TECHNICAL REPORT

Dielectric-Covered, Cavity-Backed Aper tures

Pisti B. Katehi

Department of Electrical Engineering and Computer Science

The University of Michigan
Ann Arbor, MI

RL-842 = RL-842

June 1987
1. General formulation of the problem

An important special case of a cavity-backed aperture is that where the cavity is a finite rectangular waveguide. The aperture exists at one end of the the cylindrical waveguide and is covered by a dielectric slab. The other end of the waveguide is completely covered by a conductor. The cavity can then be viewed as a short-circuited waveguide and waveguide theory can be applied. Figure 1 represents a typical problem, where the excitation is by an incident plane wave from the half-space region. The equivalence principle is used to divide the problem into two equivalent problems as shown in figure 2. In region a, the field is produced by the incident field \( E^i, H^i \) and the equivalent magnetic current

\[
\vec{M} = \hat{n} \times \vec{E}
\]

(1)

over the aperture region, with the aperture covered by an electric conductor. In region c, the field is produced by the equivalent magnetic current \( \vec{M} \) over the aperture region, with the aperture covered by an electric conductor. The fact that the equivalent current in region c is the negative of that in region b ensures that the tangential component of the electric field is continuous across the aperture. The remaining boundary condition to be applied is continuity of the tangential component of the magnetic field across the aperture.

The tangential component \( H_y \) of the magnetic field in region b over the aperture is the sum of that due to the impressed sources
Figure 1
(a) Equivalent problem b

(b) Equivalent problem c

Figure 2
\( H_t \) plus that due to the equivalent source \( \mathbf{M} \), denoted by \( H_t(M) \).

Therefore

\[
\mathbf{H}_t = \mathbf{H}_t + H_t(M)
\]  

(2)

Both \( H_t \) and \( H_t(M) \) are computed with a conductor covering the aperture. In region \( \mathbf{c} \) the equivalent source \( -\mathbf{M} \) is the only source. Hence, the tangential component of magnetic field in region \( \mathbf{c} \) over the aperture is

\[
\mathbf{H}_t = \mathbf{H}_t(-\mathbf{M}) = -H_t(M)
\]  

(3)

Since \( H_t \) should be equal to \( \mathbf{H}_t \), equations (2) and (3) give

\[
\mathbf{H}_t(M) + H_t(M) = -\mathbf{H}_t
\]  

(4)

or

\[
\mathbf{n} \times \left( \mathbf{H}_t(M) + H_t(M) \right) = -\mathbf{n} \times \mathbf{H}_t
\]  

(5)

Equation (5) is the basic operator equation for determining the equivalent magnetic current \( \mathbf{M} \).

The magnetic fields \( \mathbf{H}_t \) and \( \mathbf{H} \) are given in terms of the magnetic current \( \mathbf{M} \) as shown in the following two integral equations.

\[
\mathbf{H}_t = \iint_{\text{slot}} \frac{j}{\omega \varepsilon \mu_d} \left[ k_d \mathbf{I} + \nabla \times \mathbf{\nabla} \right] \cdot F(\mathbf{r}/\mathbf{r}') \cdot M(\mathbf{r}') \, ds'
\]  

(6)

and

\[
\mathbf{H} = \iint_{\text{slot}} \frac{j}{\omega \varepsilon \mu_c} \left[ k_c \mathbf{I} + \nabla \times \mathbf{\nabla} \right] \cdot G(\mathbf{r}/\mathbf{r}') \cdot M(\mathbf{r}') \, ds'
\]  

