunction with (B.29) and our normalization condition that the voltage equals 1 V, the transformed electric field becomes

$$\hat{\mathcal{E}}_{Ax}(\alpha) = \frac{k_0}{2\pi} J_0(k_0 w_A \alpha/2) \tag{B.31}$$

B.2.2 Simplified Norm Integral Now that we have a trial function for the slot electric field, the norm integral (B.27) reduces to

$$
\begin{aligned}
N_A = \frac{\beta_A}{\pi\eta}\int_0^\infty \left[\frac{J_0(k_0 w_A \alpha/2)}{\psi}\right]^2 \Bigg\{ & \frac{\alpha^2+n_A^2}{2k_0 s_0} + \frac{R_m}{\mu_r S_m}\left(\frac{\alpha^2 R_e}{S_e}+\frac{n_A^2 R_m}{S_m}\right) \\
& \times \left[\frac{\sinh(2k_0 d s_1)-2k_0 d s_1}{4k_0 s_1}\right] + \frac{\alpha^2+n_A^2}{\mu_r}\left[\frac{\sinh(2k_0 d s_1)+2k_0 d s_1}{4k_0 s_1}\right] \\
& + \frac{1-\cosh(2k_0 d s_1)}{4\mu_r k_0 s_1}\left[\alpha^2\left(\frac{R_e}{S_e}+\frac{R_m}{S_m}\right)+\frac{2n_A^2 R_m}{S_m}\right] \\
& + \frac{s_1}{2k_0 S_m}\left(\frac{\epsilon_r \alpha^2}{S_e}+\frac{n_A^2 s_1}{s_0 S_m}\right) + \alpha^2\Bigg\{\frac{\epsilon_r}{2k_0 S_e}\left(\frac{\epsilon_r}{s_0 S_e}+\frac{s_1}{S_m}\right) \\
& + \frac{1}{2k_0 s_0}\left(1+\frac{1}{s_0^2}\right) + \frac{\alpha^2+n_A^2}{\mu_r s_1^2}\left[\frac{\sinh(2k_0 d s_1)-2k_0 d s_1}{4k_0 s_1}\right] \\
& + \frac{R_e}{S_e}\left(\frac{\epsilon_r R_e}{s_1^2 S_e}+\frac{R_m}{\mu_r S_m}\right)\left[\frac{\sinh(2k_0 d s_1)+2k_0 d s_1}{4k_0 s_1}\right] \\
& + \frac{1-\cosh(2k_0 d s_1)}{4k_0 s_1}\left[\frac{1}{\mu_r}\left(\frac{R_e}{S_e}+\frac{R_m}{S_m}\right)+\frac{2\epsilon_r R_e}{s_1^2 S_e}\right]\Bigg\}\Bigg\}\, d\alpha
\end{aligned}
\tag{B.32}
$$

This result was obtained by combining (B.27), (B.31), and (B.3)–(B.5).

The above integral was evaluated using the FORTRAN program ZA which is listed in Appendix D. The characteristic impedance of the slotline is obtained from

$$Z_A = \frac{V^2}{N_A} = \frac{1}{N_A} \tag{B.33}$$

This follows from the power-voltage definition for the characteristic impedance and the 1 V normalization.

Table B.1: Slotline characteristic impedance

Set 1				
w_A (mm)	ϵ_A	$Z_A\ (\Omega)$		Δ %
		Eq. B.32	Mariani	
.04	4.81	45.40	45.80	-0.9
.20	4.65	62.93	62.73	0.3
.40	4.52	75.38	74.96	0.6
.80	4.33	93.57	93.96	-0.4
1.20	4.19	108.56	108.82	-0.2
1.60	4.06	122.13	122.18	0.0
2.00	3.95	134.89	134.78	0.1
3.00	3.70	164.83	165.26	-0.3
4.00	3.49	193.18	193.29	-0.1
Set 2				
w_A (mm)	ϵ_A	$Z_A\ (\Omega)$		Δ %
		Eq. B.32	Mariani	
.02	9.05	32.01	32.21	-0.6
.10	8.56	44.22	44.41	-0.4
.20	8.21	52.85	53.05	-0.4
.40	7.77	65.26	65.26	0.0
.40	7.34	75.67	75.96	-0.4
.80	7.00	84.99	84.98	0.0
1.00	6.72	93.66	93.43	0.2
1.50	6.13	113.90	113.52	0.3
2.00	5.61	133.11	133.43	-0.2

The results of this analysis were verified by comparing the characteristic impedance values with published data from Mariani, et al [8]. The impedances match quite well as can be seen in Table B.2.2. The first set of impedances were calculated using the parameters $d = 2.0$ mm, $f = 6$ GHz, $\epsilon_r = 9.6$, and $\mu_r = 1.0$. The second set was calculated with $d = 1.0$ mm, $f = 6$ GHz, $\epsilon_r = 20$, and $\mu_r = 1.0$. The effective dielectric constants, ϵ_A, were obtained from Hoffmann [3].

APPENDIX C

SOME NUMERICAL ANALYSIS

The inner products of Chapter 5 had to be evaluated numerically. A number of mathematical tricks had to be employed to ensure accurate and speedy computation of the five integrals. The integrals can now be computed in as little as four minutes on a Hewlett-Packard series 370 workstation with its math coprocessor.

The following sections describe the analytical work that had to be performed on the six Sommerfeld integrals of Chapter 5 to make them amenable to computer solution. A large number of closed-form integrals were obtained from tables compiled by Gradshteyn and Ryzhik [23] or by Abramowitz and Stegun [24]. These were used in extracting various terms from the integrands.

The integrals derived here were often obtained by applying the principle of limiting absorption. As was already mentioned in Chapter 5, this simply involves making the effective refractive indexes slightly complex, say $n = n' - in''$, and taking the limit $n'' \to 0$. Thus, some of the results found in this appendix only hold when applying this principle.

C.1 I_{MM}

Recall the last section of Chapter 5 which discusses the difficulties encountered when trying to evaluate the Sommerfeld integrals. For the integrals which were converted to polar form, the order of integration was chosen to be first the γ-integration then the θ-integration. This is much more efficient since the limits of the outer integral are finite while the upper limit of the inner integral is infinite.

The general procedure for eliminating the numerical hurdles in the integral I_{MM}, given by (5.9), consists of the following:

1. Extract the double pole of the integrand.
2. Find the closed-form asymptotic solution for the resulting single θ-integral.
3. Subtract and add the dominant asymptotic term from the new double integral to help with the γ-convergence.
4. Find the closed-form asymptotic solution for the resulting single θ-integral.
5. Compute the residue for the D_ϵ-pole.
6. Find the asymptotic terms of the final double integral for large γ. Compute the γ-integration from γ_l to infinity in closed form, where γ_l is a large value of γ.
7. Perform the outer θ-integration. Step 3 results in a square-root type singularity in θ which requires that the outer integral itself be split into two parts. To the singular part we apply a special version of the Gaussian integration procedure which can handle exactly such a singularity.

As it turns out, steps 2 and 4 are actually the most important. It is the single integrals resulting from the double-pole extraction and the asymptotic-term subtraction which provide the most significant contribution to I_{MM}. Worse yet, the real parts of the two single integrals often cancel in the first two significant digits which requires that they be computed to one or two additional significant digits than the desired accuracy of I_{MM}.

C.1.1 Double-Pole Extraction Let's begin by writing the integral I_{MM} as

$$I_{MM} = -\frac{4i\eta n_M^2}{\pi^2} \int_0^{\pi/2} \int_0^{\infty} \frac{\gamma J_0^2(r\gamma)}{(\gamma^2 - q^2)^2} F_1(\gamma^2)\, d\gamma\, d\theta \tag{C.1}$$

where $r = k_0 w_M \sin\theta/2$, $q = n_M/\cos\theta$, and

$$F_1(\gamma^2) = \sinh(k_0 d u_1)\left(\frac{u_0 u_1}{D_e \cos^2\theta} - \frac{\mu_r \sin^2\theta}{D_m \cos^4\theta}\right) \tag{C.2}$$

In order to extract the double pole, we want to add and subtract terms from the integrand which can be found in closed form and which combine with the original term to yield a second-order zero thus canceling the double pole. We do this by expanding the function F_1 in a Taylor series around $\gamma^2 = q^2$. Letting primes denote differentiation with respect to γ^2, we have

$$F_1(\gamma^2) = F_1(q^2) + (\gamma^2 - q^2)F_1'(q^2) + \frac{(\gamma^2 - q^2)^2}{2} F_1''(q^2) + \cdots \tag{C.3}$$

so that by subtracting the first two terms from $F_1(\gamma^2)$, we will be left with a function that has the required second-order zero at $\gamma^2 = q^2$. Of course, we must add the same two terms to our final result in closed form. To this end

we need the following two integrals

$$\begin{aligned} I_1' &= \int_0^\infty \frac{\gamma J_0^2(r\gamma)}{\gamma^2 - q^2}\, d\gamma \\ &= -\frac{i\pi}{2} J_0(rq) H_0^{(2)}(rq) \\ &= -\frac{\pi}{2} J_0(rq)[iJ_0(rq) + Y_0(rq)] \end{aligned} \tag{C.4}$$

and

$$\begin{aligned} I_1 &= \int_0^\infty \frac{\gamma J_0^2(r\gamma)}{(\gamma^2 - q^2)^2}\, d\gamma \\ &= -\frac{\pi r}{4q}\{J_0(rq)[Y_1(rq) + 2iJ_1(rq)] + J_1(rq)Y_0(rq)\} \end{aligned} \tag{C.5}$$

The first of these integrals is obtained by applying the principle of limiting absorption to a related integral obtained from [23] by writing the denominator as $\gamma^2 + (iq)^2$. The second integral follows by differentiating the first with respect to q and by making use of the relationships between Bessel functions and their derivatives. We can now rewrite I_{MM} as a sum of two integrals

$$I_{MM} = -\frac{4i\eta n_M^2}{\pi^2}\left\{ I_d + \int_0^{\pi/2} \left[F_1(q^2) I_1 + F_1'(q^2) I_1'\right] d\theta \right\} \tag{C.6}$$

where

$$I_d = \int_0^{\pi/2} \int_0^\infty \frac{\gamma J_0^2(r\gamma)}{(\gamma^2 - q^2)^2} \left[F_1(\gamma^2) - F_1(q^2) - (\gamma^2 - q^2) F_1'(q^2)\right] d\gamma\, d\theta \tag{C.7}$$

The derivative of F_1 with respect to γ^2 is given by

$$
\begin{aligned}
F_1' &= \frac{1}{2D_e^2\cos^2\theta}\left\{\left[\frac{\epsilon_r u_0^2\cosh(k_0du_1)}{u_1}+\frac{u_1^2\sinh(k_0du_1)}{u_0}\right]\right. \\
&\quad \left.\times\sinh(k_0du_1)+\epsilon_r k_0du_0^2\right\} \\
&\quad +\frac{\mu_r\sin^2\theta}{2u_1D_m^2\cos^4\theta}\left\{\left[\cosh(k_0du_1)+\frac{\mu_r u_1}{u_0}\sinh(k_0du_1)\right]\right. \\
&\quad \left.\times\sinh(k_0du_1)-k_0du_1\right\} \qquad \text{(C.8)}
\end{aligned}
$$

The single integral in (C.6) has a singularity at $\theta = \pi/2$. This could not be eliminated numerically since the integrand oscillates infinitely many times with ever increasing amplitude as the singularity is approached. We had to use the first two terms in the asymptotic expansion for the Bessel and Neumann functions in conjunction with the asymptotic forms of F_1 and F_1' to derive the following closed-form solution (the symbol "$\sim$" means asymptotically equal to):

$$
\int_{\pi/2-\delta}^{\pi/2}\left[F_1(q^2)I_1+F_1'(q^2)I_1'\right]d\theta \sim (\bar{\mu}-\bar{\epsilon})\left(\frac{\cos\zeta-i\sin\zeta}{\zeta}\right)-\frac{2i\bar{\mu}}{\zeta} \qquad \text{(C.9)}
$$

where $\zeta = n_M k_0 w_M/\delta$, $\bar{\mu} = \mu_r/[2n_M^3(1+\mu_r)]$, and $\bar{\epsilon} = 1/[2n_M(1+\epsilon_r)]$. This term is added to the numerical solution of the integral up to $\theta = \pi/2-\delta$. The value of δ is chosen such that $rq \geq 50$.

C.1.2 Asymptotic-Term Subtraction Calculating the inner γ-integral of I_d of (C.7) accurately and in an acceptable amount of time requires that we encourage a rapid decay with increasing γ. This is done by subtracting from the integrand a term which asymptotically cancels the dominant term. An asymptotic analysis of I_d shows that subtracting the

following term I_a achieves this objective:

$$I_a = -\int_0^{\pi/2}\int_0^{\infty} \frac{\gamma J_0^2(r\gamma)F_1'(q^2)}{\gamma^2+g^2} d\gamma\, d\theta \tag{C.10}$$

This results in our modified version of I_d:

$$I_d = \bar{I}_d - \int_0^{\pi/2} F_1'(q^2)I_0(rg)K_0(rg)d\theta \tag{C.11}$$

where $\bar{I}_d = I_d - I_a$. We have once again made use of the table of integrals [23]. The functions $I_0(rg)$ and $K_0(rg)$ are the modified Bessel and Neumann functions of zeroth order. The variable g is chosen so that it cancels the $1/\cos\theta$ singularity in F_1', or $g = 10/\sqrt{\cos\theta}$. The constant of 10 is used for numerical reasons.

The new single integral appearing above has its own difficulties as $\theta \to \pi/2$. The following closed-form result was added to the numerically computed solution from $\theta = 0$ to $\theta = \pi/2 - \delta$:

$$-\int_{\pi/2-\delta}^{\pi/2} F_1'(q^2)I_0(rg)K_0(rg)\, d\theta \sim -\frac{(\bar{\mu}-\bar{\epsilon})\sqrt{\delta}}{5k_0 w_M} \tag{C.12}$$

C.1.3 D_e-Pole Residue The double integral now needs to be broken into two sections so as to avoid the pole of the function $1/D_e$. This pole corresponds to the effective refractive index of the TM_0 surface wave of the grounded dielectric slab [21]. It's value is calculated by the program `DEPOL` using Newton's method. Since the integration is along the γ-axis, we only need half of the residue of this pole. Letting γ_p denote the location of the pole along this axis, the half-residue is easily found:

$$I_r = -\frac{4\eta n_M^2 \gamma_p u_0 u_1 \sinh(k_0 d u_1)}{\pi \dot{D}_e(\gamma_p)} \int_0^{\pi/2} \frac{J_0^2(r\gamma_p)\, d\theta}{(\gamma_p^2 - q^2)^2 \cos^2\theta} \tag{C.13}$$

All terms which are functions of γ are evaluated at γ_p. The function $\dot{D}_e$ is the derivative of D_e with respect to γ and is given by

$$\dot{D}_e = \gamma \cosh(k_0 d u_1)\left(k_0 d + \frac{\epsilon_r}{u_0}\right) + \frac{\gamma \sinh(k_0 d u_1)}{u_1}\left(1 + \epsilon_r k_0 d u_0\right) \qquad \text{(C.14)}$$

The residue is simply added to I_{MM}. The integration in γ must be performed to within a small distance, Δ, away from γ_p. Δ is one of the variable integration parameters found in the programs. Generally, it was found that a value of $\Delta = 10^{-5}$ was more than close enough. Of course, the location of the pole has to be determined to double precision for the integration to remain accurate (at least 10 significant digits).

C.1.4 Closed-Form Asymptotic Solutions Even with all this manipulation, the double integral $\bar{I}_d$ obtained above still does not converge fast enough. Hence it is necessary to compute all the closed-form solutions of the asymptotic portions of the integrand. To this end, we choose a large value of γ based on $\gamma_l = \max(100/r, q)$. This assures that the asymptotic forms of the Bessel and Neumann functions hold and that the integrand is past the double pole.

The derivation for the asymptotic results is by no means trivial. However, it is straightforward enough that it will not be repeated here. Breaking the integral $\bar{I}_d$ into two portions

$$\bar{I}_d = \int_0^{\gamma_l} \tilde{I}_d \, d\gamma + \int_{\gamma_l}^{\infty} \tilde{I}_d \, d\gamma \qquad \text{(C.15)}$$

allows us to write the asymptotic solutions as

$$\int_{\gamma_l}^{\infty} \tilde{I}_d \, d\gamma \sim \Lambda_1 + \Lambda_2 + \Lambda_3 + \Lambda_4 + \Lambda_5 \qquad \text{(C.16)}$$

where the Λ's are defined by

$$\Lambda_1 = \frac{-\mu_r \sin^2\theta\, A_1}{\pi(1+\mu_r) r \cos^4\theta} \tag{C.17}$$

$$\Lambda_2 = \frac{[q^2 F_1'(q^2) - F_1(q^2)] A_2}{\pi r} \tag{C.18}$$

$$\Lambda_3 = \frac{A_3}{\pi(1+\epsilon_r) r \cos^2\theta} \tag{C.19}$$

$$\Lambda_4 = \frac{-F_1'(q^2)\, A_4}{\pi r} \tag{C.20}$$

and

$$\Lambda_5 = \frac{F_1'(q^2)}{\pi r g}\Big\{\cosh(2rg)\Im[\mathrm{si}(\nu)] - \sinh(2rg)\Re[\mathrm{Ci}(\nu)] - \tan^{-1}(\gamma_l/g) + \pi/2\Big\} \tag{C.21}$$

In the last equation $\nu = 2r(\gamma_l + ig)$. The sine and cosine integrals are as defined in Abramowitz [24]. The terms A_i are defined by the following integrals

$$A_1 = \int_{\gamma_l}^{\infty} \frac{[1+\sin(2r\gamma)]\, d\gamma}{\gamma(\gamma^2 - q^2)^2} \tag{C.22}$$

$$A_2 = \int_{\gamma_l}^{\infty} \frac{[1+\sin(2r\gamma)]\, d\gamma}{(\gamma^2 - q^2)^2} \tag{C.23}$$

$$A_3 = \int_{\gamma_l}^{\infty} \frac{\gamma[1+\sin(2r\gamma)]\, d\gamma}{(\gamma^2 - q^2)^2} \tag{C.24}$$

$$A_4 = \int_{\gamma_l}^{\infty} \frac{\gamma^2[1+\sin(2r\gamma)]\, d\gamma}{(\gamma^2 - q^2)^2} \tag{C.25}$$

When solved by using partial fractions and a table of integrals, they reduce to

$$A_1 = \frac{1+\sin(2r\gamma_l)}{2q^2(\gamma_l^2-q^2)} + \frac{1}{2q^4}\Big\{\Big[\cos(2rq)+rq\sin(2rq)\Big]\tilde{\Sigma} + \Big[\sin(2rq)-rq\cos(2rq)\Big]\Upsilon - \ln\Big(\frac{\gamma_l^2}{\gamma_l^2-q^2}\Big) - 2\,\mathrm{si}(2r\gamma_l)\Big\} \quad \text{(C.26)}$$

$$A_2 = \frac{\gamma_l[1+\sin(2r\gamma_l)]}{2q^2(\gamma_l^2-q^2)} + \frac{1}{4q^3}\Big\{\Big[\cos(2rq)+2rq\sin(2rq)\Big]\Sigma + \Big[\sin(2rq)-2rq\cos(2rq)\Big]\tilde{\Upsilon} + \ln\Big(\frac{\gamma_l-q}{\gamma_l+q}\Big)\Big\} \quad \text{(C.27)}$$

$$A_3 = \frac{1+\sin(2r\gamma_l)}{2(\gamma_l^2-q^2)} + \frac{r}{2q}\Big[\sin(2rq)\tilde{\Sigma} - \cos(2rq)\Upsilon\Big] \quad \text{(C.28)}$$

$$A_4 = \frac{\gamma_l[1+\sin(2r\gamma_l)]}{2(\gamma_l^2-q^2)} - \frac{1}{4q}\Big\{\Big[\cos(2rq)-2rq\sin(2rq)\Big]\Sigma + \Big[\sin(2rq)+2rq\cos(2rq)\Big]\tilde{\Upsilon} + \ln\Big(\frac{\gamma_l-q}{\gamma_l+q}\Big)\Big\} \quad \text{(C.29)}$$

with $\kappa = 2r(\gamma_l - q)$, $\tilde{\kappa} = 2r(\gamma_l + q)$, $\Sigma = \mathrm{si}(\kappa) - \mathrm{si}(\tilde{\kappa})$, $\tilde{\Sigma} = \mathrm{si}(\kappa) + \mathrm{si}(\tilde{\kappa})$, $\Upsilon = \mathrm{Ci}(\kappa) - \mathrm{Ci}(\tilde{\kappa})$, and $\tilde{\Upsilon} = \mathrm{Ci}(\kappa) + \mathrm{Ci}(\tilde{\kappa})$. The results of this section have been implemented in the program `PIMM` located in Appendix D.

