A METHOD FOR EVALUATING MUTUAL COUPLING BETWEEN DIELECTRIC COVERED SLOTS FED BY A BOXED STRIPLINE AND ITS IMPLEMENTATION INTO AN ARRAY DESIGN PROCEDURE

by

Pisti B. Katehi
Radiation Laboratory
3240 EECS Building
The University of Michigan
Ann Arbor, MI 48109-2122
Phone: (313) 747-1796

RL-858 = RL-858

January, 1989

I. INTRODUCTION

The importance of stripline-fed slots in array applications has been extensively discussed in many papers [1] - [3]. Owing to the wide applicability of these radiating structures, very accurate analysis and design techniques have been developed which account for electromagnetic coupling and the finite thickness of the walls [1]-[5]. In all these models, the exterior surface of the slot is left open into free space and the coupling is due to space waves only. In recent applications, for many practical considerations, the slots are coated with a layer of insulating material which has a great effect on the performance of the array. The presence of the insulating material affects the resonant characteristics of the slots and their electromagnetic coupling, as well, due to the excitation of surface waves in the dielectric.

The present paper solves the problem of mutual coupling between two electromagnetically coupled dielectric covered slots fed by boxed striplines and describes how it can be incorporated into an array design procedure. Since the slots are electromagnetically shielded in the interior region with the use of shorting pins as shown in Figure 1, there is no interior coupling between them. The method applied for the evaluation of the exterior coupling is based on the reaction theorem [6] and accounts for the interactions between the two slots due to space and surface waves excited in the air-dielectric region (see Figure 2). The major assumption in the mathematical development which will be presented here is that the slots are narrow and close to resonance. All the other parameters of the problem are arbitrary including cover thickness and relative permittivity.

II. THEORY

By mutual coupling we mean the electromagnetic interactions between elements of an array. Since the strength of these interactions can be measured in different ways, the definition of the mutual coupling is not unique. In fact, mutual coupling is defined in such a way that can be easily incorporated into the equations which characterize the array and govern its design. In this presented work the design procedure parallels very closely with the one developed by Elliott et al and, therefore, the adopted definition for the mutual coupling is consistent with this design procedure.

(a) Evaluation of Mutual Coupling

Figure 1 shows a top view of two stripline-fed slots excited by the dominant stripline mode. Throughout the mathematical derivations it has been assumed that the thickness of the slot walls is very small compared to the wavelength in free space (λ_o). In addition, both slots have been assumed narrow and operating under resonant conditions. At this point, we define as mutual coupling the voltage applied across slot No. 1 due to the field excited on the aperture of slot No. 2. This voltage $V_{1,m}^{s}$ indicates the effect on the field on the apeture of the slot due to electromagnetic coupling.

The mutual coupling in the form of a voltage as it has been defined above can be deduced with the aid of the reciprocity theorem as has been described in [6] and is repeated here for completeness. At this point, let the dominant TEM stripline mode of amplitude A^a excite slot No. 1 as shown in Figure 3. Then a slot voltage V_1^s develops at the center of the slot and, as a result, a backscattering TEM mode is launched to the left of the slot. The TEM fields at the reference plane aa' on the first stripline are given by

$$E^a_{x,y} = e^{TEM}_{x,y}\left[A^a e^{-jk(z-z_{aa'})} + B^a e^{jk(z-z_{aa'})} + D^a e^{jk(z-z_{aa'})}\right] \qquad (1)$$

$$H_{y,x} = \pm\frac{1}{\eta} e^{TEM}_{x,y}\left[A^a e^{-jk(z-z_{aa'})} - B^a e^{jk(z-z_{aa'})} - D^a e^{jk(z-z_{aa'})}\right] \qquad (2)$$

where $z_{aa'}$ denotes the position of the reference plane aa', B^a is the amplitude of the backscattered wave due to the presence of the slot and D^a is the amplitude of the backscattered wave due to the stripline load termination. Also in equations (1) and (2) $e^{TEM}_{x,y}$ is an anlytic function of x,y which gives the dependence of the dominant stripline mode on the x,y coordinates, k is the propagation constant of the dominant mode and η is the intrinsic impedance of the medium surrounding the stripline.

The electric field distribution $\bar{E}_s^1$ on the aperture of slot No. 1 induces electric currents at the edges of slot No. 2 resulting in an electric field distribution $\bar{E}_s^2$ on its aperture. Assuming the above electric field distributions the following two sets of sources may be considered on the reference plane aa'

$$\bar{J}^a = \hat{z} \times \bar{H}^a = - \hat{x} H_y^a + \hat{y} H_x^a \tag{3}$$

$$\bar{M}^a = \bar{E}^a \times \hat{z} = \hat{x} E_y^a - \hat{y} E_x^a \tag{4}$$

and

$$\bar{M}^b = \bar{E}_s^2 \times \hat{n} \tag{5}$$

on the exterior surface of slot No. 2 with the opening covered by a perfect conductor (see Figure 4). With the aid of the reciprocity theorem, these sources are related as it follows:

$$\iint_{Sa} \left(\bar{E}^b \cdot \bar{J}^a - \bar{H}^b \cdot \bar{M}^a\right) ds = - \iint_{S_2} \bar{H}_{ext}^a \cdot \bar{M}^b \, ds \tag{6}$$

where $\bar{H}_{ext}^a$ is the field radiated by slot No. 1 through the air-dielectric region on the aperture of slot No. 2. In equation (6), $\bar{E}^b$, $\bar{H}^b$ are the fields on the reference plane aa' of stripline No. 1 due to the equivalent source $\bar{M}^b$. These fields are given by

$$E^{b}_{x,y} = e^{TEM}_{x,y} \left[B^{b} + D^{b} \right] e^{jk\,(z-z_{aa'})} \tag{7}$$

and

$$H^{b}_{y,x} = + \frac{1}{\eta}\, e^{TEM}_{x,y} \left[B^{b} + D^{b} \right] e^{jk\,(z-z_{aa'})} \tag{8}$$

where B^b, D^b are the backscattered waves from the slot and stripline load due to the electromagnetic coupling between the slots. In view of equations (1) to (5) and (7), (8), the LHS of equation (6) takes the form:

$$\iint_{S_a} dx\ dy \left\{ E^{b}_{y} H^{a}_{x} - E^{b}_{x} H^{a}_{y} - H^{b}_{x} E^{a}_{y} + H^{b}_{y} E^{a}_{x} \right\} =$$

$$= \iint_{S_a} dx\ dy \left\{ \left[- \frac{1}{\eta} \left(e^{TEM}_{y} \right)^{2} - \frac{1}{\eta} \left(e^{TEM}_{x} \right)^{2} \right] \left(B^{b} + D^{b} \right) \left(A^{a} - B^{a} - D^{a} \right) + \right.$$

$$\left. + \left[- \frac{1}{\eta} \left(e^{TEM}_{y} \right)^{2} - \frac{1}{\eta} \left(e^{TEM}_{x} \right)^{2} \right] \left(B^{b} + D^{b} \right) \left(A^{a} + B^{a} + D^{a} \right) \right\}$$

$$= - \frac{2}{\eta} A^{a} \left(B^{b} + D^{b} \right) \iint_{S_a} \left\{ \left(e^{TEM}_{x} \right)^{2} + \left(e^{TEM}_{y} \right)^{2} \right\} \tag{9}$$

By comparing equations (6) and (9) we conclude that

$$A^{a} \ (B^{b} + D^{b}) = \frac{1}{2S_{TEM}} \iint_{S_2} \bar{H}^{a}_{ext} \bullet M^{b}\ ds \tag{10}$$

in which

$$S_{TEM} = -\frac{1}{\eta}\iint_{S_a}\left\{\left(e_x^{TEM}\right)^2 + \left(e_y^{TEM}\right)^2\right\} \quad . \tag{11}$$

From the equivalent circuit for the individual slot No. 1 in Figure 3 we can easily find that

$$D^b = C^b \frac{\bar{Z}_{L1} - 1}{\bar{Z}_{L1} + 1} \tag{12}$$

where C^b is the forward scattering coefficient of the slot due to electromagnetic coupling between the slots and $\bar{Z}_{L1}$ is the normalized load termination. As it has been shown in [7] the backscattering and forward scattering coefficient B^b and C^b are related through the following simple expression.

$$B^b = -C^b \quad . \tag{13}$$

In view of (12) and (13), (10) takes the form:

$$B^b = \frac{1}{4S_{TEM}} \cdot \frac{1}{A^a} (\bar{Z}_{L1} + 1) \iint_{S_2} \bar{H}^a_{ext} \cdot \bar{M}^b \, ds \quad . \tag{14}$$

Furthermore, the amplitude of the backscattering wave is related to the voltage across the slot through a function which depends only on the geometrical and electrical characteristics of the stripline, the cavity created by the shorting pins and the slot aperture cut on its upper wall [7]. As a result, the following relations may be considered:

$$B^b = F_n V^s_{1,m} \tag{15}$$

$$\frac{V_1^s}{A^a} = \frac{1}{F_n} \frac{\bar{Z}_{s1}}{\bar{Z}_{s1} + \bar{Z}_{L1}} \tag{16}$$

where F_n is called the backscattering function and can be derived by solving the problem of an isolated stripline-fed slot. Also, $V_{1,m}^s$ is the voltage applied across slot No. 1 due to mutual coupling, V_1^s is the total voltage applied across slot No. 1 and $\bar{Z}_{s1}$ is the normalized self impedance of slot No. 1 with slot No. 2 absent. From equations (10), (11) and (12) we can conclude that

$$V_{1,m}^s = \frac{1}{4S_{TEM}} \frac{V_1^s}{A^a} (\bar{Z}_{L1} + 1) \frac{1}{F_n} \iint_{S_2} \frac{\bar{H}_{ext}^a}{V_1^s} \bullet \bar{M}^b \, ds \tag{17}$$

or

$$V_{1,m}^s = \frac{1}{4S_{TEM}} \frac{\bar{Z}_{s1} (\bar{Z}_{L1} + 1)}{\bar{Z}_{s1} + \bar{Z}_{L1} + 1} \frac{1}{F_n^2} \iint_{S_2} \frac{\bar{H}_{ext}^a}{V_1^s} \bullet \bar{M}^b \, ds \quad . \tag{18}$$

Equation (18) could be simplified to the following form:

$$V_{1,m}^s = - \frac{1}{4S_{TEM}} \frac{\bar{Z}_{s1} (\bar{Z}_{L1} + 1)}{\bar{Z}_{s1} + \bar{Z}_{L1} + 1} \frac{1}{F_n^2} V_2^s Y_{12}^{ext} \tag{19}$$

with Y_{12}^{ext} given by

$$Y_{12}^{ext} = - \iint_{S_2} \frac{\bar{H}_{ext}^a}{V_1^s} \bullet \frac{\bar{M}^b}{V_2^s} \, ds \tag{20}$$

and V_2^s the total voltage applied on slot No. 2.

In (20), Y_{12}^{ext} has the units of an admittance and is called the ideal mutual admittance between the two slots. In addition, $\bar{H}_{ext}^a$ has the form [8]

$$\vec{H}_{ext}^{a} = \iint_{S_1} (k_d^2 \,\bar{\bar{I}} + \overline{\nabla}\,\overline{\nabla}) \cdot \bar{\bar{G}}^{d} \cdot \vec{M}^{a} \, ds \qquad (21)$$

with $\vec{M}^a$ the equivalent magnetic current on the aperture of slot No. 1 and $\bar{\bar{G}}^d$ the dyadic Green's function for the air-dielectric region. Since the slots have been assumed narrow the equivalent magnetic currents can take the form:

$$\bar{M}^{a} = \hat{y} \sum_{n=1}^{N1} V_{1n} \, f_{1n}(y) \, g_{1n}(z) \qquad (22)$$

and

$$\bar{M}^{b} = \hat{y} \sum_{m=1}^{N2} V_{2m} \, f_{1m}(y) \, g_{1m}(z) \qquad (23)$$

where V_{1n} and V_{2m} are known constants, $\{f_{1,2n}(y),\ n = 1,2,\ldots\}$ are piecewise sinusoidal functions each one extending over a subsection of width w and length l_y on the slot apertures [6], [7] and $\{g_{1,2n}(z) \,.\, n = 1,2,\ldots\}$ are functions chosen so that the edge conditions on the slots are satisfied.

By substituting equations (21)-(23) into (20) the ideal mutual admittance takes the form

$$Y_{12}^{ext} = - \sum_{n=1}^{N1} V_{1n} \sum_{m=1}^{N2} V_{2m}\, y_{nm}^{ext} \tag{24}$$

with y_{nm}^{ext} given by [8]

$$y_{nm}^{ext} = \left< f_n\, \delta(z-z_{1o}),\ K_{yy}^{d},\ f_m\, g_m \right> \quad . \tag{25}$$

and

$$K_{yy}^{d} = k_d^2 \left(G_{yy}^{d} + \frac{\partial^2}{\partial y^2} G_{yy}^{d}\right) + \frac{\partial^2}{\partial x \partial y} G_{xy}^{d} \tag{26}$$

In equation (25), z_{1o} indicates the position of the slot along the stripline.

If the slots are around resonance and are excited in phase, then both field distributions are equiphase and cosinusoidal. As a result, it is appropriate to assume that the equivalent magnetic currents are given by:

$$\bar{M}^{a} = \hat{y}\, V_1^{s} \sum_{n=1}^{N1} \cos\left(\frac{\pi(n-1)}{N1-1} - \frac{\pi}{2}\right) f_{1n}(y)\, g_{1n}(z) \tag{27}$$

$$\bar{M}^{b} = \hat{y}\, V_2^{s} \sum_{n=1}^{N2} \cos\left(\frac{\pi(n-1)}{N2-1} - \frac{\pi}{2}\right) f_{2n}(y)\, g_{2n}(z) \qquad (28)$$

Due to the fact that the expression for y_{12}^{ext} is variational in nature the above assumption has a minor effect on the value of mutual admittance.

(b) Design Equations

As it has been explained previously, the consideration of mutual interactions in an array design procedure is very important. In order to incorporate the effect of mutual coupling in a simple and effective way, a design procedure was developed by Elliott et al which has proven to be accurate, practical and applicable to slot arrays of any size. This design method is an iterative procedure and is based on two equations. The derivation of these equations for the problem of stripline-fed slots is presented in detail below. Let us consider the equivalent circuit for a slot No. n in an array environment (see Figure 5). For the case where this slot is electromagnetically coupled to other neighboring array elements the self impedance becomes the active self impedance $\bar{Z}_{s}^{a}$. If the two ports of this circuit are at a distance equal to one wavelength of the dominant stripline mode (λ_{TEM}), then it can be easily proved that

$$B_n = -C_n = \frac{1}{2}\bar{Z}_{sn}^{a} I_n \qquad (29)$$

where I_n is the mode current at the juncture where the series element is placed. In view of (29) equation (15) becomes

$$\bar{Z}^{a}_{sn} = 2 F_n V^s_n / I_n \quad . \tag{30}$$

Equation (30) is a principal result of the analysis and is called the first design equation. Furthermore, the normalized active impedance $\bar{Z}^{a}_{sn}$ can be expressed in terms of the scattered and incident waves at the nth slot. Specifically, the input impedance to the right of the slot is

$$\bar{Z}_{Ln} = \frac{A_n + C_n + D_n}{A_n + C_n - D_n} = \frac{A_n - B_n + D_n}{A_n - B_n - D_n} \quad . \tag{31}$$

Similarly, the input impedance to the left of the same slot is

$$\bar{Z}^{a}_{sn} + \bar{Z}_{Ln} = \frac{A_n + B_n + D_n}{A_n - B_n - D_n} \quad . \tag{32}$$

From equations (31) and (32) we conclude that

$$\bar{Z}^{a}_{sn} = \frac{2 B_n}{A_n - B_n - D_n} \quad . \tag{33}$$

Since the wave reflected by the terminating load has an amplitude D_n given by

$$D_n = (A_n - B_n) \frac{\bar{Z}_{Ln} - 1}{\bar{Z}_{Ln} + 1} \quad , \tag{34}$$

then equation (33) becomes

$$\bar{Z}^{a}_{sn} = (\bar{Z}_{Ln} + 1)\,\frac{B_n}{A_n - B_n} \qquad (35)$$

or

$$A_n = \frac{\bar{Z}^{a}_{sn} + \bar{Z}_{Ln} + 1}{\bar{Z}^{a}_{sn}}\,B_n \quad . \qquad (36)$$

For the case of an isolated slot the above equation takes the form

$$A_n = \frac{\bar{Z}_{sn} + \bar{Z}_{Ln} + 1}{\bar{Z}_{sn}}\,F_n\,V^{s}_{n,s} \qquad (37)$$

where $V^{s}_{n,s}$ is the voltage applied on the single slot.

In view of the above, equation (33) becomes

$$\bar{Z}^{a}_{sn} = (\bar{Z}_{Ln} + 1)\,\frac{F_n\,V^{s}_{n}}{\left(1 + \dfrac{\bar{Z}_{Ln} + 1}{\bar{Z}_{sn}}\right) F_n\,V^{s}_{n,s} - F_n\,V^{s}_{n}} \qquad (38)$$

where V^{s}_{n} is the total voltage applied on the slot and is given by

$$V^{s}_{n} = V^{s}_{n,s} + V^{s}_{n,m} \quad . \qquad (39)$$

Equation (38) may be rewritten to give

$$\bar{Z}_{sn}^{a} = (\bar{Z}_{Ln} + 1) \frac{F_n V_n^s}{\dfrac{\bar{Z}_{Ln} + 1}{\bar{Z}_{sn}} F_n V_n^s - \left(\dfrac{\bar{Z}_{Ln} + 1}{\bar{Z}_{sn}} + 1 \right) F_n V_{n,m}^s} \tag{40}$$

or

$$\bar{Z}_{sn}^{a} = \frac{1}{\dfrac{1}{\bar{Z}_{sn}} - \left(\dfrac{1}{\bar{Z}_{sn}} + \dfrac{1}{\bar{Z}_{Ln} + 1} \right) \dfrac{V_{n,m}^s}{V_n^s}} \quad . \tag{41}$$

Equation (41) is the second design equation and together with (30) permit the design of an array of slots fed by boxed striplines.

(c) Design Procedure

As preliminary preparations to the design procedure of the array we have to compute the following

(a) The desired voltage distribution V_i^s/V_j^s from the pattern requirements, where i is any slot in the array and j is the reference slot.

(b) The slots' self impedances $\bar{Z}_{sn}$ and backscattering functions F_n as functions of the length and offset of the slots.

(c) The ratios I_i/I_j from the feeding network requirements where $I_{i,j}$ are the currents flowing on the striplines exciting the ith slot and reference slot respectively.

After computing the above one can assume a set of original lengths and offsets for all the slots in the array. Specifically, these will be the resonant lengths for the desired initial offsets. With these lengths and offsets, we evaluate the ratio $\frac{V^s_{n,m}}{V^s_n}$ which includes interaction from all neighboring slots. With this computed, a computer search is performed to find new values of slots lengths or offsets if necessary so that the denominator of (41) becomes real. In this manner, we find many possible lengths or offsets for each slot which can satisfy the above requirement. As a next step we use equation (30) in the following form

$$\frac{\bar{Z}^a_{sn}}{\bar{Z}^a_{sj}} = \frac{F_n}{F_j} \frac{V^s_n}{V^s_j} \frac{I_j}{I_n} \tag{42}$$

where subscript j denotes quantities related to the reference slot. Then, from all values of lengths and offsets found above, we try to identify those which satisfy equation (42). From these values which may correspond to more than one solution we keep the most realistic one. This choice can be made easier if other requirements about the feeding netowrk are considered.

The described procedure must be iterated because the chosen values will undoubtedly not agree with the original guess for slot lengths and offsets.

With the array design procedure carefully described, our next step is to accurately characterize the slots as isolated elements and evaluate mutual coupling between them.

The purpose of this work is to numerically evaluate mutual coupling between stripline-fed slots in such a manner so that it can be easily implemented into the described design procedure. Numerical results for the mutual coupling following the technique presented in previous sections are presented below.

III. NUMERICAL RESULTS

As we mentioned previously, the definition of mutual coupling is not unique and could be given either in the form of an admittance (see ideal mutual admittance defined above) or an impedance. Here, using Booker's relation [9] we define an ideal mutual impedance Z_{12}^{ext} given by

$$Z_{12}^{ext} = \frac{\eta_o^2}{2} Y_{12}^{ext} \tag{43}$$

where Y_{12}^{ext} is given by (24) and η_o is the intrinsic impedance of free-space (=120π).

Figure 6 shows the real and imaginary part of the ideal mutual impedance as functions of the distance between slots with dielectric cover replaced by air. The two slots are identical and their lengths are equal to the resonant length of the isolated slot [10] ($L_{d1} = L_{d2} = 0.3924\ \lambda_o$). Figures 7, 8, 9 show the same impedance as function of the inter-slot distance for the case of a insulating cover with dielectric constant $\varepsilon_r = 2.2$ and thickness $h = 0.035\ \lambda_o$, $0.07\ \lambda_o$ and 0.106 respectively. In all these cases the slot lengths are equal to the resonant length of the isolated slots [10]. Specifically for $h = 0.035\ \lambda_o$, $L_{d1} = L_{d2} = 0.035\ \lambda_o$ (see Figure 7), for $h = 0.07\ \lambda_o$, $L_{d1} = L_{d2} = 0.342\ \lambda_o$ (see Figure 8) and for $h = 0.106\ \lambda_o$, $L_{d1} = L_{d2} = 0.337\ \lambda_o$ (see Figure 9). These figures, as it is expected, show that the real part of the ideal mutual impedance is rather insensitive to thin dielectric covers as oppose to the imaginary part which can be seriously affected by the presence of the insulating material.

IV. DISCUSSION AND CONCLUSIONS

This paper has presented a procedure for designing a dielectric covered slot array fed by a boxed stripline. This is an iterative procedure and takes into account all of the physical phenomena including mutual coupling due to the excitation of space and surface waves in the dielectric. In addition this work has presented an accurate and efficient method for evaluating the electromagnetic interactions between the elements of the array and has described in detail how the effect of these interactions can be incorporated into the design procedure.

However, in order to design an array additional information is needed about the characteristics of the isolated elements. Specifically the backscattering functions, the resonant length and self admittance have to be known as functions of various geometrical and electrical parameters. The above functions can be derived by polyfitting either theoretical or experimental results. The theoretical characterization of an isolated dielectric-covered slot fed by a boxed stripline will be presented in future work.

REFERENCES

[1] E. Strumwasser, R. J. Stegen, J. A. Short and J. R. Miller, "Slot Study in Rectangular TEM Transmission Line," Hughes Aircraft Company, Rep. TM 265, January 1952.

[2] P.K. Park and P. S. Elliott, "Design of Collinear Longitudinal Slot Arrays Fed by a Boxed Stripline," IEEE Trans. on Antennas and Propag., vol. AP-29, No. 1, January 1981.

[3] R. Shavit and R. S. Elliott, "Design of an Array of Transverse Slots Fed by a Boxed Stripline," IEEE Trans. on Antennas and Propag., vol. AP-31, No. 7, July 1983.

[4] R. S. Robertson and R. S. Elliott, "The Design of Transverse Slot Arrays Fed by the Meandering Strip of a Boxed Stripline," IEEE Trans. on Antennas and Propag., vol. AP-35, No. 3, March 1987.

[5] R.S. Elliott and G. J. Stern, "The Design of Microstrip Dipole Arrays Including Mutual Coupling, Part I: Theory," IEEE Trans. on Antennas and Propag., vol. AP-29, No. 5, September 1981.

[6] R. S. Elliott, "An Improved Design Procedure for Small Arrays of Shunt Slots," IEEE Trans. Antennas and Propag., vol. AP-31, pp. 48-53, January 1983.

[7] Rewven Shavit, "Design of an Array of Transverse Slots Fed by a Boxed Stripline," Ph.D. Dissertation, Electrical Engineering Department, University of California, Los Angeles, June 1982.

[8] P. B. Katehi, "A Space Domain Integral Equation Approach in the Analysis of Dielectric Covered Slots," to appear in Radio Science in January 1989.

[9] R.S. Elliott, "Antenna Theory and Design," Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[10] J. D. Miller, "Dielectric Cover Thickness Effects on Stripline Slot Impedance and Frequence," Technical Report BCD TR-155, Allied Bendix Corporation Communication Division, January 1987.

