ISBN 0-315-53287-4

Field-Theory Based CAD Procedure for Millimeter Wave Integrated Circuits

by

Poman Pok-Man So

A thesis presented to
The School of Graduate Studies and Research
of the University of Ottawa
in partial fulfillment of the requirement
for the degree of
Master of Applied Science in Electrical Engineering

Contents

Page

Thesis Contributions

Commercially available microwave CAD programs such as Touchstone™ Senior and Super-Compact™ do not include E-plane circuit elements. Although these CAD programs allow users to define their own models for circuit design, implementation of E-plane circuit elements is not an easy task because of the large number of variable parameters that must be included in the analytical expressions. Such formulas are often inaccurate, long to compute and restricted in range. The contributions of this thesis are:

(1) A detailed formulation for the spectral domain analysis of antipodal finline is presented.

(2) A novel CAD procedure of E-plane circuits has been developed which is based on discontinuity models and field theory based lookup tables.

(3) An innovative interpolation technique based on physically realistic functions is developed to extract data from lookup tables.

(4) Some finline circuit elements are modeled with the commercially available CAD program Touchstone™ Senior; thus taking full advantage of all capabilities inherent in it.

This CAD procedure expands the power of commercially available microwave CAD software into the millimeter wave circuit domain and replaces the conventional trial-and-error design method for E-plane circuits with a systematic design and optimization procedure.

Acknowledgments

I wish to thank my supervisor Dr. W. J. R. Hoefer for his endless encouragement and financial support. He gave me close supervision and invaluable suggestions throughout the development of this CAD procedure. He encouraged me to submit the CAD procedure of this thesis to the IEEE MTT symposium in New York, May 1988. The paper was accepted and we demonstrated this CAD procedure in the symposium.

I want to thank all persons who were involved in the development of two Spectral Domain programs, *UniFast.Pas* and *BiFast.Pas*. I modified those programs to became *UniFastS.Pas* and *BiFastS.Pas* for lookup table generations. Thanks are also due to Dr. P. Saguet for allowing me to use his unpublished shunt discontinuity model, Mr. C. Verver from the Communications Research Centre in Ottawa for the measurement data on two finline filters, and Mr. Alain Dugas for proofreading and correcting the manuscript of this thesis. In addition, I am indebted to EEsof Inc. for providing a free version of their Touchstone™ Senior software. This project was funded by the Natural Science and Engineering Research Council of Canada.

I must also say thanks to all the professors in this laboratory who helped me directly and indirectly during my studies here, and my fiancée Joanne Cheung who helped me in preparing the final version of this thesis and the thesis defence transpancies. Last but not least, I would like to thank my parents who supported my expensive education in Canada.

Poman So

Abstract

A novel CAD procedure for E-plane circuits is developed. This CAD procedure employs field-theory based lookup tables and equivalent circuit discontinuity models. An innovative interpolation scheme which is based on physically realistic functions is also implemented. This interpolation method reduces the size of the lookup tables considerably since only three frequency points are required to characterize a given finline geometry over the entire operating frequencies range. A straightforward linear interpolation scheme, which requires larger lookup tables, is also implemented for the cases in which the new interpolation scheme is not applicable.

Field-theory based lookup table generation programs are among the building blocks of this CAD procedure. Because the spectral domain method is an efficient algorithm for calculating the propagation characteristics of E-plane structrues, it is chosen to perform the table-generation task. The application of the spectral domain method to E-plane structures, especially to antipodal finline, is discussed.

The accuracy of the implemented E-plane circuit elements has been verified by measurement. The simulation results are in good agreement with the experimental results and confirm the validity of the computer models.

A simple, but detailed, CAD example which uses all the programs developed for this CAD procedure is presented. It is followed by a number of CAD and experimental results for transformers, tapers and bandpass filters. Some CAD results obtained by the two interpolation methods are compared. Limitations of the implemented E-plane circuit elements and of this CAD procedure are discussed before the conclusion.

All programs, Touchstone™ Senior circuit files, measurement result and related materials are provided in the appendices.

CHAPTER 0

With the increasing interest in millimeter wave circuits, E-plane passive circuit components start playing important roles such as filtering and impedance matching. Therefore, development of efficient millimeter wave CAD software is important.

Conventional microwave CAD programs depend heavily on closed form empirical expressions obtained by curve fitting. This is especially true in the microstrip circuit CAD domain. Despite the simple geometry of microstrip circuit elements and the quasi-TEM nature of the associated electromagnetic fields, the corresponding empirical expressions are very complicated. These expressions give no insight to the physical behaviour of the circuit components and are also computer-time expensive.

E-plane circuit components (see Figure 0-1) have an even more complicated geometry than mircostrip components [1]. The complicated structures impose stringent boundary conditions that can only be satisfied by complicated field solutions. The variable parameters involved in E-plane circuit elements are the inner dimension of the waveguide housing (a, b), substrate thickness and permittivity (s, ϵ_r), slot width (d), lateral position of the fins (h) and frequency (f). It is very impractical, if not impossible, to develop closed-form expressions for these structures because too many variable parameters must be included in the empirical expressions. Such formulas are often inaccurate, long to compute and restricted in range. As Jansen [2] has pointed out for the case of monolithic circuit computer-aided design, programs that are based on field-theory (such as the Finite Element Methods—FEM, the Transmission Line Matrix—TLM and the Spectral Domain Methods—SDM) are much more accurate and flexible for modelling planar lines and circuit elements.

Commercially available CAD software such as Touchstone™ Senior and Super-Compact™ allow users to define circuit models. However, field-theory based numerical programs are too slow to be used directly to implement circuit models because circuit optimization requires a large number of analyses. In this thesis, a novel CAD procedure that can take advantage of both the accuracy of field-theoretical tools as well as the capabilities (such as analysis, optimization, graphics and components integration) of commercially available CAD software is developed for millimeter wave circuits.

This novel CAD procedure relies on simple discontinuity models and field-theory based lookup tables which give the effective dielectric constant (ϵ_{eff}) and effective characteristic impedance (Z_{eff}) of the transmission medium such as unilateral and bilateral finlines†. In order to minimize the size of the lookup tables, a new interpolation scheme

---

†

$$\epsilon_{eff} = \left(\frac{\lambda}{\lambda_g}\right)^2$$

Where λ_g is the wavelength in longitudinal direction in the transmission line. This which is based on physically realistic functions is developed. These tools are the basis for the finline circuit models of this CAD procedure. The finline elements are integrated with Touchstone™ Senior and are used to design some finline circuit components. The CAD results are found to be in good agreement with experimental results. Hence this CAD procedure expands the power of commercially available microwave CAD software into the millimeter wave circuit domain and replaces the traditional trial-and-error design method with a systematic design procedure.

There is only a relatively small number of E-plane element discontinuity models available in the literature. Among them are the step and the shunt discontinuity models for finlines. These types of discontinuity elements are found in transformer, taper, and filter circuit components, and are therefore considered in this thesis.

The shunt discontinuity element shown in Figure 0-2(a) is modelled by Saguet and Coumes [3] as a shunt inductance at the middle of a short transmission line having an effective dielectric constant ϵ_{eff} and an effective characteristic impedance Z_{eff}. The inductance and the length of the transmission line are functions of a, b, w, h, ϵ_{eff}, Z_{eff} and $frequency$. The step discontinuity element in Figure 0-2(b) is modelled by Tsui and Hoefer [4] as a step inductance which is a function of $a, b, d_1, d_2, \epsilon_{1eff}, \epsilon_{2eff}, Z_{1eff}, Z_{2eff}$ and $frequency$. The dimensions which are not found in Figure 0-2 are defined in Figure 0-1.

The empirical expressions for these discontinuity models are not computer-time expensive. However it is quite a task to find the effective dielectric constant and effective characteristic impedance at various frequencies and gapwidths. This task is one of the main subjects covered in this thesis.

---

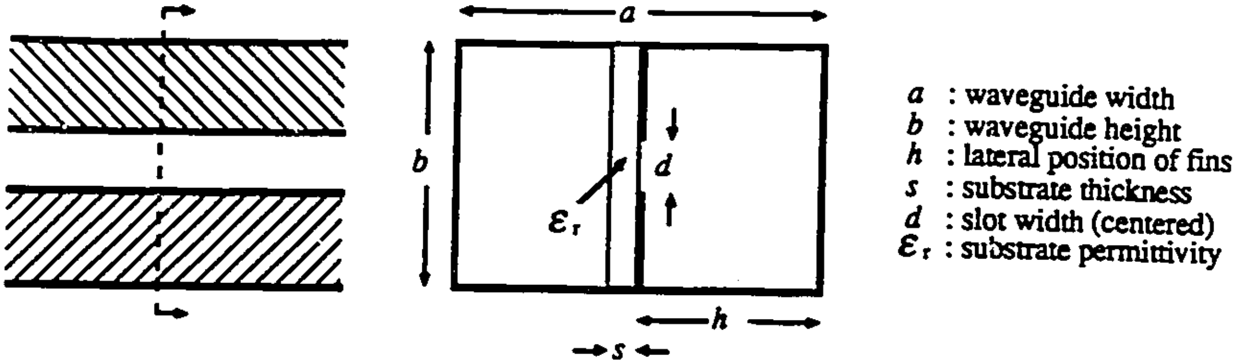
definition has been adopted widely for non-TEM E-plane structures and will be used throughout this thesis.

For application to E-plane structures, one of the following three impedance definitions is usually used:

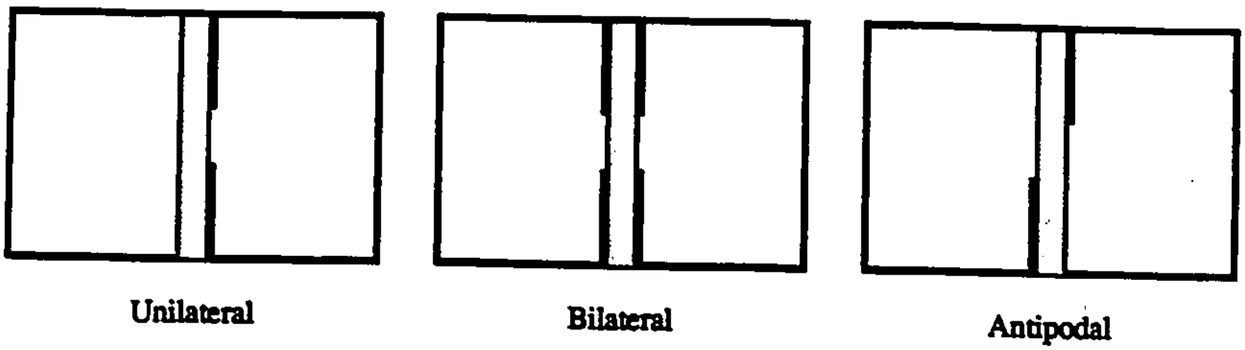
$$Z_o(V,I) = V/I \qquad \text{(voltage-current impedance)}$$
$$Z_o(V,P) = V^2/(2P) \qquad \text{(voltage-power impedance)}$$
$$Z_o(P,I) = 2P/I^2 \qquad \text{(power-current impedance)}$$

In finline structures, the voltage-power impedance definition is usually chosen. In this thesis, effective characteristic impedance (Z_{eff}) means $Z_o(V,P)$ at the frequency of interest.

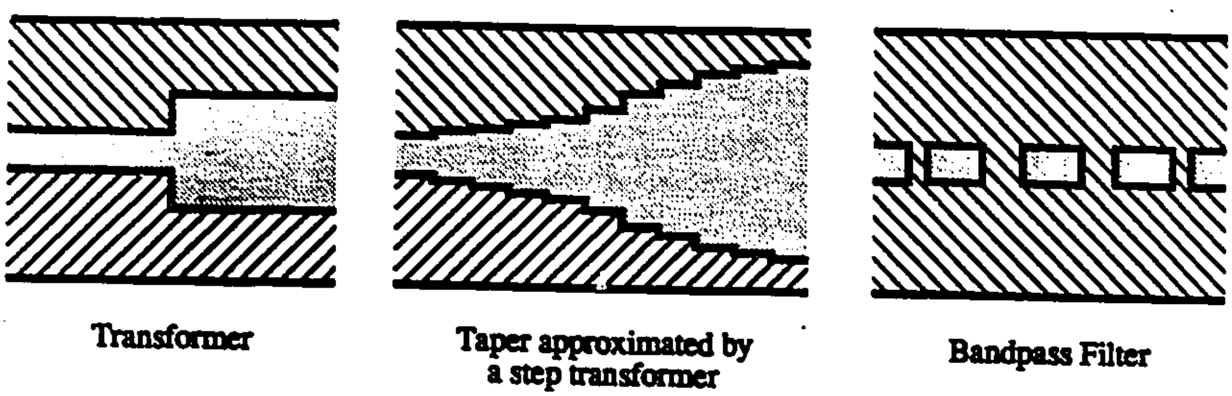
Reference: ELG7100B course note (term 1987-1988).



(a): Unilateral finline sectional views.

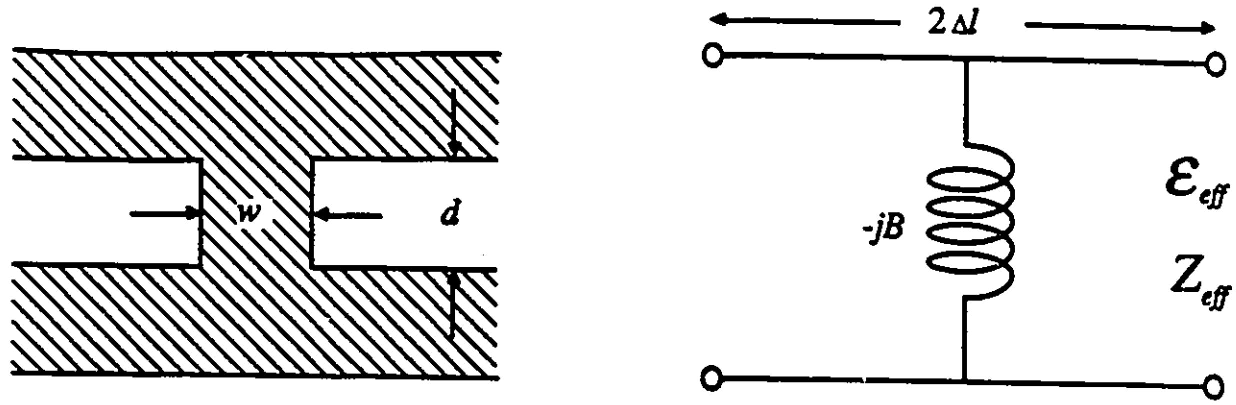


(b): Finline configurations.

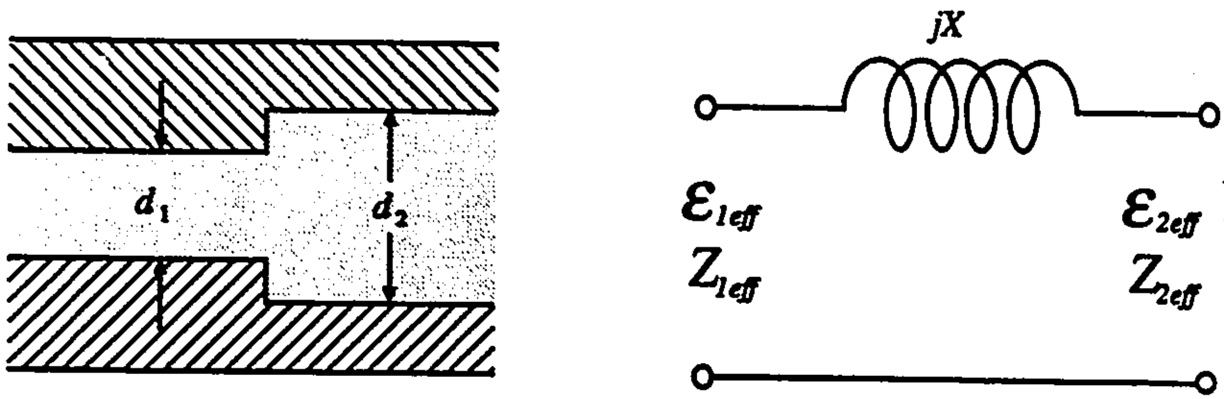


(c): Finline components.

Figure 0-1: Some finline configurations and circuit components.



(a): Finline shunt discontinuity and it equivalent model



(b): Finline step discontinuity and its equivalent model

Figure 0-2: Two discontinuity models for finline.

CHAPTER 1 The Applications of SDA to E-Plane Structures

The spectral domain analysis (SDA) is by far the most efficient numerical method for the computation of E-Plane line characteristics. It has therefore been selected for the generation of lookup tables and forms an integral part of the CAD procedure presented in this thesis. A very successful formulation of the spectral domain method in terms of the equivalent transmission line concept is due to Itoh and Schmidt [10]. This approach will be used to explain the application of the SDA to E-plane structures in the following sections.

The application of SDA to unilateral and bilateral finlines has been discussed extensively in the literature (It is not practical to list all references on this subject. Interested readers should consult the references [5] and [10]–[12]). Therefore the following discussion concentrates on the application of SDA to antipodal finline for which no explicit spectral domain formulation has been published so far.

An antipodal finline cross-section is shown in Figure 1-1. It has a rectangular waveguide enclosure with inner dimensions a and b. The substrate of thickness s and permittivity ϵ_r bridges the broad wall at the middle and carries metal fins of zero thickness on both sides, a valid approximation in most applications. The distance along x between the fin edges is d. It becomes negative in the case of overlap as in Figure 1-1.

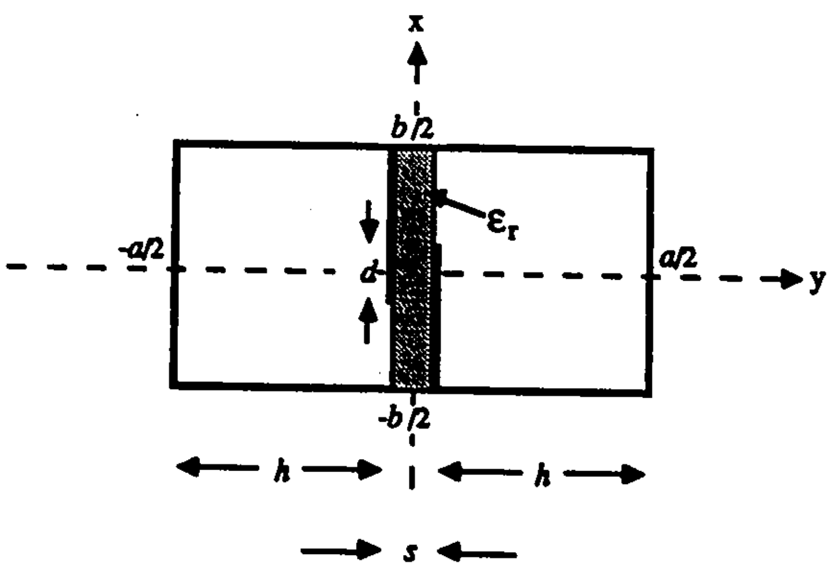


Figure 1-1: Antipodal finline cross-section.

The structure in Figure 1-1 can be analysed by using a superposition of LSE (E_y=0) and LSM (H_y=0) waves at the operating frequency f. LSE and LSM waves are TE and TM waves with respect to the y-axis. Once the corresponding longitudinal components are known, the other field components can be obtained using Maxwell's equations. The goal of the following analysis is to obtain the solution for the field components that will fulfill all boundary and interface conditions and to obtain the propagation constant in the z-direction at the operating frequency. In order to eliminate the x-dependence of the fields, all field components are Fourier transformed with respected to the x-direction. The Fourier transform of the y-components of the field quantities, E_y for the LSM-mode and H_y for the LSE-mode, are:

$$\widetilde{E}_y(\alpha, y) = \int_{-\infty}^{+\infty} E_y(x, y)\; e^{j\alpha x}\; dx \tag{1}$$

$$\widetilde{H}_y(\alpha, y) = \int_{-\infty}^{+\infty} H_y(x, y)\; e^{j\alpha x}\; dx \tag{2}$$

Because the housing is finite in the x-direction, the Fourier variable α becomes discrete: $\alpha = \alpha_n = \frac{n\pi}{b}$, where $n = ..., -1, 0, 1, 2, ...$, which implies that $\widetilde{E}_y(\alpha_n, y)$ and $\widetilde{H}_y(\alpha_n, y)$ are the coefficients of two Fourier series:

$$E_y(x, y) = b \cdot \sum_{n=-\infty}^{+\infty} \widetilde{E}_y(\alpha_n, y)\; e^{-j\alpha_n x} \tag{3}$$

$$H_y(x, y) = b \sum_{n=-\infty}^{+\infty} \widetilde{H}_y(\alpha_n, y)\; e^{-j\alpha_n x} \tag{4}$$

Where

$$\widetilde{E}_y(\alpha_n, y) = \int_{-\frac{b}{2}}^{+\frac{b}{2}} E_y(x, y)\; e^{j\alpha_n x}\; dx$$

$$\widetilde{H}_y(\alpha_n, y) = \int_{-\frac{b}{2}}^{+\frac{b}{2}} H_z(x, y)\; e^{j\alpha_n x}\; dx$$

(3) and (4) can be rewritten as

$$E_y(x, y)\; e^{-j\beta z} = b \sum_{n=-\infty}^{+\infty} \widetilde{E}_y(\alpha_n, y)\; e^{-j(\alpha_n x + \beta z)} \tag{5}$$

$$H_y(x, y)\; e^{-j\beta z} = b \sum_{n=-\infty}^{+\infty} \widetilde{H}_y(\alpha_n, y)\; e^{-j(\alpha_n x + \beta z)} \tag{6}$$

Equation (5) and (6) suggests that the field in the structure is a superposition of plane waves, inhomogeneous with respect to the y-direction, and propagating in the $(\alpha_n x + \beta z)$-direction. A new co-ordinate system (u, v) such that u points in the direction of propagation of the wave and v is orthogonal to u can be defined as in Figure 1-2.

$$u = x\cos(\theta) + z\sin(\theta) = xN_x + zN_z \tag{7}$$
$$v = z\cos(\theta) - x\sin(\theta) = zN_x - xN_z \tag{8}$$

where

$$N_x = \cos(\theta) = \frac{\alpha_n}{\sqrt{\alpha_n^2 + \beta^2}}$$
$$N_z = \sin(\theta) = \frac{\beta}{\sqrt{\alpha_n^2 + \beta^2}}$$

and

$$N_x^2 + N_z^2 = \cos^2(\theta) + \sin^2(\theta) = 1$$

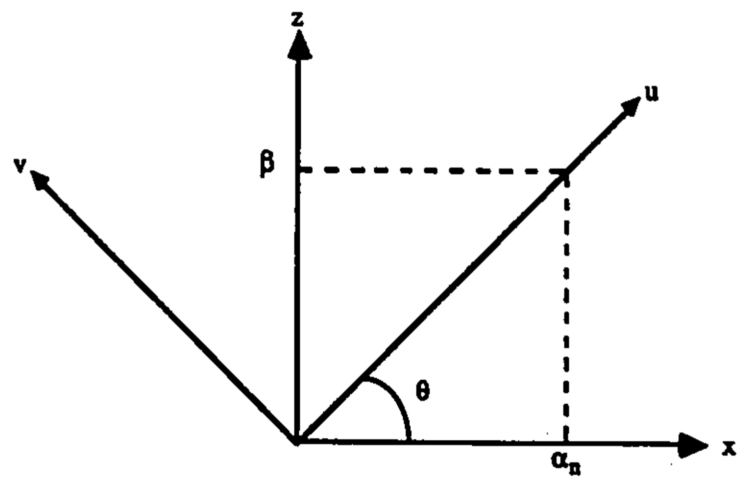


Figure 1-2: Relationship between (x, y) and (u, v) co-ordinate systems.

Note that for each α_n, u points to a different direction in the (x, z)-plane. But for each α_n, the transformed field depends only on u and y; i.e. $\partial/\partial v = 0$. The LSE (TE-to-y) wave has components $\tilde{E}_v, \tilde{H}_y$ and $\tilde{H}_u$ while the LSM (TM-to-y) wave has components $\tilde{E}_y, \tilde{H}_v$ and $\tilde{E}_u$. According to Schmidt and Itoh [11] the transverse condition can be described by two transmission line equivalent circuits as shown in Figure 1-3. The "voltages" $\tilde{E}_u, \tilde{E}_v$ and the "currents" $\tilde{H}_u, \tilde{H}_v$ in the equivalent circuits are related through the wave impedances:

$$Z_{LSE,i} = \left|\frac{\tilde{E}_{v,i}}{\tilde{H}_{u,i}}\right| = \frac{j\omega\mu}{\gamma_i} = \frac{1}{Y_{LSE,i}} \tag{9}$$
$$Z_{LSM,i} = \left|\frac{\tilde{E}_{u,i}}{\tilde{H}_{v,i}}\right| = \frac{\gamma_i}{j\omega\epsilon_o\epsilon_{r,i}} = \frac{1}{Y_{LSM,i}} \tag{10}$$

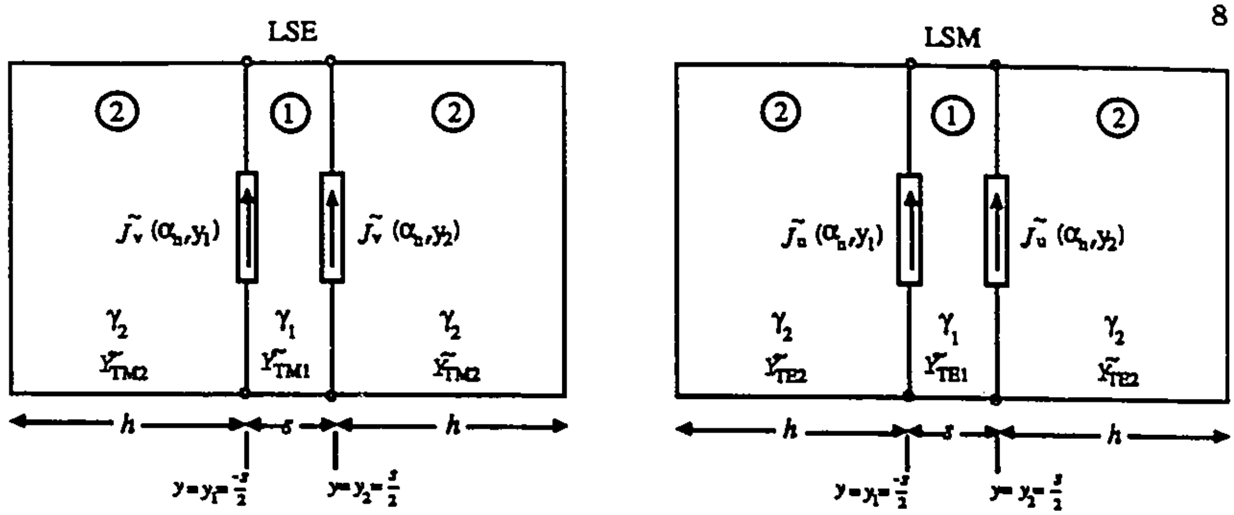


Figure 1-3: Transmission line equivalent circuit at transverse resonance.

where

$$\gamma_i = \sqrt{\alpha_n^2 + \beta^2 - \epsilon_{r,i}k_o^2}$$
$$i \in \{1 : \text{substrate}, 2 : \text{air}\}$$

In order to satisfy the boundary condition at the substrate-to-air interface, the tangential electric field must be continuous and the change in tangential magnetic field must be equal to the current density in the interface. These conditions are satisfied by injecting currents of intensities $\tilde{J}_v(\alpha_n, y_1)$, $\tilde{J}_v(\alpha_n, y_2)$, $\tilde{J}_u(\alpha_n, y_1)$ and $\tilde{J}_u(\alpha_n, y_2)$ as shown in Figure 1-3. In the spectral domain the electric field and the current density are related through the following formulae:

$$\tilde{J}_v(\alpha_n, y_1) = \tilde{Y}^e_{11}\tilde{E}_v(\alpha_n, y_1) + \tilde{Y}^e_{12}\tilde{E}_v(\alpha_n, y_2)$$
$$\tilde{J}_u(\alpha_n, y_1) = \tilde{Y}^h_{11}\tilde{E}_u(\alpha_n, y_1) + \tilde{Y}^h_{12}\tilde{E}_u(\alpha_n, y_2)$$
$$\tilde{J}_v(\alpha_n, y_2) = \tilde{Y}^e_{21}\tilde{E}_v(\alpha_n, y_1) + \tilde{Y}^e_{22}\tilde{E}_v(\alpha_n, y_2)$$
$$\tilde{J}_u(\alpha_n, y_2) = \tilde{Y}^h_{21}\tilde{E}_u(\alpha_n, y_1) + \tilde{Y}^h_{22}\tilde{E}_u(\alpha_n, y_2)$$

or in matrix notation,

$$\begin{bmatrix} \tilde{J}_v(\alpha_n, y_1) \\ \tilde{J}_u(\alpha_n, y_1) \\ \tilde{J}_v(\alpha_n, y_2) \\ \tilde{J}_u(\alpha_n, y_2) \end{bmatrix} = \begin{bmatrix} \tilde{Y}^e_{11} & 0 & \tilde{Y}^e_{12} & 0 \\ 0 & \tilde{Y}^h_{11} & 0 & \tilde{Y}^h_{12} \\ \tilde{Y}^e_{21} & 0 & \tilde{Y}^e_{22} & 0 \\ 0 & \tilde{Y}^h_{21} & 0 & \tilde{Y}^h_{22} \end{bmatrix} \times \begin{bmatrix} \tilde{E}_v(\alpha_n, y_1) \\ \tilde{E}_u(\alpha_n, y_1) \\ \tilde{E}_v(\alpha_n, y_2) \\ \tilde{E}_u(\alpha_n, y_2) \end{bmatrix} \qquad (11)$$

Where $\tilde{Y}^e_{11}$ is the driving point admittance at $y = y_1$. $\tilde{Y}^e_{12}$ is the transfer admittance which expresses the contribution of the source at $y = y_2$ to the current density at $y = y_1$. Other

Other quantities may be similarly defined. From transmission line theory and Figure 1-3

$$\widetilde{Y}_2^e = \widetilde{Y}_{TM2} coth(\gamma_2 h)$$
$$\widetilde{Y}_1^e = \widetilde{Y}_{TM1} \frac{\widetilde{Y}_{TM1} + \widetilde{Y}_2^e coth(\gamma_1 s)}{\widetilde{Y}_2^e + \widetilde{Y}_{TM1} coth(\gamma_1 s)}$$
$$\widetilde{Y}_{11}^e = \widetilde{Y}_1^e + \widetilde{Y}_2^e$$
$$\widetilde{Y}_{12}^e = \frac{\widetilde{J}_v(y_2)}{\widetilde{E}_{v_{12}}(y_1)} = \frac{(\widetilde{Y}_1^e + \widetilde{Y}_2^e)(\widetilde{Y}_2^e sinh(\gamma_1 s) + \widetilde{Y}_{TM1} cosh(\gamma_1 s))}{\widetilde{Y}_{TM1}}$$

Where $\widetilde{E}_{v_{12}}(y_1)$ is the field at y_1 due to the current density at y_2. Because of symmetry:

$$\widetilde{Y}_{22}^e = \widetilde{Y}_{11}^e$$
$$\widetilde{Y}_{21}^e = \widetilde{Y}_{12}^e$$

$\widetilde{Y}_{11}^h, \widetilde{Y}_{12}^h, \widetilde{Y}_{21}^h$ and $\widetilde{Y}_{22}^h$ are similar to $\widetilde{Y}_{11}^e, \widetilde{Y}_{12}^e, \widetilde{Y}_{21}^e$ and $\widetilde{Y}_{22}^e$; the difference is that all $\widetilde{Y}_{TM1}$ and $\widetilde{Y}_{TM2}$ must be changed to $\widetilde{Y}_{TE1}$ and $\widetilde{Y}_{TE2}$ respectively. Applying equations (7) and (8) to (11) gives:

$$\begin{bmatrix} \widetilde{J}_z(\alpha_n, y_1) \\ \widetilde{J}_x(\alpha_n, y_1) \\ \widetilde{J}_z(\alpha_n, y_2) \\ \widetilde{J}_x(\alpha_n, y_2) \end{bmatrix} = \begin{bmatrix} \widetilde{Y}_{zz}^{11} & \widetilde{Y}_{zx}^{11} & \widetilde{Y}_{zz}^{12} & \widetilde{Y}_{zx}^{12} \\ \widetilde{Y}_{xz}^{11} & \widetilde{Y}_{xx}^{11} & \widetilde{Y}_{xz}^{12} & \widetilde{Y}_{xx}^{12} \\ \widetilde{Y}_{zz}^{21} & \widetilde{Y}_{zx}^{21} & \widetilde{Y}_{zz}^{22} & \widetilde{Y}_{zx}^{22} \\ \widetilde{Y}_{xz}^{21} & \widetilde{Y}_{xx}^{21} & \widetilde{Y}_{xz}^{22} & \widetilde{Y}_{xx}^{22} \end{bmatrix} \times \begin{bmatrix} \widetilde{E}_z(\alpha_n, y_1) \\ \widetilde{E}_x(\alpha_n, y_1) \\ \widetilde{E}_z(\alpha_n, y_2) \\ \widetilde{E}_x(\alpha_n, y_2) \end{bmatrix} \quad (12)$$

where

$$\begin{bmatrix} \widetilde{Y}_{zz}^{11} & \widetilde{Y}_{zx}^{11} & \widetilde{Y}_{zz}^{12} & \widetilde{Y}_{zx}^{12} \\ \widetilde{Y}_{xz}^{11} & \widetilde{Y}_{xx}^{11} & \widetilde{Y}_{xz}^{12} & \widetilde{Y}_{xx}^{12} \\ \widetilde{Y}_{zz}^{21} & \widetilde{Y}_{zx}^{21} & \widetilde{Y}_{zz}^{22} & \widetilde{Y}_{zx}^{22} \\ \widetilde{Y}_{xz}^{21} & \widetilde{Y}_{xx}^{21} & \widetilde{Y}_{xz}^{22} & \widetilde{Y}_{xx}^{22} \end{bmatrix} =$$

$$\begin{bmatrix} (N_x^2 \widetilde{Y}_{11}^h + N_z^2 \widetilde{Y}_{11}^e) & N_x N_z (\widetilde{Y}_{11}^h - \widetilde{Y}_{11}^e) & (N_x^2 \widetilde{Y}_{12}^h + N_z^2 \widetilde{Y}_{12}^e) & N_x N_z (\widetilde{Y}_{12}^h - \widetilde{Y}_{12}^e) \\ N_x N_z (\widetilde{Y}_{11}^h - \widetilde{Y}_{11}^e) & (N_z^2 \widetilde{Y}_{11}^h + N_x^2 \widetilde{Y}_{11}^e) & N_x N_z (\widetilde{Y}_{12}^h - \widetilde{Y}_{12}^e) & (N_z^2 \widetilde{Y}_{12}^h + N_x^2 \widetilde{Y}_{12}^e) \\ (N_x^2 \widetilde{Y}_{21}^h + N_z^2 \widetilde{Y}_{21}^e) & N_x N_z (\widetilde{Y}_{21}^h - \widetilde{Y}_{21}^e) & (N_x^2 \widetilde{Y}_{22}^h + N_z^2 \widetilde{Y}_{22}^e) & N_x N_z (\widetilde{Y}_{12}^h - \widetilde{Y}_{22}^e) \\ N_x N_z (\widetilde{Y}_{21}^h - \widetilde{Y}_{21}^e) & (N_z^2 \widetilde{Y}_{21}^h + N_x^2 \widetilde{Y}_{21}^e) & N_x N_z (\widetilde{Y}_{22}^h - \widetilde{Y}_{22}^e) & (N_z^2 \widetilde{Y}_{22}^h + N_x^2 \widetilde{Y}_{22}^e) \end{bmatrix}$$

Because

$$\widetilde{Y}^k_{ij} = \widetilde{Y}^k_{ji} \qquad \text{for } i \neq j, k = e \text{ or } h$$

$$\widetilde{Y}^k_{11} = \widetilde{Y}^k_{22}$$

Therefore

$$\begin{bmatrix} \widetilde{Y}^{11}_{zz} & \widetilde{Y}^{11}_{zx} & \widetilde{Y}^{12}_{zz} & \widetilde{Y}^{12}_{zx} \\ \widetilde{Y}^{11}_{xz} & \widetilde{Y}^{11}_{xx} & \widetilde{Y}^{12}_{xz} & \widetilde{Y}^{12}_{xx} \\ \widetilde{Y}^{21}_{zz} & \widetilde{Y}^{21}_{zx} & \widetilde{Y}^{22}_{zz} & \widetilde{Y}^{22}_{zx} \\ \widetilde{Y}^{21}_{xz} & \widetilde{Y}^{21}_{xx} & \widetilde{Y}^{22}_{xz} & \widetilde{Y}^{22}_{xx} \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ B & E & D & F \\ C & D & A & B \\ D & F & B & E \end{bmatrix}$$

Where

$$A = (N_x^2 \widetilde{Y}^h_{11} + N_z^2 \widetilde{Y}^e_{11})$$
$$B = N_x N_z (\widetilde{Y}^h_{11} - \widetilde{Y}^e_{11})$$
$$C = (N_x^2 \widetilde{Y}^h_{12} + N_z^2 \widetilde{Y}^e_{12})$$
$$D = N_x N_z (\widetilde{Y}^h_{12} - \widetilde{Y}^e_{12})$$
$$E = (N_z^2 \widetilde{Y}^h_{11} + N_x^2 \widetilde{Y}^e_{11})$$
$$F = (N_z^2 \widetilde{Y}^h_{12} + N_x^2 \widetilde{Y}^e_{12})$$

Note that $\widetilde{J}_z(\alpha_n, y_i), \widetilde{J}_x(\alpha_n, y_i)$ and $\widetilde{E}_z(\alpha_n, y_i), \widetilde{E}_x(\alpha_n, y_i)$ are the Fourier transforms of the unknown current densities on the fins and the unknown fields in the slot; where i=1 corresponds to y=y_1 and i=2 corresponds to y=y_2. The elements of the Y-matrix in equation (12) are unknown because they contain the unknown constant β. However the orthogonal relationship between the electric field components and the current density components can be used to compute this unknown constant β. Mathematically speaking, the orthogonal relationship are:

$$\int_{-\frac{s}{2}}^{+\frac{s}{2}} J_z(x, y_1) E_z^*(x, y_1) dx = 0 \qquad (13)$$

$$\int_{-\frac{s}{2}}^{+\frac{s}{2}} J_x(x, y_1) E_x^*(x, y_1) dx = 0 \qquad (14)$$

$$\int_{-\frac{s}{2}}^{+\frac{s}{2}} J_z(x, y_2) E_z^*(x, y_2) dx = 0 \qquad (15)$$

$$\int_{-\frac{b}{2}}^{+\frac{b}{2}} J_z(x, y_2) E_z^*(x, y_2) dx = 0 \tag{16}$$

Let

$$E_z(x, y_1) = \sum_{r=1}^{R} a_r^{y_1} f_{z,r}^{y_1}(x)$$
$$E_x(x, y_1) = \sum_{s=1}^{S} b_s^{y_1} f_{x,s}^{y_1}(x)$$
$$E_z(x, y_2) = \sum_{r=1}^{R} a_r^{y_2} f_{z,r}^{y_2}(x)$$
$$E_x(x, y_2) = \sum_{s=1}^{S} b_s^{y_2} f_{x,s}^{y_2}(x)$$

Where $a_r^{y_1}, a_r^{y_2}, b_s^{y_1}$ and $b_s^{y_2}$ are unknown amplitude coefficients. Both sums must vanish on the metal surfaces. Therefore it is logical to choose $f_{z,r}^{y_1}(x), f_{z,r}^{y_2}(x), f_{x,s}^{y_1}(x)$ and $f_{x,s}^{y_2}(x)$ such that they have zero amplitude on the fins, i.e.

$$f_{z,r}^{y_1}(x) = f_{x,s}^{y_1}(x) = 0 \qquad -\frac{d}{2} < x \leq \frac{b}{2}$$
$$f_{z,r}^{y_2}(x) = f_{x,s}^{y_2}(x) = 0 \qquad -\frac{b}{2} < x \leq \frac{d}{2}$$

Because the structure is symmetrical with respect to the origin, $(x, y) = (0, 0)$, and the waves are propagating in the z-direction:

$$f_{z,r}^{y_2}(x) = f_{z,r}^{y_1}(-x)$$
$$f_{x,s}^{y_2}(x) = f_{x,s}^{y_1}(-x)$$

Therefore, the above set of equations can be rewritten as:

$$E_z(x, y_1) = \sum_{r=1}^{R} a_r f_{z,r}^{y_1}(x) = \sum_{r=1}^{R} a_r f_{z,r}(x)$$
$$E_x(x, y_1) = \sum_{s=1}^{S} b_s f_{x,s}^{y_1}(x) = \sum_{s=1}^{S} b_s f_{x,s}(x)$$
$$E_z(x, y_2) = \sum_{r=1}^{R} a_r f_{z,r}^{y_1}(-x) = \sum_{r=1}^{R} a_r f_{z,r}(-x)$$
$$E_x(x, y_2) = \sum_{s=1}^{S} b_s f_{x,s}^{y_1}(-x) = \sum_{s=1}^{S} b_s f_{x,s}(-x)$$

Note that the superscript, y_1, is omitted from the basis functions and it must be kept in mind that $f_{x,r}(x)$ and $f_{x,s}(x)$ will give the correct representation of the electric fields at $y=y_1$ (and $y=y_2$). Since $f_{x,r}(x)$ and $f_{x,s}(x)$ are real-and-even, their Fourier transforms are also real-and-even. In the spectral domain the above system of equations becomes:

$$\widetilde{E}_x(\alpha_n, y_1) = \sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) \tag{17}$$

$$\widetilde{E}_x(\alpha_n, y_1) = \sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) \tag{18}$$

$$\widetilde{E}_x(\alpha_n, y_2) = \sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) \tag{19}$$

$$\widetilde{E}_x(\alpha_n, y_2) = \sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n) \tag{20}$$

Substituting these equations into equation (12) gives:

$$\begin{bmatrix} \widetilde{J}_x(\alpha_n, y_1) \\ \widetilde{J}_x(\alpha_n, y_1) \\ \widetilde{J}_x(\alpha_n, y_2) \\ \widetilde{J}_x(\alpha_n, y_2) \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ B & E & D & F \\ C & D & A & B \\ D & F & B & E \end{bmatrix} \times \begin{bmatrix} \sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) \\ \sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) \\ \sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) \\ \sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n) \end{bmatrix} \tag{21}$$

or

$$\widetilde{J}_x(\alpha_n, y_1) = A\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) + B\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) + C\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) + D\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n)$$

$$\widetilde{J}_x(\alpha_n, y_1) = B\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) + E\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) + D\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) + F\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n)$$

$$\widetilde{J}_x(\alpha_n, y_2) = C\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) + D\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) + A\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) + B\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n)$$

$$\widetilde{J}_x(\alpha_n, y_2) = D\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(\alpha_n) + F\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(\alpha_n) + B\sum_{r=1}^{R} a_r \widetilde{f}_{x,r}(-\alpha_n) + E\sum_{s=1}^{S} b_s \widetilde{f}_{x,s}(-\alpha_n)$$

According to Parseval's theorem:

$$b \int_{-\frac{1}{2}}^{\frac{1}{2}} J(x) f^*(x) dx = \sum_{n=-\infty}^{+\infty} \widetilde{J}(\alpha_n) \widetilde{f}^*(\alpha_n) \tag{22}$$

If the above system of equations is multiplied by its corresponding orthogonal basis, and the spectral components are summed from $n = -\infty$ to $n = +\infty$, then the following system of equations is obtained (i.e. apply equations (13)–(16) to (21) and use Parseval's theorem (22)):

$$\sum_{r=1}^{R} a_r P^{11}_{r'r} + \sum_{s=1}^{S} b_s Q^{11}_{r's} + \sum_{r=1}^{R} a_r P^{12}_{r'r} + \sum_{s=1}^{S} b_s Q^{12}_{r's} = 0 \tag{23}$$

$$\sum_{r=1}^{R} a_r R^{11}_{s'r} + \sum_{s=1}^{S} b_s T^{11}_{s's} + \sum_{r=1}^{R} a_r R^{12}_{s'r} + \sum_{s=1}^{S} b_s T^{12}_{s's} = 0 \tag{24}$$

$$\sum_{r=1}^{R} a_r P^{21}_{r'r} + \sum_{s=1}^{S} b_s Q^{21}_{r's} + \sum_{r=1}^{R} a_r P^{22}_{r'r} + \sum_{s=1}^{S} b_s Q^{22}_{r's} = 0 \tag{25}$$

$$\sum_{r=1}^{R} a_r R^{21}_{s'r} + \sum_{s=1}^{S} b_s T^{21}_{s's} + \sum_{r=1}^{R} a_r R^{22}_{s'r} + \sum_{s=1}^{S} b_s T^{22}_{s's} = 0 \qquad (26)cr$$

where $r' = 1..R$, $s' = 1..S$, and

$$P^{11}_{r'r} = A \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,r'}(\alpha_n) \widetilde{f}_{x,r}(\alpha_n)$$

$$Q^{11}_{r's} = B \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,r'}(\alpha_n) \widetilde{f}_{x,s}(\alpha_n)$$

$$P^{12}_{r'r} = C \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,r'}(\alpha_n) \widetilde{f}_{x,r}(-\alpha_n)$$

$$Q^{12}_{r's} = D \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,r'}(\alpha_n) \widetilde{f}_{x,s}(-\alpha_n)$$

$$R^{11}_{s'r} = B \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,s'}(\alpha_n) \widetilde{f}_{x,r}(\alpha_n)$$

$$T^{11}_{s's} = E \sum_{n=-\infty}^{+\infty} \widetilde{f}^*_{x,s'}(\alpha_n) \widetilde{f}_{x,s}(\alpha_n)$$

$$R^{12}_{s'r} = D \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(\alpha_n)\widetilde{f}_{x,r}(-\alpha_n)$$

$$T^{12}_{s's} = F \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(\alpha_n)\widetilde{f}_{x,s}(-\alpha_n)$$

$$P^{21}_{r'r} = C \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,r'}(-\alpha_n)\widetilde{f}_{x,r}(\alpha_n)$$

$$Q^{21}_{r's} = D \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,r'}(-\alpha_n)\widetilde{f}_{x,s}(\alpha_n)$$

$$P^{22}_{r'r} = A \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,r'}(-\alpha_n)\widetilde{f}_{x,r}(-\alpha_n)$$

$$Q^{22}_{r's} = B \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,r'}(-\alpha_n)\widetilde{f}_{x,s}(-\alpha_n)$$

$$R^{21}_{s'r} = D \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(-\alpha_n)\widetilde{f}_{x,r}(\alpha_n)$$

$$T^{21}_{s's} = F \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(-\alpha_n)\widetilde{f}_{x,s}(\alpha_n)$$

$$R^{22}_{s'r} = B \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(-\alpha_n)\widetilde{f}_{x,r}(-\alpha_n)$$

$$T^{22}_{s's} = E \sum_{n=-\infty}^{+\infty} \widetilde{f}^{*}_{x,s'}(-\alpha_n)\widetilde{f}_{x,s}(-\alpha_n)$$

For $R = 2$ and $S = 1$:

$$\begin{bmatrix} P^{11}_{11} & P^{11}_{12} & Q^{11}_{11} & P^{12}_{11} & P^{12}_{12} & Q^{12}_{11} \\ P^{11}_{21} & P^{11}_{22} & Q^{11}_{21} & P^{12}_{21} & P^{12}_{22} & Q^{12}_{21} \\ R^{11}_{11} & R^{11}_{12} & T^{11}_{11} & R^{12}_{11} & R^{12}_{12} & T^{12}_{11} \\ P^{21}_{11} & P^{21}_{12} & Q^{21}_{11} & P^{22}_{11} & P^{22}_{12} & Q^{22}_{11} \\ P^{21}_{21} & P^{21}_{22} & Q^{21}_{21} & P^{22}_{21} & P^{22}_{22} & Q^{22}_{21} \\ R^{21}_{11} & R^{21}_{12} & T^{21}_{11} & R^{22}_{11} & R^{22}_{12} & T^{22}_{11} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_1 \\ a_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$\begin{bmatrix} (P_{11}^{11}+P_{11}^{12}) & (P_{12}^{11}+P_{12}^{12}) & (Q_{11}^{11}+Q_{11}^{12}) \\ (P_{21}^{11}+P_{21}^{12}) & (P_{22}^{11}+P_{22}^{12}) & (Q_{21}^{11}+Q_{21}^{12}) \\ (R_{11}^{11}+R_{11}^{12}) & (R_{12}^{11}+R_{12}^{12}) & (T_{11}^{11}+T_{11}^{12}) \\ (P_{11}^{21}+P_{11}^{22}) & (P_{12}^{21}+P_{12}^{22}) & (Q_{11}^{21}+Q_{11}^{22}) \\ (P_{21}^{21}+P_{21}^{22}) & (P_{22}^{21}+P_{22}^{22}) & (Q_{21}^{21}+Q_{21}^{22}) \\ (R_{11}^{21}+R_{11}^{22}) & (R_{12}^{21}+R_{12}^{22}) & (T_{11}^{21}+T_{11}^{22}) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Because $f_{z,r}(x)$ and $f_{z,s}(x)$ are real-and-even, their Fourier transforms, $\tilde{f}_{z,r}(\alpha_n)$ and $\tilde{f}_{z,s}(\alpha_n)$ are also real-and-even, i.e.:

$$\begin{aligned} \tilde{f}^*_{z,r}(\alpha_n) &= \tilde{f}_{z,r}(\alpha_n) \\ \tilde{f}_{z,r}(-\alpha_n) &= \tilde{f}_{z,r}(\alpha_n) \\ \tilde{f}^*_{z,s}(\alpha_n) &= \tilde{f}_{z,s}(\alpha_n) \\ \tilde{f}_{z,s}(-\alpha_n) &= \tilde{f}_{z,s}(\alpha_n) \end{aligned}$$

Therefore the top-half and the bottom-half of the 6-by-3 matrix are equal, i.e.:

$$\begin{bmatrix} (P_{11}^{11}+P_{11}^{12}) & (P_{12}^{11}+P_{12}^{12}) & (Q_{11}^{11}+Q_{11}^{12}) \\ (P_{21}^{11}+P_{21}^{12}) & (P_{22}^{11}+P_{22}^{12}) & (Q_{21}^{11}+Q_{21}^{12}) \\ (R_{11}^{11}+R_{11}^{12}) & (R_{12}^{11}+R_{12}^{12}) & (T_{11}^{11}+T_{11}^{12}) \end{bmatrix} =$$

$$\begin{bmatrix} (P_{11}^{21}+P_{11}^{22}) & (P_{12}^{21}+P_{12}^{22}) & (Q_{11}^{21}+Q_{11}^{22}) \\ (P_{21}^{21}+P_{21}^{22}) & (P_{22}^{21}+P_{22}^{22}) & (Q_{21}^{21}+Q_{21}^{22}) \\ (R_{11}^{21}+R_{11}^{22}) & (R_{12}^{21}+R_{12}^{22}) & (T_{11}^{21}+T_{11}^{22}) \end{bmatrix}$$

and only one of the above 3-by-3 systems is required for solving β:

$$\begin{bmatrix} (P_{11}^{11}+P_{11}^{12}) & (P_{12}^{11}+P_{12}^{12}) & (Q_{11}^{11}+Q_{11}^{12}) \\ (P_{21}^{11}+P_{21}^{12}) & (P_{22}^{11}+P_{22}^{12}) & (Q_{21}^{11}+Q_{21}^{12}) \\ (R_{11}^{11}+R_{11}^{12}) & (R_{12}^{11}+R_{12}^{12}) & (T_{11}^{11}+T_{11}^{12}) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ b_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (27)$$

Non-trival solutions of the above system of equations can be found by obtaining the values of β for which the determinant of the $PQRT$-matrix is zero. Real values of β correspond to a propagation mode and imaginary values of β signify evanescent modes. Once β is obtained, the expansion coefficients in equation (27) can be solved by a standard library procedure such as Gaussian-elimination. These coefficients are useful in finding the x-dependence of the field at the fin positions. Because β is known, the field inside the structure can be found via Maxwell's equations by assuming a suitable variation of the field components along the x and y-direction.