C.2 I_{AA}

Luckily, the approach used to make I_{AA} manageable is perfectly analogous to that used for I_{MM}. The primary difference is that the integrand of I_{AA} also contains a $1/u_0$ singularity which must be avoided. Thus, the γ-integration is broken into segments around both the singularity and the D_e-pole. Since these two can be very close to each other, especially

at low frequencies where $\gamma_p \approx 1$, the program PIAA has to make sure that the values of Δ and Δ_2 which are chosen to integrate up to the pole and the singularity, respectively, aren't too large. The program must also dynamically modify the relative error parameter for the adaptive integration routine. This is necessary because the adaptive integration routine produces incorrect results when the relative error is set too small on a function displaying singular behavior. The algorithms which perform these tasks were developed empirically.

We begin as in the previous section by writing I_{AA} from (5.13) as

$$I_{AA} = -\frac{4in_A^2}{\eta\pi^2}\int_0^{\pi/2}\int_0^{\infty}\frac{\gamma J_0^2(a\gamma)}{(\gamma^2-b^2)^2}F_2(\gamma^2)\,d\gamma\,d\theta \tag{C.30}$$

where $a = k_0 w_A \cos\theta/2$, $b = n_A/\sin\theta$, and

$$F_2(\gamma^2) = \frac{\mathcal{F}\cos^2\theta}{D_e\sin^4\theta} + \frac{\mathcal{G}}{D_m\sin^2\theta} \tag{C.31}$$

Here we made use of our definitions following (3.42) for $\mathcal{F}$ and $\mathcal{G}$. The procedure for extracting the double pole is identical to the one leading up to (C.6). We simply apply the transformation $r \Rightarrow a$ and $q \Rightarrow b$. This allows us to write

$$I_{AA} = -\frac{4in_A^2}{\eta\pi^2}\left\{I_e + \int_0^{\pi/2}\left[F_2(b^2)I_2 + F_2'(b^2)I_2'\right]d\theta\right\} \tag{C.32}$$

I_2 and I_2' are the same functions as I_1 and I_1' except with the above variable change applied. The integral I_e is just like I_d

$$I_e = \int_0^{\pi/2}\int_0^{\infty}\frac{\gamma J_0^2(a\gamma)}{(\gamma^2-b^2)^2}\left[F_2(\gamma^2) - F_2(b^2) - (\gamma^2-b^2)F_2'(b^2)\right]d\gamma\,d\theta \tag{C.33}$$

The derivative of F_2 with respect to γ^2 is given by

$$\begin{aligned} F_2' &= \frac{-k_0 d\cosh(k_0 d u_1)}{2u_1 \sinh(k_0 d u_1)\sin^2\theta}\left(\frac{\mathcal{F}\cos^2\theta}{D_e\sin^2\theta}+\frac{\mathcal{G}}{D_m}\right)+\frac{1}{\sinh(k_0 d u_1)\sin^2\theta} \\ &\times\left\{\frac{1}{\mu_r}\left[D_m'\left(1+\frac{u_1^2}{D_m^2}\right)-\frac{1}{D_m}\right]+\frac{\cos^2\theta}{u_0 u_1\sin^2\theta}\left[\frac{D_e}{2}\left(\frac{1}{u_0^2}+\frac{1}{u_1^2}\right)\right.\right. \\ &\left.\left.-D_e'+\frac{\epsilon_r^2}{D_e}\left(\frac{1}{2}-\frac{u_0^2}{2u_1^2}-\frac{u_0^2 D_e'}{D_e}\right)\right]\right\} \end{aligned} \tag{C.34}$$

where

$$D_e' = \frac{\cosh(k_0 d u_1)}{2}\left(k_0 d+\frac{\epsilon_r}{u_0}\right)+\frac{\sinh(k_0 d u_1)}{2u_1}\left(1+\epsilon_r k_0 d u_0\right) \tag{C.35}$$

$$D_m' = \frac{\sinh(k_0 d u_1)}{2}\left(k_0 d+\frac{\mu_r}{u_0}\right)+\frac{\cosh(k_0 d u_1)}{2u_1}\left(1+\mu_r k_0 d u_0\right) \tag{C.36}$$

The singularity in the single integral of (C.32) is now located at $\theta = 0$, but the asymptotic portion of the integral comes out very similar to our previous result for I_{MM}

$$\int_0^\delta \left[F_2(b^2)I_2+F_2'(b^2)I_2'\right]d\theta \sim \left(\frac{\tilde{\epsilon}-\tilde{\mu}}{2}\right)\left(\frac{\cos\bar{\zeta}-i\sin\bar{\zeta}}{\bar{\zeta}}\right)-\frac{i\tilde{\epsilon}}{\bar{\zeta}} \tag{C.37}$$

where $\bar{\zeta} = n_A k_0 w_A/\delta$, $\tilde{\mu} = (1+\mu_r)/(n_A\mu_r)$, and $\tilde{\epsilon} = (1+\epsilon_r)/n_A^3$. This term is added to the numerical solution of the integral up to $\theta = \pi/2-\delta$. The value of δ is chosen such that $ab \geq 50$.

Next, we subtract the following asymptotic term from I_e to speed up the γ-convergence:

$$I_b = -\int_0^{\pi/2}\int_0^\infty \frac{\gamma J_0^2(a\gamma)F_2'(b^2)}{\gamma^2+h^2}d\gamma\, d\theta \tag{C.38}$$

The double integral is now given by

$$I_e = \bar{I}_e - \int_0^{\pi/2} F_2'(b^2) I_0(ah) K_0(ah) d\theta \tag{C.39}$$

where $\bar{I}_e = I_e - I_b$. The variable h is chosen so that it cancels the $1/\sin\theta$ singularity in F_2', i.e., $h = 10/\sqrt{\sin\theta}$. The single integral appearing above has a singularity at $\theta = 0$ which is handled analytically via

$$-\int_0^{\delta} F_2'(b^2) I_0(ah) K_0(ah)\, d\theta \sim -\frac{(\tilde{\epsilon} + \tilde{\mu})\sqrt{\delta}}{5 k_0 w_A} \tag{C.40}$$

The half-residue corresponding to the D_e-pole is calculated as

$$I_r = -\frac{4 n_A^2 \epsilon_r^2 \gamma_p u_0}{\eta \pi u_1 \sinh(k_0 d u_1) \dot{D}_e} \int_0^{\pi/2} \frac{J_0^2(a\gamma_p)\cos^2\theta\, d\theta}{(\gamma_p^2 - b^2)^2 \sin^4\theta} \tag{C.41}$$

The closed-form asymptotic solutions for the γ-integration consist of five terms as before

$$\int_{\gamma_l}^{\infty} \tilde{I}_e\, d\gamma \sim \Lambda_6 + \Lambda_7 + \Lambda_8 + \Lambda_9 + \Lambda_{10} \tag{C.42}$$

Λ_{10} is identical in form to Λ_5 if we remember to replace r, q, g, and F_1 with a, b, h, and F_2, respectively. The remaining four terms are given by

$$\Lambda_6 = -\frac{(1 + \epsilon_r)\cos^2\theta\, A_6}{\pi a \sin^4\theta} \tag{C.43}$$

$$\Lambda_7 = \frac{[b^2 F_2'(b^2) - F_2(b^2)] A_7}{\pi a} \tag{C.44}$$

$$\Lambda_8 = \frac{(1 + \mu_r) A_8}{\mu_r \pi a \sin^2\theta} \tag{C.45}$$

$$\Lambda_9 = -\frac{F_2'(b^2)\, A_9}{\pi a} \tag{C.46}$$

The terms A_6, A_7, A_8, and A_9 correspond to (C.26)–(C.29) when the above mentioned variable changes are made. The results have been coded up in the program PIAA.

C.3 I_{AM}

The remaining integrals are much more straightforward to handle. Recall I_{AM} from (5.15)

$$I_{AM} = -\frac{4n_A n_M}{\pi^2} \int_0^{\pi/2} \int_0^{\infty} \frac{\gamma J_0(a\gamma) J_0(r\gamma)}{(\gamma^2 - b^2)(\gamma^2 - q^2)} F_3(\gamma^2)\, d\gamma\, d\theta \tag{C.47}$$

where

$$F_3(\gamma^2) = \left(\frac{\epsilon_r u_0}{D_e \sin^2\theta} + \frac{u_1}{D_m \cos^2\theta} \right) \tag{C.48}$$

This integral contains only two simple poles which are readily extracted by subtracting the first term of the Taylor series expansion for F_3 around each pole. The integral relation we use for this purpose is found in [23]

$$\int_0^{\infty} \frac{\gamma J_0(u_> \gamma) J_0(u_< \gamma)}{\gamma^2 - v^2}\, d\gamma = -\frac{i\pi}{2} J_0(u_< v) H_0^{(2)}(u_> v) \tag{C.49}$$

where $u_<$ and $u_>$ are the smaller and greater of two numbers, respectively, and v is some constant. In our case, $u_< = \min(a, r)$ and $u_> = \max(a, r)$, while v will equal either b or q.

Applying this identity to I_{AM} results in the following form of the integral which was coded as the program PIAM:

$$\begin{aligned} I_{AM} \;=\; & -\frac{4n_A n_M}{\pi^2} \int_0^{\pi/2} \int_0^{\infty} \frac{1}{(q^2 - b^2)} \left\{ \frac{\gamma J_0(u_< \gamma) J_0(u_> \gamma)}{\gamma^2 - q^2} \left[F_3(\gamma^2) - F_3(q^2) \right] \right. \\ & \left. - \frac{\gamma J_0(u_< \gamma) J_0(u_> \gamma)}{\gamma^2 - b^2} \left[F_3(\gamma^2) - F_3(b^2) \right] \right. \end{aligned}$$

$$- \frac{i\pi}{2}\Big[F_3(q^2)J_0(u_<q)H_0^{(2)}(u_>q)$$

$$- F_3(b^2)J_0(u_<b)H_0^{(2)}(u_>b)\Big]\Big\}d\gamma\, d\theta \qquad \text{(C.50)}$$

The half-residue due to the D_e-pole is obtained as

$$I_r = \frac{4in_M n_A \epsilon_r \gamma_p u_0}{\pi \dot{D}_e} \int_0^{\pi/2} \frac{J_0(u_>\gamma_p)J_0(u_<\gamma_p)\, d\theta}{(\gamma_p^2 - q^2)(\gamma_p^2 - b^2)\sin^2\theta} \qquad \text{(C.51)}$$

which is simply added to the above result for I_{AM}.

C.4 I_{MI}

Upon inspection of the integral I_{MI} in (5.18), one finds that it actually consists of two independent single integrals in α and λ. We can quickly solve the λ-integral by closing the contour of integration at infinity in the upper half-plane. Recalling our principle of limiting absorption, the residue of the pole $\lambda = -n_A$ is found to be

$$\int_{-\infty}^{\infty}\left[\frac{J_0(\bar{\lambda})e^{ik_0 w_M(\lambda+n_A)/2}}{(\lambda + n_A)}\right] d\lambda = 2\pi i J_0(n_A k_0 w_M/2) \qquad \text{(C.52)}$$

The α-integral contains a simple pole which we extract in the standard manner. We find that

$$I_{MI} = J_0(n_A k_0 w_M/2)\Big\{\frac{2in_M}{\pi}\int_0^{\infty}\frac{J_0(\bar{\alpha})}{\alpha^2 - n_M^2}\Big[F_4(\alpha^2) - F_4(n_M^2)\Big]\, d\alpha$$

$$+ F_4(n_M^2)\Big[J_0(n_M k_0 w_A/2) - i\mathbf{H}_0(n_M k_0 w_A/2)\Big]\Big\} \qquad \text{(C.53)}$$

where

$$F_4(\alpha^2) = \left[\cosh(k_0 d s_1) - \frac{\sinh(k_0 d s_1)}{\alpha^2 + n_A^2}\left(\frac{\alpha^2 R_e}{S_e} + \frac{n_A^2 R_m}{S_m}\right)\right] \qquad \text{(C.54)}$$

and $\mathbf{H}_0(x)$ is the Struve function of zeroth order. The modified integral I_{MI} converges quickly. The program PIMI made use of the fact that F_4 can be reduced by using the definitions (B.3)–(B.11). After some algebra, one obtains

$$F_4(\alpha^2) = \frac{1}{\alpha^2 + n_A^2}\left(\frac{\epsilon_r \alpha^2 s_0}{S_e} + \frac{n_A^2 s_1}{S_m}\right) \tag{C.55}$$

C.5 I_{AG}

The integral I_{AG} is almost an exact dual of I_{MI}. In this case, the α-integration is performed by closing the contour at infinity. One obtains

$$\int_{-\infty}^{\infty}\left[\frac{J_0(\bar{\alpha})e^{ik_0 w_A(\alpha+n_M)/2}}{(\alpha+n_M)}\right]d\alpha = 2\pi i J_0(n_M k_0 w_A/2) \tag{C.56}$$

The pole that needs extracting is located at $\lambda = n_A$. The result is

$$\begin{aligned} I_{AG} \;=\; & J_0(n_M k_0 w_A/2)\Bigg\{\frac{-2in_A}{\pi}\int_0^{\infty}\frac{J_0(\bar{\lambda})}{\lambda^2 - n_A^2}\left[F_5(\lambda^2) - F_5(n_A^2)\right]d\lambda \\ & - F_5(n_A^2)\left[J_0(n_A k_0 w_M/2) - i\mathbf{H}_0(n_A k_0 w_M/2)\right]\Bigg\} \end{aligned} \tag{C.57}$$

where

$$F_5(\alpha^2) = \frac{1}{\lambda^2 + n_M^2}\left(\frac{\epsilon_r n_M^2 t_0}{T_e} + \frac{\lambda^2 t_1}{T_m}\right) \tag{C.58}$$

The above result is computed by the program PIAG.

C.6 I'_{AG}

The last integral can be done entirely in closed form by performing contour integrations similar to those which were done for I_{MI} and I_{AG}. A quicker way to obtain the same result is to recognize that the space-domain integration is mathematically identical to our Fourier transform definition

(3.4) if one makes the changes of variable $\alpha \Rightarrow -n_M$ and $\lambda \Rightarrow n_A$. We obtain the following solution for (5.27)

$$I'_{AG} = -\frac{J_0(n_M k_0 w_A/2) J_0(n_A k_0 w_M/2)}{n_A^2 + n_M^2} \left(\frac{\epsilon_r n_M^2 t_0}{T_e} + \frac{n_A^2 t_1}{T_m} \right) \Bigg|_{\lambda = n_A} \qquad \text{(C.59)}$$

This result has been incorporated into the program PIAG.

APPENDIX D

FORTRAN PROGRAMS FOR THE CALCULATION OF THE CROSSOVER SCATTERING PARAMETERS

This appendix contains a listing of the FORTRAN programs which were used to calculate the Sommerfeld integrals of Chapter 5. A total of eleven programs were written for the computation of the scattering parameters of the microstrip-slotline crossover. The adaptive integration routines were adapted from a similar routine published by Forsythe [25]. The values for the Gaussian integration routines came from Abramowitz and Stegun [24]. The program `TCAA` uses a golden section search to bracket and locate a function minimum. This was adapted from the very useful handbook by Press, et al [26].

The programs were written in a version of FORTRAN 77 which runs on Hewlett-Packard computers, specifically the HP 9000 Series 300 engineering workstations. The operating system used was HP-UX 6.02.

The programs were originally written for a 1968 vintage Cyber mainframe running the KRONONS operating system. On this computer, it was not clear to the author how subroutines could be compiled independently and linked to a main program. As a result, the programs contain a fair amount of duplication. To save space in this thesis, the common subroutines have been printed only once and replaced by a comment line in the other programs. This comment line begins with a % symbol in column 1 of the

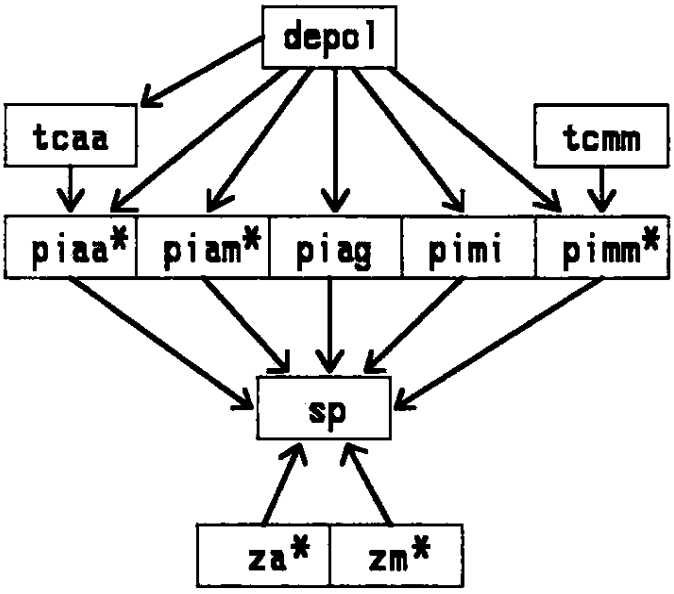
programs. To obtain working copies of the programs, simply insert the referenced subroutines at the indicated positions using a text editor.

D.1 Control Programs

Following are the Unix operating system programs and input files which are needed by the remaining eleven FORTRAN programs. These are listed by name in the indicated sections.

Program Flowchart

The following chart shows the relationships between the programs. All eleven programs use the input file phys.param, but only the files marked with an * use the input file program.param.



Control Program

When running these programs unmodified under a UNIX operating system, the program and data files have to be organized in three directories. The two input files, `phys.param` and `program.param`, are in the directory `defined.data`. The eleven output files are saved in the directory `computed.data`. All eleven programs and this control program are in the directory `programs`. Execution of the programs must take place in this last directory.