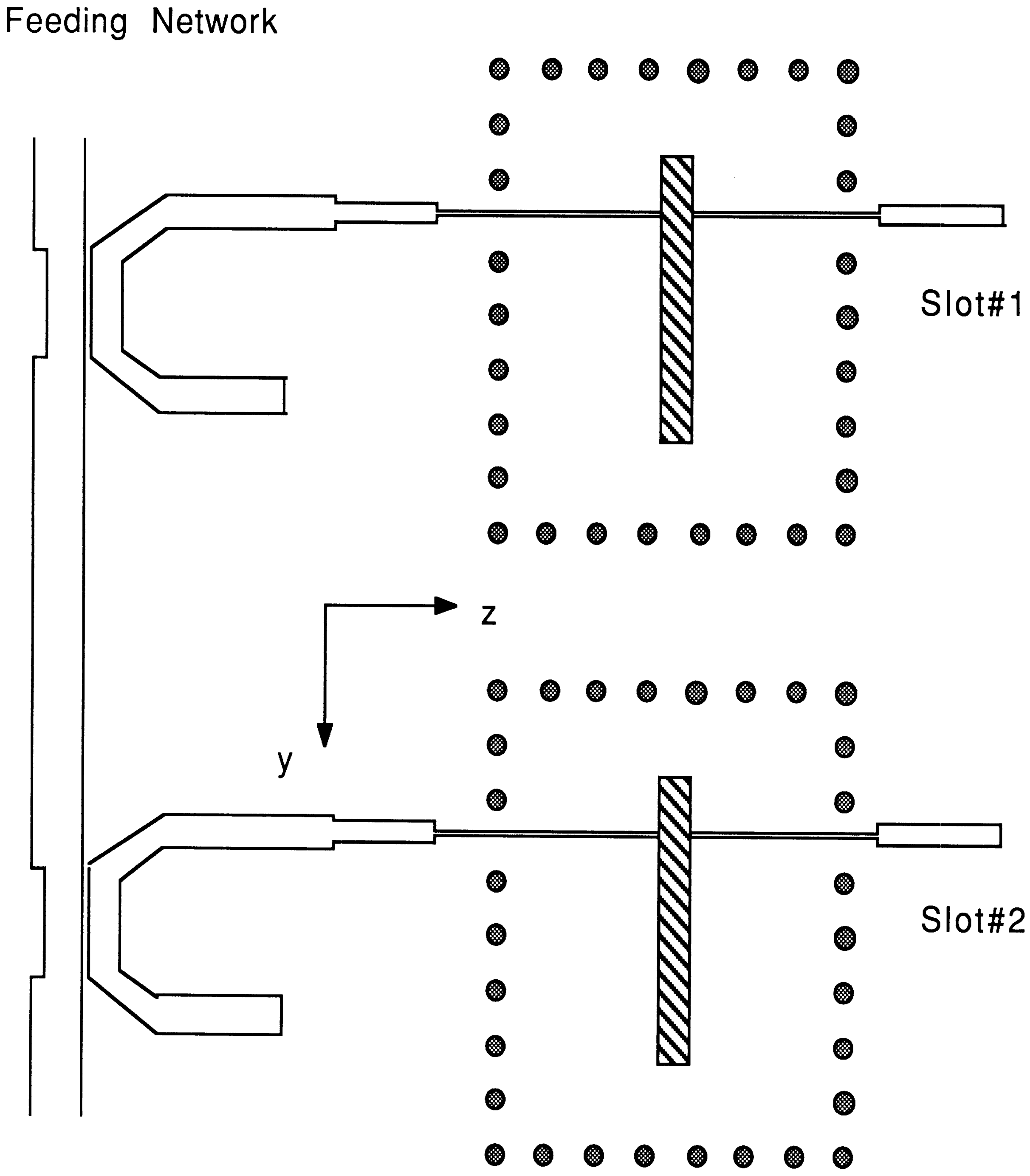


Figure 1: Two dielectric covered collinear slots excited by boxed striplines.

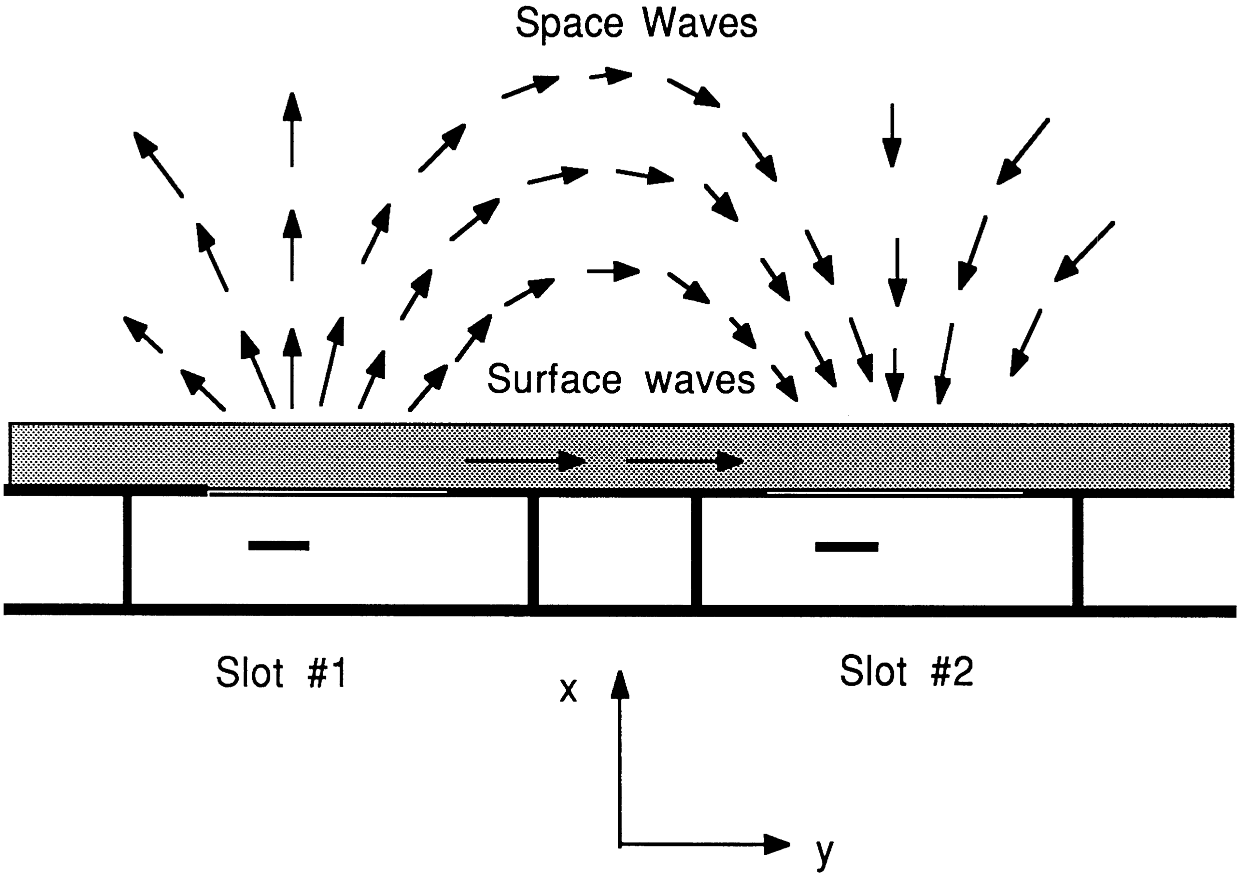


Figure 2: Two slots electromagnetically coupled through surface and space waves

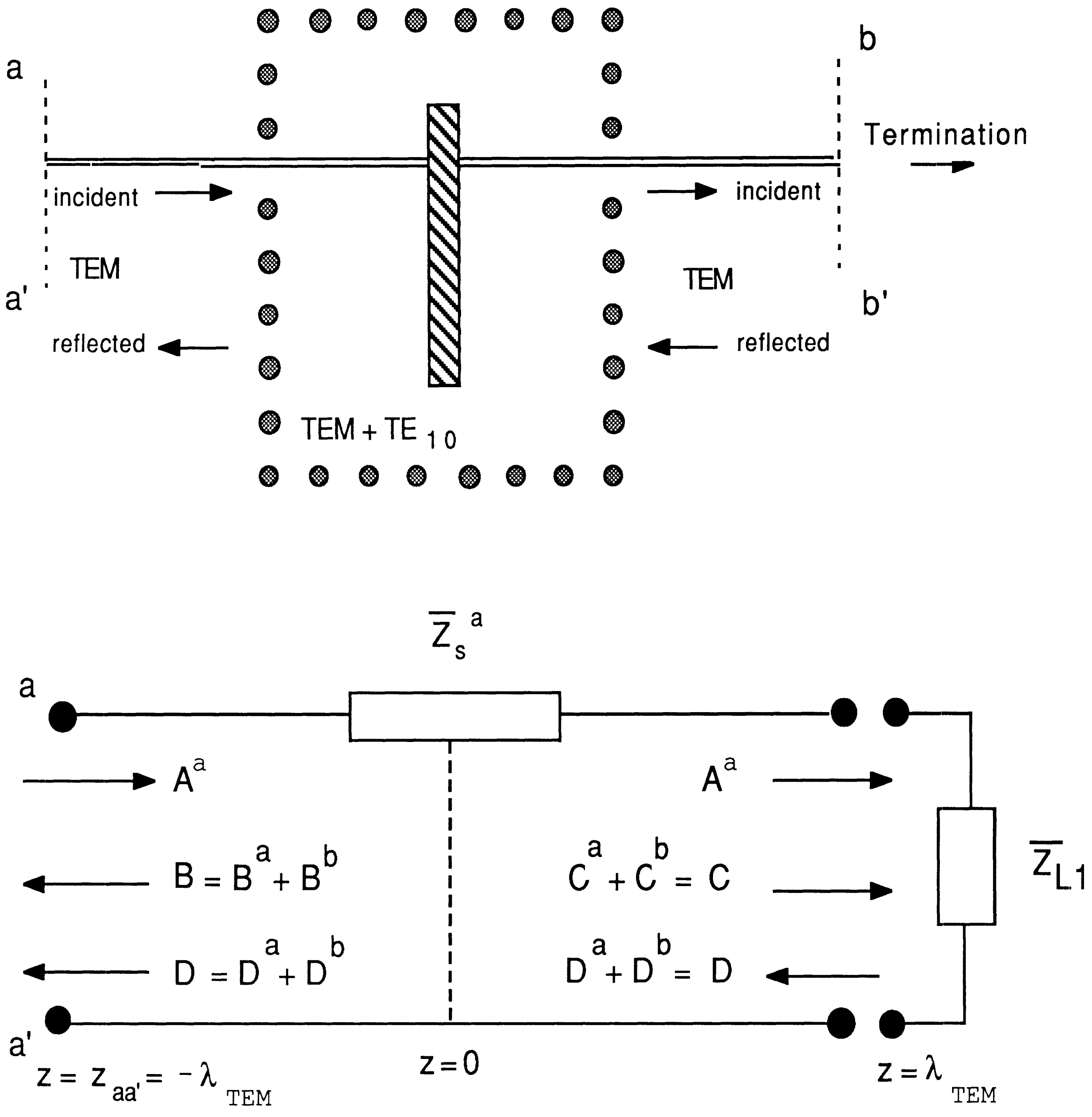


Figure 3 : A slot excited by the dominant TEM stripline mode and its equivalent circuit

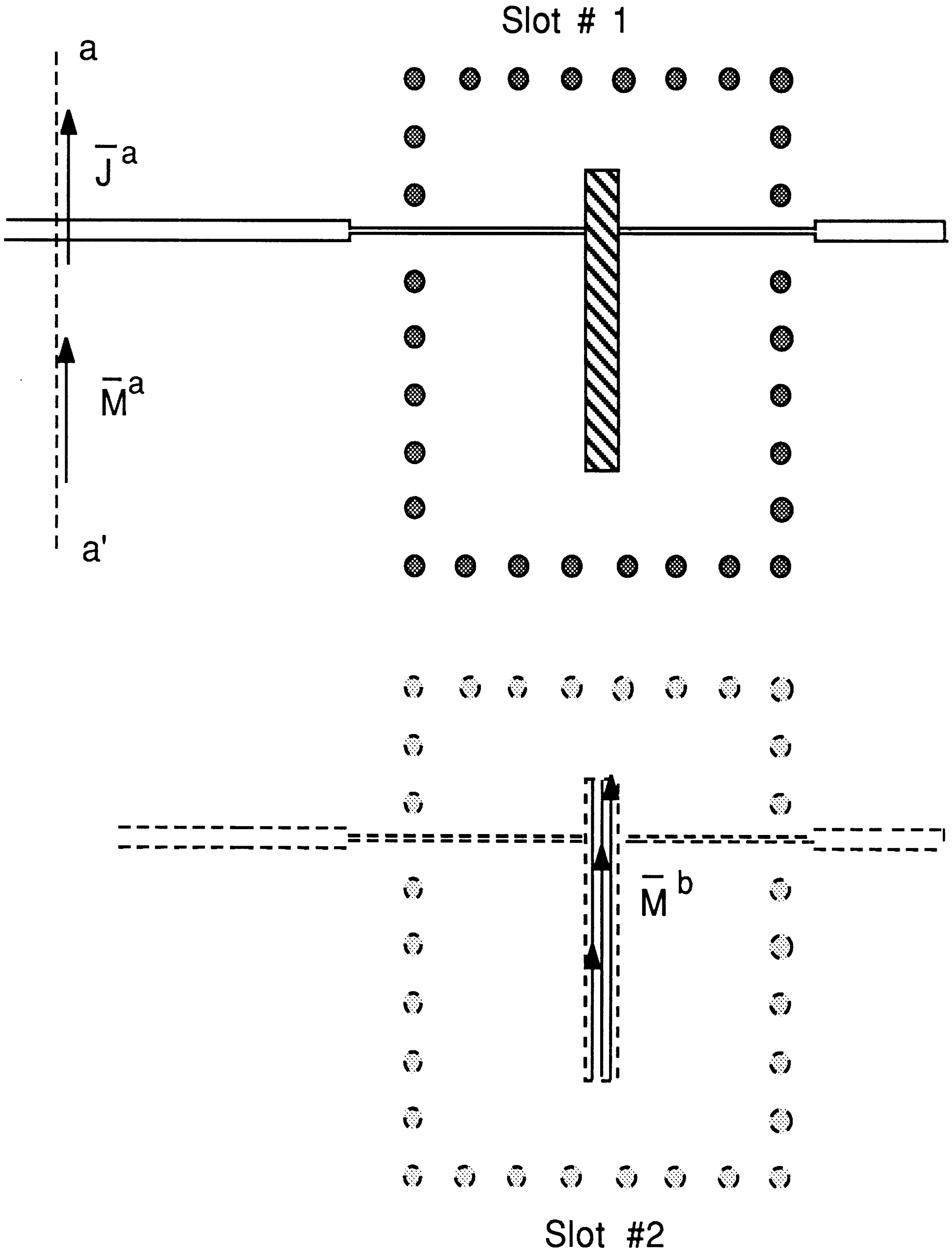


Figure 4 : Equivalent sources used in the evaluation of mutual coupling .

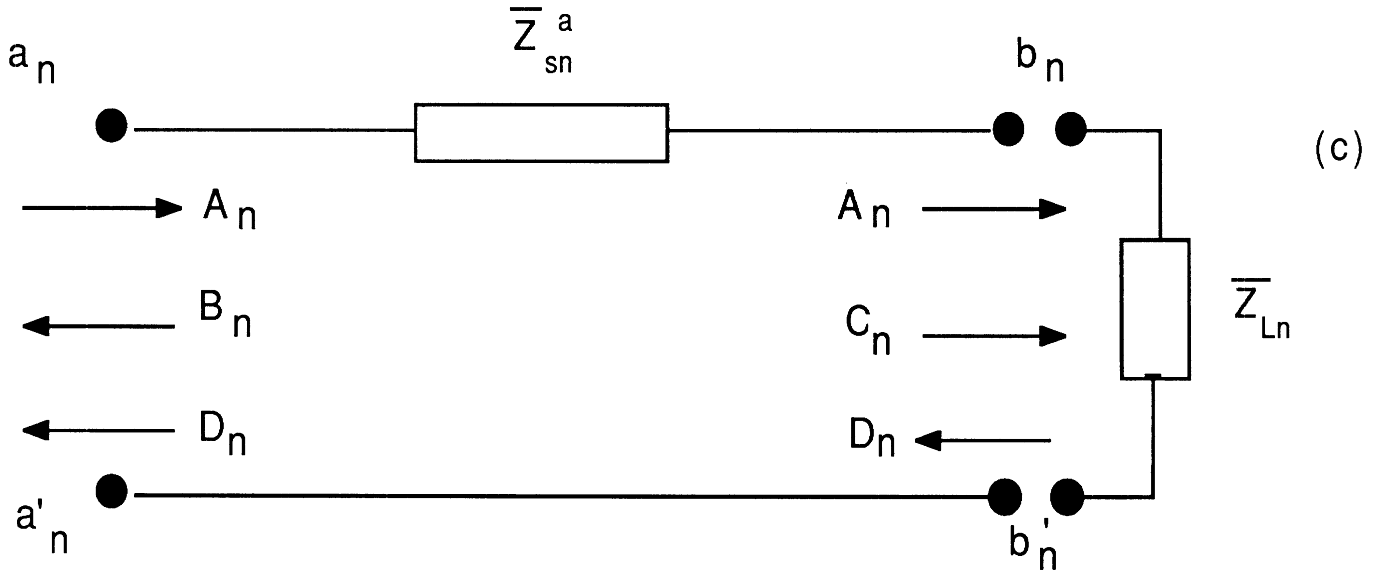


Figure 5 : Equivalent circuit for the nth element of a dielectric covered slot array

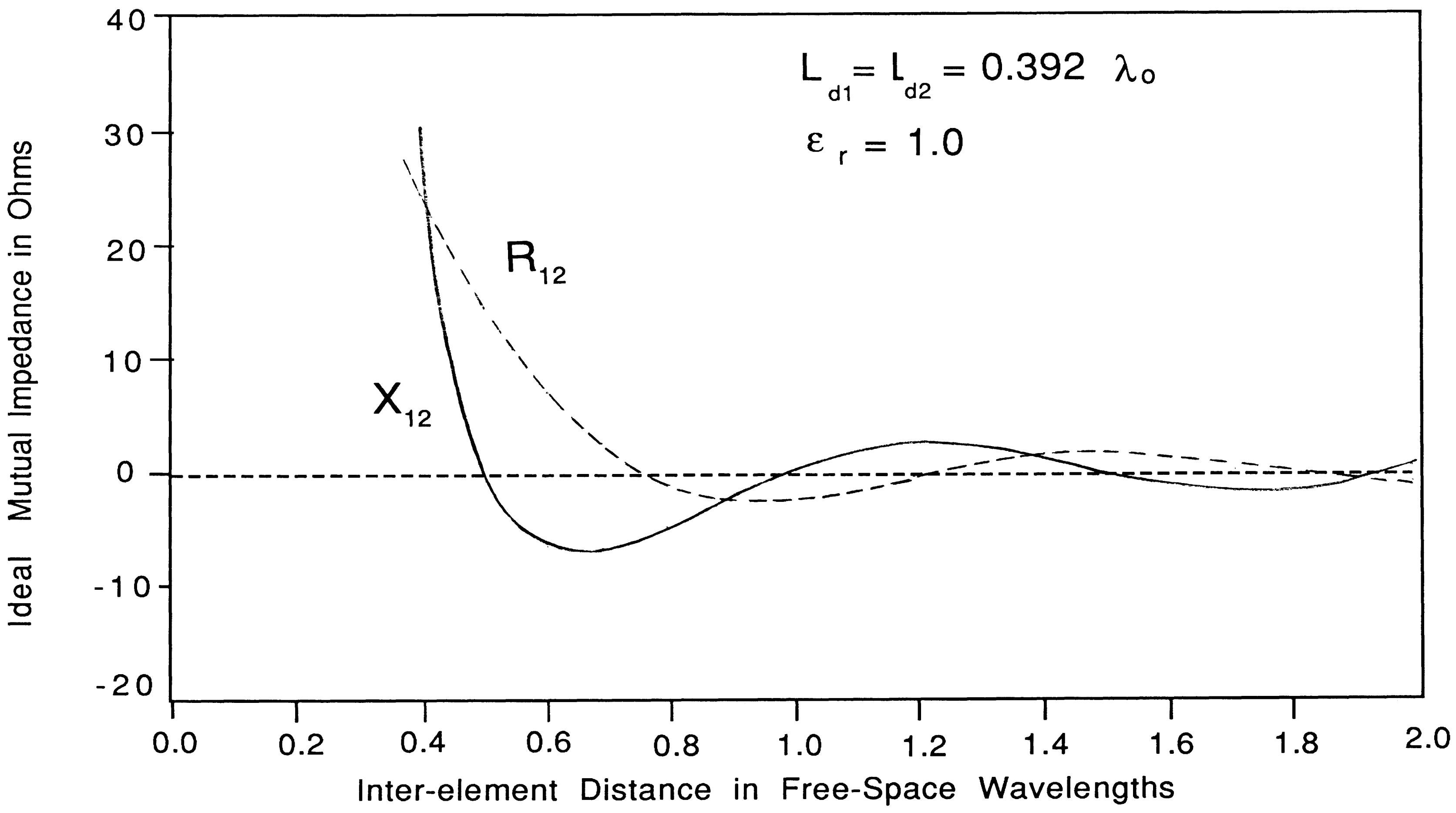

$L_{d1} = L_{d2} = 0.392\ \lambda_o$
$\varepsilon_r = 1.0$
R_{12}
X_{12}
Ideal Mutual Impedance in Ohms
40
30
20
10
0
-10
-20
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Inter-element Distance in Free-Space Wavelengths