(7)
In equation (6), $\varepsilon_d$ is the dielectric constant of the dielectric substrate, $\mu_d$ is the magnetic permeability and $F(r/r')$ is the dyadic Green's function for the equivalent problem of Fig. 2a. Similarly, $\varepsilon_c$ is the dielectric constant of the material in the cavity, $\mu_c$ is the magnetic permeability and $G(r/r')$ is the dyadic Green's function for the equivalent problem b. In view of (6) and (7), equation (5) gives the following coupled integral equations

$$
\iint_{\text{slot}} \left[ K_{xx}(r/r') M_x(r') + K_{xy}(r/r') M_y(r') \right] ds' = -H_x^i \quad (z=-h) \quad (13)
$$

and

$$
\iint_{\text{slot}} \left[ K_{yx}(r/r') M_x(r') + K_{yy}(r/r') M_y(r') \right] ds' = -H_y^i \quad (z=-h) \quad (14)
$$

where

$$
K_{ij}(r'/r) = j\omega \hat{u} \cdot \left\{ \frac{\varepsilon}{(F + G)} + \nabla \nabla \cdot \left( \frac{\varepsilon}{k_d^2} \frac{\varepsilon_c}{k_c^2} \right) \right\} \hat{J} \quad (15)
$$

and $i, j = x, y$.

For a thin slot which is open at the center of the cavity wall as shown in Figure 3, the transverse component of the electric field is much smaller than the longitudinal $E_x<<E_y$. This results in one directional magnetic current and therefore a single integral equation:

$$
\iint_{\text{slot}} K_{xx}(r/r') M_x(r') ds' = -H_x^i \quad (z=-h) \quad (16)
$$
In this case, \( K_{xx}(\textbf{r}/\textbf{r}') \) takes a simpler form as shown below

\[
K_{xx} = \left( F_{xx} + G_{xx} \right) + \frac{\partial}{\partial x^2} \left( \frac{F_{xx}}{k_d^2} + \frac{G_{xx}}{k_c^2} \right) + \frac{\partial}{\partial x \partial x} \frac{F_{zx}}{k_d^2} \tag{17}
\]

with [1]

\[
F_{xx} = -\frac{j \omega e_d}{2 \pi k_d^2} \int_0^\infty J_0 \left( \lambda |\textbf{r}-\textbf{r}'| \right) \frac{u \text{cosh}(uz) - \varepsilon_{rd} u \text{sinh}(uz)}{\varepsilon_{rd} u \text{cosh}(uh) + u \text{sinh}(uh)} \frac{\lambda}{u} \, d\lambda \tag{18}
\]

\[
F_{zx} = -\frac{j \omega e_d (1 - \varepsilon_{rd})}{2 \pi k_d^2} \cos \phi \int_0^\infty J_1 \left( \lambda |\textbf{r}-\textbf{r}'| \right) \frac{\lambda^2}{\varepsilon_{rd} u \text{cosh}(uh) + u \text{sinh}(uh)} \frac{\sinh [u(h+z)]}{u \text{cosh}(uh) + u \text{o} \text{sinh}(uh)} \, d\lambda \tag{19}
\]

and (Appendix A)

\[
G_{xx}(\textbf{r}/\textbf{r}') = j \frac{4 \omega e_c}{ab} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \varepsilon_n \varepsilon_m \frac{1}{k_z \text{sin}(k_c)} \sin \left( \frac{n \pi}{a} (x + x_o) \right) \sin \left( \frac{n \pi}{a} (x' + x_o) \right) \cdot \cos \left( \frac{m \pi}{b} (y - y_o) \right) \cos \left( \frac{m \pi}{b} (y' - y_o) \right) \cdot \cos \left[ k_z (c + z + h) \right]. \tag{20}
\]
2. Method of Moments

To obtain the unknown magnetic current \( M_x(r') \) in equation (16), we apply the method of moments. The unknown current is assumed to be in the form:

\[
M_x(r') = \sum_n V_n f_n(x') g(y')
\]  

(21)

where

\[
g(y') = \frac{2}{\pi w} \int_{-w/2}^{w/2} \frac{1}{\sqrt{1 - \left(\frac{2y'}{w}\right)^2}}
\]  

(22)

and \( \{f_n(x'), n=1, 2..., N\} \) is a set of expansion functions overlapping in nature. These expansion functions are given by

\[
f_n(x') = \begin{cases} 
\frac{\sin[k(x'-x_{n-1})]}{\sin(kl_x)} & X_{n-1} \leq x' \leq X_n \\
\frac{\sin[k(x_{n+1} - x')]}{\sin(kl_x)} & X_n \leq x \leq X_{n+1} \\
0 & \text{elsewhere}
\end{cases}
\]  