The following is an HP-UX shell script. It goes to the directory called `/users/alexa/thesis/programs` and executes the compiled FORTRAN programs in the order shown. A file called `tout` is created when running `depol.x`, and all the other programs append their output to it. `tout` is a log which contains intermediary results useful for verifying numerical convergence of the integration routines. The last line prints `tout` to the default line printer.

```
cd /users/alexa/thesis/programs
depol.x>tout
tcaa.x>>tout
tcmm.x>>tout
pimm.x>>tout
piaa.x>>tout
piam.x>>tout
piag.x>>tout
pimi.x>>tout
zm.x>>tout
za.x>>tout
sp.x>>tout
lp tout ../computed.data/SPout
```

Sample Input Data Files

The file `program.param` consists of miscellaneous parameters used by the integration routines. It is read by the programs PIMM, PIAA, PIAM, ZA, & ZM. Data fields must be specified exactly as below. The parameters are:

program.param

```
1.D-8, 1.570796326
1.D-8
6, 4, 20
1.D-5
4, 8
1.D-5
1.D6
6, 4
1.D-7
```

```
 1st&2nd = Approximations for THETA = 0, THETA = PI/2
     3rd = Approximation for GAMMA = 0
 4th&5th = Number of segments & number of functions evaluated per
           segment for the outer theta integration in PIMM
     6th = Number of functions evaluated for the singular portion
           of the theta integration in PIMM and PIAA
     7th = The initial value of DELTA = the distance up to the
           1/DE pole at which the numerical integration stops
 8th&9th = Number of segments & number of functions evaluated per
           segment for the outer theta integration in PIAA
    10th = Initial value of DLTA2 = the distance up to the 1/U0
           singularity at which the numerical integration stops
           in PIAA
    11th = Approximation for GAMMA = infinity (used for
           integrals in PIAM, ZA, & ZM)
12th&13th = Number of segments & number of functions evaluated per
           segment for the outer theta integration in PIAM
    14th = Relative error used in the single integration routines
           in ZA and ZM
```

The file `phys.param` contains the physical parameters used in the microstrip-slotline analysis. All programs use these parameters. The file format must be specified exactly as noted below the data. The parameters and their definitions are:

Data file phys.param

```
 20.D+09
 .6D-03
 9.7
 1.0
 .6D-03
 7.25
 .6D-03
 3.97

1st = Frequency in Hertz
2nd = Substrate thickness in meters
3rd = Relative dielectric constant
4th = Relative permeability
5th = Microstrip width in meters
6th = Effective dielectric constant for microstrip
7th = Slot line width in meters
8th = Effective dielectric constant for slot line
```

Sample Output Data Files

GAMA0

```
  .10372953599401E+01
```

THETAC.IAA

```
 1.78433408782877E-002
```

IAA

```
 (-1.37937590064409E-002,7.39165432438598E-003)
```

THETAC.IMM

```
 1.470796325304774
```

IMM

```
 (-100.9131604453773,32.79281158295177)
```

IAM

```
 (.347871315778705,-.814169840051225)
```

IAG

```
 (-.763846070473675,.494732219202146)
 (-.763846070473675,-.0)
```

IMI

```
 (.763846070473675,-.367859450028737)
```

Zslot

```
 134.5380994413673
```

Zstrip

```
 51.90954610229248
```

SPout

```
 S11 =      .4535 AT  -12.8657 DEGREES
 S21 =      .5669 AT   10.2609 DEGREES
 S33 =      .4054 AT  171.3822 DEGREES
 S43 =      .6022 AT    5.7894 DEGREES
 S41 =      .5045 AT -151.6085 DEGREES
 U1 =  1.036144604777592 U2 =  1.036100141073986
 CU3 =  (-1.73596455651619E-002,6.82313522315672E-002)
```

D.2 Program DEPOL

PROGRAM DEPOL

```
************************************************************************
*                                                                      *
*     This program computes the location of the zero of the function   *
*     De (D sub e) using Newton's method.  The value of gamma squared  *
*     corresponds to the effective dielectric constant of the          *
*     dominant surface wave mode of the grounded dielectric slab.      *
*                                                                      *
*     The inverse of this function shows up in most of the integrals   *
*     computed.  This results in a simple pole through which the       *
*     integration cannot be performed numerically.  The pole has to    *
*     be computed accurate to at least seven or eight significant      *
*     figures.  This eliminates numerical errors when the integration  *
*     is performed a small distance up to and away from the pole.      *
*                                                                      *
************************************************************************


      PROGRAM     DEPOL
      IMPLICIT    COMPLEX*16 (C)
      IMPLICIT    REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16  U0, U1, KDU1, DE, DM
      REAL*8      K0, MUR
      COMMON      /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON      /L2/ U0, U1, KDU1, DE, DM, CCH, CSH
      COMMON      /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX

      OPEN (1, FILE = '../defined.data/phys.param')
      OPEN (8, FILE = '../computed.data/GAMA0')

      READ (1,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      K0  = 2.*PI*FREQ/SPEEDC
      WMK = WM*K0/2.
      WAK = WA*K0/2.

      Y = 1.001

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     DEPR is the derivative of DE.  Y is set close to 1 because      *
*     this is the rough location of the surface wave pole.  It        *
*     cannot be less than 1 and is smaller than the corresponding     *
*     value for microstrip effective dielectric constant.             *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      DO 5 J = 1, 100
         Y2    = Y**2
         XC2   = 1.
         XS2   = 1.
         CDUMY = CG(Y2)
         RDE   = DREAL(DE)
         DEPR  = DREAL(Y*((K0*D+EPSR/U0)*CCH
     &                  +(1.+EPSR*K0*D*U0)*CSH/U1))
```

```
      GAMA0 = Y - RDE/DEPR

      IF (DABS(RDE) .LT. 1.E-10) THEN
         GO TO 7
      ELSE IF (GAMA0 .LT. 1.) THEN
         Y  = (Y + GAMA0)/2.
      ELSE
         Y  = GAMA0
      END IF
    5 CONTINUE

      PRINT *, 'ROOT NOT FOUND IN 100 STEPS', GAMA0
      STOP

    7 CONTINUE

      WRITE (6,*) 'DE POLE AT GAMMA =', GAMA0
      WRITE (6,*) ' '
      WRITE (8,80) GAMA0
      CLOSE (1)
      CLOSE (8)
   80 FORMAT (1X, 1E20.14)
   99 STOP
      END

*************************************************************************
*                                                                       *
*     The following function computes variables needed for locating     *
*     the DE pole.                                                      *
*                                                                       *
*************************************************************************

      COMPLEX*16  FUNCTION CG(Y2)
      IMPLICIT    COMPLEX*16 (C)
      IMPLICIT    REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16  U0, U1, KDU1, DE, DM
      REAL*8      K0, MUR
      COMMON      /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON      /L2/ U0, U1, KDU1, DE, DM, CCH, CSH
      COMMON      /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX

      U0   = ZSQRT(DCMPLX(Y2 - 1., 0.))
      U1   = ZSQRT(DCMPLX(Y2 - EPSR*MUR, 0.))
      KDU1 = K0*D*U1

      IF (DREAL(KDU1) .GT. 600.) THEN
         CG  = (0., 0.)
      ELSE
         CEP = ZEXP(KDU1)
         CEN = ZEXP(-KDU1)
         CCH = (CEP + CEN)/2.
         CSH = (CEP - CEN)/2.
         DE  = U1*CSH + EPSR*U0*CCH
         DM  = U1*CCH + MUR*U0*CSH
         CG  = (EPSR*U0*XC2/DE + U1*XS2/DM)
      END IF
      RETURN
      END
```

D.3 Program TCAA

PROGRAM TCAA

```
************************************************************************
*                                                                      *
*     TCAA computes a cutoff value for the outer theta integration     *
*     of the integral computed in PIAA.  The function that results     *
*     when the inner integration over gamma is performed (that is,     *
*     f(theta) where f is obtained by integrating out gamma) has a     *
*     one over square root type singularity at theta = 0.              *
*                                                                      *
*     To allow accurate evaluation of the outer integral, two          *
*     different integration techniques have to be implemented.         *
*     One of these is applied from theta = 0 to theta = THETAC.        *
*     The other is applied from theta = THETAC to theta = PI2.         *
*                                                                      *
************************************************************************


      PROGRAM     TCAA
      IMPLICIT    REAL*8      (A, B, D - H, O - Z)
      COMPLEX*16  CFAA, CI
      REAL*8      KO, MUR
      EXTERNAL    CFAA, YI
      COMMON      /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA, DK
      COMMON      /B/  FLAGY, BADY, EYEST, NOFUN
      COMMON      /E/  PI, PI2, CI
      COMMON      /G/  G, G2
      COMMON      /AG/ AG

      OPEN (3, FILE = '../computed.data/GAMA0')
      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (8, FILE = '../computed.data/THETAC.IAA')

      READ (3,*) GAMA0
      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA

      PI  = 3.14159265358979
      PI2 = 1.57079632679489
      G   = 10.
      G2  = G**2

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*     G is an arbitrary constant which shows up when                   *
*     extracting the dominant asymptotic behavior of                   *
*     IAA.  G = 10 turns out to be a good value.                       *
*                                                                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)
      KO     = 2.*PI*FREQ/SPEEDC
      DK     = KO*D
      WAK    = KO*WA/2.
      AG     = GAMA0 + .1
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     The primary contribution to the inner gamma integral occurs             *
*     beyond the pole of the function 1/DE.  Thus AG is defined as            *
*     above.  The inner integral is calculated from AG to infinity.           *
*                                                                             *
*     The algorithm for locating THETAC is based on empirical                 *
*     observation of the function which results when the gamma                *
*     integration has been performed.  The real part of this function         *
*     has a 1/SQRT(theta) singularity at theta = 0 and thus goes              *
*     to infinity.  It then drops off to a minimum, taking on a               *
*     negative value in the process.  As theta approaches PI/2,               *
*     the function approaches zero.  This algorithm brackets the              *
*     minimum based on these observations.  This minimum has to               *
*     be located accurate to one or two figures so that the outer             *
*     integration routines will work correctly.                               *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      X = 1.D-3

   10 FX = YI(CFAA, X)
      IF (FX .LT. 0.) THEN
         X = X/5.
         GO TO 10
      END IF

      AX  = X
      CX  = 1.5
      BX  = (CX + 1.618*AX)/2.618
      FTC = GOLDEN(AX, BX, CX, YI, 1.D-1, THETAC)

   30 WRITE (6,*) 'THETA CUTOFF FOR IAA =', THETAC
      WRITE (8,*) THETAC

      CLOSE (3)
      CLOSE (7)
      CLOSE (8)
   99 STOP
      END

*******************************************************************************

      REAL*8 FUNCTION GOLDEN(AX,BX,CX,F,TOL,XMIN)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     This routine is from the book "Numerical Recipes" by                    *
*     W. Press, B. Flannery, S. Teukolsky, and W. Vetterling,                 *
*     Cambridge University Press, 1986, p. 282.  It performs                  *
*     a golden section search on the function F for the minimum               *
*     which is bracketed by AX, BX, and CX.                                   *
*     (F(BX) less than both F(AX) and F(CX).)                                 *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 CFAA
      EXTERNAL   CFAA, F
      PARAMETER  (R=.61803399, C= 1.-R)

      X0 = AX
      X3 = CX
```

```
      IF (DABS(CX-BX) .GT. DABS(BX-AX)) THEN
         X1 = BX
         X2 = BX+C*(CX-BX)
      ELSE
         X2 = BX
         X1 = BX - C*(BX-AX)
      END IF

      F1 = F(CFAA, X1)
      F2 = F(CFAA, X2)

    1 IF (DABS(X3-X0) .GT. TOL*(DABS(X1)+DABS(X2))) THEN
         IF (F2 .LT. F1) THEN
            X0 = X1
            X1 = X2
            X2 = R*X1 + C*X3
            F0 = F1
            F1 = F2
            F2 = F(CFAA, X2)
         ELSE
            X3 = X2
            X2 = X1
            X1 = R*X2 + C*X0
            F3 = F2
            F2 = F1
            F1 = F(CFAA, X1)
         END IF
      GO TO 1
      END IF

      IF (F1 .LT. F2) THEN
         GOLDEN = F1
         XMIN   = X1
      ELSE
         GOLDEN = F2
         XMIN   = X2
      END IF
      RETURN
      END

*************************************************************************

      COMPLEX*16 FUNCTION CFAA(Y)
      IMPLICIT   COMPLEX*16 (C)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      REAL*8     K0, MUR
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP

      Y2 = Y**2
      CALL FG(Y2, CFGY)
      CALL J0(Y*XC*WAK, RJ0)
      CFAA = RJ0**2*Y*((CFGY-CFGB+(B2-Y2)*CFGP)/(Y2-B2)**2+
     &       CFGP/(Y2+G2/XS))
      RETURN
      END

*************************************************************************

%     Insert SUBROUTINE FG(Y2, CFG) from program PIAA here!
```

```
*******************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

*******************************************************************************

%     Insert SUBROUTINE CCSI(C, CCI, CSI) from program PIAA here!

*******************************************************************************

%     Insert SUBROUTINE SCIS(X, RSI, RCI) from program PIAA here!

*******************************************************************************

%     Insert SUBROUTINE SCIL(X, RSI, RCI) from program PIAA here!

*******************************************************************************

%     Insert SUBROUTINE SCI(X, RSI, RCI) from program PIAA here!

*******************************************************************************

      REAL*8 FUNCTION YI(CF, X)
      IMPLICIT   COMPLEX*16 (C)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 UO, UO2, U1, U12, KDU1, DE, DM, DEP, DMP
      REAL*8     C1, C3, CBP, KO, MUR
      EXTERNAL   CF
      COMMON /A/  EPSR, MUR, KO, D, WNK, EPSM, WAK, EPSA, DK
      COMMON /B/  FLAGY, BADY, EYEST, NOFUN
      COMMON /E/  PI, PI2, CI
      COMMON /G/  G, G2
      COMMON /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON /L2/ UO, UO2, U1, U12, KDU1, DE, DM, CCH, CSH
      COMMON /AG/ AG
      PARAMETER   (LEVMY = 60)
      COMPLEX*16  AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION   Y(17), CY(17), CYSV(8,LEVMY), YSV(8,LEVMY)

      AY  = AG
      XS  = DSIN(X)
      XS2 = XS**2
      XC  = DCOS(X)
      XC2 = XC**2
      B   = DSQRT(EPSA)/XS
      B2  = B**2
      CALL  FG(B2, CFGB)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CFGP = .5*(XC2*(1./(UO*UO2) + EPSR/(U1*U12))/XS2 + 1./UO
     &         + 1./(MUR*U1))/XS2
      ELSE
         DEP  = .5*(CCH*(DK+EPSR/UO)+CSH*(1.+EPSR*DK*UO)/U1)
         DMP  = .5*(CSH*(DK+MUR/UO) +CCH*(1.+ MUR*DK*UO)/U1)
         CFGP = (-DK*CCH*CFGB/(2.*U1) + ((DMP*(1.+U12/DM**2)-1./DM)
     &         /MUR + XC2*(EPSR**2*(.5 - UO2/(2.*U12) -
     &         UO2*DEP/DE)/DE + DE * (1./UO2+1./U12)/2. - DEP)
     &         /(UO*U1*XS2))/XS2)/CSH
      END IF
```

```
      C4 = -CFGP
      C3 = (1.+1./MUR)/XS2
      C2 = B2*CFGP - CFGB
      C1 = -XC2*(1.+EPSR)/XS2**2

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     Note that C1 and C3 are REAL.                                   *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      A  = WAK*XC
      GL = 100./A
      IF (GL .LE. B) GL = 1.1*B
      GL2 = GL**2
      BY  = GL

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     In order to prevent undersampling the function, a               *
*     minimum number of function evaluations will be                  *
*     required.  MINFUN as calculated below will force                *
*     the adaptive routine to zoom in to a panel                      *
*     of approximate width 10.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      MINFUN = 8*IDINT(DNINT(DLOG(BY)/ALOG(2.)))

      W0 =   3956./14175.
      W1 =  23552./14175.
      W2 =  -3712./14175.
      W3 =  41984./14175.
      W4 = -18160./14175.

      RELERY = 1.D-5
      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AY
      Y(17)  = BY
      STONEY = (BY-AY)/16.
      Y(9)   = (Y(1) + Y(17))/2.
      Y(5)   = (Y(1) + Y(9)) /2.
      Y(13)  = (Y(9) + Y(17))/2.
      Y(3)   = (Y(1) + Y(5))/2.
      Y(7)   = (Y(5) + Y(9))/2.
      Y(11)  = (Y(9) + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
   25    CY(J) = CF(Y(J))

      FLAGY = 0.
      EYEST = 0.
      NOFUN = 9
      APREV = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                W4*CY(9))
      AREA  = APREV

   30 DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
   35    CY(J) = CF(Y(J))
```

```
      NOFUN  = NOFUN + 8
      STEPY  = (Y(17) - Y(1))/16.

      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &          W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &          W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &               W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &               W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERY*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUN  .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70

   40 DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
   50    YSV(J,LEVY)  = Y(J+9)

      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
   55    Y(19-2*J)  = Y(10-J)
      GO TO 30

   60 FLAGY = FLAGY + 1.
      BADY  = Y(9)

   70 CAREA = CAREA + ANOW + ADIFW
      EYEST = EYEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
   78    Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

   80 CONTINUE

      BP = 2.*A*B
      GP = 2.*A*GL
      H  = GP + BP
      HT = GP - BP
      IF (H .GE. 100.) THEN
         CALL SCIL(H, HSI, HCI)
      ELSE
         CALL SCI(H, HSI, HCI)
      END IF
```

```
      IF (HT .GE. 100.) THEN
         CALL SCIL(HT, HTSI, HTCI)
      ELSE IF (HT .LE. 1.) THEN
         CALL SCIS(HT, HTSI, HTCI)
      ELSE
         CALL SCI(HT, HTSI, HTCI)
      END IF

      E   = HTSI + HSI
      ET  = HTSI - HSI
      F   = HTCI + HCI
      FT  = HTCI - HCI
      SGP = DSIN(GP)
      CBP = DCOS(BP)
      SBP = DSIN(BP)
      GB  = .5/(GL2 - B2)

      IF (GP .GE. 100.) THEN
         CALL SCIL(GP, GSI, GCI)
      ELSE
         CALL SCI(GP, GSI, GCI)
      END IF

      D1 = (1.+SGP)*GB/B2 + (((CBP+.5*BP*SBP)*E+(SBP-.5*BP*CBP)*FT
     &     - DLOG(2.*GL2*GB))/2. - GSI)/B2**2
      D2 = GL*GB*(1.+SGP)/B2 + ((CBP+BP*SBP)*ET + (SBP-BP*CBP)*F +
     &     DLOG(HT/H))/(4.*B*B2)
      D3 = (1.+SGP)*GB + .5*A*(SBP*E - CBP*FT)/B
      D4 = (1.+SGP)*GL*GB - ((CBP-BP*SBP)*ET + (SBP+BP*CBP)*F +
     &     DLOG(HT/H))/(4.*B)

      CASINT = (C1*D1 + C2*D2 + C3*D3 + C4*D4)/(PI*A)
      GC     = G/DSQRT(XS)
      GCP    = 2.*A*GC

      IF (GCP .GT. 12.) THEN
         TRM = 0.
      ELSE
         CN  = DCMPLX(GP,GCP)
         CALL  CCSI(CN, CCI, CSI)
         RCI = DREAL(CCI)
         RSI = DIMAG(CSI)
         TRM = DSINH(GCP)*RCI - DCOSH(GCP)*RSI
      END IF

      CASMP = C4*(DATAN(GL/GC) - PI2 + TRM)/(GCP*PI2)
      YI    = DREAL(CAREA + CASINT + CASMP)
      RETURN
      END
```