Figure 6

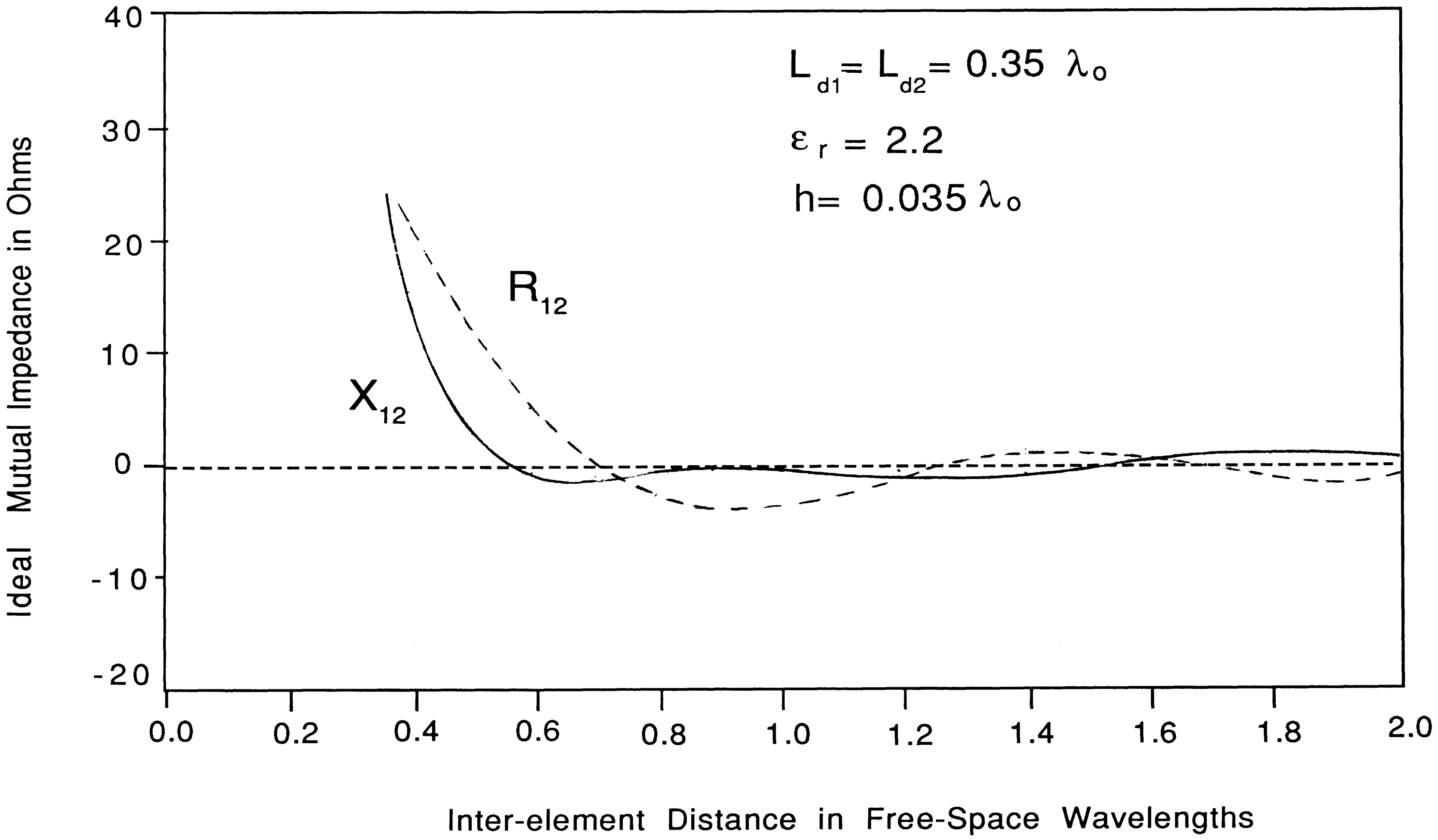
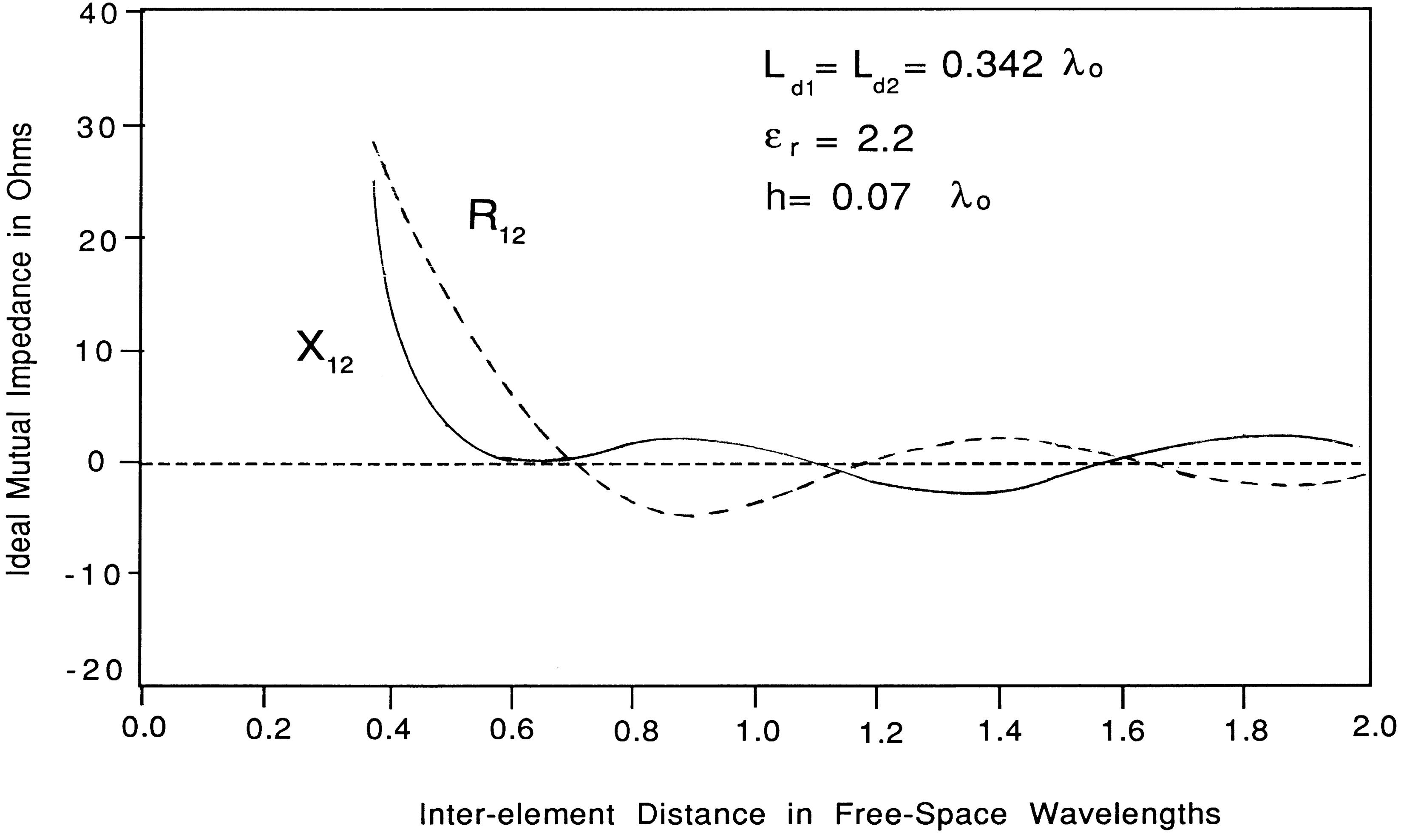
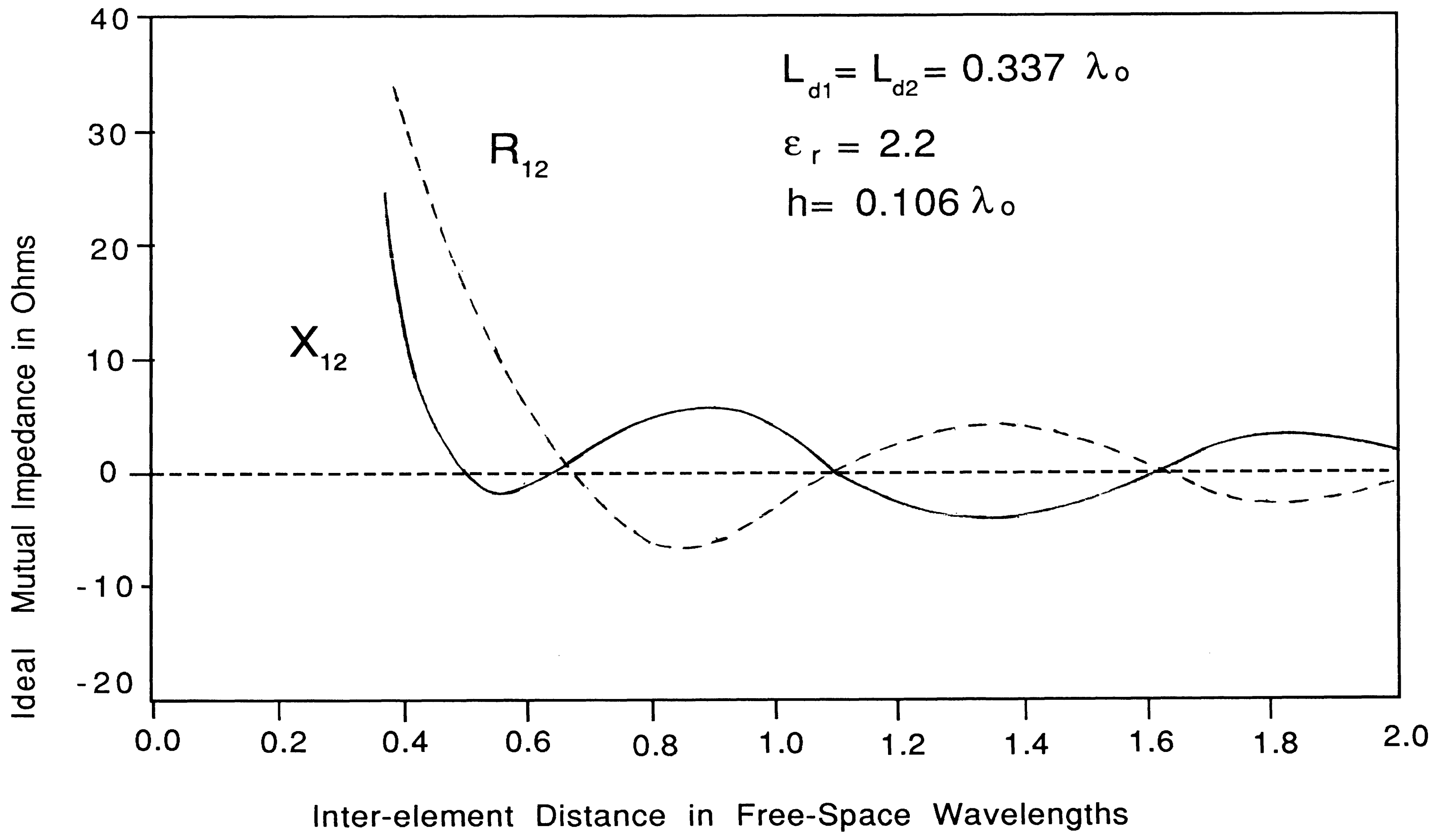
40
30
20
10
0
-10
-20
Ideal Mutual Impedance in Ohms
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Inter-element Distance in Free-Space Wavelengths
$L_{d1} = L_{d2} = 0.35\ \lambda_o$
$\varepsilon_r = 2.2$
$h = 0.035\ \lambda_o$
R_{12}
X_{12}

Figure 7



40
30
20
10
0
-10
-20
Ideal Mutual Impedance in Ohms
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Inter-element Distance in Free-Space Wavelengths
$L_{d1} = L_{d2} = 0.342\ \lambda_o$
$\varepsilon_r = 2.2$
$h = 0.07\ \lambda_o$
R_{12}
X_{12}

Figure 8



40
30
20
10
0
-10
-20
Ideal Mutual Impedance in Ohms
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Inter-element Distance in Free-Space Wavelengths
$L_{d1} = L_{d2} = 0.337\ \lambda_o$
$\varepsilon_r = 2.2$
$h = 0.106\ \lambda_o$
R_{12}
X_{12}

Figure 9

LIST OF PROGRAMS

PROGRAM: SLOT_DESIGN.FTN

This program evaluates the mutual coupling between two dielectric-covered slots fed by boxed striplines for various inter-slot distances S. This distance S varies between S_{min} and S_{max} in steps of S_{min}.

COMMENTS

1) The dielectric constant **ER** (see files **data_wave_mutual** and **slot_design.ftn**) should be always different than 1. It can be as small as 1.00001.

2) The substrate thickness **H** (see files **data_wave_mutual** and **slot_design.ftn**) should be larger than 0.02. The substrate thickness can become smaller than that with some minor modifications in the program.

3) The first of the **logitudinal offsets of the slots** (see **data_wave_mutual**) should be equal to 1 and the second should be equal to $(S_{max}+(L_{d1}+L_{d2})/2)/DLX$ where S_{max} is the maximum separation between the slots, $L_{d1,d2}$ are the lengths of the two dipoles and DLX is the subsection length.

4) The subsection length **DLX** (see **data_wave_mutual**) should be equal to $L_{d1,d2}/N_{d1,d2}$ where $N_{d1,d2}$ are the number of points on the slots (see data_wave_mutual).

5) In the main program on the fistr line of **DO-LOOP 108** the parameter **IZ_STEP** should be equal to $INT(S_{min}/DLX)$ where S_{min} is the minimum separation between the slots. Also, **IZ_MIN** should be equal to $S_{min}+1$.

apollo domain

CAEN/Apollo

```
K    K     A    TTTTTTT EEEEEEE H     H  III
K   K     A A      T    E       H     H   I
K  K     A   A     T    E       H     H   I
KKK     A     A    T    EEEEE   HHHHHHH   I
K  K    AAAAAAA    T    E       H     H   I
K   K   A     A    T    E       H     H   I
K    K  A     A    T    EEEEEEE H     H  III
```

```
ddddd    aa   ttttt    aa          w    w   aa   v    v eeeeee        m    m u    u ttttt  u    u   aa   l
d    d  a  a    t     a  a         w    w  a  a  v    v e             mm  mm u    u   t    u    u  a  a  l
d    d a    a   t    a    a        w    w a    a v    v eeeee         m mm m u    u   t    u    u a    a l
d    d aaaaaa   t    aaaaaa        w ww w aaaaaa v    v e             m    m u    u   t    u    u aaaaaa l
d    d a    a   t    a    a        ww  ww a    a  v  v  e             m    m u    u   t    u    u a    a l
ddddd  a    a   t    a    a ______ w    w a    a   vv   eeeeee ______ m    m  uuuu    t     uuuu  a    a llllll
```