(23)

In view of equation (21), equation (16) takes the form

\[
\sum_n V_n \int_{-w/2}^{w/2} \int_{-w/2}^{w/2} K_{xx}(\vec{r}/\vec{r}') f_n(x') g(y') = -H_x^i + \Delta H_x^i
\]  

(24)

In order to minimize the error \( \Delta H_x^i \), we impose the following condition
\begin{equation}
\langle \Delta H_x^i, \phi_m \rangle = 0 \quad m = 1, 2, \ldots, N
\end{equation}

or

\begin{equation}
\int_{x_{n-1}}^{x_{n+1}} \phi_m(x) \Delta H_x^i \, dx = 0 \quad \text{at } y = 0
\end{equation}

The testing functions \( \phi_m(x) \), in equation (26), have been chosen equal to the expansion functions (Galerkin's method). The above condition imposed on equation (24) reduce the integral equation into a set of linear equations which can determine the coefficients \( V_n \) and the magnetic current \( \vec{M} \) according to equation (21). Once \( \vec{M} \) is known, the fields and field-related parameters may be computed by standard methods.

The above solution can be put into a matrix notation as follows:

\begin{equation}
\begin{bmatrix} V_n \end{bmatrix} = \left[ Y \right]^{-1} \begin{bmatrix} I_n \end{bmatrix}
\end{equation}

where \( \begin{bmatrix} I_n \end{bmatrix} \) is the excitation vector given by

\begin{equation}
I_n = \langle -\Delta H_x^i, f_m \rangle
\end{equation}

The admittance matrix \( [Y] \) can be split into two parts:

\begin{equation}
\begin{bmatrix} Y_{mn} \end{bmatrix}_b = \text{aperture admittance matrix for equivalent problem } b
\end{equation}

and

\begin{equation}
\begin{bmatrix} Y_{mn} \end{bmatrix}_c = \text{aperture admittance matrix for equivalent problem } c.
\end{equation}
In this manner, equation (26) takes the form:

\[
[V_n] = \left( \left[ Y_{mn} \right]_b + \left[ Y_{mn} \right]_c \right)^{-1} [I_n]
\]  

(28)

with

\[
Y_{mn}^b = \iint_{\text{slot}} \langle f_m^b K_{xx}^b \rangle f_n g \, dx' dy'
\]  

(29)

\[
Y_{mn}^c = \iint_{\text{slot}} \langle f_m^c K_{mm}^c \rangle f_n g \, dx' dy'
\]  

(30)

and

\[
K_{xx}^b = \left( 1 + \frac{1}{k_d^2} \frac{d^2}{dx^2} \right) F_{xx} + \frac{1}{k_d^2} \frac{d^2}{dx \partial z} F_{zx}
\]  

(31)

\[
K_{xx}^c = \left( 1 + \frac{1}{k_c^2} \frac{d^2}{dx^2} \right) F_{xx} + \frac{1}{k_c^2} \frac{d^2}{dx \partial z} F_{zx}
\]  

(32)

It is important to note that computation of \([Y]_b\) involves only regions a, b and computation of \([Y]_c\) involves only region c.

Therefore we have divided the problem into two parts each of which can be formulated independently. The elements of the two admittance matrices are given by:

\[
y_{mn}^b = \frac{2}{\pi w} \cdot \frac{1}{\sin^2 \left( k_d l_x \right)} \sqrt{\frac{1}{w/2}} \cdot \sqrt{1 - \left( \frac{2y'}{w} \right)^2} \cdot \int_0^{l_x} dx \int_0^{l_x} dx' \sin[k_d (l_x-x')] \sin[k_d (l_x-x')] \cdot D(\mathbf{r}-\mathbf{r}')
\]
\begin{equation}
\begin{aligned}
+ k_d \int_0^l dx \int_0^l dx' \sin[k_d(l-x)S(|r-r'|)] \\
\quad \left\{ \begin{array}{c}
z=z'=-h \\
y=0
\end{array} \right. 
\end{aligned}
\end{equation}