D.4 Program TCMM

PROGRAM TCMM

```
*************************************************************************
*                                                                       *
*     TCMM computes a cutoff value for the outer theta integration      *
*     of the integral computed in PIMM.  The function that results      *
*     when the inner integration over gamma is performed (that is,      *
*     f(theta) where f is obtained by integrating out gamma) has a      *
*     one over square root type singularity at theta = PI/2.            *
*                                                                       *
*     To allow accurate evaluation of the outer integral, two           *
*     different integration techniques have to be implemented.          *
*     One of these is applied from theta = 0 to theta = THETAC.         *
*     The other, which was developed to deal with functions             *
*     containing 1/SQRT(x-x0) singularities, is applied from            *
*     theta = THETAC to theta = PI2.                                    *
*                                                                       *
*************************************************************************


      PROGRAM    TCMM
      IMPLICIT   COMPLEX*16 (C)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      REAL*8     K0, MUR
      EXTERNAL   CFMM
      COMMON     /A/   EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/   FLAGY, BADY, EYEST, NOFUN
      COMMON     /C/   W0, W1, W2, W3, W4
      COMMON     /D/   RELERY
      COMMON     /E/   PI, PI2, CI
      COMMON     /G/   G, G2
      COMMON     /Z/   AI, BI, AO, BO

      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (8, FILE = '../computed.data/THETAC.IMM')

      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)
      G      = 10.
      G2     = G**2

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     G is an arbitrary constant which shows up when                    *
*     extracting the dominant asymptotic behavior of                    *
*     IMM.  G = 10 turns out to be a good value.                        *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      W0     =  3956./14175.
      W1     = 23552./14175.
      W2     = -3712./14175.
      W3     = 41984./14175.
      W4     = -18160./14175.
```

```
      K0     = 2.*PI*FREQ/SPEEDC
      WMK    = K0*WM/2.
      AT     = .01
      BT     = 1.57
      AI     = EPSR*MUR + 1.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     The primary contribution to the inner integral occurs beyond        *
*     the branch points of U0 and U1.  Thus, to speed up this             *
*     program, the gamma integration begins at EPSR*MUR+1.                *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      RELERY = 1.D-6
      F      = 3.*ZABS(CYI(CFMM, 1.D-1))

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     The algorithm used for calculating THETAC is based on empirical     *
*     observation of the behavior of IMM when the gamma integration       *
*     has been performed.  The function has a small slope and value       *
*     until it approaches theta = PI/2 at which time the function         *
*     and its slope increase rapidly.  A suitable cutoff for              *
*     theta occurs when f(theta) has grown to three times its             *
*     starting value at f(0.1).                                           *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      DO 20 I = 1,15
         X  = PI2 - .1*I
         FX = ZABS(CYI(CFMM, X))
         IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
         IF (FX .LT. F) THEN
            THETAC = X
            GO TO 30
         END IF
   20    CONTINUE

      PRINT *, 'THETA CUTOFF NOT LOCATED - POTENTIAL ERROR IN IMM'
      THETAC = .1
   30 WRITE (6,*) 'THETA CUTOFF FOR IMM =', THETAC
      WRITE (8,*) THETAC

      CLOSE (7)
      CLOSE (8)
   99 STOP
      END

*************************************************************************

      COMPLEX*16 FUNCTION CFMM(Y)
      IMPLICIT   COMPLEX*16 (C)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      REAL*8     K0, MUR
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CF1B, CF2B, CF1P, CF2P, CFUN
      COMMON     /Z/  AI, BI, AO, BO
```

```
      Y2   = Y**2
      CALL   F1F2(Y2, CF1Y, CF2Y)
      CT1  = XS2*(CF1Y - CF1B + (B2-Y2)*CF1P)
      CT2  = XC2*(CF2Y - CF2B + (B2-Y2)*CF2P)
      CALL   JO(Y*XS*WMK, RJO)
      CFMM = RJO**2*Y*((CT1+CT2)/(Y2*XC2-EPSM)**2 - CFUN/(Y2+G2/XC))

      RETURN
      END

*************************************************************************

%     Insert SUBROUTINE F1F2(Y2, CF1, CF2) from program PIMM here!

*************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE CCSI(C, CCI, CSI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCIS(X, RSI, RCI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCIL(X, RSI, RCI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCI(X, RSI, RCI) from program PIAA here!

*************************************************************************

      COMPLEX*16 FUNCTION CYI(CFMM,X)

*     Computes inner gamma integral

      IMPLICIT   COMPLEX*16 (C)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      REAL*8     C1, C3, CBP, KO, MUR
      COMPLEX*16 UO, U1, KDU1, DE, DM
      EXTERNAL   CFMM
      COMMON     /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/  FLAGY, BADY, EYEST, NOFUN
      COMMON     /C/  WO, W1, W2, W3, W4
      COMMON     /D/  RELERY
      COMMON     /E/  PI, PI2, CI
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CF1B, CF2B, CF1P, CF2P, CFUN
      COMMON     /L2/ UO, U1, KDU1, DE, DM, CCH, CSH
      COMMON     /Z/  AI, BI, AO, BO
      PARAMETER  (LEVMY = 60)
      COMPLEX*16 AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION  Y(17), CY(17), YSV(8, LEVMY), CYSV(8, LEVMY)

      XS    = DSIN(X)
      XS2   = XS**2
      XC    = DCOS(X)
      XC2   = XC**2
      B     = DSQRT(EPSM)/XC
      B2    = B**2
```

```
      CALL    F1F2(B2, CF1B, CF2B)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CF1P = MUR*(1.+MUR*U1/U0)/(2.*U1*DM**2)
         CF2P = .5*(EPSR*U0**2/U1+U1**2/U0)/DE**2
      ELSE
         CF1P = MUR*(CSH*(CCH+MUR*U1*CSH/U0)-KDU1)/(2.*U1*DM**2)
         CF2P = .5*(CSH*(EPSR*U0**2*CCH/U1+U1**2*CSH/U0)
     &          + EPSR*K0*D*U0**2)/DE**2
      END IF
      T    = XS2/XC2
      C4   = -T*CF1P - CF2P
      C3   = 1./(1.+EPSR)
      C2   = -B2*C4 - T*CF1B - CF2B
      C1   = -MUR*T/(1.+MUR)
      CFUN = C4/XC2
      A    = WMK*XS
      GL   = 100./A
      IF     (GL .LE. B) GL = 1.1*B
      GL2  = GL**2
      BI   = GL

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     In order to prevent undersampling the function, a                   *
*     minimum number of function evaluations will be                      *
*     required.  MINFUN as calculated below will force                    *
*     the adaptive routine to zoom in to a panel                          *
*     of approximate width 10.                                            *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      MINFUN = 8*IDINT(DNINT(DLOG(BI)/ALOG(2.)))

      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AI
      Y(17)  = BI
      STONEY = (BI-AI)/16.
      Y(9)   = (Y(1) + Y(17))/2.
      Y(5)   = (Y(1) + Y(9)) /2.
      Y(13)  = (Y(9) + Y(17))/2.
      Y(3)   = (Y(1) + Y(5))/2.
      Y(7)   = (Y(5) + Y(9))/2.
      Y(11)  = (Y(9) + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
   25    CY(J) = CFMM(Y(J))

      FLAGY = 0.
      EYEST = 0.
      NOFUN = 9
      APREV = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                W4*CY(9))
      AREA  = APREV

   30 DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
   35    CY(J) = CFMM(Y(J))

      NOFUN  = NOFUN + 8
      STEPY  = (Y(17) - Y(1))/16.
```

```
      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &          W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &          W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &                W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &                W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERY*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUN  .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70

   40 DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
   50    YSV(J,LEVY)  = Y(J+9)

      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
   55    Y(19-2*J)  = Y(10-J)
      GO TO 30

   60 FLAGY = FLAGY + 1.
      BADY  = Y(9)

   70 CAREA = CAREA + ANOW + ADIFW
      EYEST = EYEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
   78    Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

   80 CONTINUE

      BP = 2.*A*B
      GP = 2.*A*GL
      H  = GP + BP
      HT = GP - BP
      IF (H .GE. 100.) THEN
         CALL SCIL(H, HSI, HCI)
      ELSE
         CALL SCI(H, HSI, HCI)
      END IF
```

```
      IF (HT .GE. 100.) THEN
         CALL SCIL(HT, HTSI, HTCI)
      ELSE IF (HT .LE. 1.) THEN
         CALL SCIS(HT, HTSI, HTCI)
      ELSE
         CALL SCI(HT, HTSI, HTCI)
      END IF
      E   = HTSI + HSI
      ET  = HTSI - HSI
      F   = HTCI + HCI
      FT  = HTCI - HCI
      SGP = DSIN(GP)
      CBP = DCOS(BP)
      SBP = DSIN(BP)
      GB  = .5/(GL2 - B2)
      IF (GP .GE. 100.) THEN
         CALL SCIL(GP, GSI, GCI)
      ELSE
         CALL SCI(GP, GSI, GCI)
      END IF
      D1 = (1.+SGP)*GB/B2 + (((CBP+.5*BP*SBP)*E+(SBP-.5*BP*CBP)*FT
     &                     - DLOG(2.*GL2*GB))/2. - GSI)/B2**2
      D2 = GL*GB*(1.+SGP)/B2 + ((CBP+BP*SBP)*ET + (SBP-BP*CBP)*F +
     &                           DLOG(HT/H))/(4.*B*B2)
      D3 = (1.+SGP)*GB + .5*A*(SBP*E - CBP*FT)/B
      D4 = (1.+SGP)*GL*GB - ((CBP-BP*SBP)*ET + (SBP+BP*CBP)*F +
     &                       DLOG(HT/H))/(4.*B)
      CASINT = (C1*D1 + C2*D2 + C3*D3 + C4*D4)/(PI*A*XC2)
      GC     = G/DSQRT(XC)
      GCP    = 2.*A*GC

      IF (GCP .GT. 12.) THEN
         TRM = 0.
      ELSE
         CN  = DCMPLX(GP,GCP)
         CALL  CCSI(CN, CCI, CSI)
         RCI = DREAL(CCI)
         RSI = DIMAG(CSI)
         TRM = DSINH(GCP)*RCI - DCOSH(GCP)*RSI
      END IF
      CASMP  = CFUN*(DATAN(GL/GC) - PI2 + TRM)/(GCP*PI2)
      CYI    = CAREA + CASINT + CASMP

      RETURN
      END
```

D.5 Program PIAA

PROGRAM PIAA

```
************************************************************************
*                                                                      *
*     This program calculates the integral IAA which represents        *
*     the inner product between scattered electric field with          *
*     itself in the junction geometry.                                 *
*                                                                      *
************************************************************************


      PROGRAM    PIAA
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 DPOLE, DPOLEA, ASTRM, ASTRMA, RES,
     &           U0, U02, U1, U12, KDU1, DE, DM
      REAL*8     K0, KD, MUR, MURI, NA
      EXTERNAL   CFPOL, CFRES, CFASMP
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /B/  FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/  W0, W1, W2, W3, W4
      COMMON     /D/  RELERY
      COMMON     /E/  PI, PI2, CI
      COMMON     /F/  GAMA0
      COMMON     /G/  G, G2
      COMMON     /S/  FLAGS, BADS, ESEST, NOFUNS
      COMMON     /Z/  KEY, AI, BI, A0, B0
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /L2/ U0, U02, U1, U12, KDU1, DE, DM, CCH, CSH
      DIMENSION  AT(2), BT(2), CV(2)

      OPEN (3, FILE = '../computed.data/GAMA0')
      OPEN (4, FILE = '../computed.data/THETAC.IAA')
      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (9, FILE = '../defined.data/program.param')
      OPEN (8, FILE = '../computed.data/IAA')

      READ (3,*) GAMA0
      READ (4,*) BT(1)
      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (9,*) AT(1), BT(2), AG, DMY1, DMY2, MS, DELTA, NG, MG, DLTA2

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      G      = 10.
      G2     = G**2
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      W0 =  3956./14175.
      W1 = 23552./14175.
      W2 = -3712./14175.
      W3 = 41984./14175.
      W4 = -18160./14175.

      AT(2) = BT(1)
      K0    = 2.*PI*FREQ/SPEEDC
      KD    = K0*DSQRT(EPSR)*D
```

```
      IF (KD .LE. 0.05) THEN
         RELERY = 1.D-6
      ELSE
         RELERY = 1.D-7
      END IF

      DK    = KO*D
      WAK   = KO*WA/2.
      NA    = DSQRT(EPSA)
      Q     = NA*KO*WA
      SMALL = DATAN(Q/100.)

      WRITE (6,*) ' '
      WRITE (6,*) 'SMALL =', SMALL

      DPOLE  = CSINT(CFPOL, SMALL, BT(2), 1.D-7)
      SMLI   = Q/SMALL
      EPSI   = (1.+EPSR)/(EPSA*NA)
      MURI   = (1.+1./MUR)/NA
      S      = DSIN(SMLI)
      T      = DCOS(SMLI)
      DPOLEA = (-CI*EPSI + (EPSI-MURI)*(T-CI*S)/2.)/SMLI
      DPOLE  = DPOLE + DPOLEA

      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
      WRITE (6,*) 'DPOLE =', DPOLE, NOFUNS

      ASTRM  = CSINT(CFASMP, SMALL, BT(2), 1.D-7)
      ASTRMA = -DSQRT(SMALL)*(MURI+EPSI)/(KO*WA*G)
      ASTRM  = ASTRM + ASTRMA

      WRITE (6,*) 'ASTRM =', ASTRM, NOFUNS
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS

      XS2  = 1.
      XC2  = 1.
      RES  = CSINT(CFRES, AT(1), BT(2), 1.D-7)
      GO2  = GAMAO**2
      CALL   FG(GO2, CDUMY)
      DEPR = DREAL(GAMAO*((KO*D+EPSR/UO)*CCH+(1.+EPSR*KO*D*UO)*CSH/U1))
      RES  = -RES*CI*PI*UO*GAMAO*EPSR**2/(DEPR*U1*CSH)

      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
      WRITE (6,*) 'RESIDUE =', RES, NOFUNS

*  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     DLTA2 and DELTA need to be small enough so that the numerical   *
*     integration is correct around the 1/UO singularity as well as   *
*     the adjacent 1/DE pole.                                         *
*                                                                     *
*  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

   10 IF (DLTA2 .GT. (GAMAO-1.)/10.) THEN
         DLTA2 = DLTA2/10.
         GO TO 10
      END IF

   15 IF (DELTA .GT. (GAMAO-1.-DLTA2)/10.) THEN
         DELTA = DELTA/10.
         GO TO 15
      END IF
```

```
      DO 95 L = 1,2

         AO  = AT(L)
         BO  = BT(L)
         AI  = AG
         BI  = 1. - DLTA2
         KEY = 1
         CV1 = CGAUSS(NG,MG)

         WRITE (6,*) ' '
         IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
         WRITE (6,*) 'CV1(AG,1-DLTA2)', CV1, NOFUNY

         AI  = 1. - DLTA2
         BI  = 1.
         KEY = 3
         CV2 = CGAUSS(NG,MG)
         WRITE (6,*) 'CV2(1-DLTA2,1.)', CV2

         AI  = 1.
         BI  = 1. + DLTA2
         KEY = 4
         CV3 = CGAUSS(NG,MG)
         WRITE (6,*) 'CV3(1,1+DLTA2)', CV3

         AI  = 1. + DLTA2
         BI  = GAMA0 - DELTA
         KEY = 1
         CV4 = CGAUSS(NG,MG)

         IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
         WRITE (6,*) 'CV4(1+DLTA2,GAMA0-DELTA)', CV4, NOFUNY

         AI  = GAMA0 + DELTA
         KEY = 2

         IF (L .EQ. 1) THEN
            CV5 = CQSNG(MS)
         ELSE
            CV5 = CGAUSS(NG,MG)
         END IF

         IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
         WRITE (6,*) 'CV5(GAMA0+DELTA,GL)', CV5, NOFUNY

         CV(L) = CV1 + CV2 + CV3 + CV4 + CV5

      95 CONTINUE

         CVT = CV(1) + CV(2) + RES + DPOLE + ASTRM

         WRITE (6,*) 'CVT =', CVT

         CIAA = -4.*CVT*CI*EPSA/(ETA*PI**2)

         WRITE (6,*) ' '
         WRITE (6,*) 'IAA =', CIAA
         WRITE (8,*) CIAA

         CLOSE (3)
         CLOSE (4)
         CLOSE (7)
         CLOSE (8)
         CLOSE (9)
```

```
   99 STOP
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFAA(YA)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /Z/  KEY, AI, BI, AO, BO

      Y = YA
      IF (KEY .EQ. 1) Y = AI + BI - Y

      Y2   = Y**2
      CALL   FG(Y2, CFGY)
      CALL   JO(Y*XC*WAK, RJO)
      CFAA = RJO**2*Y*((CFGY-CFGB+(B2-Y2)*CFGP)/(Y2-B2)**2+
     &       CFGP/(Y2+G2/XS))
      RETURN
      END

*******************************************************************************

      SUBROUTINE FG(Y2, CFG)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 UO, UO2, U1, U12, KDU1, DE, DM
      REAL*8     KO, MUR
      COMMON     /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /L2/ UO, UO2, U1, U12, KDU1, DE, DM, CCH, CSH

      UO2  = DCMPLX(Y2 - 1., 0.)
      UO   = ZSQRT(UO2)
      U12  = DCMPLX(Y2 - EPSR*MUR, 0.)
      U1   = ZSQRT(U12)
      KDU1 = DK*U1

      IF (DREAL(KDU1) .GT. 10.) THEN
         DE  = U1 + EPSR*UO
         DM  = U1 + MUR*UO
         CFG = (DM/MUR-XC2*DE/(UO*U1*XS2))/XS2
      ELSE
         CEP = ZEXP(KDU1)
         CEN = ZEXP(-KDU1)
         CCH = (CEP+CEN)/2.
         CSH = (CEP-CEN)/2.
         DE  = U1*CSH + EPSR*UO*CCH
         DM  = U1*CCH + MUR*UO*CSH
         CFG = (XC2*(EPSR**2*UO/DE-DE/UO)/(U1*XS2)+(DM-U12/DM)/MUR)
     &         /(CSH*XS2)
      END IF
      RETURN
      END
```

```
*************************************************************************

      COMPLEX*16 FUNCTION CFPOL(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U02, U1, U12, KDU1, DE, DM, DEP, DMP
      REAL*8     KO, MUR
      COMMON     /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /E/  PI, PI2, CI
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /L2/ U0, U02, U1, U12, KDU1, DE, DM, CCH, CSH

      XS  = DSIN(X)
      XS2 = XS**2
      XC  = DCOS(X)
      XC2 = XC**2
      B2  = EPSA/XS2
      CALL  FG(B2, CFG)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CFGP = .5*(XC2*(1./(U0*U02) + EPSR/(U1*U12))/XS2 + 1./U0
     &          + 1./(MUR*U1))/XS2
      ELSE
         DEP  = .5*(CCH*(DK+EPSR/U0)+CSH*(1.+EPSR*DK*U0)/U1)
         DMP  = .5*(CSH*(DK+MUR/U0) +CCH*(1.+ MUR*DK*U0)/U1)
         CFGP = (-DK*CCH*CFG/(2.*U1) + ((DMP*(1.+U12/DM**2)-1./DM)/MUR
     &          + XC2*(EPSR**2*(.5 - U02/(2.*U12) - U02*DEP/DE)/DE + DE
     &          * (1./U02+1./U12)/2. - DEP)/(U0*U1*XS2))/XS2)/CSH
      END IF

      A     = XC*WAK
      B     = DSQRT(B2)
      AB    = A*B
      CALL    BES0 (AB, RJ0, RY0)
      CALL    BES1 (AB, RJ1, RY1)
      CIC   = PI*A*(RJ0*(RY1+2.*CI*RJ1) + RJ1*RY0)/(4.*B)
      CJC   = -PI2*RJ0*(RY0 + CI*RJ0)
      CFPOL = CIC*CFG + CJC*CFGP
      RETURN
      END

*************************************************************************

      COMPLEX*16 FUNCTION CFRES(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/ EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /F/ GAMA0

      XC    = DCOS(X)
      XC2   = XC**2
      XS2   = DSIN(X)**2
      CALL    J0(WAK*XC*GAMA0, RJ0)
      FCT   = XC2*(RJ0/(XS2*GAMA0**2-EPSA))**2
      CFRES = DCMPLX(FCT, 0.)
      RETURN
      END
```

```
*******************************************************************************

      COMPLEX*16 FUNCTION CFASMP(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U02, U1, U12, KDU1, DE, DM, DEP, DMP
      REAL*8     K0, MUR
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /L2/ U0, U02, U1, U12, KDU1, DE, DM, CCH, CSH

      XC  = DCOS(X)
      XC2 = XC**2
      XS  = DSIN(X)
      XS2 = XS**2
      SXS = DSQRT(XS)
      B2  = EPSA/XS2
      CALL  FG(B2, CFG)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CFGP = .5*(XC2*(1./(U0*U02) + EPSR/(U1*U12))/XS2 + 1./U0
     &         + 1./(MUR*U1))/XS2
      ELSE
         DEP  = .5*(CCH*(DK+EPSR/U0)+CSH*(1.+EPSR*DK*U0)/U1)
         DMP  = .5*(CSH*(DK+MUR/U0) +CCH*(1.+ MUR*DK*U0)/U1)
         CFGP = (-DK*CCH*CFG/(2.*U1) + ((DMP*(1.+U12/DM**2)-1./DM)/MUR
     &         + XC2*(EPSR**2*(.5 - U02/(2.*U12) - U02*DEP/DE)/DE + DE
     &         * (1./U02+1./U12)/2. - DEP)/(U0*U1*XS2))/XS2)/CSH
      END IF

      CALL     BESM(WAK*XC*G/SXS, RIK)
      CFASMP = - RIK*CFGP
      RETURN
      END

*******************************************************************************

      SUBROUTINE BESM(ARG, RIK)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     Routine returns RIK(ARG), the product of the modified                   *
*     Bessel functions I0(ARG) and K0(ARG).                                   *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)

      IF (ARG .LE. 3.75) THEN
         T   = (ARG/3.75)**2
         RIO = ((((.0045813*T+.0360768)*T+.2659732)*T+1.2067492)*T
     &             +3.0899424)*T+3.5156229)*T+1.
      ELSE
         T   = 3.75/ARG
         RIO = (((((((.00392377*T-.01647633)*T+.02635537)*T
     &         -.02057706)*T+.00916281)*T-.00157565)*T+.00225319)*T
     &         +.01328592)*T+.39894228
      END IF
```

```
      IF (ARG .LE. 2.) THEN
         T   = (ARG/2.)**2
         RK0 = (((((7.4D-6*T+1.075D-4)*T+.00262698)*T+.0348859)*T
     &          +.23069756)*T+.4227842)*T-.57721566-RI0*DLOG(ARG/2.)
         RIK = RI0*RK0
      ELSE IF (ARG .LE. 3.75) THEN
         T   = 2./ARG
         RK0 = (((((5.3208D-4*T-2.5154D-3)*T+5.87872D-3)*T
     &          -.01062446)*T+.02189568)*T-.07832358)*T+1.25331414
         RIK = RI0*RK0*DEXP(-ARG)/DSQRT(ARG)
      ELSE
         T   = 2./ARG
         RK0 = (((((5.3208D-4*T-2.5154D-3)*T+5.87872D-3)*T
     &          -.01062446)*T+.02189568)*T-.07832358)*T+1.25331414
         RIK = RI0*RK0/ARG
      END IF
      RETURN
      END

*****************************************************************************

      SUBROUTINE BES0(ARG, RJ0, RY0)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
*     Routine returns RJ0(ARG) and RY0(ARG), the zeroth order               *
*     Bessel and Neumann function, respectively.                            *
*                                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 CI
      COMMON     /E/ PI, PI2, CI

      IF (ARG .LE. 3.) THEN
         Y   = (ARG/3.)**2
         RJ0 =(((((0.21000000D-3 *Y-0.39444000D-2)*Y+0.44447900D-1)
     &          *Y-0.31638660E00)*Y+0.12656208D+1)*Y-0.22499997D+1)
     &          *Y+0.10000000D+1
         RY0 =(((((-.24846000D-3 *Y+0.42791600D-2)*Y-0.42612140D-1)
     &          *Y+0.25300117E00)*Y-0.74350384E00)*Y+0.60559366E00)
     &          *Y+0.36746691E00 + 2.*RJ0*DLOG(ARG/2.)/PI

      ELSE

         Y     = 3.0/ARG
         F0    = (((((0.14476000D-3 *Y-0.72805000D-3)*Y+0.13723700D-2)
     &            *Y-0.95120000D-4)*Y-0.55274000D-2)*Y-0.77000000D-6)
     &            *Y+0.79788456
         THETA = (((((0.13558000D-3 *Y-0.29333000D-3)*Y-0.54125000D-3)
     &            *Y+0.26257300D-2)*Y-0.39540000D-4)*Y-0.41663970D-1)
     &            *Y-0.78539816+ARG
         RJ0 = F0*DCOS(THETA)/DSQRT(ARG)
         RY0 = F0*DSIN(THETA)/DSQRT(ARG)
      END IF
      RETURN
      END
```

```
*************************************************************************

      SUBROUTINE BES1(ARG, RJ1, RY1)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     Routine returns RJ1(ARG) and RY1(ARG), the first order            *
*     Bessel and Neumann function, respectively.                        *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 CI
      COMMON     /E/ PI, PI2, CI

      IF (ARG .LE. 3.) THEN
         Y   = (ARG/3.)**2
         RJ1 = ((((((.00001109*Y-.00031761)*Y+.00443319)*Y
     &          -.03954289)*Y+.21093573)*Y-.56249985)*Y+.5)*ARG
         RY1 = ((((((.0027873*Y-.0400976)*Y+.3123951)*Y
     &          -1.3164827)*Y+2.1682709)*Y+.2212091)*Y-.6366198)/ARG
     &          +2.*RJ1*DLOG(ARG/2.)/PI

      ELSE

         Y     = 3.0/ARG
         F1    = (((((-.20033000D-3 *Y+0.11365300D-2)*Y-0.24951100D-2)
     &             *Y+0.17105000D-3)*Y+0.16596670D-1)*Y+0.15600000D-5)
     &             *Y+0.79788456
         THETA = (((((-.29166000D-3 *Y+0.79824000D-3)*Y+0.74348000D-3)
     &             *Y-0.63787900D-2)*Y+0.56500000D-4)*Y+0.12499612E00)
     &             *Y-2.35619449+ARG

         RJ1 = F1*DCOS(THETA)/DSQRT(ARG)
         RY1 = F1*DSIN(THETA)/DSQRT(ARG)
      END IF
      RETURN
      END

*************************************************************************

      SUBROUTINE JO(ARG, RJO)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     Subroutine returns the zeroth order Bessel function RJO for       *
*     real argument ARG.                                                *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT REAL*8 (A, B, D - H, O - Z)

      IF (ARG .LT. 0.) ARG = -ARG
```

```
      IF (ARG .LE. 3.) THEN
         Y   = (ARG/3.)**2
         RJ0 =(((((0.21000000D-3 *Y-0.39444000D-2)*Y+0.44447900D-1)
     &          *Y-0.31638660E00)*Y+0.12656208D+1)*Y-0.22499997D+1)
     &          *Y+0.10000000D+1
      ELSE
         Y     = 3.0/ARG
         F0    = (((((0.14476000D-3 *Y-0.72805000D-3)*Y+0.13723700D-2)
     &            *Y-0.95120000D-4)*Y-0.55274000D-2)*Y-0.77000000D-6)
     &            *Y+0.79788456
         THETA = (((((0.13558000D-3 *Y-0.29333000D-3)*Y-0.54125000D-3)
     &            *Y+0.26257300D-2)*Y-0.39540000D-4)*Y-0.41663970D-1)
     &            *Y-0.78539816+ARG
         RJ0   = F0*DCOS(THETA)/DSQRT(ARG)
      END IF
      RETURN
      END

*******************************************************************************

      SUBROUTINE CCSI(C, CCI, CSI)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     Complex sine and cosine integrals for ABS(C) .GE. 50.                   *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT COMPLEX*16 (C)

      C2  = C**2
      CF  = (1.-2./C2)/C
      CG  = (1.-6./C2)/C2
      CCI =  CF*ZSIN(C) - CG*ZCOS(C)
      CSI = -CF*ZCOS(C) - CG*ZSIN(C)

      RETURN
      END

*******************************************************************************

      SUBROUTINE SCIS(X, RSI, RCI)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     Real Sine and cosine integrals for X .LT. 1.                            *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


      IMPLICIT REAL*8 (A, B, D - H, O - Z)

      X2  = X**2
      PI2 = 1.57079632679489
      RSI = X*(((1./600.-X2/35280.)*X2-1./18.)*X2+1.)-PI2
      RCI = DLOG(X)+X2*(((X2/322560.-1./4320.)*X2+1./96.)*X2-1./4.)+
     &      .5772156649

      RETURN
      END
```

```
*******************************************************************************

      SUBROUTINE SCIL(X, RSI, RCI)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     Real sine and cosine integrals for X .GT. 50                            *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT REAL*8 (A, B, D - H, O - Z)

      X2  = X**2
      F   = (1.-2./X2)/X
      G   = (1.-6./X2)/X2
      RCI =  F*DSIN(X) - G*DCOS(X)
      RSI = -F*DCOS(X) - G*DSIN(X)

      RETURN
      END

*******************************************************************************

      SUBROUTINE SCI(X, RSI, RCI)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*     Real sine and cosine integrals for 1 .LE. X .LT. infinity               *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT REAL*8 (A, B, D - H, O - Z)
      REAL*8   A(4,4), H(4)

      X2 = X**2
      A(1,1) =  38.027264
      A(1,2) = 265.187033
      A(1,3) = 335.677320
      A(1,4) =  38.102495
      A(2,1) =  40.021433
      A(2,2) = 322.624911
      A(2,3) = 570.236280
      A(2,4) = 157.105423
      A(3,1) =  42.242855
      A(3,2) = 302.757865
      A(3,3) = 352.018498
      A(3,4) =  21.821899
      A(4,1) =  48.196927
      A(4,2) = 482.485984
      A(4,3) =1114.978885
      A(4,4) = 449.690326

      DO 20 I = 1, 4
         H(I) = X2**4

         DO 10 J = 1, 4
   10       H(I) = H(I) + A(I, 5-J)*X2**(J-1)

   20 CONTINUE

      F   = H(1)/(X*H(2))
      G   = H(3)/(X2*H(4))
      RSI = -F*COS(X) - G*SIN(X)
      RCI =  F*SIN(X) - G*COS(X)
```

```
      RETURN
      END

*****************************************************************************

      COMPLEX*16 FUNCTION CGAUSS(N,M)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
*     This routine performs a Gaussian quadrature integration on            *
*     the function CININT which is itself an integral.  CININT              *
*     corresponds to the inner integral of the external function            *
*     CFAA.  Some definitions:                                              *
*                                                                           *
*     AO   : lower limit of outer integral                                  *
*     BO   : upper limit of outer integral                                  *
*     N    : number of segments or subintervals                             *
*     M    : number of points evaluated in each subinterval                 *
*     CFAA : complex external function of a real variable                   *
*                                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      EXTERNAL   CFAA
      COMMON     /B/ FLAGY, BADY, EYEST, NOFUNY
      COMMON     /Z/ KEY, AI, BI, AO, BO
      INTEGER    NPOINT(5)
      REAL*8     Z(10), W(10)

      DATA       NPOINT /4,6,8,10,12/

      M2 = M/2

      DO 200 I = 1,5
         IF (M.EQ.NPOINT(I)) GO TO 205
  200 CONTINUE
      PRINT *, 'Wrong input value(s) for CGAUSS  -- Fatal error'
      STOP
  205 GO TO (4,6,8,10,12),I

    4 Z(1)=.339981043584856
      Z(2)=.861136311594053
      W(1)=.652145154862546
      W(2)=.347854845137454
      GO TO 100
    6 Z(1)=.238619186083197
      Z(2)=.661209386466265
      Z(3)=.932469514203152
      W(1)=.467913934572691
      W(2)=.360761573048139
      W(3)=.171324492379170
      GO TO 100
    8 Z(1)=.183434642495650
      Z(2)=.525532409916329
      Z(3)=.796666477413627
      Z(4)=.960289856497536
      W(1)=.362683783378362
      W(2)=.313706645877887
      W(3)=.222381034453374
      W(4)=.101228536290376
      GO TO 100
```

```
   10 Z(1)=.148874338981631
      Z(2)=.433395394129247
      Z(3)=.679409568299024
      Z(4)=.865063366688985
      Z(5)=.973906528517172
      W(1)=.295524224714753
      W(2)=.269266719309996
      W(3)=.219086362515982
      W(4)=.149451349150581
      W(5)=.066671344308688
      GO TO 100
   12 Z(1)=.125233408511469
      Z(2)=.367831498998180
      Z(3)=.587317954286617
      Z(4)=.769902674194305
      Z(5)=.904117256370475
      Z(6)=.981560634246719
      W(1)=.249147045813403
      W(2)=.233492536538355
      W(3)=.203167426723066
      W(4)=.160078328543346
      W(5)=.106939325995318
      W(6)=.047175336386512
      GO TO 100
  100 CONTINUE

      FLAGY  = 0.
      EYEST  = 0.
      NOFUNY = 0
      BAH    = (BO-AO)/2.
      BAHN   = BAH/N
      CSEG   = (0.0,0.0)

      DO 310 I = 1, N
         BAM  = AO+(2*I-1)*BAHN
         CSUM = (0.0,0.0)

         DO 300 J=1,M2
            CSUM = CSUM + W(J)*(CININT(CFAA, Z(J)*BAHN+BAM)+
     &                          CININT(CFAA,-Z(J)*BAHN+BAM))
  300       CONTINUE

  310    CSEG = CSEG + BAHN*CSUM

      CGAUSS = CSEG
      RETURN
      END
```