The characteristic impedance of E-plane structures is application dependent. For finlines the generally accepted definition is the voltage power impedance:

$$Z_o(V,P) = \frac{V_{slot}^2}{2P} \tag{28}$$

$$V_{slot} = \int_{\frac{-d}{2}}^{\frac{d}{2}} E_x(x, y_{slot})dx \tag{29}$$

$$P_{total} = Re \int_{\frac{-b}{2}}^{\frac{b}{2}} \int_{\frac{-a}{2}}^{\frac{a}{2}} (E_x H_y^* - E_y H_x^*)dydx = \int_{\frac{-a}{2}}^{\frac{a}{2}} (\widetilde{E}_x \widetilde{H}_y^* - \widetilde{E}_y \widetilde{H}_x^*)dy \tag{30}$$

where V_{slot} is the voltage calculated along the shortest distance between the fin edges and P_{total} is the total power carried by the structure.

Unilateral and bilateral finlines are more often used in general applications; antipodal finline is seldom used. Hence, only programs computing the effective dielectric constant and voltage-power impedance of unilateral and bilateral finlines have been implemented in this CAD procedure.

CHAPTER 2

A: Generation and Manipulation

The previous chapter indicates that the effective dielectric constant and effective characteristic impedance of finline must be found by some means before the proposed discontinuity models can be ultilized. Field-theory based numerical programs such as Finite Element Method—FEM, Transmssion Line Matrix—TLM and Spectral Domain Methods—SDM are suitable for this purpose. However, combining these numerical programs directly with the discontinuity models is not a suitable implementation for computer aided design application because even the fastest field-theorictal tool is too slow for CAD purposes. This is because optimization relies on a repeated analysis of a circuit model for a large number of frequency points. A good alternative is to generate lookup tables. One attractive advantage of this alternative is that it is not associated with a specific table generation method. The only requirement is that the lookup table conform to a prescribed format. Another advantage is that the lookup tables need only be calculated once. Thus very "fast" E-plane circuit models can be realized. In this thesis, an accelerated Spectral Domain program [5], which is one order of magnitude faster than a regular SDM program, has been chosen to generate lookup tables of ϵ_{eff} and Z_{eff} for finlines. However, any other numerical technique may be used for this purpose.

A suitable lookup table format must be determined before implementing table generation programs; a lookup table of the smallest possible dimension is the goal. Figure 1-1(a) shows the physical parameters of an unilateral finline structure: the inner dimensions of the waveguide housing (a, b), substrate thickness and permittivity (s, ϵ_r), lateral position of the fin (h) and gapwidth (d). Another parameter needed to be considered is frequency (f). Obviously, the parameters to be varied most frequently during analysis and optimization are the gapwidth and frequency. The remaining parameters are usually fixed in advance and are not modified during a typical phase of the design procedure. Therefore, a two-dimensional lookup table, which has *gapwidth* and *frequency* as the variable parameters, is a suitable choice for most applications.

Lookup table generation programs for unilateral and bilateral finlines (*UniFastS* and *BiFastS*) have been developed by using the accelerated Spectral Domain method (see Appedix A). These programs are very user-friendly. They can generate lookup tables of size m–by–n, where $m, n \in [2, 11]$ inclusively. An input menu of *BiFastS* is shown in Figure 2A-1. Lookup tables of size 11-by-11 (see Table B1 in Appendix B) can be generated in less than two minutes by running these programs under a 16-MHz clock rate Intel–80386 (with coprocessor 80287) based personal computer.

Lookup tables are just an array of numbers which are only useful as a basis for computations. It is important to ensure that ϵ_{eff} and Z_{eff} are well behaved in the chosen frequency and gapwidth range before the lookup tables are used for CAD application. The best way to do this is to depict the lookup tables graphically. A three-dimensional graphic program (*3D-Graf*) has been written for this purpose (see Appendix C). A screen output of this program is shown in Figure 2A-2. This program allows the graphical image of the lookup tables to be viewed at various orientations. The size and orientation of the picture can be changed interactively within seconds. This allows the user to inspect the physical behaviour of ϵ_{eff} and Z_{eff} effectively.

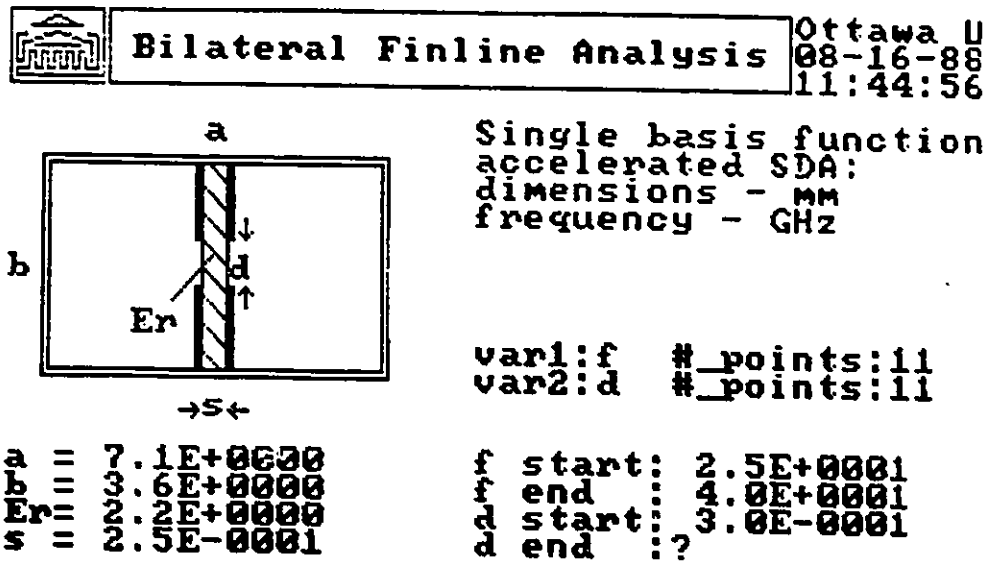


Figure 2A-1: An input menu of a table generation program — *BiFastS*. The question mark is an input prompt indicator. See Appendix A for more details.

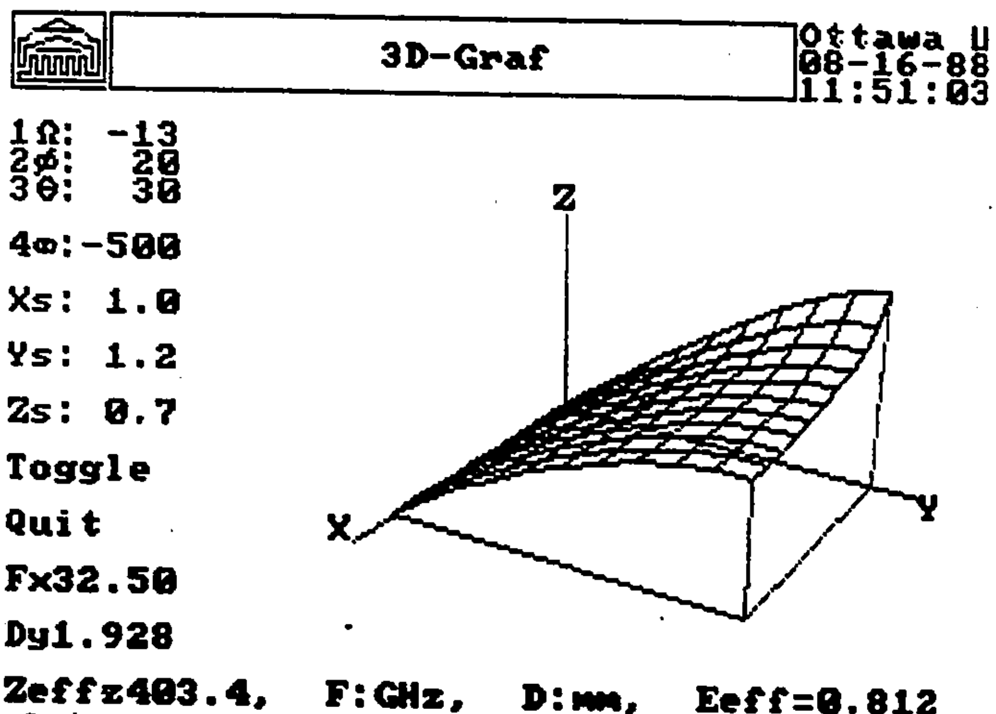


Figure 2A-2: A screen output of *3D-Graf*. This is a graphical image of Z_{eff} versus f and d, which is given by Table B1 in Appendix B. In this graph, the x-axis respresents f, the y-axis respresents d and the z-axis represents Z_{eff}. See Appendix C for more details about this graphic program.

B: Data Interpolation

Two dimensional interpolations must be performed in order to extract information from the lookup tables. Clearly, the accuracy of the results is directly related to the size of the lookup tables and the method of interpolation. By using the proper interpolation functions, the size of the lookup table can be reduced to a minimum without sacrificing the accuracy of the results. Hence it is important to explore the physical behaviour of the effective dielectric constant, ϵ_{eff}, and the effective characteristic impedance, Z_{eff}, of the transmission medium.

Meier [6] has shown that the effective dielectric constant and the characteristic impedance of finline can be approximately expressed as follows:

$$\epsilon_{eff} \approx k_e - \left(\frac{\lambda}{\lambda_g}\right)^2 \tag{1}$$

$$Z_{eff} \approx \frac{Z_o}{\sqrt{k_e - \left(\frac{\lambda}{\lambda_g}\right)^2}} \quad \text{or} \quad Z_{eff}^{-2} \approx Z_o^{-2}\left(k_e - \left(\frac{\lambda}{\lambda_g}\right)^2\right) \tag{2}$$

where Z_o is the voltage-power impedance. k_e is the so-called equivalent permittivity to be determined by a test measurement. A better approximation is obtained by assuming that k_e is frequency dependent:

$$k_e = k_{e0} + k_{e1} f \tag{3}$$

so that in the most general form, ϵ_{eff} and Z_{eff} are very closely emulated by the expression:

$$y = A + Bf + Cf^{-2} \tag{4}$$

where y stands for ϵ_{eff} or Z_{eff}^{-2}. A, B and C are unknown coefficients to be determined by using the lookup tables. Since three coefficients are present in formula (4), three data points are required to determine the coefficients. The system of equations required to be solved at a given gapwidth is:

$$y_1 = A + Bf_1 + Cf_1^{-2}$$
$$y_2 = A + Bf_2 + Cf_2^{-2}$$
$$y_3 = A + Bf_3 + Cf_3^{-2}$$

Using elimination the and back-substitution technique (Appendix D),

$$C = \frac{k_2(y_1 - y_2) - k_3(y_1 - y_3)}{k_5} \tag{5}$$

$$B = \frac{y_1 - y_2 - Ck_4}{k_3} \tag{6}$$

$$A = y_1 - Bf_1 - Ck_1 \tag{7}$$

where

$$k_1 = f_1^{-2}$$
$$k_2 = f_1 - f_3$$
$$k_3 = f_1 - f_2$$
$$k_4 = k_1 - f_2^{-2}$$
$$k_5 = k_2 k_4 - k_3(k_1 - f_3^{-2})$$

Note that k_1, k_2, k_3, k_4 and k_5 are functions of f_1, f_2 and f_3 only; they do not depend on frequency, f, nor on gapwidth, d. They can be stored with the lookup tables, and equations (4) to (7) can be applied directly to determine the corresponding ϵ_{eff} and Z_{eff} at various frequency points. This analysis shows that only three frequency points are required for each gapwidth.

The dependence of ϵ_{eff} and Z_{eff} on the gapwidth is more complicated. If parabolas are used to perform interpolation with respect to gapwidth then lookup tables of size 3–by–3 would be sufficient. Figure 2B-1 depicts a possible interpolation scheme. Because ϵ_{eff} and Z_{eff} are usually quite insensitive to frequency [2], interpolation is first performed with respect to frequency, then with respect to gapwidth. The interpolation procedure is:

(I) Apply equations (4) to (7) three times at three different gapwidths, d, to determine y_a, y_b and y_c.

(II) Use y_a, y_b and y_c with a parabolic interpolation method to determine y_o.

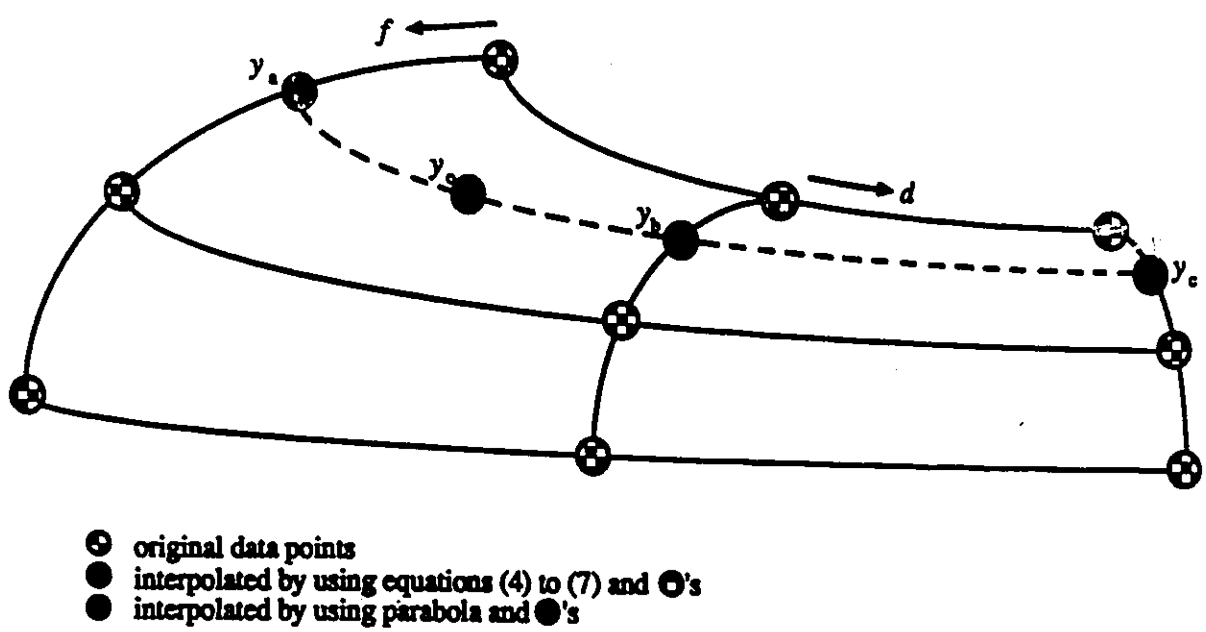


Figure 2B-1: An interpolation scheme which uses a lookup table of size 3–by–3.

The 3–by–3 interpolation scheme described in Figure 2B-1 is very attractive because it allows the size of the lookup table to be as small as 3–by–3. A question to be asked is: "How accurate is this interpolation scheme?". A method of comparison must be established before this question can be answered.

One feasible method is to construct a 3–by–3 table† by extracting nine data points from an accurate 11–by–11 table. Then reconstruct an 11–by–11 lookup table by using the 3–by–3 table and the 3–by–3 interpolation scheme in Figure 2B-1. If table $\mathbf{A}=[a_{ij}]$ is the accurate table and table $\mathbf{R}=[r_{ij}]$ is the reconstructed table then an error table $\mathbf{E}=[e_{ij}]$ can be constructed as follows:

$$e_{ij} = \frac{a_{ij} - r_{ij}}{a_{ij}} = 1 - \frac{r_{ij}}{a_{ij}} \qquad i,j \in \{1,2,3,\ldots,11\}$$

The infinity norm and maximum norm of $\mathbf{E}$ are:

$$|\mathbf{E}|_\infty = \sum_{ij} |e_{ij}|$$

$$|\mathbf{E}|_{\max} = \max\{|e_{ij}|\}$$

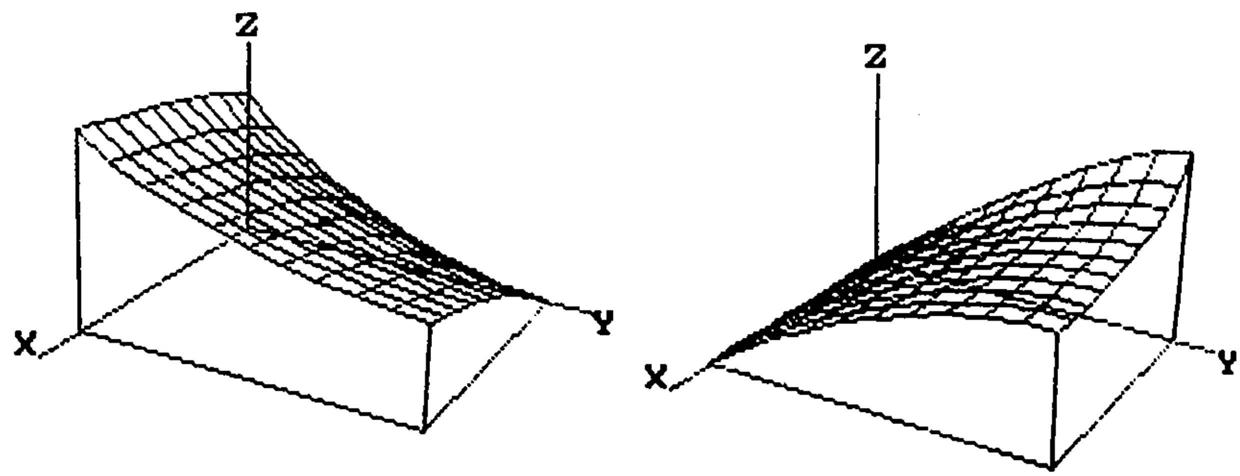
These two norms are measurements of the accuracy of this 3–by–3 interpolation method. The content of $\mathbf{E}$ can be displayed by using *3D-Graf*; such a graphical image of $\mathbf{E}$ can reveal the quality of this interpolation method effectively.

The sequence of diagrams in Figure 2B-2 illustrates this idea graphically. The 11–by–11 tables in Figure 2B-2(a) are generated by using *UniFastS*. Figure 2B-2(b) shows two 3–by–3 tables extracted from the corresponding 11–by–11 tables. Using array index notation, the (f,d) pairs correspondence between these two pairs of lookup tables are shown in Table 2B-1. Applying the 3–by–3 interpolation scheme in Figure 2B-1, some $[i,j]$ entries can be reconstructed by using entries $([1,j],[6,j],[11,j])$ and physically realistic functions, i.e. equation (4) to (7), where $i \in \{2,3,4,5,7,8,9,10\}$ and $j \in \{1,6,11\}$. The remaining $[i,j]$ entries can be reconstructed by using $([i,1],[i,6],[i,11])$ and parabolas, where $i \in \{1,2,3,\cdots,11\}$ and $j \in \{2,3,4,5,7,8,9,10\}$. A program called *Intpol33* that can perform this reconstruction is given in Listing E1 in Appendix E. The tables that are reconstructed from the 3–by–3 tables in Figure 2B-2(b) are shown in Figure 2B-2(c).

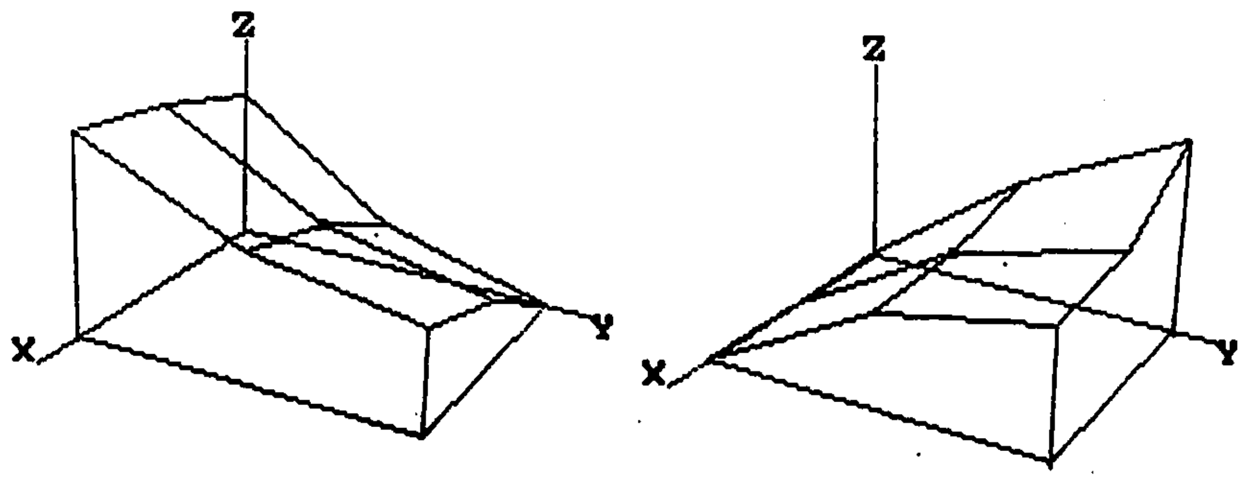
size	index								
11-by-11	1,1	1,6	1,11	6,1	6,6	6,11	1,1	11,6	11,11
3-by-3	1,1	1,2	1,3	2,1	2,2	2,3	3,1	3,2	3,3

Table 2B-1: The node number correspondence between Figures 2B-2(a) and (b).

† Note that a lookup table consists of two tables ϵ_{eff} versus (f,d) and Z_{eff} versus (f,d).

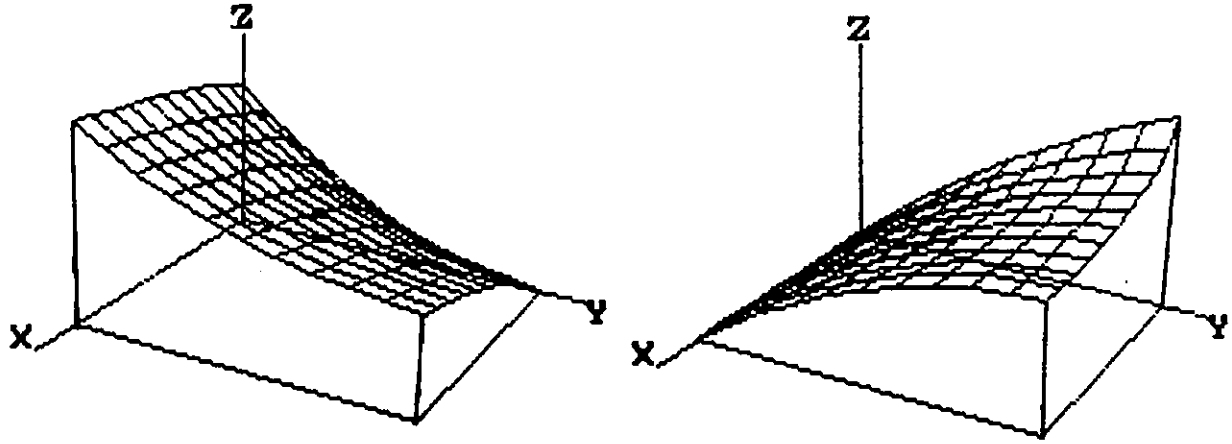


(a): Two 11–by–11 tables. The left table represents ϵ_{eff} (z-axis) versus f (x-axis) and d (y-axis) while the right table shows Z_{eff} versus f and d.

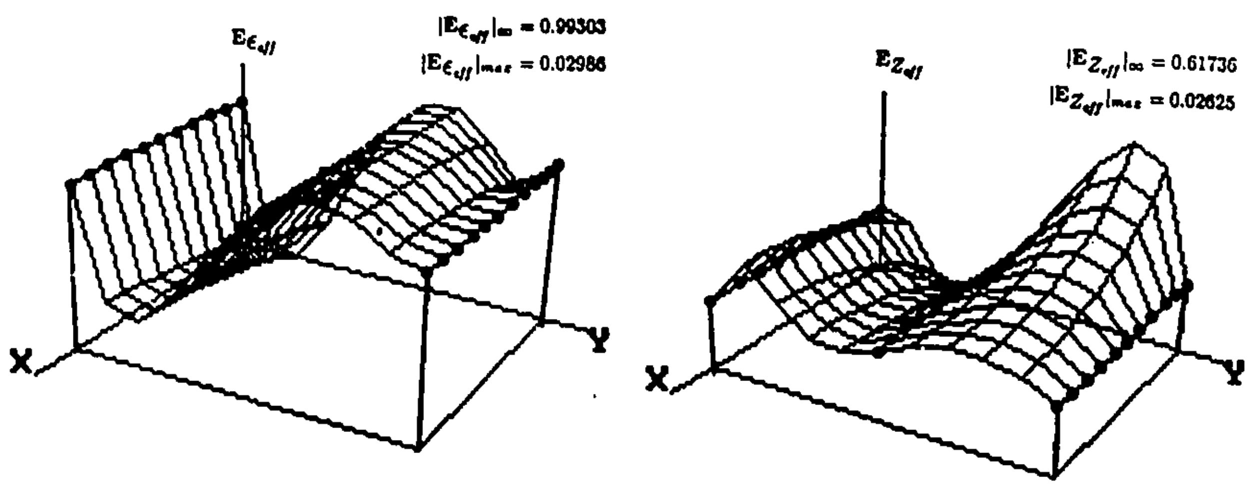


(b): Two 3–by–3 tables extracted from the 11–by–11 tables in (a).

Figure 2B-2: A sequence of diagrams which illustrate the error introduced by the 3–by–3 interpolation scheme in Figure 2B-1.



(c): Two 11–by–11 tables that are reconstructed by using the 3–by–3 tables in (b) and the 3–by–3 interpolation scheme in Figure 2B-1.



(d): $E_{\epsilon_{eff}}$ and $E_{Z_{eff}}$. Note that the dotted values have small error because they are reconstructed by using physically realistic functions. These graphs show that interpolations with respect to gapwidth are not accurate due to the use of parabolic function.

Figure 2B-2 (cont'): A sequence of diagrams which illustrates the error introduced by the 3–by–3 interpolation scheme in Figure 2B-1.

Two error tables $E_{\epsilon_{eff}}$ and $E_{Z_{eff}}$ are computed by a simple program called *Error*, which is given in Listing E3 in Appendix E. The contents of these error tables are depicted in Figure 2B-2(d). Note that the dotted values have relatively small errors, less than 0.05 percent. This is expected because these points are generated by using physically realistic functions. Another expected behaviour is that the errors are large in the regions close to cutoff frequency and small gapwidth. The infinity norms and maximum norms of $E_{\epsilon_{eff}}$ and $E_{Z_{eff}}$ are not small enough for practical design requirements.

Figure 2B-2(d) shows that the large error of the previous interpolation scheme is mainly due to the poor performance of the parabolic functions which are applied to carry out the interpolation with respect to gapwidth. This suggests that either higher-order polynomials or denser lookup tables are needed. Using higher-order polynomials implies slower program execution speed; this is not desirable in CAD application. Therefore, the latter alternative is preferable. A possible scheme is to use eleven data points and linear interpolation to determine the ϵ_{eff} and Z_{eff} at various gapwidth. This 3-by-11 interpolation scheme is an alternative to the previous method; it uses eleven gapwidth points rather than three and uses linear interpolation rather than parabolic interpolation (see Listing E2 in Appendix E). The error norms obtained by applying these two interpolation methods to unilateral and bilateral finlines in WR22 and WR28 waveguides are shown in Table 2B-2. The result shows that this 3-by-11 interpolation scheme is much more accurate than the 3-by-3 interpolation scheme discussed previously.

Table 2B-2 shows that for unilateral and bilateral finlines the 3-by-11 interpolation scheme is sufficiently accurate for most practical purpose. However for other transmission media that do not behave like unilateral and bilateral finlines, such as antipodal finlines, this 3-by-11 interpolation scheme might fail. Hence it is necessary to have a general interpolation scheme in the system to handle this type of problem. Since *UniFastS* and *BiFastS* can generate tables of size up to 11-by-11 (in general m-by-n), it is convenient to choose lookup tables of size 11-by-11 and use linear interpolation with respect to both frequency and gapwidth.

3-by-3 Scheme			
norm	WR	infinity	maximum
ϵ_{eff}	221	1.61174	0.04351
	222	2.03550	0.05419
	281	0.99303	0.02986
	282	1.59536	0.04364
Z_{eff}	221	0.63589	0.02315
	222	0.85385	0.03603
	281	0.61736	0.02625
	282	0.67214	0.02337

3-by-11 Scheme			
norm	WR	infinity	maximum
ϵ_{eff}	221	0.00191	0.00011
	222	0.00126	0.00004
	281	0.00489	0.00015
	282	0.00353	0.00013
Z_{eff}	221	0.02376	0.00393
	222	0.01162	0.00037
	281	0.01432	0.00059
	282	0.01419	0.00059

Table 2B-2: Comparison of error norms which are obtained by using two different interpolation schemes. Note that WR221, WR222, WR281 and WR282 mean unilateral and bilateral finlines in WR22 and WR28 waveguides (See Section D for more details about the naming convention.).

C: Data Compression

The lookup tables in Appendix C are designed to be readable. Many unnecessary characters are added into the text files to make them look good, at least to human readers. These lookup tables require a large amount of memory. On the other hand, if the lookup tables are stored in a binary format with all the unnecessary characters removed, only one-tenth of the memory is needed. Since the amount of main memory is usually limited, it is better† to compress the tables into a binary format while they are in the main memory. In addition to reducing memory usage, binary file format increases the input-output speed by eliminating the decoding procedure which is required to convert ASCII characters to floating point numbers. Therefore, two programs (*Make-Re4* and *Make-311*) are written to perform data compression for the 11–by–11 and 3–by–11 lookup tables (see Appendix F).

D: Naming Convention

Meaningful filenames are used to name the lookup tables so that important information about the lookup tables can easily be found by reading the filenames. The naming convention of waveguide is modified to name these lookup tables so that the waveguide dimension, table size as well as the finline type are stated clearly in the filenames. This convention is summarized in Table 2D-1:

Lookup Tables File Name Convention			
Format	Size	Convention	Note
Text	3-by-11	WRmnt311.Dat	recommended
	11-by-11	WRmnt.Dat	recommended
Binary	3-by-11	WRmntA.311	required
	11-by-11	WRmnt.Re4	required

mn: waveguide number; 22, 28 etc.
t: finline type; 1 for unilateral, 2 for bilateral.
recommended: it is not necessary to use this convention.
required: TsiFin, a CAD program to be introduced in Chapter4 (Appendix I), uses this convention to access lookup tables.

Table 2D-1: Lookup table naming convention.

† This compression step is necessary in order to integrate these lookup tables with Touchstone™ Senior. It is due to the static memory limitation of Microsoft Pascal and the 640K memory limitation of DOS.

CHAPTER 3 Implementation of E-Plane Circuit Elements

Two-port and one-port E-plane circuit elements have been implemented using the aforementioned discontinuity models and lookup tables. The two-port elements are the straight finline section, shunt junction, and step junction. The one-port element is a perfect matching load.

(I) Finline Section and Its Equivalent Model

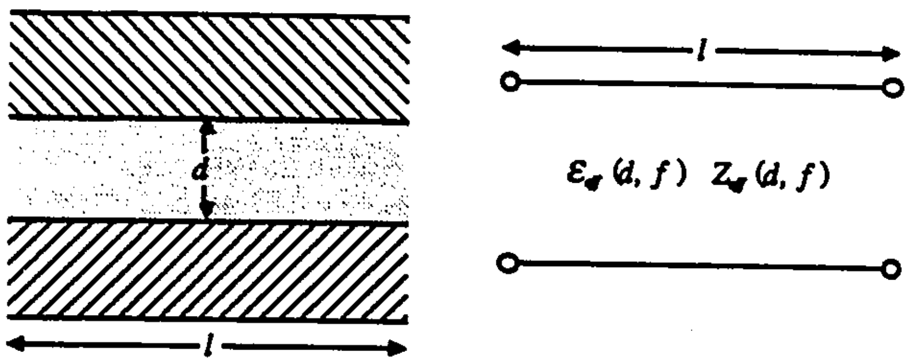


Figure 3-1: A finline line section and its equivalent model.

The S-matrix of a line section of length l, impedance Z and reference impedance Z_0 is, [7]:

$$S = \frac{1}{D_s}\begin{bmatrix} (Z^2 - Z_0^2)Sh & 2ZZ_0 \\ 2ZZ_0 & (Z^2 - Z_0^2)Sh \end{bmatrix}$$

where

$$D_s = 2ZZ_0Ch + (Z^2 + Z_0^2)Sh$$
$$Sh = \sinh(\gamma l)$$
$$Ch = \cosh(\gamma l)$$

For finline:

$$Z = Z_{eff}$$
$$\gamma = \alpha_{eff} + j\beta_{eff}$$
$$\beta_{eff} = \frac{2\pi f}{c}\sqrt{\epsilon_{eff}}$$

ϵ_{eff} and Z_{eff} are found by using lookup tables. α_{eff} is the attenuation parameter and it is a function of gapwidth and frequency. However, it can be assumed zero in most applications. For the cases where the loss cannot be neglected, a table can be generated for α_{eff}.

(II) Shunt Junction and Its Equivalent Model

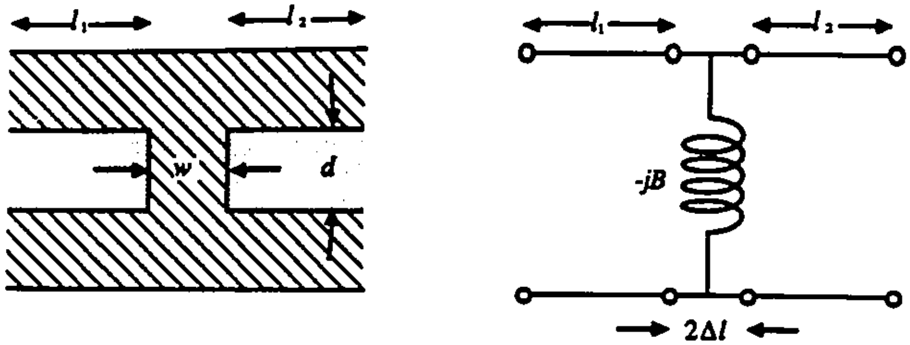


Figure 3-2: Shunt junction and its equivalent model[3].

The ABCD-matrix of a shunt admittance $Y = -jB$ is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{shunt} = \begin{bmatrix} 1 & 0 \\ Y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -jB & 1 \end{bmatrix}$$

The ABCD-matrix of a transmission line section of length l and impedance Z is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Z,l} = \begin{bmatrix} Ch & Z\,Sh \\ Z/Sh & Ch \end{bmatrix}$$

where

$$Ch = \cosh(\gamma l)$$
$$Sh = \sinh(\gamma l)$$

Z and γ are given in model (I).

The ABCD-matrix of the equivalent model is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{model} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Z,(l_1+\Delta l)} \times \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{shunt} \times \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Z,(l_2+\Delta l)}$$

The S-matrix of the shunt junction can be found by using the ABCD-to-S transformation formulas, [7]:

$$S_{11} = \frac{AZ_{02} + B - CZ_{01}Z_{02} - DZ_{01}}{AZ_{02} + B + CZ_{01}Z_{02} + DZ_{01}}$$
$$S_{12} = \frac{2(AD - BC)\sqrt{Z_{01}Z_{02}}}{AZ_{02} + B + CZ_{01}Z_{02} + DZ_{01}}$$
$$S_{21} = \frac{2\sqrt{Z_{01}Z_{02}}}{AZ_{02} + B + CZ_{01}Z_{02} + DZ_{01}}$$
$$S_{22} = \frac{-AZ_{02} + B - CZ_{01}Z_{02} + DZ_{01}}{AZ_{02} + B + CZ_{01}Z_{02} + DZ_{01}}$$

Where Z_{01} and Z_{02} are reference impedances, usually $50\,\Omega$, in port 1 and port 2 respectively.

(III) Step Junction and Its Equivalent Model

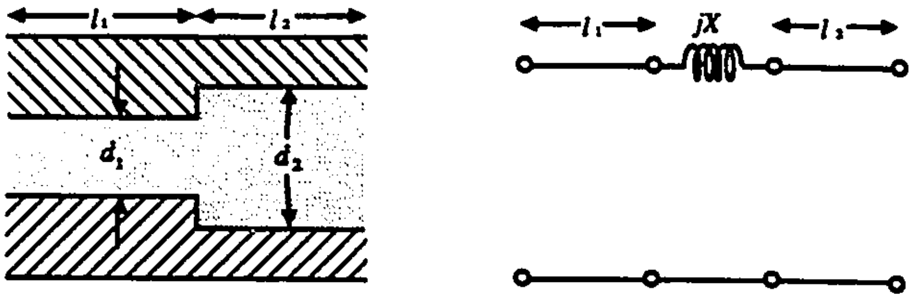


Figure 3-3: Step junction and its equivalent model[3].

The ABCD-matrix of a series impedance $Z = jX$ is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{series} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & jX \\ 0 & 1 \end{bmatrix}$$

The ABCD-matrix of the step junction is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{step} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Z_1,l_1} \times \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{series} \times \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{Z_2,l_2}$$

The S-matrix of the step junction can be found by using the transformation formulas given on the previous page.

(IV) A Wideband Matching Load

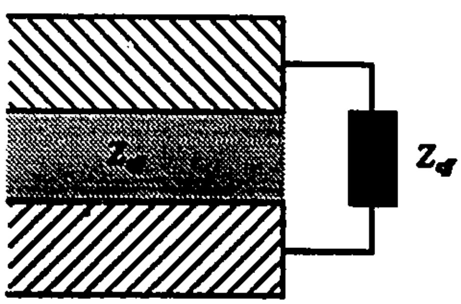


Figure 3-4: A wideband matching load

The S-matrix of this one-port element is just Γ:

$$\Gamma = \frac{Z_{eff} - Z_0}{Z_{eff} + Z_0}$$

Where Z_0 is the reference impedance.

The above circuit elements were integrated with Touchstone™ Senior, see Listing G1 in Appendix G. The calling syntax for these elements are given in Appendix H. Their model names are shown in Figure 3-5:

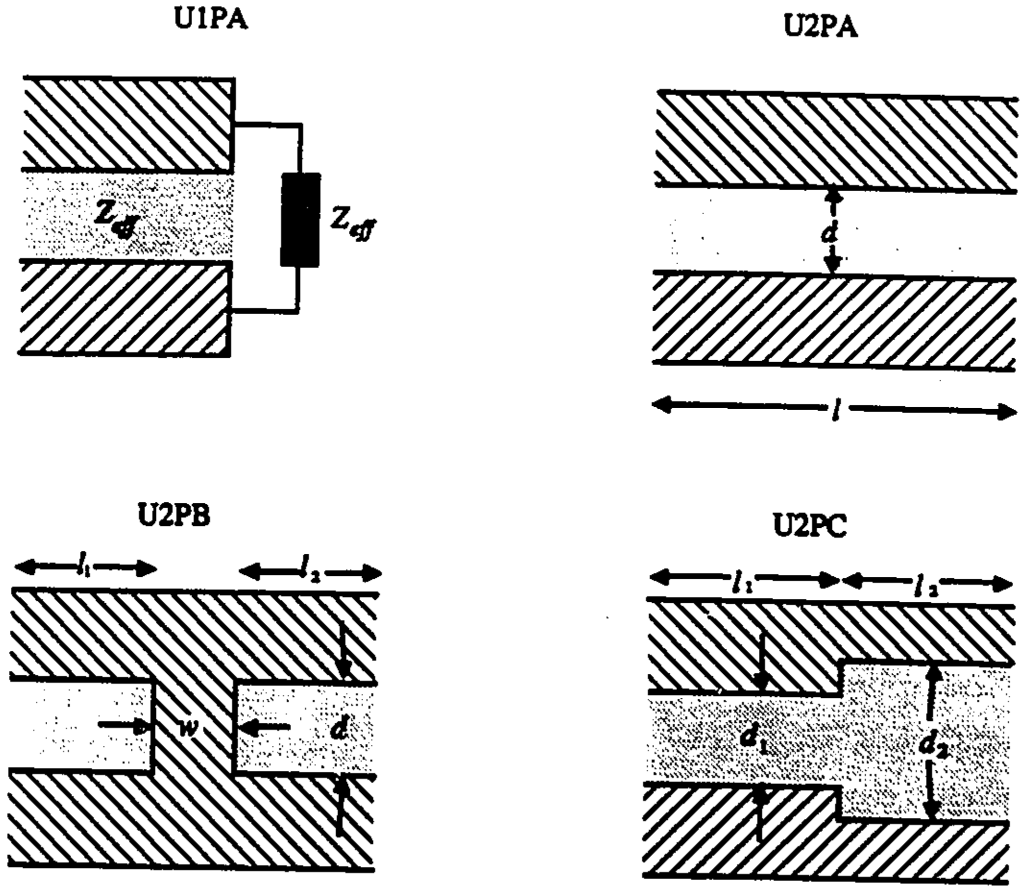


Figure 3-5: Model names of the implemented finline circuit elements.

Two interpolation schemes are implemented to obtain ϵ_{eff} and Z_{eff} which are required by the circuit elements. The first scheme is a 3-by-11 interpolation method which is based on physically realistic functions interpolated with respect to frequency, linear interpolation with respect to gapwidth and lookup table of size 3-by-11. The second scheme is an 11-by-11 interpolation method which is based on linear interpolation with respect to both frequency and gapwidth and lookup tables of size 11-by-11. In unilateral and bilateral finline cases, these two schemes give the same ‡ results. However the second scheme is a more general method and is useful for other more complicated transmission media that do not have the same dispersion characteristics as unilateral and bilateral finlines.

‡ Within one tenth of a percent difference, see Chapter 6.

CHAPTER 4

E-Plane Circuit CAD Procedure

A number of programs are introduced in Chapter 2. They are *UniFastS, BiFastS, 3D-Graf, Make-311* and *Make-Re4*. *UniFastS* and *BiFastS* are table generation programs for unilateral and bilateral finlines. *3D-Graf*, a 3-dimensional graphic program, was developed to display the contents of the lookup tables generated by *UniFastS* and *BiFastS*. *Make-311* and *Make-Re4* are table compression programs which can be used to compress a text format lookup table to a IEEE 4-byte real number format lookup table so that commercially available CAD software can use these tables more efficiently. A naming convention for the lookup tables is given at the end of Chapter 2. This naming convention is aimed to help recall the basic information of the lookup tables.

A number of E-plane circuit elements were developed in Chapter 3. These circuit elements are based on the discontinuity models in Chapter 0 and the lookup tables presented in Chapter 2. These circuit elements are integrated with Touchstone™ Senior in order to take full advantage of this software. The implemented E-plane elements are U1PA, U2PA, U2PB and U2PC, which are summarized in Figure 3A-5.

The first step in designing finline circuit components is to choose the inner dimensions of the waveguide housing (a, b), the substrate thickness and permittivity (s, ϵ_r) and the metalization distance from the waveguide wall (h); see Figure 0-1(a). The next step is to determine the operating frequency range and the allowed range of gapwidth variation. Once these parameters are known, lookup tables† can be generated by using *UniFastS* or *BiFastS*. The contents of these lookup tables could be inspected so that it can be ensured that ϵ_{eff} and Z_{eff} are well behaved in the chosen frequency and gapwidth range. This can be done easily by using *3D-Graf*. If ϵ_{eff} and Z_{eff} are not well behaved‡, the lower bound of the frequency range or the gapwidth range (or both) should be increased. This iterative process should be repeated until well-behaved lookup tables are obtained; see Figure 4-1.

The text format lookup tables must be compressed to a binary format before they can be used by the E-plane circuit elements. If the size of the lookup tables are 3–by–11 then *Make-311* can be used to compress the tables. If the size is 11–by–11 then *Make-Re4* can be used. If the lookup tables are not of the correct size, they cannot be used in this CAD procedure.

The above steps are summarized in the top part of the flow graph in Figure 4-1. The binary format lookup tables can be used together with the E-plane circuit elements to design E-plane circuit components under the Touchstone™ environment; see the bottom

† The size of the lookup tables must be either 3–by–11 or 11–by–11. The first sweeping variable must be frequency and the second one must be gapwidth.

‡ ϵ_{eff} and Z_{eff} are highly nonlinear at the region where frequency is close to cut-off and gapwidth is very small.

part of the flow graph in Figure 4-1. The calling syntax for these elements and the way to associate these elements with a particular lookup table are given in Appendix H.

A design example is given in Appendix I. This example illustrates all the capabilities of this new CAD procedure, from lookup table generation to circuit optimization. It also demonstrates the usefulness of the flow graph in Figure 4-1.

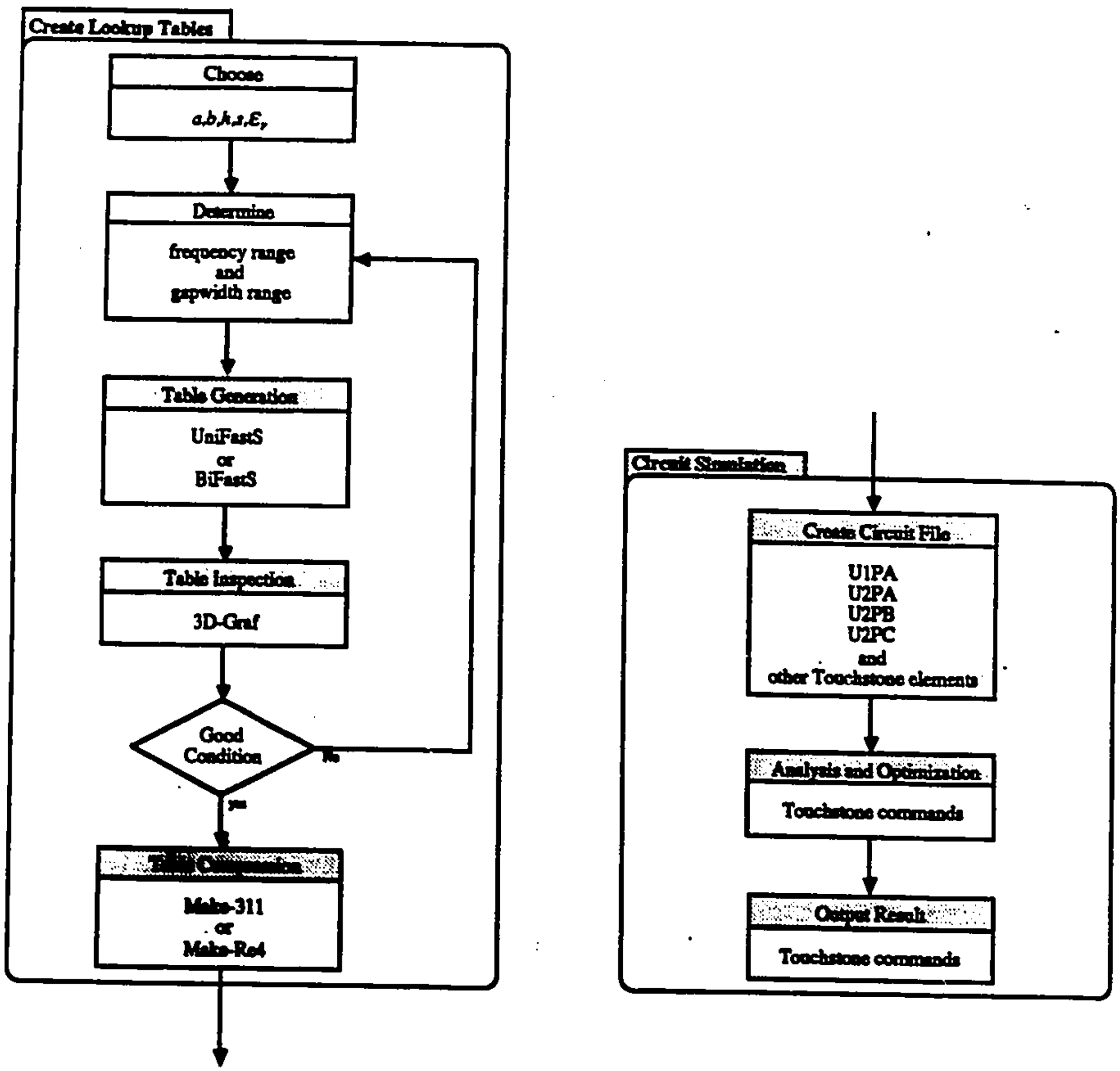


Figure 4-1: A flow graph for the novel CAD procedure.