//tera/users/katehi/tape/data_wave_mutual

LAST MODIFIED ON: 89/01/25 1:13 PM
FILE PRINTED: 89/01/25 1:30 PM

```

     ---- Dielectric constant ---

     2.62

     ---- Substrate Thickness ---

     0.02544

     ---- Conductor Thickness ---

     0.00001

     ---- Number of Slots ----

     2

     ---- Transverse offsets of the slots ----

     0.3166
     0.60864

     ---- Longitudinal offets of the slot ----

     1
     57

     ---- Slot widths ----

     0.048
     0.048

     ---- Slot Excess Widths ----

     0.0
     0.0

     ---- Subsection Length ----

     0.0125
     0.0125

     ---- Lower Limit of the Tail Contribution ----

     100.0

     ---- Number of Points on the Slots ----

     29
     29

     ---- Error in the evaluation of the series ----

     1.D-6
```

```
################                 a p o l l o   d o m a i n                  ################
################                                                            ################
################                        CAEN/Apollo                         ################
################                                                            ################
```

```
K    K     A    TTTTTTT EEEEEEE H     H   III
K   K     A A      T    E       H     H    I
K  K     A   A     T    E       H     H    I
KKK     A     A    T    EEEEE   HHHHHHH    I
K  K    AAAAAAA    T    E       H     H    I
K   K   A     A    T    E       H     H    I
K    K  A     A    T    EEEEEEE H     H   III
```

```
 ssss   l        oooo    ttttt         ddddd   eeeeee   ssss      i     gggg   n    n          ffffff   ttttt  n    n
s       l       o    o     t           d    d  e       s          i    g    g  nn   n          f          t    nn   n
 ssss   l       o    o     t           d    d  eeeee    ssss      i    g       n n  n          fffff      t    n n  n
     s  l       o    o     t           d    d  e            s     i    g  ggg  n  n n   ...    f          t    n  n n
s    s  l       o    o     t           d    d  e       s    s     i    g    g  n   nn   ...    f          t    n   nn
 ssss   llllll   oooo      t   _______ ddddd   eeeeee   ssss      i     gggg   n    n   ...    f          t    n    n
```

```
                     //tera/users/katehi/tape/slot_design.ftn

################                                                            ################
################         LAST MODIFIED ON:  89/01/26  9:09 AM               ################
################             FILE PRINTED:  89/01/26  9:11 AM               ################
################                                                            ################
```

```
C...........................................................................
C                            SLOT_DESIGN.FTN
C      This program designs dielectric covered slot arrays
C
C...........................................................................
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 RCUR,AICUR
      COMPLEX YS,YS1S2,CI,SUM_MD,SUM_MW,SUM_M,CUR_RES,Z_SELF_RES
      COMPLEX YS_ADM,YSW_ADM,CONSTN,CONSTM,Z12_MD,Z12_MW
      EXTERNAL F_EER
C
C...................................................................
C                        COMMENTS
C
C     Input file:    DATA_WAVE_MUTUAL
C     Output file:   OUT_WAVE_MUTUAL
C
C     This program calls the following subroutines:
C
C     DATA   :         This subroutine reads the values of the substrate
C                      parameters from the input file
C     F_EER  :         This subroutine specifies the appropriate normalization
C                      constant
C     GENERATE  :      This subroutine evaluates the resonant length, resonant
C                      conductance and field distribution at resonance.
C     CUBSPL  :        This subroutine segments the slots into subsections
C                      of the  same length starting from the centers
C                      to the end points
C     MUTAL_SLOT  :    This subroutine finds the mutual coupling between
C                      the subsections of one slot and the subsections
C                      of the neighboring slots
C     ARRANGE_SLOT  :  This subroutine combines these interactions between
C                      subsections to get the total mutual coupling
C                      between slots
C
C....................................................................
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
     *CUR_RES(30,60)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
      COMMON/MAN/IBMATR(260,260)
C
      OPEN(UNIT=05,FILE='DATA_WAVE_MUTUAL',STATUS='OLD')
      OPEN(UNIT=06,FILE='OUT_WAVE_MUTUAL',STATUS='OLD')
C
C     Subroutine DATA reads the values of the geometrical
C     parameters
C
      CALL DATA
C
      CALL F_EER
      CI=(0.0,1.0)
      ICUR=1
C
C     ICUR=0  resonant field derived from GENERATE
C     ICUR=1  we assume a form for the resonant field
C
      IF (ICUR.EQ.0) THEN
```

```
C---------------------------------------------------------------
C     Call GENERATE to find the resonant lengths of various
C     slots
C
C      DO 1 I_SLOT=1,NSLOTS
C           CALL GENERATE(I_SLOT)
C  1   CONTINUE
C-----------------------------------------------------------------
      END IF
      DO 7 I_SLOT=1,NSLOTS
           NSL(I_SLOT)=NSL(I_SLOT)+2
           N_MAX=NSL(I_SLOT)-1
           DO 8 N=1,N_MAX
                CUR_RES(I_SLOT,N+1)=CUR_RES(I_SLOT,N)
  8        CONTINUE
           CUR_RES(I_SLOT,1)=(0.0,0.0)
           CUR_RES(I_SLOT,N_MAX+1)=(0.0,0.0)
  7   CONTINUE
C
C     Find common subsection length
C
      N_SLOT=1
      IF (N_SLOT.LE.4) THEN
          I_MIN=1
      ELSE
          I_MIN=N_SLOT-3
      END IF
C
      IF (N_SLOT.GT.(NSLOTS-4)) THEN
          I_MAX=NSLOTS
      ELSE
          I_MAX=N_SLOT+3
      END IF
      DLX=DLX_RES(I_MIN)
      DO 2 I=I_MIN,I_MAX
           IF (DLX_RES(I).LT.DLX) THEN
               DLX=DLX_RES(I)
           END IF
  2   CONTINUE
C
C     Interpolate the current of n_slot
C
      DLX_DIF=DABS(DLX_RES(N_SLOT)-DLX)
      IF (DLX_DIF.GT.1.D-5) THEN
          CALL CUBSPL(ICUR,DLX,1,N_SLOT,1)
          CALL CUBSPL(ICUR,DLX,1,N_SLOT,2)
          DLX_RES(N_SLOT)=DLX
          L_MAX=NSL(N_SLOT)
          DO 9 L=1,L_MAX
               CUR_RES(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
               WRITE (6,77) N_SLOT,L,CUR_RES(N_SLOT,L)
 77            FORMAT(5X,'N_SLOT=',I4,2X,'L=',I4,2X,'CUR=',
    *          E14.7,2X,E14.7)
  9       CONTINUE
      ELSE  IF (ICUR.EQ.1) THEN
          L_MAX=NSL(N_SLOT)
          DO 502 L=1,L_MAX
                 RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                 AICUR(L)=0.0
                 WRITE (6,601) L,RCUR(L),AICUR(L)
 601             FORMAT(2X,'L=',I4,2X,'RCUR(L)=',E14.7,2X,
    *            'AICUR(L)=',E14.7)
 502     CONTINUE
         DO 505 L=1,L_MAX
                CUR_RES(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
 505     CONTINUE
```

```
      END IF
C
C     Interpolate the current on the other slots
C
      DO 33  M=I_MIN,I_MAX
              IF (M.EQ.N_SLOT) GO TO 33
C
C          Interpolate the current of m slot
C
              DLX_DIF=DABS(DLX_RES(M)-DLX)
              IF (DLX_DIF.GT.1.D-5) THEN
                     CALL CUBSPL(ICUR,DLX,1,M,1)
                     CALL CUBSPL(ICUR,DLX,1,M,2)
                     DLX_RES(M)=DLX
                     L_MAX=NSL(M)
                     DO 34 L=1,L_MAX
                          CUR_RES(M,L)=RCUR(L)+CI*AICUR(L)
                          WRITE (6,77) M,L,CUR_RES(M,L)
  34                 CONTINUE
              ELSE IF (ICUR.EQ.1) THEN
                     L_MAX=NSL(M)
                     DO 503 L=1,L_MAX
                            RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                            AICUR(L)=0.0
                            WRITE (6,601) L,RCUR(L),AICUR(L)
 503                 CONTINUE
                     DO 506 L=1,L_MAX
                            CUR_RES(M,L)=RCUR(L)+CI*AICUR(L)
 506                 CONTINUE
             END IF
  33  CONTINUE
C
C
C     Call MUTUAL_SLOT to find mutual coupling between slot
C     n_slot and the neighboring slots
C
      CALL MUTUAL_SLOT(N_SLOT)
C
C
      DO 11 I=I_MIN,I_MAX
            J_MIN=I_MIN+1
            J_MAX=I_MAX
            DO 12 J=J_MIN,J_MAX
                  IJMAX=NSSL(I,J)
                  WRITE (6,13) I,J
  13              FORMAT(10X,'Interactions between slots',I2,' and ',
     *                  I2//)
                  DO 14 IJ=1,IJMAX
                        WRITE (6,15) IJ,YS_ADM(I,J,IJ)
  15                    FORMAT(1X,'IJ=',I4,1X,'YSD=',E14.7,2X,E14.7,
     *                  )
  14              CONTINUE
  12        CONTINUE
  11  CONTINUE
C
      IZ_MIN=48
      IZ_MAX=NXOFF(2)
      IZ_STEP=47
      DO 108 IZ=IZ_MIN,IZ_MAX,IZ_STEP
             NXOFF(2)=IZ
             CALL ARRANGE_MUTUAL
C
C     Find the center of n_slot
C
             NC0=(NSL(N_SLOT)+1)/2
C
```

```
C     Find the corresponding row for IBMATR
C
             I_ROW=0
             DO 20 I=I_MIN,N_SLOT
                   IF (I.GT.1) I_ROW=I_ROW+NSL(I-1)
  20         CONTINUE
C
C     Find the mutual coupling terms due to dielectric ,waveguide
C
             SUM_MD=(0.0,0.0)
             I_COL=0
             DO 3 M=I_MIN,I_MAX
                  IF (M.EQ.N_SLOT) THEN
                      IF (M.GT.1) I_COL=I_COL+NSL(M-1)
                      GO TO 3
                  END IF
C
C          Find the center of m slot
C
                  NCI=(NSL(M)+1)/2
C
C          Find corresponding collumn in IBMATR
C

                  IF (M.GT.1) I_COL=I_COL+NSL(M-1)
C
C          ICUR = 0   : We derive the current from GENERATE
C          ICUR = 1   : We assume a form for the current
C
                  IN_MIN=1
                  IN_MAX=NSL(N_SLOT)
                  DO 4 IN=IN_MIN,IN_MAX
                       CONSTN=CUR_RES(N_SLOT,IN)/CUR_RES(N_SLOT,NC0)
C
C                      WRITE (6,88) N_SLOT,IN,CONSTN
C  88                  FORMAT(2X,'N=',I4,2X,'IN=',I4,5X,'CONSTN=',
C     *                E14.7,2X,E14.7//)
C
                       IM_MIN=1
                       IM_MAX=NSL(M)
                       DO 5 IM=IM_MIN,IM_MAX
                            CONSTM=CUR_RES(M,IM)/CUR_RES(M,NCI)
                            CON=CONSTN*CONSTM
                            IJ=I_ROW+IN
                            KJ=I_COL+IM
                            IK=IBMATR(IJ,KJ)
C
C                     WRITE (6,89) IM,CONSTM,IJ,KJ,IK
C  89                 FORMAT(10X,'IM=',I4,2X,'CONSTM=',E14.7,2X,E14.7/
C     *               10X,'IJ=',I4,2X,'KJ=',I4,2X,'IK=',I4)
C
                           SUM_MD=SUM_MD+SNGL(CON)*YS_ADM(N_SLOT,M,IK)
  5                    CONTINUE
  4               CONTINUE
  3          CONTINUE
             DIST_X=(NXOFF(2)-NXOFF(1))*DLX
C
             WRITE (6,52) DIST_X
  52         FORMAT(///2X,'LONGITUDINAL DISTANCE =', E14.7/)
             WRITE (6,60) SUM_MD
 60          FORMAT (/10X,'SUM_MD=',E14.7,5X,E14.7/)
C
             Z12_MD=-(120.0*SNGL(PI))**2*SUM_MD/2.0
             WRITE (6,61) Z12_MD
  61         FORMAT(/10X,'Z12_MD=',E14.7,2X,E14.7//)
C
```

```
 108  CONTINUE
 1000 CONTINUE
      STOP
      END
C...............................................................
C     The name of this subroutine is        DATA
C     and gives all the data used by the main program and the other
C     subroutines.
C................................................................
      SUBROUTINE DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION WORK(7,7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
     *CUR_RES(30,60)
C
      PI=3.141592653589D0
C
C
C     ---- Dielectric constant ---
      READ (5,1) ER
  1   FORMAT (///6X,D16.9)
      WRITE (6,2) ER
  2   FORMAT(10X,'Dielectric Constant of the Substrate'/10X,E14.7//)
C
C     ---- Substrate Thickness ---
C
      READ (5,1) H
      WRITE (6,3) H
  3   FORMAT(10X,'Substrate Thickness'/10X,E14.7//)
C
C     ---- Conductor Thickness ---
C
      READ (5,1) T
      WRITE (6,4) T
  4   FORMAT(10X,'Conductor Thickness'/10X,E14.7//)
C
C     ---- Number of Slots ----
C
      READ (5,20) NSLOTS
  20  FORMAT(///6X,I4)
      WRITE (6,6) NSLOTS
   6  FORMAT(10X,'Number of Slots'/10X,'NSLOTS=',I4//)
C
C     ---- Limit for offsets: Small Offset< OFFLIM ----
C                             Large Offset> OFFLIM
C
      OFFLIM=0.1
C
C     ---- Transverse Offsets of the Slots ----
C
      READ(5,1) YOFF(1)
      WRITE (6,7) YOFF(1)
  7   FORMAT(10X,'Transverse Offsets of the Slots'/10X,
     *'YOFF(1)=',E14.7)
```

```
      IF (NSLOTS.GT.1) THEN
             DO 8 I=2,NSLOTS
                  READ(5,10) YOFF(I)
 10               FORMAT(6X,D14.7)
                  WRITE (6,9) I,YOFF(I)
  9               FORMAT(10X,'YOFF(',I2,')=',E14.7)
  8          CONTINUE
      END IF
      WRITE(6,60)
  60  FORMAT(10X,//)
C
C     ---- Longitudinal Offsets of the Slots ( in dlx )
C
      READ (5,20) NXOFF(1)
      WRITE (6,11) NXOFF(1)
  11  FORMAT(10X,'Longitudinal Offset of the Slots'/
     *10X,'NXOFF(1)=',I4)
      IF (NSLOTS.GT.1) THEN
             DO 12 I=2,NSLOTS
                   READ(5,30) NXOFF(I)
  30               FORMAT(6X,I4)
                   WRITE (6,13) I,NXOFF(I)
  13               FORMAT(10X,'NXOFF(',I2,')=',I4)
  12         CONTINUE
      END IF
      WRITE(6,60)
C
C     ---- Slot Widths ----
C
      READ(5,1) WS(1)
      WRITE(6,14) WS(1)
  14  FORMAT(10X,'Slot Widths'/10X,'WS(1)=',E14.7)
      IF (NSLOTS.GT.1) THEN
             DO 15 I=2,NSLOTS
                   READ(5,10) WS(I)
                   WRITE (6,16) I,WS(I)
  16               FORMAT(10X,'WS(',I2,')=',E14.7)
  15         CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Slots Excess Widths ----
C
      READ(5,1) WSDELTA(1)
      WRITE (6,17) WSDELTA(1)
  17  FORMAT(10X,'Slots Excess Widths'/10X,'WSDELTA=',
     *E14.7)
      IF (NSLOTS.GT.1) THEN
             DO 18 I=2,NSLOTS
                   READ(5,10) WSDELTA(I)
                   WRITE(6,19) I,WSDELTA(I)
  19               FORMAT(10X,'WSDELTA(',I2,')=',E14.7)
  18         CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Subsection Length ----
C
      READ (5,1) DLX_RES(1)
      WRITE (6,21) DLX_RES(1)
  21  FORMAT(10X,'Subsection Length'/10X,E14.7//)
      IF (NSLOTS.GT.1) THEN
            DO 40 I=2,NSLOTS
                  READ(5,10) DLX_RES(I)
                  WRITE(6,46) I,DLX_RES(I)
  46              FORMAT(10X,'DLX_RES(',I2,')=',E14.7)
```

```
 40         CONTINUE
      END IF
      WRITE (6,60)
C
C
C     ---- Lower Limit of the Tail Contribution ----
C
      READ (5,1) A
      WRITE (6,22) A
 22   FORMAT(10X,'Lower Limit of Tail Contribution'/10X,E14.7//)
C
C     ---- Number of Points on Each Slot ----
C
      READ (5,20) NSL(1)
      WRITE (6,23) NSL(1)
 23   FORMAT(10X,'Number of Points on Each Slot including the ends',
     */10X,'NSL(1)=',I4)
      IF (NSLOTS.GT.1) THEN
            DO 24 I=2,NSLOTS
                  READ(5,25) NSL(I)
 25               FORMAT(6X,I4)
                  WRITE(6,26) I,NSL(I)
 26               FORMAT(10X,'NSL(',I2,')=',I4)
 24         CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Error in the evaluation of the series ----
C
      READ (5,1) ERROR
      WRITE (6,27) ERROR
 27   FORMAT(10X,'Error in the evaluation of the series'/
     *10X,'ERROR=',E14.7//)
C
C     Initialize OFFSET( ) to 0
C
      DO 37 I=1,7
            OFFSET(I)=0.D0
 37   CONTINUE
C
C     Initialize NOFF to 1
C
      NOFF=1
      RETURN
      END
C----------------------------------------------------------------------
C     This function evaluates the normalization constant
C----------------------------------------------------------------------
      SUBROUTINE F_EER
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---- Normalization Constant ----
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C      EER=ER+(1.D0-ER)*(W/H)/(1.D0+W/H)
C
      EER=1.0
      WRITE (6,100) EER
      WRITE(*,100) EER
100   FORMAT(10X,'Normalization Constant'/10X,E14.7/)
      RETURN
      END
```

```
C---------------------------------------------------------------
C      This subroutine denormalizes with respect to CNORM_OLD
C      and normalizes again with respect to  CNORM_NEW
C---------------------------------------------------------------
      SUBROUTINE NORM(CNORM_OLD,CNORM_NEW)
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      CNORM=CNORM_OLD/CNORM_NEW
C
      PI=3.141592654
C
      AK0=2.D0*PI*CNORM_NEW
      AKK=2.D0*PI
      AK=AK0*DSQRT(ER)
C
      H=H*CNORM
      AW=AW*CNORM
      BW=BW*CNORM
      T=T*CNORM
      DLX=DLX*CNORM
      OFFLIM=OFFLIM*CNORM
C
      YOFF(1)=YOFF(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 8 I=2,NSLOTS
                  YOFF(I)=YOFF(I)*CNORM
  8          CONTINUE
      END IF
C
      WS(1)=WS(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 15 I=2,NSLOTS
                   WS(I)=WS(I)*CNORM
  15         CONTINUE
      END IF
C
      WSDELTA(1)=WSDELTA(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 18 I=2,NSLOTS
                   WSDELTA(I)=WSDELTA(I)*CNORM
  18         CONTINUE
      END IF
      RETURN
      END
C**********************************************************************
C.......................... Spline Interpolation ......................
C**********************************************************************
      SUBROUTINE  CUBSPL(ICUR,DLX,IEND,N_SLOT,IRX)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX CURRENT,CUR_RES,Z_SELF_RES,CC
      REAL*4 RCUR,AICUR,REAL_CUR,AIMAG_CUR
      DIMENSION S(260),A(260,4),X(260),Y(260),AI(260),BI(260),
     *CI(260),DI(260)
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
```

```
     *CUR_RES(30,60)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
C     This routine computes the matrix for finding the coefficients of a
C     cubic spline through a set of data.
C     The  system is then solved to obtain the second derivative values,
C     and the coefficients of the cubic spline between each pair of points.
C       -----------------------------------------------------------------------
C      Parameters are
C      X,Y       Arrays of X and Y values to be fitted
C
C      DLX        Subsection length (if all points have same spacing)
C
C      S         Array of second derivative values at the points
C
C      N         Number of points
C
C      IEND      Type of end condition to be used
C                    IEND=1, Linear ends, S(1)=S(N)=0
C                    IEND=2, Parabolic ends, S(1)=S(2), S(N)=S(N-1)
C                    IEND=3, Cubic ends S(1),S(N) are extrapolated
C
C      A          Augmented matrix of coefficients and R.H.S. for finding S
C
C      IRX       1  : Interpolate the real part of the current
C                2  : Interpolate the imaginary part of the current
C
C      ICUR     =0  resonant field derived from GENERATE
C      ICUR     =1  we assume a form for the resonant field
C      ------------------------------------------------------------------------
       PI=3.141592654
       N=NSL(N_SLOT)
       CC=(0.0,1.0)
C
C      Computation of matrices X,Y
C
       NC0_OLD=(NSL(N_SLOT)+1)/2
       NSLOT_NEW=2*NINT((NSL(N_SLOT)-1)*DLX_RES(N_SLOT)/(2.0*DLX))+1
       NC0_NEW=(NSLOT_NEW+1)/2
       ITEST=(NSLOT_NEW+1)-NC0_NEW*2
       I_CUR=(NSLOT_NEW+1)/2
       I_MIN=1
       I_MAX=NSL(N_SLOT)
       L_MAX=I_MAX
       IF (ICUR.EQ.1) GO TO 500
        DO 1 I=I_MIN,I_MAX
             X(I)=DLX_RES(N_SLOT)*FLOAT(I-1)
             REAL_CUR=REAL(CUR_RES(N_SLOT,I))
             CURRENT=-CC*CUR_RES(N_SLOT,I)
             AIMAG_CUR=REAL(CURRENT)
             IF  (IRX.EQ.1) Y(I)=DBLE(REAL_CUR)
             IF  (IRX.EQ.2) Y(I)=DBLE(AIMAG_CUR)
             WRITE (6,67) I,X(I),Y(I)
 67          FORMAT(10X,'I=',I4,2X,'X=',E14.7,2X,'Y=',E14.7)
  1     CONTINUE
C     -------------------------------------------------------------------------
C     | Compute the N-2 rows |
C     ------------------------
C
        NM2=N-2
        NM1=N-1
        DX1=X(2)-X(1)
        DY1=(Y(2)-Y(1))/DX1*6.D0
        DO 10 I=1,NM2
          DX2=X(I+2)-X(I+1)
```

```
          DY2=(Y(I+2)-Y(I+1))/DX2*6.D0
          A(I,1)=DX1
          A(I,2)=2.D0*(DX1+DX2)
          A(I,3)=DX2
          A(I,4)=DY2-DY1
          DX1=DX2
          DY1=DY2
10      CONTINUE
C
C    Adjust first and last rows to end condition
C
        GO TO  (20,50,80), IEND
C
C    for IEND = 1 no change is needed
C
20      GO TO 100
C
C    for IEND = 2, S(1)=S(2), S(N)=S(N-1), parabolic ends.
C
50      A(1,2)=A(I,2)+X(2)-X(1)
        A(NM2,2)=A(NM2,2)+X(N)-X(NM1)
        GO TO 100
C
C    for IEND = 3, cubic ends, S(1), S(N) are extrapolated.
C
80      DX1=X(2)-X(1)
        DX2=X(3)-X(2)
        A(1,2)=(DX1+DX2)*(DX1+2.D0*DX2)/DX2
        A(1,3)=(DX2*DX2-DX1*DX1)/DX2
        DXN2=X(NM1)-X(NM2)
        DXN1=X(N)-X(NM1)
        A(NM2,1)=(DXN2*DXN2-DXN1*DXN1)/DXN2
        A(NM2,2)=(DXN1+DXN2)*(DXN1+2.D0*DXN2)/DXN2
        GO TO 100
C
C    Now we solve the tridiagonal system.  First reduce
C
100     DO 110 I=2,NM2
          A(I,2)=A(I,2)-A(I,1)/A(I-1,2)*A(I-1,3)
          A(I,4)=A(I,4)-A(I,1)/A(I-1,2)*A(I-1,4)
110     CONTINUE
C
C    Back substitution
C
        A(NM2,4)=A(NM2,4)/A(NM2,2)
        DO 120 I=2,NM2
          J=NM1-I
          A(J,4)=(A(J,4)-A(J,3)*A(J+1,4))/A(J,2)
120     CONTINUE
C
C     Place values in S-vector
C
        DO 130 I=1,NM2
          S(I+1)=A(I,4)
130     CONTINUE
C
C     Set S(1) and S(N) according to end conditions
C
        GO TO (150,160,170), IEND
C
C     Linear ends
C
150     S(1)=0.
        S(N)=0.
        GO TO 200
C
```

```
C     Parabolic ends
C
160     S(1)=S(2)
        S(N)=S(N-1)
        GO TO 200
C
C     For cubic ends
C
170     S(1)=((DX1+DX2)*S(2)+DX1*S(3))/DX2
        S(N)=((DXN2+DXN1)*S(NM1)-DXN1*S(NM2))/DXN2
C
C     Find spline fit coefficients
C
C
C     Evaluation of the coefficients ai,bi,ci,di - Store into AI,BI
C     CI,DI
C
200      DO 210 I=1,NM1
           AI(I)=(S(I+1)-S(I))/(6.D0*DLX_RES(N_SLOT))
           BI(I)=S(I)/2.D0
           CI(I)=(Y(I+1)-Y(I))/DLX_RES(N_SLOT)-(2.D0*S(I)+S(I+1))
     *             *DLX_RES(N_SLOT)/6.D0
210        DI(I)=Y(I)
C
C      Re-evaluate nsl(n_slot) and cur_res(n_slot)
C
      DO 2 I=1,I_CUR
           IF (ITEST.EQ.0) NCP=NC0_NEW+I-1
           IF (ITEST.EQ.1) NCP=NC0_NEW+I
           NCM=NC0_NEW-I+1
           DISTP=(NCP-1)*DLX
           DISTM=(NCM-1)*DLX
           RIP=DISTP/DLX_RES(N_SLOT)
           IP=INT(RIP)
           IF ((RIP-IP).GT.0.999) IP=IP+1
           IF(IP.EQ.NSLOT_NEW) IP=IP-1
           RIM=DISTM/DLX_RES(N_SLOT)
           IM=INT(RIM)
           IF((RIM-IM).GT.0.999) THEN
                IM=IM+1
           END IF
           DIFP=DISTP-FLOAT(IP)*DLX_RES(N_SLOT)
           DIFM=DISTM-FLOAT(IM)*DLX_RES(N_SLOT)
           DIFP2=DIFP*DIFP
           DIFM2=DIFM*DIFM
           DIFP3=DIFP2*DIFP
           DIFM3=DIFM2*DIFM
           IF (IRX.EQ.1) THEN
              IP=IP+1
              IM=IM+1
              RCUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                  CI(IP)*DIFP+DI(IP))
              RCUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                  CI(IM)*DIFM+DI(IM))
              WRITE (6,666) NCP,IP,NCM,IM,RCUR(NCP),RCUR(NCM)
 666          FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *        I4/30X,'RCUR(NCP)=',E14.7,2X,'RCUR(NCM)=',E14.7)
           END IF
           IF (IRX.EQ.2) THEN
              IP=IP+1
              IM=IM+1
              AICUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                   CI(IP)*DIFP+DI(IP))
              AICUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                   CI(IM)*DIFM+DI(IM))
              WRITE (6,777) NCP,IP,NCM,IM,AICUR(NCP),AICUR(NCM)
```

```
 777          FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *        I4/10X,'AICUR(NCP)=',E14.7,2X,'AICUR(NCM)=',E14.7)
          END IF
  2   CONTINUE
C
      IF (IRX.EQ.2) NSL(N_SLOT)=NSLOT_NEW
      RETURN
C
 500  CONTINUE
      DO 502 L=1,L_MAX
             IF (IRX.EQ.1) THEN
                  RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                  WRITE (6,601) L,RCUR(L)
 601              FORMAT(10X,'L=',I4,2X,'RCUR=',E14.7)
             ELSE IF (IRX.EQ.2) THEN
                  AICUR(L)=0.0
                  WRITE (6,602) L,AICUR(L)
 602              FORMAT(2X,'L=',I4,2X,'AICUR=',E14.7)
             END IF
 502     CONTINUE
      RETURN
      END
```

```
################                                                              ################
################                   apollo domain                              ################
################                                                              ################
################                    CAEN/Apollo                               ################
```

```
K    K     A    TTTTTTT EEEEEEE H     H  III
K   K     A A      T    E       H     H   I
K  K     A   A     T    E       H     H   I
KKK     A     A    T    EEEEE   HHHHHHH   I
K  K    AAAAAAA    T    E       H     H   I
K   K   A     A    T    E       H     H   I
K    K  A     A    T    EEEEEEE H     H  III
```

```
m    m  u    u  ttttt  u    u   aa    l               ssss   l       oooo    ttttt         ffffff  ttttt  n    n
mm  mm  u    u    t    u    u  a  a   l              s       l      o    o     t           f         t    nn   n
m mm m  u    u    t    u    u a    a  l               ssss   l      o    o     t           fffff     t    n n  n
m    m  u    u    t    u    u aaaaaa  l                   s  l      o    o     t     ...   f         t    n  n n
m    m  u    u    t    u    u a    a  l              s    s  l      o    o     t     ...   f         t    n   nn
m    m   uuuu     t     uuuu  a    a  llllll _______  ssss   llllll  oooo      t     ...   f         t    n    n
```

```
                          //tera/users/katehi/tape/mutual_slot.ftn

###############                                                               ###############
###############            LAST MODIFIED ON:  88/10/25  2:59 PM               ###############
###############                FILE PRINTED:  89/01/25  1:34 PM               ###############
###############                                                               ###############
```

```
C...........................................................................
C                              MUTUAL_SLOT.FTN
C         This program evaluates the mutual coupling terms for the second
C                                 design equation
C...........................................................................
      SUBROUTINE MUTUAL_SLOT(N_SLOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YS,YS1S2,CI
      COMPLEX YS_ADM,YSW_ADM
      EXTERNAL F_EER
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/OUT/GS(250),GS1S2(7,250)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/B01/BJ0,BJ1
C
      COMMON/MAN/IBMATR(260,260)
C
C     Subroutine DATA_MUTUAL_SLOT prepares the parameters for the
C     evaluation of the mutual coupling terms
C
      CALL DATA_MUTUAL_SLOT(N_SLOT)
C
      CNORM_OLD=1.D0
      CNORM_NEW=1.D0/DSQRT(EER)
      CALL NORM(CNORM_OLD,CNORM_NEW)
C
C     Subroutine YIJ_DIEL evaluates the contribution to the elements
C     of the admittance matrix coming from the dielectric substrate
C
      CALL YIJ_DIEL
C
      CNORM_OLD=1.D0/DSQRT(EER)
      CNORM_NEW=1.D0
      CALL NORM(CNORM_OLD,CNORM_NEW)
      RETURN
      END
C............................................................
C     The name of this subroutine is       DATA_MUTUAL_SLOT
C.............................................................
```

```fortran
      SUBROUTINE DATA_MUTUAL_SLOT(N_SLOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION WORK(7,7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      IF (N_SLOT.LE.4) THEN
          I_MIN=1
      ELSE
          I_MIN=N_SLOT-3
      END IF
C
      IF (N_SLOT.GT.(NSLOTS-4)) THEN
          I_MAX=NSLOTS
      ELSE
          I_MAX=N_SLOT+3
      END IF
C
C     ---- Evaluation of the Elements for the Mutual Interactions
C
      WRITE (6,70)
  70  FORMAT(///10X,'Number of elements to be evaluated for',
     *' the mutual interactions'/)
          DO 28 I=I_MIN,I_MAX
                J_MIN=I+1
                J_MAX=I_MAX
                DO 29 J=J_MIN,J_MAX
                      WORK(I,J)=ABS(YOFF(I)-YOFF(J))
                      NSSL(I,J)=NXOFF(J)+(NSL(J)-1)/2-(NXOFF(I)-
     *                         (NSL(I)+1)/2)
                      WRITE (6,50) I,J,NSSL(I,J),WORK(I,J)
  50                  FORMAT(10X,'I=',I4,2X,'J=',I4,2X,'NSSL=',I4,
     *                E14.7)
  29            CONTINUE
  28      CONTINUE
C
C     ---- Evaluation of the offsets for the dielectric ----
C
      IJ=1
      OFFSET(1)=DABS(YOFF(I_MAX)-YOFF(I_MIN))
      NOFFS(1)=NSSL(I_MIN,I_MAX)
      DO 31 I=I_MIN,I_MAX
            J_MIN=I+1
            J_MAX=I_MAX
            DO 32 J=J_MIN,J_MAX
                  TEST=WORK(I,J)
                  IMIN=I
                  JMIN=J
                  DO 33 L=I_MIN,I_MAX
                        K_MIN=L+1
                        K_MAX=I_MAX
                        DO 34 K=K_MIN,K_MAX
                              IF (TEST.GT.WORK(L,K)) THEN
                                  TEST=WORK(L,K)
                                  IMIN=L
                                  JMIN=K
                              END IF
   34                   CONTINUE
   33             CONTINUE
```

```
                  DO 35 N=1,IJ
                        IF (TEST.EQ.OFFSET(N)) THEN
                            INSS(IMIN,JMIN)=N
                            IF(NOFFS(N).LT.NSSL(IMIN,JMIN)) THEN
                                   NOFFS(N)=NSSL(IMIN,JMIN)
                            END IF
                            GO TO 36
                        END IF
   35             CONTINUE
                  IJ=IJ+1
                  OFFSET(IJ)=TEST
                  INSS(IMIN,JMIN)=IJ
                  NOFFS(IJ)=NSSL(IMIN,JMIN)
   36             WORK(IMIN,JMIN)=100
   32       CONTINUE
   31 CONTINUE
C
      WRITE (6,80)
  80  FORMAT(///10X,'Offsets for the dielectric layer and number',
     *' of corresponding elements'/)
      DO 51 I=I_MIN,I_MAX
            WRITE (6,52) I,OFFSET(I),NOFFS(I)
  52        FORMAT(10X,'I=',I4,2X,'OFFSET=',E14.7,2X,'NOFFS=',I4)
  51  CONTINUE
      WRITE (6,90)
  90  FORMAT (///10X,'SLOTS and corresponding offsets in the dielectric'
     *,/)
      DO 53 I=I_MIN,I_MAX
            J_MIN=I+1
            J_MAX=I_MAX
            DO 54 J=J_MIN,J_MAX
                  IJ=INSS(I,J)
                  WRITE(6,55) I,J,INSS(I,J),OFFSET(IJ)
  55              FORMAT(10X,'I=',I4,2X,'J=',I4,2X,'INSS=',I4,
     *            2X,'OFFSET=',E14.7/)
  54        CONTINUE
  53  CONTINUE
C
C     ---- Evaluation of the Max Number of Offsets for the Diel. ----
C
      NOFF=IJ
C
      WRITE (6,56) NOFF
  56  FORMAT(///10X,'Max number of offsets in the dielectric'/
     *10X,'NOFF=',I4//)
C
      RETURN
      END
```

```
*** WARNING ***
1 line was too long and was truncated.
*** WARNING ***
```

```
################              a p o l l o   d o m a i n               ################
################                                                      ################
################                   CAEN/Apollo                        ################
################                                                      ################
```

```
K    K     A    TTTTTTT EEEEEEE H     H   III
K   K     A A      T    E       H     H    I
K  K     A   A     T    E       H     H    I
KKK     A     A    T    EEEEE   HHHHHHH    I
K  K    AAAAAAA    T    E       H     H    I
K   K   A     A    T    E       H     H    I
K    K  A     A    T    EEEEEEE H     H   III
```

```
ppppp    oooo   l      eeeeee   ssss          m    m  u     u   ttttt  u     u    aa    l             ffffff  ttttt  n    n
p    p  o    o  l      e       s              mm  mm  u     u     t    u     u   a  a   l             f         t    nn   n
p    p  o    o  l      eeeee    ssss          m mm m  u     u     t    u     u  a    a  l             fffff     t    n n  n
ppppp   o    o  l      e            s         m    m  u     u     t    u     u  aaaaaa  l        ...  f         t    n  n n
p       o    o  l      e       s    s         m    m  u     u     t    u     u  a    a  l        ...  f         t    n   nn
p        oooo   llllll eeeeee   ssss  ______  m    m   uuuu       t     uuuu  a    a  llllll   ...  f         t    n    n
```

```
                     //tera/users/katehi/tape/poles_mutual.ftn

################                                                      ################