and

\begin{equation}
Y_{mn}^c = j \frac{64}{\eta_c \omega} k_c^2 \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \epsilon_n \epsilon_m J_0 \left( \frac{mn\omega}{2b} \right) \cos \left( \frac{mnY_c}{b} \right) \cdot \quote
\frac{k_c^2 - \left( \frac{mn}{a} \right)^2}{k_c^2 - \left( \frac{mn}{a} \right)^2} \cos \left( \frac{k_c z}{a} \right) \cdot \quote
\sin^2 \left[ \frac{1}{2} \left( k_d + \frac{mn}{a} \right) \right] \sin^2 \left[ \frac{1}{2} \left( k_d - \frac{mn}{a} \right) \right] \cdot \quote
\sin \left[ \frac{mn}{a} (x_n + X_o) \right] \cdot \sin \left[ \frac{mn}{a} (x_n + X_o) \right] \cdot \quote
\end{equation}

In equation (33), the functions $D(|r-r'|)$ and $S(|r-r'|)$ have the following expressions

\begin{equation}
D(|r-r'|) = - \frac{j\omega e_d}{2\pi k_d} \int_0^\infty \frac{ucosh(uh) + \varepsilon_{rd_o} u sinh(uh)}{\varepsilon_{rd_o} u cosh(uh) + usinh(uh)} \lambda \sum_{i=1}^4 J_0(\lambda \rho_i) d\lambda
\end{equation}

\begin{equation}
S(|r-r'|) = - \frac{j\omega e_d}{2\pi k_d} \int_0^\infty \frac{\lambda}{\varepsilon_{rd_o} u cosh(uh) + usinh(uh)}
\end{equation}
\[
\left\{ \cosh(uh) + \varepsilon_{rd} \frac{u \sinh(uh)}{u} - \frac{u(\varepsilon_{rd} - 1)}{u \cosh(uh) + u \sinh(uh)} \right\} \\
\sum_{i=1}^{2} \int_0 J_{\rho_i} \left[ \delta(x' + l_{x'}) + \delta(x' - l_{x'}) - 2 \cos (k_{d} l_{x'}) \delta(x') \right] \\
\rho_1 = \left\{ \left[ (x + x') + (x_m - x_n) \right]^2 + y^2 \right\}^{1/2} \\
\rho_2 = \left\{ \left[ -(x + x') + (x_m - x_n) \right]^2 + y^2 \right\}^{1/2} \\
\rho_3 = \left\{ \left[ (x - x') + (x_m - x_n) \right]^2 + y^2 \right\}^{1/2} \\
\rho_4 = \left\{ \left[ -(x - x') + (x_m - x_n) \right]^2 + y^2 \right\}^{1/2}
\]
3. **Excitation of the Slot**

As it was mentioned in section 1, the slot is excited by an incident plane wave. This wave can be either parallel or vertically polarized with respect to the plane of incidence. The x-component of the incident magnetic field for both polarizations will be in the form:

\[
H_x = H_0 \, e^{-jk_d x \sin \theta_t} \tag{38}
\]

with

\[
H_0 = \frac{1}{\eta_d} \frac{2E_o^i \cos(k_d (z+h) \cos \theta_t) \sin \phi_i}{\eta_o \cos(k_d h \cos \theta_t) + j \cos \theta_t \sin(k_d h \cos \theta_t)} \tag{39}
\]

\[
H_c^i = \frac{1}{\eta_d} \frac{2 \, E_c^i \cos[k_d (z+h) \cos \theta_t] \cos \phi_i}{\eta_o \cos k_d h \cos \theta_t + j \sin(k_d h \cos \theta_t)} \tag{40}
\]

In view of equations (38)-(40), the elements of the excitation vector \([I]\) are given by the following expression

\[
I_m = 2 \, H_0 \left[ \frac{\cos(k_d^1 x \sin \theta_t) - \cos(k_d^1 x)}{k_d (1 - \sin^2 \theta_t)} \right]^{-jk \sin \theta_t x} \tag{41}
\]
4. Electric Field Distribution on the Slot—Numerical Results

Based on the formulation shown in previous sections, the electric field distribution on cavity-backed slot is shown in figures 1-5 for various slot, substrate and cavity characteristics.
### Figures 1-3