CHAPTER 5

Computer-Simulated Results and Experimental Results

A new CAD procedure for E-plane circuits has been presented in the previous chapter. This chapter is devoted to the analysis and optimization of finline circuit components by using this new CAD procedure. The components to be dealt with are two-step Chebyshev transformers, tapers and bandpass filters. Comparisons between simulated results, theoretical results and experimental results are also presented wherever appropriate. All the Touchstone™ circuit and output files used in this chapter are found in Appendix J.

A: Two-Step Chebyshev Transformers

A complete analysis of the Chebyshev transformer is given by Collin, pp.229-233 in [8]. For a two-step transformer having a maximum reflection coefficient ρ_m over the passband, the impedance of the two quarter-wave line-sections in Figure 5A-1 can be found approximately by following an example in [8], pp.232-233. For $Z_S = 200\,\Omega, Z_L = 400\,\Omega$ and $\rho_m = 0.01$ (i.e. $-40\,\text{dB}$); the second-order Chebyshev polynomial $T_2(sec\theta_m)$ is:

$$T_2(\sec\theta_m) = 2\sec^2\theta_m - 1 = \frac{Z_L - Z_S}{(Z_L + Z_S)\rho_m} = \frac{100}{3}$$

which gives

$$\sec\theta_m = 4.143$$

$$\rho_0 = \frac{1}{2}\rho_m \sec^2\theta_m = 0.08583$$

$$\rho_1 = \rho_m(\sec^2\theta_m - 1) = 0.1617$$

$$Z_1 = \frac{1+\rho_0}{1-\rho_0} Z_S = 237.6\,\Omega$$

$$Z_2 = \frac{1+\rho_1}{1-\rho_1} Z_1 = 329.2\,\Omega$$

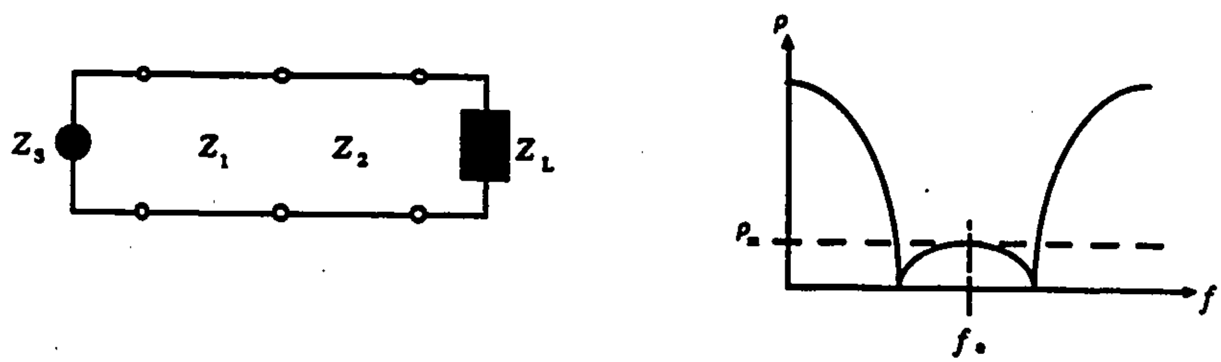
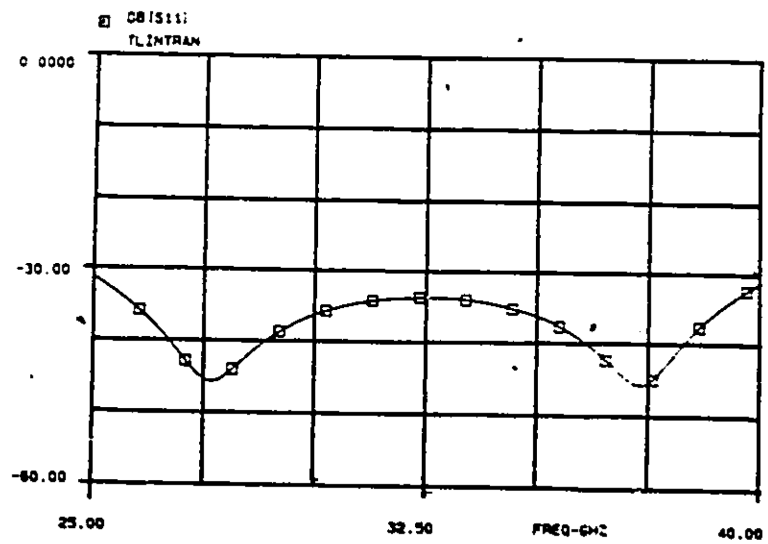
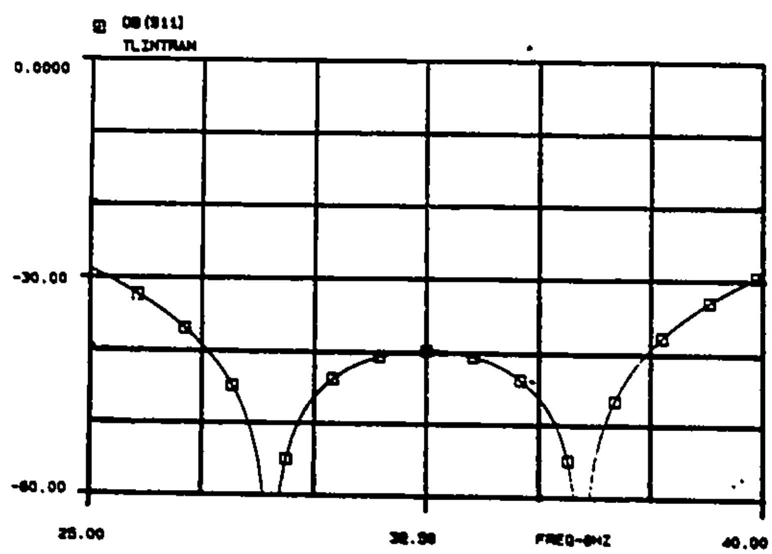


Figure 5A-1: A two-step Chebyshev transformer and its reflection characteristic.

If the calculated impedance values are entered into Touchstone™ by using an ideal transmission line model (see Listing J4 in Appendix J) the response in Figure 5A-2(a) would be obtained. The graph shows that ρ_m is greater than −40 dB. This is expected because the previous solution is only an approximation. Figure 5A-2(b) depicts an optimized result. The impedance values are Z_1=239.00Ω and Z_2=334.59Ω which are close, within two percent, to the original approximations.



(a): Return loss before optimization.



(b): Return loss after optimization.

Figure 5A-2: Return loss characteristics of a two-step Chebyshev transformer which is implemented by using ideal transmission lines.

In practice the transformer must be implemented by using a non-ideal transmission medium such as finline. One straightforward method to realize such a finline transformer is to find the gapwidth and physical length of each finline section so that at the center frequency, the impedance and the electrical length of each finline section are equivalent to their ideal transmission line counterparts. Assuming that it is required to implement this transformer using unilateral finline in WR28 waveguide, Table 5A-1 can be used to find the gapwidth and physical length of the two finline sections.

At $Z = 239\,\Omega$

$$d = d_1 + (d_2 - d_1) \times \frac{Z - Z_1}{Z_2 - Z_1} = 0.6505\,\text{mm}$$

$$\epsilon_{eff} = \epsilon_1 + (\epsilon_2 - \epsilon_1) \times \frac{Z - Z_1}{Z_2 - Z_1} = 1.0536$$

$$\frac{\lambda_g}{4} = \frac{c}{4f\sqrt{\epsilon_{eff}}} = 2.2834\,\text{mm}$$

where

$$c = 3 \times 10^{11}\ \text{mm/sec}$$

$$f = 32.5\ \text{GHz}$$

At $Z = 334.60\,\Omega$

$$d = d_3 + (d_4 - d_3) \times \frac{Z - Z_3}{Z_4 - Z_3} = 1.3356\,\text{mm}$$

$$\epsilon_{eff} = \epsilon_3 + (\epsilon_4 - \epsilon_1 3 \times \frac{Z - Z_3}{Z_4 - Z_3} = 0.9041$$

$$\frac{\lambda_g}{4} = \frac{c}{4f\sqrt{\epsilon_{eff}}} = 2.4649\,\text{mm}$$

This result is summarized in Table 5A-2.

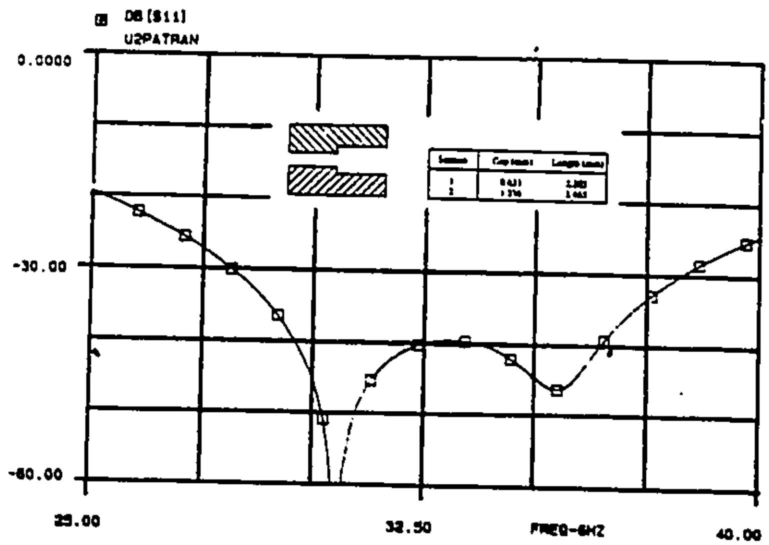
#	d mm	ϵ_{eff}	Z_{eff} Ω
1	0.62560	1.05799	235.30052
2	0.95120	0.97794	283.72473
3	1.27680	0.91387	327.40797
4	1.60240	0.85959	367.24743

Table 5A-1: ϵ_{eff} and Z_{eff} versus d at f=32.5GHz. The data are extracted from Table B1 in Appendix B.

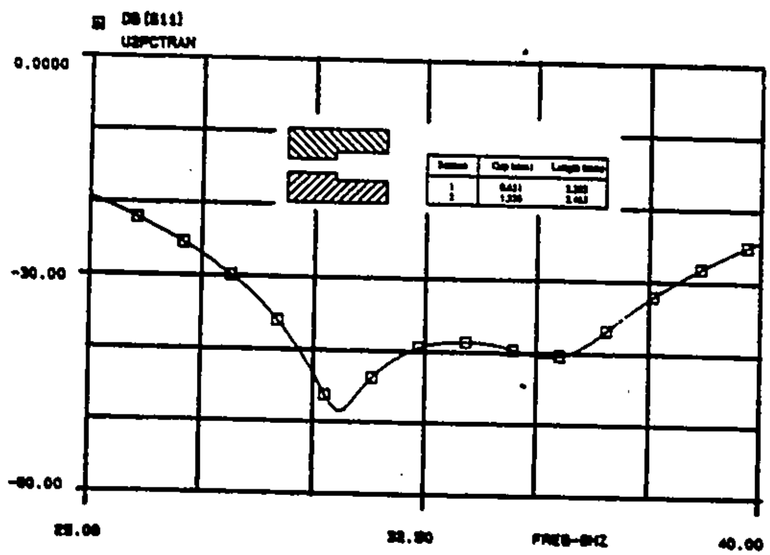
Parameters	Section 1	Section 2
gapwidth mm	0.6505	1.3356
length mm	2.2834	2.4649

Table 5A-2: The gapwidths and physical lengths of a two-step finline transformer.

There are two ways to simulate the transformer by using the dimensions in Table 5A-2. One method is to cascade two finline line-sections, U2PA elements, together without considering the step discontinuity effect. A better method is to use an U2PC element which can take into account the discontinuity effect. The circuit files of these two implementations are given in Appendix J, Listing J5 and J6. The simulation results are depicted in Figure 5A-3.



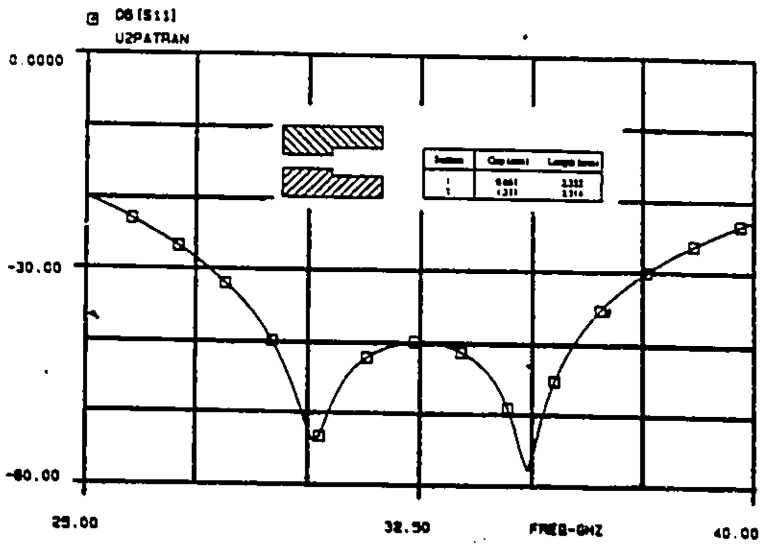
(a): Return loss of a finline transformer generated by using two U2PA elements with dimensions in Table 5A-2.



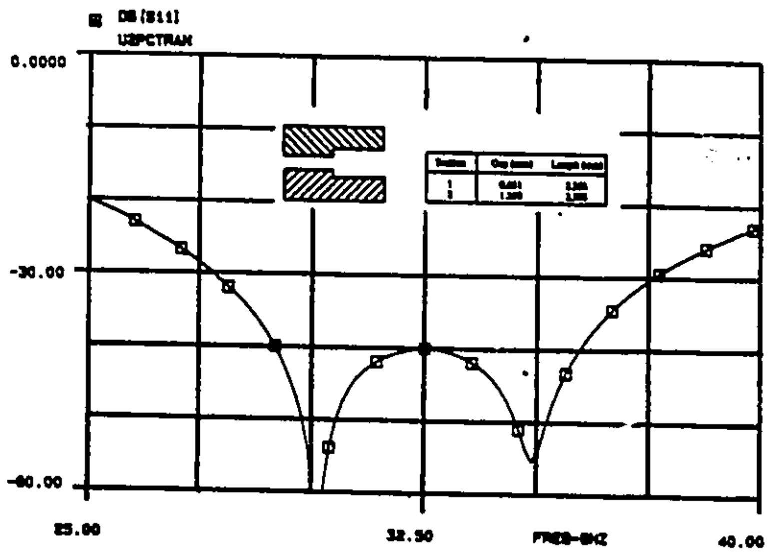
(b): Return loss of a finline transformer generated by using an U2PC element with dimensions in Table 5A-2.

Figure 5A-3: Return loss curves of two finline transformers generated by using two different circuit elements.

Figure 5A-3 shows that the dispersion and step discontinuity effects of finline degrade the overall response of the transformer. The dimensions of the two transformers can be optimized to keep ρ_m less than -40 dB over a frequency band which is centered at 32.5 GHz and has a maximum possible bandwidth. The optimized results are shown in Figure 5A-4.



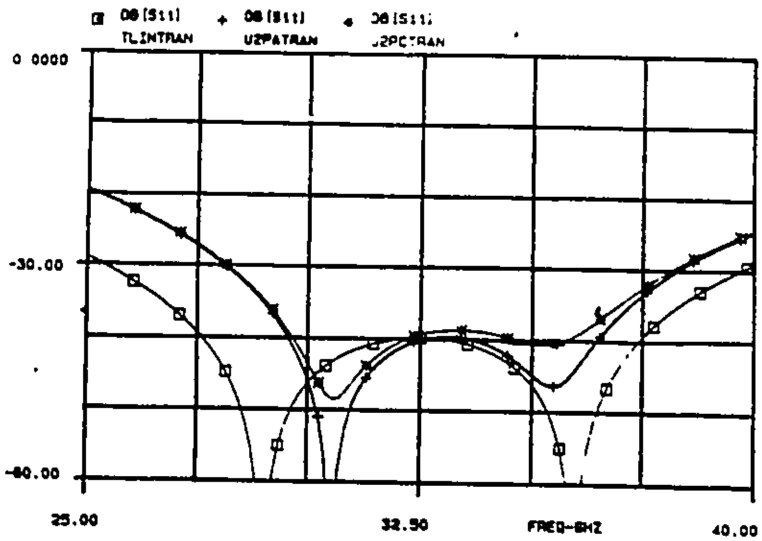
(a): Optimized dimensions and return loss of a finline transformer which is implemented by using two U2PA elements.



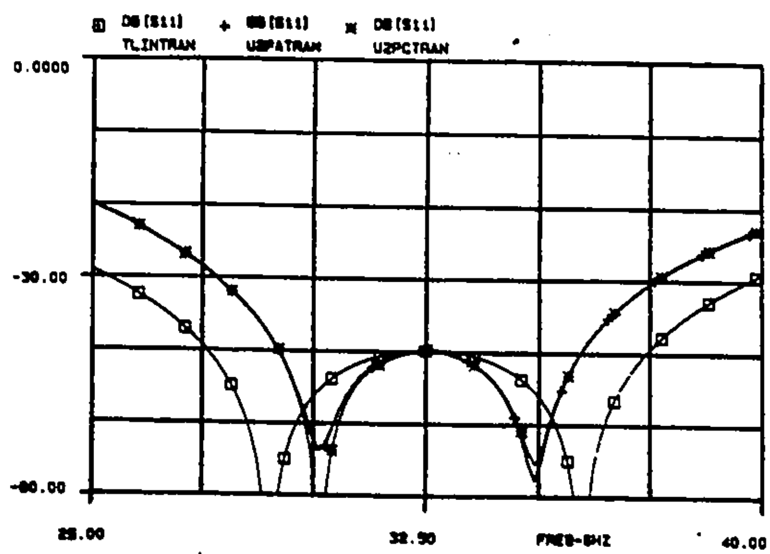
(b): Optimized dimensions and return loss of a finline transformer which is implemented by using an U2PC element.

Figure 5A-4: Return loss characteristics and dimensions of two finline transformers which are implemented by using two different circuit elements.

Figure 5A-5 depicts the response of all three transformers. For the purpose of comparison, Figure 5A-5(a) combines Figures 5A-2(b) and 5A-3 while Figure 5A-5(b) combines Figures 5A-2(b) and 5A-4. These comparisons show that the dispersion and step discontinuity effect of finline cannot be neglected. Figure 5A-5(b) shows that the Chebyshev type response can be realized by using finline elements, but the bandwidth is only about 68% of its ideal transmission line model counterpart. These effects can hardly be demonstrated without using this CAD procedure.



(a): Comparison between Figures 5A-2(b) and 5A-3. The nonideal effects of finline degrade the overall response of the transformer.



(b): Comparison between Figures 5A-2(b) and 5A-4. The bandwidth of the finline transformers are approximately 68% of the ideal transmission line implementation.

Figure 5A-5: Comparison of return loss characteristics among transformers which are implemented by using different circuit elements.

B: Tapers

The bandwidths of the two finline transformers in the previous section are approximately one half of the WR28 waveguide frequency band. Bandwidth can be increased with the use of a taper. Since there are no closed-form expressions available for optimal finline taper design, one feasible way to simulate a taper is by step approximation. This is done by cascading a large number of finline line sections and keeping the overall length fixed at the same time.

The required length of a Chebyshev taper of maximum acceptable ripple ρ_m implemented by using a non-dispersive transmission medium is given by Collin, pp. 248-251 in [8]:

$$L = \frac{1}{\beta_o}\cosh^{-1}\left(\frac{ln(Z_L/Z_S)}{2\rho_m}\right)$$
$$= \frac{c}{2\pi f_o}\cosh^{-1}\left(\frac{ln(Z_L/Z_S)}{2\rho_m}\right)$$

where $c = 3 \times 10^{11}$ mm/sec for $\epsilon_r = 1$. Z_S, Z_L, L and ρ_m are defined in Figure 5B-1.

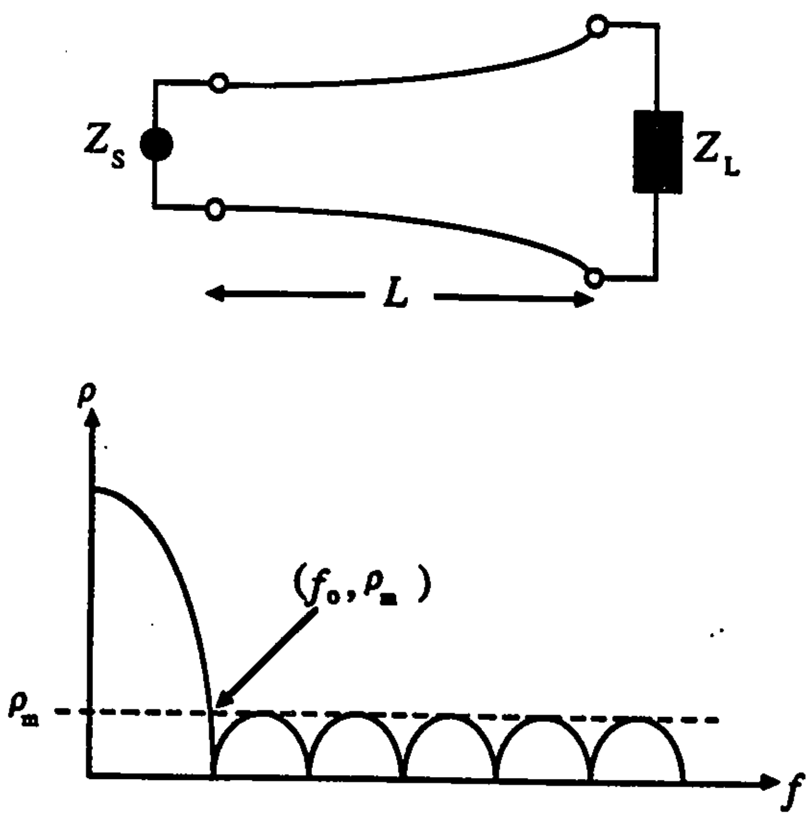


Figure 5B-1: A taper profile and Chebyshev response.

In practice, matching is only required over a certain frequency band. Therefore, the number of steps can be quite small. It would be of engineering interest to find out the minimum number of steps which are needed to achieve a matching requirement at a given taper length. A typical practical problem is to match $Z_S = 200\,\Omega$ and $Z_L = 400\,\Omega$ over the whole WR28 waveguide band (26.5 to 40 GHz) with a minimum return loss of 40 dB, i.e. $\rho_m \leq 0.01$. The minimum length, L_{min} of a taper implemented by ideal transmission lines with $\epsilon_r = 1$ is:

$$L_{min} = \frac{c}{2\pi f_o} \cosh^{-1}\left(\frac{ln(Z_L/Z_S)}{2\rho_m}\right) = 7.636 \text{ mm}$$

That is, if the number of steps is infinitely large and the medium is non-dispersive with $\epsilon_r = 1$, the length of the taper would be 7.636 mm. Since finline is a dispersive medium, and the goal here is to use the minimum number of steps due to limited computing resources†; $L = 10\,\text{mm} \approx 1.30 \times L_{min}$ is chosen as the required taper length. The problem to be solved is to find the minimum number of steps, the gapwidth and length of each step of a taper such that $\rho_m \leq 0.01$ over the frequency band of 26.5 to 40 GHz given that $Z_S = 200\,\Omega$, $Z_L = 400\,\Omega$ and $L = 10\,\text{mm}$. The following sequence of diagrams depicts the simulation results of four step-tapers (2-step, 3-step, 4-step and 5-step) which are obtained by using this novel CAD procedure. The corresponding circuit files are given in Appendix J, Listings J7 to J14.

Figures 5B-2 and 5B-3 show that the 2-step and 3-step profile cannot achieve the specified reflection requirement. Figure 5B-4 shows that the 4-step profile can meet the requirement but it gives little or no error margin. The 5-step profile in Figure 5B-5 gives a very satisfactory response. Therefore no more than five steps are necessary unless either larger bandwidth or a better reflection characteristic is needed.

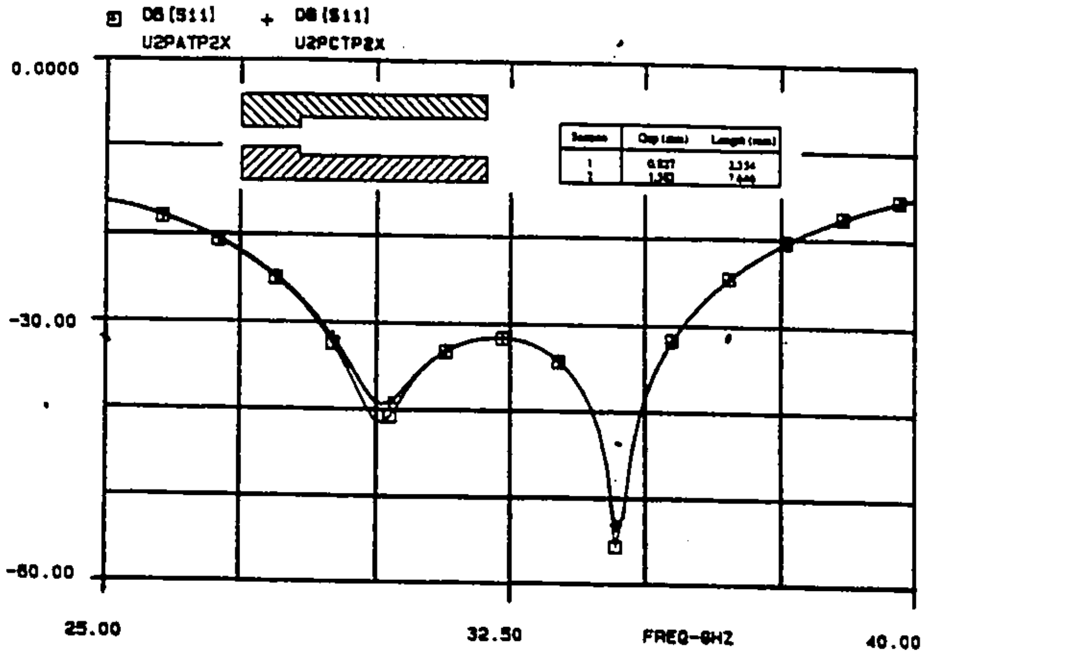
It is not straightforward to optimize circuit components that contain a large number of variables‡. Listings J7 to J18 in Appendix J all have the same optimization constraints. The constraints and reasons (if present) are:

(1) The taper length must be 10 mm.
(2) The reflection must be less than −40 dB over the frequency band of 26.5 to 40 GHz.
(3) The profile must be monotonically increasing — this can speed up the optimization process by eliminating a class of unrealistic profiles.
(4) The minimum gapwidth is 0.3 mm — The lower limit of the lookup table is 0.3 mm. Indeed the smallest gapwidth that gives $Z_{eff} = 200\,\Omega$ is 0.3915 mm at 25 GHz. At other frequency points the gapwidth must be larger in order to have $Z_{eff} = 200\,\Omega$.
(5) The maximum gapwidth is 2.00 mm — The largest gapwidth that gives $Z_{eff} = 400\,\Omega$ is 1.973 mm at 40 GHz. At other frequency points the gapwidth must be smaller in order to have $Z_{eff} = 400\,\Omega$.

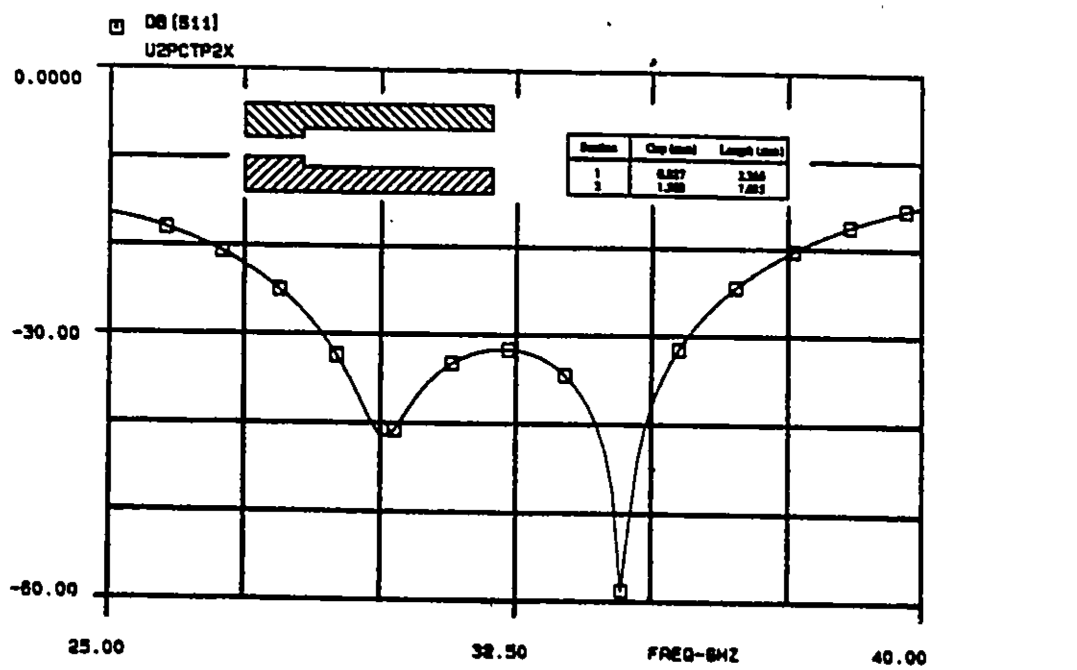
Only constraints 1 and 2 must be included in the circuit file. The remaining constraints are helpful only when the number of steps is large.

† The computation time required to optimize a taper geometry increases exponentially with the number of steps.

‡ The 5-step taper in Figure 5B-5 has 10 variables.

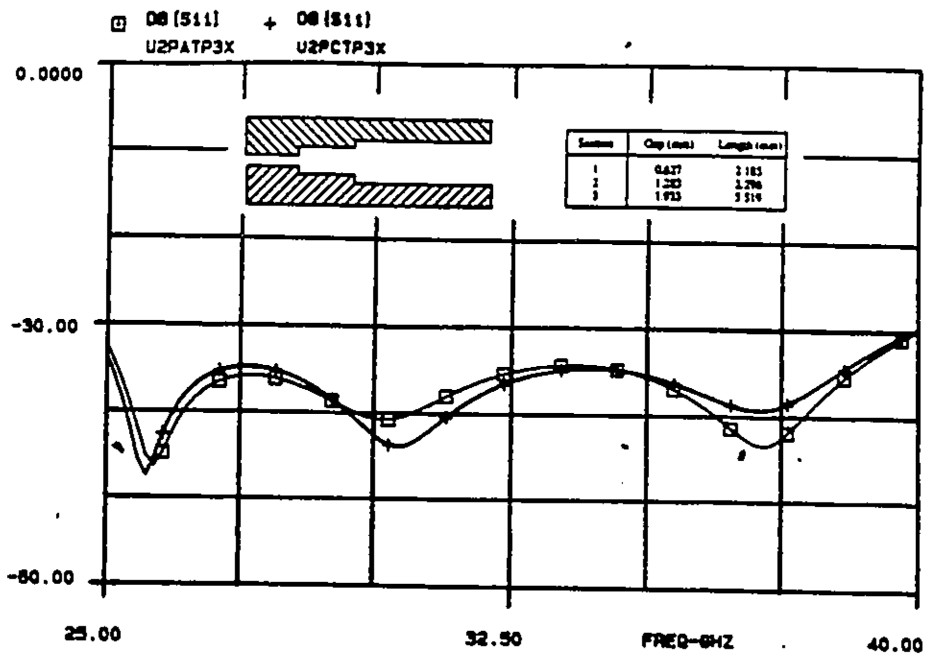


(a): Taper dimensions and return loss from U2PATP2X.CKT and U2PCTP2X.CKT. The step discontinuity effect can be seen clearly.

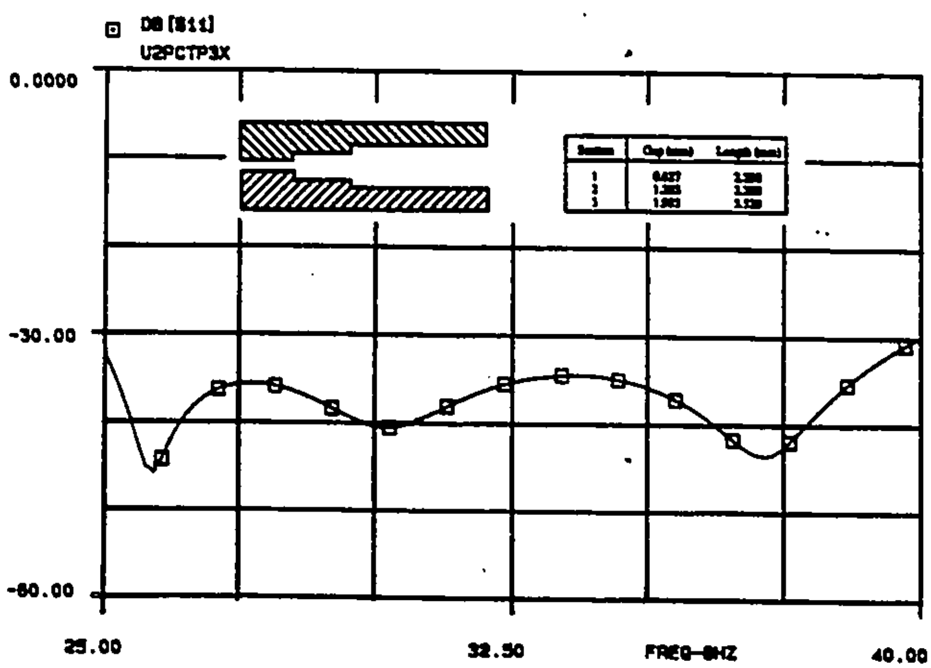


(b): Taper dimensions and return loss from U2PCTP2X.OPT. The dimensions are very close to that in (a).

Figure 5B-2: Dimensions, profile and return loss of three tapers implemented by using 2-step approximations.

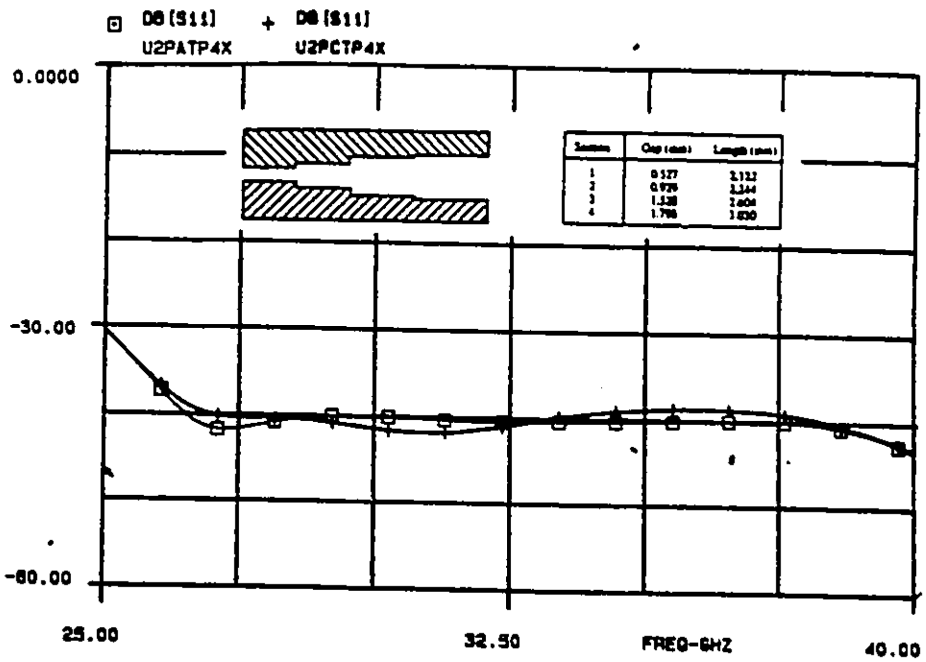


(a): Taper dimensions and return loss from U2PATP3X.CKT and U2PCTP3X.CKT.

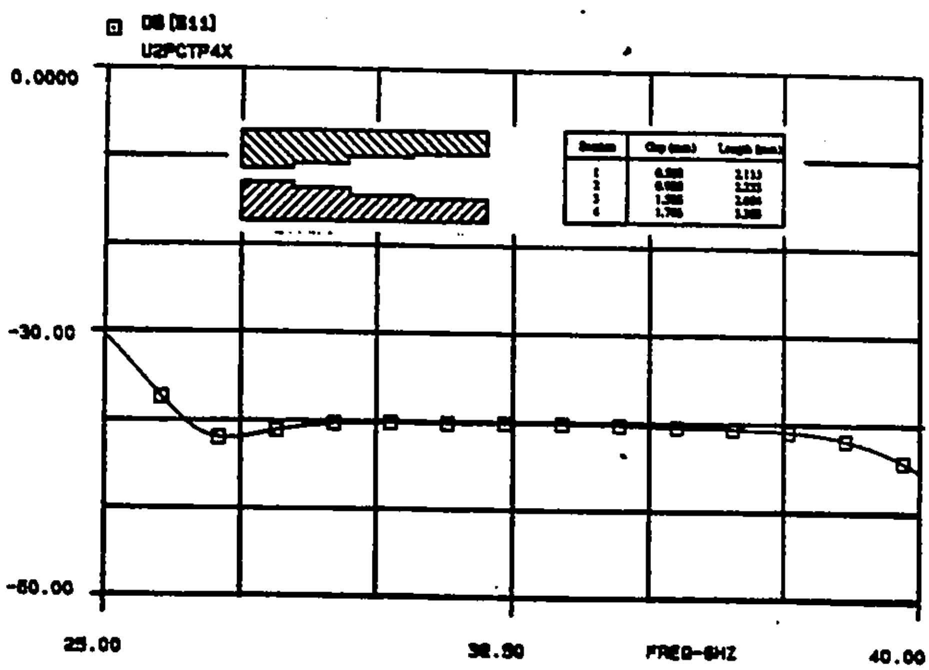


(b): Taper dimensions and return loss from U2PCTP3X.OPT.

Figure 5B-3: Dimensions, profile and return loss of three tapers implemented by using 3-step approximations.

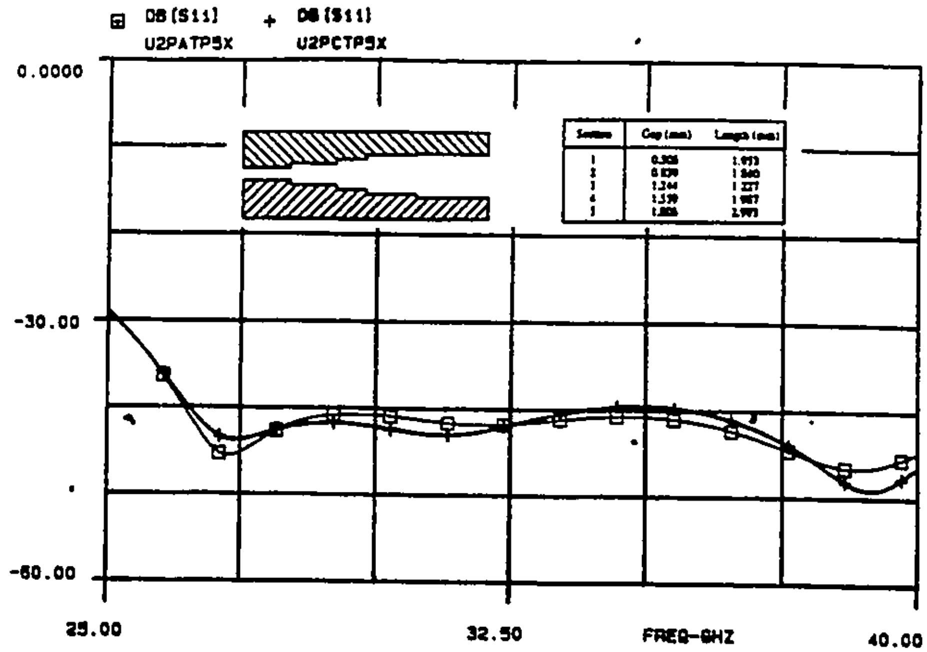


(a): Taper dimensions and return loss from U2PATP4X.CKT and U2PCTP4X.CKT. U2PATP4X meets the return loss specification but there islittle or no error margin.

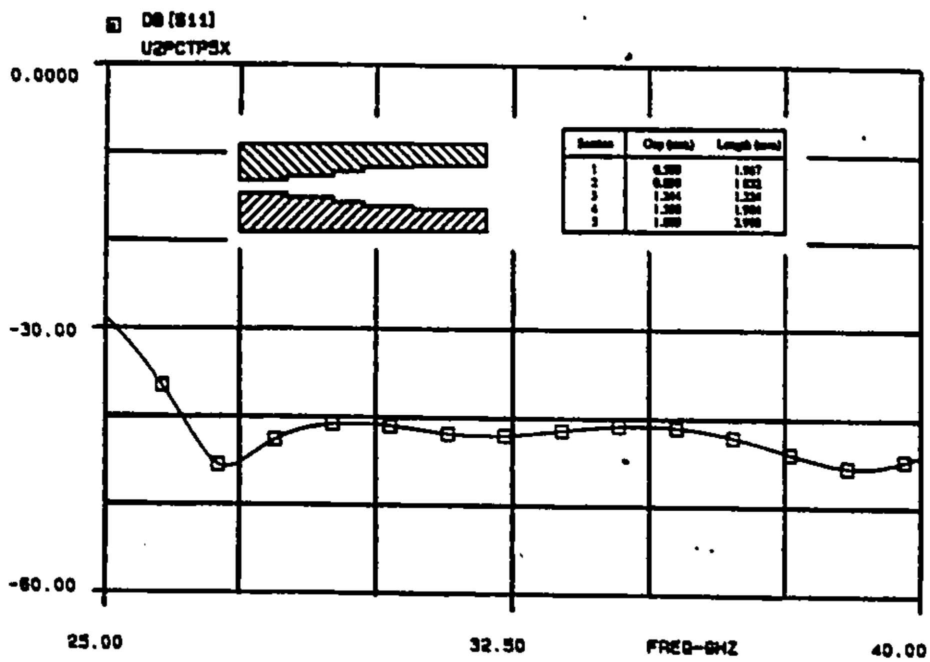


(b): Taper dimensions and return loss from U2PCTP4X.OPT.

Figure 5B-4: Dimensions, profile and return loss of three tapers implemented by using 4-step approximations.



(a): Taper dimensions and return loss from U2PATP5X.CKT and U2PCTP5X.CKT. U2PATP5X meets the return loss requirment with 1 dB error margin.



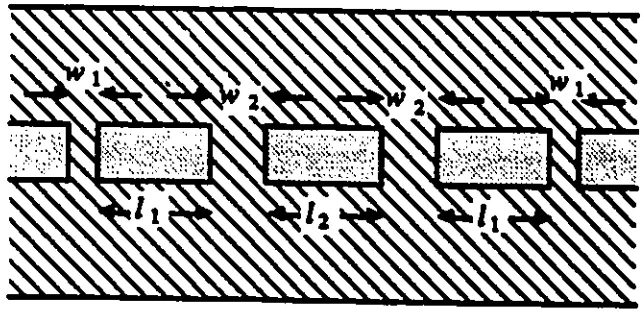
(b): Taper dimensions and return loss from U2PCTP5X.OPT.

Figure 5B-5: Dimensions, profile and return loss of three tapers implemented by using 5-step approximations.

C: Bandpass Filters

Bandpass filters are important circuit components. In a finline environment, they can be realized by using shunting strips. The gapwidth, distance between strips, width of each strip and the number of strips are some important factors that affect the filter characteristics. This section is devoted to comparing computer-simulated results with experimental results.

The filter geometry and dimensions given in Table 5C-1 were provided by the Communications Research Center in Ottawa. These two filters were designed by using empirical methods. The measurement results are shown in Figures 5C-2 to 5C-5 together with the computer-simulated results. The circuit files that described these two filters are given in Appendix J, Listing J19 and J20. Note that element U1PA is used in these circuit files to simulate an input impedance that is similar to that of an ideal taper with a fixed resistive load at one end.



Parameters	Filter 1	Filter 2
w_1	0.1016 mm	0.03048 mm
w_2	0.8890 mm	0.79502 mm
l_1	3.4036 mm	4.3840 mm
l_2	3.4798 mm	4.5466 mm
a	5.588 mm	7.112 mm
b	2.794 mm	3.556 mm
d	0.381 mm	0.508 mm
s	0.128 mm	0.254 mm
h	2.794 mm	3.556 mm
ϵ_r	2.22	2.22

Figure 5C-1: Filter geometry and dimensions.

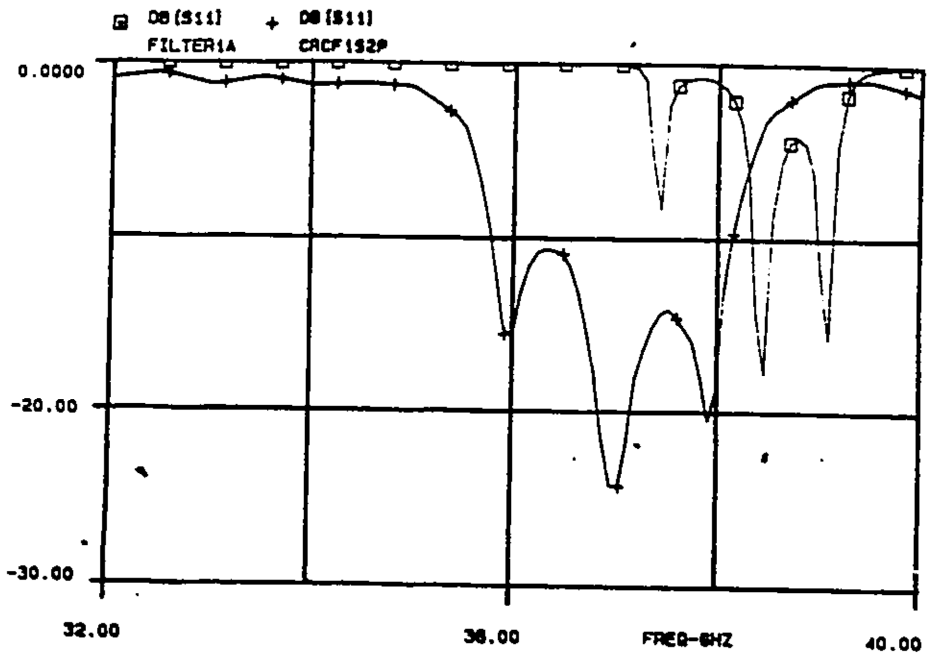


Figure 5C-2: Return loss of filter-1. The curve CRCF1S2P is obtained from measurement and Filter1A is computer-simulated.

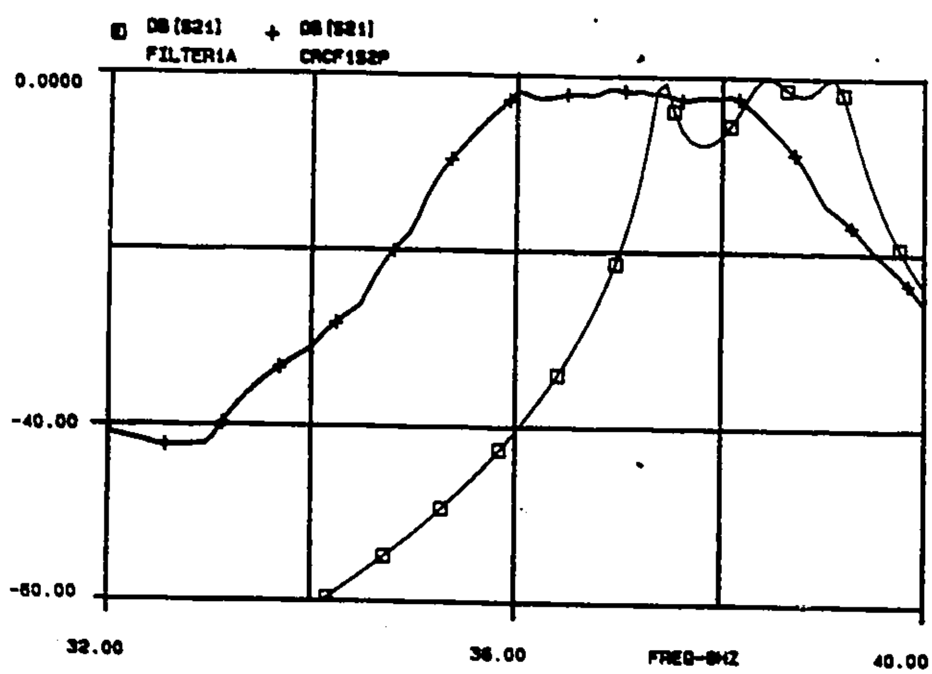


Figure 5C-3: Insertion loss of filter-1. The curve CRCF1S2P is obtained from measurement and Filter1A is computer-simulated.