################          LAST MODIFIED ON:  88/10/25  2:59 PM        ################
################              FILE PRINTED:  89/01/25  1:35 PM        ################
################                                                      ################
```

```
C***********************************************************************
C     The name of this file is .......... POLES_MUTUAL.FTN ...............
C***********************************************************************
      SUBROUTINE SPOLES
      IMPLICIT REAL*8(A-H,O-Z)
C.......................................................................
C
C     ER    :....Dielectric constant
C
C     H     :....Height of the dielectric substrate
C
C     NE    :....Number of TE surface waves
C
C     NM    :....Number of tm surface waves
C
C     XS    :....Matrix of poles contributing to TE surface waves
C
C     XR    :....Matrix of poles contributing to TM surface waves
C
C     ERR   :....Error in the computation of the poles
C
C.......................................................................
      DIMENSION XS(40),XR(40),LOR(40)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0_GENER,
     *AK_GENER,AKK_GENER,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),TMP(20),TEP(20),
     *AM(41),DM(41),TPO(40),VXXM(20),VZXM(20),VZXE(20),BPOINT(10),
     *BCOAL(10),MPOINT,NPOINT,NK0,MA,NM,NE,NK0K,IFIRST
C
      AER=DSQRT(EER)
      ER2=ER*ER
      PI2=PI*PI
      MAXE=5
      ERR=0.0000001D0
      DP=H/AER
C---------------------------------------------
C     PART I : TE MODES
C---------------------------------------------
      AK0=2.D0*PI
      AK=DSQRT(ER)*AK0
      X0=DP*DSQRT(AK**2-AK0**2)
C-----------------------------------------------------------
C     WRITE (6,300) AK0,AK,X0,PI
C 300 FORMAT(10X,'AK0=',E14.7,2X,'AK=',E14.7,2X,'X0=',E14.7,
C    *2X,'PI=',E14.7/)
C----------------------------------------------------------
      AN=X0/PI+0.5D0
      NE=AN
      IF (NE.EQ.0) GO TO 310
      DO 2 I=1,NE
           IF (X0-(2.D0*FLOAT(I)+1.D0)*PI/2.D0) 3,3,4
  4        XS0=(2.D0*FLOAT(I)-1.D0)*PI/2.D0+ERR
           XS1=(2.D0*FLOAT(I)+1.D0)*PI/2.D0-ERR
           GO TO 5
  3        XS0=(2.D0*FLOAT(I)-1.D0)*PI/2.D0+ERR
           XS1=X0
  5        CONTINUE
           IF (DABS(XS0-XS1)-ERR) 22,7,7
  7        XSA=(XS0+XS1)/2.D0
           Y=-DTAN(XSA)*DSQRT(X0**2-XSA**2)-XSA
           IF (Y) 8,9,10
  9        XS(I)=XSA
```

```
           GO TO 222
   8       XS1=XSA
           GO TO 5
  10       XS0=XSA
           GO TO 5
  22       XS(I)=(XS0+XS1)/2.D0
 222       XS(I)=DSQRT(AK**2-XS(I)**2/DP**2)
   2  CONTINUE
C---------------------------------------------------------------
      WRITE (6,301) ER,H
301   FORMAT(//10X,' Dielectric Constant=',D16.9/10X,'Substrate '
     *,'Thickness',D16.9///)
C----------------------------------------------------------------
 310  IF (NE.EQ.0) WRITE (6,304)
 304  FORMAT(/////10X,'No TE waves excited in the substrate'//)
      IF (NE.EQ.0) GO TO 312
      IF (NE.GT.0) WRITE (6,305) NE
 305  FORMAT(///10X,'There are',I4,
     *' TE waves excited in the substrate'//)
      DO 302 I=1,NE
             TEP(I)=XS(I)/AER
             IF (I.GT.1) THEN
                 I_MAX=I-1
                 DO 502 I_I=1,I_MAX
                        TEP_MIN=TEP(I_I)
                        IF (TEP(I).LT.TEP(I_I)) THEN
                            TEP(I_I)=TEP(I)
                            TEP(I)=TEP_MIN
                        END IF
 502             CONTINUE
             END IF
             DO 503 I=1,NE
                    WRITE (6,303) I,TEP(I)
 303                FORMAT (10X,I4,2X,D16.9)
 503         CONTINUE
 302  CONTINUE
 312  CONTINUE
C
C     END OF PART I
C
C--------------------------------------------
C     PART II : TM MODES
C--------------------------------------------
      AN=X0/PI+1.D0
      NM=AN
      DO 13 I=1,NM
            IF (X0-(2.D0*FLOAT(I)+1.D0)*PI/2.D0) 14,14,15
  15        XS1=FLOAT(I)*PI-PI/3.D0-0.01D0
            GO TO 16
  14        XS1=X0
  16        XS0=FLOAT(I-1)*PI+ERR
  17        CONTINUE
            IF (DABS(XS0-XS1)-ERR) 113,19,19
  19        XRA=(XS0+XS1)/2.D0
C--------------------------------------------
C           WRITE (6,301) XRA
C 301       FORMAT(10X,'XRA=',E14.7/)
C--------------------------------------------
            Y=DSQRT(ER)**2*(1.D0/DTAN(XRA))*DSQRT(X0**2-XRA**2)-XRA
            IF (Y) 20,21,24
  21        XR(I)=XRA
            GO TO 333
  20        XS1=XRA
            GO TO 17
  24        XS0=XRA
            GO TO 17
```

```
 113        XR(I)=(XS0+XS1)/2.D0
 333        XR(I)=DSQRT(AK**2-XR(I)**2/DP**2)
  13  CONTINUE
      WRITE (6,307) NM
 307  FORMAT(///10X,'There are',I4,' TM waves excited in the substrate'/
     */)
      DO 308 I=1,NM
             TMP(I)=XR(I)/AER
             IF (I.GT.1) THEN
                 I_MAX=I-1
                 DO 508 I_I=1,I_MAX
                        TMP_MIN=TMP(I_I)
                        IF (TMP(I).LT.TMP(I_I)) THEN
                            TMP(I_I)=TMP(I)
                            TMP(I)=TMP_MIN
                        END IF
 508             CONTINUE
             END IF
             DO 509 I=1,NM
                    WRITE (6,306) I,TMP(I)
 306                FORMAT (10X,I4,2X,D16.9)
 509         CONTINUE
 308  CONTINUE
 322  CONTINUE
C
      NK=NE+NM
      IF (NE.EQ.0) GO TO 350
      DO 411 IQW=1,NE
             TPO(IQW)=TEP(IQW)
             LOR(IQW)=1
  411 CONTINUE
  350 CONTINUE
      DO 412 IQW=1,NM
             TPO(NE+IQW)=TMP(IQW)
             LOR(NE+IQW)=0
  412 CONTINUE
C
      IF (NK.EQ.1) GO TO 416
      NNK=NK-1
      DO 415 IIP=1,NNK
             IK=IIP+1
             DO 413 IIF=IK,NK
                    QWR=TPO(IIP)
                    IIW=LOR(IIP)
                    IF (TPO(IIP).LT.TPO(IIF)) GO TO 413
                    TPO(IIP)=TPO(IIF)
                    LOR(IIP)=LOR(IIF)
                    TPO(IIF)=QWR
                    LOR(IIF)=IIW
  413        CONTINUE
  415 CONTINUE
      IF (LOR(1).EQ.0) IFIRST=0
      IF (LOR(1).EQ.1) IFIRST=1
      GO TO 417
C
  416 IFIRST=2
  417 CONTINUE
      RETURN
      END
```

```
*** WARNING ***
7 lines were too long and were truncated.
*** WARNING ***
```

```
################                                                   ################
################            a p o l l o   d o m a i n              ################
################                                                   ################
################                   CAEN/Apollo                     ################



                    K    K     A    TTTTTTT EEEEEEE H     H   III
                    K   K     A A      T    E       H     H    I
                    K  K     A   A     T    E       H     H    I
                    KKK     A     A    T    EEEEE   HHHHHHH    I
                    K  K    AAAAAAA    T    E       H     H    I
                    K   K   A     A    T    E       H     H    I
                    K    K  A     A    T    EEEEEEE H     H   III



y   y   i       j        ddddd   i   eeeeee l          m   m  u   u  ttttt  u   u   aa   l         ffffff
 y y    i       j        d   d   i   e      l          mm mm  u   u    t    u   u  a  a  l         f
  y     i       j        d   d   i   eeeee  l          m mm m u   u    t    u   u a    a l         fffff
  y     i       j        d   d   i   e      l          m   m  u   u    t    u   u aaaaaa l     ... f
  y     i   j   j        d   d   i   e      l          m   m  u   u    t    u   u a    a l     ... f
  y     i    jjjj ______ ddddd   i   eeeeee llllll ______ m   m   uuuu    t     uuuu a    a llllll ... f



                         //tera/users/katehi/tape/yij_diel_mutual.ftn

###############                                                    ###############
###############            LAST MODIFIED ON:  88/10/25  2:59 PM    ###############
###############                FILE PRINTED:  89/01/25  1:36 PM    ###############
###############                                                    ###############
```

```
C*******************************************************************
C     ...................  YIJ_DIEL_MUTUAL.FTN   .................................
C
C     This program evaluates the part of the elements of the admittance
C     matrix coming from th dielectric substrate.
C     This program is good for any substrate thickness h, er and
C     any dimensions of the slot.
C
C
C
C
C
C
C*******************************************************************
      SUBROUTINE YIJ_DIEL
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 CONST,GSK,GS1S2K
      COMPLEX YS,YS1S2,CI,YS_ADM,YSW_ADM
      DIMENSION MOFFS(7)
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/OUT/GS(250),GS1S2(7,250)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/B01/BJ0,BJ1
C
      W=WS(1)
      WDELTA=WSDELTA(1)
      WREAL=W
      W=W*(1.D0+2.D0*WDELTA/W)
C
C     Subroutine POLES evaluates the poles of the Green's function
C     and orders them according to their magnitude
C
      CALL SPOLES
C
C     This subroutines gives data for the numerical integration
C
      CALL DATA_SLOT
```

```
C
      CI=(0.00,1.00)
C
      DO 1 I=1,NOFF
           MOFFS(I)=NOFFS(I)
           IF (NOFFS(I).GT.200) NOFFS(I)=200
           NMAX(I)=NOFFS(I)+2
 1    CONTINUE
C
C
C
      ADL=AKK*DLX
      YSIN=DSIN(ADL)
      YCOS=DCOS(ADL)
C
C     For the normalization of the current along the y axis
C
                  CVON=W*PI/2.D0
C
C     Computation of lamda-integration limits between 0 and A
C
C
C
      CALL LIMIT
C
C     Evaluation of the Green's function at different points
C     in the interval [0,A]. The Bessel function has been excluded
C
      CALL GREEN
C
C     Evaluation of the tail contribution (from a to infinity)
C
      CALL TAIL
C
      CONST=-(1.D0/CVON)*DSQRT(EER)/(480.D0*(PI**3)*YSIN*YSIN)
      WRITE(6,10)
 10   FORMAT(///10X,'Contribution to admittance from the dielectric'///)
      KMAX=MOFFS(1)
      DO 2 K=1,KMAX
           YS(K)=YS(K)*CONST
           GSK=REAL(GS(K))*CONST
C          WRITE (6,11) K,YS(K),GSK
 11        FORMAT(1X,I4,2X,'YS=',E14.7,2X,E14.7,2X,
     *             'GSK=',E14.7)
           YS(K)=(YS(K)+GSK)*CI
C          WRITE (6,12) K,YS(K)
 12        FORMAT(5X,I4,5X,'YS=',E14.7,2X,E14.7)
 2    CONTINUE
      DO 3 I=1,NOFF
           WRITE(6,13) I
 13        FORMAT(///5X,'OFFSET #',I4///)
           KMIN=I+1
           KMAX=MOFFS(I)
           DO 4 K=KMIN,KMAX
                YS1S2(I,K)=YS1S2(I,K)*CONST
                GS1S2K=REAL(GS1S2(I,K))*CONST
C               WRITE(6,14) K,YS1S2(I,K),GS1S2K
 14             FORMAT(1X,I4,2X,'YS1S2=',(E14.7,2X,E14.7),
     *          2X,'GS1S2K=',E14.7)
                YS1S2(I,K)=(YS1S2(I,K)+GS1S2K)*CI
C               WRITE (6,15) K,YS1S2(I,K)
 15             FORMAT(5X,I4,5X,'YS1S2=',(E14.7,2X,E14.7))
 4         CONTINUE
 3    CONTINUE
      DO 5 I=1,NSLOTS
           DO 6 J=I,NSLOTS
                KMAX=NSSL(I,J)
                DO 7 K=1,KMAX
```

```
                     IF (I.EQ.J) THEN
                         YS_ADM(I,J,K)=YS(K)
                     ELSE
                         IJ=INSS(I,J)
                         IF(IJ.EQ.1) YS_ADM(I,J,K)=YS(K)
                         IF(IJ.GE.2) YS_ADM(I,J,K)=YS1S2(IJ,K)
                     END IF
  7             CONTINUE
  6        CONTINUE
  5   CONTINUE
C
 1000 CONTINUE
      RETURN
      END
C..................................................................
C..................................................................
C     This subroutine evaluates the limits of integration in
C     the interval [0,A].
C     Specifically:
C          1) It divides the interval [0,k0] to 10 equal
C             subsections and then apply fixed-point Gaussian
C             Quadrature
C          2) It divides the interval [k0,k] into so many
C             subsections as the number of poles and in
C             such a way that each subsection includes one
C             pole only away from the ends of the subsection
C          3) It divides the interval [k,A] into 20 equal
C             subsections and then apply fixed-point Gaussian
C             Quadrature
C...................................................................
      SUBROUTINE LIMIT
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL WSPE,WTPE,WSPM
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
C------------------------------------------------+
C     Step 1 : Evaluation of vector CN           |
C              it gives the end points of the    |
C              intervals considered in (0,k0)    |
C------------------------------------------------+
      DELTA=AK0/FLOAT(NK0)
      CN(1)=0.D0
      DO 1 I=1,NK0
           CN(I+1)=DELTA*FLOAT(I)
  1   CONTINUE
C------------------------------------------------+
C     Step 2 : Evaluation of vector BM           |
C              it gives the end points of the    |
C              intervals considered in (k,A)     |
C------------------------------------------------+
      DELTA=(A/DSQRT(EER)-AK)/FLOAT(MA)
      BM(1)=AK
      DO 2 I=1,MA
           BM(I+1)=DELTA*FLOAT(I)+AK
  2   CONTINUE
C------------------------------------------------+
C     Step 3 : Evaluation of the vectors AM,DM   |
C              "AM" gives the end points around  |
```

```
C                   the TM poles                  |
C              "DM" gives the end points around   |
C                   the TE poles                  |
C                                                 |
C     IFIRST=   2   only one TM pole              |
C               1   TE0<TM0                       |
C               0   TM0<TE0                       |
C-------------------------------------------------+
      AM(1)=AK0
      DM(1)=AK0
      NMAX=NTE+NTM-1
      IF (IFIRST.EQ.2) GO TO 3
      DO 4 I=1,NMAX
           AM(I+1)=(POLES(I+1)+POLES(I))/2.D0
           DM(I+1)=AM(I+1)
   4  CONTINUE
      AM(NMAX+2)=AK
      DM(NMAX+2)=AK
      IF (IFIRST.EQ.1) GO TO 5
          DM(NMAX+1)=AM(NMAX+2)
          DO 6 I=1,NMAX
               DM(NMAX-I+1)=AM(NMAX-I+2)
  6       CONTINUE
          GO TO 7
  5       AM(NMAX+1)=DM(NMAX+2)
          DO 8 I=1,NMAX
               AM(NMAX-I+1)=DM(NMAX-I+2)
  8       CONTINUE
          GO TO 7
C
  3       DELTA=(AK-AK0)/FLOAT(NK0K)
          AM(1)=AK0
          DO 9 I=1,NK0K
               AM(I+1)=DELTA*FLOAT(NK0K)+AK0
  9       CONTINUE
  7       CONTINUE
C-------------------------------------------------+
C     Step 4 : evaluation of vectors VZXE         |
C-------------------------------------------------+
      IF (IFIRST.EQ.2) GO TO 10
          DO 11 I=1,NTE
                ARG=POLTE(I)
                VZXE(I)=HZXE(ARG)
  11      CONTINUE
  10  CONTINUE
C-------------------------------------------------+
C     Step 5 : evaluation of vector VXXM,VZXM     |
C-------------------------------------------------+
      DO 12 I=1,NTM
            ARG=POLTM(I)
            VXXM(I)=GXXM(ARG)
            VZXM(I)=GZXM(ARG)
  12  CONTINUE
      RETURN
      END
C.....................................................................
C.....................................................................
C     This subroutine evaluates the values of the integrand of
C     the Green's function at different points in the interval
C     [0,A]. Then it evaluetes the space integrals of the Bessel
C     function at the same points and multiply these values with
C     the corresponding values of the Green's function.
C     Finally , it multiplies these products with known coeffic.
C     and it adds them up. This way, the moments'-method
C     space integrals of the first part of the Green's function are
C     evaluated and are stored in the complex vectors ZS,ZS1S2
```

```
C.....................................................................
C.....................................................................
      SUBROUTINE GREEN
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YS,YS1S2,CI
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
C-----------------------------------------------+
C     Evaluation of the coefficients for the    |
C            FF's functions                     |
C-----------------------------------------------+
      F1X=1.D0
      F1Z=2.D0*(1.D0-ER)/((1.D0+ER)*(1.D0+E2)*(1.D0+0.5D0*E1))
      IF ((ER-1.D0).LT.0.005) F1Z=0.D0
C
      CALL ARIS
C
      DO 1 I=1,NPOINT
            INCON=I
            IY=I
            AI=COAL(I)
            TI=POINT(I)
C
C     evaluation of intervals 1 and 2
C
            IAD=1
            DO 2 N=1,NK0
                 AUP=CN(N+1)
                 ALOW=CN(N)
                 CALL FUNCT(IAD,AUP,ALOW,N,INCON)
   2        CONTINUE
C
C     evaluation of intervals 3 and 4
C
            NTTM=NTM
            IF (IFIRST.EQ.2) NTTM=NK0K
            DO 3 IAD=3,4
                 IFD=0
                 DO 4 N=1,NTTM
                      IFD=IFD+1
                      AUP=AM(IFD+1)
                      ALOW=AM(IFD)
                      CALL FUNCT(IAD,AUP,ALOW,N,INCON)
```

```
                      IFD=IFD+1
   4             CONTINUE
   3       CONTINUE
           IF (IFIRST.EQ.2) GO TO 9
C
C     evaluation of the intervals 5 and 6,9,11
C
           DO 5 IAD=5,6
                 IFD=0
                 DO 6 N=1,NTE
                      IFD=IFD+1
                      AUP=DM(IFD+1)
                      ALOW=DM(IFD)
                      CALL FUNCT(IAD,AUP,ALOW,N,INCON)
                      IFD=IFD+1
   6             CONTINUE
   5       CONTINUE
   9       CONTINUE
C
C     evaluation of the interval 7
C
           IAD=7
           DO 7 N=1,MA
                 AUP=BM(N+1)
                 ALOW=BM(N)
                 CALL FUNCT(IAD,AUP,ALOW,N,INCON)
  7        CONTINUE
  1   CONTINUE
C
C     evaluation of the intervals 8,10
C
      IAD=8
      IFD=0
      DO 8 N=1,NTM
           IFD=IFD+1
           AUP=AM(IFD+1)
           ALOW=AM(IFD)
           CALL FUNCT(IAD,AUP,ALOW,N,INCON)
           IFD=IFD+1
   8  CONTINUE
      RETURN
      END
C...................................................................
C             Functions :  GXXM,GZXM,HZXE
C
C     These functions evaluate the residues from the different poles
C...................................................................
      FUNCTION GXXM(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RM0H=RM0*H
      RMT=RM*(-H+T)
      SXN=RM*DCOS(RMT)-ER*RM0*DSIN(RMT)
      SXD=(ER+RM0H)*(RM/RM0)*DCOS(RMH)+(1.D0+ER*RM0H)*DSIN(RMH)
      GXXM=SXN/SXD
```

```
      RETURN
      END
C
C..............................................................
C
      FUNCTION GZXM(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RM0H=RM0*H
      RMT=RM*T
      CST=DCOS(RMT)
      CSH=DCOS(RMH)
      SNH=DSIN(RMH)
      SXN=RM*CST
      SXD=(RM*CSH+RM0*SNH)*((ER+RM0H)*CSH/RM0+(1.D0+ER*RM0H)*SNH/RM)
      GZXM=SXN/SXD
      RETURN
      END
C
C
C............................................................
C
      FUNCTION HZXE(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RMT=RM*T
      RM0H=RM0*H
      CSH=DCOS(RMH)
      CST=DCOS(RMT)
      SNH=DSIN(RMH)
      SXN=RM*CST
      SXD=(ER*RM0*CSH-RM*SNH)*(1.D0+RM0H)*(SNH/RM0-CSH/RM)
      HZXE=SXN/SXD
      RETURN
      END
C..........................................................
C     1) This subroutine evaluates the integrand of the Green's
C        function at different points (subroutine Grei).
C     2) It evaluates the space integrals comming from the
C        application of moments' method (subroutine adonis)
C     3) Multiply these two valueswith appropriate weighting
C        coefficients and it adds them up
C..........................................................
      SUBROUTINE FUNCT(IAD,AUP,ALOW,N,INCON)
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 S1,S2
      COMPLEX YS,YS1S2,CI
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      CI=(0.0,1.0)
      NCON=0
      X=AUP-ALOW
      Y=AUP+ALOW
      AK02=AK0*AK0
      AK2=AK*AK
      AKK2=AKK*AKK
      ER1=1.D0-ER
      IF (IAD.GT.2) GO TO 1
         ALI=0.5D0*(TI*X+Y)
         GCONX=AI*X*0.5D0
         FCONX=GCONX
         GCONZ=GCONX*ER1
         IF (DABS(ER1).LT.0.005) GCONZ=0.D0
         FCONZ=FCONX
         AIMA=1.D0
         CALL GREI(ALI,0.D0,0.D0,IAD,0.D0)
         GO TO 10
    1 IF (IAD.NE.3) GO TO 2
         ALI=0.5D0*(TI*X+Y)
         XTM=POLTM(N)
         TMTM=(2.D0*XTM-Y)/X
         GCONX=AI/(TI-TMTM)
         GCONZ=GCONX*ER1
         FCONX=AI*X*0.5D0
         FCONZ=FCONX
         AIMA=0.D0
         IF (DABS(ER1).LT.0.005) THEN
                  GCONX=0.D0
                  GCONZ=0.D0
                  FCONX=0.D0
                  FCONZ=0.D0
         END IF
         CALL GREI(ALI,XTM,0.D0,IAD,0.D0)
         GO TO 10
    2 IF (IAD.NE.4) GO TO 3
         ALI=POLTM(N)
         TM=(2.D0*ALI-Y)/X
```

```
         GCONX=-AI/(TI-TM)
         GCONZ=GCONX*ER1
         FCONX=0.D0
         FCONZ=0.D0
         AIMA=0.D0
         RX=VXXM(N)
         RZ=VZXM(N)
         IF (DABS(ER1).LT.0.005) THEN
                  GCONX=0.D0
                  GCONZ=0.D0
                  FCONX=0.D0
                  FCONZ=0.D0
         END IF
         GO TO 10
  3   IF (IFIRST.EQ.2) GO TO 5
      IF (IAD.NE.5) GO TO 4
         ALI=0.5D0*(TI*X+Y)
         XTE=POLTE(N)
         TMTE=(2.D0*XTE-Y)/X
         GCONX=AI*X*0.5D0
         GCONZ=AI*ER1/(TI-TMTE)
         FCONX=GCONX
         FCONZ=FCONX
         AIMA=0.D0
         CALL GREI(ALI,0.D0,XTE,IAD,TMTE)
         IF (DABS(ER1).LT.0.005) THEN
                  GCONX=0.D0
                  GCONZ=0.D0
                  FCONX=0.D0
                  FCONZ=0.D0
         END IF
         GO TO 10
  4   IF (IAD.NE.6) GO TO 5
         NCON=6
         ALI=POLTE(N)
         TM=(2.D0*ALI-Y)/X
         GCONX=0.D0
         GCONZ=-AI*ER1/(TI-TM)
         FCONX=0.D0
         FCONZ=0.D0
         AIMA=0.D0
         RZ=VZXE(N)
         IF (DABS(ER1).LT.0.005) THEN
                  GCONX=0.D0
                  GCONZ=0.D0
                  FCONX=0.D0
                  FCONZ=0.D0
         END IF
         GO TO 10
  5   IF (IAD.NE.7) GO TO 6
         ALI=0.5D0*(TI*X+Y)
         GCONX=AI*X*0.5D0
         GCONZ=GCONX*ER1
         IF (DABS(ER1).LT.0.005) GCONZ=0.D0
         FCONX=GCONX
         FCONZ=FCONX
         AIMA=0.D0
         CALL GREI(ALI,0.D0,0.D0,IAD,0.D0)
         GO TO 10
  6      NCON=8
         ALI=POLTM(N)
         TM=(2.D0*ALI-Y)/X
         FCONX=0.D0
         FCONZ=0.D0
         AIMA=0.D0
         RX=VXXM(N)
```

```
         RZ=VZXM(N)
         GO TO 28
C
  10  CONTINUE
         GXXR=GCONX*RX-FCONX*FRX
         GXXX=AIMA*GCONX*XX
         GZXR=GCONZ*RZ-FCONZ*FRZ
         GZXX=AIMA*GCONZ*XZ
  27  CONTINUE
         VARX=(AK2-AKK2)*GXXR+AKK2*GZXR
         VARZ=AKK*(GXXR-GZXR)
         GXXR=VARX
         GZXR=VARZ
         VARX=(AK2-AKK2)*GXXX+AKK2*GZXX
         VARZ=AKK*(GXXX-GZXX)
         GXXX=VARX
         GZXX=VARZ
      PLI=ALI
C
      CALL ADONIS
      KMAX=NOFFS(1)
      DO 13 K=1,KMAX
            S1=REAL(GXXR*SSJ0(K,1)+GZXR*SAJ0(K,1))
            S2=REAL(GXXX*SSJ0(K,1)+GZXX*SAJ0(K,1))
            YS(K)=YS(K)+S1-CI*S2
  13  CONTINUE
      DO 14 I=2,NOFF
            KMAX=NOFFS(I)
            DO 15 K=1,KMAX
                  S1=REAL(GXXR*SSJ0(K,I)+GZXR*SAJ0(K,I))
                  S2=REAL(GXXX*SSJ0(K,I)+GZXX*SAJ0(K,I))
                  YS1S2(I,K)=YS1S2(I,K)+S1-CI*S2
  15        CONTINUE
  14  CONTINUE
  28  IF (NCON.EQ.0) GO TO 24
      IF (INCON.LT.NPOINT) GO TO 24
          GCONX1=0.0
          GCONX2=0.0
          GCONZ1=ER1*DLOG((1.D0-TM)/(1.D0+TM))
          GCONZ2=ER1*PI
          IF (NCON.EQ.6) GO TO 29
              GCONX1=GCONZ1/ER1
              GCONX2=GCONZ2/ER1
  29      CONTINUE
              GXXR=GCONX1*RX
              GXXX=GCONX2*RX
              GZXR=GCONZ1*RZ
              GZXX=GCONZ2*RZ
              FXXR=0.D0
              FZXR=0.D0
              IF (DABS(ER1).LT.0.005) THEN
                    GXXR=0.D0
                    GXXX=0.D0
                    GZXR=0.D0
                    GZXX=0.D0
              END IF
  25      CONTINUE
          NCON=0
          GO TO 27
  24      CONTINUE
          RETURN
          END
C.................................................................
C     This subroutine evaluates the integrand of the green's
C     function at different points
C.................................................................
```

```
      SUBROUTINE GREI(X,XFM,XFE,IAD,TM)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      X2=X*X
      AK2=AK*AK
      AK02=AK0*AK0
      RM=DSQRT(DABS(AK2-X2))
      RM0=DSQRT(DABS(X2-AK02))
      RMH=RM*H
      RMT=RM*T
      RMHT=RM*(-H+T)
C
      CSH=DCOS(RMH)
      SNH=DSIN(RMH)
      CST=DCOS(RMT)
      SNT=DSIN(RMT)
      CSHT=DCOS(RMHT)
      SNHT=DSIN(RMHT)
C
      RM2=RM*RM
      RM02=RM0*RM0
      CSH2=CSH*CSH
      ERM0=ER*RM0
      ERM02=ERM0*ERM0
C
      EXX=DEXP(-X*T/FA)/FA
      EXZ=DEXP(-X*(2.D0*H)/FA)/FA
      IF (IAD.NE.7) GO TO 100
          EX=DEXP(RMH)
          TANH=(EX-1.D0/EX)/(EX+1.D0/EX)
          CSHH=(EX+1.D0/EX)/2.D0
          EX=DEXP(RMT)
          CSHT=0.5D0*(EX+1.D0/EX)
          SNHT=0.5D0*(EX-1.D0/EX)
          TANT=SNHT/CSHT
          EX=DEXP(RMHT)
          CSHHT=0.5D0*(EX+1.D0/EX)
          SNHHT=0.5D0*(EX-1.D0/EX)
          TANHT=SNHHT/CSHHT
C
  100 IF (IAD.NE.1) GO TO 1
          DEN=RM2+(ERM02-RM2)*CSH2
          RNOM=-RM2*SNT+(RM2-ERM02)*CSH*SNHT
          XNOM=ER*RM*RM0*CST
          C1=X/RM
C
          RX=C1*RNOM/DEN
          IF ((ER-1.D0).LT.0.005) RX=0.D0
          XX=C1*XNOM/DEN
          FRX=F1X*EXX
C
          DEN=DEN*(RM02+AK02*(ER-1.D0)*CSH2)
          RNOM=-CST*(RM2+ER*RM02)*CSH*SNH
          XNOM=CST*RM*RM0*(-1.D0+(1.D0+ER)*CSH2)
          C1=X*RM
          RZ=-C1*RNOM/DEN
          XZ=C1*XNOM/DEN
```

```
          FRZ=F1Z*EXZ
          RETURN
  1   IF (IAD.NE.3) GO TO 2
          C1=X-XFM
          IF (DABS(AK-X).LT.1.D-6) GO TO 10
              DEN=ERM0*CSH-RM*SNH
              RNOM=(RM*CSHT-ERM0*SNHT)
              C2=X/RM
              RX=C1*C2*RNOM/DEN
C
              DEN=DEN*(RM*CSH+RM0*SNH)
              RNOM=CST
              C3=X*RM
              RZ=C1*C3*RNOM/DEN
C
              FRX=F1X*EXX
              FRZ=F1Z*EXZ
              RETURN
C
   10     RNOM=1.D0-ERM0*(-H+T)
          RX=C1*X*RNOM/ERM0
          FRX=F1X*EXX
C
          RZ=X*C1/(ERM0*(1.D0+RM0*H))
          FRZ=F1Z*EXZ
          RETURN
  2   IF (IAD.NE.5) GO TO 4
          C1=X-XFE
          IF (DABS(AK-X).LT.1.D-6) GO TO 13
              RNOM=RM*CSHT-ERM0*SNHT
              DEN=ERM0*CSH-RM*SNH
              RX=(X/RM)*RNOM/DEN
              FRX=F1X*EXX
C
              RNOM=RM*CST
              DEN=DEN*(RM*CSH+RM0*SNH)
              RZ=X*C1*RNOM/DEN
              FRZ=F1Z*EXZ
              RETURN
   13     RX=X*(1.D0-ERM0*(-H+T))/ERM0
          FRX=F1X*EXX
C
          RZ=X*C1/(ERM0*(1.D0+RM0*H))
          FRZ=F1Z*EXZ
          RETURN
  4   IF (IAD.NE.7) GO TO 6
          IF (DABS(X-AK).LT.1.D-6) GO TO 15
              DEN=ERM0+RM*TANH
              RNOM=(RM+ERM0*TANH)*CSHT-DEN*SNHT
              RX=(X/RM)*RNOM/DEN
              FRX=F1X*EXX
C
              RNOM=X*(RM*CSHT)/(CSHH*CSHH)
              DEN=DEN*(RM+RM0*TANH)
              RZ=RNOM/DEN
              FRZ=F1Z*EXZ
              RETURN
 15   RX=X*(1.D0-ERM0*(-H+T))/ERM0
      FRX=F1X*EXX
      RZ=(X/ERM0)/(1.D0+RM0*H)
      FRZ=F1Z*EXZ
 6    CONTINUE
      RETURN
      END
C................................................................
C                             ARIS
```

```
C..................................................................
      SUBROUTINE ARIS
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
C    +--------------------------------------------+
C    | Formation of the matrices:  DIST,          |
C    |     DARG,RCOE                              |
C    |                                            |
C    +--------------------------------------------+
      W2=W/2.D0
      U=WREAL/W
      THMIN=DATAN(DSQRT(1.D0/(U*U)-1.D0))
      THMAX=PI-THMIN
      AX=(THMAX-THMIN)/2.D0
      BX=(THMAX+THMIN)/2.D0
      X=PI/4.D0
      DO 1 J=1,NOFF
           MAX=NMAX(J)
           LPOINT=MPOINT
           IF (OFFSET(J).LE.OFFLIM) LPOINT=NPOINT
           DO 2 I=1,LPOINT
                POIN=BPOINT(I)
                IF (OFFSET(J).LE.OFFLIM) POIN=POINT(I)
                FI=X*(POIN+1.D0)
                THETA=AX*POIN+BX
                AS=DSIN(FI)
                AC=DCOS(FI)
                DARG(J,I,1)=W2*AC
                DARG(J,I,2)=AC
                DARG(J,I,3)=AS
                DARG(J,I,4)=X
                DO 3 N=1,MAX
                     AXN=FLOAT(N-2)*DLX
                     IF (OFFSET(J).GT.OFFLIM) GO TO 4
                        DIST(N,J,I)=AXN*AS
                        GO TO 5
  4                  AXN2=AXN*AXN
                     BXN=OFFSET(J)-W*DCOS(THETA)/2.D0
                     BXN2=BXN*BXN
                     DIST(N,J,I)=DSQRT(AXN2+BXN2)
                     SIG=DIST(N,J,I)
                     SIG2=SIG*SIG
                     SIG3=SIG2*SIG
                DSIG=DABS(AXN)/SIG
                DSIG2=BXN2/SIG3
                DSIG3=-3.D0*DSIG*DSIG2/SIG
                DSIG4=-3.D0*DSIG2*(DSIG2-4.D0*DSIG**2/SIG)/SIG
                DSIG5=-3.D0*(-15.D0*DSIG2**2*DSIG+(20.D0/SIG)*
     *                DSIG2*DSIG**3)/SIG2
                DSIG6=-3.D0*(-15.D0*DSIG2**3+(180.D0/SIG)*DSIG2
```

```
     *               **2*DSIG**2-(120.D0/SIG2)*DSIG2*DSIG**4)/
     *               SIG2
               DSIG7=-3.D0*(525.D0*DSIG2**3*DSIG-(2100.D0/SIG)*
     *               DSIG2**2*DSIG**3+(840.D0/SIG2)*DSIG2*DSIG
     *               **5)/SIG3
               DSIG8=-3.D0*(525.D0*DSIG2**4-(12600.D0/SIG)*DSIG2
     *               **3*DSIG**2+(25200.D0/SIG2)*DSIG2**2*DSIG**4
     *               -(6720.D0/SIG3)*DSIG2*DSIG**6)/SIG3
C
C           Evaluation of the coefficients Gij
C
               G21=DSIG2
               G22=DSIG**2
C----------------
               G41=DSIG4
               G42=4.D0*DSIG3*DSIG+3.D0*DSIG2**2
               G43=6.D0*DSIG2*DSIG**2
               G44=DSIG**4
C----------------
               G61=DSIG6
               G62=6.D0*DSIG5*DSIG+15.D0*DSIG4*DSIG2+10.D0*DSIG3**2
               G63=15.D0*DSIG4*DSIG**2+60.D0*DSIG3*DSIG2*DSIG+15.D0
     *             *DSIG2**3
               G64=20.D0*DSIG3*DSIG**3+45.D0*DSIG2**2*DSIG**2
               G65=15.D0*DSIG2*DSIG**4
               G66=DSIG**6
C----------------
               G81=DSIG8
               G82=8.D0*DSIG7*DSIG+28.D0*DSIG6*DSIG2+56.D0*DSIG5
     *             *DSIG3+35.D0*DSIG4**2
               G83=28.D0*DSIG6*DSIG**2+168.D0*DSIG5*DSIG2*DSIG+
     *             280.D0*DSIG4*DSIG3*DSIG+210.D0*DSIG4*DSIG2**2+
     *             280.D0*DSIG3**2*DSIG2
               G84=56.D0*DSIG5*DSIG**3+420.D0*DSIG4*DSIG2*DSIG**2
     *             +280.D0*DSIG3**2*DSIG**2+840.D0*DSIG3*DSIG2**2
     *             *DSIG+105.D0*DSIG2**4
               G85=70.D0*DSIG4*DSIG**4+560.D0*DSIG3*DSIG2*DSIG**3
     *             +420.D0*DSIG2**3*DSIG**2
               G86=56.D0*DSIG3*DSIG**5+210.D0*DSIG2**2*DSIG**4
               G87=28.D0*DSIG2*DSIG**6
               G88=DSIG**8
C----------------
               RCOE(2,N,J,I)=-0.5D0*(G22+SIG*G21)
               RCOE(1,N,J,I)=0.5D0*(G22-SIG*G21)
C----------------
               SX=0.5D0*SIG*(G42-SIG*G41)
               S30=-0.5D0*SIG*(G42+SIG*G41)
               S31=0.25D0*(SX+3.D0*G43)
               S33=0.25D0*(SX-G43)
               RCOE(3,N,J,I)=0.5D0*(SIG*S33/3.D0+G44/4.D0)
               RCOE(4,N,J,I)=0.5D0*(SIG*S31+SIG*S33/3.D0-G44)
               RCOE(5,N,J,I)=0.5D0*(SIG*S31+3.D0*G44/4.D0)
               RCOE(6,N,J,I)=SIG*S30
C----------------
               SX=SIG*S33/3.D0+G64/4.D0
               ST=SIG*S31+SIG*S33/3.D0-G64
               S5M3=SIG2*S30
               S5M1=0.5D0*SIG*(SIG*S31+3.D0*G64/4.D0)
               S51=0.25D0*(0.5D0*SIG*ST-5.D0*G65/2.D0)
               S53=0.25D0*(0.5D0*SIG*ST+0.25D0*SIG*SX+0.5D0*G65/
     *             4.D0)
               S55=0.125D0*(0.5D0*SIG*SX-0.5*G65)
               RCOE(7,N,J,I)=0.5D0*(SIG*S55/5.D0+G66/16.D0)
               RCOE(8,N,J,I)=0.5D0*(SIG*S53/3.D0+SIG*S55/5.D0-
     *                       6.D0*G66/16.D0)
               RCOE(9,N,J,I)=0.5D0*(SIG*S51+SIG*S53/3.D0+15.D0*
```

```
     *                       G66/16.D0)
                RCOE(10,N,J,I)=0.5D0*(SIG*S51-10.D0*G66/16.D0)
                RCOE(11,N,J,I)=SIG*S5M1
                RCOE(12,N,J,I)=SIG*S5M3
C----------------
                S7M5=SIG2*S5M3
                S7M3=SIG2*S5M1
                S7M1=0.5D0*SIG*(SIG*S51-10.D0*G86/16.D0)
                S71=0.5D0*(0.25D0*SIG*(SIG*S51+SIG*S53/3.D0+
     *              15.D0*G86/16.D0)+35.D0*G87/32.D0)
                S73=0.5D0*(0.25D0*SIG*(SIG*S51+SIG*S53/3.D0+15.D0
     *              *G86/16.D0)+0.125D0*SIG*(SIG*S53/3.D0+SIG*