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Cavity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_r = 2.35$</td>
<td>$x_c = 0.4\lambda_o$</td>
</tr>
<tr>
<td>$h = 0.1\lambda_o$</td>
<td>$y_c = 0.25\lambda_o$</td>
</tr>
<tr>
<td>$z_c = 0.124\lambda_o$</td>
<td>$\varepsilon_{zc} = 4.0$</td>
</tr>
<tr>
<td>$\mu_{zc} = 1.0$</td>
<td>$\delta \varepsilon = 0.0$</td>
</tr>
<tr>
<td>$\delta \mu = 0.0$</td>
<td></td>
</tr>
</tbody>
</table>
Electric Field Distribution on a
Cavity-Backed Slot

\((\phi = 0^\circ, \theta = 0^\circ)\)
Electric Field Distribution on a
Cavity-Backed Slot

$\left(\phi = 0^\circ, \theta = 0^\circ\right)$

![Graph showing the electric field distribution on a cavity-backed slot.](image-url)
Electric Field Distribution on a Cavity-Backed Slot

\((\phi = 90^\circ, \theta = 60^\circ)\)

**Figure 3**

![Graph showing the electric field distribution on a cavity-backed slot. The graph plots Amplitude \(E_{slot}/E_i\) and Phase \(E_{slot}/E_i\) against Length/Wavelength. The Amplitude and Phase are shown as separate lines, with Amplitude having a peak at a certain point and Phase showing a consistent trend.](image-url)
Figures 4, 5

**Substrate**

\[ \varepsilon_r = 2.35 \]

\[ h = 0.1\lambda_0 \]

**Cavity**

\[ x_c = 0.4\lambda_0 \]

\[ y_c = 0.25\lambda_0 \]

\[ \varepsilon_{rc} = 4.0 \]

\[ \mu_{rc} = 1.0 \]

\[ \delta_\varepsilon = 0.0 \]

\[ \delta_\mu = 0.0 \]

**Incident Wave**

\[ \phi = 12^\circ \]

\[ \theta = 0^\circ \]
Amplitudes for various Zc's

Figure 4
Figure 5

Phases for various Zc's

Length/Wavelength

Phase E\_Signal
Figures 6-9

Substrate

$\varepsilon_r = 2.35$

$h = 0.1\lambda_0$

Cavity

$x_C = 0.4\lambda_0$

$y_C = 0.25\lambda_0$

$z_C = 0.125\lambda_0$

$\mu_{rc} = 1.0\lambda_0$

$\delta_\varepsilon = 0.0$

$\delta_\mu = 0.0$

Incident Wave

$\phi = 12^\circ$

$\theta = 0^\circ$
Phases for various ErC's

Figure 7

[Graph showing phases for various ErC's with markers representing different ErC values: Phase(2.0), Phase(4.57), Phase(7.143), Phase(9.714).]
Figure 9

Phases for Various ERs

Length/Wavelength

Phase E1 / E2

Phases (2.00)
Phases (1.74)
Phases (1.48)
References

Appendix A

Green's Function for a Magnetic Current on the Wall of a Rectangular Cavity

Let us begin by considering an x-directed Hertzian magnetic dipole (HMD) at the position \((x', y', z')\) as shown in figure (A.1).

With an assumed \(e^{j\omega t}\) time dependency, Maxwell's equations take the form,

\[
\begin{align*}
\nabla \times \mathbf{H} &= j\omega \mathbf{D} \\
\nabla \times \mathbf{E} &= -j\omega \mathbf{B} - \mathbf{M} \\
\n\nabla \cdot \mathbf{D} &= 0 \\
\n\nabla \cdot \mathbf{B} &= 0
\end{align*}
\tag{1}
\]

and

\[
\begin{align*}
\mathbf{D} &= \varepsilon_c \mathbf{E} \\
\mathbf{B} &= \mu_c \mathbf{H}
\end{align*}
\tag{2}
\]

with

\[
\begin{align*}
\varepsilon_c &= \varepsilon_c'(1 - j \tan \delta) \\
\mu_c &= \mu_c'(1 - j \tan \delta)
\end{align*}
\tag{3}
\tag{4}
\]