****

```
************************************************************************

      COMPLEX*16 FUNCTION CQSNG(M)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     This routine performs a Gaussian quadrature integration on a    *
*     complex function which has a 1/SQRT(x) singularity.             *
*                                                                     *
*     AO   : lower limit                                              *
*     BO   : upper limit                                              *
*     M    : number of points evaluated                               *
*     CFAA : complex external function of a real variable             *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     X(20), Z(20), W(20)
      EXTERNAL   CFAA
      COMMON     /B/ FLAGY, BADY, EYEST, NOFUNY
      COMMON     /E/ PI, PI2, CI
      COMMON     /Z/ KEY, AI, BI, AO, BO
      INTEGER    NPOINT(5)
      DATA       NPOINT /4,8,10,16,20/

      DO 200 I = 1,5
         IF (M .EQ. NPOINT(I)) GO TO 205
  200 CONTINUE
      PRINT *, 'Wrong input value for CQSNG  -- Fatal Error'
      STOP

  205 GO TO (4,8,10,16,20), I

    4 Z(1)=.183434642495650
      Z(2)=.525532409916329
      Z(3)=.796666477413627
      Z(4)=.960289856497536
      W(1)=.362683783378362
      W(2)=.313706645877887
      W(3)=.222381034453374
      W(4)=.101228536290376
      GO TO 100
    8 Z(1)=.095012509837637
      Z(2)=.281603550779259
      Z(3)=.458016777657227
      Z(4)=.617876244402644
      Z(5)=.755404408355003
      Z(6)=.865631202387832
      Z(7)=.944575023073233
      Z(8)=.989400934991650
      W(1)=.189450610455068
      W(2)=.182603415044924
      W(3)=.169156519395003
      W(4)=.149595988816577
      W(5)=.124628971255534
      W(6)=.095158511682493
      W(7)=.062253523938648
      W(8)=.027152459411754
      GO TO 100
```

```
10 Z(1)=.076526521133497
   Z(2)=.227785851141645
   Z(3)=.373706088715420
   Z(4)=.510867001950827
   Z(5)=.636053680726515
   Z(6)=.746331906460151
   Z(7)=.839116971822219
   Z(8)=.912234428251326
   Z(9)=.963971927277914
   Z(10)=.993128599185095
   W(1)=.152753387130726
   W(2)=.149172986472604
   W(3)=.142096109318382
   W(4)=.131688638449177
   W(5)=.118194531961518
   W(6)=.101930119817240
   W(7)=.083276741576705
   W(8)=.062672048334109
   W(9)=.040601429800387
   W(10)=.017614007139152
   GO TO 100
16 Z(1) = .048307665687738
   Z(2) = .144471961582796
   Z(3) = .239287362252137
   Z(4) = .331868602282127
   Z(5) = .421351276130635
   Z(6) = .506899908932229
   Z(7) = .587715757240762
   Z(8) = .663044266930215
   Z(9) = .732182118740289
   Z(10)= .794483795967942
   Z(11)= .849367613732569
   Z(12)= .896321155766052
   Z(13)= .934906075937739
   Z(14)= .964762255587506
   Z(15)= .985611511545268
   Z(16)= .997263861849481
   W(1) = .096540088514727
   W(2) = .095638720079274
   W(3) = .093844399080804
   W(4) = .091173878695763
   W(5) = .087652093004403
   W(6) = .083311924226946
   W(7) = .078193895787070
   W(8) = .072345794108848
   W(9) = .065822222776361
   W(10)= .058684093478535
   W(11)= .050998059262376
   W(12)= .042835898022226
   W(13)= .034273862913021
   W(14)= .025392065309262
   W(15)= .016274394730905
   W(16)= .007018610009470
   GO TO 100
20 Z(1) = .038772417506050
   Z(2) = .116084070675255
   Z(3) = .192697580701371
   Z(4) = .268152185007253
   Z(5) = .341994090825758
   Z(6) = .413779204371605
   Z(7) = .483075801686178
   Z(8) = .549467125095128
   Z(9) = .612553889667980
   Z(10)= .671956684614179
```

```
      Z(11)= .727318255189927
      Z(12)= .778305651426519
      Z(13)= .824612230833311
      Z(14)= .865959503212259
      Z(15)= .902098806968874
      Z(16)= .932812808278676
      Z(17)= .957916819213791
      Z(18)= .977259949983774
      Z(19)= .990726238699457
      Z(20)= .998237709710559
      W(1) = .077505947978424
      W(2) = .077039818164247
      W(3) = .076110361900626
      W(4) = .074723169057968
      W(5) = .072886582395804
      W(6) = .070611647391286
      W(7) = .067912045815233
      W(8) = .064804013456601
      W(9) = .061306242492928
      W(10)= .057439769099391
      W(11)= .053227846983936
      W(12)= .048695807635072
      W(13)= .043870908185673
      W(14)= .038782167974472
      W(15)= .033460195282547
      W(16)= .027937006980023
      W(17)= .022245849194166
      W(18)= .016421058381907
      W(19)= .010498284531152
      W(20)= .004521277098533

  100 CONTINUE

      FLAGY  = 0.
      EYEST  = 0.
      NOFUNY = 0

      DO 110 I = 1, M
         W(I)  = 2.*W(I)
  110    X(I)  = 1.- Z(I)**2

      BA   = BO - AO
      CSUM = (0.,0.)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     This routine is applicable when the singularity lies at the    *
*     upper integration limit.  Since for IAA it is at theta = 0,    *
*     the function has to rotated about the midpoint of the          *
*     integration interval, hence the argument BO-THETA+AO below.    *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      DO 300 J = 1, M
         THETA = AO + BA*X(J)
  300    CSUM  = CSUM + W(J)*CININT(CFAA,AO+BO-THETA)*DSQRT(BO-THETA)

      CQSNG = DSQRT(BA)*CSUM
      RETURN
      END
```

```
************************************************************************

      COMPLEX*16 FUNCTION CININT(CFAA,X)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     This routine calculates the inner gamma integral for the        *
*     double integral IAA.  The result is returned to the             *
*     quadrature rules, CGAUSS or CQSNG.                              *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     C1, C3, CBP, K0, MUR, Z(10), W(10)
      COMPLEX*16 U0, U02, U1, U12, KDU1, DE, DM, DEP, DMP, CCI, CSI
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA, DK
      COMMON     /B/  FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/  W0, W1, W2, W3, W4
      COMMON     /D/  RELERY
      COMMON     /E/  PI, PI2, CI
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CFGB, CFGP
      COMMON     /L2/ U0, U02, U1, U12, KDU1, DE, DM, CCH, CSH
      COMMON     /Z/  KEY, AI, BI, A0, B0
      PARAMETER  (LEVMY = 60)
      COMPLEX*16 AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION  Y(17), CY(17), CYSV(8,LEVMY), YSV(8,LEVMY)

      XS  = DSIN(X)
      XS2 = XS**2
      XC  = DCOS(X)
      XC2 = XC**2
      B   = DSQRT(EPSA)/XS
      B2  = B**2
      CALL  FG(B2, CFGB)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CFGP = .5*(XC2*(1./(U0*U02) + EPSR/(U1*U12))/XS2 + 1./U0
     &          + 1./(MUR*U1))/XS2
      ELSE
         DEP = .5*(CCH*(DK+EPSR/U0)+CSH*(1.+EPSR*DK*U0)/U1)
         DMP = .5*(CSH*(DK+MUR/U0) +CCH*(1.+ MUR*DK*U0)/U1)
         CFGP = (-DK*CCH*CFGB/(2.*U1) + ((DMP*(1.+U12/DM**2)-1./DM)/MUR
     &          + XC2*(EPSR**2*(.5 - U02/(2.*U12) - U02*DEP/DE)/DE + DE
     &          * (1./U02+1./U12)/2. - DEP)/(U0*U1*XS2))/XS2)/CSH
      END IF

      C4 = -CFGP
      C3 = (1.+1./MUR)/XS2
      C2 = B2*CFGP - CFGB
      C1 = -XC2*(1.+EPSR)/XS2**2
```

```
      IF (KEY .EQ. 2) THEN
         A   = WAK*XC
         GL  = 100./A
         IF (GL .LE. B) GL = 1.1*B
         GL2 = GL**2
         BI  = GL
      ELSE IF ((KEY .EQ. 3) .OR. (KEY .EQ. 4)) THEN
         Z(1)=.076526521133497
         Z(2)=.227785851141645
         Z(3)=.373706088715420
         Z(4)=.510867001950827
         Z(5)=.636053680726515
         Z(6)=.746331906460151
         Z(7)=.839116971822219
         Z(8)=.912234428251326
         Z(9)=.963971927277914
         Z(10)=.993128599185095
         W(1)=.152753387130726
         W(2)=.149172986472604
         W(3)=.142096109318382
         W(4)=.131688638449177
         W(5)=.118194531961518
         W(6)=.101930119817240
         W(7)=.083276741576705
         W(8)=.062672048334109
         W(9)=.040601429800387
         W(10)=.017614007139152

         DO 110 I = 1,10
            W(I) = 2.*W(I)
 110        Y(I) = 1.- Z(I)**2

         BA   = BI - AI
         CSUM = (0.,0.)

         DO 120 J = 1,10
            YY = AI + BA*Y(J)
            IF (KEY .EQ. 3) THEN
               CF = CFAA(YY)
            ELSE
               CF = CFAA(AI+BI-YY)
            END IF
 120        CSUM = CSUM + W(J)*CF*DSQRT(BI-YY)

         CININT = DSQRT(BA)*CSUM
         GO TO 99
      END IF

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     The above statements compute the inner gamma integral in           *
*     the immediate vicinity of the 1/U0 singularity.  Then              *
*     the routine is exited (line 99 is the RETURN).                     *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     In order to prevent undersampling the function, a               *
*     minimum number of function evaluations will be                  *
*     required.  MINFUN as calculated below will force                *
*     the adaptive routine to zoom in to a panel                      *
*     of approximate width 10.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      MINFUN = 8*IDINT(DNINT(DLOG(BI)/ALOG(2.)))

      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AI
      Y(17)  = BI
      STONEY = (BI-AI)/16.
      Y(9)   = (Y(1)  + Y(17))/2.
      Y(5)   = (Y(1)  + Y(9)) /2.
      Y(13)  = (Y(9)  + Y(17))/2.
      Y(3)   = (Y(1)  + Y(5))/2.
      Y(7)   = (Y(5)  + Y(9))/2.
      Y(11)  = (Y(9)  + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
   25    CY(J) = CFAA(Y(J))

      NOFUN = 9
      NOFUNY = NOFUNY + 9
      APREV = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                W4*CY(9))
      AREA  = APREV

   30 DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
   35    CY(J) = CFAA(Y(J))

      NOFUN  = NOFUN + 8
      NOFUNY = NOFUNY + 8
      STEPY  = (Y(17) - Y(1))/16.

      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &         W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &         W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &               W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &               W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERY*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUN  .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70
```

```
   40 DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
   50    YSV(J,LEVY)  = Y(J+9)

      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
   55    Y(19-2*J)  = Y(10-J)
      GO TO 30

   60 FLAGY = FLAGY + 1.
      BADY  = Y(9)

   70 CAREA = CAREA + ANOW + ADIFW
      EYEST = EYEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
   78    Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

   80 CONTINUE

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     The following lines compute the asymptotic portion of the           *
*     inner gamma integral from BI to infinity.                           *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IF (KEY .EQ. 2) THEN
         BP = 2.*A*B
         GP = 2.*A*GL
         H  = GP + BP
         HT = GP - BP

         IF (H .GE. 100.) THEN
            CALL SCIL(H, HSI, HCI)
         ELSE
            CALL SCI(H, HSI, HCI)
         END IF

         IF (HT .GE. 100.) THEN
            CALL SCIL(HT, HTSI, HTCI)
         ELSE IF (HT .LE. 1.) THEN
            CALL SCIS(HT, HTSI, HTCI)
         ELSE
            CALL SCI(HT, HTSI, HTCI)
         END IF

         E   = HTSI + HSI
         ET  = HTSI - HSI
         F   = HTCI + HCI
         FT  = HTCI - HCI
         SGP = DSIN(GP)
```

```
      CBP = DCOS(BP)
      SBP = DSIN(BP)
      GB  = .5/(GL2 - B2)

      IF (GP .GE. 100.) THEN
         CALL SCIL(GP, GSI, GCI)
      ELSE
         CALL SCI(GP, GSI, GCI)
      END IF

      D1 = (1.+SGP)*GB/B2 + (((CBP+.5*BP*SBP)*E+(SBP-.5*BP*CBP)*FT
     &       - DLOG(2.*GL2*GB))/2. - GSI)/B2**2
      D2 = GL*GB*(1.+SGP)/B2 + ((CBP+BP*SBP)*ET + (SBP-BP*CBP)*F +
     &      DLOG(HT/H))/(4.*B*B2)
      D3 = (1.+SGP)*GB + .5*A*(SBP*E - CBP*FT)/B
      D4 = (1.+SGP)*GL*GB - ((CBP-BP*SBP)*ET + (SBP+BP*CBP)*F +
     &      DLOG(HT/H))/(4.*B)
      CASINT = (C1*D1 + C2*D2 + C3*D3 + C4*D4)/(PI*A)
      GC = G/DSQRT(XS)
      GCP = 2.*A*GC

      IF (GCP .GT. 12.) THEN
         TRM = 0.
      ELSE
         CN  = DCMPLX(GP,GCP)
         CALL  CCSI(CN, CCI, CSI)
         RCI = DREAL(CCI)
         RSI = DIMAG(CSI)
         TRM = DSINH(GCP)*RCI - DCOSH(GCP)*RSI
      END IF

      CASMP  = C4*(DATAN(GL/GC) - PI2 + TRM)/(GCP*PI2)

      CININT = CAREA + CASINT + CASMP

   ELSE

      CININT = CAREA

   END IF
99 RETURN
   END

*****************************************************************************


      COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     COMPUTES SINGLE INTEGRAL OF CF FROM AY TO BY                        *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMMON     /C/ W0, W1, W2, W3, W4
      COMMON     /S/ FLAGS, BADS, ESEST, NOFUNS
      PARAMETER  (LEVMY = 60)
      COMPLEX*16 AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION  Y(17), CY(17), CYSV(8,LEVMY), YSV(8,LEVMY)
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     In order to prevent undersampling the function, a               *
*     minimum number of function evaluations will be                  *
*     required.  MINFUN as calculated below will force                *
*     the adaptive routine to zoom in to a panel                      *
*     of approximate width 10.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      MINFUN = 8*IDINT(DNINT(DLOG(BY)/ALOG(2.)))

      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AY
      Y(17)  = BY
      STONEY = (BY-AY)/16.
      Y(9)   = (Y(1)  + Y(17))/2.
      Y(5)   = (Y(1)  + Y(9)) /2.
      Y(13)  = (Y(9)  + Y(17))/2.
      Y(3)   = (Y(1)  + Y(5))/2.
      Y(7)   = (Y(5)  + Y(9))/2.
      Y(11)  = (Y(9)  + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
   25    CY(J) = CF(Y(J))
      FLAGS  = 0.
      ESEST  = 0.
      NOFUNS = 9
      APREV  = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                 W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                 W4*CY(9))
      AREA   = APREV

   30 DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
   35    CY(J) = CF(Y(J))
      NOFUNS = NOFUNS + 8
      STEPY  = (Y(17) - Y(1))/16.
      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &          W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &          W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &                W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &                W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERS*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUNS .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70

   40 DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
   50    YSV(J,LEVY)  = Y(J+9)
```

```
      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
55       Y(19-2*J)  = Y(10-J)
      GO TO 30

60    FLAGS = FLAGS + 1.
      BADS  = Y(9)

70    CAREA = CAREA + ANOW + ADIFW
      ESEST = ESEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
78       Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

80    CSINT = CAREA
      RETURN
      END
```