     *              S55/5.D0-6.D0*G86/16.D0)-21.D0*G87/32.D0)
                S75=0.5D0*(0.125D0*SIG*(SIG*S53/3.D0+SIG*S55/5.D0-
     *              6.D0*G86/16.D0)+(SIG/12.D0)*(SIG*S55/5.D0+
     *              G86/16.D0)+7.D0*G87/32.D0)
                S77=0.5D0*((SIG/12.D0)*(SIG*S55/5.D0+G86/16.D0)-
     *              G87/32.D0)
                RCOE(13,N,J,I)=0.5D0*(SIG*S77/7.D0+G88/64.D0)
                RCOE(14,N,J,I)=0.5D0*(SIG*S75/5.D0+S77*SIG/7.D0
     *                         -8.D0*G88/64.D0)
                RCOE(15,N,J,I)=0.5D0*(SIG*S73/3.D0+SIG*S75/5.D0
     *                         +28.D0*G88/64.D0)
                RCOE(16,N,J,I)=0.5D0*(SIG*S71+SIG*S73/3.D0-56.D0
     *                         *G88/64.D0)
                RCOE(17,N,J,I)=0.5D0*(SIG*S71+35.D0*G88/64.D0)
                RCOE(18,N,J,I)=SIG*S7M1
                RCOE(19,N,J,I)=SIG*S7M3
                RCOE(20,N,J,I)=SIG*S7M5
    5               CONTINUE
    3           CONTINUE
    2       CONTINUE
    1 CONTINUE
C
C             Formation of the series s(dlx) . Storage in
C                  vectors SERS(5),SERA(5)
C
C
      U1=2.D0*THMIN/FLOAT(NSER)
      DO 6 JN=1,NSER
           S2=(2.D0*FLOAT(JN)-1.D0)
           S2=S2/(2.D0*FLOAT(NSER))
           S3=DCOS(S2*THMIN)
           S(JN,2)=S3*W/2.D0
           S(JN,1)=U1
   6  CONTINUE
      ADL=AKK*DLX
      ADL2=ADL*ADL
      ADL3=ADL2*ADL
      ADL4=ADL3*ADL
      ADL5=ADL4*ADL
      ADL6=ADL5*ADL
      YSIN=DSIN(ADL)
      YCOS=DCOS(ADL)
C
      SER1=(1.D0-YCOS)*2.D0/AKK
C
      SER2=-YSIN/3.D0+ADL*YCOS/4.D0+ADL2*YSIN/10.D0-ADL3*YCOS/36.D0
     *     -ADL4*YSIN/168.D0+ADL5*YCOS/960.D0+ADL6*YSIN/6480.D0
C
      SER3=YSIN/60.D0-ADL*5.D0*YCOS/360.D0-ADL2*YSIN/168.D0+ADL3
     *     *YCOS/560.D0+ADL4*YSIN/2592.D0-ADL5*YCOS/12960.D0-ADL6
     *     *YSIN/95040.D0
C
      SER4=-YSIN/2520.D0+ADL*YCOS/2880.D0+ADL2*YSIN/6480.D0-ADL3
     *     *YCOS/21600.D0-ADL4*YSIN/95040.D0+ADL5*YCOS/518400.D0
C
```

```
      SER5=YSIN/181440.D0-ADL*YCOS/201600.D0-ADL2*YSIN/443520.D0+
     *     ADL3*YCOS/1442775.9D0
C
      SERS(1)=SER1*SER1
      SERS(2)=DLX*2.D0*SER1*SER2
      SERS(3)=DLX*(DLX*SER2*SER2+2.D0*SER1*SER3)
      SERS(4)=DLX*(2.D0*SER1*SER4+2.D0*DLX*SER2*SER3)
      SERS(5)=DLX*(DLX*SER3*SER3+2.D0*DLX*SER2*SER4)
C
      SERA(1)=SER1
      SERA(2)=DLX*SER2
      SERA(3)=DLX*SER3
      SERA(4)=DLX*SER4
      SERA(5)=DLX*SER5
  111 CONTINUE
      RETURN
      END
C....................................................................
C                         ADONIS
C     This subroutine evaluates the space integarls of the bessel
C         function
C
C...............................................................
      SUBROUTINE ADONIS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2),DERIV(9,3)
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/BSS/ARG(10),AARG
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      ARX=W*AX/2.D0
      W1=2.D0*YCOS
      PR1=PLI*DLX
      PR2=PR1*PR1
      PR4=PR2*PR2
      PR6=PR4*PR2
      PR8=PR6*PR2
      DO 1 J=1,NOFF
           MAX=NMAX(J)
           DO 2 N=1,MAX
                SSJ0(N,J)=0.D0
                SAJ0(N,J)=0.D0
  2        CONTINUE
  1   CONTINUE
C
      DO 11 J=1,NOFF
            LPOINT=MPOINT
```

```
          IF (OFFSET(J).GT.OFFLIM) GO TO 12
              LPOINT=NPOINT
              DO 13 I=1,NPOINT
                    ARG(I)=PLI*DARG(J,I,1)
13            CONTINUE
              CALL BESS1(BJ)
12        DO 14 I=1,LPOINT
                DO 17 NK=1,5
                      DERIV(NK,1)=0.D0
                      DERIV(NK,2)=0.D0
17              CONTINUE
                ASIN=ARX*BCOAL(I)
                IF (OFFSET(J).GT.OFFLIM) GO TO 15
                      ASIN=W*DARG(J,I,4)*COAL(I)
                      AROF=PLI*OFFSET(J)*DARG(J,I,2)
                      COFF=DCOS(AROF)
                      SSUM=0.D0
                      DO 16 JN=1,NSER
                            ARAF=PLI*S(JN,2)*DARG(J,I,2)
                            CAFF=DCOS(ARAF)
                            SSUM=SSUM+S(JN,1)*CAFF
16                    CONTINUE
15              CONTINUE
                KMAX=NMAX(J)
                DO 18 K=1,KMAX
                      DO 20 NK=1,5
                            DERIV(NK,1)=DERIV(NK,2)
                            DERIV(NK,2)=DERIV(NK,3)
20                    CONTINUE
                      IF (OFFSET(J).GT.OFFLIM) GO TO 21
                            SIN1=DARG(J,I,3)
                            SIN2=SIN1*SIN1
                            COS1=DCOS(PLI*DIST(K,J,I))
                            TERM=COFF*(BJ(I,1)-SSUM/PI)*COS1
                            DERIV(1,3)=TERM
                            SIN1=SIN2
                            DERIV(2,3)=-TERM*SIN1
                            SIN1=SIN1*SIN2
                            DERIV(3,3)=TERM*SIN1
                            SIN1=SIN1*SIN2
                            DERIV(4,3)=-TERM*SIN1
                            SIN1=SIN1*SIN2
                            DERIV(5,3)=TERM*SIN1
                            GO TO 22
21                  AARG=PLI*DIST(K,J,I)
                    ARG2=AARG*AARG
                    ARG4=ARG2*ARG2
                    ARG6=ARG4*ARG2
                    CALL BESS2(BJ)
                    DERIV(1,3)=BJ(1,2)
                    DERIV(2,3)=RCOE(1,K,J,I)*BJ(3,2)+
    *                          RCOE(2,K,J,I)*BJ(1,2)
                    DERIV(3,3)=RCOE(3,K,J,I)*BJ(5,2)+
    *                          RCOE(4,K,J,I)*BJ(3,2)+(RCOE(5,K,J,I)
    *                          +RCOE(6,K,J,I)/ARG2)*BJ(1,2)
                    DERIV(4,3)=RCOE(7,K,J,I)*BJ(7,2)+
    *                          RCOE(8,K,J,I)*BJ(5,2)+RCOE(9,K,J,I)*
    *                          BJ(3,2)+(RCOE(10,K,J,I)+RCOE(11,K,
    *                          J,I)/ARG2+RCOE(12,K,J,I)/ARG4)*
    *                          BJ(1,2)
                    DERIV(5,3)=RCOE(13,K,J,I)*BJ(9,2)+
    *                          RCOE(14,K,J,I)*BJ(7,2)+RCOE(15,K,J,
    *                          I)*BJ(5,2)+RCOE(16,K,J,I)*BJ(3,2)+
    *                          (RCOE(17,K,J,I)+RCOE(18,K,J,I)/ARG2
    *                          +RCOE(19,K,J,I)/ARG4+RCOE(20,K,J,I)
    *                          /ARG6)*BJ(1,2)
```

```
  22                  IF (K.LT.3) GO TO 18
                      SUMS=SERS(1)*DERIV(1,2)-PR2*SERS(2)*DERIV(2,2)
     *                     +PR4*SERS(3)*DERIV(3,2)-PR6*SERS(4)*DERIV
     *                     (4,2)+PR8*SERS(5)*DERIV(5,2)
C
                      CH1=SERA(1)*(DERIV(1,1)+DERIV(1,3)-W1*DERIV
     *                    (1,2))
                      CH2=SERA(2)*(DERIV(2,1)+DERIV(2,3)-W1*DERIV
     *                    (2,2))*PR2
                      CH3=SERA(3)*(DERIV(3,1)+DERIV(3,3)-W1*DERIV
     *                    (3,2))*PR4
                      CH4=SERA(4)*(DERIV(4,1)+DERIV(4,3)-W1*DERIV
     *                    (4,2))*PR6
                      CH5=SERA(5)*(DERIV(5,1)+DERIV(5,3)-W1*DERIV
     *                    (5,2))*PR8
                      SUMA=CH1-CH2+CH3-CH4+CH5
                      KJ=K-2
                      SSJ0(KJ,J)=SSJ0(KJ,J)+ASIN*SUMS
                      SAJ0(KJ,J)=SAJ0(KJ,J)+ASIN*SUMA
CCCC
C                     IF (KJ.EQ.1)WRITE (6,665) KJ,J,SSJ0(KJ,J),
C    *                SUMS,SAJ0(KJ,J),SUMA
C665                  FORMAT(10X,'KJ=',I4,2X,'J=',I4/10X,'SSJ0=',
C    *                E14.7,2X,'SUMS=',E14.7/10X,'SAJ0=',E14.7,
C    *                2X,'SUMA=',E14.7/)
CCCC
  18            CONTINUE
  14       CONTINUE
  11  CONTINUE
      RETURN
      END
C...................................................................
C                           BESS1
C     This subroutine gives values for the zeroth order
C     Bessel functions. It is used for small offsets
C...................................................................
      SUBROUTINE BESS1(BJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2)
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/BSS/ARG(10),AARG
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      PI=3.141592653589D0
      DO 1 IJ=1,NPOINT
           X=ARG(IJ)
           IF (X.GT.0.001D0) GO TO 10
              X3=X/3.D0
              X32=X3*X3
              X34=X32*X32
              X36=X34*X32
              BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *            *X36
              BJ(IJ,1)=BJ0
              GO TO 1
  10       IF (X.GT.3.D0) GO TO 12
              X3=X/3.D0
              X32=X3*X3
```

```
              X34=X32*X32
              X36=X34*X32
              X38=X36*X32
              X310=X38*X32
              X312=X310*X32
              BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *           *X36+0.0444479D0*X38-0.0039444D0*X310+0.00021000
     *            D0*X312
              BJ(IJ,1)=BJ0
              GO TO 1
   12      CONTINUE
           X3=3.D0/X
           X32=X3*X3
           X33=X32*X3
           X34=X33*X3
           X35=X34*X3
           X36=X35*X3
           FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.0000
     *         9512D0*X33+0.00137237D0*X34-0.00072805D0*X35+0.00014
     *         476D0*X36
           TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00
     *         262573D0*X33-0.00054125D0*X34-0.00029333D0*X35+0.000
     *         13558D0*X36
           WCON=DSQRT(1.D0/X)
           BJ(IJ,1)=WCON*FJ0*DCOS(TJ0)
   1       CONTINUE
      RETURN
      END
C.................................................................
C                             TAIL
C     This subroutine evaluates the tail contribution
C.................................................................
      SUBROUTINE TAIL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YS,YS1S2
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/OUT/GS(250),GS1S2(7,250)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/WIDTH/W,WDELTA
C
      COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XNT(40,3),
     *CNT(40),NDP,NTP,NSP
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
C     This vector contains the values of t in the integrals h0
C
      Z1=T
      Z2=2.D0*H
C
C     This vector contains the values of the coefficient C in
C          the integrals h0
C
```

```
      C1=FA
C
C
C     This vector contains the values of the coefficient A in
C              the integrals h0
C
      AK2=AK*AK
      AKK2=AKK*AKK
      AK02=AK0*AK0
      W2=W/2.D0
      THMIN=WREAL/W
      THMIN=DATAN(DSQRT(1.D0/THMIN**2-1.D0))
      THMAX=PI-THMIN
      PI2=PI/2.D0
      PI4=PI/4.D0
      DLX2=DLX/2.D0
      DLX4=DLX2*DLX2
C
      YCOS=DCOS(AKK*DLX)
      CCS=DCOS(2.D0*AKK*DLX)
      YSIN=DSIN(AKK*DLX)
      SSN=DSIN(2.D0*AKK*DLX)
C
C     +---------------------------------+
C     | Evaluation of S1,S2,S3,S4,S5,S6 |
C     |      (Single Integrals)         |
C     +---------------------------------+
C
C
      DO 201 J=1,7
             DO 202 K=1,205
                    DO 203 JK=1,4
                           S1(JK,K,J)=0.D0
                           D1(JK,K,J)=0.D0
                           D2(JK,K,J)=0.D0
                           T1(JK,K,J)=0.D0
                           T2(JK,K,J)=0.D0
                           T3(JK,K,J)=0.D0
                           T4(JK,K,J)=0.D0
  203               CONTINUE
  202        CONTINUE
  201 CONTINUE
C
      ZP1=Z1/C1
      ZP2=Z2/C1
C
      ZP12=ZP1*ZP1
      ZP22=ZP2*ZP2
      DO 1 J=1,NOFF
           KMAX=NMAX(J)+2
           IF (OFFSET(J).LT.1.D-6) THMAX=PI
           DSP=(THMAX-THMIN)/4.D0
           DDP=DSP*DLX2
           DTP=DSP*DLX4
           COEF1=(THMAX-THMIN)/2.D0
           IF (OFFSET(J).LT.1.D-6) COEF1=(PI/2.D0-THMIN)/2.D0
           COEF2=(THMAX+THMIN)/2.D0
           IF (OFFSET(J).LT.1.D-6) COEF2=(PI/2.D0+THMIN)/2.D0
           DO 10 I=1,NSP
                 THI=COEF1*XNS(I)+COEF2
                 C1=DCOS(THI)
                 C2=W2*C1
                 C2=OFFSET(J)-C2
                 CW=C2*C2
                 AASIN=CNS(I)*DSP
                 DO 11 K=1,KMAX
```

```
                      XN=FLOAT(K-3)*DLX
                      RAD2=XN*XN+CW
                      TRAD1=DSQRT(RAD2+ZP12)
                      TRAD2=DSQRT(RAD2+ZP22)
                      S1(1,K,J)=S1(1,K,J)+DLOG(2.D0*(TRAD1+XN))*AASIN
                      S1(2,K,J)=S1(2,K,J)+DLOG(2.D0*(TRAD2+XN))*AASIN
  11            CONTINUE
  10      CONTINUE
C
C    +------------------------------------------------------------+
C    | EVALUATION OF D1,D2,D4,D5                                  1
C    +------------------------------------------------------------+
          DO 20 I=1,NDP
                THI=COEF1*XND(I,1)+COEF2
                XI=DLX2*(XND(I,2)+1.D0)
                C1=DCOS(THI)
                C2=W2*C1
                C2=OFFSET(J)-C2
                CW=C2*C2
                AASIN=CND(I)*DDP
                SV1=DSIN(AKK*(DLX-XI))
                SV2=-SV1
                SV4=DSIN(AKK*XI)
                C2=DCOS(AKK*(DLX-XI))
                DO 21 K=1,KMAX
                     XNP=XI+FLOAT(K-2)*DLX
                     XNM=-XI+FLOAT(K-2)*DLX
                     RADP2=XNP*XNP+CW
                     RADM2=XNM*XNM+CW
                     TRAP1=DSQRT(RADP2+ZP12)
                     TRAP2=DSQRT(RADP2+ZP22)
C
                     TRAM1=DSQRT(RADM2+ZP12)
                     TRAM2=DSQRT(RADM2+ZP22)
C
                     XA1=AKK*XNP
                     XA2=AKK*XNM
                     XAP=DSIN(XA1)
                     XAM=DSIN(XA2)
C
                     SANP1=XAP*DLOG(2.D0*(TRAP1+XNP))
                     SANP2=XAP*DLOG(2.D0*(TRAP2+XNP))
C
                     SANM1=XAM*DLOG(2.D0*(TRAM1+XNM))
                     SANM2=XAM*DLOG(2.D0*(TRAM2+XNM))
C
                     XAP=DSIN(XA1/2.D0)
                     XAM=DSIN(XA2/2.D0)
                     SONP1=XAP/TRAP1
                     SONP2=XAP/TRAP2
C
                     SONM1=XAM/TRAM1
                     SONM2=XAM/TRAM2
C
                     Y1=-XNM/2.D0-DLX
                     Y2=-XNP/2.D0+DLX
                     CY1=DCOS(AKK*Y1)
                     CY2=DCOS(AKK*Y2)
                     SY1=DSIN(AKK*Y1)
                     SY2=DSIN(AKK*Y2)
C
                     D1(1,K,J)=D1(1,K,J)+(SANP1+SANM1)*SV2*AASIN
                     D2(1,K,J)=D2(1,K,J)+(CY1*SONP1-CY2*SONM1)*AASIN
                     D1(2,K,J)=D1(2,K,J)+(SANP2+SANM2)*SV2*AASIN
                     D2(2,K,J)=D2(2,K,J)+(CY1*SONP2-CY2*SONM2)*AASIN
  21           CONTINUE
```

```
  20      CONTINUE
C
C      evaluation of T1,T2,T3,T4
C
          DO 30 I=1,NTP
                THI=COEF1*XNT(I,1)+COEF2
                XI=DLX2*(XNT(I,2)+1.D0)
                XIP=DLX2*(XNT(I,3)+1.D0)
                C1=DCOS(THI)
                C2=W2*C1
                C2=OFFSET(J)-C2
                CW=C2*C2
                SV1=DSIN(AKK*(DLX-XI))
                SV2=-SV1
                SV3=DSIN(AKK*(DLX-XIP))
                AASIN=DTP*CNT(I)
                DO 31 K=1,KMAX
                      XNPP=(XI+XIP)+FLOAT(K-1)*DLX
                      XNPM=(XI-XIP)+FLOAT(K-1)*DLX
                      XNMP=(-XI+XIP)+FLOAT(K-1)*DLX
                      XNMM=(-XI-XIP)+FLOAT(K-1)*DLX
                      RADPP2=XNPP*XNPP+CW
                      RADPM2=XNPM*XNPM+CW
                      RADMP2=XNMP*XNMP+CW
                      RADMM2=XNMM*XNMM+CW
                      TAPP1=DSQRT(RADPP2+ZP12)
                      TAPP2=DSQRT(RADPP2+ZP22)
                      TAPM1=DSQRT(RADPM2+ZP12)
                      TAPM2=DSQRT(RADPM2+ZP22)
                      TAMP1=DSQRT(RADMP2+ZP12)
                      TAMP2=DSQRT(RADMP2+ZP22)
                      TAMM1=DSQRT(RADMM2+ZP12)
                      TAMM2=DSQRT(RADMM2+ZP22)
                      CST1=DCOS(AKK*(XNPM/2.D0+DLX))*DSIN(AKK*XNPP
     *                    /2.D0)
                      CST2=DCOS(AKK*(-XNMP/2.D0+DLX))*DSIN(AKK*XNMM
     *                    /2.D0)
                      CST3=DCOS(AKK*(XNMM/2.D0+DLX))*DSIN(AKK*XNMP
     *                    /2.D0)
                      CST4=DCOS(AKK*(-XNPP/2.D0+DLX))*DSIN(AKK*XNPM
     *                    /2.D0)
                      T1(1,K,J)=T1(1,K,J)+SV2*AASIN*CST1/TAPP1
                      T2(1,K,J)=T2(1,K,J)+SV1*AASIN*CST2/TAMM1
                      T3(1,K,J)=T3(1,K,J)+SV1*AASIN*CST3/TAMP1
                      T4(1,K,J)=T4(1,K,J)+SV2*AASIN*CST4/TAPM1
                      T1(2,K,J)=T1(2,K,J)+SV2*AASIN*CST1/TAPP2
                      T2(2,K,J)=T2(2,K,J)+SV1*AASIN*CST2/TAMM2
                      T3(2,K,J)=T3(2,K,J)+SV1*AASIN*CST3/TAMP2
                      T4(2,K,J)=T4(2,K,J)+SV2*AASIN*CST4/TAPM2
  31            CONTINUE
  30      CONTINUE
   1  CONTINUE
C
C
C     Evaluation of GS,GS1S2
C
C
      CZX=2.D0*(1.D0-ER)/((1.D0+ER)*(1.D0+E2)*(1.D0+0.5D0*E1))
      IF ((ER-1.D0).LT.0.005) CZX=0.D0
      CXX=1.D0
      CSX=(AK2-AKK2)*CXX/FA
      CSZ=AKK2*CZX/FA
      CAX=AKK*CXX/FA
      CAZ=AKK*CZX/FA
      DO 4 J=1,NOFF
           NJMAX=NOFFS(J)
```

```
          DO 62 N=1,NJMAX
                NP1=N+2
                N0=N+1
                NM1=N
                STX=-D1(1,NP1,J)+2.D0*YCOS*D1(1,N0,J)-D1(1,NM1,J)
     *             +2.D0*(T1(1,N,J)+T2(1,N,J)-T3(1,N,J)-T4(1,N,J))
                STZ=-D1(2,NP1,J)+2.D0*YCOS*D1(2,N0,J)-D1(2,NM1,J)
     *             +2.D0*(T1(2,N,J)+T2(2,N,J)-T3(2,N,J)-T4(2,N,J))
                MP2=N+4
                MP1=N+3
                M0=N+2
                MM1=N+1
                MM2=N
                SINP2=DSIN(AKK*FLOAT(N+1)*DLX)
                SINP1=DSIN(AKK*FLOAT(N)*DLX)
                SIN0=DSIN(AKK*FLOAT(N-1)*DLX)
                SINM1=DSIN(AKK*FLOAT(N-2)*DLX)
                SINM2=DSIN(AKK*FLOAT(N-3)*DLX)
                ATX=SINP2*S1(1,MP2,J)-4.D0*YCOS*SINP1*S1(1,MP1,J)
     *              +2.D0*(2.D0+CCS)*SIN0*S1(1,M0,J)-4.D0*YCOS
     *              *SINM1*S1(1,MM1,J)+SINM2*S1(1,MM2,J)
                ATZ=SINP2*S1(2,MP2,J)-4.D0*YCOS*SINP1*S1(2,MP1,J)
     *              +2.D0*(2.D0+CCS)*SIN0*S1(2,M0,J)-4.D0*YCOS
     *              *SINM1*S1(2,MM1,J)+SINM2*S1(2,MM2,J)
                AAX=-2.D0*(D2(1,NP1,J)-2.D0*YCOS*D2(1,N0,J)
     *              +D2(1,NM1,J))
                AAZ=-2.D0*(D2(2,NP1,J)-2.D0*YCOS*D2(2,N0,J)
     *              +D2(2,NM1,J))
                AX=ATX+AAX
                AZ=ATZ+AAZ
                ZW=W*(CSX*STX+CSZ*STZ+CAX*AX-CAZ*AZ)
                IF (J.EQ.1) GS(N)=ZW
                IF(J.GE.2) GS1S2(J,N)=ZW
   62      CONTINUE
   4  CONTINUE
      RETURN
      END
C...................................................................
C   This subroutine evaluates the higher order bessel functions using
C   the ascenting series expression or hankel's expansion.
C..................................................................
      SUBROUTINE BESS2 (BJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2),U(4),RBJ(50,2)
      COMMON/B01/BJ0,BJ1
      COMMON/BSS/ARG(10),X
C
      PI=3.141592653589
C
C     Evaluation of J0,J1
C
      CALL BSJ0(X)
      RBJ(1,2)=BJ0
      RBJ(2,2)=BJ1
C
      NCON=1
      N=IDINT(2.4D0*X)
      IF (N.LT.10) N=10
      IF (X.LT.3.D0) GO TO 10
C
C     EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS UP TO
C     ORDER LESS THEN THE ARGUMENT
C
      NIMAX=IDINT(X)-1
      IF (NIMAX.GT.9) NIMAX=9
      DO 1 I=2,NIMAX
```

```
           NJ1=I
           NJ2=I-1
           NB=I+1
           RBJ(NB,2)=FLOAT(2*NJ2)*RBJ(NJ1,2)/X-RBJ(NJ2,2)
 1    CONTINUE
      IF (NIMAX.EQ.9) GO TO 20
      NCON=NIMAX
C
C     DEBYE'S ASYMPTOTIC EXPANSION-EVALUATION OF JN
C
 10   DO 11 J=1,2
            JN=N-J+1
            XA=X/FLOAT(JN)
            XA=1.D0/XA
            XE=XA+DSQRT(XA*XA-1.D0)
            A=DLOG(XE)
            CTH=(XE+1.D0/XE)/(XE-1.D0/XE)
            CALL F(CTH,U)
            TNH=1.D0/CTH
            R1=DEXP(FLOAT(JN)*(TNH-A))
            R2=DSQRT(2.D0*PI*FLOAT(JN)*TNH)
            BN1=JN
            BN2=JN*JN
            BN3=BN2*JN
            BN4=BN3*JN
            RBJ(JN+1,2)=(R1/R2)*(1.D0+U(1)/BN1+U(2)/BN2+U(3)/BN3+
     *                 U(4)/BN4)
 11   CONTINUE
C
C     EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS WHEN X<10
C
      NJMAX=N-2-NCON
      DO 2 I=1,NJMAX
           NJB=N-I
           NJB1=NJB+1
           NJB2=NJB1+1
           RBJ(NJB,2)=2.D0*FLOAT(NJB)*RBJ(NJB1,2)/X-RBJ(NJB2,2)
 2    CONTINUE
 20   CONTINUE
      DO 3 I=1,9
           BJ(I,2)=RBJ(I,2)
 3    CONTINUE
      RETURN
      END
C...........................................................................
C...........................................................................
      SUBROUTINE BSJ0(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/B01/BJ0,BJ1
C
C
C    Evaluation of J0 using the series expansion given in
C    Abramowitz.
C
      PI=3.141592653589D0
      IF (X.GT.3.D0) GO TO 20
      X3=X/3.D0
      X32=X3*X3
      X34=X32*X32
      X36=X32*X34
      X38=X32*X36
      X310=X38*X32
      X312=X310*X32
      BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0*X36+
     *    0.0444479D0*X38-0.0039444D0*X310+0.00021000D0*X312
      BJ1=X*(0.5D0-0.56249985D0*X32+0.21093573D0*X34-0.03954289D0
```

```
     *    *X36+0.00443319D0*X38-0.00031761D0*X310+0.00001109D0
     *    *X312)
      GO TO 21
C
   20 X3=3.D0/X
      X32=X3*X3
      X33=X32*X3
      X34=X33*X3
      X35=X34*X3
      X36=X35*X3
      FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.00009512D0
     *    *X33+0.00137237D0*X34-0.00072805D0*X35+0.00014476D0*X36
      FJ1=0.79788456D0+0.00000156D0*X3+0.01659667D0*X32+0.00017105D0
     *    *X33-0.00249511D0*X34+0.00113653D0*X35-0.00020033D0*X36
      TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00262573D0
     *    *X33-0.00054125D0*X34-0.00029333D0*X35+0.00013558D0*X36
      TJ1=X-2.35619449D0+0.12499612D0*X3+0.00005650D0*X32-0.00637879D0
     *    *X33+0.00074348D0*X34+0.00079824D0*X35-0.00029166D0*X36
      WCON=DSQRT(1.D0/X)
      BJ0=WCON*FJ0*DCOS(TJ0)
      BJ1=WCON*FJ1*DCOS(TJ1)
  21  CONTINUE
      RETURN
      END
C..........................................................................
C..................................................................
      SUBROUTINE F(X,U)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(4)
      X2=X*X
      X3=X2*X
      X4=X3*X
      X5=X4*X
      X6=X5*X
      X7=X6*X
      X8=X7*X
      X9=X8*X
      X10=X9*X
      X11=X10*X
      X12=X11*X
C
      U(1)=(3.D0*X-5.D0*X3)/24.D0
      U(2)=(81.D0*X2-462.D0*X4+385.D0*X6)/1152.D0
      U(3)=(30375.D0*X3-369603.D0*X5+765765.D0*X7-425425.D0*X9)/
     *      414720.D0
      U(4)=(4465125.D0*X4-94121676.D0*X6+349922430.D0*X8-446185740.D0*
     *      X10+185910725.D0*X12)/39813120.D0
      RETURN
      END
C.........................................................................
C             SUBROUTINE DATA_SLOT
C     This subroutine gives all the data for integration used in
C     subroutine SLOT.FTN
C........................................................................
      SUBROUTINE DATA_SLOT
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
```

```
      COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XNT(40,3),
     *CNT(40),NDP,NTP,NSP
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(7,10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      PI=3.141592653589D0
C
C
      TPI=2.D0*PI
      TPI2=TPI*TPI
C     +-------------------+
C     | ERROR FUNCTIONS   |
C     +-------------------+
C
      A1=A*A/ER-TPI2
      A2=TPI2-TPI2/ER
      E1=0.5D0*A2/A1
      E2=ER*E1/(1.D0+ER)
      FA=DSQRT(1.D0+TPI2/A1)
C     +----------------------------------------------------+
C     |          Data for the poles                        |
C     |  IFIRST= 0 : dominant mode is TM wave (many poles) |
C     |          1 : dominant mode is TE wave (many poles) |
C     |          2 : only one TM surface wave              |
C     +----------------------------------------------------+
C   +---------------------------+
C   |  Data for the Integration |
C   +---------------------------+
      NK0=20
      NK0K=1
      MA=40
      NSER=10
C
      NPOINT=10
C-----------
C     Vector COAL
C-----------
      COAL(1)=0.0666713443D0
      COAL(2)=0.14945134915D0
      COAL(3)=0.21908636251D0
      COAL(4)=0.26926671931D0
      COAL(5)=0.29552422471D0
      COAL(6)=COAL(5)
      COAL(7)=COAL(4)
      COAL(8)=COAL(3)
      COAL(9)=COAL(2)
      COAL(10)=COAL(1)
C     ------------
C     Vector POINT
C     ------------
      POINT(1)=0.973906528517D0
      POINT(2)=0.865063366688D0
      POINT(3)=0.679409568299D0
      POINT(4)=0.433395394129D0
      POINT(5)=0.148874338981D0
      POINT(6)=-POINT(5)
      POINT(7)=-POINT(4)
      POINT(8)=-POINT(3)
      POINT(9)=-POINT(2)
      POINT(10)=-POINT(1)
C
      MPOINT=5
C     ------------
```

```
C     Vector BCOAL
C     ------------
      BCOAL(1)=0.2369268851D0
      BCOAL(2)=0.4786286705D0
      BCOAL(3)=0.5688888888D0
      BCOAL(4)=BCOAL(2)
      BCOAL(5)=BCOAL(1)
C     -------------
C     Vector BPOINT
C     -------------
      BPOINT(1)=0.9061798459D0
      BPOINT(2)=0.5384693101D0
      BPOINT(3)=0.D0
      BPOINT(4)=-BPOINT(2)
      BPOINT(5)=-BPOINT(1)
C     ------------------
C     Single integration
C     ------------------
C
      NSP=31
      RS1=0.99708748181D0
      RS2=0.98468590966D0
      RS3=0.96250392509D0
      RS4=0.93075699789D0
      RS5=0.88976002994D0
      RS6=0.83992032014D0
      RS7=0.78173314841D0
      RS8=0.71577678458D0
      RS9=0.64270672292D0
      RS10=0.56324916140D0
      RS11=0.47819378204D0
      RS12=0.38838590160D0
      RS13=0.29471806998D0
      RS14=0.19812119933D0
      RS15=0.09955531215D0
      RS16=0.D0
C
      XNS(1)=RS1
      XNS(2)=RS2
      XNS(3)=RS3
      XNS(4)=RS4
      XNS(5)=RS5
      XNS(6)=RS6
      XNS(7)=RS7
      XNS(8)=RS8
      XNS(9)=RS9
      XNS(10)=RS10
      XNS(11)=RS11
      XNS(12)=RS12
      XNS(13)=RS13
      XNS(14)=RS14
      XNS(15)=RS15
      XNS(16)=RS16
      XNS(17)=-RS15
      XNS(18)=-RS14
      XNS(19)=-RS13
      XNS(20)=-RS12
      XNS(21)=-RS11
      XNS(22)=-RS10
      XNS(23)=-RS9
      XNS(24)=-RS8
      XNS(25)=-RS7
      XNS(26)=-RS6
      XNS(27)=-RS5
      XNS(28)=-RS4
      XNS(29)=-RS3
```

```
      XNS(30)=-RS2
      XNS(31)=-RS1
C
      CNS(1)=0.0074708315792D0
      CNS(2)=0.0173186207903D0
      CNS(3)=0.0270090191849D0
      CNS(4)=0.0364322739123D0
      CNS(5)=0.0454937075272D0
      CNS(6)=0.0541030824249D0
      CNS(7)=0.0621747865610D0
      CNS(8)=0.0696285832354D0
      CNS(9)=0.0763903865987D0
      CNS(10)=0.0823929917615D0
      CNS(11)=0.0875767406084D0
      CNS(12)=0.0918901138936D0
      CNS(13)=0.0952902429123D0
      CNS(14)=0.0977433353863D0
      CNS(15)=0.0992250112266D0
      CNS(16)=0.0997205447934D0
      CNS(17)=CNS(15)
      CNS(18)=CNS(14)
      CNS(19)=CNS(13)
      CNS(20)=CNS(12)
      CNS(21)=CNS(11)
      CNS(22)=CNS(10)
      CNS(23)=CNS(9)
      CNS(24)=CNS(8)
      CNS(25)=CNS(7)
      CNS(26)=CNS(6)
      CNS(27)=CNS(5)
      CNS(28)=CNS(4)
      CNS(29)=CNS(3)
      CNS(30)=CNS(2)
      CNS(31)=CNS(1)
C
C
C
C
C     2)  Double Integration
C         ------------------
C
      NDP=16
      R1=DSQRT((15.D0-2.D0*DSQRT(30.D0))/35.D0)
      R2=-R1
      S1=DSQRT((15.D0+2.D0*DSQRT(30.D0))/35.D0)
      S2=-S1
      A1=4.D0*(59.D0+6.D0*DSQRT(30.D0))/864.D0
      A2=4.D0*(59.D0-6.D0*DSQRT(30.D0))/864.D0
      A3=4.D0*49.D0/864.D0
C
      XND(1,1)=R1
      XND(1,2)=R1
      CND(1)=A1
C
      XND(2,1)=R2
      XND(2,2)=R1
      CND(2)=A1
C
      XND(3,1)=R1
      XND(3,2)=R2
      CND(3)=A1
C
      XND(4,1)=R2
      XND(4,2)=R2
      CND(4)=A1
C
      XND(5,1)=S1
```

```
      XND(5,2)=S1
      CND(5)=A2
C
      XND(6,1)=S1
      XND(6,2)=S2
      CND(6)=A2
C
      XND(7,1)=S2
      XND(7,2)=S1
      CND(7)=A2
C
      XND(8,1)=S2
      XND(8,2)=S2
      CND(8)=A2
C
      XND(9,1)=R1
      XND(9,2)=S1
      CND(9)=A3
C
      XND(10,1)=R1
      XND(10,2)=S2
      CND(10)=A3
C
      XND(11,1)=S1
      XND(11,2)=R1
      CND(11)=A3
C
      XND(12,1)=S2
      XND(12,2)=R1
      CND(12)=A3
C
      XND(13,1)=R2
      XND(13,2)=S1
      CND(13)=A3
C
      XND(14,1)=R2
      XND(14,2)=S2
      CND(14)=A3
C
      XND(15,1)=S1
      XND(15,2)=R2
      CND(15)=A3
C
      XND(16,1)=S2
      XND(16,2)=R2
      CND(16)=A3
C
C     3)  Triple Integration
C         ------------------
C
      NTP=34
      RS1=0.9317380000D0
      RS2=-RS1
      UU1=0.9167441779D0
      UU2=-UU1
      SS1=0.4086003800D0
      SS2=-SS1
      TT1=0.7398529500D0
      TT2=-TT1
      B1=8.D0*0.03558180896D0
      B2=8.D0*0.01247892770D0
      B3=8.D0*0.05286772991D0
      B4=8.D0*0.02672752182D0
C
      XNT(1,1)=RS1
      XNT(1,2)=0.D0
```

```
      XNT(1,3)=0.D0
      CNT(1)=B1
C
      XNT(2,1)=RS2
      XNT(2,2)=0.D0
      XNT(2,3)=0.D0
      CNT(2)=B1
C
      XNT(3,1)=0.D0
      XNT(3,2)=RS1
      XNT(3,3)=0.D0
      CNT(3)=B1
C
      XNT(4,1)=0.D0
      XNT(4,2)=RS2
      XNT(4,3)=0.D0
      CNT(4)=B1
C
      XNT(5,1)=0.D0
      XNT(5,2)=0.D0
      XNT(5,3)=RS1
      CNT(5)=B1
C
      XNT(6,1)=0.D0
      XNT(6,2)=0.D0
      XNT(6,3)=RS2
      CNT(6)=B1
C
      XNT(7,1)=UU1
      XNT(7,2)=UU1
      XNT(7,3)=0.D0
      CNT(7)=B2
C
      XNT(8,1)=UU2
      XNT(8,2)=UU1
      XNT(8,3)=0.D0
      CNT(8)=B2
C
      XNT(9,1)=UU1
      XNT(9,2)=UU2
      XNT(9,3)=0.D0
      CNT(9)=B2
C
      XNT(10,1)=UU2
      XNT(10,2)=UU2
      XNT(10,3)=0.D0
      CNT(10)=B2
C
      XNT(11,1)=UU1
      XNT(11,2)=0.D0
      XNT(11,3)=UU1
      CNT(11)=B2
C
      XNT(12,1)=UU1
      XNT(12,2)=0.D0
      XNT(12,3)=UU2
      CNT(12)=B2
C
      XNT(13,1)=UU2
      XNT(13,2)=0.D0
      XNT(13,3)=UU1
      CNT(13)=B2
C
      XNT(14,1)=UU2
      XNT(14,2)=0.D0
      XNT(14,3)=UU2
```

```
      CNT(14)=B2
C
      XNT(15,1)=0.D0
      XNT(15,2)=UU1
      XNT(15,3)=UU1
      CNT(15)=B2
C
      XNT(16,1)=0.D0
      XNT(16,2)=UU1
      XNT(16,3)=UU2
      CNT(16)=B2
C
      XNT(17,1)=0.D0
      XNT(17,2)=UU2
      XNT(17,3)=UU1
      CNT(17)=B2
C
      XNT(18,1)=0.D0
      XNT(18,2)=UU2
      XNT(18,3)=UU2
      CNT(18)=B2
C
      XNT(19,1)=SS1
      XNT(19,2)=SS1
      XNT(19,3)=SS1
      CNT(19)=B3
C
      XNT(20,1)=SS1
      XNT(20,2)=SS1
      XNT(20,3)=SS2
      CNT(20)=B3
C
      XNT(21,1)=SS1
      XNT(21,2)=SS2
      XNT(21,3)=SS1
      CNT(21)=B3
C
      XNT(22,1)=SS1
      XNT(22,2)=SS2
      XNT(22,3)=SS2
      CNT(22)=B3
C
      XNT(23,1)=SS2
      XNT(23,2)=SS1
      XNT(23,3)=SS1
      CNT(23)=B3
C
      XNT(24,1)=SS2
      XNT(24,2)=SS1
      XNT(24,3)=SS2
      CNT(24)=B3
C
      XNT(25,1)=SS2
      XNT(25,2)=SS2
      XNT(25,3)=SS1
      CNT(25)=B3
C
      XNT(26,1)=SS2
      XNT(26,2)=SS2
      XNT(26,3)=SS2
      CNT(26)=B3
C
      XNT(27,1)=TT1
      XNT(27,2)=TT1
      XNT(27,3)=TT1
      CNT(27)=B4
```

```
C
      XNT(28,1)=TT1
      XNT(28,2)=TT1
      XNT(28,3)=TT2
      CNT(28)=B4
C
      XNT(29,1)=TT1
      XNT(29,2)=TT2
      XNT(29,3)=TT1
      CNT(29)=B4
C
      XNT(30,1)=TT1
      XNT(30,2)=TT2
      XNT(30,3)=TT2
      CNT(30)=B4
C
      XNT(31,1)=TT2
      XNT(31,2)=TT1
      XNT(31,3)=TT1
      CNT(31)=B4
C
      XNT(32,1)=TT2
      XNT(32,2)=TT1
      XNT(32,3)=TT2
      CNT(32)=B4
C
      XNT(33,1)=TT2
      XNT(33,2)=TT2
      XNT(33,3)=TT1
      CNT(33)=B4
C
      XNT(34,1)=TT2
      XNT(34,2)=TT2
      XNT(34,3)=TT2
      CNT(34)=B4
C
      RETURN
      END
```

```
################              apollo domain               ################
################                                          ################
################                                          ################
################               CAEN/Apollo                ################
```

```
K    K     A    TTTTTTT EEEEEEE H     H   III
K   K     A A      T    E       H     H    I
K  K     A   A     T    E       H     H    I
KKK     A     A    T    EEEEE   HHHHHHH    I
K  K    AAAAAAA    T    E       H     H    I
K   K   A     A    T    E       H     H    I
K    K  A     A    T    EEEEEEE H     H   III
```

```
  aa    rrrrr   rrrrr     aa    n    n   gggg   eeeeee          m    m  u    u   ttttt  u    u    aa    l                       ffffff   ttttt
 a  a   r    r  r    r   a  a   nn   n  g    g  e               mm  mm  u    u     t    u    u   a  a   l               f          t
a    a  r    r  r    r  a    a  n n  n  g       eeeee           m mm m  u    u     t    u    u  a    a  l               fffff      t
aaaaaa  rrrrr   rrrrr   aaaaaa  n  n n  g  ggg  e               m    m  u    u     t    u    u  aaaaaa  l        ...    f          t
a    a  r   r   r   r   a    a  n   nn  g    g  e               m    m  u    u     t    u    u  a    a  l        ...    f          t
a    a  r    r  r    r  a    a  n    n   gggg   eeeeee  ______  m    m   uuuu      t     uuuu   a    a  llllll   ...    f          t
```