We can now introduce an electric vector potential function \(\mathbf{F}\) such that

\[
\mathbf{D} = \nabla \times \mathbf{F}.
\tag{5}
\]

When we substitute equation (5) into (1a), the following relation is derived

\[
\nabla \times [\mathbf{H} - j\omega \mathbf{F}] = 0.
\tag{6}
\]
Equation (6) is equivalent to

$$\vec{H} = j\omega \vec{E} + \nabla \phi_m.$$ (7)

From equations (7) and (1b), we can get the following result

$$\nabla \times \nabla \times \vec{F} = \nabla (\nabla \cdot \vec{F}) - \nabla^2 \vec{F} = k_c^2 \vec{F} - j\omega \varepsilon_c\varepsilon \nabla \phi_m - \varepsilon M.$$ (8)

Since equation (5) alone cannot specify \( \vec{F} \), we can select its divergence so that

$$\nabla \cdot \vec{F} = -j\omega \varepsilon \mu \phi_m.$$ (9)

Equations (8) and (9) then give

$$\left( \nabla^2 + k_c^2 \right) \vec{F} = \varepsilon_c M.$$ (10)

The electric and magnetic fields, in terms of \( \vec{F} \), are given as follows

$$\vec{E} = \frac{1}{\varepsilon_c} \nabla \times \vec{F}$$ (11)

$$\vec{H} = \frac{j}{\omega \varepsilon \mu_c} \left[ k_c^2 \vec{F} + \nabla \nabla \cdot \vec{F} \right]$$ (12)

with

$$k_c^2 = \omega^2 \varepsilon_c \mu_c.$$

For a magnetic current \( \vec{M} \) with more than one component, the electric potential can be considered as a dyadic function of the form

$$\vec{F} = \hat{x}x F_{xx} + \hat{y}y F_{yy} + \hat{z}z F_{zz}.$$ (11)

For the case of Hertzian dipoles \( \vec{F} \) is being considered as the Green's function of the problem and satisfies the following differential


\[ \left( \nabla^2 + k_c^2 \right) G = \varepsilon_c \Xi \quad (12) \]

where \( \Xi \) is the unit dyadic. For the problem of figure A.1, \( G_{yy} = G_{zt} = 0 \)

and

\[
E_x = 0 \\
E_y = \frac{1}{\varepsilon_c} \frac{\partial G_{xx}}{\partial z} \\
E_z = -\frac{1}{\varepsilon_c} \frac{\partial G_{xx}}{\partial y}
\quad (13)
\]

Also, the magnetic field is given by

\[
H_x = \frac{j}{\omega \varepsilon \mu c_c} \left( k_c^2 + \frac{\partial^2}{\partial x^2} \right) G_{xx} \\
H_y = \frac{j}{\omega \varepsilon \mu c_c} \frac{\partial^2}{\partial y \partial x} G_{xx} \\
H_z = \frac{j}{\omega \varepsilon \mu c_c} \frac{\partial^2}{\partial z \partial x} G_{xx}
\quad (14)
\]

The boundary conditions for this problem are:

\[
E_y^I = 0 \quad z = 0 \quad x = 0, a \\
E_z^I = 0 \quad z = 0, a \quad y = 0, b \\
E_y^{II} = 0 \quad z = c \quad x = 0, a \\
E_z^{II} = 0 \quad x = 0, a \quad y = 0, b
\quad (15)
\]
The solution to equation (12) which satisfies the above boundary conditions is in the form:

**region I**

\[
G_{xx}^I = \frac{4\varepsilon_c}{ab} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \varepsilon_n \varepsilon_m \sin\left(\frac{m\pi x'}{a}\right) \cos\left(\frac{m\pi y'}{b}\right) \cdot \sin\left(\frac{n\pi x}{a}\right) \cos\left(\frac{m\pi y}{b}\right) \cdot \frac{\cos\left[k_z(c-z')\right]}{k_z} \cdot \frac{\cos(k_z z)}{\sin(k_z c)} \tag{16}
\]