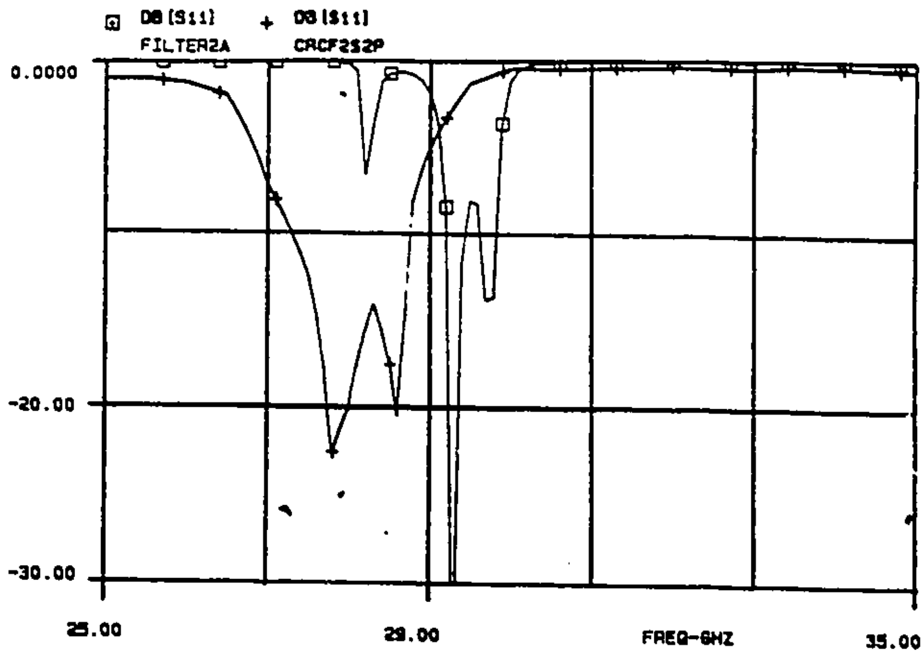


Figure 5C-4: Return loss of filter-2. The curve CRCF2S2P is obtained from measurement and Filter2A is computer-simulated.

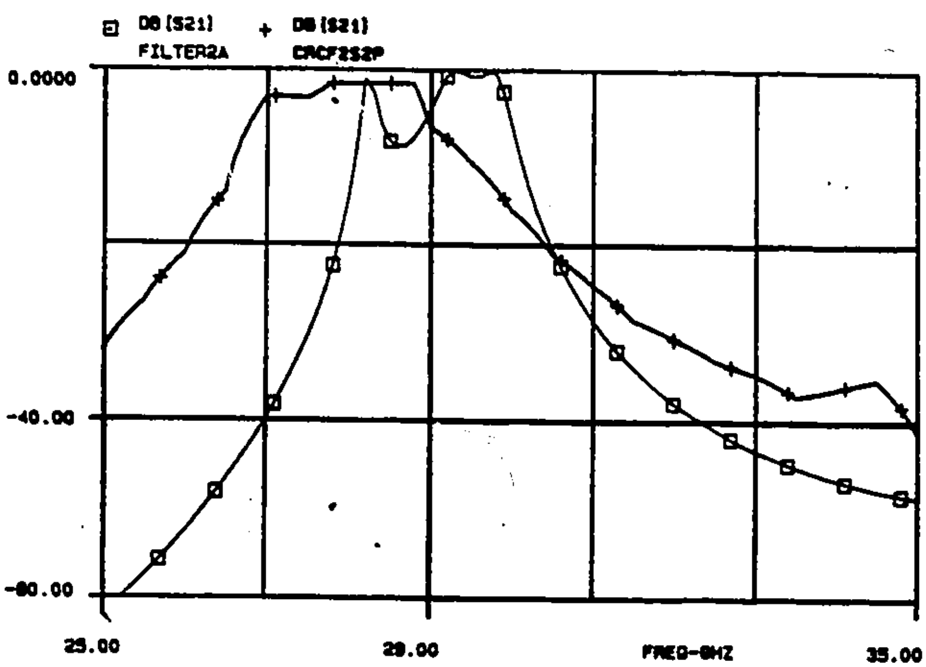


Figure 5C-5: Insertin loss of filter-2. The curve CRCF2S2P is obtained from measurement and Filter2A is computer-simulated.

Figures 5C-2 to 5C-5 show that the CAD results are similar to the experimental results. The CAD results have approximately 1 GHz up shift in frequency. The reflection at the lower pass band is also large in comparison to the reflection at the upper pass band. This suggests that the shunt discontinuity element, U2PB, does not model the discontinuity effect accurately enough. Indeed, the shunt discontinuity model is not accurate for small strip widths‡. Both filters simulated in this section have two very narrow stubs at the two ends; this is a major source of error. The next source of error is that higher order mode interactions are not accounted for by the shunt model or by this CAD procedure.

‡ See Chapter 7: Limitations of this CAD Procedure.

CHAPTER 6 Comparison of CAD Results Obtained by Using 3-by-11 and 11-by-11 Lookup Tables

As mentioned in Chapter 3, two different interpolation schemes which are based on 3–by–11 and 11–by–11 lookup tables have been implemented in this novel CAD procedure. Some results from these two interpolation methods are compared in Figure 6-1 to 6-3. The curves labelled "A311" are generated by using 3–by–11 lookup tables and the proposed new interpolation method, while the curves labelled "DAT" are generated by 11–by–11 lookup tables and straightforward linear interpolation. The following sequence of diagrams shows that the innovative 3–by–11 interpolation scheme indeed gives very accurate results. Therefore this 3–by–11 interpolation scheme can be used in most unilateral and bilateral finline applications.

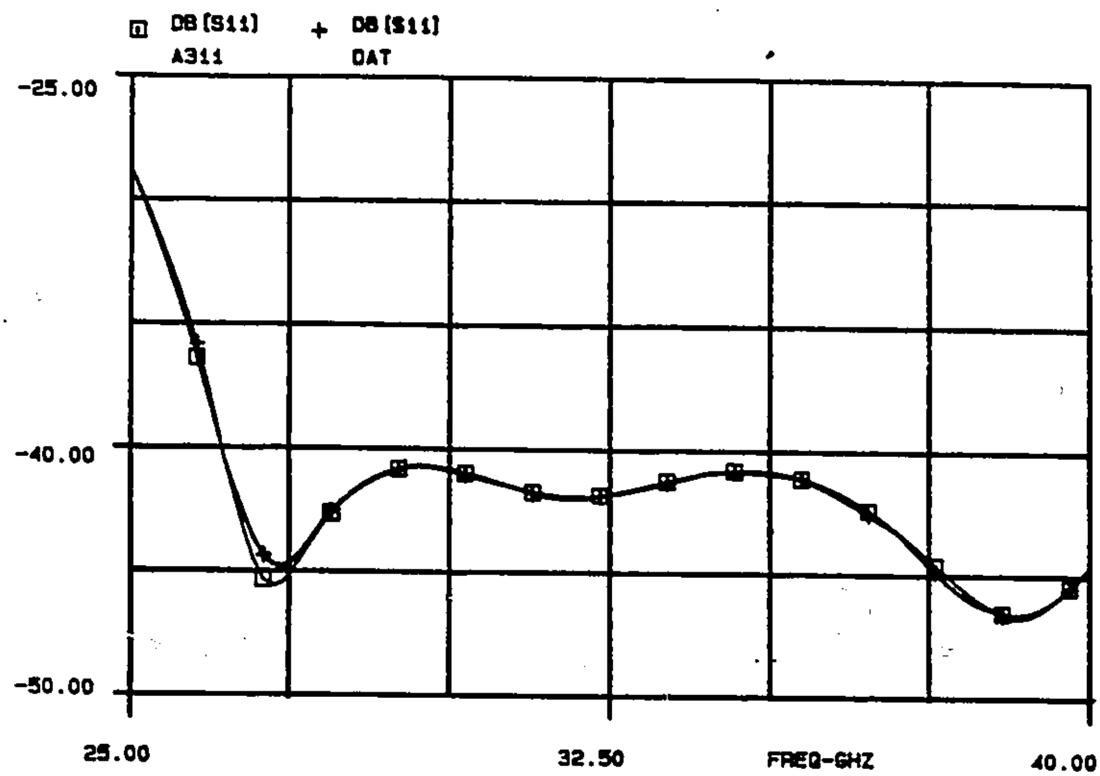
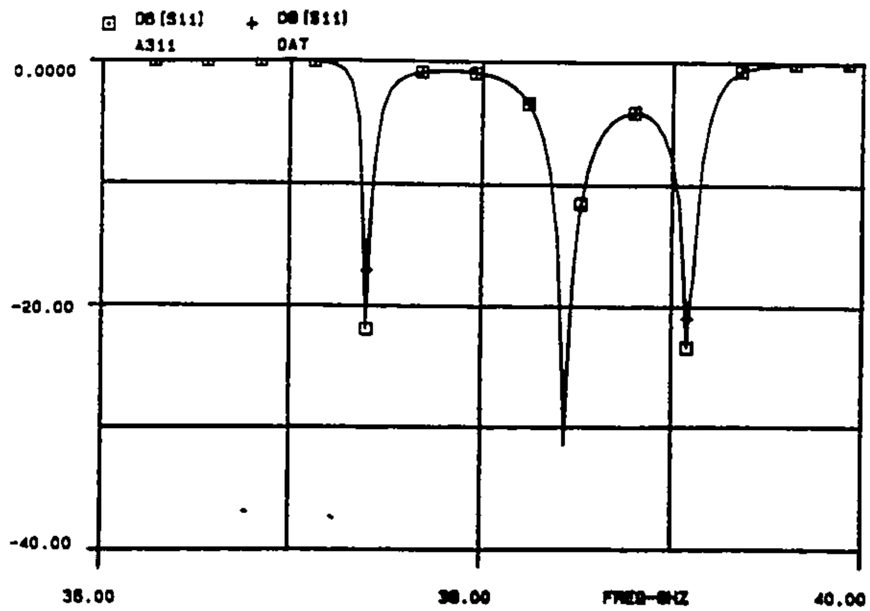
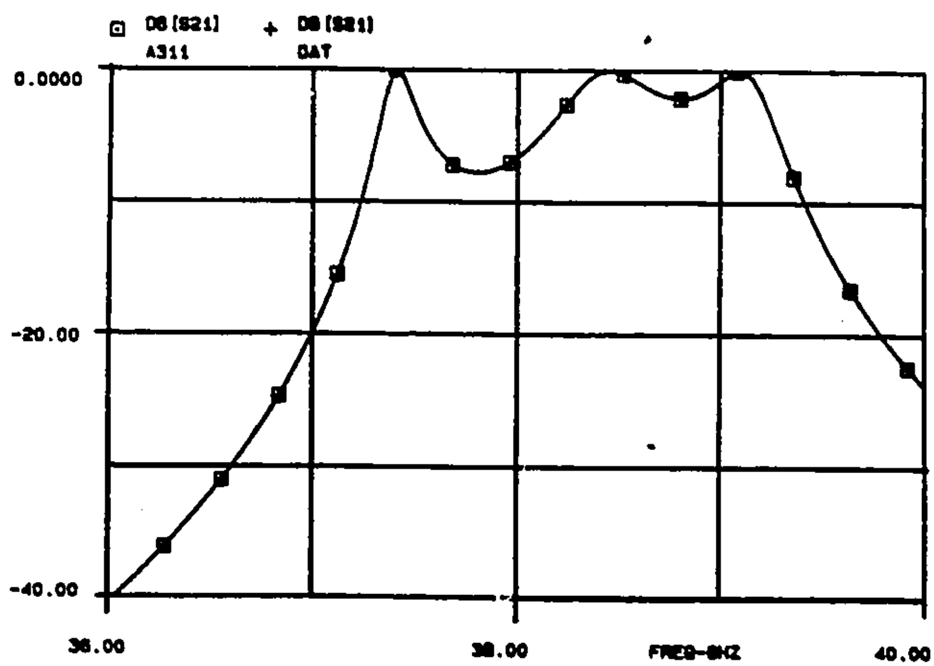


Figure 6-1: Comparison of the return loss curves of a taper (U2PCTP5X.OPT in Chapter 5) generated by using the 3–by–11 and 11–by–11 interpolation schemes.

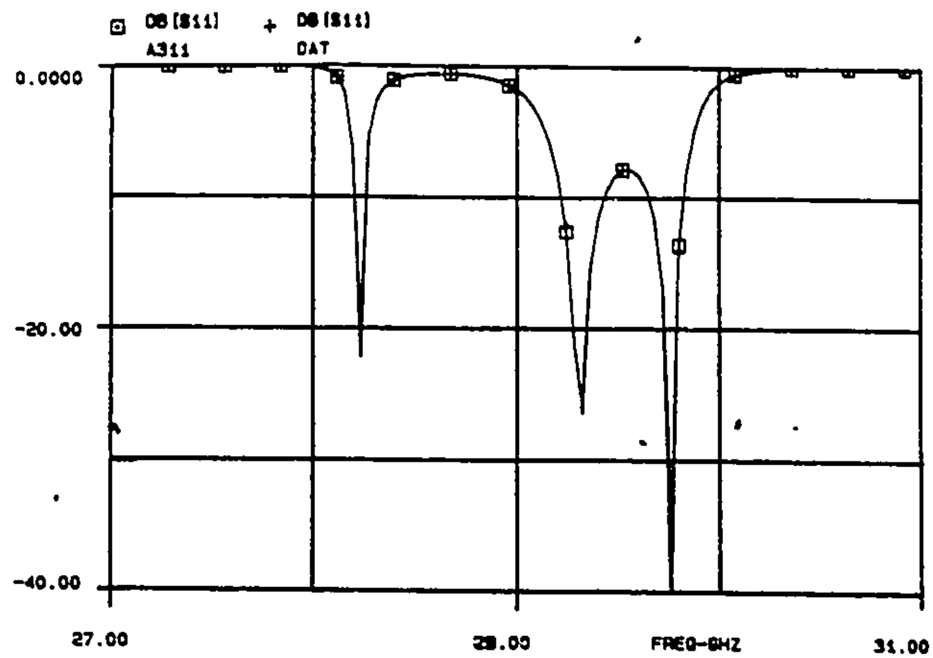


(a): Comparison of two return loss curves

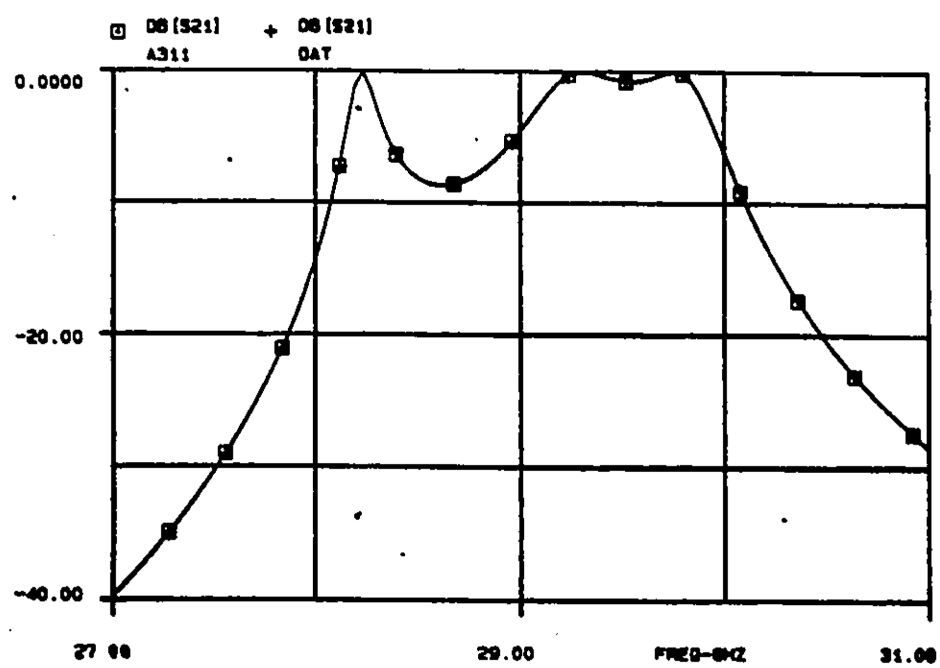


(b): Comparison of two insertion loss curves

Figure 6-2: Comparison of the return loss and insertion loss curves of Filter-1 in Chapter 5 which are generated by using the 3-by-11 and 11-by-11 interpolation schemes.



(a): Comparison of two return loss curves



(b): Comparison of two insertion loss curves

Figure 6-3: Comparison of the return loss and insertion loss curves of Filter-2 in Chapter 5 which are generated by using the 3–by–11 and 11–by–11 interpolation schemes.

CHAPTER 7 Limitations and Experimental Verification of the Discontinuity Models

A: Limitations

The discontinuity models implemented in this CAD procedure are based on a number of empirical expressions and field-theory based lookup tables. Since there are too many variable parameters ($a, b, d, s, h, \epsilon_{eff}$ and f) to be considered, the discontinuity-model expressions can only cover a narrow variable-range.

The shunt discontinuity model is valid for both unilateral and bilateral finlines. The validity range of w/b is (0..1), i.e. $0 < w/b < 1$. The shunt susceptance, B, for $w/b \leq 0.4$ is given by [3]:

$$B = 0.455(\lambda_g/b)(d/b)^{-0.715} + 326(1 - 1.3b/\lambda_g)^3(w/b)exp(-2.63d/b)$$

For w=0, B=$0.455(\lambda_g/b)(d/b)^{-0.715}$, which corresponds to the shunt inductance of a thin wire of zero thickness.

The step discontinuity model is only good for unilateral finline and it is not accurate at small step size. The step inductance is given by [4]:

$$L \text{ (in pH)} = 19.14 - 31.275(d_2/d_1) + 14.56(d_2/d_1)^2 - 0.5014(d_2/d_1)^3$$

For d_2=d_1, L=1.9236pH$\neq$0. Hence in the limiting case, d_2=d_1, this model is completely inaccurate. However, the value for L is so small that it introduces no measurable error in a practical application. Other constraints of this model are:

$$s/a = 1/28$$
$$b/a = 1/2$$
$$\epsilon_r = 2.2$$
$$0.32 \leq \frac{b}{\lambda} \leq 0.47$$

The lookup tables are generated by theory-based programs which are sufficiently accurate for most practical applications, but cannot account for higher-order mode interactions among circuit elements. This is a common problem for most commercially available microwave CAD programs. One way to solve this problem is to use the generalized n-port scattering matrix. This requires more complicated discontinuity models. Another way to attack this problem is to generate lookup tables at the component level rather than at the element level. This would improve the accuracy of the result, but more sophisticated table generation programs are needed.

B: Experimental Verification

Since the discontinuity-model expressions and the lookup tables are not perfectly accurate, a natural question to ask is "How much error would be introduced into the simulation result by the circuit models?" The experimental results presented in Chapter 5 have shown that the center frequency of a bandpass filter obtained by this simulation procedure was 1 GHz higher than the measured value, 28 GHz, i.e. 3.57% higher. Since the filter response was simulated by cascading four shunt elements (U2PB), it is logical to conclude that each shunt element may introduce approximately 0.9% error. However, this can only be verified by measuring the characteristics of individual discontinuities and comparing them with their analytical model.

There are at least two ways to verify the accuracy of the models. One is to perform swept measurements with an automatic network analyzer. This method is very attractive but the instrumentation is complicated. It also requires high quality transitions in order to obtain reasonably accurate results. An other method is to measure the shift of resonant frequencies of a cavity containing the discontinuities. This second method was chosen for experimental verification because it requires only simple instrumentation and gives high sensitivity. A disadvantage of this method is that only discrete resonant frequencies can be obtained.

The shape and dimensions of the resonant cavity used in the experiment is depicted in Figure 7B-1. The large size of the cavity allows a good control of the dimensional tolerances. The grooves in the middle of the cavity cover hold the finline substrate. The various circuit elements used in the experiment are depicted in Figures 7B-2 to 7-4. Measurements were performed to obtain the resonant frequencies of the cavity with and without these elements. The apparatus used in the experiment is listed in Table 7B-1.

#	quantity	item	serial #
1	1	Frequency counter EIP model 545A	00257
2	1	HP8690B sweep oscillator	959–02048
3	1	HP182T oscilloscope	1507A00135
4	1	xy-plotter	
5	1	dual directional coupler 776D(0.940–1.900GHz)	
6	1	dual directional coupler 797D(1.900–4.100 GHz)	
7	1	detector 1166A	1413A02790
8	1	modulator 11665B	00484
9	1	oversize rectangular cavity (Figure7B-1)	
10	1	unilateral finline element (Figure 7B-2)	
11	1	step discontinuity elements (Figure 7B-3)	
12	1	shunt discontinuity elements (Figure 7B-4)	

Table 7B-1: A listing of the major apparatus used in the experiment.

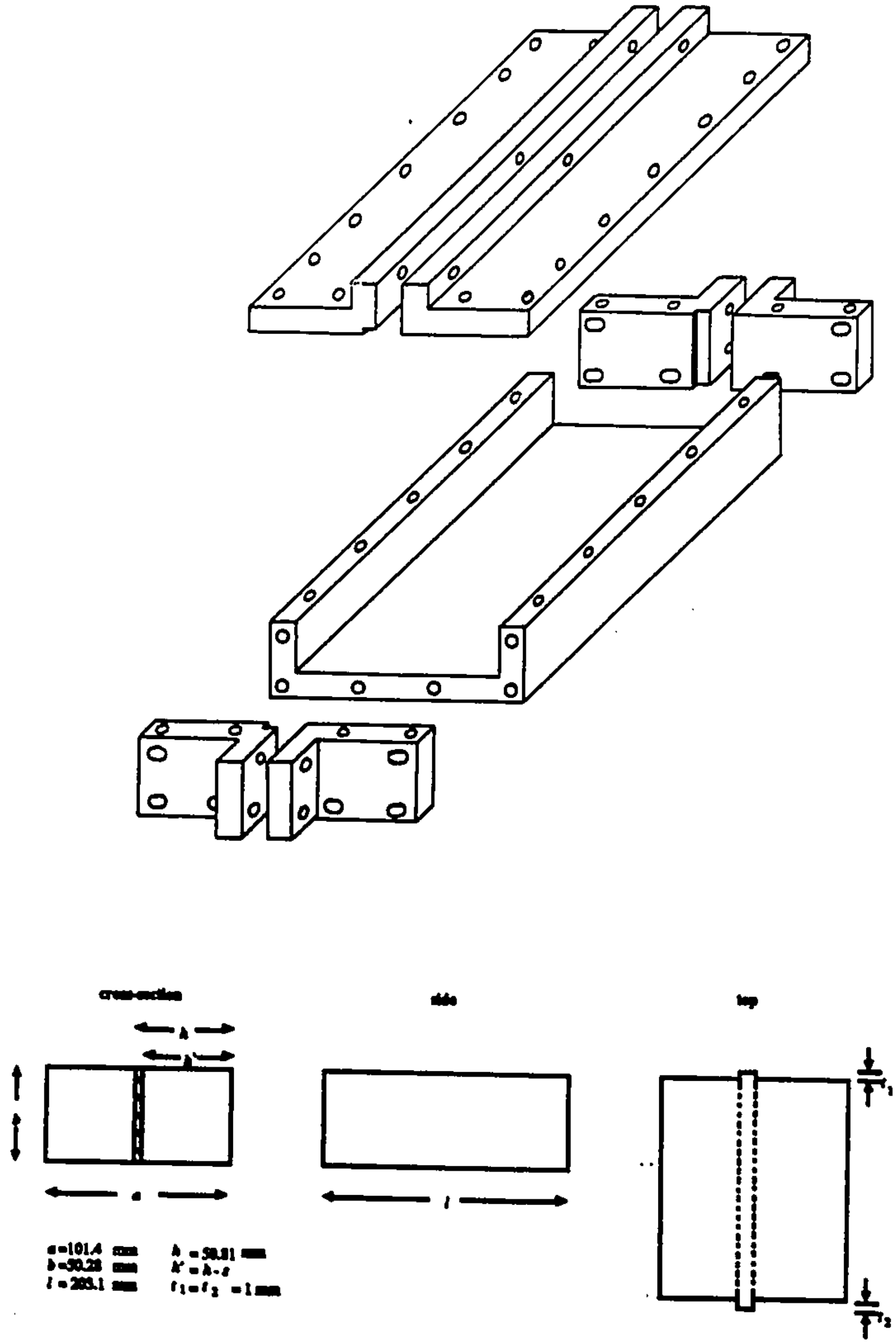


Figure 7B-1: The cavity used in the experiment.

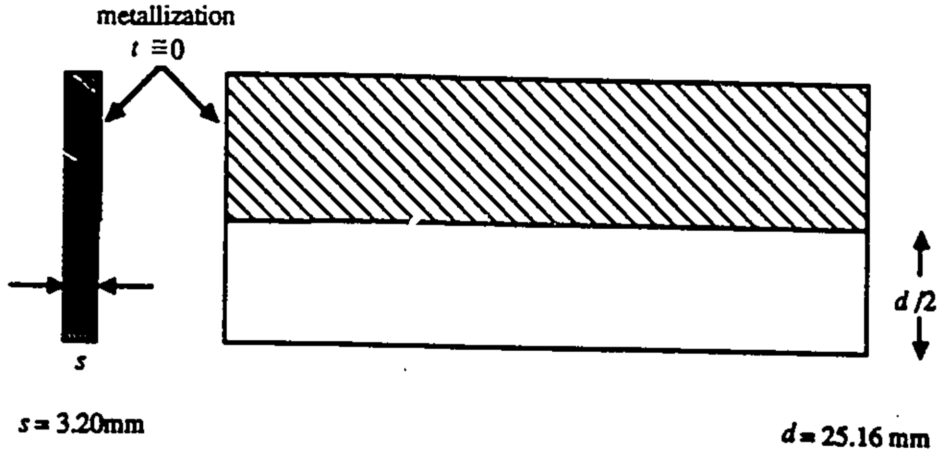
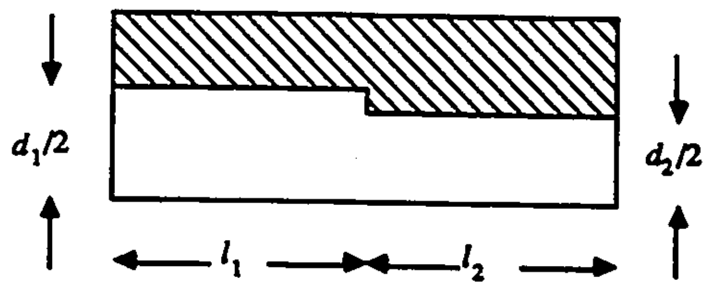
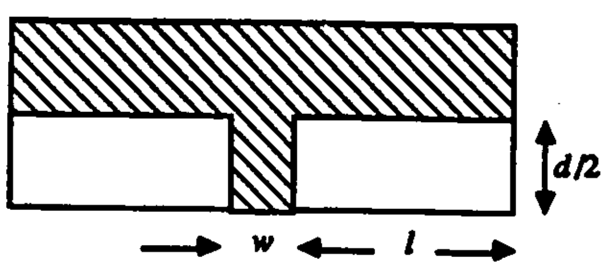


Figure 7B-2: The line-section used in the experiment.



	1	2	3	4
s	3.175	3.135	3.220	3.220
d_1	5.890	12.68	5.840	12.53
d_2	2.970	3.790	2.640	3.400
l_2	101.1	101.5	50.73	50.50

Figure 7B-3: The four step-discontinuities used in the experiment.



	1	2
s	3.225	3.200
d	6.070	12.86
l	100.0	99.55
w	5.070	5.185

Figure 7B-4: The two shunt-discontinuities used in the experiment.

B.1: Empty Cavity

The apparatus was set up as shown in Figure 7B-5. The sweeper was initially set to a sweep mode for 0–2GHz (and then for 2–4GHz). The output power was increased gradually until resonant peaks could be seen on the scope. The transmission characteristic was then recorded by using the xy-plotter. The theoretical resonant frequencies were calculated first; then the sweeper was set to the manual-mode and the resonant frequencies in the vicinity of the theoretical values were recorded precisely by using the frequency counter. This procedure was repeated 10 times for each frequency of interest. These repetitive steps are necessary in order to avoid non-systematic error due to the uncertainty in setting the frequency to the maximum of the resonance peaks, see Figure 7B-6. The means of the resonant frequencies $f_{101}, f_{102}, f_{103}, f_{104}$ and f_{105} are given in Table 7B-2 together with the theoretical values.

The equivalent circuit in Figure 7B-7 was used to perform simulations (see Appendix I for simulation procedure). The simulation results are summarized in Table 7B-2 and Figure 7B-8.

The good agreement among the theoretical, measurement and simulation results shows that:

(1) The small mounting groove at the middle of the cavity cover does not sensibly affect the resonant frequencies of the empty cavity.

(2) The loading effect due to the input and output coupling loops is negligible.

(3) The equivalent circuit in Figure 7B-7 is suitable for resonant frequency simulation.

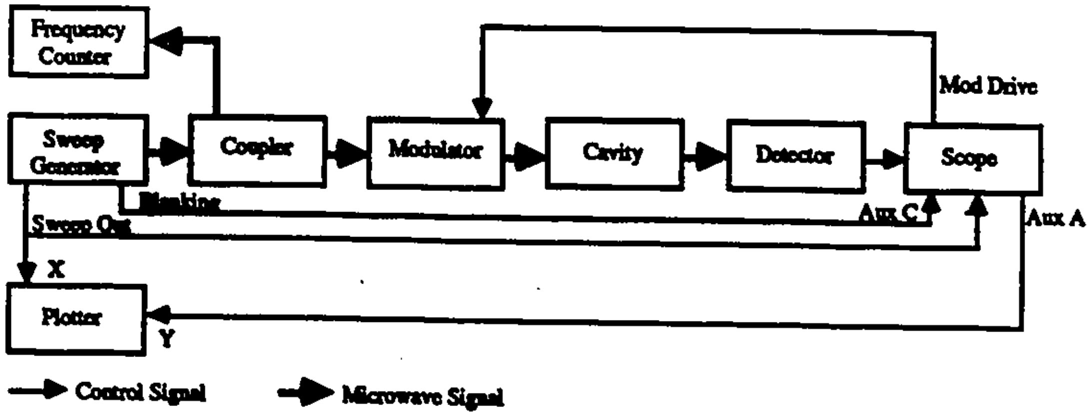


Figure 7B-5: The experimental arrangement used in the experiment. The resonant frequencies were found in the sweep-mode and then carefully measured in the CW-mode by using the microwave frequency counter.

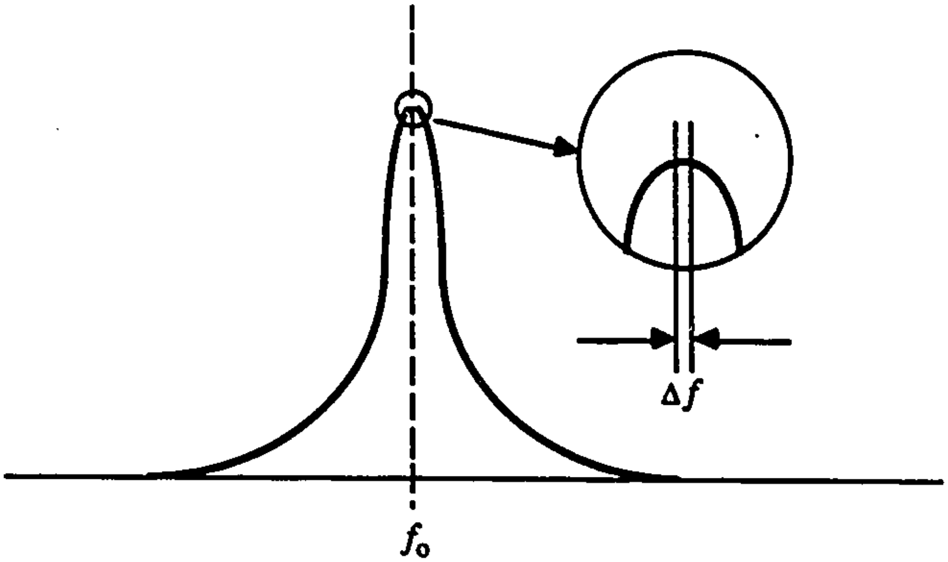


Figure 7B-6: A sample resonant peak pattern. The maximum of the resonance curve is not well defined: $f_{res} = f_o \pm \Delta f$. In order to eliminate non-systematic error, the measurement was repeated 10 times for each resonant freqency.

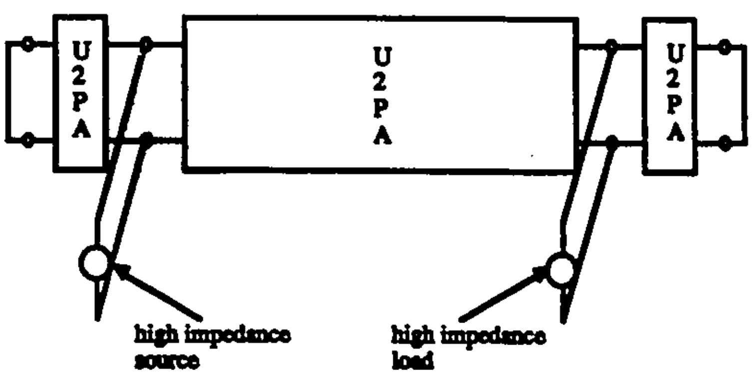


Figure 7B-7: The equivalent circuit used to produce the simulation result in Table 7B-2. See Chapter 3 (Figure 3–5) and Appendix H (Figure H–2) for information about U2PA.

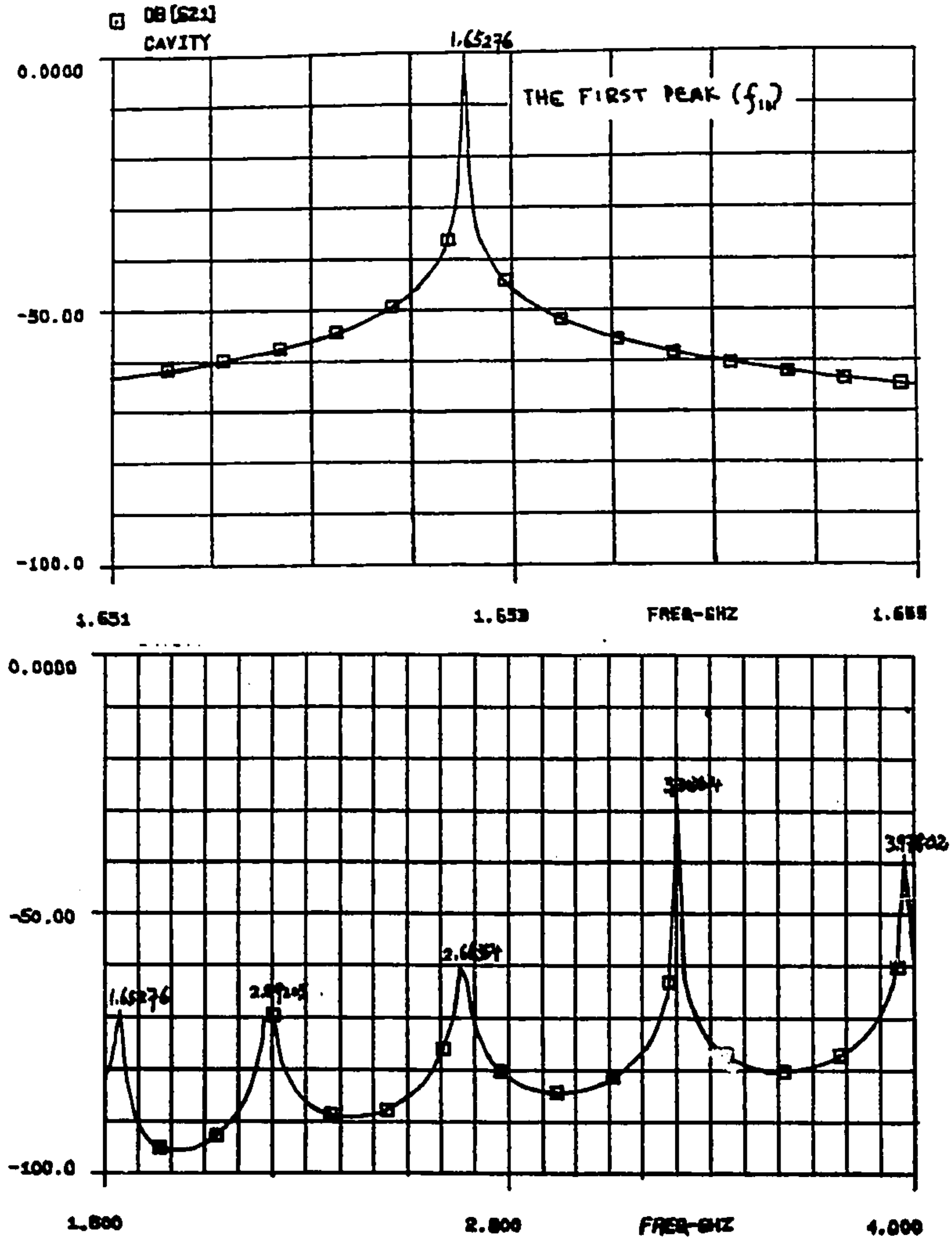


Figure 7B-8: The simulated transmission curve. Since loss is not accounted for by the equivalent circuit in Figure 7B-7, the numerical value of the amplitude of the transmission curve is arbitrary.

Measurement	Theory	Difference in %
1.651	1.653	-0.151
2.086	2.090	-0.200
2.657	2.664	-0.254
3.303	3.304	-0.026
3.967	3.978	-0.237

Measurement	Simulation	Difference in %
1.651	1.653	-0.119
2.086	2.092	-0.028
2.657	2.666	-0.308
3.303	3.304	-0.017
3.967	3.978	-0.234

Table 7B-2: The first five resonant frequencies, $f_{101}, f_{102}, f_{103}, f_{104}$ and f_{105} of the empty cavity in Figure 7B-1 obtained by three different methods — measurement, theory and simulation. The good agreement among these methods shows that the effect of the mounting grooves and of the coupling loops on the resonant frequencies are negligibly small.

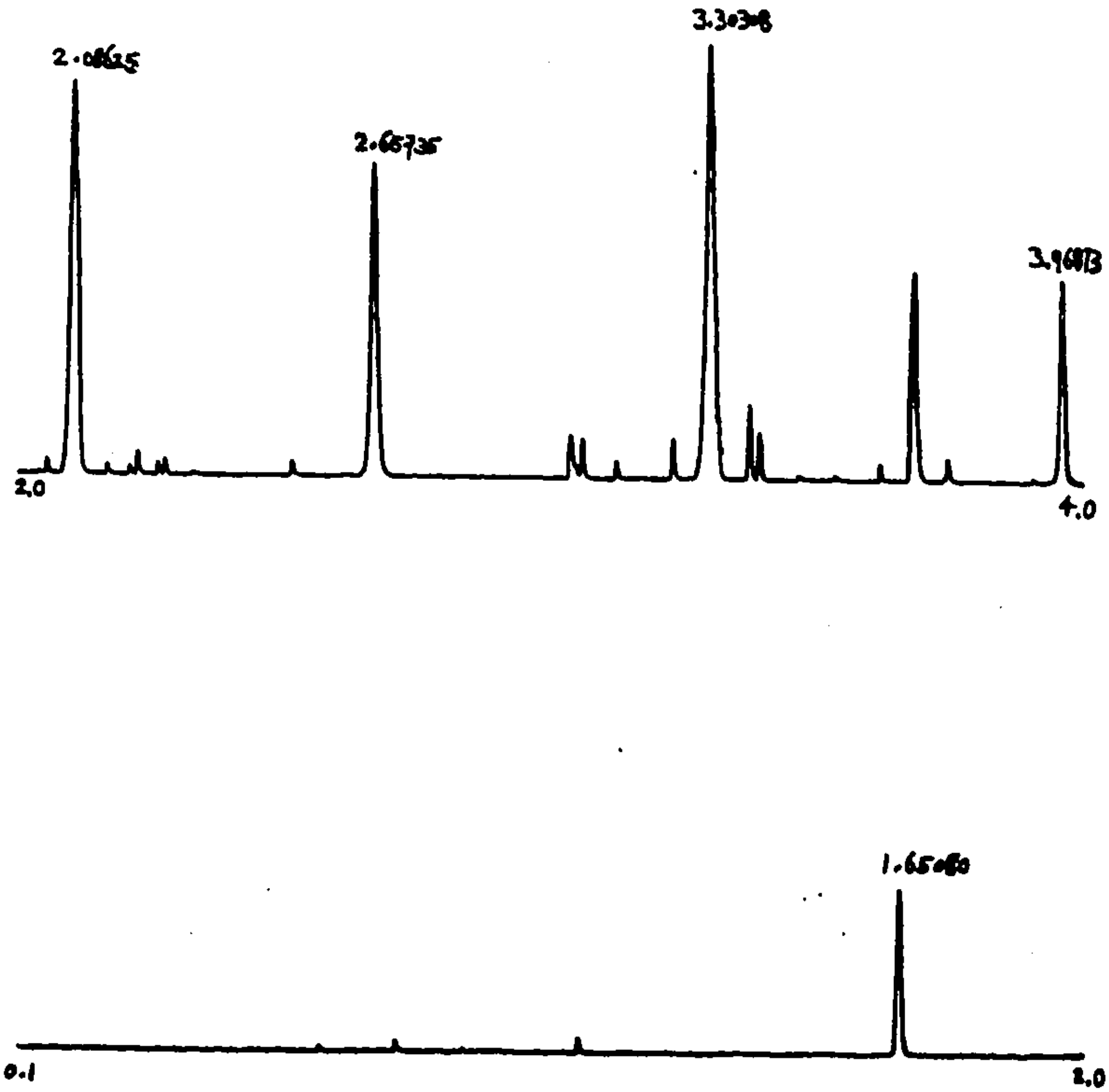


Figure 7B-9: The measured transmission characteristics of the empty cavity.

B.2: Cavity with Discontinuity Elements

The first five resonant frequencies of the cavity with discontinuity elements were first determined by simulation. The discontinuity elements were then placed in the cavity and the resonant frequencies in the vicinity of the simulated values were recorded precisely as described in (B.1). The results are depicted in Table 7B-3. Note that because the groove is not located right at the center of the cavity, the fin position can be in center or off-center, see Figure 7B-10.

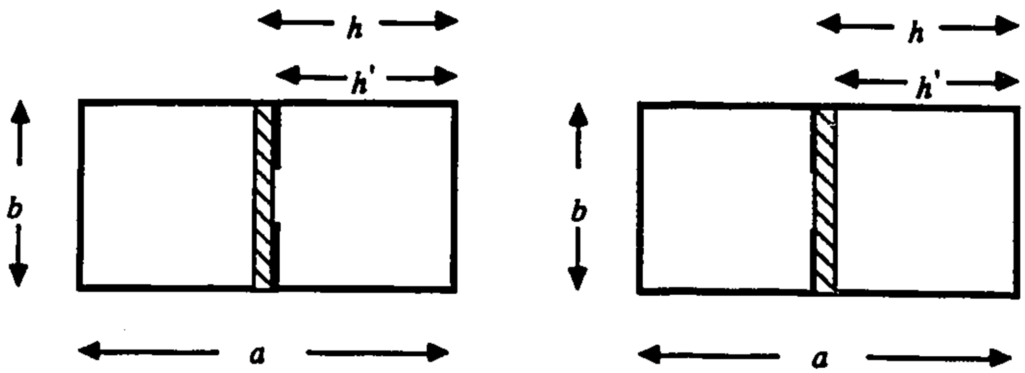


Figure 7B-10: The two possible ways of placing the finline elements in the cavity, with the metalization surface at position h' or to place it at position h.

UFIN at h'

Measurement	Simulation	Difference in %
1.419	1.406	0.953
1.852	1.851	0.084
2.403	2.406	-0.134
3.001	3.015	-0.460
3.636	3.646	-0.269

UFIN at h

Measurement	Simulation	Difference in %
1.417	1.404	0.946
1.850	1.849	0.045
2.401	2.405	-0.158
3.000	3.014	-0.469
3.638	3.645	-0.188

STEP1 at h'

Measurement	Simulation	Difference in %
1.040	1.052	-1.110
1.538	1.532	0.385
2.098	2.099	-0.045
2.694	2.699	-0.172
3.303	3.306	-0.107

STEP1 at h

Measurement	Simulation	Difference in %
1.038	1.051	-1.244
1.534	1.531	0.201
2.095	2.098	-0.143
2.692	2.699	-0.269
3.306	3.306	0.001

Table 7B-3: The first five resonant frequencies $f_{101}, f_{102}, f_{103}, f_{104}$ and f_{105} of the cavity with the finline elements. The values are obtained by measurement and by simulation. The accuracy of the elements U2PA, U2PB and U2PC are better than 1.0%, 4.0% and 1.5% respectively.

STEP2 at *h'*

Measurement	Simulation	Difference in %
1.100	1.091	0.835
1.625	1.627	-0.170
2.151	2.158	-0.355
2.765	2.788	-0.843
3.353	3.373	-0.615

STEP2 at *h*

Measurement	Simulation	Difference in %
1.097	1.088	0.882
1.623	1.627	-0.210
2.145	2.158	-0.417
2.764	2.788	-0.845
3.373	3.373	0.011

STEP3 at *h'*

Measurement	Simulation	Difference in %
1.102	1.117	-1.357
1.548	1.542	0.360
2.091	2.091	0.009
2.714	2.716	-0.064
3.351	3.350	0.012

STEP3 at *h*

Measurement	Simulation	Difference in %
1.100	1.115	-1.418
1.548	1.541	0.451
2.092	2.091	0.070
2.716	2.716	-0.003
3.359	3.350	0.269

STEP4 at *h'*

Measurement	Simulation	Difference in %
1.238	1.245	-0.568
1.629	1.631	-0.095
2.154	2.169	-0.709
2.803	2.823	-0.720
3.462	3.482	-0.574

STEP4 at *h*

Measurement	Simulation	Difference in %
1.236	1.243	-0.621
1.627	1.630	-0.164
2.151	2.168	-0.791
2.802	2.823	-0.760
3.508	3.481	-0.765

SHUNT1 at *h'*

Measurement	Simulation	Difference in %
1.506	1.541	-2.324
1.585	1.585	-0.003
2.587	2.631	-1.739
2.764	2.763	0.048
3.625	3.761	-3.771

SHUNT1 at *h*

Measurement	Simulation	Difference in %
1.505	1.541	-2.371
1.582	1.584	-0.118
2.590	2.631	-1.605
2.758	2.762	-0.165
3.694	3.761	-1.826

SHUNT2 at *h'*

Measurement	Simulation	Difference in %
1.595	1.643	-3.009
1.705	1.702	0.145
2.630	2.685	-2.074
2.874	2.875	-0.048
3.651	3.790	-3.813

SHUNT2 at *h*

Measurement	Simulation	Difference in %
1.594	1.642	-2.994
1.703	1.701	0.117
2.638	2.684	-1.756
2.874	2.875	-0.045
3.695	3.790	-2.561

Table 7B-3 (con't): The first five resonant frequencies $f_{101}, f_{102}, f_{103}, f_{104}$ and f_{105} of the cavity with the finline elements.

Conclusion

The first five resonant frequencies of an empty cavity were measured. The experimental values were compared with the values obtained by both analytical formulae and computer simulation. Since the agreement among the values from all three methods was excellent it was concluded that the effect of the mounting grooves and of the coupling loops on the resonant frequencies are negligible small.

The same procedure was employed to measure the resonant frequencies of the cavity containing the discontinuities. The results show that the models for the elements U2PA, U2PB and U2PC permit us to predict the resonant frequencies of the loaded cavity within 1%, 4% and 1.5% respectively. Note that the cavity is a narrowband device, which is much more sensitive to small inaccuracies in the models than wideband ciruits, where the discontinuity effects are only of the second order. Therefore, the modeling errors are acceptable in the vast majority of practical applications. In devices such as narrow bandpass filters, higher-order mode interaction between discontinuities becomes dominant, and a more comprehensive analysis must be undertaken anyway.

CHAPTER 8

Conclusion

A novel CAD procedure for E-plane circuits has been presented. This procedure is based on E-plane circuit elements which are implemented by using simple discontinuity models and field-theory based lookup tables. The lookup tables are generated by using two accelerated Spectral Domain programs for unilateral and bilateral finlines. A spectral domain program for antipodal finline has not been developed, however a theoretical formulation for such a program is given. Two different interpolation schemes have been developed to extract data from the lookup tables. The first scheme is based on physically realistic functions and requires lookup tables of size 3–by–11. The second one is a straightforward linear interpolation scheme which requires lookup tables of size 11–by–11.

A number of E-plane circuit elements have been implemented. They are straight finline sections, shunt discontinuities, step discontinuities and wideband matching loads. The accuracy of these elements has been verified by measurement. These elements are used to simulate transformers, tapers and bandpass filters. The transformer and taper CAD results revealed the importance of dispersion and discontinuity effects of finline circuit components. The filter CAD results are in good agreement with the experimental results. Some CAD results obtained by using the two interpolation schemes are also compared and they are in excellent agreement with each other.

Limitations of this E-plane CAD procedure are discussed. These limitations must be respected while using this CAD procedure. Suggestions for further improvement are also given. Those improvements should be implemented in order to enhance the performance of this CAD procedure. The accuracy of each individual element has also been verified by resonant frequency measurements.