D.6 Program PIMM

PROGRAM PIMM

```
*************************************************************************
*                                                                       *
*     This program calculates the integral IMM which represents         *
*     the inner product between scattered microstrip current with       *
*     itself in the junction geometry.                                  *
*                                                                       *
*************************************************************************

      PROGRAM    PIMM
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 DPOLE, DPOLEA, ASTRM, ASTRMA, RES
      COMPLEX*16 U0, U1, KDU1, DE, DM
      REAL*8     K0, KD, MUR, MURI, NM
      EXTERNAL   CFPOL, CFRES, CFASMP
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/  FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/  W0, W1, W2, W3, W4
      COMMON     /D/  RELERY
      COMMON     /E/  PI, PI2, CI
      COMMON     /F/  GAMA0
      COMMON     /G/  G, G2
      COMMON     /S/  FLAGS, BADS, ESEST, NOFUNS
      COMMON     /Z/  KEY, AI, BI, A0, B0
      COMMON     /L2/ U0, U1, KDU1, DE, DM, CCH, CSH
      DIMENSION  AT(2), BT(2), CV(2)

      OPEN (3, FILE = '../computed.data/GAMA0')
      OPEN (4, FILE = '../computed.data/THETAC.IMM')
      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (9, FILE = '../defined.data/program.param')
      OPEN (8, FILE = '../computed.data/IMM')

      READ (3,*) GAMA0
      READ (4,*) BT(1)
      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (9,*) AT(1), BT(2), AG, NG, MG, MS, DELTA

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      G      = 10.
      G2     = G**2
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      W0 =  3956./14175.
      W1 = 23552./14175.
      W2 = -3712./14175.
      W3 = 41984./14175.
      W4 = -18160./14175.

      AT(2) = BT(1)
      K0    = 2.*PI*FREQ/SPEEDC
      KD    = K0*DSQRT(EPSR)*D
```

```
IF (KD .LE. 0.05) THEN
   RELERY = 1.D-6
ELSE
   RELERY = 1.D-7
END IF

WMK   = KO*WM/2.
NM    = DSQRT(EPSM)
Q     = NM*KO*WM
PI2A  = DATAN(100./Q)
SMALL = PI2 - PI2A

WRITE (6,*) ' '
WRITE (6,*) 'PI2A =', PI2A

DPOLE  = CSINT(CFPOL, AT(1), PI2A, 1.D-7)
SMLI   = Q/SMALL
EPSI   = .5/((1.+EPSR)*NM)
MURI   = .5*MUR/(NM*EPSM*(1.+MUR))
S      = DSIN(SMLI)
T      = DCOS(SMLI)
DPOLEA = (-2.*CI*MURI + (MURI-EPSI)*(T-CI*S))/SMLI
DPOLE  = DPOLE + DPOLEA

IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
WRITE (6,*) 'DPOLE =', DPOLE, NOFUNS

ASTRM  = CSINT(CFASMP, AT(1), PI2A, 1.D-7)
ASTRMA = -2.*DSQRT(SMALL)*(MURI+EPSI)/(KO*WM*G)
ASTRM  = ASTRM + ASTRMA

IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
WRITE (6,*) 'ASTRM =', ASTRM, NOFUNS

RES  = CSINT(CFRES, AT(1), BT(2), 1.D-7)
GO2  = GAMAO**2
CALL   F1F2(GO2, CDUM1, CDUM2)
DEPR = DREAL(GAMAO*((KO*D+EPSR/UO)*CCH+(1.+EPSR*KO*D*UO)*CSH/U1))
RES  = -RES*CI*PI*UO*U1*CSH*GAMAO/DEPR

IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
WRITE (6,*) 'RESIDUE =', RES, NOFUNS

DO 95 L = 1,2

   AI  = AG
   BI  = GAMAO - DELTA
   AO  = AT(L)
   BO  = BT(L)
   KEY = 1
   CVF = CGAUSS(NG,MG)

   WRITE (6,*) ' '
   IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
   WRITE (6,*) 'CVF =', CVF, NOFUNY

   AI  = GAMAO + DELTA
   KEY = 2
```

```
      IF (L .EQ. 1) THEN
         CVI = CGAUSS(NG,MG)
      ELSE
         CVI = CQSNG(MS)
      END IF

      IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
      WRITE (6,*) 'CVI =', CVI, NOFUNY

      CV(L) = CVF + CVI

   95 CONTINUE

      CVT = CV(1) + CV(2) + RES + DPOLE + ASTRM

      WRITE (6,*) 'CVT =', CVT

      CIMM = -4.*CVT*CI*ETA*EPSM/PI**2

      WRITE (6,*) ' '
      WRITE (6,*) 'IMM =', CIMM
      WRITE (8,*) CIMM

      CLOSE (3)
      CLOSE (4)
      CLOSE (7)
      CLOSE (8)
      CLOSE (9)
   99 STOP
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFMM(YA)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/  EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2, CF1B, CF2B, CF1P, CF2P, CFUN
      COMMON     /Z/  KEY, AI, BI, AO, BO

      Y = YA
      IF (KEY .EQ. 1) Y = AI + BI - Y

      Y2   = Y**2
      CALL   F1F2(Y2, CF1Y, CF2Y)
      CT1  = XS2*(CF1Y - CF1B + (B2-Y2)*CF1P)
      CT2  = XC2*(CF2Y - CF2B + (B2-Y2)*CF2P)
      CALL   J0(Y*XS*WMK, RJ0)
      CFMM = RJ0**2*Y*((CT1+CT2)/(Y2*XC2-EPSM)**2-
     &       CFUN/(Y2+G2/XC))
      RETURN
      END
```

```
*******************************************************************************

      SUBROUTINE F1F2(Y2, CF1, CF2)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U1, KDU1, DE, DM
      REAL*8     K0, MUR
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /L2/ U0, U1, KDU1, DE, DM, CCH, CSH

      U0   = ZSQRT(DCMPLX(Y2-1.,0.))
      U1   = ZSQRT(DCMPLX(Y2-EPSR*MUR,0.))
      KDU1 = K0*D*U1

      IF (DREAL(KDU1) .GT. 10.) THEN
         DE  = U1 + EPSR*U0
         DM  = U1 + MUR*U0
         CF1 = -MUR/DM
         CF2 = U0*U1/DE
      ELSE
         CEP = ZEXP(KDU1)
         CEN = ZEXP(-KDU1)
         CCH = (CEP+CEN)/2.
         CSH = (CEP-CEN)/2.
         DE  = U1*CSH + EPSR*U0*CCH
         DM  = U1*CCH + MUR*U0*CSH
         CF1 = -MUR*CSH/DM
         CF2 = U0*U1*CSH/DE
      END IF
      RETURN
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFPOL(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U1, KDU1, DE, DM
      REAL*8     K0, MUR
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /E/  PI, PI2, CI
      COMMON     /L2/ U0, U1, KDU1, DE, DM, CCH, CSH

      XC2 = DCOS(X)**2
      XT  = DTAN(X)
      B2  = EPSM/XC2
      CALL  F1F2(B2, CF1, CF2)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CF1P = MUR*(1.+MUR*U1/U0)/(2.*U1*DM**2)
         CF2P = .5*(EPSR*U0**2/U1+U1**2/U0)/DE**2
      ELSE
         CF1P = MUR*(CSH*(CCH+MUR*U1*CSH/U0)-KDU1)/(2.*U1*DM**2)
         CF2P = .5*(CSH*(EPSR*U0**2*CCH/U1+U1**2*CSH/U0)
     &           + EPSR*K0*D*U0**2)/DE**2
      END IF

      A     = DSIN(X)*WMK
      B     = DSQRT(B2)
      AB    = A*B
      CALL    BES0 (AB, RJ0, RY0)
      CALL    BES1 (AB, RJ1, RY1)
      CIC   = PI*A*(RJ0*(RY1+2.*CI*RJ1) + RJ1*RY0)/(4.*B)
      CJC   = -PI2*RJ0*(RY0 + CI*RJ0)
```

```
      CFPOL = (XT**2*(CIC*CF1+CJC*CF1P)+CIC*CF2+CJC*CF2P)/XC2
      RETURN
      END

*****************************************************************************

      COMPLEX*16 FUNCTION CFRES(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     K0, MUR
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /F/ GAMA0

      XC2   = DCOS(X)**2
      CALL    J0(WMK*DSIN(X)*GAMA0, RJ0)
      FCT   = (RJ0/(GAMA0**2-EPSM/XC2))**2/XC2
      CFRES = DCMPLX(FCT, 0.)
      RETURN
      END

*****************************************************************************

      COMPLEX*16 FUNCTION CFASMP(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U1, KDU1, DE, DM
      REAL*8     K0, MUR
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /G/ G, G2

      XC   = DCOS(X)
      XC2  = XC**2
      SXC  = DSQRT(XC)
      XS   = DSIN(X)
      B2   = EPSM/XC2
      U0   = DSQRT(B2-1.)
      U1   = ZSQRT(DCMPLX(B2-EPSR*MUR, 0.))
      KDU1 = K0*D*U1

      IF (DREAL(KDU1) .GT. 10.) THEN
         DE   = U1 + EPSR*U0
         DM   = U1 + MUR*U0
         CF1P = MUR*(1.+MUR*U1/U0)/(2.*U1*DM**2)
         CF2P = .5*(EPSR*U0**2/U1+U1**2/U0)/DE**2
      ELSE
         CEP  = ZEXP(KDU1)
         CEN  = ZEXP(-KDU1)
         CCH  = (CEP+CEN)/2.
         CSH  = (CEP-CEN)/2.
         DE   = U1*CSH + EPSR*U0*CCH
         DM   = U1*CCH + MUR*U0*CSH
         CF1P = MUR*(CSH*(CCH+MUR*U1*CSH/U0)-KDU1)/(2.*U1*DM**2)
         CF2P = .5*(CSH*(EPSR*U0**2*CCH/U1+U1**2*CSH/U0)
     &           + EPSR*K0*D*U0**2)/DE**2
      END IF

      CALL      BESM(WMK*XS*G/SXC, RIK)
      CFASMP = -RIK*(XS**2*CF1P/XC2+CF2P)/XC2
      RETURN
      END
```

```
*************************************************************************

%     Insert SUBROUTINE BESM(ARG, RIK) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE BES0(ARG, RJ0, RY0) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE BES1(ARG, RJ1, RY1) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE J0(ARG, RJ0) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE CCSI(C, CCI, CSI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCIS(X, RSI, RCI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCIL(X, RSI, RCI) from program PIAA here!

*************************************************************************

%     Insert SUBROUTINE SCI(X, RSI, RCI) from program PIAA here!

*************************************************************************

%     -- IMPORTANT NOTE -- IMPORTANT NOTE -- IMPORTANT NOTE --

%     Insert COMPLEX*16 FUNCTION CGAUSS(N,M) and
%            COMPLEX*16 FUNCTION CQSNG(M)    from program PIAA here!

%     These two routines will need to be modified slightly
%     before they will work in this program.  8 lines need
%     to be modified, 3 comment lines and 5 program lines.
%     7 of the changes are easy to do - simply replace the first
%     7 occurences of CFAA by CFMM.  The new lines will look as
%     follows (in order of occurence):

*     CFMM.  Some definitions:                                        *
*     CFMM : complex external function of a real variable             *
      EXTERNAL   CFMM
           CSUM = CSUM + W(J)*(CININT(CFMM, Z(J)*BAHN+BAM)+
     &                         CININT(CFMM,-Z(J)*BAHN+BAM))
*     CFMM : complex external function of a real variable             *
      EXTERNAL   CFMM

%     The last change is in line 300 of CQSNG.  For the integral
%     IMM the 1/sqrt singularity occurs at THETA=PI/2.  This is
%     taken into account by the modification shown here:

  300    CSUM  = CSUM + W(J)*CININT(CFMM,THETA)*DSQRT(PI2-THETA)

%     The above line is VERY IMPORTANT!  PLEASE NOTE!

*************************************************************************
```

```
*************************************************************************

      COMPLEX*16 FUNCTION CININT(CFMM,X)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     This routine calculates the inner gamma integral for the          *
*     double integral IMM.  The result is returned to the               *
*     quadrature rules, CGAUSS or CQSNG.                                *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     C1, C3, CBP, K0, MUR
      COMPLEX*16 U0, U1, KDU1, DE, DM, CCI, CSI
      COMMON     /A/  EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/  FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/  W0, W1, W2, W3, W4
      COMMON     /D/  RELERY
      COMMON     /E/  PI, PI2, CI
      COMMON     /G/  G, G2
      COMMON     /L1/ XS, XS2, XC, XC2, B2,
     &               CF1B, CF2B, CF1P, CF2P, CFUN
      COMMON     /L2/ U0, U1, KDU1, DE, DM, CCH, CSH
      COMMON     /Z/  KEY, AI, BI, AO, BO
      PARAMETER  (LEVMY = 60)
      COMPLEX*16 AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION  Y(17), CY(17), CYSV(8,LEVMY), YSV(8,LEVMY)

      XS  = DSIN(X)
      XS2 = XS**2
      XC  = DCOS(X)
      XC2 = XC**2
      B   = DSQRT(EPSM)/XC
      B2  = B**2
      CALL  F1F2(B2, CF1B, CF2B)

      IF (DREAL(KDU1) .GT. 10.) THEN
         CF1P = MUR*(1.+MUR*U1/U0)/(2.*U1*DM**2)
         CF2P = .5*(EPSR*U0**2/U1+U1**2/U0)/DE**2
      ELSE
         CF1P = MUR*(CSH*(CCH+MUR*U1*CSH/U0)-KDU1)/(2.*U1*DM**2)
         CF2P = .5*(CSH*(EPSR*U0**2*CCH/U1+U1**2*CSH/U0)
     &          + EPSR*K0*D*U0**2)/DE**2
      END IF

      T    = XS2/XC2
      C4   = -T*CF1P - CF2P
      C3   = 1./(1.+EPSR)
      C2   = -B2*C4 - T*CF1B - CF2B
      C1   = -MUR*T/(1.+MUR)
      CFUN = C4/XC2

      IF (KEY .EQ. 2) THEN
         A  = WMK*XS
         GL = 100./A
         IF (GL .LE. B) GL = 1.1*B
         GL2 = GL**2
         BI  = GL
      END IF
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     In order to prevent undersampling the function, a               *
*     minimum number of function evaluations will be                  *
*     required.  MINFUN as calculated below will force                *
*     the adaptive routine to zoom in to a panel                      *
*     of approximate width 10.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      MINFUN = 8*IDINT(DNINT(DLOG(BI)/ALOG(2.)))

      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AI
      Y(17)  = BI
      STONEY = (BI-AI)/16.
      Y(9)   = (Y(1)  + Y(17))/2.
      Y(5)   = (Y(1)  + Y(9)) /2.
      Y(13)  = (Y(9)  + Y(17))/2.
      Y(3)   = (Y(1)  + Y(5))/2.
      Y(7)   = (Y(5)  + Y(9))/2.
      Y(11)  = (Y(9)  + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
   25    CY(J) = CFMM(Y(J))

      NOFUN = 9
      NOFUNY = NOFUNY + 9
      APREV = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                W4*CY(9))
      AREA  = APREV

   30 DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
   35    CY(J) = CFMM(Y(J))

      NOFUN  = NOFUN + 8
      NOFUNY = NOFUNY + 8
      STEPY  = (Y(17) - Y(1))/16.

      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &          W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &          W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &                W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &                W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERY*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUN  .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70

   40 DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
```

```
50    YSV(J,LEVY)  = Y(J+9)

      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
55       Y(19-2*J)  = Y(10-J)
      GO TO 30

60 FLAGY = FLAGY + 1.
      BADY  = Y(9)

70 CAREA = CAREA + ANOW + ADIFW
      EYEST = EYEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
78       Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

80 CONTINUE

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     The following lines compute the asymptotic portion of the      *
*     inner gamma integral from BI to infinity.                       *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IF (KEY .EQ. 2) THEN
         BP = 2.*A*B
         GP = 2.*A*GL
         H  = GP + BP
         HT = GP - BP

         IF (H .GE. 100.) THEN
            CALL SCIL(H, HSI, HCI)
         ELSE
            CALL SCI(H, HSI, HCI)
         END IF

         IF (HT .GE. 100.) THEN
            CALL SCIL(HT, HTSI, HTCI)
         ELSE IF (HT .LE. 1.) THEN
            CALL SCIS(HT, HTSI, HTCI)
         ELSE
            CALL SCI(HT, HTSI, HTCI)
         END IF

         E   = HTSI + HSI
         ET  = HTSI - HSI
         F   = HTCI + HCI
         FT  = HTCI - HCI
         SGP = DSIN(GP)
         CBP = DCOS(BP)
         SBP = DSIN(BP)
```

```
      GB  = .5/(GL2 - B2)

      IF (GP .GE. 100.) THEN
         CALL SCIL(GP, GSI, GCI)
      ELSE
         CALL SCI(GP, GSI, GCI)
      END IF

      D1 = (1.+SGP)*GB/B2 + (((CBP+.5*BP*SBP)*E+(SBP-.5*BP*CBP)*FT
     &     - DLOG(2.*GL2*GB))/2. - GSI)/B2**2
      D2 = GL*GB*(1.+SGP)/B2 + ((CBP+BP*SBP)*ET + (SBP-BP*CBP)*F +
     &     DLOG(HT/H))/(4.*B*B2)
      D3 = (1.+SGP)*GB + .5*A*(SBP*E - CBP*FT)/B
      D4 = (1.+SGP)*GL*GB - ((CBP-BP*SBP)*ET + (SBP+BP*CBP)*F +
     &     DLOG(HT/H))/(4.*B)

      CASINT = (C1*D1 + C2*D2 + C3*D3 + C4*D4)/(PI*A*XC2)
      GC = G/DSQRT(XC)
      GCP = 2.*A*GC

      IF (GCP .GT. 12.) THEN
         TRM = 0.
      ELSE
         CN  = DCMPLX(GP,GCP)
         CALL  CCSI(CN, CCI, CSI)
         RCI = DREAL(CCI)
         RSI = DIMAG(CSI)
         TRM = DSINH(GCP)*RCI - DCOSH(GCP)*RSI
      END IF

      CASMP = CFUN*(DATAN(GL/GC) - PI2 + TRM)/(GCP*PI2)
      CININT = CAREA + CASINT + CASMP

   ELSE
       CININT = CAREA
    END IF

99 RETURN
   END

*************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

*************************************************************************
```