**region II**

\[
G_{xx}^{II} = \frac{4\varepsilon_c}{ab} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \varepsilon_n \varepsilon_m \sin\left(\frac{m\pi x'}{a}\right) \cos\left(\frac{m\pi y'}{b}\right) \cdot \sin\left(\frac{n\pi x}{a}\right) \cos\left(\frac{m\pi y}{b}\right) \cdot \frac{\cos(k_z' z')}{k_z'} \cdot \frac{\cos\left[k_z(c-z)\right]}{k_z} \cdot \frac{\cos(k_z c)}{\sin(k_z c)} \tag{17}
\]

with

\[
k_z^2 = k_c^2 - \left(\frac{n\pi}{a}\right)^2 - \left(\frac{m\pi}{b}\right)^2 \tag{18}
\]

Using the above expressions and for the case of a magnetic dipole \(\mathbf{M}_x\) at \(z' = 0\), the x-component of the magnetic field in the cavity
takes the form

\[ H_x^c = \iint_S \left[ 1 + \frac{1}{k_c^2} \frac{\partial^2}{\partial x^2} \right] G^{II}_{x,x} (x, y, z; x', y') \ M_x(x', y') \ dx' \ dy' . \] (19)
This program evaluates the elements of the admittance matrix in the following problem:

"Scattering by two slots on infinite dielectric substrate with cavities at the back."

This program is good for any substrate thickness $h$, $\varepsilon_r$ and any dimensions of the cavities.

SUBROUTINE SLOT
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 CONST, GSK, GS1S2K
COMPLEX ZS, ZS1S2, CI, BMATR, CUR

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)

COMMON/COMP/ZS(50), ZS1S2(350), NS, NS1S2

COMMON/OUT/GS(50), GS1S2(350)

COMMON/MAT/PLI, AI, TI, V(3), IY

COMMON/FUT/SSJ0(250,7), SAJ0(250,7), YSIN, YCOS

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)

COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TP12, PI, W, E1,
* E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, PI,
* NS1, NS2, NS3, NOFF

COMMON/DATT/COAL(20), POINT(20), CN(51), BM(51), POLTM(20),
* POLTE(20), AM(41), DM(41), POLES(40), VXKM(20), VZKM(20), VZXE(20),
* BPOINT(10), BCOAL(10), MPOINT, NPOINT, NK0, MA, NMT, NTE, NK0K, IFIRST

COMMON/COEF/RX, XX, RZ, XZ, FRX, FRZ, FLX, FLZ

COMMON/IOFF/INS, INS1S2

COMMON/B01/BJ0, BJ1

CALL DATA_SLOT

Subroutine POLES evaluates the poles of the Green's function and orders them according to their magnitude

IFIRST=0 : dominant mode is a TM wave
1 : dominant mode is a TE wave
2 : only one TM wave

CALL POLES
PRINT *, 'POLES has run'
H=H*DSQRT(ER)
CI=(0.00,1.00)

NS=NS1
IF (NS1.LT.NS2) NS=NS2
MS=NS
IF (NOFF.EQ.1) GO TO 50
NS1S2=NS2+NSS2-1
MS1S2=NS1S2
IF (NS1S2.GT.200) NS1S2=200

C
50 IF (NMAX(INS).LE.(NS+2)) NMAX(INS)=NS+2
IF (NOFF.EQ.1) GO TO 51
IF (NMAX(INS1S2).LE.(NS1S2+2)) NMAX(INS1S2)=NS1S2+2

C
51 ADL=AKK*DLX
YSIN=D SIN(ADL)
YCOS=DCOS(ADL)

For the normalization of the current along the y axis
CVON=W*PI/2.D0

Computation of lambda-integration limits between 0 and A

CALL LIMIT

Evaluation of the Green's function at different points in the interval [0,A]. The Bessel function has been excluded

CALL GREEN

Evaluation of the tail contribution (from a to infinity)