//tera/users/katehi/tape/arrange_mutual.ftn

```
################                                          ################
################     LAST MODIFIED ON:  88/10/25  3:00 PM ################
################         FILE PRINTED:  89/01/25  1:40 PM ################
################                                          ################
```

```
C*****************************************************************
C     The name of this file is ......... ARRANGE_MUTUAL................
C*****************************************************************
      SUBROUTINE ARRANGE_MUTUAL
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/MAN/IBMATR(260,260)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C.................................................................
C                 DATA
C.................................................................
      NOEL1=NSL(1)
      NOEL2=NSL(2)
      NS12=NXOFF(2)-NXOFF(1)
      NOR=NOEL1+NOEL2
      WRITE(6,222) NOEL1,NOEL2,NS12
 222  FORMAT (10X,'NOEL1=',I4/10X,'NOEL2=',I4/10X,'NS12=',
     *I4,//////)
C
C     .....First Diagonal Matrix.........
C
      IMIN=1
      IMAX=NOEL1
      DO 4 I=IMIN,IMAX
           IXN=0
           DO 5 KI=I,IMAX
                IXN=IXN+1
                IBMATR(IXN,KI)=I
                IBMATR(KI,IXN)=IBMATR(IXN,KI)
  5        CONTINUE
  4   CONTINUE
C
C     .......Second Diagonal Matrix ........
C
C
      INI=NOEL1
      IMIN=NOEL1+1
      IMAX=NOEL1+NOEL2
      DO 6 I=IMIN,IMAX
           IXN=INI
           DO 7 KI=I,IMAX
                IXN=IXN+1
                IBMATR(IXN,KI)=I-INI
                IBMATR(KI,IXN)=IBMATR(IXN,KI)
   7       CONTINUE
   6  CONTINUE
C
C     ...1...  First off-diagonal matrix
C
C        1)   Upper Part
C
      IAI=NOEL1-NOEL2
      IMI=IABS(IAI)+1
      IMIN=NOEL1+1
      IMAX=NOEL1+NOEL2
      DO 12 I=IMIN,IMAX
            IXN=0
            LXN=IABS(NS12+I-IMIN)+1
            IF (IAI.LT.0) GO TO 13
                   KIMIN=I
                   KIMAX=IMAX
                   GO TO 14
```

```
  13        KIMIN=I
            KIMAX=I+NOEL1
            IF ((I-IMIN+1).GE.IMI) KIMAX=IMAX
  14        DO 15 KI=KIMIN,KIMAX
                  IXN=IXN+1
                  IBMATR(IXN,KI)=LXN
                  IBMATR(KI,IXN)=IBMATR(IXN,KI)
  15        CONTINUE
  12  CONTINUE
C
C   ......2) lower Part .............
C
      IMIN=2
      IMAX=NOEL1
      DO 16 I=IMIN,IMAX
            IXN=I-1
            LXN=IABS(NS12-I+IMIN-1)+1
            IF (IAI.GT.0) GO TO 17
                  KIMIN=NOEL1+1
                  KIMAX=2*NOEL1-I+IMIN-1
                  GO TO 18
  17        KIMIN=NOEL1+1
            KIMAX=NOEL1+NOEL2
            IIMI=I-IMIN+2
            IF (IIMI.GE.IMI) KIMAX=NOEL1+NOEL2-IIMI+IMI
  18        DO 19 KI=KIMIN,KIMAX
                  IXN=IXN+1
                  IBMATR(IXN,KI)=LXN
                  IBMATR(KI,IXN)=IBMATR(IXN,KI)
  19        CONTINUE
  16  CONTINUE
      RETURN
      END
```