D.7 Program PIAM

PROGRAM PIAM

```
************************************************************************
*                                                                      *
*     This program calculates the integral IAM which represents        *
*     the inner product between scattered electric field  and          *
*     scattered microstrip current in the junction geometry.           *
*                                                                      *
************************************************************************


      PROGRAM    PIAM
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U1, KDU1, DE, DM, RES
      REAL*8     K0, KD, MUR
      EXTERNAL   CFRES
      COMMON     /A/   EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/   FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/   W0, W1, W2, W3, W4
      COMMON     /D/   RELERY
      COMMON     /E/   PI, PI2, CI
      COMMON     /F/   GAMA0, GO2
      COMMON     /L2/  U0, U1, KDU1, DE, DM, CCH, CSH
      COMMON     /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX
      COMMON     /S/   FLAGS, BADS, ESEST, NOFUNS
      COMMON     /Z/   KEY, AI, BI, AO, BO

      OPEN (4, FILE = '../computed.data/GAMA0')
      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (9, FILE = '../defined.data/program.param')
      OPEN (8, FILE = '../computed.data/IAM')

      READ (4,*) GAMA0
      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (9,*) AT, BT, AG, DMY1, DMY2, DMY3, DELTA, DMY4, DMY5,
     &           DMY6, BG, NG, MG

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      W0 =  3956./14175.
      W1 = 23552./14175.
      W2 = -3712./14175.
      W3 = 41984./14175.
      W4 = -18160./14175.

      K0 = 2.*PI*FREQ/SPEEDC
      KD = K0*DSQRT(EPSR)*D

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*     For electrically thin substrates, a smaller relative error       *
*     produces more accurate inner integrals.  Hence, the next         *
*     statement.                                                       *
*                                                                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      IF (KD .LE. 0.05) THEN
         RELERY = 1.D-6
      ELSE
         RELERY = 1.D-7
      END IF

      WMK   = WM*KO/2.
      WAK   = WA*KO/2.
      XC2   = 1.
      XS2   = 1.
      GO2   = GAMAO**2
      RES   = CSINT(CFRES, AT, BT, 1.D-8)
      CDUMY = CG(GO2)
      DEPR  = DREAL(GAMAO*((KO*D+EPSR/UO)*CCH+(1.+EPSR*KO*D*UO)*CSH/U1))
      RES   = -RES*CI*PI*UO*GAMAO*EPSR/DEPR

      WRITE (6,*) ' '
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
      WRITE (6,*) 'RESIDUE =', RES, NOFUNS

      AO  = AT
      BO  = BT
      AI  = AG
      BI  = GAMAO - DELTA
      KEY = 1
      CVF = CGAUSS(NG, MG)

      IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
      WRITE (6,*) 'CVF =', CVF, NOFUNY

      AI  = GAMAO + DELTA
      BI  = BG
      KEY = 2
      CVI = CGAUSS(NG, MG)

      IF (FLAGY .NE. 0.) WRITE (6,*) 'FLAGY', FLAGY, BADY
      WRITE (6,*) 'CVI =', CVI, NOFUNY

      CVT  = CVF + CVI + RES
      WRITE (6,*) 'CVT =', CVT

      CIAM = -4.*DSQRT(EPSM*EPSA)*CVT/PI**2

      WRITE (6,*) ' '
      WRITE (6,*) 'IAM =', CIAM
      WRITE (8,*) CIAM

      CLOSE (4)
      CLOSE (7)
      CLOSE (8)
      CLOSE (9)
   99 STOP
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFAM(YA)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/   EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX
      COMMON     /Z/   KEY, AI, BI, AO, BO
```

```
      Y  = YA

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                         *
*     Since the adaptive integration routine tends to zoom in to          *
*     to the left side of an interval, it performs better when            *
*     regions of steep slope appear towards the origin.  The next         *
*     statement rotates the function, so that the 1/DE pole is at         *
*     the origin.                                                         *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IF (KEY .EQ. 1) Y = AI + BI - Y
      Y2   = Y**2
      CGY2 = CG(Y2)
      CALL   J0(AMN*Y, RMN)
      CALL   J0(AMX*Y, RMX)
      CFAM = Y*RMN*RMX*((CGY2-CGBM)/(Y2-BM2)-(CGY2-CGBA)/(Y2-BA2))
      RETURN
      END

*************************************************************************

      COMPLEX*16 FUNCTION CG(Y2)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U0, U1, KDU1, DE, DM
      REAL*8     K0, MUR
      COMMON     /A/   EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX
      COMMON     /L2/  U0, U1, KDU1, DE, DM, CCH, CSH

      U0   = ZSQRT(DCMPLX(Y2 - 1., 0.))
      U1   = ZSQRT(DCMPLX(Y2 - EPSR*MUR, 0.))
      KDU1 = K0*D*U1

      IF (DREAL (KDU1) .GT. 600) THEN
         CG = (0., 0.)
      ELSE
         CEP = ZEXP(KDU1)
         CEN = ZEXP(-KDU1)
         CCH = (CEP + CEN)/2.
         CSH = (CEP - CEN)/2.
         DE  = U1*CSH + EPSR*U0*CCH
         DM  = U1*CCH + MUR*U0*CSH
         CG  = (EPSR*U0*XC2/DE + U1*XS2/DM)
      END IF
      RETURN
      END

*************************************************************************

      COMPLEX*16 FUNCTION CFRES(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     K0, MUR
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /F/ GAMA0, G02

      XC  = DCOS(X)
      XS  = DSIN(X)
      XC2 = XC**2
      XS2 = XS**2
      CALL  J0(WMK*XS*GAMA0, RJ0M)
```

```
      CALL  JO(WAK*XC*GAMA0, RJOA)
      FCT = RJOM*RJOA/((GO2-EPSM/XC2)*(GO2*XS2-EPSA))

      CFRES = DCMPLX(FCT, 0.)
      RETURN
      END

*************************************************************************

      SUBROUTINE JOHO(ARG, RJO, HO2)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*    Returns Bessel fct. RJO(ARG) and Hankel fct. HO2(ARG) (2nd kind)   *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 HO2, CI
      COMMON     /E/ PI, PI2, CI

      IF (ARG .LE. 0.) THEN
         WRITE (6,*) 'No negative argument allowed in JOHO -- Fatal'
      END IF

      IF (ARG .LE. 3.) THEN
         Y = (ARG/3.)**2
         RJO =(((((0.21000000E-3 *Y-0.39444000E-2)*Y+0.44447900E-1)
     &          *Y-0.31638660E00)*Y+0.12656208E+1)*Y-0.22499997E+1)
     &          *Y+0.10000000E+1
         RYO =(((((-.24846000E-3 *Y+0.42791600E-2)*Y-0.42612140E-1)
     &          *Y+0.25300117E00)*Y-0.74350384E00)*Y+0.60559366E00)
     &          *Y+0.36746691E00 + 2.*RJO*DLOG(ARG/2.)/PI
      ELSE
         Y = 3./ARG
         FO    = (((((0.14476000E-3 *Y-0.72805000E-3)*Y+0.13723700E-2)
     &            *Y-0.95120000E-4)*Y-0.55274000E-2)*Y-0.77000000E-6)
     &            *Y+0.79788456
         THETA = (((((0.13558000E-3 *Y-0.29333000E-3)*Y-0.54125000E-3)
     &            *Y+0.26257300E-2)*Y-0.39540000E-4)*Y-0.41663970E-1)
     &            *Y-0.78539816+ARG
         RJO = FO*DCOS(THETA)/DSQRT(ARG)
         RYO = FO*DSIN(THETA)/DSQRT(ARG)
      END IF

      HO2 = RJO - CI*RYO
      RETURN
      END


*************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

*************************************************************************

%     -- IMPORTANT NOTE -- IMPORTANT NOTE -- IMPORTANT NOTE --

%     Insert COMPLEX*16 FUNCTION CGAUSS(N,M) from program PIAA here!

%     This routine will need to be modified slightly
%     before it will work in this program.  5 lines need
%     to be modified, 2 comment lines and 3 program lines.
%     The changes are easy to do - simply replace the
```

```
%     5 occurences of CFAA by CFAM.  The new lines will look as
%     follows (in order of occurence):

*     CFAM.  Some definitions:                                          *
*     CFAM : complex external function of a real variable               *
      EXTERNAL   CFAM
           CSUM = CSUM + W(J)*(CININT(CFAM, Z(J)*BAHN+BAM)+
     &                         CININT(CFAM,-Z(J)*BAHN+BAM))


**********************************************************************

      COMPLEX*16 FUNCTION CININT(CFAM,X)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     This routine calculates the inner gamma integral for the        *
*     double integral IAM.  The result is returned to the             *
*     quadrature rule, CGAUSS.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 HO2M, HO2A
      REAL*8     KO, MUR
      COMMON     /A/   EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /B/   FLAGY, BADY, EYEST, NOFUNY
      COMMON     /C/   W0, W1, W2, W3, W4
      COMMON     /D/   RELERY
      COMMON     /E/   PI, PI2, CI
      COMMON     /LAM/ XS2, XC2, BM2, BA2, CGBM, CGBA, AMN, AMX
      COMMON     /Z/   KEY, AI, BI, AO, BO
      PARAMETER  (LEVMY = 60)
      COMPLEX*16 AREA, ANOW, APREV, ALEFT, ADIFF, ADIFW, ARIGHT(LEVMY)
      DIMENSION  Y(17), CY(17), CYSV(8,LEVMY), YSV(8,LEVMY)

      XS   = DSIN(X)
      XC   = DCOS(X)
      XS2  = XS**2
      XC2  = XC**2
      AM   = XS*WMK
      AA   = XC*WAK
      BM   = DSQRT(EPSM)/XC
      BA   = DSQRT(EPSA)/XS
      BM2  = BM**2
      BA2  = BA**2
      AMN  = DMIN1(AM, AA)
      AMX  = DMAX1(AM, AA)
      CGBM = CG(BM2)
      CGBA = CG(BA2)
      XPOL = 1./(XS2*XC2*(BM2-BA2))

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     In order to prevent undersampling the function, a               *
*     minimum number of function evaluations will be                  *
*     required.  MINFUN as calculated below will force                *
*     the adaptive routine to zoom in to a panel                      *
*     of approximate width 10.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      MINFUN = 8*IDINT(DNINT(DLOG(BI)/ALOG(2.)))

      CAREA  = (0.,0.)
      LEVY   = 1
      Y(1)   = AI
      Y(17)  = BI
      STONEY = (BI-AI)/16.
      Y(9)   = (Y(1)  + Y(17))/2.
      Y(5)   = (Y(1)  + Y(9)) /2.
      Y(13)  = (Y(9)  + Y(17))/2.
      Y(3)   = (Y(1)  + Y(5))/2.
      Y(7)   = (Y(5)  + Y(9))/2.
      Y(11)  = (Y(9)  + Y(13))/2.
      Y(15)  = (Y(13) + Y(17))/2.

      DO 25 J = 1, 17, 2
25       CY(J) = CFAM(Y(J))

      NOFUN = 9
      NOFUNY = NOFUNY + 9
      APREV = STONEY*(W0*(CY(1)+CY(17)) + W1*(CY(3)+CY(15)) +
     &                W2*(CY(5)+CY(13)) + W3*(CY(7)+CY(11)) +
     &                W4*CY(9))
      AREA  = APREV

30    DO 35 J = 2, 16, 2
         Y(J)  = (Y(J-1) + Y(J+1))/2.
35       CY(J) = CFAM(Y(J))

      NOFUN  = NOFUN + 8
      NOFUNY = NOFUNY + 8
      STEPY  = (Y(17) - Y(1))/16.

      ALEFT  = (W0*(CY(1)+CY(9)) + W1*(CY(2)+CY(8)) +
     &          W2*(CY(3)+CY(7)) + W3*(CY(4)+CY(6)) +
     &          W4*CY(5))*STEPY

      ARIGHT(LEVY) = (W0*(CY(9)+CY(17)) + W1*(CY(10)+CY(16)) +
     &                W2*(CY(11)+CY(15)) + W3*(CY(12)+CY(14)) +
     &                W4*CY(13))*STEPY

      ANOW   = ALEFT + ARIGHT(LEVY)
      ADIFF  = ANOW - APREV
      ADIFW  = ADIFF/1023.
      AREA   = AREA + ADIFW
      ESTERY = ZABS(ADIFW)
      TOLERR = ZABS(AREA)*RELERY*STEPY/STONEY

      IF (LEVY   .GE. LEVMY)  GO TO 60
      IF (NOFUN  .LT. MINFUN) GO TO 40
      IF (ESTERY .LE. TOLERR) GO TO 70

40    DO 50 J = 1, 8
         CYSV(J,LEVY) = CY(J+9)
50       YSV(J,LEVY)  = Y(J+9)

      LEVY  = LEVY + 1
      APREV = ALEFT

      DO 55 J = 1, 8
         CY(19-2*J) = CY(10-J)
55       Y(19-2*J)  = Y(10-J)
      GO TO 30
```

```
   60 FLAGY = FLAGY + 1.
      BADY  = Y(9)

   70 CAREA = CAREA + ANOW + ADIFW
      EYEST = EYEST + ESTERY
      LEVY  = LEVY - 1

      IF (LEVY .LE. 0) GO TO 80

      APREV = ARIGHT(LEVY)
      Y(1)  = Y(17)
      CY(1) = CY(17)

      DO 78 J = 1, 8
         CY(2*J+1) = CYSV(J,LEVY)
   78    Y(2*J+1)  = YSV(J,LEVY)
      GO TO 30

   80 CONTINUE

      IF (KEY .EQ. 2) THEN
         CALL     JO(AMN*BM, RJOM)
         CALL     JO(AMN*BA, RJOA)
         CALL     JOHO(AMX*BM, DUM, HO2M)
         CALL     JOHO(AMX*BA, DUM, HO2A)
         CPOL   = CI*PI2*(CGBA*RJOA*HO2A - CGBM*RJOM*HO2M)
         CININT = XPOL*(CAREA + CPOL)
      ELSE
         CININT = XPOL*CAREA
      END IF
      RETURN
      END

*************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

*************************************************************************
```

D.8 Program PIAG

PROGRAM PIAG

```
************************************************************************
*                                                                      *
*     This program calculates the integral IAG which represents        *
*     the inner product between scattered electric field in the        *
*     junction geometry and ground plane current of the incident       *
*     microstrip mode geometry.                                        *
*                                                                      *
************************************************************************


      PROGRAM    PIAG

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     K0, MUR
      EXTERNAL   CFCT, CFF
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /C/ W0, W1, W2, W3, W4
      COMMON     /E/ PI, PI2, CI
      COMMON     /F/ CFNA
      COMMON     /S/ FLAGS, BADS, ESEST, NOFUNS

      OPEN (9, FILE = '../defined.data/phys.param')
      OPEN (8, FILE = '../computed.data/IAG')

      READ (9,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      W0 =   3956./14175.
      W1 =  23552./14175.
      W2 =  -3712./14175.
      W3 =  41984./14175.
      W4 = -18160./14175.

      K0   = 2.*PI*FREQ/SPEEDC
      WMK  = K0*WM/2.
      RNM  = DSQRT(EPSM)
      WAK  = K0*WA/2.
      RNA  = DSQRT(EPSA)
      XMAX = DSQRT((100./(K0*D))**2 + EPSR*MUR - EPSM)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*     XMAX is the outer limit of integration and is determined by the  *
*     exponential decay of the integrand.  When XMAX is calculated as  *
*     above, the integrand is down in amplitude by a factor of         *
*     EXP(-100).                                                       *
*                                                                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      WRITE (6,*) ' '
      CFNA = CFF(RNA)
      CINT = CSINT(CFCT, 1.D-7, XMAX, 1.D-8)
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS

      CALL JO(WAK*RNM, RJOA)
```

```
      CALL JO(WMK*RNA, RJOM)
      CALL STRUV(WMK*RNA, HO)

      CIAG = RJOA*(-CI*RNA/PI2*CINT+CFNA*(CI*HO-RJOM))

      WRITE (6,*) 'IAG =', CIAG, NOFUNS
      WRITE (8,*) CIAG

      CIAGP = -RJOA*RJOM*CFNA
      WRITE (6,*) 'IAGP =', CIAGP
      WRITE (8,*) CIAGP

      CLOSE (8)
      CLOSE (9)
   99 STOP
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFCT(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/ EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /E/ PI, PI2, CI
      COMMON     /F/ CFNA

      CALL   JO(WMK*X, RJO)
      CFCT = RJO*(CFF(X)-CFNA)/(X**2-EPSA)
      RETURN
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFF(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 TO, T1, KDT1, TE, TM
      REAL*8     KO, MUR
      COMMON     /A/ EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /E/ PI, PI2, CI
      COMMON     /F/ CFNA

      X2   = X**2
      TO   = ZSQRT(DCMPLX(X2 + EPSM - 1., 0.))
      T1   = ZSQRT(DCMPLX(X2 + EPSM - EPSR*MUR, 0.))
      KDT1 = KO*D*T1

      IF (DREAL(KDT1) .GT. 100.) THEN
         CFF = (0., 0.)
      ELSE
         CP  = ZEXP(KDT1)
         CN  = ZEXP(-KDT1)
         CCH = (CP + CN)/2.
         CSH = (CP - CN)/2.
         TE  = T1*CSH + EPSR*TO*CCH
         TM  = T1*CCH + MUR*TO*CSH
         CFF = (EPSR*EPSM*TO/TE + X2*T1/TM)/(X2+EPSM)
      END IF
      RETURN
      END
```

```
************************************************************************

      SUBROUTINE STRUV(X, HO)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     This subroutine computes the zeroth order Struve function,      *
*     HO, for the argument X.  For X less than or equal to 20, a      *
*     power series is used.  For X greater than 20, an asymptotic     *
*     expansion involving the zeroth order Neumann function is used.  *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      COMPLEX*16 CI
      COMMON     /E/ PI, PI2, CI

      X2 = X**2

      IF (X .GT. 20.) GO TO 30

      TERM = 1.
      SUM1 = 1.
      DO 10 K = 1, 100
         TERM = -X2*TERM/(2*K+1)**2
         SUM2 = SUM1 + TERM
         IF ((DABS(SUM1/SUM2-1.) .LT. 1.D-9) .OR.
     &               (DABS(SUM2) .LT. 1.D-20)) THEN
            GO TO 20
         ELSE
            SUM1 = SUM2
         END IF
   10 CONTINUE

      PRINT *, 'STRUVE FUNCTION DID NOT CONVERGE WITH 100 TERMS'
      STOP
   20 CONTINUE

      HO = X*SUM2/PI2
      GO TO 99

   30 CONTINUE
      X8 = 8.*X
      ARG = X - PI2/2.
      XI = 1./(X*PI2)
      YO = DSQRT(XI)*(DSIN(ARG)*(1.-4.5/X8**2) + DCOS(ARG)*
     &               (-1./X8 + 37.5/X8**3))
      HO = YO + XI*(1.-1./X2)
   99 RETURN
      END

************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

************************************************************************
```