All the finline circuit elements developed in this thesis are integrated in Touchstone™ Senior. Therefore the proposed CAD procedure takes advantage of all the capabilities inherent in this commercially available software. In fact, this novel CAD method expands the power of commercially available microwave CAD software into the E-plane circuit domain and replaces the conventional trial and error method for E-plane circuit design with a systematic design and optimization procedure.

APPENDIX A

UniFastS.Pas and *BiFastS.Pas* are two lookup table generation programs for unilateral and bilateral finlines. These programs are the modified versions of *UniFast.Pas* and *BiFast.Pas* — two accelerated Spectral Domain programs developed by a group of persons headed by Dr. Hoefer. Since Spectral Domain method is not the theme of this thesis and *UniFastS.Pas* and *BiFastS.Pas* are two fairly big programs, their source codes are not included here. The purpose of this appendix is to demonstrate the usefulness of these two programs. The format of the lookup tables output from these programs is given in Appendix B.

UniFastS.Pas is a versatile table generation program. The input menu in Figure A1 shows that there are seven variables; two of the seven variables can be chosen as sweeping variables. Figure A2 depicts a graphical image of Z_{eff} versus f and h which is given by an 11-by-11 lookup table generated by this program. This lookup table would be useful if it is required to design a transmission section having uniform gapwidth but non-uniform metalization distance from the waveguide wall.

BiFastS.Pas is similar to *UniFastS.Pas* in the user's point of view. The only difference is that *BiFastS.Pas* fixed the substrate at the middle of the waveguide. Figure A3 shows an input menu of *BiFastS.Pas*, note that h is missing from the option menu. Figure A4 depicts a graphical image of ϵ_{eff} versus f and s which is given by an 11-by-11 lookup table generated by this program. Such a lookup table would be useful if it is required to design a transmission section having uniform gapwidth but non-uniform substrate thickness.

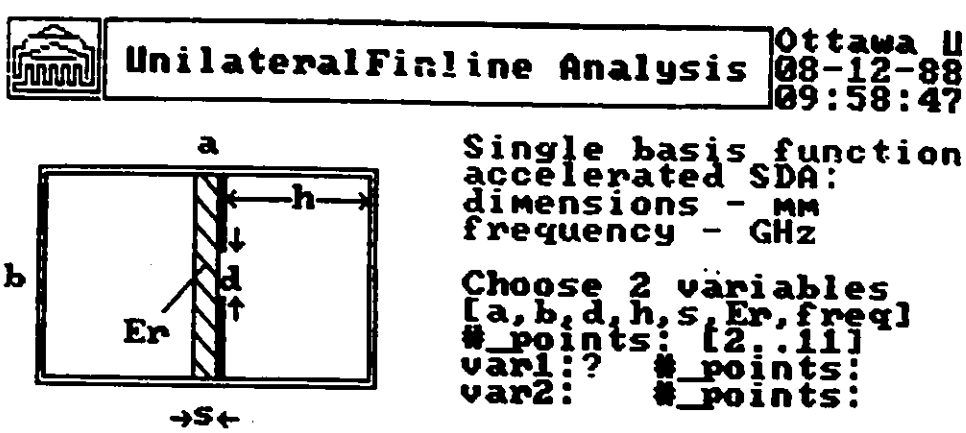


Figure A1: An *UniFastS.Pas* input menu.

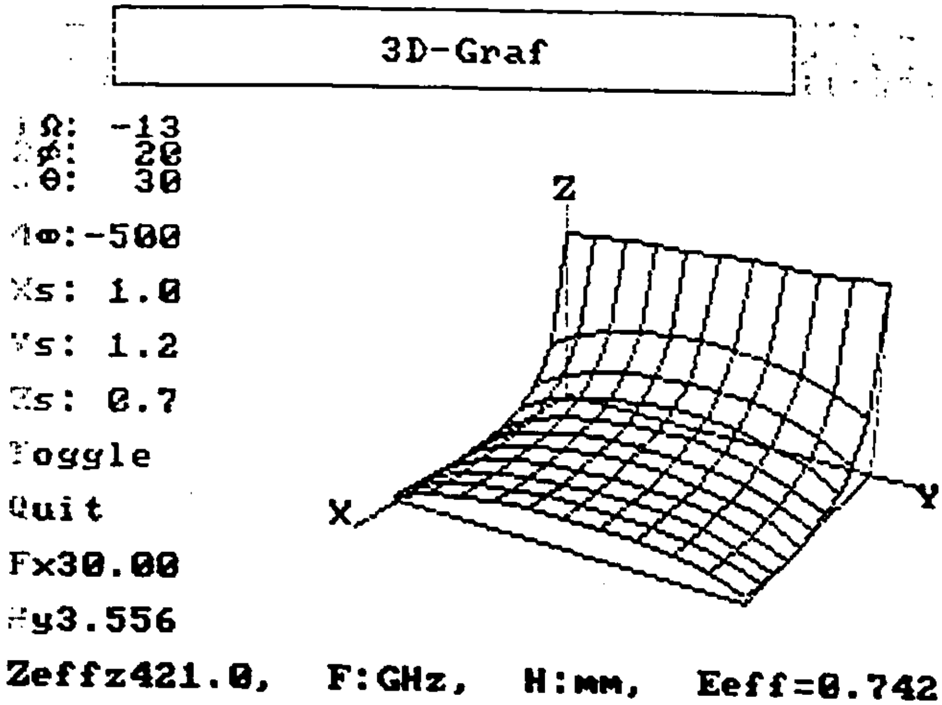


Figure A2: A graphical image of Z_{eff} versus f and h from an 11-by-11 lookup table which is generated by *UniFasts.Pas*.

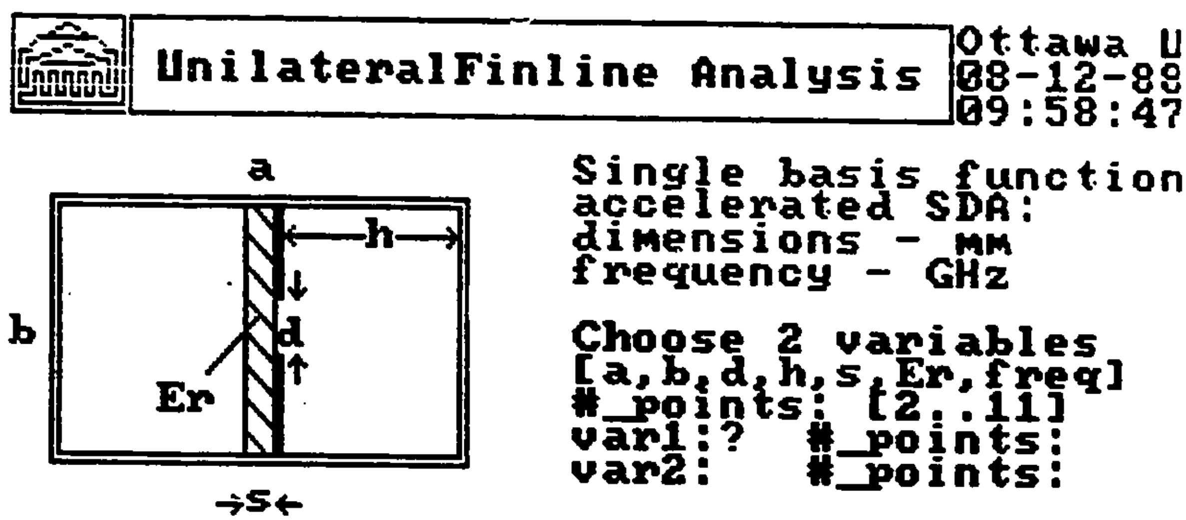


Figure A3: A *BiFastS.Pas* input menu.

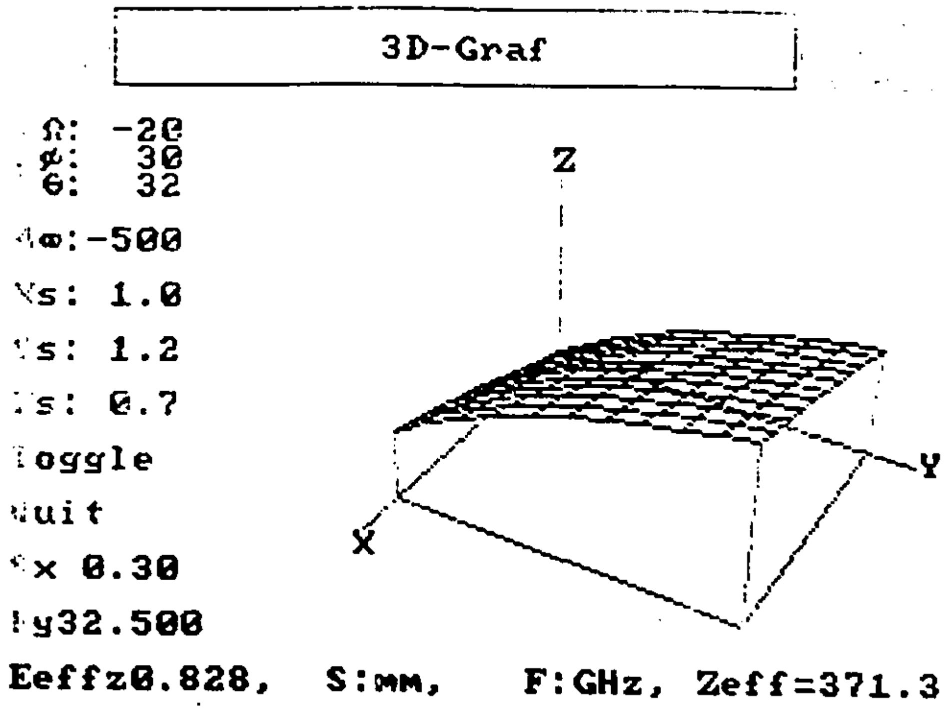


Figure A4: A graphical image of ϵ_{eff} versus f and s from an 11–by–11 lookup table which is generated by *BiFastS.Pas*.

APPENDIX B

Lookup Tables

This novel CAD procedure can make use of lookup tables of two different sizes. The interpolation methods associated with them are described in Chapter 2. The format of these lookup tables is given in this appendix. Should the format be changed the input-table procedures of *3D-Graf.Pas, Make-Re4.Pas* and *Make-S11.Pas* must also be modified.

Table B1 is an 11-by-11 lookup table and Table B2 is a 3-by-11 lookup table. These two tables have the same style, the only difference is that Table B1 is relatively bigger because it has more frequency points. The first three lines of the tables are ignored by *3D-Graf.Pas, Make-Re4.Pas* and *Make-S11.Pas*¶. These three lines should not be deleted but can be modified to record information that is of interest to the user. The next four lines contain information about frequency: `Fmin`, `Fmax`, `DeltaF` and `NF`. Note that

$$\texttt{DeltaF} = \frac{\texttt{Fmax - Fmin}}{\texttt{NF - 1}}$$

The first character of the first of these four lines is called a "hot" character. In this case the hot character is "`F`". This hot character is used by *3D-Graf.Pas* to set up its display menu, and it is also saved in the binary files created by *Make-Re4.Pas* and *Make-S11.Pas* so that *TsiFin* can double check the integrity of the binary file. This is a precaution step that allows *TsiFin* to verify if the lookup tables used are indeed the correct tables. This step is needed because *UniFastS.Pas* and *BiFastS.Pas* can generate lookup tables of sweeping variables other than frequency and gapwidth. The units "`GHz`" following the numbers are used by *3D-Graf.Pas* only. *Make-Re4.Pas* and *Make-S11.Pas* assume frequency is in "`GHz`".

Following the above-mentioned four lines are two blank lines, then 4 lines contain information about gapwidth: `Dmin`, `Dmax`, `DeltaD` and `ND`. Note that

$$\texttt{DeltaD} = \frac{\texttt{Dmax - Dmin}}{\texttt{ND - 1}}$$

In this case the hot character is "`D`". It serves the same purpose as the previous hot character. The units, "`mm`" following the numbers are used by *3D-Graf.Pas* only. *Make-Re4.Pas* and *Make-S11.Pas* assume gapwidth is in "`mm`". Following these four lines is the body of the lookup table. It contains `NF`† number of frequency blocks. Each frequency blocks contains six comment lines and `ND`‡ number of data lines. Each data line contains three fields: gapwidth, effective dielectric constant and effective characteristic impedance§ in Ohms. The first field is for human reader only and is discarded by the utility programs.

¶ From here on they are collectively called as utility programs.

† `NF`=3 in Table B1 and 11 in Table B2.

‡ `ND`=11 in both Table B1 and B2.

§ Voltage power impedance definition is used in *UniFastS.Pas* and *BiFastS.Pas*.

At the end of the table are a number of comment lines. Currently a useful piece of information is stored there and is extrated by *Make-Re4.Pas* and *Make-911.Pas*. It is the height of the waveguide, i.e. *b*. This piece of information is needed by *TsiFin*.

Table B1: An 11-by-11 lookup table generated by using *UniFastS.Pas*.

Laboratory for Electromagnetics and Microwaves, Faculty of Engineering,
Department of Electrical Engineering, University of Ottawa, Canada.

Fmin = 25.00000 GHz
Fmax = 40.00000 GHz
DeltaF = 1.50000 GHz
NF = 11

Dmin = 0.30000 mm
Dmax = 3.55600 mm
DeltaD = 0.32560 mm
ND = 11

f = 25.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.05076	182.50610
0.62560	0.90835	244.85565
0.95120	0.80331	301.19517
1.27680	0.71679	355.17314
1.60240	0.64188	407.90944
1.92800	0.57555	459.76597
2.25360	0.51699	509.62460
2.57920	0.46629	556.22283
2.90480	0.42438	597.26362
3.23040	0.39329	629.05184
3.55600	0.37846	644.23699

f = 26.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.08236	180.93584
0.62560	0.94811	241.29687
0.95120	0.84990	294.91660
1.27680	0.76947	345.32364
1.60240	0.70016	393.49108
1.92800	0.63897	439.64576
2.25360	0.58500	482.66541
2.57920	0.53826	521.46528
2.90480	0.49956	554.31090
3.23040	0.47076	578.73431
3.55600	0.45695	589.98201

f = 28.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.10929	179.87045
0.62560	0.98192	238.78595
0.95120	0.88939	290.44579
1.27680	0.81408	338.31757
1.60240	0.74948	383.31829
1.92800	0.69260	425.63235
2.25360	0.64250	464.19161
2.57920	0.59909	498.08065
2.90480	0.56308	525.94674
3.23040	0.53619	546.03286
3.55600	0.52323	555.00794

f = 29.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.13245	179.19386
0.62560	1.01093	237.06150
0.95120	0.92322	287.28885
1.27680	0.85226	333.31869
1.60240	0.79164	376.04844
1.92800	0.73841	415.65388
2.25360	0.69157	451.12545
2.57920	0.65098	481.68007
2.90480	0.61725	506.22770
3.23040	0.59198	523.46820
3.55600	0.57973	530.96142

f = 31.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.15256	178.82084
0.62560	1.03604	235.94309
0.95120	0.95247	285.12259
1.27680	0.88520	329.79942
1.60240	0.82799	370.86683
1.92800	0.77788	408.50377

2.25360	0.73382	441.74886
2.57920	0.69563	469.91994
2.90480	0.66383	492.11169
3.23040	0.63994	507.34069
3.55600	0.62830	513.78260

f = 32.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.17020	178.68990
0.62560	1.05799	235.30052
0.95120	0.97794	283.72473
1.27680	0.91387	327.40797
1.60240	0.85959	367.24743
1.92800	0.81216	403.42797
2.25360	0.77049	435.02371
2.57920	0.73435	461.42677
2.90480	0.70421	481.86844
3.23040	0.68148	495.59300
3.55600	0.67036	501.24204

f = 34.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.18578	178.75484
0.62560	1.07731	235.04075
0.95120	1.00033	282.93692
1.27680	0.93902	325.90070
1.60240	0.88715	364.82899
1.92800	0.84214	399.93031
2.25360	0.80253	430.28384
2.57920	0.76817	455.34191
2.90480	0.73946	474.43579
3.23040	0.71774	486.98793
3.55600	0.70706	492.00708

f = 35.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.19966	178.97994
0.62560	1.09443	235.09356
0.95120	1.02014	282.64368
1.27680	0.96121	325.10281
1.60240	0.91155	363.37249
1.92800	0.86854	397.66918
2.25360	0.83074	427.08122
2.57920	0.79790	451.09756
2.90480	0.77043	469.12619
3.23040	0.74957	480.73107
3.55600	0.73927	485.22969

f = 37.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.21210	179.33711
0.62560	1.10975	235.40428
0.95120	1.03779	282.75806
1.27680	0.98095	324.88656
1.60240	0.93321	362.69481
1.92800	0.89196	396.40536
2.25360	0.85571	425.10272
2.57920	0.82421	448.30414
2.90480	0.79780	465.47262
3.23040	0.77769	476.29006
3.55600	0.76771	480.34379

f = 38.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.22334	179.80392
0.62560	1.12352	235.93095
0.95120	1.05360	283.21254
1.27680	0.99862	325.15477
1.60240	0.95256	362.66263
1.92800	0.91284	395.96170
2.25360	0.87795	424.12504
2.57920	0.84761	446.68349
2.90480	0.82213	463.14767
3.23040	0.80267	473.29436
3.55600	0.79297	476.95686

f = 40.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.23355	180.36214
0.62560	1.13599	236.63971
0.95120	1.06787	283.95587
1.27680	1.01451	325.83267
1.60240	0.96993	363.17373
1.92800	0.93156	396.20647
2.25360	0.89787	423.98196
2.57920	0.86855	446.03752
2.90480	0.84388	461.91496
3.23040	0.82497	471.47825

3.55600	0.81551	474.78828

Input parameters (the stop values of the swept parameters) are:
Waveguide width, a : 7.11200
Waveguide height, b : 3.55600
Finline gap width, d : 3.55600
Dielectric constant, eps : 2.22000
Frequency, f : 40.00000
Metallization distance from wall, h : 3.55600
Substrate thickness, s : 0.25400

Contents of this file may be graphed by running
3D-graf <file.Dat>

Computation time is 0: 1:30.46

Table B2: A 3–by–11 lookup table generated by using *UniFastS.Pas.*

Laboratory for Electromagnetics and Microwaves, Faculty of Engineering,
Department of Electrical Engineering, University of Ottawa, Canada.

Fmin = 25.00000 GHz
Fmax = 40.00000 GHz
DeltaF = 7.50000 GHz
NF = 3

Dmin = 0.30000 mm
Dmax = 3.55600 mm
DeltaD = 0.32560 mm
ND = 11

f = 25.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.05076	182.50610
0.62560	0.90835	244.85565
0.95120	0.80331	301.19517
1.27680	0.71679	355.17314
1.60240	0.64188	407.90944
1.92800	0.57555	459.76597
2.25360	0.51699	509.62460
2.57920	0.46629	556.22283
2.90480	0.42438	597.26362
3.23040	0.39329	629.05184
3.55600	0.37846	644.23699

f = 32.50000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.17020	178.68990
0.62560	1.05799	235.30052
0.95120	0.97794	283.72473
1.27680	0.91387	327.40797
1.60240	0.85959	367.24743
1.92800	0.81216	403.42797
2.25360	0.77049	435.02371
2.57920	0.73435	461.42677
2.90480	0.70421	481.86844
3.23040	0.68148	495.59300
3.55600	0.67036	501.24204

f = 40.00000 GHz
Sweep over a range of d

d in mm	Epsilon Effective	Zo (Ohms)
0.30000	1.23355	180.36214
0.62560	1.13599	236.63971
0.95120	1.06787	283.95587
1.27680	1.01451	325.83267
1.60240	0.96993	363.17373
1.92800	0.93156	396.20647
2.25360	0.89787	423.98196
2.57920	0.86855	446.03752
2.90480	0.84388	461.91496
3.23040	0.82497	471.47825
3.55600	0.81551	474.78828

Input parameters (the stop values of the swept parameters) are:
Waveguide width, a : 7.11200
Waveguide height, b : 3.55600
Finline gap width, d : 3.55600
Dielectric constant, eps : 2.22000
Frequency, f : 40.00000
Metallization distance from wall, h : 3.55600
Substrate thickness, s : 0.25400

Contents of this file may be graphed by running
3D-Graf <file.Dat>

Computation time is 0: 0:24.99

APPENDIX C

3D-Graf.Pas is a three-dimensional graphic program developed to display the content of the lookup tables created by *UniFastS.Pas* and *BiFastS.Pas*. In fact any data file that has the same format as the lookup tables can be graphed by this program.

Figure C1 depicts an output from this program. The commands on the display menu at the left of the figure can be used to change the orientation and size of the graph. As show in Figure C1, the default setting ($\Omega = -13, \Phi = 20$ and $\Theta = 30$) may not yield a nice picture. The orientation of the graph can be change by changing the values of Ω, Φ and Θ; these are angles of rotation about the x, y and z-axis respectively. Figure C2 to C4 display the same graph at three different angle settings. The perspective of the graph can also be modified by changing the value of ∞. Figure C2 to C4 have $\infty = -5000000$, i.e. the infinity point is set to -5000000 units behind the surface of the paper. Figure C5 has $\infty = -250$, the perspective effect can be seen clearly by comparing this diagram with Figure C4. The scaling factors for each axis (Xs, Ys and Zs) can also be changed. Figure C5 has Ys=1.2 while Figure C6 has Ys=1.8. The difference between these two graphs are quite clear.

Figure C7 has $\Omega = -20, \Phi = 30$ and $\Theta = 32$. This graph reveals the behaviour of ϵ_{eff} (z-axis) versus substrate thickness (S, x-axis) and frequency (F, y-axis) more effectively then Figure C1. The "Toggle" key can be used to switch between the ϵ_{eff} graph and the Z_{eff} graph, see Figure C8. There is a big dot at the middle of Figure C8†. At that position, frequency is 32.5 GHz, substrate thickness is 0.3 mm, effectively dielectric constant is 0.828 and effective characteristic impedance is 371.3 Ω.

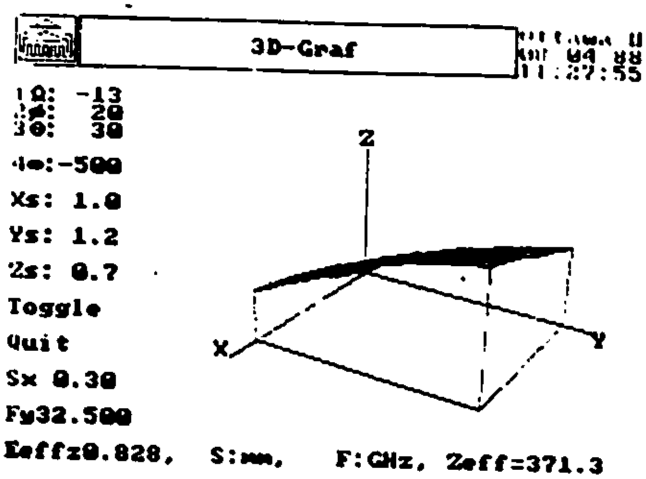


Figure C1: A table displayed by using *3D-Graf.Pas*.

† Acturally there is also a big dot in the middle of Figure C7 but it does not show up clearly on a black-and-white diagram. On a color monitor this big dot is a big-red-dot and it can be seen clearly no matter what the orientation of the graph is.

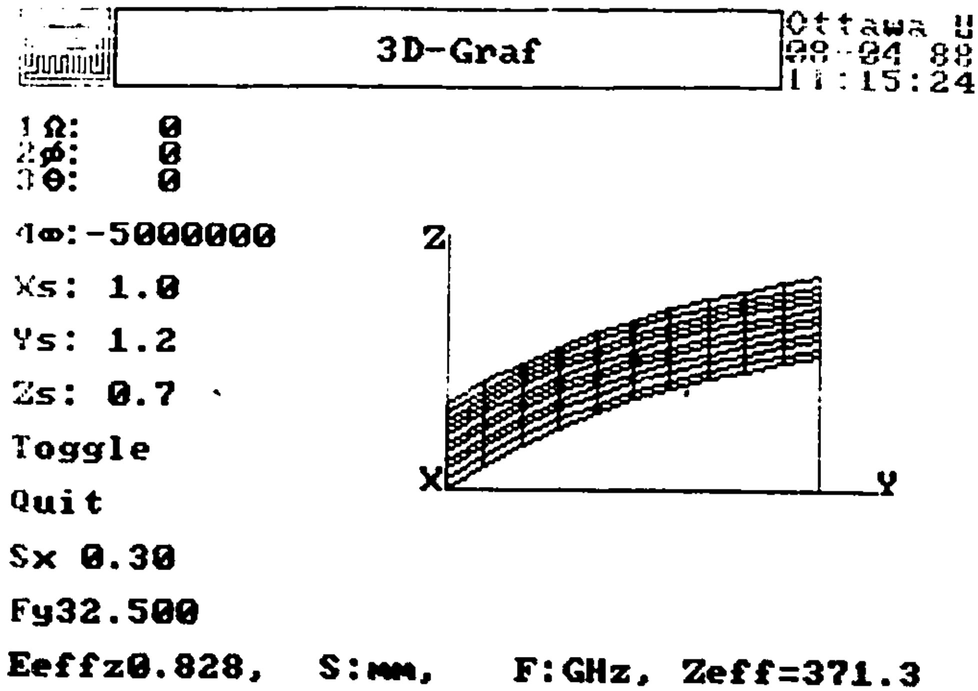


Figure C2: The table of Figure C1 at Ω=0, Φ=0 and Θ=0.

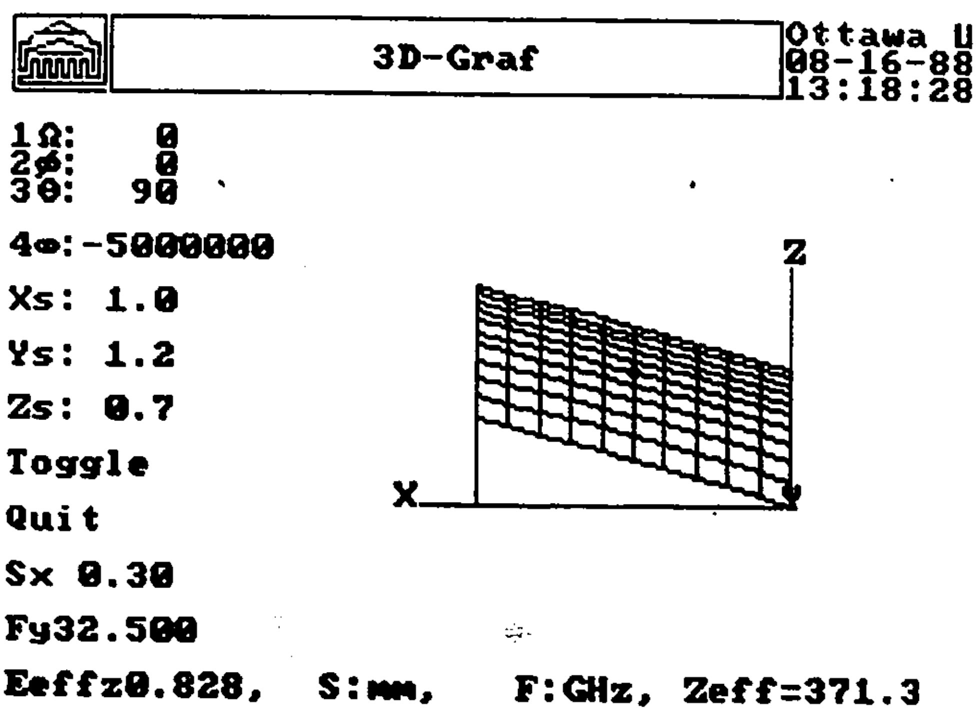


Figure C3: The table of Figure C1 at $\Omega = 0, \Phi = 0$ and $\Theta = 90$.

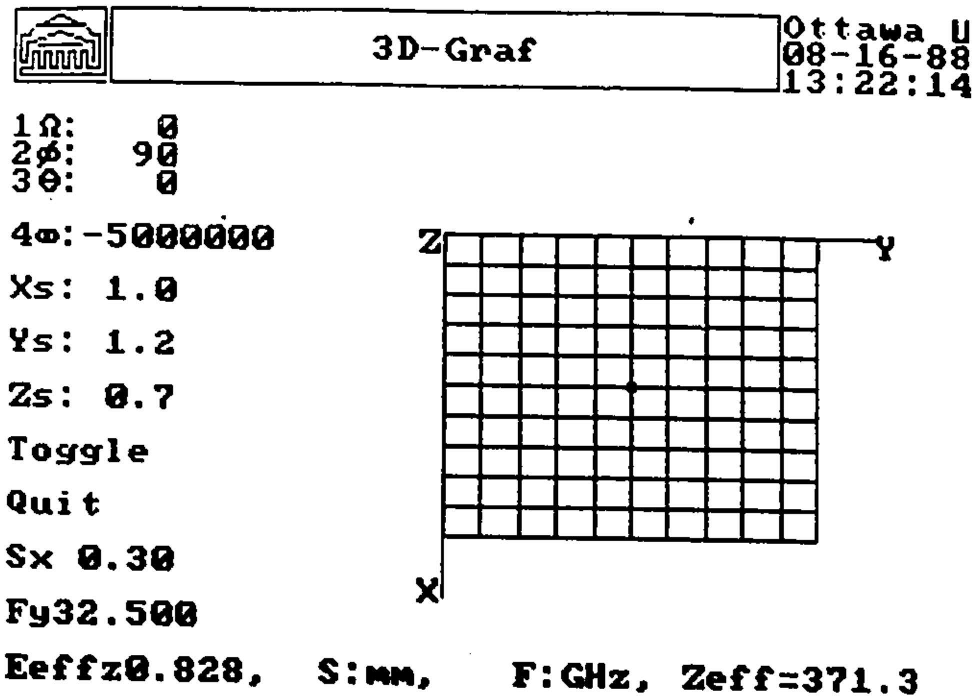


Figure C4: The table of Figure C1 at $\Omega = 0, \Phi = 90$ and $\Theta = 0$.

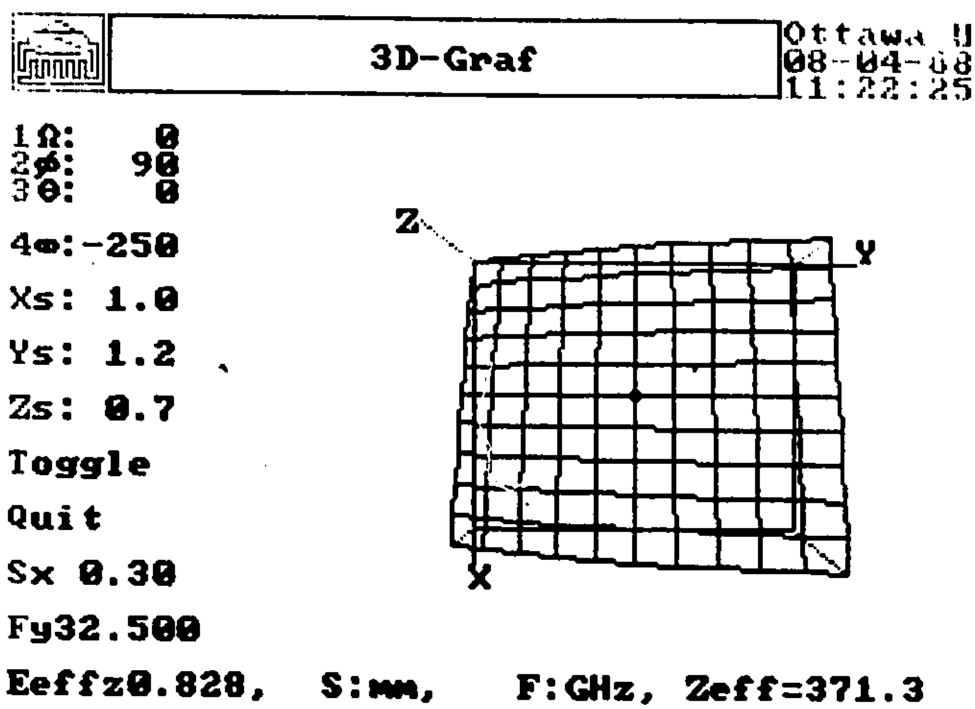


Figure C5: A perspective view of Figure C4.

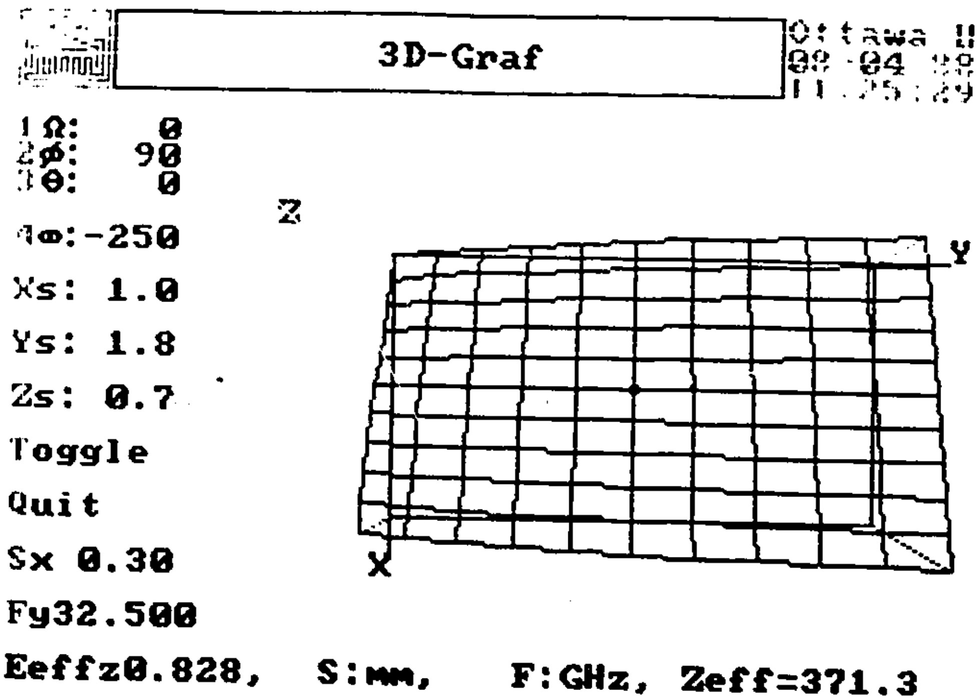


Figure C6: An enlarged version of Figure C5.

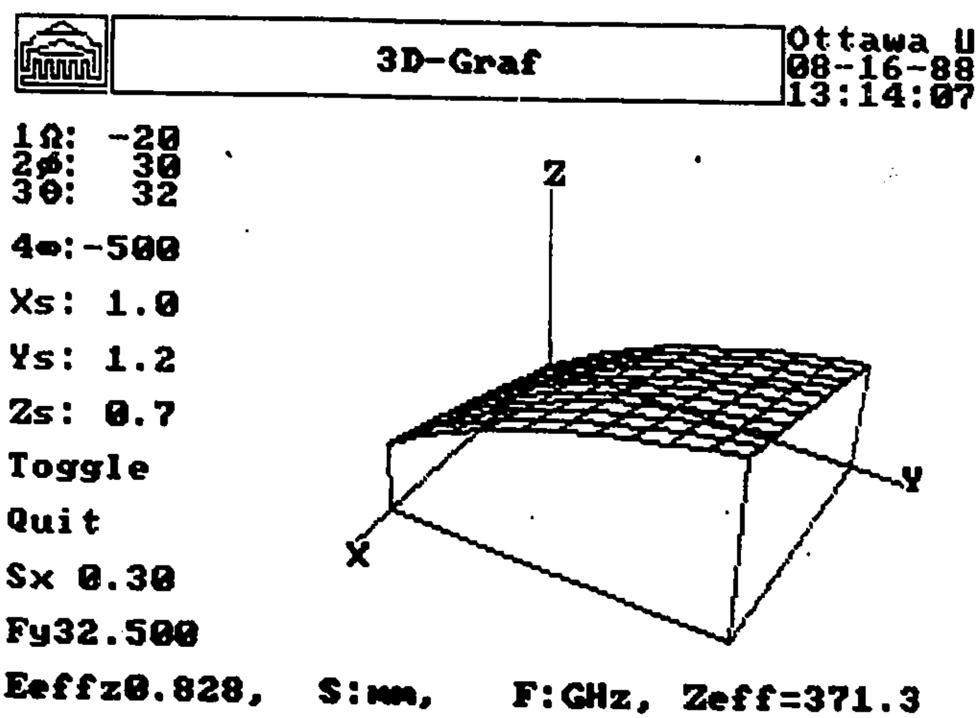


Figure C7: A better viewing angle of the same table in Figure C1.

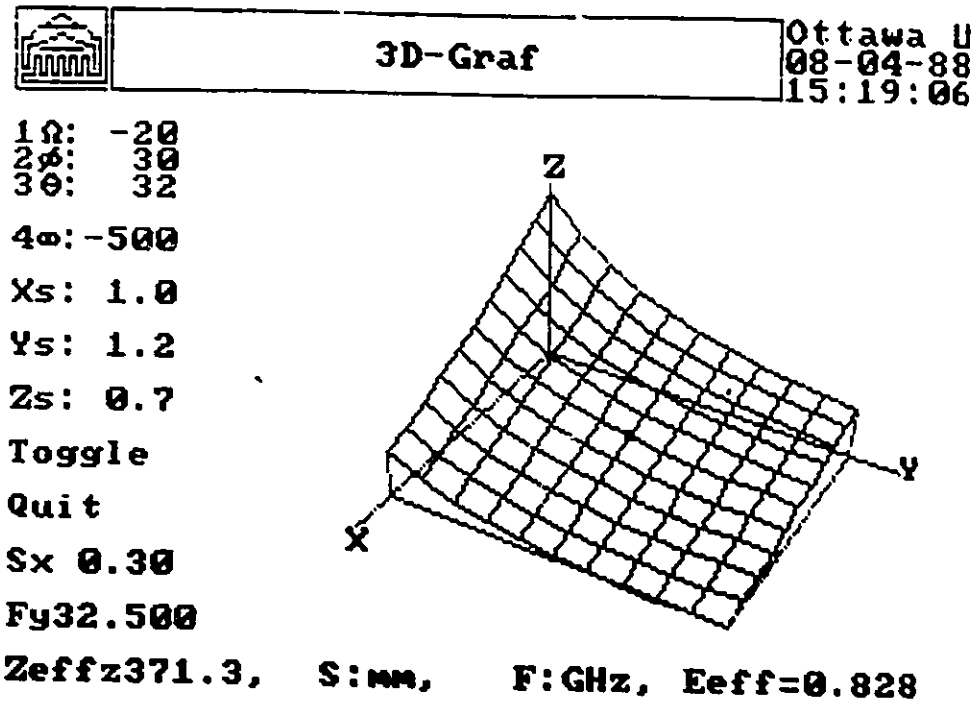


Figure C8: Z_{eff} versus substrate thickness and frequency.