apollo domain

CAEN/Apollo

```
K    K     A    TTTTTTT EEEEEEE H     H  III
K   K     A A      T    E       H     H   I
K  K     A   A     T    E       H     H   I
KKK     A     A    T    EEEEE   HHHHHHH   I
K  K    AAAAAAA    T    E       H     H   I
K   K   A     A    T    E       H     H   I
K    K  A     A    T    EEEEEEE H     H  III
```

```
rrrrr   u    u  n    n          m    m  u    u  ttttt   u    u    aa    l
r    r  u    u  nn   n          mm  mm  u    u    t     u    u   a  a   l
r    r  u    u  n n  n          m mm m  u    u    t     u    u  a    a  l
rrrrr   u    u  n  n n          m    m  u    u    t     u    u  aaaaaa  l
r   r   u    u  n   nn          m    m  u    u    t     u    u  a    a  l
r    r   uuuu   n    n _______  m    m   uuuu     t      uuuu   a    a  llllll
```

//tera/users/katehi/tape/run_mutual

LAST MODIFIED ON: 89/01/25 11:47 AM
FILE PRINTED: 89/01/25 1:41 PM

```
BIND SLOT_DESIGN.BIN MUTUAL_SLOT.BIN POLES_MUTUAL.BIN YIJ_DIEL_MUTUAL.BIN  ARRANGE_MUTUAL.BIN -
```

```
*** WARNING ***
1 line was too long and was truncated.
*** WARNING ***
```

apollo domain

CAEN/Apollo

```
K    K     A    TTTTTTT EEEEEEE H     H   III
K   K     A A      T    E       H     H    I
K  K     A   A     T    E       H     H    I
KKK     A     A    T    EEEEE   HHHHHHH    I
K  K    AAAAAAA    T    E       H     H    I
K   K   A     A    T    E       H     H    I
K    K  A     A    T    EEEEEEE H     H   III
```

```
 oooo   u    u   ttttt          w    w    aa    v    v  eeeeee          m    m  u    u   ttttt  u    u    aa    l
o    o  u    u     t            w    w   a  a   v    v  e               mm  mm  u    u     t    u    u   a  a   l
o    o  u    u     t            w    w  a    a  v    v  eeeee           m mm m  u    u     t    u    u  a    a  l
o    o  u    u     t            w ww w  aaaaaa  v    v  e               m    m  u    u     t    u    u  aaaaaa  l
o    o  u    u     t            ww  ww  a    a   v  v   e               m    m  u    u     t    u    u  a    a  l
 oooo    uuuu      t    ______  w    w  a    a    vv    eeeeee  ______  m    m   uuuu      t     uuuu   a    a  llllll
```

//tera/users/katehi/tape/out_wave_mutual

```
LAST MODIFIED ON:  89/01/26  9:33 AM
    FILE PRINTED:  89/01/26  9:39 AM
```

```
          Dielectric Constant of the Substrate
           0.2620000E+01


          Substrate Thickness
           0.2544000E-01


          Conductor Thickness
           0.1000000E-04


          Number of Slots
          NSLOTS=   2


          Transverse Offsets of the Slots
          YOFF(1)= 0.3166000E+00
          YOFF( 2)= 0.6086400E+00



          Longitudinal Offset of the Slots
          NXOFF(1)=   1
          NXOFF( 2)=  57



          Slot Widths
          WS(1)= 0.4800000E-01
          WS( 2)= 0.4800000E-01



          Slots Excess Widths
          WSDELTA= 0.0000000E+00
          WSDELTA( 2)= 0.0000000E+00



          Subsection Length
           0.1250000E-01


          DLX_RES( 2)= 0.1250000E-01



          Lower Limit of Tail Contribution
           0.1000000E+03


          Number of Points on Each Slot including the ends
          NSL(1)=  29
          NSL( 2)=  29



          Error in the evaluation of the series
          ERROR= 0.1000000E-05


          Normalization Constant
           0.1000000E+01

  L=    1  RCUR(L)= 0.0000000E+00  AICUR(L)= 0.0000000E+00
```

```
  L=   2  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
  L=   3  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
  L=   4  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
  L=   5  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
  L=   6  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
  L=   7  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
  L=   8  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
  L=   9  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
  L=  10  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
  L=  11  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
  L=  12  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
  L=  13  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
  L=  14  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
  L=  15  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
  L=  16  RCUR(L)= 0.1000000E+01  AICUR(L)= 0.0000000E+00
  L=  17  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
  L=  18  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
  L=  19  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
  L=  20  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
  L=  21  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
  L=  22  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
  L=  23  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
  L=  24  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
  L=  25  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
  L=  26  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
  L=  27  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
  L=  28  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
  L=  29  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
  L=  30  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
  L=  31  RCUR(L)= 0.7932658E-12  AICUR(L)= 0.0000000E+00
  L=   1  RCUR(L)= 0.0000000E+00  AICUR(L)= 0.0000000E+00
  L=   2  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
  L=   3  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
  L=   4  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
  L=   5  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
  L=   6  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
  L=   7  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
  L=   8  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
  L=   9  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
  L=  10  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
  L=  11  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
  L=  12  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
  L=  13  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
  L=  14  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
  L=  15  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
  L=  16  RCUR(L)= 0.1000000E+01  AICUR(L)= 0.0000000E+00
  L=  17  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
  L=  18  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
  L=  19  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
  L=  20  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
  L=  21  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
  L=  22  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
  L=  23  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
  L=  24  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
  L=  25  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
  L=  26  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
  L=  27  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
  L=  28  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
  L=  29  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
  L=  30  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
  L=  31  RCUR(L)= 0.7932658E-12  AICUR(L)= 0.0000000E+00


          Number of elements to be evaluated for the mutual interactions
```

```
        I=   1  J=   2  NSSL=  87 0.2920400E+00


        Offsets for the dielectric layer and number of corresponding elements

        I=   1  OFFSET= 0.2920400E+00  NOFFS=  87
        I=   2  OFFSET= 0.0000000E+00  NOFFS=   0


        SLOTS and corresponding offsets in the dielectric

        I=   1  J=   2  INSS=   1  OFFSET= 0.2920400E+00



        Max number of offsets in the dielectric
        NOFF=   1




        No TE waves excited in the substrate




        There are   1 TM waves excited in the substrate

           1   0.631427169E+01


        Contribution to admittance from the dielectric




    OFFSET #   1



        Interactions between slots 1 and  2


IJ=    1 YSD=-0.7824835E-06   0.1558373E-05
IJ=    2 YSD=-0.7817870E-06   0.1552693E-05
IJ=    3 YSD=-0.7796996E-06   0.1535848E-05
IJ=    4 YSD=-0.7762282E-06   0.1508451E-05
IJ=    5 YSD=-0.7713833E-06   0.1471411E-05
IJ=    6 YSD=-0.7651809E-06   0.1425862E-05
IJ=    7 YSD=-0.7576400E-06   0.1373079E-05
IJ=    8 YSD=-0.7487855E-06   0.1314513E-05
IJ=    9 YSD=-0.7386444E-06   0.1251922E-05
IJ=   10 YSD=-0.7272494E-06   0.1187477E-05
IJ=   11 YSD=-0.7146360E-06   0.1123704E-05
IJ=   12 YSD=-0.7008439E-06   0.1063042E-05
```

```
 IJ=  13 YSD=-0.6859163E-06   0.1006943E-05
 IJ=  14 YSD=-0.6699007E-06   0.9549244E-06
 IJ=  15 YSD=-0.6528462E-06   0.9043670E-06
 IJ=  16 YSD=-0.6348060E-06   0.8517802E-06
 IJ=  17 YSD=-0.6158364E-06   0.7954015E-06
 IJ=  18 YSD=-0.5959959E-06   0.7377696E-06
 IJ=  19 YSD=-0.5753450E-06   0.6858309E-06
 IJ=  20 YSD=-0.5539477E-06   0.6471066E-06
 IJ=  21 YSD=-0.5318685E-06   0.6235971E-06
 IJ=  22 YSD=-0.5091747E-06   0.6079404E-06
 IJ=  23 YSD=-0.4859345E-06   0.5864324E-06
 IJ=  24 YSD=-0.4622173E-06   0.5488644E-06
 IJ=  25 YSD=-0.4380942E-06   0.4982312E-06
 IJ=  26 YSD=-0.4136361E-06   0.4511244E-06
 IJ=  27 YSD=-0.3889146E-06   0.4254873E-06
 IJ=  28 YSD=-0.3640019E-06   0.4242090E-06
 IJ=  29 YSD=-0.3389715E-06   0.4295155E-06
 IJ=  30 YSD=-0.3138884E-06   0.4162005E-06
 IJ=  31 YSD=-0.2888288E-06   0.3744660E-06
 IJ=  32 YSD=-0.2638660E-06   0.3213586E-06
 IJ=  33 YSD=-0.2390565E-06   0.2871093E-06
 IJ=  34 YSD=-0.2144823E-06   0.2865827E-06
 IJ=  35 YSD=-0.1902016E-06   0.3027549E-06
 IJ=  36 YSD=-0.1662792E-06   0.3012312E-06
 IJ=  37 YSD=-0.1427780E-06   0.2640741E-06
 IJ=  38 YSD=-0.1197573E-06   0.2094553E-06
 IJ=  39 YSD=-0.9727464E-07   0.1748476E-06
 IJ=  40 YSD=-0.7538512E-07   0.1785652E-06
 IJ=  41 YSD=-0.5414028E-07   0.1994006E-06
 IJ=  42 YSD=-0.3358976E-07   0.1977392E-06
 IJ=  43 YSD=-0.1377886E-07   0.1577346E-06
 IJ=  44 YSD= 0.5249304E-08   0.1053305E-06
 IJ=  45 YSD= 0.2345550E-07   0.8094082E-07
 IJ=  46 YSD= 0.4080386E-07   0.9514656E-07
 IJ=  47 YSD= 0.5726202E-07   0.1163907E-06
 IJ=  48 YSD= 0.7280136E-07   0.1056789E-06
 IJ=  49 YSD= 0.8739682E-07   0.6046866E-07
 IJ=  50 YSD= 0.1010273E-06   0.1792455E-07
 IJ=  51 YSD= 0.1136752E-06   0.1287583E-07
 IJ=  52 YSD= 0.1253268E-06   0.3790763E-07
 IJ=  53 YSD= 0.1359724E-06   0.5182596E-07
 IJ=  54 YSD= 0.1456057E-06   0.2735760E-07
 IJ=  55 YSD= 0.1542244E-06  -0.1711874E-07
 IJ=  56 YSD= 0.1618298E-06  -0.3917364E-07
 IJ=  57 YSD= 0.1684268E-06  -0.2223567E-07
 IJ=  58 YSD= 0.1740240E-06   0.5034281E-08
 IJ=  59 YSD= 0.1786332E-06   0.3011337E-08
 IJ=  60 YSD= 0.1822699E-06  -0.3172977E-07
 IJ=  61 YSD= 0.1849526E-06  -0.6266885E-07
 IJ=  62 YSD= 0.1867033E-06  -0.5750394E-07
 IJ=  63 YSD= 0.1875463E-06  -0.2691218E-07
 IJ=  64 YSD= 0.1875094E-06  -0.1115973E-07
 IJ=  65 YSD= 0.1866229E-06  -0.3105652E-07
 IJ=  66 YSD= 0.1849196E-06  -0.6301207E-07
 IJ=  67 YSD= 0.1824344E-06  -0.6841788E-07
 IJ=  68 YSD= 0.1792048E-06  -0.4100843E-07
 IJ=  69 YSD= 0.1752701E-06  -0.1404192E-07
 IJ=  70 YSD= 0.1706712E-06  -0.1890794E-07
 IJ=  71 YSD= 0.1654507E-06  -0.4660967E-07
 IJ=  72 YSD= 0.1596529E-06  -0.5960180E-07
 IJ=  73 YSD= 0.1533227E-06  -0.3873151E-07
 IJ=  74 YSD= 0.1465064E-06  -0.6675805E-08
 IJ=  75 YSD= 0.1392509E-06   0.9264340E-09
 IJ=  76 YSD= 0.1316037E-06  -0.2001843E-07
 IJ=  77 YSD= 0.1236126E-06  -0.3761033E-07
 IJ=  78 YSD= 0.1153255E-06  -0.2454465E-07
```

```
 IJ=   79 YSD= 0.1067905E-06   0.7759468E-08
 IJ=   80 YSD= 0.9805507E-07   0.2417755E-07
 IJ=   81 YSD= 0.8916630E-07   0.1041008E-07
 IJ=   82 YSD= 0.8017092E-07  -0.9506152E-08
 IJ=   83 YSD= 0.7111441E-07  -0.4317599E-08
 IJ=   84 YSD= 0.6204162E-07   0.2479453E-07
 IJ=   85 YSD= 0.5299594E-07   0.4632329E-07
 IJ=   86 YSD= 0.4401960E-07   0.3890204E-07
 IJ=   87 YSD= 0.3515319E-07   0.1814317E-07
          Interactions between slots 2 and  2


          NOEL1=  31
          NOEL2=  31
          NS12=  47







  LONGITUDINAL DISTANCE = 0.5875000E+00


          SUM_MD= 0.1292908E-04      0.3637009E-04


          Z12_MD=-0.9187551E+00  -0.2584500E+01
```