D.9 Program PIMI

PROGRAM PIMI

```
************************************************************************
*                                                                      *
*     This program calculates the integral IMI which represents        *
*     the inner product between scattered microstrip current in the    *
*     junction geometry with the tangential electric field             *
*     of the incident slot line mode.  The inner product is taken      *
*     over the surface of the microstrip.                              *
*                                                                      *
************************************************************************


      PROGRAM    PIMI

      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     K0, MUR
      EXTERNAL   CFCT, CFF
      COMMON     /A/ EPSR, MUR, K0, D, WMK, EPSM, WAK, EPSA
      COMMON     /C/ W0, W1, W2, W3, W4
      COMMON     /E/ PI, PI2, CI
      COMMON     /F/ CFNM
      COMMON     /S/ FLAGS, BADS, ESEST, NOFUNS

      OPEN (9, FILE = '../defined.data/phys.param')
      OPEN (8, FILE = '../computed.data/IMI')

      READ (9,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA

      PI     = 3.14159265358979
      PI2    = 1.57079632679489
      ETA    = 376.7
      SPEEDC = 299792458.
      CI     = (0.,1.)

      W0 =   3956./14175.
      W1 =  23552./14175.
      W2 =  -3712./14175.
      W3 =  41984./14175.
      W4 = -18160./14175.

      K0   = 2.*PI*FREQ/SPEEDC
      WMK  = K0*WM/2.
      RNM  = DSQRT(EPSM)
      WAK  = K0*WA/2.
      RNA  = DSQRT(EPSA)
      XMAX = DSQRT((100./(K0*D))**2 + EPSR*MUR - EPSM)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*     XMAX is the outer limit of integration and is determined by the  *
*     exponential decay of the integrand.  When XMAX is calculated as  *
*     above, the integrand is down in amplitude by a factor of         *
*     EXP(-100).                                                       *
*                                                                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      WRITE (6,*) ' '

      CFNM = CFF(RNM)
      CINT = CSINT(CFCT, 1.D-7, XMAX, 1.D-8)
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
```

```
      CALL   JO(WMK*RNA, RJOM)
      CALL   JO(WAK*RNM, RJOA)
      CALL   STRUV(WAK*RNM, HO)
      CIMI = RJOM*(CI*RNM/PI2*CINT+CFNM*(RJOA-CI*HO))

      WRITE (6,*) 'IMI =', CIMI, NOFUNS
      WRITE (8,*) CIMI

      CLOSE (8)
      CLOSE (9)
   99 STOP
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFCT(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMMON     /A/ EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /E/ PI, PI2, CI
      COMMON     /F/ CFNM

      CALL   JO(WAK*X, RJO)
      CFCT = RJO*(CFF(X)-CFNM)/(X**2-EPSM)
      RETURN
      END

*******************************************************************************

      COMPLEX*16 FUNCTION CFF(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      COMPLEX*16 SO, S1, KDS1, SE, SM
      COMMON     /A/ EPSR, MUR, KO, D, WMK, EPSM, WAK, EPSA
      COMMON     /E/ PI, PI2, CI

      X2 = X**2
      SO = ZSQRT(DCMPLX(X2 + EPSA - 1., 0.))
      S1 = ZSQRT(DCMPLX(X2 + EPSA - EPSR*MUR, 0.))
      KDS1 = KO*D*S1

      IF (DREAL(KDS1) .GT. 100.) THEN
         CFF = (0., 0.)
      ELSE
         CP  = ZEXP(KDS1)
         CN  = ZEXP(-KDS1)
         CCH = (CP + CN)/2.
         CSH = (CP - CN)/2.
         SE  = S1*CSH + EPSR*SO*CCH
         SM  = S1*CCH + MUR*SO*CSH
         CFF = (EPSR*X2*SO/SE + EPSA*S1/SM)/(X2+EPSA)
      END IF
      RETURN
      END

*******************************************************************************

%     Insert SUBROUTINE STRUV(X, HO) from program PIAG here!
```

```
*************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

*************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

*************************************************************************
```

D.10 Program ZM

PROGRAM ZM

```
************************************************************************
*                                                                      *
*     This program computes the characteristic impedance of the        *
*     microstrip. The current-power definition is used for this        *
*     line.  When the strip current is normalized to 1 Amp, the        *
*     impedance equals the microstrip norm.                            *
*                                                                      *
************************************************************************


      PROGRAM    ZM
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     KO, MUR
      EXTERNAL   CFZM
      COMMON     /A/ EPSR, MUR, KO, D, D2, WM, EPSM, CI
      COMMON     /C/ WO, W1, W2, W3, W4
      COMMON     /S/ FLAGS, BADS, ESEST, NOFUNS

      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (9, FILE = '../defined.data/program.param')
      OPEN (8, FILE = '../computed.data/Zstrip')

      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (9,*) DMY1,DMY2, AY, DMY3,DMY4,DMY5,DMY6,DMY7,DMY8,DMY9,
     &           BY, DMY10,DMY11, RELERS

      WO =   3956./14175.
      W1 =  23552./14175.
      W2 =  -3712./14175.
      W3 =  41984./14175.
      W4 = -18160./14175.

      CI     = (0.,1.)
      PI     = 3.14159265358979
      ETA    = 376.7
      SPEEDC = 299792458.
      D2     = D/2.
      KO     = 2.*PI*FREQ/SPEEDC

      CAREA  = CSINT(CFZM, AY, BY, RELERS)
      ZSTRIP = ETA*DSQRT(EPSM)*DREAL(CAREA)/(2.*PI)

      WRITE (6,*) ' '
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
      WRITE (6,*) 'MICROSTRIP CHARACTERISTIC IMPEDANCE, ZSTRIP = '
     &                                                 , ZSTRIP
      WRITE (8,*) ZSTRIP

      CLOSE (7)
      CLOSE (8)
      CLOSE (9)
   99 STOP
      END
```

```
*************************************************************************

      COMPLEX*16 FUNCTION CFZM(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U1, DE, DM, SHU1, SHU12
      REAL*8     K0, MUR, CHU1, KDU1
      COMMON     /A/ EPSR, MUR, K0, D, D2, WM, EPSM, CI

      CALL     J0(K0*WM*X/2., RJ0)
      X2     = X**2
      GAMMA2 = EPSM+X2
      U02    = GAMMA2-1.
      U0     = DSQRT(U02)
      U12    = GAMMA2-EPSR*MUR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     This function represents the norm, that is E x H integrated       *
*     over the infinite surface perpendicular to the propagation        *
*     direction of the microstrip fundamental mode.                     *
*                                                                       *
*     For speed and due to the author's original ignorance, the         *
*     function below is evaluated in a slightly indirect fashion.       *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IF (U12 .GE. 0.) THEN
         U1   = DSQRT(U12)
         KDU1 = K0*D*U1
         IF (KDU1 .GT. 10.) THEN
            DEA  = EPSR*U0 + U1
            DMA  = MUR*U0 + U1
            CFZM = RJ0**2/GAMMA2*(EPSM/DEA**2*(EPSR*U02/U1+U12/U0)
     &             +X2*(U0+MUR*U1)/DMA*(2./DEA+MUR/(DMA*U0*U1)))
         ELSE
            SHU1  = DSINH(KDU1)
            SH2U1 = DSINH(2.*KDU1)/(4.*K0*U1)
            CHU1  = DCOSH(KDU1)
         END IF
      ELSE
         U12   = -U12
         U1    = DSQRT(U12)
         KDU1  = K0*D*U1
         SHU1  = CI*DSIN(KDU1)
         SH2U1 = DSIN(2.*KDU1)/(4.*K0*U1)
         CHU1  = DCOS(KDU1)
         U1    = CI*U1
         U12   = -U12
      END IF
      IF (KDU1 .LE. 10.) THEN
         DE    = U1*SHU1+EPSR*U0*CHU1
         DM    = MUR*U0*SHU1+U1*CHU1
         SHU12 = SHU1**2/(2.*K0*U0)
         CFZM  = RJ0**2/GAMMA2*((EPSM/DE*(EPSR*U02*(SH2U1+D2)+U12*SHU12)
     &           +X2*U0*U1/DM*(SH2U1+D2+MUR*SHU12))/DE+X2/DM
     &           *(U0*U1/DE+MUR/DM)*(SH2U1-D2+MUR*SHU12))*2.*K0
      END IF
      RETURN
      END
```

```
*************************************************************************

%     Insert SUBROUTINE JO(ARG, RJO) from program PIAA here!

*************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

*************************************************************************
```

D.11 Program ZA

PROGRAM ZA

```
************************************************************************
*                                                                      *
*     This program computes the characteristic impedance of the        *
*     slot line.  The voltage-power definition is used for this        *
*     line.  When the slot voltage is normalized to 1 Volt, the        *
*     impedance equals the inverse of the slot norm.                   *
*                                                                      *
************************************************************************


      PROGRAM    ZA
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      REAL*8     K0, MUR
      EXTERNAL   CFZA
      COMMON     /A/ EPSR, MUR, K0, D, D2, WA, EPSA, CI
      COMMON     /C/ W0, W1, W2, W3, W4
      COMMON     /S/ FLAGS, BADS, ESEST, NOFUNS

      OPEN (7, FILE = '../defined.data/phys.param')
      OPEN (9, FILE = '../defined.data/program.param')
      OPEN (8, FILE = '../computed.data/Zslot')

      READ (7,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (9,*) DMY1, DMY2, AY, DMY3, DMY4, DMY5, DMY6, DMY7,
     &           DMY8, DMY9, BY, DMY10,DMY11, RELERS

      W0 =   3956./14175.
      W1 =  23552./14175.
      W2 =  -3712./14175.
      W3 =  41984./14175.
      W4 = -18160./14175.

      CI     = (0.,1.)
      PI     = 3.14159265358979
      ETA    = 376.7
      SPEEDC = 299792458.
      D2     = D / 2.
      K0     = 2.*PI*FREQ/SPEEDC

      CAREA  = CSINT(CFZA, AY, BY, RELERS)
      ZSLOT  = ETA*PI/(DSQRT(EPSA)*K0*DREAL(CAREA))

      WRITE (6,*) ' '
      IF (FLAGS .NE. 0.) WRITE (6,*) 'FLAGS', FLAGS, BADS
      WRITE (6,*) 'SLOTLINE CHARACTERISTIC IMPEDANCE, ZSLOT = ', ZSLOT
      WRITE (8,*) ZSLOT

      CLOSE (7)
      CLOSE (8)
      CLOSE (9)
   99 STOP
      END
```

```
*************************************************************************

      COMPLEX*16 FUNCTION CFZA(X)
      IMPLICIT   REAL*8     (A, B, D - H, O - Z)
      IMPLICIT   COMPLEX*16 (C)
      COMPLEX*16 U1, DE, DM, SHU1, FE, FM, FEDE, FMDM, FEDM
      REAL*8     KO, MUR, CHU1, KDU1
      COMMON     /A/ EPSR, MUR, KO, D, D2, WA, EPSA, CI

      CALL     JO(KO*WA*X/2., RJO)
      X2     = X**2
      GAMMA2 = EPSA+X2
      U02    = GAMMA2-1.
      UO     = DSQRT(U02)
      U12    = GAMMA2-EPSR*MUR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*     This function represents the norm, that is E x H integrated      *
*     over the infinite surface perpendicular to the propagation       *
*     direction of the slot line fundamental mode.                      *
*                                                                       *
*     For speed and due to the author's original ignorance, the         *
*     function below is evaluated in a slightly indirect fashion.       *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      IF (U12 .GE. 0.) THEN
         U1   = DSQRT(U12)
         KDU1 = KO*D*U1

         IF (KDU1 .GT. 10.) THEN
            CFZA = RJO**2*(X2/GAMMA2*((1.+1/U02)/UO+(EPSR/U12+1./MUR)
     &              /U1)+1./UO+1./(MUR*U1))/(2.*KO)
         ELSE
            SHU1 = DSINH(KDU1)
            SH24 = DSINH(2.*KDU1)/(4.*KO*U1)
            CHU1 = DCOSH(KDU1)
            CH24 = (1.-DCOSH(2.*KDU1))/(4.*KO*U1)
         END IF

      ELSE
         U12  = -U12
         U1   = DSQRT(U12)
         KDU1 = KO*D*U1
         SHU1 = CI*DSIN(KDU1)
         SH24 = DSIN(2.*KDU1)/(4.*KO*U1)
         CHU1 = DCOS(KDU1)
         U1   = CI*U1
         CH24 = (1.-DCOS(2.*KDU1))/(4.*KO*U1)
         U12  = -U12
      END IF

      IF (KDU1 .LE. 10.) THEN
         DE   = U1*SHU1+EPSR*UO*CHU1
         DM   = MUR*UO*SHU1+U1*CHU1
         FE   = EPSR*UO*SHU1+U1*CHU1
         FM   = U1*SHU1+MUR*UO*CHU1
         FEDE = FE/DE
         FMDM = FM/DM
         FEDM = FEDE + FMDM
```

```
          CFZA = RJ0**2/GAMMA2*(X2*((1.+1./U02)/(2.*K0*U0)+(EPSR/U12
     &           +1./MUR)*(SH24-D2)+FEDE*(EPSR*FEDE/U12+FMDM/MUR)*(SH24
     &           +D2)+CH24*(2.*EPSR*FEDE/U12+FEDM/MUR)+EPSR/(2.*K0*DE)
     &           *(EPSR/(U0*DE)+U1/DM))+GAMMA2/(2.*K0*U0)+FMDM/MUR
     &           *(X2*FEDE+EPSA*FMDM)*(SH24-D2)+GAMMA2/MUR*(SH24
     &           +D2)+CH24/MUR*(X2*FEDM+2.*EPSA*FMDM)+U1/(2.*K0
     &           *DM)*(EPSR*X2/DE+EPSA*U1/(U0*DM)))

          END IF
          RETURN
          END

*******************************************************************************

%     Insert SUBROUTINE J0(ARG, RJ0) from program PIAA here!

*******************************************************************************

%     Insert COMPLEX*16 FUNCTION CSINT(CF, AY, BY, RELERS) from
%     program PIAA here!

*******************************************************************************
```

D.12 Program SP

PROGRAM SP

```
************************************************************************
*                                                                      *
*      This is the last program in the series.  It combines all of     *
*      the previous results and calculates the scattering parameters.  *
*                                                                      *
************************************************************************


      PROGRAM     SP
      IMPLICIT    COMPLEX*16 (C)
      IMPLICIT    REAL*8 (A, B, D - H, O - Z)
      REAL*8      MUR, NRMM, NRMA
      COMMON      PI

      OPEN (1, FILE = '../computed.data/IMM')
      OPEN (2, FILE = '../computed.data/IAA')
      OPEN (3, FILE = '../computed.data/IAM')
      OPEN (4, FILE = '../computed.data/IAG')
      OPEN (5, FILE = '../computed.data/IMI')
      OPEN (7, FILE = '../computed.data/Zstrip')
      OPEN (8, FILE = '../computed.data/Zslot')
      OPEN (9, FILE = '../computed.data/SPout')
      OPEN (10, FILE = '../defined.data/phys.param')

      READ (10,*) FREQ, D, EPSR, MUR, WM, EPSM, WA, EPSA
      READ (1,*) CIMM
      READ (2,*) CIAA
      READ (3,*) CIAM
      READ (4,*) CIAG, CIAGP
      READ (5,*) CIMI
      READ (7,*) ZSTRIP
      READ (8,*) ZSLOT

    1 FORMAT (1X, A6, F9.4, ' AT ', F9.4, ' DEGREES')

      PI = 3.14159265358979

      WRITE (6,*) ' '
      WRITE (6,*) 'FREQUENCY =', FREQ
      WRITE (6,*) 'SUBSTRATE THICKNESS =', D
      WRITE (6,*) 'RELATIVE PERMITTIVITY =', EPSR
      WRITE (6,*) 'RELATIVE PERMEABILITY =', MUR
      WRITE (6,*) 'MICROSTRIP WIDTH =  ', WM
      WRITE (6,*) 'MICROSTRIP EFFECTIVE DIELECTRIC CONSTANT =', EPSM
      WRITE (6,*) 'SLOTLINE WIDTH =  ', WA
      WRITE (6,*) 'SLOTLINE EFFECTIVE DIELECTRIC CONSTANT =', EPSA

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*     THE MIRCOSTRIP NORM WHEN INCIDENT MODE CURRENT EQUALS 1 AMP IS   *
*     SIMPLY THE MICROSTRIP CHARACTERISTIC IMPEDANCE.  THE SLOTLINE    *
*     NORM  WHEN SLOT VOLTAGE OF INCIDENT MODE IS SET EQUAL TO         *
*     1 VOLT IS  GIVEN BY THE INVERSE OF THE SLOTLINE CHARACTERISTIC   *
*     IMPEDANCE.                                                       *
*                                                                      *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      NRMM = ZSTRIP
      NRMA = 1./ZSLOT
```

```
      CD = 1./(CIAM**2+CIMM*CIAA)

      CS11 = -CIMM*CIAG**2*CD/(2.*NRMM)
      S11M = ZABS(CS11)
      S11P = PHASE(CS11)
      WRITE (6,*) ' '
      WRITE (6,*) 'S11 (real, imag) = ', CS11
      WRITE (6,*) 'S11 (mag, phase in degrees) = ', S11M, S11P
      WRITE (9,1) 'S11 = ', S11M, S11P

      CS21 = (1.,0.) - CS11
      S21M = ZABS(CS21)
      S21P = PHASE(CS21)
      WRITE (6,*) ' '
      WRITE (6,*) 'S21 (real, imag) = ', CS21
      WRITE (6,*) 'S21 (mag, phase in degrees) = ', S21M, S21P
      WRITE (9,1) 'S21 = ', S21M, S21P

      CS33 = CIAA*CIMI**2*CD/(2.*NRMA)
      S33M = ZABS(CS33)
      S33P = PHASE(CS33)
      WRITE (6,*) ' '
      WRITE (6,*) 'S33 (real, imag) = ', CS33
      WRITE (6,*) 'S33 (mag, phase in degrees) = ', S33M, S33P
      WRITE (9,1) 'S33 = ', S33M, S33P

      CS43 = (1.,0.) + CS33
      S43M = ZABS(CS43)
      S43P = PHASE(CS43)
      WRITE (6,*) ' '
      WRITE (6,*) 'S43 (real, imag) = ', CS43
      WRITE (6,*) 'S43 (mag, phase in degrees) = ', S43M, S43P
      WRITE (9,1) 'S43 = ', S43M, S43P

      CS41 = (-CIAM*CIAG*CIMI*CD+CIAGP)*DSQRT(ZSLOT/ZSTRIP)/2.
      S41M = ZABS(CS41)
      S41P = PHASE(CS41)
      WRITE (6,*) ' '
      WRITE (6,*) 'S41 (real, imag) = ', CS41
      WRITE (6,*) 'S41 (mag, phase in degrees) = ', S41M, S41P
      WRITE (9,1) 'S41 = ', S41M, S41P

      U1  = DREAL(2.*CS41*DCONJG(CS41) + CS11*DCONJG(CS11) +
     &            CS21*DCONJG(CS21))
      U2  = DREAL(2.*CS41*DCONJG(CS41) + CS33*DCONJG(CS33) +
     &            CS43*DCONJG(CS43))
      CU3 = DCONJG(CS41)*(CS11-CS21) + CS41*(DCONJG(CS33+CS43))

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*     The above three quantities should be as follows when            *
*     loss is negligible for the crossover:  U1 = U2 = 1, CU3 = 0.    *
*     The accuracy of the S-matrix may be ascertained by comparing    *
*     the actual values of U1, U2, and CU3 to the above results which *
*     would hold for a unitary matrix (no radiation loss or other).   *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      WRITE (6,*) ' '
      WRITE (6,*) 'U1 = ', U1, 'U2 = ', U2
      WRITE (6,*) 'CU3 = ', CU3
      WRITE (9,*) 'U1 = ', U1, 'U2 = ', U2
      WRITE (9,*) 'CU3 = ', CU3
```

```
      CLOSE (1)
      CLOSE (2)
      CLOSE (3)
      CLOSE (4)
      CLOSE (5)
      CLOSE (7)
      CLOSE (8)
      CLOSE (9)
      CLOSE (10)
   99 STOP
      END

*************************************************************************

      REAL*8      FUNCTION PHASE(Z)
      IMPLICIT    REAL*8   (A, B, D - H, O - Z)
      COMPLEX*16  Z
      COMMON      PI

      ZR = DREAL(Z)
      ZI = DIMAG(Z)

      IF (ZR .EQ. 0.) THEN

         IF (ZI .GT. 0.) THEN
            PHASE = 90.
         ELSE IF (ZI .LT. 0.) THEN
            PHASE = -90.
         ELSE
            PHASE = 0.
         END IF

      ELSE IF (ZR .GT. 0.) THEN
         PHASE = DATAN(ZI/ZR)*180./PI
      ELSE IF (ZI .GE. 0.) THEN
         PHASE = DATAN(ZI/ZR)*180./PI + 180.
      ELSE
         PHASE = DATAN(ZI/ZR)*180./PI - 180.
      END IF
      RETURN
      END
```