Listing C1: 3D-Graf.Pas

```
                              (* File 3D-Graf.Pas *)
(*-----------------------------------------------------------------------------
 This is a three dimensional graphic program developed to graph the contents
 of lookup tables created by UniFastS.Pas and BiFastS.Pas. The size of the
 lookup tables should be m-by-n, where m, n is in [2,11] inclusively. The
 calling syntax is:

                           3D-Graf TableIn.Dat

 This program can be compiled directly by using Turbo Pascal Version 4.0
 compiler without any modifications.  To compile this program by using
 earlier version compiler follow the following steps:
   (1) Delete the uses clause - uses crt, dos, graph3;
   (2) Change all 'Single' declarations to 'Real' declarations.
   (3) Compile the modified program.

                                  Poman So

                 Laboratory for Electromagnetics and Microwaves
                            University of Ottawa
                              August 4, 1988.

------------------------------------------------------------------------------*)
Program ThreeD_Graf;
uses crt,dos,graph3;
Const
  imax = 100;
  co0 = 0;
  co1 = 1;
  co2 = 2;
  co3 = 3;
  MaxXa = 11;
  MaxYa = 11;
  NV =  2;
  Xmax = 160;
  Ymax = 90;
  Afact= 0.87;
  X1 = 56;
  Y1 = 30;
  X2 = 319;
  Y2 = 189;
  Aoff = 1.4;
  Xorg : Integer = 150;
  Yorg : Integer = 80;

Type
  Table = Record
    Xname  : Char;
```

```
      XaMin   : Single;
      XaMax   : Single;
      DeltaXa: Single;
      NXa     : Integer;
      Yname   : Char;
      YaMin   : Single;
      YaMax   : Single;
      DeltaYa: Single;
      NYa     : Integer;
      Datum   : Array[1..MaxXa,1..MaxYa,1..NV] of Single;
    End;

    Point = Record
      x : Integer;
      y : Integer;
    End;

    Axis = String[4];
    String3 = STRING[3];
    String8 = STRING[8];
    String40 = STRING[40];

  Var ou: array [1..imax] of point;

  (* Calendar.Lib                        Poman SO  Aug 25, 1987.
    This procedure return the date, time and week of the day;
    type declarations required are:
    TYPE
      String3 = STRING[3];
      String8 = STRING[8];
  *)
  PROCEDURE Date_Time_Week( VAR Date, Time: String8; VAR Week: String3 );
  TYPE String2 = STRING[2];

  FUNCTION TwoChar( I: word): String2;
  VAR temp: String2;
  BEGIN
    Str( Round(100*Frac(I/100.0)):2, temp );
    IF Pos(' ',temp )=1 THEN
    BEGIN
      Delete( temp, 1, 1 );
      TwoChar := '0'+temp;
    END
    ELSE
      TwoChar := temp;
  END; (*-TwoChar-*)

  Var Hour, Min, Sec, Frac, Week_Day, Year, Month, Day, temp: Word;

  BEGIN
    gettime( hour, min, sec, temp );
    getdate( year, month, day, week_day );
    Date := TwoChar(Month) + '-' + TwoChar(Day) + '-' + TwoChar(Year);
    Time := TwoChar(Hour)  + ':' + TwoChar(Min) + ':' + TwoChar(Sec);
    Case Week_Day of
      0: Week := 'SUN';
      1: Week := 'MON';
      2: Week := 'TUE';
      3: Week := 'WED';
      4: Week := 'THU';
      5: Week := 'FRI';
      6: Week := 'SAT';
     ELSE Week := '???';
    END;
  END; (*-Date_Time_Week-*)

  procedure readou;
  var picfile: text;
      i:integer;
  begin
    {*$I-*)
    Assign( PicFile, '\POMAN\SENIOR\Picture\OttawaU.Pic' );
    Reset(PicFile );
    i := IOresult;
    (*$I+*)
    IF i=0 THEN
    BEGIN
      REPEAT
        i := i + 1;
        Read( PicFile, ou[i].x, ou[i].y );
      UNTIL ((ou[i].x<0) and (ou[i].y<0)) or (i=imax);
      Close( PicFile )
    END
    ELSE
    Write ( 'IO ERROR #', i )
  end;

  PROCEDURE DrawPic( Xorg, Yorg, Xm, Ym: INTEGER; color: integer );
  VAR
    i, x1, y1, x2, y2: INTEGER;
  BEGIN
    i := 0;
    REPEAT
      i := i + 1;
      x2 := Xorg + ou[i].x*Xm;
      y2 := Yorg + ou[i].y*Ym;
      i := i + 1;
      REPEAT
        x1 := x2;
        y1 := y2;
        x2 := Xorg + ou[i].x*Xm;
        y2 := Yorg + ou[i].y*Ym;
        draw( x1, y1, x2, y2, color );
        i := i + 1;
```

```
    UNTIL (i>imax) or (ou[i].x<0)
  UNTIL (i>imax) or (ou[i].y<0)
END; (*-DrawPic-*)

(*----------------------------------------------------------------------
  This procedure displays the Ottawa U symbol and the current data, time
  and week.
----------------------------------------------------------------------*)
PROCEDURE Display_Heading;
VAR
  Date, Time: String8;
  Week: String3;
BEGIN
  DrawPic( 0, 0, 1, 1, 1 );
  draw( 32, 0, 254, 0, co2 );
  draw( 254, 0, 254, 22, co2 );
  draw( 254, 22, 32, 22, co2 );
  draw( 32, 22, 32, 0, co2 );
  Date_Time_Week ( Date, Time, Week );
  TextColor( co3 );
  GotoXY( 6, 2 ); Write( '          3D-Graf' );
  TextColor( co1 );
  GotoXY( 33, 1 ); Write( 'Ottawa U' );
  GotoXY( 33, 2 ); Write( Date );
  GotoXY( 33, 3 ); Write( Time );
END; (*Display_Heading*)

  Procedure ReadData( Var Surface: Table );
  Var
    i, j, k : Integer;
    temp    : Single;
    infile  : text;
    garbage : String[9];
    fname   : String[80];
  Begin
    i := Paramcount;
    (*$I-*)
    Repeat
      IF i<>0 Then
        Assign( infile, ParamStr(1) )
      ELSE
      Begin
        Write( 'Please specify input text file: [abc.Dat]    > ' );
        Readln( fname );
        Assign( infile, Copy(fname,1,14) );
      End;
      Reset( infile );
      j := IOResult;
      IF j<>0 Then i:=0;
    Until j=0;
    (*$I+*)

    (********************)
    (* Input data block *)
    (********************)
    (* Skip three lines of headings *)
    Readln( infile );
    Readln( infile );
    Readln( infile );
    With Surface Do
    Begin
      Readln( infile, garbage, XaMin );   XaMin   := XaMin*1e9;
      Xname := garbage[1];
      Readln( infile, garbage, XaMax );   XaMax   := XaMax*1e9;
      Readln( infile, garbage, DeltaXa ); DeltaXa := DeltaXa*1e9;
      Readln( infile, garbage, NXa );
      Readln( infile ); Readln( infile );
      Readln( infile, garbage, YaMin );   YaMin   := YaMin*1e-3;
      Yname := garbage[1];
      Readln( infile, garbage, YaMax );   YaMax   := YaMax*1e-3;
      Readln( infile, garbage, DeltaYa ); DeltaYa := DeltaYa*1e-3;
      Readln( infile, garbage, k );
      If NXa>MaxXa Then
        NXa:=MaxXa;
      If k >MaxYa Then
        NYa:=MaxYa
      Else
        NYa := k;
      For i:=1 To NXa Do
      Begin
        Readln( infile ); Readln( infile ); Readln( infile );
        Readln( infile ); Readln( infile ); Readln( infile );
        For j:=1 To NYa Do
          Readln( infile, temp, Datum[i,j,1], Datum[i,j,2] );
        For j:=NYa+1 To k DO Readln( infile );
      End;
    End; (*With*)
    Close( infile );
  End;

  Procedure SurfaceValues( xs, ys: Single; Var Surface: Table;
                           Var Eeff, Zeff: Single );
  Var
    nx1, nx2, ny1, ny2: Integer;
    x, y, DX, DY, Ea, Eb, Za, Zb: Single;

  Begin
    With Surface Do
    Begin
      (*****************************************)
      (* Make sure fs and ds are within bounds *)
      (*****************************************)
      If xs > XaMax Then x:=XaMax
```

```
        Else If xs < XaMin Then x:=XaMin
        Else x:=xs;
        If ys > YaMax Then y:=YaMax
        Else If ys < YaMin Then y:=YaMin
        Else y:=ys;

        {*******************************}
        {* 2 dimensional interpolation *}
        {*******************************}
        nx1 := Trunc((x-XaMin)/DeltaXa) + 1;
        If nx1 < NXa Then nx2:=nx1+1
        Else nx2:=nx1;
        DX := (x-XaMin)/DeltaXa - nx1 + 1;
        ny1 := Trunc((y-YaMin)/DeltaYa) + 1;
        If ny1 < NYa Then ny2:=ny1+1
        Else ny2:=ny1;
        DY := (y-YaMin)/DeltaYa - ny1 + 1;
        Ea := Datum[nx1,ny1,1] + DX * (Datum[nx2,ny1,1] - Datum[nx1,ny1,1]);
        Eb := Datum[nx1,ny2,1] + DX * (Datum[nx2,ny2,1] - Datum[nx1,ny2,1]);
        Eeff := Ea + DY * (Eb-Ea);
        Za := Datum[nx1,ny1,2] + DX * (Datum[nx2,ny1,2] - Datum[nx1,ny1,2]);
        Zb := Datum[nx1,ny2,2] + DX * (Datum[nx2,ny2,2] - Datum[nx1,ny2,2]);
        Zeff := Za + DY * (Zb-Za);
     End; (*With*)
   End; (*SurfaceValues*)


 Var
   title: String[40];
   infile : file of Table;
   answer : char;
   Surface : Table;
   garbage : String[9];
   fname   : String[14];
   x : Array[1..MaxXa] of Single;
   y : Array[1..MaxYa] of Single;
   DataPoint : Array[1..MaxXa, 1..MaxYa] of Point;
   xa, ya, xb, yb, xc, yc, i, j, NXa, NYa : integer;
   Mx,My,Mz,vy,vx,Xinf,radians,phi,theta,omiga,temp,Lmax,Emax,Zmax,Emin:Single;
   Eeff,Zeff,cosphi,costheta,cosomiga,sinphi,sintheta,sinomiga,Zmin:Single;


Procedure Mapping( x, y, z: Single; Var xa, ya: Integer);
Var
  x1, x2, y1, y2, z1, z2: Single;
Begin
  y1 := y*cosomiga - z*sinomiga;
  z1 := z*cosomiga + y*sinomiga;

  x2 := x*cosphi*costheta + y1*sintheta + z1*sinphi*costheta;
  y2 := y1*costheta - z1*sinphi*sintheta - x*cosphi*sintheta;
  z2 := z1*cosphi - x*sinphi;

  x1 := (1-x2/Xinf)*y2;
  y1 := Afact*(1-x2/Xinf)*z2;
  xa := Round(Xorg + x1);
  ya := Round(Yorg - y1);
End; (* Mapping *)


Procedure ReadXY( Out: Char; X, Y:Integer; Var Z: Single );
Var
  I, J, K: Integer;
Begin
  answer := 'g';
  (*$I-*)
  Repeat
    GotoXY( X, Y );
    Write( Out,^H );
    Read( Z );
    I := IOResult;
    IF I<>0 THEN
    Begin
      Write( ^G );
      J := WhereX;
      GotoXY( X, Y );
      FOR K:=X TO J DO Write( ' ' );
    End;
  Until I=0;
  (*$I+*)
End; (* ReadXY *)


Procedure DrawGraph( k: Integer; Max, Min: Single; zname: Axis );
Var
  NewMx, NewMy, NewMz, vz, Oper, Gmax, Gmin: Single;
  line1, line2 : String[9];

Procedure F1;
Begin
  Mapping( Mx*x[1],  My*y[1], GMin, xa, ya );
  Mapping( Aoff*Mx, My*y[1], GMin, xc, yc );
  Draw( xa, ya, xc, yc, 2 );
  xc := Round( (4+xc+X1)/8 );
  yc := Round( (4+yc+Y1)/8 );
  IF (xc<41) and (xc>8) and (yc<25) and (yc>1) Then
  Begin
    Gotoxy( xc, yc );
    TextColor(3);
    Write( 'X' );
  End;
End;

Procedure F2;
Begin
  Mapping( Mx*x[NXa], My*y[1],  GMin, xa, ya );
```

```
    Mapping( Mx*x[NXa], My*y[NYa], GMin, xb, yb );
    Draw( xa, ya, xb, yb, 2 );
  End;

  Procedure F3;
  Begin
    Mapping( Mx*x[NXa], My*y[NYa], GMin, xa, ya );
    Mapping( Mx*x[1],  My*y[NYa], GMin, xb, yb );
    Draw( xa, ya, xb, yb, 2 );
  End;

  Procedure F4;
  Begin
    Mapping( Mx*x[1], My*y[1],  GMin, xa, ya );
    Mapping( Mx*x[1], Aoff*My, GMin, xc, yc );
    Draw( xa, ya, xc, yc, 2 );
    xc := Round( (4+xc+X1)/8 );
    yc := Round( (4+yc+Y1)/8 );
    IF (xc<41) and (xc>8) and (yc<25) and (yc>1) Then
    Begin
      GotoXY( xc, yc );
      TextColor(3);
      Write( 'Y' );
    End;
  End;

  Procedure F5;
  Begin
    Mapping( Mx*x[1],  My*y[1], GMin, xa, ya );
    Mapping( Mx*x[1], My*y[1], Aoff*GMax, xc, yc );
    Draw( xa, ya, xc, yc, 1);
    xc := Round( (4+xc+X1)/8 );
    yc := Round( (4+yc+Y1)/8 );
    IF (xc<41) and (xc>8) and (yc<25) and (yc>1) Then
    Begin
      GotoXY( xc, yc );
      TextColor(3);
      Write( 'Z' );
    End;
  End;

  Procedure F6;
  Begin
    Mapping( Mx*x[NXa], My*y[1],  GMin, xa, ya );
    Draw( DataPoint[NXa,1].x, DataPoint[NXa,1].y, xa, ya, 1 );
  End;

  Procedure F7;
  Begin
    Mapping( Mx*x[NXa], My*y[NYa], GMin, xa, ya );
    Draw( DataPoint[NXa,NYa].x, DataPoint[NXa,NYa].y, xa, ya, 1 );
  End;

  Procedure F8;
  Begin
    Mapping( Mx*x[1], My*y[NYa], GMin, xa, ya );
    Draw( DataPoint[1,NYa].x, DataPoint[1,NYa].y, xa, ya, 1 );
  End;

  Procedure F9;
  Var y1, y2: Integer;
  Begin
    For i:=1 To NXa Do
      For j:=1 To NYa-1 Do
        Draw( DataPoint[i,j].x, DataPoint[i,j].y, DataPoint[i,j+1].x,
              DataPoint[i,j+1].y, 3 );
    For j:=1 To NYa Do
      For i:=1 To NXa-1 Do
        Draw( DataPoint[i,j].x, DataPoint[i,j].y, DataPoint[i+1,j].x,
              DataPoint[i+1,j].y, 3 );
    With Surface Do
      Mapping(Mx*(2*vx-XaMax-XaMin)/(XaMax-XaMin),
              My*(2*vy-YaMax-YaMin)/(YaMax-YaMin),
              Mz*(2*vz-Max-Min)/Oper, xc, yc );
    y1 := yc-1;
    y2 := yc+1;
    Draw( xc-1, y1, xc-1, y2, 2 );
    Draw( xc,   y1, xc,   y2, 2 );
    Draw( xc+1, y1, xc+1, y2, 2 );
  End;

Begin;
  Oper := Max-Min;
  Repeat
    cosphi   := cos(phi);
    costheta := cos(theta);
    cosomiga := cos(omiga);
    sinphi   := sin(phi);
    sintheta := sin(theta);
    sinomiga := sin(omiga);
    Mx := Mx * Lmax;
    My := My * Lmax;
    Mz := Mz * Lmax;
    Gmin := Mz*(2*Min-Max-Min)/Oper;
    Gmax := Mz*(2*Max-Max-Min)/Oper;
    SurfaceValues( vx, vy, Surface, Eeff, Zeff );
    IF zname='Eeff' THEN
      vz:=Eeff
    ELSE
      vz:=Zeff;

    For i:=1 To NXa Do
      For j:= 1 To NYa Do
        Mapping( Mx*x[i], My*y[j],
                 Mz*(2*Surface.Datum[i,j,k]-Max-Min)/Oper,
                 DataPoint[i,j].x, DataPoint[i,j].y );
```

```
 GraphColorMode;
Palette( 2 );
TextColor(3);
Display_Heading;
GraphWindow( X1, Y1, X2, Y2 );

F1; F2; F3; F4; F5; F6; F8; F9; F7;
NewMx := Mx/Lmax;
NewMy := My/Lmax;
NewMz := Mz/Lmax;

TextColor(1); Gotoxy(1,5); Write('1');
  TextColor(2); Write( ':', omiga/radians:4:0 );
TextColor(1); Gotoxy(1,6); Write('2');
  TextColor(2); Write( 'm:', phi  /radians:4:0 );
TextColor(1); Gotoxy(1,7); Write('3');
  TextColor(2); Write( 'l:', theta/radians:4:0 );
TextColor(1); Gotoxy(1,9); Write('4');
  TextColor(2); Write( 'l:', Xinf:4:0 );
TextColor(1); Gotoxy(1,11); Write('X');
  TextColor(2); Write('s:', NewMx:4:1 );
TextColor(1); Gotoxy(1,13); Write('Y');
  TextColor(2); Write('s:', NewMy:4:1 );
TextColor(1); Gotoxy(1,15); Write('Z');
  TextColor(2); Write('s:', NewMz:4:1 );
TextColor(1); Gotoxy(1,17); Write('T');
  TextColor(2); Write('oggle' );
TextColor(1); Gotoxy(1,19); Write('Q');
  TextColor(2); Write('uit');
TextColor(1); Gotoxy(1,21); Write(Surface.Xname);
  TextColor(2); Write('x');
  TextColor(3); Write(vx/1e9:5:2);
TextColor(1); Gotoxy(1,23); Write(Surface.Yname);
  TextColor(2); Write('y');
  TextColor(3); Write(vy*1e3:5:3);
Gotoxy(1,25); Write(zname); TextColor(2); Write('z'); TextColor(3);
  IF zname='Eeff' THEN
    Write(Eeff:5:3)
  ELSE
    Write(Zeff:5:1);
  Case Surface.Xname of
    'F' : line1:=',  F:GHz,';
    'E' : line1:=',  E:[ ],';
    Else Line1 :=',  '+Surface.Xname+':mm, ';
  End;
  Case Surface.Yname of
    'F' : line2:='  F:GHz, ';
    'E' : line2:='  E:[ ], ';
    Else Line2 :='  '+Surface.Yname+':mm,  ';
  End;
  Write( line1, line2 );
TextColor(2);
  IF zname='Eeff' THEN
    Write('Zeff=',Zeff:5:1)
  ELSE
    Write('Eeff=',Eeff:5:3);

TextColor(1);
Repeat
  answer := ReadKey;
  Case answer of
    '1': Begin
           ReadXY( '?', 8, 5, omiga );
           omiga := omiga*radians;
         End;
    '2': Begin
           ReadXY( '?', 8, 6, phi );
           phi := phi*radians;
         End;
    '3': Begin
           ReadXY( '?', 8, 7, theta );
           theta := theta*radians;
         End;
    '4': Begin
           ReadXY( '?', 4, 10, Xinf );
           IF Abs(Xinf)<Abs(Mx/2) THEN
             IF Xinf<>0 Then
               Xinf := Abs(Xinf)*Mx/Xinf/2
             Else
               Xinf := -Mx/2;
         End;
    'x', 'X': Begin
                ReadXY( '?', 4, 12, NewMx );
                IF Abs(NewMx)>3 Then
                  NewMx := 3*NewMx/Abs(NewMx);
                IF Abs(NewMx/Lmax)>Abs(2*Xinf) THEN
                  NewMx := Abs(NewMx)*2*Xinf/NewMx/Lmax;
              End;
    'y', 'Y': Begin
                ReadXY( '?', 4, 14, NewMy );
                IF Abs(NewMy)>3 Then
                  NewMy := 3*NewMy/Abs(NewMy);
              End;
    'z', 'Z': Begin
                ReadXY( '?', 4, 16, NewMz );
                IF Abs(NewMz)>3 Then
                  NewMz := 3*NewMz/Abs(NewMz);
            End;
    'a','A','b','B','d','D','e','E','f','F','h','H','s','S' :
        With Surface DO
        Begin
          answer := Upcase( answer );
          IF answer=xname Then
          Begin
            ReadXY( '?', 4, 22, vx );
            vx := vx * 1e9;
```

```
              IF vx>(2*XaMax-XaMin) THEN
                vx := 2*XaMax - XaMin
              Else IF vx<(2*XaMin-XaMax) THEN
                vx := 2*XaMin - XaMax;
            End (*IF *)
            Else IF answer=yname THEN
            Begin
              ReadXY( '?', 4, 24, vy );
              vy := vy * 1e-3;
              IF vy>(2*YaMax-YaMin) THEN
                vy := 2*YaMax - YaMin
              Else IF vy<(2*YaMin-YaMax) THEN
                vy := 2*YaMin - YaMax;
            End
            Else
              answer := 'N';
          End; (*With*)
      #27: Begin
            answer := ReadKey;
            Case answer of
              #59: F1;
              #60: F2;
              #61: F3;
              #62: F4;
              #63: F5;
              #64: F6;
              #65: F7;
              #66: F8;
              #67: F9;
              #75: Xorg:=Xorg-10;
              #77: Xorg:=Xorg+10;
              #72: Yorg:=Yorg-10;
              #80: Yorg:=Yorg+10;
              #71: Begin Xorg:=Xorg-10; Yorg:=Yorg-10; End;
              #79: Begin Xorg:=Xorg-10; Yorg:=Yorg+10; End;
              #73: Begin Xorg:=Xorg+10; Yorg:=Yorg-10; End;
              #81: Begin Xorg:=Xorg+10; Yorg:=Yorg+10; answer:='g'; End;
              Else answer:='N';
            End;
            TextColor(1);
          End;
      'g','G': answer:='N'; (*Disable 'g', and 'G'*)
    End;
    Until answer in [ 'g','q','Q','t','T',#71,#72,#73,#75,#77,#79,#80 ];

    Mx := NewMx;
    My := NewMy;
    Mz := NewMz;
  Until answer in ['t', 'T', 'q', 'Q'];
End; (* DrawGraph *)

BEGIN
  ReadData( Surface );
  With Surface DO
  Begin
    Zmax := -9.0E36;
    Emax := -9.0E36;
    Zmin :=  9.0E36;
    Emin :=  9.0E36;
    If NYa>MaxYa Then NYa:=MaxYa;
    If NXa>MaxXa Then NXa:=MaxXa;
    For i:=1 To NYa Do
      For j:=1 TO NXa Do
      Begin
        If Emin>Datum[j,i,1] Then Emin:=Datum[j,i,1];
        If Zmin>Datum[j,i,2] Then Zmin:=Datum[j,i,2];
        If Emax<Datum[j,i,1] Then Emax:=Datum[j,i,1];
        If Zmax<Datum[j,i,2] Then Zmax:=Datum[j,i,2];
      End;
    Mx := 0.5*(XaMax+XaMin);
    My := 0.5*(XaMax-XaMin);
    For i:=1 To NXa DO x[i] := (XaMin + (i-1)*DeltaXa - Mx)/My;
    Mx := 0.5*(YaMax+YaMin);
    My := 0.5*(YaMax-YaMin);
    For i:=1 To NYa DO y[i] := (YaMin + (i-1)*DeltaYa - Mx)/My;
  End; (* With *)

  NXa := Surface.NXa;
  NYa := Surface.NYa;
  Lmax := Ymax / Sqrt(3);
  radians := pi/180;
  omiga := -13*radians;
  phi   :=  20*radians;
  theta :=  30*radians;
  Mx := 1;
  My := 1.2;
  Mz := 0.7;
  Xinf := -500;
  vx := 0.5 * (Surface.XaMin + Surface.XaMax);
  vy := 0.5 * (Surface.YaMin + Surface.YaMax);
  SurfaceValues( vx, vy, Surface, Eeff, Zeff );
  readou;
  Repeat
    DrawGraph( 1, Emax, Emin, 'Eeff' );
    IF answer in ['t', 'T'] THEN
      DrawGraph( 2, Zmax, Zmin, 'Zeff' );
  Until answer in ['q','Q'];
  TextMode(co80);
END.
(************************)
(* End of ThreeD_Graf *)
(************************)
```

APPENDIX D

Interpolation Formulas

$$y_1 = A + Bf_1 + Cf_1^{-2} \qquad (1)$$
$$y_2 = A + Bf_1 + Cf_2^{-2} \qquad (2)$$
$$y_3 = A + Bf_1 + Cf_3^{-2} \qquad (3)$$

Subtract (2) from (1) and then divide the result by $(f_1 - f_2)$ yields:

$$B = \frac{y_1 - y_2}{f_1 - f_2} - C\frac{f_1^{-2} - f_2^{-2}}{f_1 - f_2} \qquad (4)$$

Similarly (1) and (3) yields:

$$B = \frac{y_1 - y_3}{f_1 - f_3} - C\frac{f_1^{-2} - f_3^{-2}}{f_1 - f_3} \qquad (5)$$

Subtract (5) from (4) yields:

$$C = \frac{(y_1 - y_2)(f_1 - f_3) - (y_1 - y_3)(f_1 - f_2)}{(f_1^{-2} - f_2^{-2})(f_1 - f_3) - (f_1^{-2} - f_3^{-2})(f_1 - f_2)} \qquad (6)$$

Let

$$k_1 = f_1^{-2}$$
$$k_2 = f_1 - f_3$$
$$k_3 = f_1 - f_2$$
$$k_4 = k_1 - f_2^{-2}$$
$$k_5 = k_2k_4 - k_3(k_1 - f_3^{-2})$$

After subetitute k_1 to k_5 into (6) yields:

$$C = \frac{k_2(y_1 - y_2) - k_3(y_1 - y_3)}{k_5} \qquad (6)$$

Using (7), (4) and (1) B and A are:

$$B = \frac{y_1 - y_2 - Ck_4}{k_3} \qquad (8)$$
$$A = y_1 - Bf_1 - Ck_1 \qquad (9)$$

Note that k_1, k_2, k_3, k_4 and k_5 are functions of f_1, f_2 and f_3 only.

APPENDIX E

Intpol33.Pas, Intpol.Pas and Error.Pas

Intpol33.Pas, *Intpol.Pas* and *Error.Pas* are three simple programs written to verify the accuracy of the two interpolation schemes in Chapter 2. These three programs are not related to the CAD procedures. They are included in this appendix just for the completeness of this thesis.

Listing E1: Intpol33.Pas

```
                           (* File: Intpol33.Pas *)
(*---------------------------------------------------------------------------
 This program is written to generated an 11-by-11 lookup table from a
 3-by-3 lookup table.  The calling syntax is:

                     Intpol33 TableIn.Dat TableOut.Dat

 Where
    (1) TableIn.Dat is a 3-by-3 lookup table created by either UniFastS.Pas
        or BiFastS.Pas.
    (2) TableOut.Dat is an 11-by-11 lookup table generated by using a
        3-by-3 intpolation scheme described in the thesis.

                                  Poman So

              Laboratory for Electromagnetics and Microwaves
                          University of Ottawa
                            August 4, 1988.

----------------------------------------------------------------------------*)
PROGRAM Intpol33;

FUNCTION Dbe(dob:single):single;
Begin
  Dbe := dob
End;

FUNCTION Dce(dob:single):single;
Begin
  Dce := dob*dob
End;

FUNCTION Dbz(dob:single):single;
Begin
  Dbz := dob
End;

FUNCTION Dcz(dob:single):single;
Begin
  Dcz := dob*dob
End;

FUNCTION Fb(f:single):single;
Begin
  Fb := f
End;

FUNCTION Fc(f:single):single;
Begin
Fc := 1/f/f
End

Function Int_eD(e1, e2, e3, d1ob, d2ob, d3ob, dob: single): single;
Var k1, k2, k3, k4, A, B, C: single;
Begin
  k1 := Dbe(d1ob) - Dbe(d3ob);
  k2 := Dbe(d1ob) - Dbe(d2ob);
  k3 := Dce(d1ob) - Dce(d2ob);
  k4 := k1*k3 - k2*(Dce(d1ob)-Dce(d3ob));
  C  := (k1*(e1-e2) - k2*(e1-e3)) / k4;
  B  := (e1-e2-C*k3) / k2;
  A  := e1 - B*Dbe(d1ob) - C*Dce(d1ob);
  Int_eD := A + B*Dbe(dob) + C*Dce(dob)
End;

Function Int_zD(e1, e2, e3, d1ob, d2ob, d3ob, dob: single): single;
Var k1, k2, k3, k4, A, B, C: single;
Begin
  k1 := Dbz(d1ob) - Dbz(d3ob);
  k2 := Dbz(d1ob) - Dbz(d2ob);
  k3 := Dcz(d1ob) - Dcz(d2ob);
  k4 := k1*k3 - k2*(Dcz(d1ob)-Dcz(d3ob));
  C  := (k1*(e1-e2) - k2*(e1-e3)) / k4;
  B  := (e1-e2-C*k3) / k2;
```

```
  A  := e1 - B*Dbz(d1ob) - C*Dcz(d1ob);
  Int_zD := A + B*Dbz(dob) + C*Dcz(dob)
End;

Function Int_F(e1, e2, e3, f1, f2, f3, f: single): single;
Var k1, k2, k3, k4, A, B, C: single;
Begin
  k1 := Fb(f1) - Fb(f3);
  k2 := Fb(f1) - Fb(f2);
  k3 := Fc(f1) - Fc(f2);
  k4 := k1*k3 - k2*(Fc(f1)-Fc(f3));
  C  := (k1*(e1-e2) - k2*(e1-e3)) / k4;
  B  := (e1-e2-C*k3) / k2;
  A  := e1 - B*Fb(f1) - C*Fc(f1);
  Int_F := A + B*Fb(f) + C*Fc(f)
End;

Type data_type = array[1..3,1..3,1..2] of single;

Procedure E_and_Z( Var datum: data_type;
        d1ob, d2ob, d3ob, dob, f1, f2, f3, f: single; Var Eeff,Zeff: single);
Var ee1, ee2, ee3: single;
Begin
  ee1 := Int_F( datum[1,1,1], datum[2,1,1], datum[3,1,1], f1, f2, f3, f );
  ee2 := Int_F( datum[1,2,1], datum[2,2,1], datum[3,2,1], f1, f2, f3, f );
  ee3 := Int_F( datum[1,3,1], datum[2,3,1], datum[3,3,1], f1, f2, f3, f );
  Eeff := Int_eD( ee1, ee2, ee3, d1ob, d2ob, d3ob, dob );
  ee1 := Int_F( Sqr(1/datum[1,1,2]), Sqr(1/datum[2,1,2]),
                Sqr(1/datum[3,1,2]), f1, f2, f3, f );
  ee2 := Int_F( Sqr(1/datum[1,2,2]), Sqr(1/datum[2,2,2]),
                Sqr(1/datum[3,2,2]), f1, f2, f3, f );
  ee3 := Int_F( Sqr(1/datum[1,3,2]), Sqr(1/datum[2,3,2]),
                Sqr(1/datum[3,3,2]), f1, f2, f3, f );
  Zeff := Int_zD( Sqrt(1/ee1), Sqrt(1/ee2), Sqrt(1/ee3),
                  d1ob, d2ob, d3ob, dob );
End;


Var
  i, j, NF, ND : Integer;
  garbage : String[8];
  gar     : string[30];
  infile, outfile : Text;
  f1, f2, f3, d1, d2, d3, deltaf, deltad, k, f, d, Eeff, Zeff : single;
  b, d1ob, d2ob, d3ob: single;
  datum: data_type;

Begin
  Assign( infile, paramstr(1) );
  Reset( infile );
  Readln( infile );  Readln( infile );  Readln( infile );
  Readln( infile, garbage, f1 );
  Readln( infile, garbage, f3 );
  Readln( infile, garbage, deltaF );
  Readln( infile );  Readln( infile );  Readln( infile );
  Readln( infile, garbage, d1 );
  Readln( infile, garbage, d3 );
  Readln( infile, garbage, deltaD );
  Readln( infile );
  for i := 1 to 3 do
  Begin
    Readln( infile );  Readln( infile );  Readln( infile );
    Readln( infile );  Readln( infile );  Readln( infile );
    for j := 1 to 3 do
      Readln( infile, k, datum[i,j,1], datum[i,j,2] )
  End;
  readln( infile );
  readln( infile );
  readln( infile );
  readln( infile );
  readln( infile );
  b := b*0.001;
  Close( infile );

  ND := 11;
  NF := 11;
  DeltaF := (f3-f1)/(NF-1);
  DeltaD := (d3-d1)/(ND-1);
  Assign( outfile, paramstr(2) );
  ReWrite( outfile );
  Writeln( outfile, '__________' );
  Writeln( outfile, '__________' );
  Writeln( outfile );
  Writeln( outfile, 'Fmin   = ', f1:8:5, ' GHz' );
  Writeln( outfile, 'Fmax   = ', f3:8:5, ' GHz' );
  Writeln( outfile, 'DeltaF = ', DeltaF:8:5, ' GHz' );
  Writeln( outfile, 'NF     = ', NF );
  Writeln( outfile );
  Writeln( outfile );
  Writeln( outfile, 'Dmin   = ', d1:8:5, ' mm' );
  Writeln( outfile, 'Dmax   = ', d3:8:5, ' mm' );
  Writeln( outfile, 'DeltaD = ', DeltaD:8:5, ' mm' );
  Writeln( outfile, 'ND     = ', ND );
  Writeln( outfile );
  Writeln( outfile );
  f1 := f1 * 1e9;
  f3 := f3 * 1e9;
  f2 := 0.5*(f1+f3);
  DeltaF := DeltaF * 1e9;
  d1 := d1 * 1e-3;
  d3 := d3 * 1e-3;
  d2 := 0.5*(d1+d3);
  d1ob := d1/b;
  d2ob := d2/b;
  d3ob := d3/b;
```

```
   DeltaD := DeltaD * 1e-3;

   f := f1 - deltaf;
   For i := 1 To NF Do
   Begin
     f := f + deltaf;
     Writeln( outfile, 'f = ', f*1e-9:8:3, 'GHz' );
     Writeln( outfile, 'Sweep over a range of d' );
     Writeln( outfile, '        d in mm              Eeff                   Zeff' );
     Writeln( outfile, '---------------------------------------------------------' );
     d := d1 - deltad;
     For j := 1 to ND Do
     Begin
       d := d + deltad;
       E_and_Z( datum, d1ob, d2ob, d3ob, d/b, f1, f2, f3, f, Eeff, Zeff );
       Writeln( outfile, '      ', d*1e3:8:5, '       ', Eeff:8:5,
                         '       ', Zeff:8:5 )
     End;
     Writeln( outfile, '---------------------------------------------------------' );
     Writeln( outfile )
   End;
   Close( outfile )
 End.
 {***********************}
 {* End of IntPol33.Pas *}
 {***********************}
```

Listing E2: Intpol.Pas

```
                          (* File: Intpol.Pas *)
(*-------------------------------------------------------------------------
 This program is written to generated an 11-by-11 lookup table from a
 3-by-11 lookup table.  The calling syntax is:

                    Intpol TableIn.Dat TableOut.Dat

 Where
    (1) TableIn.Dat is a 3-by-11 lookup table created by either UniFast5.Pas
        or BiFast5.Pas.
    (2) TableOut.Dat is an 11-by-11 lookup table generated by using a
        3-by-11 intpolation scheme described in the thesis.

                                Poman So

           Laboratory for Electromagnetics and Microwaves
                         University of Ottawa
                           August 4, 1988.

-------------------------------------------------------------------------*)
PROGRAM Intpol;

Type
  Table = Record
    NYa: Integer;
    Xname,Yname: Char;
    a,b,f1,f2,f3,A1,A2,A3,A4,A5,YaMin,YaMax,DeltaYa: single;
    Datum : Array[1..3,1..11,1..2] of single;
  End;

Procedure Interpolation( Var Look_Up_Table: Table; f, d: single;
                         Var Eeff, Zeff: single );
Var
  I1, I2: Integer;
  e1, e2, e3, t1, t2, t3, w, u, v, temp: single;
Begin
  With Look_Up_Table Do
  Begin
    If d<YaMin Then d:=YaMin
    Else If d>YaMax Then d:=YaMax;
    I1 := Trunc((d-YaMin)/DeltaYa)+1;
    I2 := I1+1;
    If I2>NYa Then I2:=NYa;
    temp := (d-YaMin)/DeltaYa - I1+1;
    e1 := Datum[1,I1,1] + (Datum[1,I2,1]-Datum[1,I1,1])*temp;
    e2 := Datum[2,I1,1] + (Datum[2,I2,1]-Datum[2,I1,1])*temp;
    e3 := Datum[3,I1,1] + (Datum[3,I2,1]-Datum[3,I1,1])*temp;
    w := (A1*(e1-e2)-A2*(e1-e3))/A4;
    v := (e1-e2-w*A3)/A2;
    u := e1-v*f1-w*A5;
    Eeff := u + v*f + w/(f*f);
    t1 := Datum[1,I1,2];
    t2 := Datum[2,I1,2];
    t3 := Datum[3,I1,2];
    e1 := Sqr(1/(t1 + (Datum[1,I2,2]-t1)*temp));
    e2 := Sqr(1/(t2 + (Datum[2,I2,2]-t2)*temp));
    e3 := Sqr(1/(t3 + (Datum[3,I2,2]-t3)*temp));
    w := (A1*(e1-e2)-A2*(e1-e3))/A4;
    v := (e1-e2-w*A3)/A2;
    u := e1-v*f1-w*A5;
    Zeff := 1/Sqrt(u + v*f + w/(f*f))
  End
End;

procedure output(var look_up_table: Table);
var outfile: text;
    i,j,k: Integer;
    Eeff, Zeff, f, d, DeltaF, DeltaD: single;
```

```
Begin
  With look_up_table do Begin
  DeltaF := (f3-f1)/10;
  DeltaD := (YaMax-YaMin)/10;
  assign(outfile, paramstr(2));
  rewrite( outfile );
  writeln( outfile, 'Laboratory for Electromagnetics and Microwaves, ',
                    'Faculty of Engineering,' );
  writeln( outfile, 'Department of Electrical Engineering, ',
                    'University of Ottawa, Canada.' );
  writeln( outfile, '==============================================',
                    '=======================' );
  writeln( outfile, 'Fmin   = ', f1*1e-9:8:5, ' GHz' );
  writeln( outfile, 'Fmax   = ', f3*1e-9:8:5, ' GHZ' );
  writeln( outfile, 'DeltaF = ', DeltaF*1e-9:8:5, ' GHz' );
  writeln( outfile, 'NF     =  11' );
  writeln( outfile );
  writeln( outfile );
  writeln( outfile, 'Dmin   = ', YaMin*1e3:8:5, ' mm' );
  writeln( outfile, 'Dmax   = ', YaMax*1e3:8:5, ' mm' );
  writeln( outfile, 'DeltaD = ', DeltaD*1e3:8:5, ' mm' );
  writeln( outfile, 'ND     =  11' );
  writeln( outfile );
  writeln( outfile );
  end;
  for i:=0 to 10 do Begin
    f := Look_up_table.f1 + i*DeltaF;
    writeln( outfile, 'f = ', f*1e-9:8:5, ' GHz' );
    writeln( outfile, 'Sweep over a range of d' );
    writeln( outfile, '        d in m              Eeff              Zeff' );
    writeln( outfile, '===============================================' );
    for j:=0 to 10 do Begin
      d := Look_up_table.YaMin + j*DeltaD;
      Interpolation( Look_Up_Table, f, d, Eeff, Zeff );
      Writeln( outfile, '    ',d*1e3:10:5, '    ', Eeff:10:5, '    ', Zeff:10:5 );
    end;
    writeln( outfile, '===============================================' );
    writeln( outfile );
  end;
  close( outfile );
end;

Var infile: text;
    Look_Up_Table: Table;
    f, d, Eeff, Zeff, temp: single;
    string9: string[9];
    string37: string[37];
    XaMin, XaMax, DeltaXa: single;
    i, j, k, NXa: Integer;

Begin
  Assign( infile, paramstr(1) );
  Reset( infile );

  (*********************)
  (* Input data block *)
  (*********************)
  (* Skip three lines of headings *)
  Readln( infile );
  Readln( infile );
  Readln( infile );
  With Look_Up_Table DO
  Begin
    Readln( infile, string9, XaMin );   XaMin   := XaMin*1e9;
    Xname := string9[1];
    Readln( infile, string9, XaMax );   XaMax   := XaMax*1e9;
    Readln( infile, string9, DeltaXa ); DeltaXa := DeltaXa*1e9;
    Readln( infile, string9, NXa );
    Readln( infile ); Readln( infile );
    Readln( infile, string9, YaMin );   YaMin   := YaMin*1e-3;
    Yname := string9[1];
    Readln( infile, string9, YaMax );   YaMax   := YaMax*1e-3;
    Readln( infile, string9, DeltaYa ); DeltaYa := DeltaYa*1e-3;
    Readln( infile, string9, NYa );
    For i:=1 To 3 Do
    Begin
      Readln( infile ); Readln( infile ); Readln( infile );
      Readln( infile ); Readln( infile ); Readln( infile );
      For j:=1 TO NYa Do
        Readln( infile, temp, Datum[i,j,1], Datum[i,j,2] )
    End;
    Readln( infile ); Readln( infile ); Readln( infile );
    Readln( infile, string37, a );
    Readln( infile, string37, b );
    a  := a*0.001;
    b  := b*0.001;
    f1 := XaMin;
    f3 := XaMax;
    f2 := 0.5*(f3+f1);
    A1 := f1-f3;
    A2 := f1-f2;
    A5 := 1/f1/f1;
    A3 := A5-1/f2/f2;
    A4 := A1*A3 - A2*A5 + A2/f3/f3;
  End; (*With*)
  Close( infile );

  output( look_up_table );
End.
(*********************)
(* End of IntPol.Pas *)
(*********************)
```

Listing E3: Error.Pas

```
                              (* File: Error.Pas *)
(*--------------------------------------------------------------------------
 This program is written to compute an error table and the maximum and
 infinity norms of that error table. The calling syntax is:

                 Error Table_1.Dat Table_2.Dat Table.Err

 Where (1) Table_1.Dat and Table_2.Dat are lookup tables of the same size.
       (2) Table.Err is file contains the abovementioned error table. It is
           the difference between Table_1.Dat and Table_2.Dat; see thesis for
           detail. The maximum and infinity norms of this error table are
           stored at the end of this file.

Note that the content of Table.Err can also be graphed by using 3D-Graf.Pas


                                 Poman So

            Laboratory for Electromagnetics and Microwaves
                          University of Ottawa
                            August 4, 1988.

---------------------------------------------------------------------------*)
Program Error;
Var Emax, Zmax, Zerr, Eerr, d, e1, e2, e3, z1, z2, z3: single;
    NF, ND, i, j: Integer;
    file1, file2, file3: Text;
    garbage: String[80];

Begin
  Assign( file1, paramstr(1) );
  ReSet( file1 );
  Assign( file2, paramstr(2) );
  ReSet( file2 );
  Assign( file3, paramstr(3) );
  ReWrite( file3 );
  Zmax := 0.0;
  Emax := 0.0;
  Zerr := 0.0;
  Eerr := 0.0;

  For i := 1 To 13 Do
  Begin
    Readln( file1, garbage );
    Readln( file2 );
    Writeln( file3, garbage );
  End;

  NF := 11;
  ND := 11;

  For i:=1 To NF Do
  Begin

    For j:=1 To  6 Do
    Begin
      Readln( file1 );
      Readln( file2, garbage );
      Writeln( file3, garbage );
    End;

    For j:=1 To ND Do
    Begin
      Readln( file1, d, e1, z1 );
      Readln( file2, d, e2, z2 );
      e3 := (e1-e2)/e1;
      z3 := (z1-z2)/z1;
      Writeln( file3, '          ', d:8:5, '          ', e3:8:5,
                      '          ', z3:8:5 );
      e3 := Abs(e3);
      z3 := Abs(z3);
      Zerr := Zerr + z3;
      Eerr := Eerr + e3;
      IF Emax<e3 Then Emax:=e3;
      IF Zmax<z3 Then Zmax:=z3
    End
  End;

  Writeln( file3 );
  Writeln( file3 );
  Writeln( file3, 'Emax  = ', Emax:8:5 );
  Writeln( file3, 'Zmax  = ', Zmax:8:5 );
  Writeln( file3, 'Eerr  = ', Eerr:8:5 );
  Writeln( file3, 'Zerr  = ', Zerr:8:5 );
  Close( file1 );
  Close( file2 );
  Close( file3 )
End.
{********************}
{* End of Error.Pas *}
{********************}
```

APPENDIX F

Make-Re4.Pas and Make-311.Pas

This appendix contains the source codes of two programs that are used to compress lookup tables from text format to binary format.

Listing F1: Make-Re4.Pas

```
                              (* File: Make-Re4.Pas *)
(*---------------------------------------------------------------------------
 This program is written to convert the text format lookup tables generated
 by UniFastS.Pas and BiFastS.Pas to a binary format that can be used by
 TsiFin.  The data structure declared in this program is exactly the same
 as the one declared in TsiFin. Therefore should the data structure declared
 here be modified the one declared under TsiFin must be changed at the same
 time. The calling syntax is:

                   Make-Re4 TableIn.Dat TableOut.Re4

 Where TableIn.Dat is an text format lookup table and TableOut.Re4 is a
      binary file with real number coded in IEEE 4-byte format.

 Note that:
 (1) The size of the text format lookup tables must be 11-by-11.
 (2) This program must be compiled by Turbo Pascal version 4.0 or later
     because earlier versions does not support IEEE 4-byte real number
     format which is required by TsiFin.

                                  Poman So

              Laboratory for Electromagnetics and Microwaves
                          University of Ottawa
                            August 4, 1988.

-----------------------------------------------------------------------------*)
PROGRAM Make_Re4;

Const
  MaxXa = 11;
  MaxYa = 11;
  NV =  2;

Type
  xCounter = 1..MaxXa;
  yCounter = 1..MaxYa;
  Table = Record
    a, b : Single;
    NXa,Nya: Integer;
    Xname,Yname: Char;
    XaMin,XaMax,DeltaXa,YaMin,YaMax,DeltaYa: Single;
    Entry: Array[1..MaxXa,1..MaxYa,1..NV] of Single;
  End;

 Var
  outfile : file of Table;
  Surface : Table;
  String37: String[37];
  DAT_file, RE4_file: String[80];
  i, j     : Integer;

Procedure ReadData;
Var
  i: xCounter;
  J: yCounter;
  temp     : Single;
  infile   : text;
  garbage  : String[9];
Begin
  Assign( infile, DAT_file );
  Reset( infile );

  {*********************}
  {* Input data block *}
  {********************}
  (* Skip three lines of headings *)
  Readln( infile );
  Readln( infile );
```

```
    Readln( infile );
    With Surface DO
    Begin
      Readln( infile, garbage, XaMin );   XaMin   := XaMin*1e9;
      Xname := garbage[1];
      Readln( infile, garbage, XaMax );   XaMax   := XaMax*1e9;
      Readln( infile, garbage, DeltaXa ); DeltaXa := DeltaXa*1e9;
      Readln( infile, garbage, NXa );
      Readln( infile ); Readln( infile );
      Readln( infile, garbage, YaMin );   YaMin   := YaMin*1e-3;
      Yname := garbage[1];
      Readln( infile, garbage, YaMax );   YaMax   := YaMax*1e-3;
      Readln( infile, garbage, DeltaYa ); DeltaYa := DeltaYa*1e-3;
      Readln( infile, garbage, Nya );
      If (Nxa>MaxXa) or (Nxa<1) or (Nya>MaxYa) or (Nya<1) Then
      Begin
        Writeln( chr(7)'Data file dimension out of range!' );
        Halt(0)
      End;
      For i:=1 To NXa Do
      Begin
        Readln( infile ); Readln( infile ); Readln( infile );
        Readln( infile ); Readln( infile ); Readln( infile );
        For j:=1 TO NYa Do
          Readln( infile, temp, Entry[i,j,1], Entry[i,j,2] )
      End;
      Readln( infile ); Readln( infile ); Readln( infile );
      Readln( infile, String37, a );
      Readln( infile, String37, b );
      a := a*0.001;
      b := b*0.001
    End; (*With*)
    Close( infile );
  End;


  Begin
    Case paramcount of
      2: Begin
           Dat_file := paramstr(1);
           Re4_file := paramstr(2)
         End;
      1: Begin
           Dat_file := paramstr(1);
           Write( 'Please specify output file name > ' ); Readln( Re4_file )
         End;
      Else
         Begin
           Write( 'Please specify input file name  > ' ); Readln( Dat_file );
           Write( 'Please specify output file name > ' ); Readln( Re4_file )
         End;
    End;
    ReadData;
    Assign( outfile, RE4_file );
    ReWrite( outfile );
    Write( outfile, Surface );
    Close( outfile );
  END.
  {**********************}
  {* End of Make-Re4.Pas *}
  {**********************}
```

Listing F2: Make-311.Pas

```
                              (* File: Make-311.Pas *)
(*---------------------------------------------------------------------------
 This program is written to convert the text format lookup tables generated
 by UniFastS.Pas and BiFastS.Pas to a binary format that can be used by
 TsiFin.  The data structure declared in this program is exactly the same
 as the one declared in TsiFin. Therefore should the data structure declared
 here be modified the one declared under TsiFin must be changed at the same
 time. The calling syntax is:

                    Make-311 TableIn.Dat TableOut.311

 Where TableIn.Dat is an text format lookup table and TableOut.311 is a
      binary file with real number coded in IEEE 4-byte format.

 Note that:
 (1) The size of the text format lookup tables must be 3-by-11.
 (2) This program must be compiled by Turbo Pascal version 4.0 or later
     because earlier versions does not support IEEE 4-byte real number
     format which is required by TsiFin.

                                   Poman So

               Laboratory for Electromagnetics and Microwaves
                            University of Ottawa
                              August 4, 1988.

----------------------------------------------------------------------------*)
PROGRAM Make_311;

Const
  MaxXa = 3;
  MaxYa = 11;
```

```
  NV =  2;

 Type
  xCounter = 1..MaxXa;
  yCounter = 1..MaxYa;
  Table = Record
    Xname,Yname: Char;
    a,b,f1,f2,f3,A1,A2,A3,A4,A5,YaMin,YaMax,DeltaYa: Single;
    NYa: Integer;
    Datum : Array[1..MaxXa,1..MaxYa,1..NV] of Single;
  End;

 Var
  i         : xCounter;
  j         : ycounter;
  NXa       : Integer;
  infile    : text;
  outfile   : file of Table;
  Surface   : Table;
  string9   : String[9];
  string37  : String[37];
  DAT_IN, DAT_OUT : String[80];
  temp,XaMin,XaMax,DeltaXa: Single;

 Begin
  Case paramcount of
    2: Begin
         DAT_IN := paramstr(1);
         DAT_OUT := paramstr(2)
       End;
    1: Begin
         DAT_IN := paramstr(1);
         Write( 'Please specify output file name > ' ); Readln( DAT_OUT )
       End;
    Else
       Begin
         Write( 'Please specify input file name  > ' ); Readln( DAT_IN );
         Write( 'Please specify output file name > ' ); Readln( DAT_OUT )
       End;
  End;

  Assign( infile, DAT_IN );
  Reset( infile );
  {*******************}
  {* Input data block *}
  {*******************}
  {* Skip three lines of headings *}
  Readln( infile );
  Readln( infile );
  Readln( infile );
  With Surface DO
  Begin
    Readln( infile, string9, XaMin );   XaMin   := XaMin*1e9;
    Xname := string9[1];
    Readln( infile, string9, XaMax );   XaMax   := XaMax*1e9;
    Readln( infile, string9, DeltaXa ); DeltaXa := DeltaXa*1e9;
    Readln( infile, string9, NXa );
    Readln( infile ); Readln( infile );
    Readln( infile, string9, YaMin );   YaMin   := YaMin*1e-3;
    Yname := string9[1];
    Readln( infile, string9, YaMax );   YaMax   := YaMax*1e-3;
    Readln( infile, string9, DeltaYa ); DeltaYa := DeltaYa*1e-3;
    Readln( infile, string9, Nya );
    For i:=1 To NXa Do
    Begin
      Readln( infile ); Readln( infile ); Readln( infile );
      Readln( infile ); Readln( infile ); Readln( infile );
      For j:=1 TO NYa Do
        Readln( infile, temp, Datum[i,j,1], Datum[i,j,2] )
    End;
    Readln( infile ); Readln( infile ); Readln( infile );
    Readln( infile, string37, a );
    Readln( infile, string37, b );
    a  := a*0.001;
    b  := b*0.001;
    f1 := XaMin;
    f3 := XaMax;
    f2 := 0.5*(f3+f1);
    A1 := f1-f3;
    A2 := f1-f2;
    A5 := 1/f1/f1;
    A3 := A5-1/f2/f2;
    A4 := A1*A3 - A2*A5 + A2/f3/f3;
  End; {*With*}
  Close( infile );
  Assign( outfile, DAT_OUT );
  ReWrite( outfile );
  Write( outfile, Surface );
  Close( outfile );
 END.
{**********************}
{* End of Make-311.Pas *}
{**********************}
```

APPENDIX G

This appendix contains the source code of a module that implements the user defined models described in Chapter 3.

Listing G1: UserProc.Fin

```
                        (* FILE: UserProc.FIN *)
{*************************************************************************}
{**                                                                      **}
{**   This module defines a set of procedures to compute the S-Matrices  **}
{**   for finline line-sections and discontinuity models.                **}
{**                                                                      **}
{**   Field-theory-based lookup tables are used to compute the effective **}
{**   dielectric constant, propagation constant and voltage power impedance **}
{**   for a given finline  geometry. Closed form expressions which have  **}
{**   the above parameters as independent variables are used to compute the **}
{**   equivalent S-Matrices for a number of finline elements.            **}
{**                                                                      **}
{**                                                                      **}
{**                            Poman So                                  **}
{**                                                                      **}
{**          Laboratory for Electomagnetics and MicroWave                **}
{**              Department of Electrical Engineering                    **}
{**                    University of Ottawa                              **}
{**                   Ottawa, Ontario, Canada                            **}
{**                      August 4, 1988.                                 **}
{*************************************************************************}

(* $ floatcalls- *)     (* MS Pascal metacommand for 80x87 coprocessor *)

module userproc [];

(*************************************************************************)

{* This module contains declarations for user defined procedures and the  *}
{* Pascal code for the procedures.  All user defined procedures must call the *}
{* external procedure userdata in order to establish the values specified at *}
{* the time the procedure is invoked in the circuit file.                 *}

(*************************************************************************)

const
  pi = 3.1415927;

{*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*}
{*!                                                                     !*}
{*!  The variables denoted as [extern] (except for the boolean aritherror) !*}
{*!  must not be changed by user defined element procedures.            !*}
{*!                                                                     !*}
{*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*}

var [extern]
  fr: real;                  {* current frequency in Hertz *}
  zo: real;                  {* system characteristic impedance = 50 ohms *}
                             {* or as respecified in the TERM block *}
  aritherror: boolean;       {* error flag *}
  funit: real;               {* frequency scale factor *}
  runit: real;               {* resistance scale factor *}
  gunit: real;               {* conductance scale factor *}
  lunit: real;               {* inductance scale factor *}
  cunit: real;               {* capacitance scale factor *}
  lenunit: real;             {* length scale factor *}
  tunit: real;               {* time scale factor *}
  angunit: real;             {* angle scale factor *}
                             {* user data times scale factor equals MKS units *}

type
  udatype = array[1..10] of real;

var (*local*)                 (* public so that userdata may set values *)
  d: udatype;                {* element data from the circuit file *}
  sbeg: integer;             {* location to begin storing s-parameters *}

(*************************************************************************)

procedure userdata(exloc: integer; var d: udatype; var sbeg: integer); extern;

procedure tosn(k, i: integer; x, y: real); extern;

(*************************************************************************)

{******************** Some complex arithmetics routines *******************}
Type
```

```
   Complex = Record x,y: Real End;
   C2_Matrix = Array[1..2,1..2] of Complex;

 Function AddZ(Z1,Z2:Complex):Complex;
 Begin
   AddZ.x := Z1.x+Z2.x;
   AddZ.y := Z1.y+Z2.y
 End;

 Function SubZ(Z1,Z2:Complex):Complex;
 Begin
   SubZ.x := Z1.x-Z2.x;
   SubZ.y := Z1.y-Z2.y
 End;

 Function MulZ(Z1,Z2:Complex):Complex;
 Begin
   MulZ.x := Z1.x*Z2.x - Z1.y*Z2.y;
   MulZ.y := Z1.x*Z2.y + Z1.y*Z2.x
 End;

 Function DivZ(Z1,Z2:Complex):Complex;
 Var demo:Real;
 Begin
   demo := Z2.x*Z2.x + Z2.y*Z2.y;
   DivZ.x := (Z1.x*Z2.x + Z1.y*Z2.y) / demo;
   DivZ.y := (Z1.y*Z2.x - Z1.x*Z2.y) / demo
 End;

 Function ReMuZ(R:Real;Z:Complex):Complex;
 Begin
   ReMuZ.x := R*Z.x;
   ReMuZ.y := R*Z.y;
 End;

 Function NegZ(Z:Complex):Complex;
 Begin
   NegZ.x := -Z.x;
   NegZ.y := -Z.y
 End;

 Function RepZ(Z:Complex):Complex;
 Var demo:Real;
 Begin
   demo := Z.x*Z.x + Z.y*Z.y;
   RepZ.x := Z.x/demo;
   RepZ.y := -Z.y/demo
 End;

 Function ExpZ(Z:Complex):Complex;
 Var amp:Real;
 Begin
   amp := Exp(Z.x);
   ExpZ.x := amp*Cos(Z.y);
   ExpZ.y := amp*Sin(Z.y)
 End;

 Function CoshZ(Z:Complex):Complex;
 Const Two=Complex(2.0,0.0);
 Begin
   CoshZ := DivZ(AddZ(ExpZ(Z),ExpZ(NegZ(Z))),Two)
 End;

 Function SinhZ(Z:Complex):Complex;
 Const Two=Complex(2.0,0.0);
 Begin
   SinhZ := DivZ(SubZ(ExpZ(Z),ExpZ(NegZ(Z))),Two)
 End;

 Function Mul_C2(C1,C2:C2_Matrix):C2_Matrix;
 Begin
   Mul_C2[1,1] := AddZ(MulZ(C1[1,1],C2[1,1]),MulZ(C1[1,2],C2[2,1]));
   Mul_C2[1,2] := AddZ(MulZ(C1[1,1],C2[1,2]),MulZ(C1[1,2],C2[2,2]));
   Mul_C2[2,1] := AddZ(MulZ(C1[2,1],C2[1,1]),MulZ(C1[2,2],C2[2,1]));
   Mul_C2[2,2] := AddZ(MulZ(C1[2,1],C2[1,2]),MulZ(C1[2,2],C2[2,2]))
 End;

 {*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*}
 {*!                                                     !*}
 {*!  The following two-port matrix formulas are from    !*}
 {*!  K.C.Gupta, Ramesh Garg, Rakesh Chadha              !*}
 {*!  "Computer Aided Design of Microwave Circuits"      !*}
 {*!  Artech House, Inc., 1981.                          !*}
 {*!                                                     !*}
 {*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*}

 {*----------------------------------------------------------------------*}
 {* Convert ABCD matrix to S matrix; the system impedance is ZO on both ends. *}
 {*----------------------------------------------------------------------*}
 Function A_TO_S(A:C2_Matrix):C2_Matrix;
 Var W1,W2,W3,W4: Complex;
 Begin
   W1 := ReMuZ(ZO,A[1,1]);
   W2 := ReMuZ(Sqr(ZO),A[2,1]);
   W3 := ReMuZ(ZO,A[2,2]);
   W4 := RepZ(AddZ(AddZ(W1,A[1,2]),AddZ(W2,W3)));
   A_TO_S[1,1] := MulZ(SubZ(AddZ(W1,A[1,2]),AddZ(W2,W3)),W4);
   A_TO_S[1,2] := MulZ(ReMuZ(2*ZO,SubZ(MulZ(A[1,1],A[2,2]),
                                      MulZ(A[1,2],A[2,1]))),W4);
   A_TO_S[2,1] := ReMuZ(2*ZO,W4);
   A_TO_S[2,2] := MulZ(SubZ(AddZ(A[1,2],W3),AddZ(W1,W2)),W4)
 End;

 Function Line_A_Matrix(Z:Real; GammaL:Complex):C2_Matrix;
 Var
```

```
   Sh,Ch,Ds:Complex;
 Begin
   Sh := SinhZ(GammaL);
   Ch := CoshZ(GammaL);
   Line_A_Matrix[1,1] := Ch;
   Line_A_Matrix[1,2] := ReMuZ(Z,Sh);
   Line_A_Matrix[2,1] := ReMuZ(1/Z,Sh);
   Line_A_Matrix[2,2] := Ch
 End;

 Function Line_S_Matrix(Z:Real; GammaL:Complex):C2_Matrix;
 Var
   R:Real;
   Sh,Ch,Ds:Complex;
 Begin
   Sh := SinhZ(GammaL);
   Ch := CoshZ(GammaL);
   R  := 2*Z*ZO;
   Ds := RepZ(AddZ(ReMuZ(R,Ch),ReMuZ(Z*Z+ZO*ZO,Sh)));
   Line_S_Matrix[1,1] := MulZ(ReMuZ(Z*Z-ZO*ZO,Sh),Ds);
   Line_S_Matrix[1,2] := ReMuZ(R,Ds);
   Line_S_Matrix[2,1] := Result(Line_S_Matrix)[1,2];
   Line_S_Matrix[2,2] := Result(Line_S_Matrix)[1,1]
 End;

 Function Shunt_A_Matrix(Y:Complex):C2_Matrix;
 Begin
   Shunt_A_Matrix[1,1].x := 1;
   Shunt_A_Matrix[1,1].y := 0;
   Shunt_A_Matrix[1,2].x := 0;
   Shunt_A_Matrix[1,2].y := 0;
   Shunt_A_Matrix[2,1]   := Y;
   Shunt_A_Matrix[2,2].x := 1;
   Shunt_A_Matrix[2,2].y := 0
 End;

 Function Series_A_Matrix(Z:Complex):C2_Matrix;
 Begin
   Series_A_Matrix[1,1].x := 1;
   Series_A_Matrix[1,1].y := 0;
   Series_A_Matrix[1,2]   := Z;
   Series_A_Matrix[2,1].x := 0;
   Series_A_Matrix[2,1].y := 0;
   Series_A_Matrix[2,2].x := 1;
   Series_A_Matrix[2,2].y := 0
 End;

 (******************** End of complex arithmetic routines *********************)

 Const
   M1 = 'Record file error!';
   M2 = '<Enter> to return';
   M3 = 'Record file not exist!';

 (****************************************************************************
 This procedure uses two dimensional interpolation to find the values in
 between <name>.Entry[i,j,1] and <name>.Entry[i,j,2]. If x or/and y are
 out of range defined by Surface then they will be set to the boundary
 values nearest to them.  The calling program should ensures that:
 (1) a lookup table of the correct data structure exist in drive D, and
 (2) the return values of x and y are equal to their corresponding entry
     values for correct interpretation of the answers returned from this
     procedure.

 Note that interpolation is first performed on x-component and then the
 y-component.

 ****************************************************************************)
 Procedure LookUpTable((*I/O*)Var x,y:Real; (*Output*)Var wh,Eeff,Zeff:Real);
 Type
   Table = Record
     a,b: Real;
     NXa,Nya: Integer;
     Xname,Yname: Char;
     XaMin,XaMax,DeltaXa,YaMin,YaMax,DeltaYa: Real;
     Entry: Array[1..11,1..11,1..2] of Real;
   End;

 Var
   WR_RE4: LString(15);
   D9: LString(3);
   nx1, nx2, ny1, ny2: Integer;
   DX, DY, Ea, Eb, Za, Zb: Real;
   Surface: Table;
   Rec_file: File of Table;

 BEGIN
   Rec_file.Trap := true;
   WR_RE4 := 'E:WR.RE4';
   aritherror := not Encode(D9,Round(D[9]):3);
   Insert(D9,WR_RE4,5);
   (*****************
   Input Lookup Table
   ******************)
   Assign(Rec_file, WR_RE4);
   Reset(Rec_file);
   IF Rec_file.ERRS=0 THEN
   BEGIN
     Surface:=Rec_file^;
     Get(Rec_file);
     IF not((Surface.Xname='F') and (Surface.Yname='D')) THEN
     BEGIN
       aritherror := true;
       Writeln(Chr(7),M1);
       Writeln(M2,Chr(7) );
       Readln;
```

```
      Return
    END
  END
  ELSE
  BEGIN
    aritherror := true;
    Writeln(Chr(7),M3);
    Writeln(M2,Chr(7));
    Readln;
    Return
  END;
  Close(Rec_file);

  With Surface DO
  BEGIN
    (**********************************
    Make sure fs and ds are within bounds
    **********************************)
    If x > XaMax Then x:=XaMax;
    If x < XaMin Then x:=XaMin;
    If y > YaMax Then y:=YaMax;
    If y < YaMin Then y:=YaMin;

    (********************************
    2 dimensional linear interpolation
    ********************************)
    nx1 := Trunc((x-XaMin)/DeltaXa) + 1;
    If nx1 < NXa Then nx2:=nx1+1
    Else nx2:=nx1;
    DX := (x-XaMin)/DeltaXa - nx1 + 1;
    ny1 := Trunc((y-YaMin)/DeltaYa) + 1;
    If ny1 < NYa Then ny2:=ny1+1
    Else ny2:=ny1;
    DY := (y-YaMin)/DeltaYa - ny1 + 1;
    Ea := Entry[nx1,ny1,1] + DX * (Entry[nx2,ny1,1] - Entry[nx1,ny1,1]);
    Eb := Entry[nx1,ny2,1] + DX * (Entry[nx2,ny2,1] - Entry[nx1,ny2,1]);
    Eeff := Ea + DY * (Eb-Ea);
    Za := Entry[nx1,ny1,2] + DX * (Entry[nx2,ny1,2] - Entry[nx1,ny1,2]);
    Zb := Entry[nx1,ny2,2] + DX * (Entry[nx2,ny2,2] - Entry[nx1,ny2,2]);
    Zeff := Za + DY * (Zb-Za);
    wh := b
  End (*With*)
End; (*-LookUpTable-*)


Procedure NewTable((*I/O*)Var f,y:Real; (*Output*)Var wh,Eeff,Zeff:Real);
Type
  Table = Record
    Xname,Yname: Char;
    a,b,f1,f2,f3,A1,A2,A3,A4,A5,YaMin,YaMax,DeltaYa: Real;
    NYa: Integer;
    Datum : Array[1..3,1..11,1..2] of Real;
  End;
Var
  Rec_file: file of Table;
  Look_Up_Table: Table;
  I1, I2: Integer;
  e1, e2, a3, w, u, v, temp: Real;
  WR_RE4: LString(15);
  D9: LString(3);

Begin
  Rec_file.Trap := true;
  WR_RE4 := 'E:WRA.311';
  aritherror := not Encode(D9,Round(-D[9]):3);
  Insert(D9,WR_RE4,5);
  (*****************
  Input Lookup Table
  *****************)
  Assign(Rec_file, WR_RE4);
  Reset(Rec_file);
  IF Rec_file.ERRS=0 THEN
  BEGIN
    Look_Up_Table:=Rec_file^;
    Get(Rec_file);
    IF not ((Look_Up_Table.Xname='F') and (Look_Up_Table.Yname='D')) THEN
    BEGIN
      aritherror := true;
      Writeln(Chr(7),M1);
      Writeln(M2,Chr(7) );
      Readln;
      Return
    END
  END
  ELSE
  BEGIN
    aritherror := true;
    Writeln(Chr(7),M3);
    Writeln(M2,Chr(7));
    Readln;
    Return
  END;
  Close(Rec_file);

  With Look_Up_Table Do
  Begin
    wh := b;
    If y<YaMin Then y:=YaMin
    Else If y>YaMax Then y:=YaMax;
    if f<f1 Then f:=f1
    Else if f>f3 Then f:=f3;
    I1 := Trunc((y-YaMin)/DeltaYa)+1;
    I2 := I1+1;
    If I2>NYa Then I2:=NYa;
    temp := (y-YaMin)/DeltaYa - I1+1;
    e1 := Datum[1,I1,1] + (Datum[1,I2,1]-Datum[1,I1,1])*temp;
```

```
    e2 := Datum[2,I1,1] + (Datum[2,I2,1]-Datum[2,I1,1])*temp;
    e3 := Datum[3,I1,1] + (Datum[3,I2,1]-Datum[3,I1,1])*temp;
    w := (A1*(e1-e2)-A2*(e1-e3))/A4;
    v := (e1-e2-w*A3)/A2;
    u := e1-v*f1-w*A5;
    Eeff := u + v*f + w/(f*f);
    e1:=Sqr(1/(Datum[1,I1,2] + (Datum[1,I2,2]-Datum[1,I1,2])*temp));
    e2:=Sqr(1/(Datum[2,I1,2] + (Datum[2,I2,2]-Datum[2,I1,2])*temp));
    e3:=Sqr(1/(Datum[3,I1,2] + (Datum[3,I2,2]-Datum[3,I1,2])*temp));
    w := (A1*(e1-e2)-A2*(e1-e3))/A4;
    v := (e1-e2-w*A3)/A2;
    u := e1-v*f1-w*A5;
    Zeff := Sqrt(1/(u + v*f + w/(f*f)))
  End
End; (*-NewTable-*)

(*-------------------------------------------------------------
Procedure to alarm user when gap width is out of valid range.
-------------------------------------------------------------*)
Procedure GapError;
Begin
  aritherror := true;
  Writeln( Chr(7), 'Gap Width out of range!');
  Writeln('<Enter> to return.', Chr(7));
  Readln
End;

(*************************************************************************
     The following procedures use the equations in [1] and [2] to compute the
discountinuity effects for finline analysis.  The first procedure computes
the impedance due to the step effects and the second procedure computes the
shunt inductive effects.

Reference:
[1]   Y. L. Tsui  and   Wolfgang J. R. Hoefer
      EMPIRICAL FORMULAE FOR THE PARAMETERS OF IMPEDANCE STEPS AND
      INDUCTIVE STRIPS IN FINLINE.

[2]   P. Saguet, A. Coumes
      Filtres Hyperfrequences en Technologie Finline

Department of Electrical Engineering                   University of Ottawa
*************************************************************************)

(*------------------------------------------------------------------------
This procedure implements (1) and (2) of [1]. If d2_o_d1 is out of the valid
range defined by the formulas then d2_o_d1 would be set to the boundary
value nearest to the entry value of d2_o_d1; L and Z2_o_Z1 are inductance and
impedance ratio at the corresponding d2_o_d1 boundary. The calling program
should ensure that the entry value of d2_o_d1 equal to the return value of
d2_o_d1 for correct interpretation of the return values of L and Z2_o_Z1.

where
  d2_o_d1 : d2/d1; valid range is [1.0, 13.6]
  L       : inductance in pH -- (2) of [1]
  Z2_o_Z1 : Z2/Z1 ------ (1) of [1]

Note that the following conditions are assumed:

                      a
        ----------------------------
        |              |   |           |        b/a = 1/2
   b    |          --->| s |<--        |        s/a = 1/28
        |              |Er |           |        Er  = 2.2
        |              |   |           |        0.32 < b/Lamder < 0.47
        ----------------------------

        ---------------------------
                          ---------                         jX
        -----------------|                   ---------  ******  ---------
              d1               d2       --          Z1                   Z2
        -----------------
                          |---------
        ---------------------------                ----------------------------
-------------------------------------------------------------------------*)
Procedure ImpStep( Var (*input/output*) d2_o_d1,
                       (*---/output*) L, Z2_o_Z1: Real );
BEGIN
  (*********************************************
  Make d2_o_d1 within valid range if necessary.
  *********************************************)
  (*
  IF d2_o_d1 <1 THEN d2_o_d1:=1;
  IF d2_o_d1 >13.6 THEN d2_o_d1:=13.6;
  *)
  (*************************************
  Compute inductance and impedance ratio.
  *************************************)
  L := 19.14 - (31.275 - (14.56 - 0.5014*d2_o_d1) * d2_o_d1 ) * d2_o_d1;
  IF d2_o_d1<=8.8 THEN
    Z2_o_Z1 := 0.7025 + 0.2975*d2_o_d1
  ELSE
    Z2_o_Z1 := 1.8790 + 0.1639*d2_o_d1;
  L := L*1.0E-12; (*Convert L from pH to H*)
END; (*-ImpStep-*)
```

```
(*----------------------------------------------------------------------
This procedure implements formulas in [2].

                          a
         ------------------------------------
         |              |    |              |
         |              |    |              |
     b   |          --->|  s |<---          |
         |              |    |              |
         |              |fr  |              |
         ------------------------------------

         |<-- l -->|           |<-- l -->|
         -----------           -----------           <----- 2 delta l ----->
                                                     -----------------------
                     |          |                               |
              d      |<---W--->|      d          ==              :  -jB
         ------------|          |-----------                    :
                                                                :
         --------------------------------------                 |
                                                     -----------------------
--------------------------------------------------------------------*)
Procedure IndStrip( Var (*input/output*) w_O_b, b_O_d, b_O_n,
                        (*---/output*) B_O_Y1, Dl_O_b: Real );
BEGIN
  Dl_O_b := 0.202 * Exp(0.157*Ln(b_O_n) - 0.37*Ln(b_O_d));
  IF w_O_b < 1.0 THEN
    Dl_O_b := Dl_O_b * Exp(0.38*Ln(w_O_b));
  B_O_Y1 := 0.455/b_O_n*Exp(0.715*Ln(b_O_d)) +
            326 * Exp( 3*Ln(1-1.3*b_O_n) - 2.63/b_O_d ) * w_O_b;
  IF w_O_b > 0.4 THEN
    B_O_Y1 := B_O_Y1 + 0.16 * Exp( 0.8*Ln(b_O_d) + 1.1*Ln(Exp(1/b_O_n))) *
              (Exp(1.3*w_O_b) - 1.68 );
END; (*-IndStrip-*)

(************************ USER DEFINED MODELS ******************************)
procedure ulpa(exloc: integer) [public];
      (*--------------------------------------------------------------
      A match load for a finlin section.
      Parameters: g
      Units: SI

                ---------------------------- - - -
               |
               |                metal
               |
               |---------------------------- - - -
      input                 g: gap width     ---> INFINITE LONG
                ---------------------------- - - -       (ie. Matched)
               |
               |                metal
               |
               |---------------------------- - - -

      D[1] = g [unit length]
      D[9] = WR : file number, eg 281 stand for WR281.RE4

      D[2] to D[8] and  D[10] are not used.
      ---------------------------------------------------------------*)
var
  freq, b, g, gg, Eeff, Zeff, x, y: real;
begin
  userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
  g := D[1]*LENunit;
  gg:= g;
  freq := fr;
  (***********
  Computes S11
  ***********)
  IF D[9]>0 THEN
    LookUpTable( freq, g, b, Eeff, Zeff )
  ELSE
    NewTable( freq, g, b, Eeff, Zeff );
  IF aritherror THEN return;
  IF (gg<>g) THEN
  BEGIN
    (**********************************************
    Gap Width out of range, Eeff and Zeff are invalid.
    **********************************************)
    GapError;
    Return
  END;

  x := (Zeff-Z0)/(Zeff+Z0);
  y := 0.0;
  tosn(sbeg, 0, x, y)
end;
(*************************************************************************)
procedure ulpb(exloc: integer) [public];
(*var
```

```
  user variable declarations go here *)

 begin
   userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
   aritherror := true;    (* temporary until procedure is defined by user *)

 (* user code goes here *)

 (* the following line should become active code when procedure is defined *)
 (* tosn(sbeg, 0, x, y); *)     (* x -> s11r, y -> s11i *)
 end;

 (****************************************************************************)

 procedure u2pa(exloc: integer) [public];
       (*------------------------------------------- -------------
       A dispersive finline model.

              |<------------------- L -------------------->|
              ----------------------------------------------
              |                                            |
              |                  metal                     |
              |--------------------------------------------|

                              g: gap width
              ----------------------------------------------
              |                                            |
              |                  metal                     |
              |--------------------------------------------|

       D[1]  = g     [unit length]
       D[2]  = l     [unit length]
       D[3]  = loss  [Neper / unit length]
       D[9]  = WR    - see U1PA
       D[10] = Flag  - If set to '1', current freq would be used to
                       calculate electrical length of the line
                       despite freq out of the valid range of the
                       lookup table.

       D[4] to D[8] are not used.
       ---------------------------------------------------------*)
Var
   S: C2Matrix;
   GammaL: Complex;
   b,g,gg,L,freq,Keff,Zeff: Real;
Begin
   userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
   g := D[1]*LENunit;
   gg:= g;
   L := D[2]*LENunit;
   GammaL.x := D[3]*D[2]; (*Lossy part of the propagation constant.*)
   freq := fr;
   (*****************************************************
   Compute Keff and Zeff,note that 'freq' must first be saved;
   see documentation of 'LookUpTable'.
   *****************************************************)
   IF D[9]>0 THEN
     LookUpTable( freq, g, b, Keff, Zeff )
   ELSE
     NewTable( freq, g, b, Keff, Zeff );
   IF aritherror THEN return;
   IF (gg<>g) THEN
   BEGIN
     (**********************************************
     Gap Width out of range, Keff and Zeff are invalid.
     **********************************************)
     GapError;
     Return
   END;

   (*******************************************************************
   Use the returned freq to compute S-Matrix so that if the entry freq is
   beyond the valid range of the lookup table the user can notice a dis-
   continuity effect. This option can be over ride by the user by setting
   D[10] to 1.
   *******************************************************************)
   IF D[10]=1 THEN freq:=fr;
   GammaL.y := pi * Sqrt(Keff) * L * freq / 1.5E8;
   S := Line_S_Matrix( Zeff, GammaL );
   tosn(sbeg,0,S[1,1].x,S[1,1].y);
   tosn(sbeg,1,S[1,2].x,S[1,2].y);
   tosn(sbeg,2,S[2,1].x,S[2,1].y);
   tosn(sbeg,3,S[2,2].x,S[2,2].y)
end; (*u2pa*)

(****************************************************************************)

procedure u2pb(exloc: integer) [public];
     (*---------------------------------------
     See documentation of IndStrip.

     |<-- l1 -->|        |<-- l2 -->|

     ------------        ------------
                |        |
         g      |<--W--> |     g
     ------------        ------------
```

```
      ----------------------------
      D[1] = g     [unit length]
      D[2] = l1    [unit length]
      D[3] = w     [unit length]
      D[4] = l2    [unit length]
      D[5] = loss  [Neper / unit length]
      D[9] = WR    - see U1PA
      D[10] = Flag - see U2PA

      D[6], D[7] and D[8] are not used.
      ---------------------------------*)
  var
    GammaL,Y: Complex;
    A1,A2,S: C2_Matrix;
    freq,Eeff,Zeff,g,gg,bb,w,L1,L2,temp,
    B,L,A,w_O_b,b_O_d,b_O_n,B_O_Y,Dl_O_b:Real;
  begin
    userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
    freq := fr;
    g := LENunit*D[1];
    gg:= g;
    w := LENunit*D[3];
    A := D[5]/LENunit; (*Alpha: real part of the propagation constant.*)

    IF D[9]>0 THEN
      LookUpTable( freq, g, bb, Eeff, Zeff )
    ELSE
      NewTable( freq, g, bb, Eeff, Zeff );
    IF aritherror THEN return;
    IF (gg<>g) THEN
    BEGIN
      (*********************************************
      Gap Width out of range, Eeff and Zeff are invalid.
      *********************************************)
      GapError;
      Return
    END;

    (**************************************************************
    Use the returned freq to compute S-Matrix so that if the entry freq is
    beyond the valid range of the lookup table the user can notice a dis-
    continuity effect. This option can be over ride by the user by setting
    D[10] to 1.
    **************************************************************)
    IF D[10]=1 THEN freq:=fr;
    (*************************************
    Compute shunt inductance and delta l.
    *************************************)
    w_O_b := w/bb;
    b_O_d := bb/g;
    b_O_n := bb * sqrt(Eeff) * freq / 3.0e8;
    IndStrip( w_O_b, b_O_d, b_O_n, B_O_Y, Dl_O_b );
    L1 := Dl_O_b*bb + LENunit*D[2];
    L2 := Dl_O_b*bb + LENunit*D[4];
    temp := pi*sqrt(Eeff)*freq/1.5E8;
    (*******************************************************
    Compute ABCD-Matrixs and cascade them to form a S-Matrix.
    *******************************************************)
    GammaL.x := A * L1;
    GammaL.y := temp * L1;
    A1 := Line_A_Matrix(Zeff,GammaL);
    B := B_O_Y/Zeff;
    Y.y := -B;
    Y.x := 0;
    A2 := Mul_C2(A1,Shunt_A_Matrix(Y));
    IF L1<>L2 THEN
    Begin
      GammaL.x := A * L2;
      GammaL.y := temp * L2;
      A1 := Line_A_Matrix(Zeff,GammaL)
    End;
    S := A_TO_S(Mul_C2(A2,A1));
    tosn(sbeg,0,S[1,1].x,S[1,1].y);
    tosn(sbeg,1,S[1,2].x,S[1,2].y);
    tosn(sbeg,2,S[2,1].x,S[2,1].y);
    tosn(sbeg,3,S[2,2].x,S[2,2].y)
  end;

(******************************************************************************)
 procedure u2pc(exloc: integer) [public];
    (*----------------------------------
     |<---- l1 ---->|<---- l2 ---->|
     ------------------------------

     ---------------|  -------------

          g1               g2
     --------------
                    |--------------

     ------------------------------

     See documentation of ImpStep.

     D[1] = g1    [unit length]
     D[2] = l1    [unit length]
     D[3] = g2    [unit length]
     D[4] = l2    [unit length]
     D[5] = loss  [Neper / unit length]
     D[9] = WR    - see U1PA
     D[10] = Flag - see U2PA
```

```
    D[5] to D[8] are not used.
    ---------------------------------*)
label err;
var
  GammaL,Zs: Complex;
  S,ABCD: C2_Matrix;
  A,b,g1,gg1,gg2,g2,L,LL,L1,L2,
  d2_o_d1,Z2_o_Z1,freq,Eeff1,Eeff2,Zeff1,Zeff2: Real;
begin
  userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
  g1 := LENunit*D[1];
  gg1 := g1;
  L1 := LENunit*D[2];
  g2 := LENunit*D[3];
  gg2 := g2;
  L2 := LENunit*D[4];
  A := D[5]/LENunit; (*Alpha; real part of the propagation constant.*)
  freq := fr;
  IF g2>g1 THEN
    d2_o_d1 := g2/g1
  ELSE
    d2_o_d1 := g1/g2;
  ImpStep( d2_o_d1, LL, Z2_o_Z1 );
  (*************************************************************
  Compute ABCD-Matrixs and then cascade them to form a single S-Matrix.
  *************************************************************)
  IF D[9]>0 THEN
    LookUpTable( freq, g1, b, Eeff1, Zeff1 )
  ELSE
    NewTable( freq, g1, b, Eeff1, Zeff1 );
  IF aritherror THEN return;
  IF gg1<>g1 THEN Goto ERR;

  IF D[9]>0 THEN
    LookUpTable( freq, g2, b, Eeff2, Zeff2 )
  ELSE
    NewTable( freq, g2, b, Eeff2, Zeff2 );
  IF aritherror THEN return;
  IF gg2<>g2 THEN Goto ERR;

  IF D[10]=1 THEN freq:=fr;

  Zs.x := 0.0;
  Zs.y := 2*pi*freq*LL;
  ABCD := Series_A_Matrix(Zs);
  IF L1<>0 THEN
  Begin
    GammaL.x := L1*A;
    GammaL.y := L1*pi*sqrt(Eeff1)*freq/1.5e8;
    ABCD := Mul_C2(Line_A_Matrix(Zeff1,GammaL),ABCD)
  End;
  IF L2<>0 THEN
  Begin
    GammaL.x := L2*A;
    GammaL.y := L2*pi*sqrt(Eeff2)*freq/1.5e8;
    ABCD := Mul_C2(ABCD,Line_A_Matrix(Zeff2,GammaL))
  End;
  S := A_TO_S(ABCD);

  tosn(sbeg,0,S[1,1].x,S[1,1].y);
  tosn(sbeg,1,S[1,2].x,S[1,2].y);
  tosn(sbeg,2,S[2,1].x,S[2,1].y);
  tosn(sbeg,3,S[2,2].x,S[2,2].y);
  Return;
  (*******************************************
  Gap Width out of range, Eeff and Zeff are invalid.
  *******************************************)
ERR: GapError;
end;

(***************************************************************************)

procedure u2pd(exloc: integer) [public];

(*var
  user variable declarations go here *)

begin
  userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
  aritherror := true;       (* temporary until procedure is defined by user *)

(* user code goes here *)

(* the following 4 lines should become active code when procedure is defined *)
(* tosn(sbeg, 0, x, y); *)      (* x => s11r, y => s11i *)
(* tosn(sbeg, 1, x, y); *)      (* x => s12r, y => s12i *)
(* tosn(sbeg, 2, x, y); *)      (* x => s21r, y => s21i *)
(* tosn(sbeg, 3, x, y)  *)      (* x => s22r, y => s22i *)
end;

(***************************************************************************)

procedure u3pa(exloc: integer) [public];

(*var
  user variable declarations go here *)

begin
  userdata(exloc, d, sbeg); (* sets d[1] .. d[10], sbeg *)
  aritherror := true;       (* temporary until procedure is defined by user *)

(* user code goes here *)

(* the following 9 lines should become active code when procedure is defined *)
(* tosn(sbeg, 0, x, y); *)      (* x => s11r, y => s11i *)
(* tosn(sbeg, 1, x, y); *)      (* x => s12r, y => s12i *)
```

```
{* tosn(sbeg, 2, x, y); *}      {* x => s13r, y => s13i *}
{* tosn(sbeg, 3, x, y); *}      {* x => s21r, y => s21i *}
{* tosn(sbeg, 4, x, y); *}      {* x => s22r, y => s22i *}
{* tosn(sbeg, 5, x, y); *}      {* x => s23r, y => s23i *}
{* tosn(sbeg, 6, x, y); *}      {* x => s31r, y => s31i *}
{* tosn(sbeg, 7, x, y); *}      {* x => s32r, y => s32i *}
{* tosn(sbeg, 8, x, y)  *}      {* x => s33r, y => s33i *}
end;

(**************************************************************************)

procedure u3pb(exloc: integer) [public];

(*var
  user variable declarations go here *)

begin
  userdata(exloc, d, sbeg); {* sets d[1] .. d[10], sbeg *}
  aritherror := true;       {* temporary until procedure is defined by user *}

(* user code goes here *)

{* the following 9 lines should become active code when procedure is defined *}
{* tosn(sbeg, 0, x, y); *}      {* x => s11r, y => s11i *}
{* tosn(sbeg, 1, x, y); *}      {* x => s12r, y => s12i *}
{* tosn(sbeg, 2, x, y); *}      {* x => s13r, y => s13i *}
{* tosn(sbeg, 3, x, y); *}      {* x => s21r, y => s21i *}
{* tosn(sbeg, 4, x, y); *}      {* x => s22r, y => s22i *}
{* tosn(sbeg, 5, x, y); *}      {* x => s23r, y => s23i *}
{* tosn(sbeg, 6, x, y); *}      {* x => s31r, y => s31i *}
{* tosn(sbeg, 7, x, y); *}      {* x => s32r, y => s32i *}
{* tosn(sbeg, 8, x, y)  *}      {* x => s33r, y => s33i *}
end;

(**************************************************************************)

procedure u4pa(exloc: integer) [public];

(*var
  user variable declarations go here *)

begin
  userdata(exloc, d, sbeg); {* sets d[1] .. d[10], sbeg *}
  aritherror := true;       {* temporary until procedure is defined by user *}

(* user code goes here *)

{* the following 16 lines should become active code when procedure is defined *}
{* tosn(sbeg,  0, x, y); *}     {* x => s11r, y => s11i *}
{* tosn(sbeg,  1, x, y); *}     {* x => s12r, y => s12i *}
{* tosn(sbeg,  2, x, y); *}     {* x => s13r, y => s13i *}
{* tosn(sbeg,  3, x, y); *}     {* x => s14r, y => s14i *}
{* tosn(sbeg,  4, x, y); *}     {* x => s21r, y => s21i *}
{* tosn(sbeg,  5, x, y); *}     {* x => s22r, y => s22i *}
{* tosn(sbeg,  6, x, y); *}     {* x => s23r, y => s23i *}
{* tosn(sbeg,  7, x, y); *}     {* x => s24r, y => s24i *}
{* tosn(sbeg,  8, x, y); *}     {* x => s31r, y => s31i *}
{* tosn(sbeg,  9, x, y); *}     {* x => s32r, y => s32i *}
{* tosn(sbeg, 10, x, y); *}     {* x => s33r, y => s33i *}
{* tosn(sbeg, 11, x, y); *}     {* x => s34r, y => s34i *}
{* tosn(sbeg, 12, x, y); *}     {* x => s41r, y => s41i *}
{* tosn(sbeg, 13, x, y); *}     {* x => s42r, y => s42i *}
{* tosn(sbeg, 14, x, y); *}     {* x => s43r, y => s43i *}
{* tosn(sbeg, 15, x, y)  *}     {* x => s44r, y => s44i *}
end;

(**************************************************************************)

procedure u4pb(exloc: integer) [public];

(*var
  user variable declarations go here *)

begin
  userdata(exloc, d, sbeg); {* sets d[1] .. d[10], sbeg *}
  aritherror := true;       {* temporary until procedure is defined by user *}

(* user code goes here *)

{* the following 16 lines should become active code when procedure is defined *}
{* tosn(sbeg,  0, x, y); *}     {* x => s11r, y => s11i *}
{* tosn(sbeg,  1, x, y); *}     {* x => s12r, y => s12i *}
{* tosn(sbeg,  2, x, y); *}     {* x => s13r, y => s13i *}
{* tosn(sbeg,  3, x, y); *}     {* x => s14r, y => s14i *}
{* tosn(sbeg,  4, x, y); *}     {* x => s21r, y => s21i *}
{* tosn(sbeg,  5, x, y); *}     {* x => s22r, y => s22i *}
{* tosn(sbeg,  6, x, y); *}     {* x => s23r, y => s23i *}
{* tosn(sbeg,  7, x, y); *}     {* x => s24r, y => s24i *}
{* tosn(sbeg,  8, x, y); *}     {* x => s31r, y => s31i *}
{* tosn(sbeg,  9, x, y); *}     {* x => s32r, y => s32i *}
{* tosn(sbeg, 10, x, y); *}     {* x => s33r, y => s33i *}
{* tosn(sbeg, 11, x, y); *}     {* x => s34r, y => s34i *}
{* tosn(sbeg, 12, x, y); *}     {* x => s41r, y => s41i *}
{* tosn(sbeg, 13, x, y); *}     {* x => s42r, y => s42i *}
{* tosn(sbeg, 14, x, y); *}     {* x => s43r, y => s43i *}
{* tosn(sbeg, 15, x, y)  *}     {* x => s44r, y => s44i *}
end;

(**************************************************************************)

end.                    (* End of module UserProc.FIN *)
```

APPENDIX H

The Calling Syntax for the Finline Elements

TsiFin contains a set of user defined elements for finline line-sections and discontinuities. Field-theory based lookup tables are used to determine the effective dielectric constant, ϵ_{eff}, and voltage power impedance, Z_{eff}, of the transmission medium to be used such as unilateral or bilateral finlines. The S-matrices of the implemented discontinuities are computed by using closed form expressions which require ϵ_{eff} and Z_{eff} as independent variables. To run *TsiFin*, follow the steps listed below.

If the required lookup tables (.Re4 and .311 files) are in the Sr-Re4 directory, type TSTONE to run the program. Else run *UniFastS* or *BiFastS* to generate the required .Dat files and then run *Make-Re4* or *Make-311* to generate the required lookup tables. The naming convention of the lookup tables is described in the following page†.

The .Dat files created by running *UniFastS* and *BiFastS* can be displayed graphically by using the graphic program *3D-Graf*; see Appendix C for details about this program.

There are a large number of example circuit files in the EEsof\SR-Re4\Example directory. In fact the example files in that directory are all given in Appendix J. A simple, but detailed, design example is given in Appendix I.

The calling syntax for the finline elements is described in the following pages. All the rules applicable to the regular Touch Stone circuit elements are also applicable to the finline elements except the rule for specifing "dielectric loss tangent". The syntax for entering a user-defined element in a circuit file is:

$$\text{U}n\text{P}x < \text{node\#}s > \ \text{D1} = a1 \ \ \text{D2} = a2 \ \ \text{D3} = a3 \ldots \ \ \text{D10} = a10$$

where

n = number of ports (1 or 2) defined by the model

x = "a" for 1-port; "a", "b" or "c" for 2-port

D1, D2, D3 ... D10 = element value keywords; all keywords must be entered and assigned values even if the model does not use them.

$a1, a2, a3 \ldots a10$ = data values, use "0" if the data is not used

Element statements CANNOT be longer than 80 characters or extend to a second line. To fit all values on one line, variables may be assigned in the VAR block of the circuit file.

† A detailed description of the naming convention is given in Chapter 2 of this thesis.

U1PA — A wideband matching load, good for unilateral and bilateral finlines.

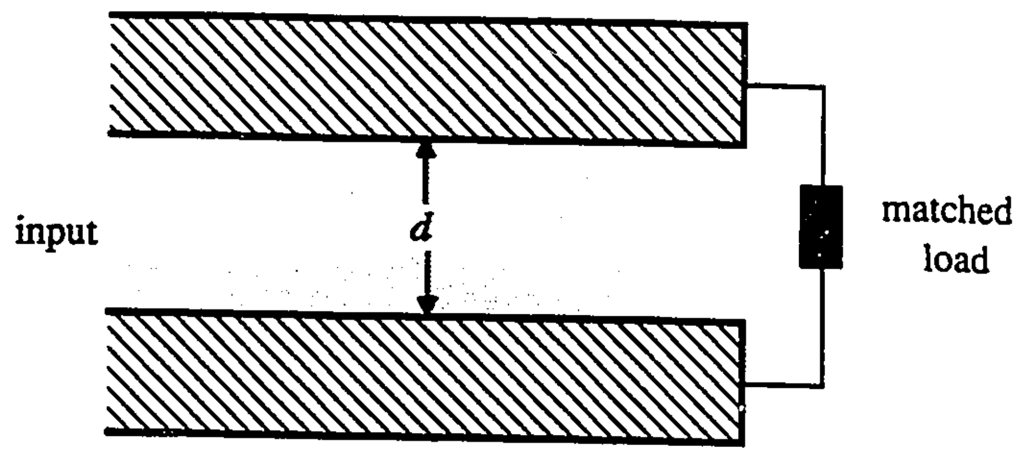


Figure H1: A wideband matching load.

D1 = d [unit length]

D9 = WR, eg 281 stand for WR281.Re4 and -281 stand for WR281A.311.

D2 to D8 and D10 are not used.

Note that .Re4 and .311 files are corresponding to lookup tables of size of 11–by–11 and 3–by–11 respectively. The naming convention of the lookup tables and their associated interpolation schemes are described in Chapter 2 and 3.

U2PA — A finline line-section, good for unilateral and bilateral finlines.

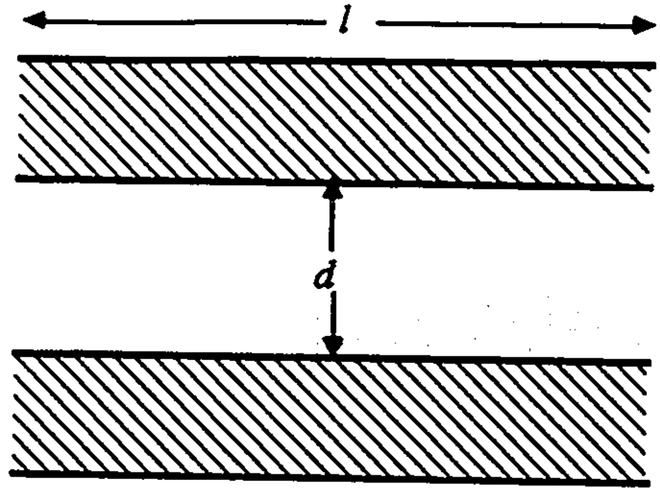


Figure H2: A finline line-section.

D1 = *d* [unit length]

D2 = *l* [unit length]

D3 = *loss* [Nepers / unit length]

D9 = `WR` — see U1PA

D10 = Flag — If set to "1", current frequency would be used to calculate electrical length of the line despite the frequency being out of range of the lookup table or else the boundary frequency would be used.

D4 to D8 are not used.

U2PB — A shunt discontinuity element, good for unilateral and bilateral finlines.

Note that this model is good under the following condition

$$0 < w/b < 1$$

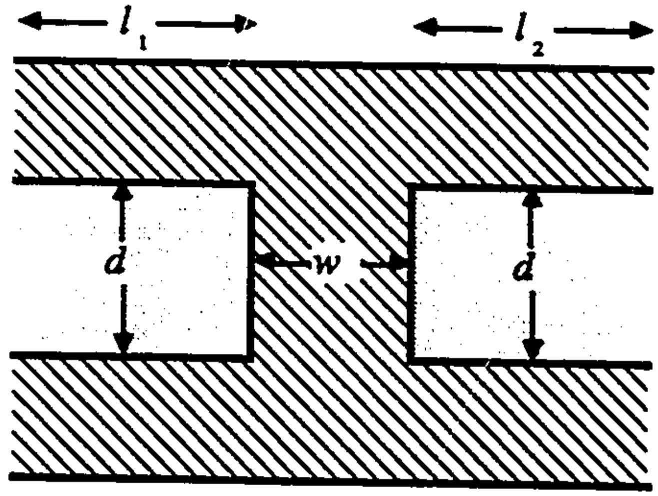


Figure H3: A shunt discontinuity element.

D1 = d [unit length]

D2 = l_1 [unit length]

D3 = w [unit length]

D4 = l_2 [unit length]

D5 = *loss* [Nepers / unit length]

D9 = WR — see U1PA

D10 = Flag — see U2PA

Reference: P. Saguet and A. Coumes, "Filtres Hyperfrequences en Technologie Finline".

U2PC — A step discontinuity element, good for unilateral finline.

Note this model is good under the following conditions:

$$1 < d_2/d_1 < 13.6$$
$$b/\lambda \in [0.32, 0.47]$$
$$s/a = 1/28$$
$$b/a = 1/2$$
$$\epsilon_r = 2.22$$

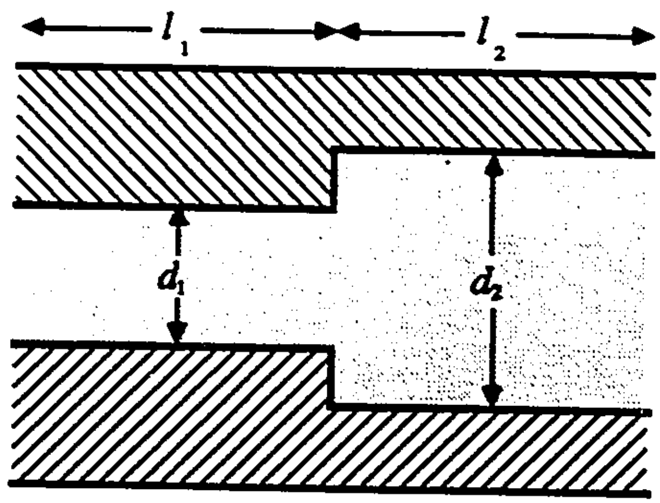


Figure H4: A step discontinuity element.

D1 = d_1 [unit length]

D2 = l_1 [unit length]

D3 = d_2 [unit length]

D4 = l_2 [unit length]

D5 = *loss* [Nepers / unit length]

D9 = WR — see U1PA

D10 = Flag — see U2PA

D6, D7 and D8 are not used.

Reference: Y. L. Tsui and Wolfgang J. R. Hoefer, "Empirical Formulae for the Parameters of Impedance Steps and Inductive Strips in Finline" in IEEE Montech'86 Symposium Digest, pp.36-38, Montreal.

APPENDIX I

A Design Example

This example is used to illustrate all the capabilities of this new CAD procedure; from lookup table generation to circuit optimization†. The problem to be solved is a simple impedance matching problem. It is required to use a unilaterial finline line section to match two impedances at 32.5 GHz. The source impedance Z_S is 200 Ω, and the load impedance Z_L is 400 Ω. The finline line section is to be placed in a WR28 waveguide. The fins are to be plated on RT duroid substrate, $\epsilon_r = 2.22$, of thickness 0.254 mm. The specifications are summarized in Figure I1.

$a = 7.112$ mm
$b = 3.556$ mm
$h = 3.556$ mm
$s = 0.254$ mm
$\epsilon_r = 2.22$
Source = 200 Ω
Load = 400 Ω

Find d and l to match the impedance at 32.5 GHz.

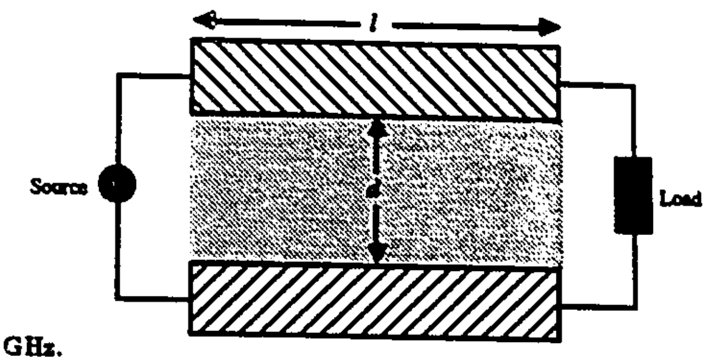


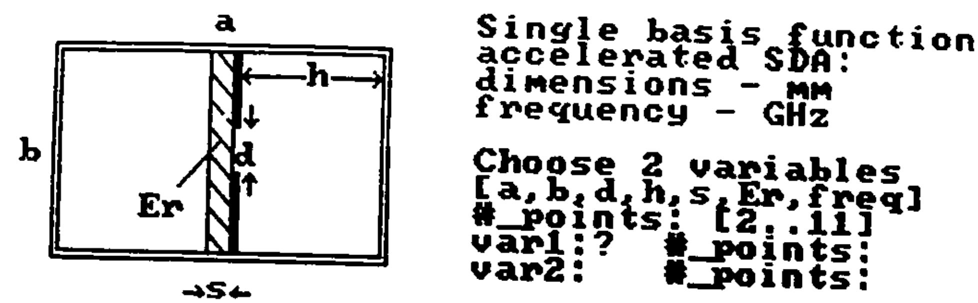
Figure I1: Design specification of a simple finline transformer.

According to Figure 4-1, generation of a lookup table is required; if there is no suitable lookup table available in the lookup table library. This can be easily done by running one of the table generation programs, *UniFastS*. To run this program just type "UNIFASTS" under DOS prompt. A graphic menu identical to the one shown in Figure I2(a) will appear on the screen. The question mark sitting right beside "var1:" is a request for input. Frequency (f) must be chosen as the first sweeping variable. After entering f the question mark will appear beside "#_points:"; "3" can be entered to choose three frequency points per gapwidth. Similarly, gapwidth (d) must be chosen as the second sweeping variable and "11" must be entered to specify eleven gapwidth points per frequency.

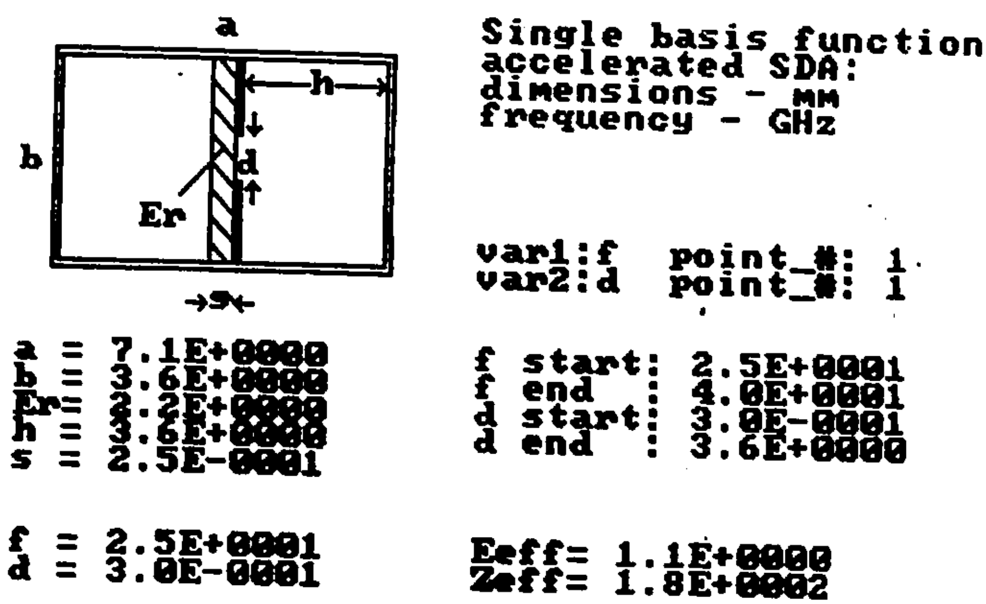
After the mesh variables (f,d) and mesh size (3,11) are specified, the remaining physical parameters can be entered easily in response to the program prompt: "7.112"

† Hence it is assumed that: (1) You have access to one of the CAD work stations in the Laboratory for Electromagnetics and Microwaves. (2) You are familiar with Touch Stone. (3) The current directory is C:\EESOF\Sr-RE4, if not type "C:" then type "CD\EESOF\SR-RE4" in response to DOS prompt.

for "a", "3.556" for "b", "2.22" for "Er", "3.556" for "h"and "0.254" for "s", "25" for "f start:", "40" for "f end:", "0.3" for "d start" and "3.556" for "d end"; where "f start", "f end", "d start" and " d end" are used to specify the frequency and gapwidth ranges respectively. When the parameters are successfully entered, an output such as Figure I2(b) would appear on the screen. This indicates that the program has successfully computed the ϵ_{eff} and Z_{eff} at mesh point (1,1) and at that mesh point, frequency is 25 GHz, gapwidth is 0.3 mm, effective dielectric constant is 1.1 and effective characteristic impedance is 180 Ω. Upon termination of this program a lookup table identical to Table B2 in Appendix B will be stored in the file *UniFastS.Dat*.



(a) A *UniFastS* input menu.



(b) *UniFastS* finished the computation at data point (1,1), at this data point $f = 25\,\mathrm{GHz}, d = 0.3\,\mathrm{mm}, \epsilon_{eff} = 1.1$ and $Z_{eff} = 180\,\Omega$.

Figure I2: Screen outputs from UniFastS.

The next step is to save the data file *UniFastS.Dat*. A meaningful filename should be used so that it is possible to recall the critical information of this lookup table. According to Table 2D-1 this file should be named *WR281311.Dat*. "WR28" means WR28 waveguide, "1" means unilateral finline, "311" means that the mesh size of this lookup table is 3-by-11 and "Dat" means it is a data file.

Figure 4-1 suggests that the content of *WR281311.Dat* should be inspected by running *3D-Graf*. Simply typing "3D-GRAF WR281311.DAT" in response to DOS prompt will display an output similar to the one in Figure I3 on the computer screen. The menu to the left of the graph indicates the command keys that can be used to change the orientation as well as the size of the graph. The functions of these command keys are discussed in detail in Appendix C. Figure I3 depicts ϵ_{eff}(z-axis) as a function of frequency (x-axis) and gapwidth (y-axis). A big dot at the center of the graph indicates the position for which the frequency is 32.5 GHz, gapwidth is 1.928 mm, and $\epsilon_{eff} = 0.812$. These numbers are shown at the lower left corner of Figure I3. The number at the lower right corner of Figure I3 indicates that $Z_{eff} = 403.4\,\Omega$ at that frequency and gapwidth. To obtain a graph of Z_{eff} versus frequency and gapwidth, the "T" key can be used. Indeed the graph that appears on the screen is similar to the one in Figure 2A-2, but the difference is that the size of the graph in Figure 2A-2 is 11-by-11 rather than 3-by-11. Notice that the value of Z_{eff} is displayed at the lower left corner and ϵ_{eff} is displayed at the lower right corner. These two displayed graphs indicate that ϵ_{eff} and Z_{eff} are well behaved in the frequency and gapwidth range of interest; the lookup table can be used for CAD application.

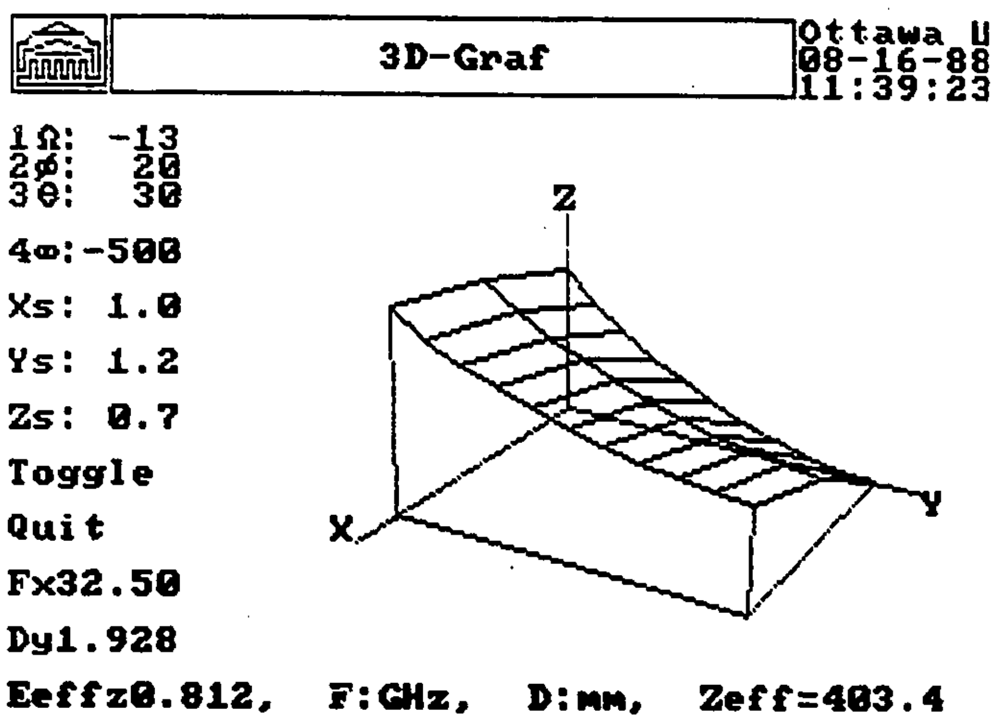
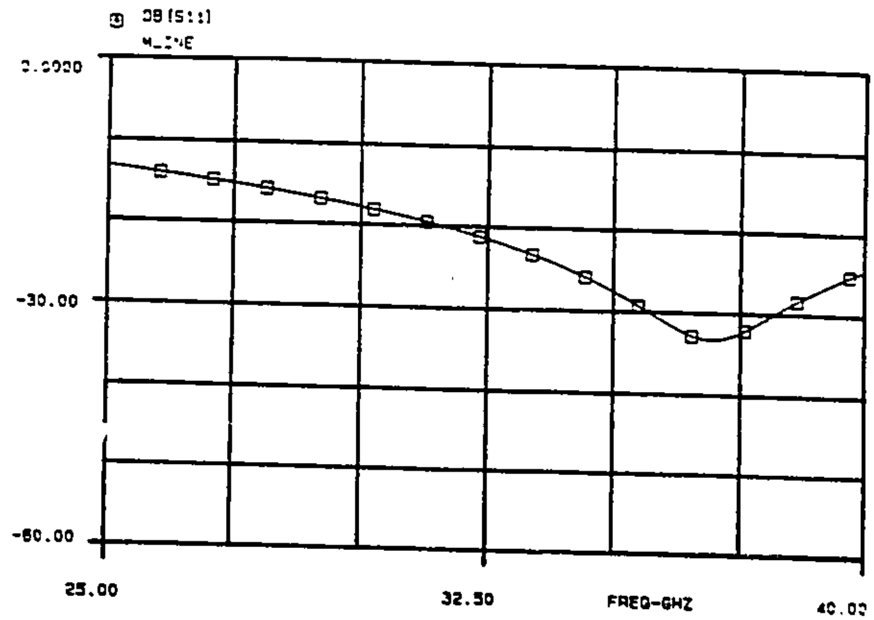


Figure I3: Graphical image (ϵ_{eff} versus f, and d) of *WR281311.Dat* displayed with *3D-Graf*.

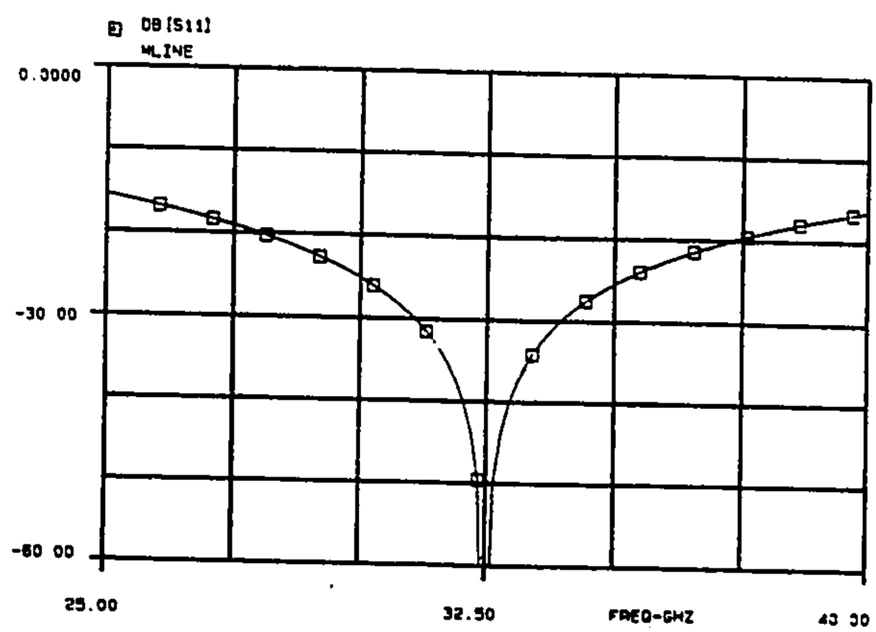
Figure 4-1 indicates that the text format lookup table must be converted to a binary file before it can be used by *TsiFin*, a user-defined version of Touchstone™ Senior. Simply type "MAKE-311 WR281311.DAT WR281A.311" in response to the DOS prompt. *WR281A.311* is the binary image of the original lookup table, "WR281" has the previous meaning while "A.311" is a special file name postfix for 3–by–11 binary format lookup tables used by *TsiFin*, see Table 3D-1. A 11–by–11 lookup table can be created similarly, if so desired, by choosing #_points: = 11 for both "f" and "d", however, this table should be renamed to *WR281.Dat*, and the compressed form should be named *WR281.Re4*. According to Figure 4-1, it is necessary to go to the Touchstone™ enviornment; simply enter "CD \EESOF". *TsiFin* can be invoked by entering "TSTONE". The binary format lookup tables that are in the \EEsof\Sr-Re4 directory are then loaded to the virtual disk automatically. The circuit file in Listing I1 describes the circuit in Figure I1.

```
!----------------------------------------------------------------------!
! File: Mline.CKT                                                      !
!                                                                      !
! This is an impedance matching example. The purpose of this example is !
! to demonstrate the calling convention of the TsiFin user defined elements. !
! Note that WR can be either positive or negative. When WR is positive, the !
! corresponding element is associated with an 11-by-11 lookup table and the !
! 11-by-11 interpolation scheme. When WR is negative, the corresponding !
! element is associated with a 3-by-11 lookup table and the 3-by-11 inter- !
! polation scheme.                                                     !
!                                                                      !
!                              Poman So                                !
!            Laboratory for Electromagnetics and Microwaves            !
!                       University of Ottawa                           !
!                          August 9, 1988.                             !
!                                                                      !
!----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281    !   281: WR281.Re4 — 11-by-11 lookup table
                 !  -281: WR281A.311 —  3-by-11 lookup table
  Flag = 0       !  Ensure that the sweep frequency is within the valid range
                 !  of the lookup table.
  RS = 200       !  source impedance
  RL = 400       !  load impedance

!------------------------------------------------------!
! Values to be optimized, initial values are 1 and 2.  !
!------------------------------------------------------!
  GAP1 # 0.5 0.94515 1.0
  LEN1 # 2.0 2.33151 2.5

CKT
!---------------------------------!
! Nodal description of the circuit !
!---------------------------------!
  U2PA 1 2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 2 MLINE
  RES 1 0 R^RS
  DEF1P 1 RSOURCE
  RES 1 0 R^RL
  DEF1P 1 RLOAD

TERM
  MLINE RSOURCE RLOAD

OUT
  MLINE DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1 -60 0 10

OPT
  RANGE 32 33
  MLINE MAG[S11] = 0
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing I1: A *TsiFin* circuit file that describes the circuit in Figure I1.



(a): Initial analysis



(b): Optimized result

Figure I4: Initial analysis and optimized result of the simple finline impedance transformer shown in Figure I1.

Listing I1 is self-explainatory for those who are familiar with circuit simulation programs. An initial analysis of this circuit can be performed easily by using the sweep command of Touchstone™. First select the output to be graph-1, GR1, by using the F4 key. Then use the F8 key to issue the frequency sweeping command. A graph similar to the one in Figure I4(a) appears on the screen . Because in the circuit file "`WR`" is set to "-281", a 3–by–11 lookup table (*WR281A.311*) and its corresponding interpolation method are used. To try out the 11–by–11 interpolation method simply set "`WR`" to "`281`" by using the tune key, F7. The graphs generated by these two different methods are the same.

The unknown values of d and l can be found be using the optimization capability of Touchstone™. If a gradient optimization method is used, then, in less than twenty trials, an optimized result similar to the one in Figure I4(b) will be obtained. The optimal values are $d = 0.945\,\mathrm{mm}$ and $l = 2.332\,\mathrm{mm}$.

Since this example is so simple it is not difficult to verify the result by the following procedure:

(1) Goto the \EEsof\Sr-Re4 directory.

(2) Enter "`3D-GRAF WR281.DAT`"‡.

(3) Set "`f`" to 32.5 GHz.

(4) Set "`d`" to 0.945 mm.

At that frequency and gapwidth $\epsilon_{eff} = 0.979$ and $Z_{eff} = 282.8\,\Omega$. Note that:

$$\frac{\lambda_g}{4} = \frac{c}{4f\sqrt{\epsilon_{eff}}} = 2.332\,\mathrm{mm} = l$$

$$Z_{eff} = 282.8\,\Omega = \sqrt{200 \times 400}\,\Omega$$

This is the well known quarter wave transformer result.

The above CAD procedure is also applicable if it is required to realize a similar transformer by using bilateral finline. Some minor changes which must be performed are:

(1) Use *BiFastS* to generate lookup tables.

(2) Name the 3–by–11 files *WR282311.Dat* and *WR282A.311* respectively.

(3) Name the 11–by–11 files *WR282.Dat* and *WR282.Re4* respectively.

(4) Set WR to -282 or 282 in the circuit file.

‡ It is required to used *WR281.DAT* rather then *WR281311.DAT* because *3D-Graf* can only perform linear interpolation.

APPENDIX J

Touchstone™ Senior S-Parameter Files and Circuit Files

Listing J1: Measurement data of Filter1A.

```
File: CRCF1.S2P

Measurement values are provided by Mr. C. Verver from the Communication
Research Centre in Ottawa.

Note that
      (1)  The phases of the S-parameters are arbitrary chosen to be 0.
      (2)  A maximum of 25 frequency points are allowed by Touch Stone.

                              Poman So
              Laboratory for Electromagnetics and Microwaves
                         University of Ottawa
                            August 9, 1988.

# GHz S DB R 50

 f            S11                  S21                 S12                 S22
32.0      -0.9333   0.0       -40.83   0.0       -40.83   0.0       -0.9333   0.0
32.5      -0.5444   0.0       -42.38   0.0       -42.38   0.0       -0.5444   0.0
33.0      -1.3220   0.0       -42.07   0.0       -42.07   0.0       -1.3220   0.0
33.5      -0.7778   0.0       -35.00   0.0       -35.00   0.0       -0.7778   0.0
34.0      -1.3220   0.0       -31.11   0.0       -31.11   0.0       -1.3220   0.0
34.5      -1.1670   0.0       -26.06   0.0       -26.06   0.0       -1.1670   0.0
35.0      -1.3220   0.0       -17.50   0.0       -17.50   0.0       -1.3220   0.0
35.5      -3.3440   0.0       -7.777   0.0       -7.777   0.0       -3.3440   0.0
35.9      -15.560   0.0       -3.111   0.0       -3.111   0.0       -15.560   0.0
36.0      -15.320   0.0       -1.711   0.0       -1.711   0.0       -15.320   0.0
36.1      -12.440   0.0       -2.178   0.0       -2.178   0.0       -12.440   0.0
36.2      -10.890   0.0       -2.772   0.0       -2.772   0.0       -10.890   0.0
36.3      -10.500   0.0       -2.567   0.0       -2.567   0.0       -10.500   0.0
36.5      -10.890   0.0       -2.100   0.0       -2.100   0.0       -10.890   0.0
36.6      -12.210   0.0       -2.022   0.0       -2.022   0.0       -12.210   0.0
36.7      -14.780   0.0       -2.256   0.0       -2.256   0.0       -14.780   0.0
36.8      -17.890   0.0       -1.944   0.0       -1.944   0.3       -17.890   0.0
36.9      -22.940   0.0       -1.322   0.0       -1.322   0.0       -22.940   0.0
37.0      -25.280   0.0       -1.566   0.0       -1.566   0.0       -25.280   0.0
37.1      -22.940   0.0       -1.944   0.0       -1.944   0.0       -22.940   0.0
37.2      -17.890   0.0       -1.556   0.0       -1.556   0.0       -17.890   0.0
37.3      -15.940   0.0       -2.022   0.0       -2.022   0.0       -15.940   0.0
37.4      -14.540   0.0       -1.867   0.0       -1.867   0.0       -14.540   0.0
37.5      -13.840   0.0       -2.333   0.0       -2.333   0.0       -13.840   0.0
37.6      -14.540   0.0       -2.644   0.0       -2.644   0.0       -14.540   0.0
37.8      -16.330   0.0       -2.333   0.0       -2.333   0.0       -16.330   0.0
37.9      -21.000   0.0       -2.333   0.0       -2.333   0.0       -21.000   0.0
38.0      -18.670   0.0       -2.488   0.0       -2.488   0.0       -18.670   0.0
38.1      -11.670   0.0       -2.022   0.0       -2.022   0.0       -11.670   0.0
38.5      -2.8780   0.0       -6.067   0.0       -6.067   0.0       -2.8780   0.0
39.0      -0.9333   0.0       -14.31   0.0       -14.31   0.0       -0.9333   0.0
39.5      -0.7778   0.0       -19.83   0.0       -19.83   0.0       -0.7778   0.0
40.0      -1.5560   0.0       -25.82   0.0       -25.82   0.0       -1.5560   0.0
```

Listing J2: Measurement data of Filter2A.

```
File: CRCF2.S2P

Measurement values are provided by Mr. C. Verver from the Communication
Research Centre in Ottawa.

Note that
      (1)   The phases of the S-parameters are arbitrary chosen to be 0.
      (2)   A maximum of 25 frequency points are allowed by Touch Stone.

                              Roman So
              Laboratory for Electromagnetics and Microwaves
                         University of Ottawa
                           August 9, 1988.

# GHz S DB R 50
```

f	S11		S21		S12		S22	
24.0	-0.75	0.0	-42.0	0.0	-42.0	0.0	-0.75	0.0
24.5	-0.50	0.0	-35.5	0.0	-35.5	0.0	-0.50	0.0
25.0	-1.00	0.0	-32.0	0.0	-32.0	0.0	-1.00	0.0
25.5	-1.00	0.0	-26.5	0.0	-26.5	0.0	-1.00	0.0
26.0	-1.25	0.0	-21.0	0.0	-21.0	0.0	-1.25	0.0
26.5	-2.00	0.0	-14.0	0.0	-14.0	0.0	-2.00	0.0
27.0	-7.25	0.0	-3.00	0.0	-3.00	0.0	-7.25	0.0
27.5	-12.5	0.0	-3.25	0.0	-3.25	0.0	-12.5	0.0
27.8	-22.5	0.0	-1.50	0.0	-1.50	0.0	-22.5	0.0
28.0	-20.0	0.0	-1.50	0.0	-1.50	0.0	-20.0	0.0
28.3	-14.0	0.0	-1.50	0.0	-1.50	0.0	-14.0	0.0
28.5	-17.5	0.0	-1.50	0.0	-1.50	0.0	-17.5	0.0
28.65	-22.5	0.0	-1.50	0.0	-1.50	0.0	-22.5	0.0
28.8	-8.00	0.0	-1.50	0.0	-1.50	0.0	-8.00	0.0
29.0	-5.00	0.0	-6.50	0.0	-6.50	0.0	-5.00	0.0
29.5	-1.25	0.0	-10.5	0.0	-10.5	0.0	-1.25	0.0
30.0	-0.40	0.0	-16.0	0.0	-16.0	0.0	-0.40	0.0
30.5	-0.40	0.0	-21.0	0.0	-21.0	0.0	-0.40	0.0
31.0	-0.40	0.0	-24.5	0.0	-24.5	0.0	-0.40	0.0
31.5	-0.40	0.0	-28.5	0.0	-28.5	0.0	-0.40	0.0
32.0	-0.30	0.0	-30.5	0.0	-30.5	0.0	-0.30	0.0
32.5	-0.40	0.0	-33.0	0.0	-33.0	0.0	-0.40	0.0
33.0	-0.40	0.0	-34.5	0.0	-34.5	0.0	-0.40	0.0
33.5	-0.30	0.0	-37.0	0.0	-37.0	0.0	-0.30	0.0
34.0	-0.30	0.0	-36.0	0.0	-36.0	0.0	-0.30	0.0
34.5	-0.40	0.0	-35.0	0.0	-35.0	0.0	-0.40	0.0
35.0	-0.50	0.0	-41.0	0.0	-41.0	0.0	-0.50	0.0

Listing J3: Circuit file for MLine in Appendix I.

```
!--------------------------------------------------------------------------!
! File: Mline.CKT                                                          !
!                                                                          !
! This is an impedance matching example. The purpose of this example is   !
! to demonstrate the calling convention of the TsiFin user defined elements.!
! Note that WR can be either positive or negative. When WR is positive, the!
! corresponding element is associated with an 11-by-11 lookup table and the!
! 11-by-11 interpolation scheme. When WR is negative, the corresponding    !
! element is associated with a 3-by-11 lookup table and the 3-by-11 inter- !
! polation scheme.                                                         !
!                                                                          !
!                               Poman So                                   !
!          Laboratory for Electromagnetics and Microwaves                  !
!                         University of Ottawa                             !
!                          August 9, 1988.                                 !
!                                                                          !
!--------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR              !  281:  WR281.Re4 — 11-by-11 lookup table
  WR    = -281   ! -281: WR281A.311 —  3-by-11 lookup table
  Flag = 0       !  Ensure that the sweep frequency is within the valid range
                 !  of the lookup table.
  RS = 200       !  source impedance
  RL = 400       !  load impedance

!-------------------------------------------------!
! Values to be optimized, initial values are 1 and 2. !
!-------------------------------------------------!
  GAP1 # 0.5 0.94515 1.0
  LEN1 # 2.0 2.33151 2.5

CKT
!-------------------------------!
! Nodal description of the circuit !
!-------------------------------!
  U2PA 1 2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 2 MLINE
  RES 1 0 R^RS
  DEF1P 1 RSOURCE
  RES 1 0 R^RL
  DEF1P 1 RLOAD

TERM
  MLINE RSOURCE RLOAD

OUT
  MLINE DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE  25 40 2.5
  GR1 -60 0 10

OPT
  RANGE 32 33
  MLINE MAG[S11] = 0
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J4: Circuit file for a two-step transformer which is implemented by using ideal transmission lines.

```
!--------------------------------------------------------------------------!
! File: TlinTran.CKT                                                       !
!                                                                          !
! Two-step quarter wave transformer implemented by using ideal transmission !
! elements.                                                                !
!                                                                          !
! Note that the initial values (Z1=237.6, Z2=329.2) are calculated by      !
! following an example given in section 5.10 of R.E. Collin, "Foundations  !
! for Microwave Engineering", MaGraw-Hill, 1966.                           !
!                                                                          !
!                              Poman So                                    !
!           Laboratory for Electromagnetics and Microwaves                 !
!                       University of Ottawa                               !
!                         August 9, 1988.                                  !
!--------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  FO = 32.5   ! 32.5 GHz
  EO = 90     ! Quater-Wave at FO
  RS = 200    ! SOURCE IMPEDANCE
  RL = 400    ! LODA IMPEDANCE

! Initial values are 237.6 and 329.2

  Z1 # 200  239.00130  400
  Z2 # 200  334.58960  400

CKT
  TLIN 1 2 Z^Z1 E^EO F^FO
  TLIN 2 3 Z^Z2 E^EO F^FO
  DEF2P 1 3 TLINTRAN
  RES 1 0 R^RS
  DEF1P 1 RSOURCE
  RES 1 0 R^RL
  DEF1P 1 RLOAD

TERM
  TLINTRAN RSOURCE RLOAD

OUT
  TLINTRAN DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE  25 40 2.5
  GR1   -60 0 10

OPT
  RANGE 27.5 37.5
  TLINTRAN  DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J5: Circuit file for a two-step transformer which is implemented by using model U2PA, Section A, Chapter 5.

```
!------------------------------------------------------------------------!
! File: U2PATran.CKT                                                     !
!                                                                        !
! Step transforer implemented by using U2PA elements and lookup table   !
! WR281.Re4 or WR281A.311. The initial gapwidths are obtained by running !
! 3D-Graf at frequenct at 32.5 GHz and iteratively try out various gapwidths !
! until the impedances are equal to that obtained from TlinTran.CKT.  The !
! initial lengths are computed by using the corresponding effective     !
! dielectric constants from 3D-Graf.                                     !
!                                                                        !
!                              Poman So                                  !
!           Laboratory for Electromagnetics and Microwaves               !
!                        University of Ottawa                            !
!                          August 9, 1988.                               !
!------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = 281
  Flag = 1
  RS = 200
  RL = 400

! Values from TLINTRAN

! GAP1 = 0.6505
! GAP2 = 1.3352
! LEN1 = 2.2500
! LEN2 = 2.4270

! Optimized values

  GAP1 # 0.5 0.66064 1.0
  GAP2 # 0.5 1.35065 2.0
  LEN1 # 2.0 2.35175 2.5
  LEN2 # 2.0 2.51419 3.0

CKT
  U2PA 1 2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA 2 3 D1^GAP2 D2^LEN2 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 U2PATRAN
  RES 1 0 R^RS
  DEF1P 1 RSOURCE
  RES 1 0 R^RL
  DEF1P 1 RLOAD

TERM
  U2PATRAN RSOURCE RLOAD

OUT
  U2PATRAN DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE  25 40 2.5
  GR1   -60 0 10

OPT
  RANGE 29.1  35.95
  U2PATRAN DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J6: Circuit file for a two-step transformer which is implemented by using model U2PC, Section A, Chapter 5.

```
!--------------------------------------------------------------------------!
! File: U2PCTran.CKT                                                       !
!                                                                          !
! Step transforer implemented by using U2PA elements and lookup table      !
! WR281.Re4 or WR281A.311. The initial gapwidths are obtained by running   !
! 3D-Graf at frequenct at 32.5 GHz and iteratively try out various gapwidths !
! until the impedances are equal to that obtained from TlinTran.CKT.  The  !
! initial lengths are computed by using the corresponding effective        !
! dielectric constants from 3D-Graf.                                       !
!                                                                          !
!                                                                          !
!                              Poman So                                    !
!             Laboratory for Electromagnetics and Microwaves               !
!                        University of Ottawa                              !
!                          August 9, 1988.                                 !
!                                                                          !
!--------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = 281
  Flag = 1
  RS = 200
  RL = 400

! Values from TLINTRAN

! GAP1 = 0.6505
! GAP2 = 1.3352
! LEN1 = 2.2500
! LEN2 = 2.4270

! Optimized values

  GAP1 # 0.5 0.66066 1.0
  GAP2 # 0.5 1.34926 2.0
  LEN1 # 2.0 2.35570 2.5
  LEN2 # 2.0 2.50343 3.0

CKT
  U2PC 1 2 D1^GAP1 D2^LEN1 D3^GAP2 D4^LEN2 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 2 U2PCTRAN
  RES 1 0 R^RS
  DEF1P 1 RSOURCE
  RES 1 0 R^RL
  DEF1P 1 RLOAD

TERM
  U2PCTRAN RSOURCE RLOAD

OUT
  U2PCTRAN DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE  25 40 2.5
  GR1  -60 0 10

OPT
  RANGE 29.1  35.95
  U2PCTRAN DB[S11] < -40
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J7: Circuit file for a taper which is approximated by using model U2PA, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
! File: U2PATp2X.CKT                                                    !
!                                                                       !
! A 2-Step taper implemented by using U2PA elements and lookup table   !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                             Poman So                                  !
!           Laboratory for Electromagnetics and Microwaves              !
!                       University of Ottawa                            !
!                         August 9, 1988.                               !
!-----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1


  G1 # 0.1 0.64039 1.0
  G2 # 0.1 0.91934 1.0
  G3 # 0.1 0.50696 1.0
  L1 # 0.1 0.28171 1.0
  L2 # 0.1 0.91522 1.0

EQN
  GT = 1.7/(G1+G2+G3)
  LT = 10/(L1+L2)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  LEN1 = LT*L1
  LEN2 = LT*L2

CKT
  U2PA  1  2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  2  3 D1^GAP2 D2^LEN2 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 U2PATP2X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PATP2X RSOURCE RLOAD

OUT
  U2PATP2X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PATP2X DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J8: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
! File: U2PCTp2X.CKT                                                    !
!                                                                       !
! A 2-Step taper implemented by using U2PC elements and lookup table    !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                          Poman So                                     !
!          Laboratory for Electromagnetics and Microwaves               !
!                   University of Ottawa                                !
!                     August 9, 1988.                                   !
!                                                                       !
!-----------------------------------------------------------------------!
DIM
  FREQ GH2
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.64039 1.0
  G2 # 0.1 0.91934 1.0
  G3 # 0.1 0.50696 1.0
  L1 # 0.1 0.28171 1.0
  L2 # 0.1 0.91522 1.0

EQN
  GT = 1.7/(G1+G2+G3)
  LT = 10/(L1+L2)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  LEN1 = LT*L1
  LEN2 = LT*L2

CKT
  U2PC  1 2 D1^GAP1 D2^LEN1 D3^GAP2 D4^LEN2 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 2 U2PCTP2X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP2X RSOURCE RLOAD

OUT
  U2PCTP2X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP2X DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J9: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
! File: U2PATp2X.opt                                                    !
!                                                                       !
! A 2-Step taper implemented by using U2PC elements and lookup table    !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                              Poman So                                 !
!          Laboratory for Electromagnetics and Microwaves               !
!                     University of Ottawa                              !
!                        August 9, 1988.                                !
!                                                                       !
!-----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.64009 1.0
  G2 # 0.1 0.91941 1.0
  G3 # 0.1 0.50696 1.0
  L1 # 0.1 0.28343 1.0
  L2 # 0.1 0.91496 1.0

EQN
  GT = 1.7/(G1+G2+G3)
  LT = 10/(L1+L2)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  LEN1 = LT*L1
  LEN2 = LT*L2

CKT
  U2PC  1 2 D1^GAP1 D2^LEN1 D3^GAP2 D4^LEN2 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 2 U2PCTP2X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP2X RSOURCE RLOAD

OUT
  U2PCTP2X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP2X DB[S11] < -40
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J10: Circuit file for a taper which is approximated by using model U2PA, Section B, Chapter 5.

```
!---------------------------------------------------------------------------!
! File: U2PATp3X.CKT                                                        !
!                                                                           !
! A 3-Step taper implemented by using U2PA elements and lookup table        !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative!
! quantities. The actural dimensions of the gapwidth and length of each     !
! section is calculated in the EQN block.                                   !
!                                                                           !
!                           Poman So                                        !
!          Laboratory for Electromagnetics and Microwaves                   !
!                    University of Ottawa                                   !
!                      August 9, 1988.                                      !
!                                                                           !
!---------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1


  G1 # 0.1 0.48948 1.0
  G2 # 0.1 0.98024 1.0
  G3 # 0.1 0.97178 1.0
  G4 # 0.1 0.10006 1.0
  L1 # 0.1 0.36437 1.0
  L2 # 0.1 0.38302 1.0
  L3 # 0.1 0.92048 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4)
  LT = 10/(L1+L2+L3)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3

CKT
  U2PA  1  2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  2  3 D1^GAP2 D2^LEN2 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  3  4 D1^GAP3 D2^LEN3 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 4 U2PATP3X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PATP3X RSOURCE RLOAD

OUT
  U2PATP3X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PATP3X DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J11: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!------------------------------------------------------------------------!
! File: U2PCTp3X.CKT                                                     !
!                                                                        !
! A 3-Step taper implemented by using U2PC elements and lookup table     !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each  !
! section is calculated in the EQN block.                                !
!                                                                        !
!                              Poman So                                  !
!             Laboratory for Electromagnetics and Microwaves             !
!                        University of Ottawa                            !
!                          August 9, 1988.                               !
!------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.48948 1.0
  G2 # 0.1 0.98024 1.0
  G3 # 0.1 0.97178 1.0
  G4 # 0.1 0.10006 1.0
  L1 # 0.1 0.36437 1.0
  L2 # 0.1 0.38302 1.0
  L3 # 0.1 0.92048 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4)
  LT = 10/(L1+L2+L3)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3

CKT
  U2PC  1 2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0    D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2 3 D1^GAP2 D2^LEN2 D3^GAP3 D4^LEN3 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 U2PCTP3X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP3X RSOURCE RLOAD

OUT
  U2PCTP3X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP3X DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J12: Circuit file for a taper which is approximated by using model U2PC. Section B, Chapter 5.

```
!----------------------------------------------------------------------!
! File: U2PCTp3X.OPT                                                   !
!                                                                      !
! A 3-Step taper implemented by using U2PC elements and lookup table   !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                              !
!                                                                      !
!                              Poman So                                !
!              Laboratory for Electromagnetics and Microwaves          !
!                        University of Ottawa                          !
!                          August 9, 1988.                             !
!----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.48916 1.0
  G2 # 0.1 0.98121 1.0
  G3 # 0.1 0.97234 1.0
  G4 # 0.1 0.10002 1.0
  L1 # 0.1 0.36597 1.0
  L2 # 0.1 0.37925 1.0
  L3 # 0.1 0.91841 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4)
  LT = 10/(L1+L2+L3)
  GAP1 = 0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3

CKT
  U2PC  1 2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0     D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2 3 D1^GAP2 D2^LEN2 D3^GAP3 D4^LEN3 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 U2PCTP3X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP3X RSOURCE RLOAD

OUT
  U2PCTP3X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP3X DB[S11] < -40
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J13: Circuit file for a taper which is approximated by using model U2PA, Section B, Chapter 5.

```
!------------------------------------------------------------------------!
! File: U2PATp4X.CKT                                                     !
!                                                                        !
! A 4-Step taper implemented by using U2PA elements and lookup table     !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each  !
! section is calculated in the EQN block.                                !
!                                                                        !
!                              Poman So                                  !
!          Laboratory for Electromagnetics and Microwaves                !
!                        University of Ottawa                            !
!                          August 9, 1988.                               !
!                                                                        !
!------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1


  G1 # 0.1 0.37507 1.0
  G2 # 0.1 0.66298 1.0
  G3 # 0.1 0.98681 1.0
  G4 # 0.1 0.44314 1.0
  G5 # 0.1 0.33562 1.0
  L1 # 0.1 0.65500 1.0
  L2 # 0.1 0.69258 1.0
  L3 # 0.1 0.80353 1.0
  L4 # 0.1 0.93525 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4+G5)
  LT = 10/(L1+L2+L3+L4)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4

CKT
  U2PA  1  2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  2  3 D1^GAP2 D2^LEN2 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  3  4 D1^GAP3 D2^LEN3 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  4  5 D1^GAP4 D2^LEN4 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 5 U2PATP4X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PATP4X RSOURCE RLOAD

OUT
  U2PATP4X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PATP4X DB[S11] < -40.1
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J14: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!--------------------------------------------------------------------------!
! File: U2PCTp4X.CKT                                                       !
!                                                                          !
! A 4-Step taper implemented by using U2PC elements and lookup table       !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each    !
! section is calculated in the EQN block.                                  !
!                                                                          !
!                              Poman So                                    !
!          Laboratory for Electromagnetics and Microwaves                  !
!                       University of Ottawa                               !
!                          August 9, 1988.                                 !
!--------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1


  G1 # 0.1 0.37507 1.0
  G2 # 0.1 0.66298 1.0
  G3 # 0.1 0.98681 1.0
  G4 # 0.1 0.44314 1.0
  G5 # 0.1 0.33562 1.0
  L1 # 0.1 0.65500 1.0
  L2 # 0.1 0.69258 1.0
  L3 # 0.1 0.80353 1.0
  L4 # 0.1 0.93525 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4+G5)
  LT = 10/(L1+L2+L3+L4)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4

CKT
  U2PC  1  2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0     D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2  3 D1^GAP2 D2^LEN2 D3^GAP3 D4=0     D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  3  4 D1^GAP3 D2^LEN3 D3^GAP4 D4^LEN4 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 4 U2PCTP4X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP4X RSOURCE RLOAD

OUT
  U2PCTP4X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP4X DB[S11] < -40.1
!-------------------!
! End of circuit file !
!-------------------!
```

Listing J15: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
!                                                                       !
! File: U2PCTp4X.OPT                                                    !
!                                                                       !
! A 4-Step taper implemented by using U2PC elements and lookup table    !
! WR281.R44 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                              Poman So                                 !
!          Laboratory for Electromagnetics and Microwaves              !
!                        University of Ottawa                           !
!                          August 9, 1988.                              !
!                                                                       !
!-----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.37552 1.0
  G2 # 0.1 0.66125 1.0
  G3 # 0.1 0.98689 1.0
  G4 # 0.1 0.44193 1.0
  G5 # 0.1 0.33602 1.0
  L1 # 0.1 0.65828 1.0
  L2 # 0.1 0.68930 1.0
  L3 # 0.1 0.80373 1.0
  L4 # 0.1 0.93543 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4+G5)
  LT = 10/(L1+L2+L3+L4)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4

CKT
  U2PC  1  2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2  3 D1^GAP2 D2^LEN2 D3^GAP3 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  3  4 D1^GAP3 D2^LEN3 D3^GAP4 D4^LEN4 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 4 U2PCTP4X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP4X RSOURCE RLOAD

OUT
  U2PCTP4X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -40  0  10

OPT
  RANGE 26.5 40
  U2PCTP4X DB[S11] < -40.1
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

Listing J16: Circuit file for a taper which is approximated by using model U2PA, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
! File: U2PATp5X.CKT                                                    !
!                                                                       !
! A 5-Step taper implemented by using U2PA elements and lookup table    !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                              Poman So                                 !
!          Laboratory for Electromagnetics and Microwaves               !
!                       University of Ottawa                            !
!                         August 9, 1988.                               !
!-----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.47257 1.0
  G2 # 0.1 0.74278 1.0
  G3 # 0.1 0.91449 1.0
  G4 # 0.1 0.70962 1.0
  G5 # 0.1 0.55655 1.0
  G6 # 0.1 0.43857 1.0
  L1 # 0.1 0.63106 1.0
  L2 # 0.1 0.59475 1.0
  L3 # 0.1 0.39632 1.0
  L4 # 0.1 0.64189 1.0
  L5 # 0.1 0.96687 1.0
EQN
  GT = 1.7/(G1+G2+G3+G4+G5+G6)
  LT = 10/(L1+L2+L3+L4+L5)
  GAP1 =   0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  GAP5 = GAP4 + GT*G5
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4
  LEN5 = LT*L5

CKT
  U2PA  1  2 D1^GAP1 D2^LEN1 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  2  3 D1^GAP2 D2^LEN2 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  3  4 D1^GAP3 D2^LEN3 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  4  5 D1^GAP4 D2^LEN4 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PA  5  6 D1^GAP5 D2^LEN5 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 6 U2PATP5X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PATP5X RSOURCE RLOAD

OUT
  U2PATP5X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PATP5X DB[S11] < -41
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J17: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!--------------------------------------------------------------------!
! File: U2PCTp5X.CKT                                                 !
!                                                                    !
! A 5-Step taper implemented by using U2PC elements and lookup table !
! WR281.Re4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                            !
!                                                                    !
!                            Poman So                                !
!           Laboratory for Electromagnetics and Microwaves           !
!                      University of Ottawa                          !
!                        August 9, 1988.                             !
!                                                                    !
!--------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 # 0.1 0.47257 1.0
  G2 # 0.1 0.74278 1.0
  G3 # 0.1 0.91449 1.0
  G4 # 0.1 0.70962 1.0
  G5 # 0.1 0.55655 1.0
  G6 # 0.1 0.43857 1.0
  L1 # 0.1 0.63106 1.0
  L2 # 0.1 0.59475 1.0
  L3 # 0.1 0.39632 1.0
  L4 # 0.1 0.64189 1.0
  L5 # 0.1 0.96687 1.0

EQN
  GT = 1.7/(G1+G2+G3+G4+G5+G6)
  LT = 10/(L1+L2+L3+L4+L5)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  GAP5 = GAP4 + GT*G5
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4
  LEN5 = LT*L5

CKT
  U2PC  1  2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2  3 D1^GAP2 D2^LEN2 D3^GAP3 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  3  4 D1^GAP3 D2^LEN3 D3^GAP4 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  4  5 D1^GAP4 D2^LEN4 D3^GAP5 D4^LEN5 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 5 U2PCTP5X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP5X RSOURCE RLOAD

OUT
  U2PCTP5X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1  -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP5X DB[S11] < -41
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J18: Circuit file for a taper which is approximated by using model U2PC, Section B, Chapter 5.

```
!-----------------------------------------------------------------------!
! File: U2PCTp5X.OPT                                                    !
!                                                                       !
! A 5-Step taper implemented by using U2PC elements and lookup table    !
! WR281.Ra4 or WR281A.311. Note that the optimization variables are relative !
! quantities. The actural dimensions of the gapwidth and length of each !
! section is calculated in the EQN block.                               !
!                                                                       !
!                            Poman So                                   !
!           Laboratory for Electromagnetics and Microwaves              !
!                      University of Ottawa                             !
!                        August 9, 1988.                                !
!                                                                       !
!-----------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  WR   = -281
  Flag = 1

  G1 #  0.1 0.47231 1.0
  G2 #  0.1 0.74312 1.0
  G3 #  0.1 0.91463 1.0
  G4 #  0.1 0.70917 1.0
  G5 #  0.1 0.55654 1.0
  G6 #  0.1 0.43993 1.0
  L1 #  0.1 0.63566 1.0
  L2 #  0.1 0.59236 1.0
  L3 #  0.1 0.39567 1.0
  L4 #  0.1 0.64116 1.0
  L5 #  0.1 0.96755 1.0

EQN
  GT  = 1.7/(G1+G2+G3+G4+G5+G6)
  LT  = 10/(L1+L2+L3+L4+L5)
  GAP1 =  0.3 + GT*G1
  GAP2 = GAP1 + GT*G2
  GAP3 = GAP2 + GT*G3
  GAP4 = GAP3 + GT*G4
  GAP5 = GAP4 + GT*G5
  LEN1 = LT*L1
  LEN2 = LT*L2
  LEN3 = LT*L3
  LEN4 = LT*L4
  LEN5 = LT*L5

CKT
  U2PC  1  2 D1^GAP1 D2^LEN1 D3^GAP2 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  2  3 D1^GAP2 D2^LEN2 D3^GAP3 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  3  4 D1^GAP3 D2^LEN3 D3^GAP4 D4=0      D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PC  4  5 D1^GAP4 D2^LEN4 D3^GAP5 D4^LEN5 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 5 U2PCTP5X
  RES 1 0 R=200
  DEF1P 1 RSOURCE
  RES 1 0 R=400
  DEF1P 1 RLOAD

TERM
  U2PCTP5X RSOURCE RLOAD

OUT
  U2PCTP5X DB[S11] GR1

FREQ
  SWEEP 25 40 0.15

GRID
  RANGE 25 40 2.5
  GR1   -60  0  10

OPT
  RANGE 26.5 40
  U2PCTP5X DB[S11] < -41
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J19: Circuit file for a bandpass filter which is implemented by using model U2PB and U1PA, Section C, Chapter 5.

```
!--------------------------------------------------------------------------!
! File: Filter1A.CKT                                                       !
!                                                                          !
! A bandpass filter implemented by using 4-U2PC elements and lookup table  !
! WR221.Re4 or WR221A.311. Note that the dimensions are provided by        !
! Mr. C. Verver from the Communication Research Centre in Ottawa.          !
!                                                                          !
!                              Poman So                                    !
!            Laboratory for Electromagnetics and Microwaves                !
!                        University of Ottawa                              !
!                          August 9, 1988.                                 !
!                                                                          !
!--------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  Flag = 1
!
! CRC values
!
  WR   = -221
  W1   = 0.1016
  W2   = 0.8890
  GAP  = 0.3810
  L1   = 1.6637
  L2   = 1.7399
  GAPM = 0.381      ! 2.844

CKT
!
! One half of a filter
!
  U2PB  1 2 D1^GAP D2^L1 D3^W1 D4^L1 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PB  2 3 D1^GAP D2^L2 D3^W2 D4^L2 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 HFILTER
!
! A complete filter
!
  HFILTER 1 2
  HFILTER 3 2
  DEF2P   1 3 FILTER1A
!
! Termination impedance
!
  U1PA 1 D1^GAPM D2=0 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10=0
  DEF1P 1 RSOURCE
!
! Measurement done in CRC
!
  S2PA 1 2 0 CRCF1
  DEF2P 1 2 CRCF1S2P

TERM
  FILTER1A RSOURCE RSOURCE

OUT
  FILTER1A DB[S11] GR1
  FILTER1A DB[S21] GR1A
  FILTER1A DB[S11] GR2
  CRCF1S2P DB[S11] GR2
  FILTER1A DB[S21] GR3
  CRCF1S2P DB[S21] GR3

FREQ
  SWEEP 32 40 0.08

GRID
  RANGE  32 40 2
  GR1   -30 0 10
  GR1A  -60 0 20
  GR2   -30 0 10
  GR3   -60 0 20

OPT
  RANGE 36.0 38.0
  FILTER1A DB[S11] < -12
  FILTER1A DB[S21] > -2
!-------------------!
!                   !
! End of circuit file !
!                   !
!-------------------!
```

Listing J20: Circuit file for a bandpass filter which is implemented by using model U2PB and U1PA, Section C, Chapter 5.

```
!-------------------------------------------------------------------------!
!                                                                         !
! File: Filter2A.CKT                                                      !
!                                                                         !
! A bandpass filter implemented by using 4-U2PC elements and lookup table !
! WR281.Re4 or WR281A.311. Note that the dimensions are provided by       !
! Mr. C. Verver from the Communication Research Centre in Ottawa.         !
!                                                                         !
!                              Poman So                                   !
!            Laboratory for Electromagnetics and Microwaves               !
!                         University of Ottawa                            !
!                           August 9, 1988.                               !
!                                                                         !
!-------------------------------------------------------------------------!
DIM
  FREQ GHZ
  RES OH
  IND NH
  CAP PF
  LNG MM
  TIME PS
  COND /OH
  ANG DEG

VAR
  Flag = 1

!----------------------!
! Experimental values  !
!----------------------!
  WR   = -281
  GAP  = 0.50800
  W1   = 0.03048
  W2   = 0.79502
  L1   = 2.11074
  L2   = 2.27330
  GAPM = 0.508

CKT
!----------------------!
! One half of a filter !
!----------------------!
  U2PB  1 2 D1^GAP D2^L1 D3^W1 D4^L1 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  U2PB  2 3 D1^GAP D2^L2 D3^W2 D4^L2 D5=0 D6=0 D7=0 D8=0 D9^WR D10^Flag
  DEF2P 1 3 HFILTER

!-------------------!
! A complete filter !
!-------------------!
  HFILTER 1 2
  HFILTER 3 2
  DEF2P   1 3 FILTER2A

!-----------------------!
! Termination impedance !
!-----------------------!
  U1PA 1 D1^GAPM D2=0 D3=0 D4=0 D5=0 D6=0 D7=0 D8=0 D9^WR D10=0
  DEF1P 1 RSOURCE

!--------------------!
! Measurement result !
!--------------------!
  S2PA 1 2 0 CRCF2
  DEF2P 1 2 CRCF2S2P

TERM
  FILTER2A RSOURCE RSOURCE

OUT
  FILTER2A DB[S11] GR1
  FILTER2A DB[S21] GR1A
  FILTER2A DB[S11] GR2
  CRCF2S2P DB[S11] GR2
  FILTER2A DB[S21] GR3
  CRCF2S2P DB[S21] GR3

FREQ
  SWEEP 25 35 0.1

GRID
  RANGE  25 35 2
  GR1  -30 0 10
  GR1A -60 0 20
  GR2  -30 0 10
  GR3  -60 0 20

OPT
  RANGE 28 30
  FILTER2A DB[S21] = 0
  FILTER2A DB[S11] < -15
!--------------------!
!                    !
! End of circuit file !
!                    !
!--------------------!
```

APPENDIX K

System Setup Files — AutoExec.Bat, Config.Sys and Tstone.Bat

This appendix contains three system setup files. *AutoExec.Bat* and *Config.Sys* must be in the system at boot-time inorder so that the CAD environment can be setup properly. *Tstone.Bat* is needed to be in the system at run-time. The contents of these files are shown in Listing K1 to K3.

AutoExec.Bat load the extended graphic characters into the system so that Ω, Φ and Θ can be displayed in the control menu of *3D-Graf*. It also loads a resident program, *Graphics*, into the system so that the graphic image on the screen can be dumped to the printer. This file does not have to be in the system at run-time.

Config.Sys setup a virtual disk to be used by *TsiFin* to store lookup tables. This file does not have to be in the system at run-time.

Tstone.Bat copy all the binary format lookup tables form the EEsof\Sr-Re4 directory to the virtual disk.

Listing K1: Autoexec.Bat

```
rem ********************************
rem ** Some system setup commands **
rem ********************************
Prompt $P$G
Path C:MsDos
rem *******************************************************
rem ** Execute GRAPHICS and load GRAFTABL; setup system for **
rem ** for graphic programs such as 3D-Graf and UniFastS.  **
rem *******************************************************
Graphics
GrafTable
rem ***************************************************
rem ** Display Ottawa U. logo and Lab for EE messages. **
rem ***************************************************
Hello
rem **************************
rem ** END of AutoExec.Bat **
rem **************************
```

Listing K2: Config.Sys

```
rem ********************************
rem ** Some system setup commands **
rem ********************************
Buffers = 10
Files   = 20
Device  = C:MsDos\HarDrive.Sys
rem ***************************************************************
rem ** Setup a 380K ram drive ( or virtual disk ) in the extended **
rem ** memory. TsiFin needs this ram drive to store lookup tables.**
rem ***************************************************************
Device = C:MsDos\RamDrive.Sys 380 512 64 /E
rem ************************
rem ** END of Config.Sys **
rem ************************
```

Listing K1: Tstone.Bat

```
rem *******************************************************
rem ** Copy all binary format lookup tables from Sr-Re4 to **
rem ** the drive E, a ram drive.                          **
rem **                                                    **
rem ** This file must be in the EEsof directory and Sr-Re4 **
rem ** must be the subdirectory of EEsof.                 **
rem *******************************************************
Copy Sr-Re4.Re4 E:
Copy Sr-Re4.311 E:
TsiFin
rem *************************
rem ** END of Tstone.Bat **
rem *************************
```

References

[1] P. Bhartia and I. J. Bahl, *Millimeter Wave Engineering and Applications*, Chapter 6, 1984, John Wiley & Sons.

[2] R. H. Jansen, "A Novel CAD Tool and Concept Compatible with the Requirements of Multi-Layer GaAs MMIC Technology", in 1985 IEEE MTT–S International Microwave Symposium Digest, pp. 711–714, St. Louis, June 4–6, 1985.

[3] P. Saguet and A. Coumes, "Filtres Hyperfrequences en Technologie Finline".

[4] Y. L. Tsui and W. J. R. Hoefer, "Empirical Formulae for the Parameters of Impedance Steps and Inductive Strips in Finlines", in IEEE Montech'86 Symposium Digest, pp. 36–38, Sept. 29 to Oct. 3, 1986, Montreal.

[5] W. J. R. Hoefer, "Accelerated Spectral Domain Analysis of E–Plane Circuits Suitable for Computer–Aided Design", in URSI International Symposium on Electromagnetic Theory, pp. 485–495, Budapest, August 25–29, 1986.

[6] P. J. Meier, "Integrated Fin-Line Millimeter Components", in IEEE Trans. MTT–22, No. 12, pp. 1209–1216, Dec. 1974.

[7] K. C. Gupta, Ramesh Garg and Rakesh Chadha, *Computer–Aided Design of Microwave Circuits*, Chapter 2, 1981, Artech House Inc..

[8] R. E. Collin, *Foundations for Microwave Engineering*, Chapter 4 and 5, 1966, McGraw Hill.

[9] P. So, W. J. R. Hoefer and P. Saguet, "CAD of E–Plane Circuits with Field–Theory Based Lookup Tables and Discontinuity Models", in IEEE MTT–S International Microwave Symposium Digest, May 23–27, New York.

[10] T. Itoh, *Spectral Domain Immitance Approach for Dispersion Characteristics of Generalized Printed Transmission Lines*, MTT–28, pp.733–736, July 1980.

[11] L. Schmidt and T. Itoh, *Spectral Domain Analysis of Dominant and Higher Order modes in Finlines*, MTT-28, pp.981–985, September 1980.

[12] J.B. Knorr and P.M. Shauda, *Millimeter-wave Finline Characteristics*, MTT-28, pp.737–743, July 1980.

[13] T.S. Laverghetta, *Modern Microwave Measurement and Technique*, Chapter 7, 1988, Artech House.

[14] Application Note 183, High Frequency Swept Measurements, November 1975, Hewlett Packard Ltd..

[15] Touchstone™ Manual and Touchstone™ Senior Manual, EEsof Inc..

[16] Microsoft Pascal Manual Version 3.31, Microsoft Inc..

[17] Turbo Pascal Manual Version 3.0 (4.0), 1985 (1987), Borland International Inc..

[18] Disk Operating System Version 3.20, 1986, IBM Ltd..