CALL TAIL

CONST=(1.D0/CVON)*DSQRT(ER)/(480.D0*(PI**3)*YSIN*YSIN)
WRITE(6,9)
9 FORMAT('--------------------- ZS ---------------------')
WRITE(6,10) MS
10 FORMAT(1X,I4)
DO 11 K=1,MS
   ZS(K)=ZS(K)*CONST
   GSK=REAL(GS(K))*CONST
   WRITE (6,30) K,ZS(K),GSK
30 FORMAT(1X,I4,2X,'ZS=',E14.7,2X,E14.7,2X,
   'GSK=',E14.7)
   ZS(K)=(ZS(K)+GSK)*CI
   WRITE (6,12) ZS(K)
12 FORMAT(68X,E14.7,1X,E14.7)
11 CONTINUE
IF (NOFF.EQ.1) GO TO 52
WRITE (6,15)
15 FORMAT('---------------------- ZS1S2 ----------------------')
WRITE (6,10) MS1S2
DO 16 K=1,MS1S2
   ZS1S2(K)=ZS1S2(K)*CONST
   GS1S2K=REAL(GS1S2(K))*CONST
   WRITE (6,32) K,ZS1S2(K),GS1S2K
32 FORMAT(1X,I4,2X,'ZS1S2=',E14.7,2X,E14.7,2X,
   'GS1S2K=',E14.7)
   WRITE (6,12) ZS1S2(K)
   ZS1S2(K)=(ZS1S2(K)+GS1S2K)*CI
16 CONTINUE
52 CONTINUE
1000 CONTINUE
RETURN
END
This subroutine evaluates the limits of integration in
the interval [0, \( k_0 \)].
Specifically:
1) It divides the interval [0, \( k_0 \)] to 10 equal
subsections and then apply fixed-point Gaussian
Quadrature
2) It divides the interval [\( k_0, k \)] into so many
subsections as the number of poles and in
such a way that each subsection includes one
pole only away from the ends of the subsection
3) It divides the interval [\( k, A \)] into 20 equal
subsections and then apply fixed-point Gaussian
Quadrature

SUBROUTINE LIMIT
IMPLICIT REAL*8 (A-H, O-Z)
EXTERNAL WSPE, WTPE, WSPM

COMMON/DAT7/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI, W, E1,
* E2, EER, AK0, AK, AKK, F1, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, FI,
* NS1, NS2, NSS2, NOFF

COMMON/DATT/COAL(20), POINT(20), CN(51), BM(51), POLT(20),
* POLTE(20), AM(41), DM(41), POLES(40), VXOM(20), VXOM(20), VXOE(20),
* BPOINT(10), BCOAL(10), MPOINT, NPOINT, NK0, MA, NTM, NTE, NK0K, IFIRST

Step 1 : Evaluation of vector CN
it gives the end points of the
intervals considered in (0, \( k_0 \))

\[ \text{DELTA}=\text{AK0}/\text{FLOAT}(	ext{NK0}) \]
CN(1)=0.D0
DO 1 I=1, NK0
CN(I+1)=DELTA*FLOAT(I)
1 CONTINUE

Step 2 : Evaluation of vector BM
it gives the end points of the
intervals considered in (\( k, A \))

\[ \text{DELTA}=(\text{A/DSQRT}(\text{EER})-\text{AK})/\text{FLOAT}(\text{MA}) \]
BM(1)=AK
DO 2 I=1, MA
BM(I+1)=DELTA*FLOAT(I)+AK
2 CONTINUE

Step 3 : Evaluation of the vectors AM,DM
"AM" gives the end points around
the TM poles
"DM" gives the end points around
the TE poles

IFIRST= 2 only one TM pole
1 TE0<TM0
0 TM0<TE0

AM(I)=AK0
DM(I)=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
--- Dielectric constant ---
2.35

--- Substrate Thickness ---
0.1

--- Conductor Thickness ---
0.0001

--- Dimensions of the Cavity ---
0.4
0.25
0.124

--- Dielectric Permittivity and Permeability of the Medium in the Cavity ---
4.00
1.00
0.00

--- Slot width ---
0.1

--- Position of the Center of the Leading Edge of the Slot ---
0.072796
0.075

--- Subsection Length ---
0.022

--- Lower Limit of the Tail Contribution ---
100.0

--- Number of Points on Each Slot ---
23
0

--- Number of Offsets ---
1

--- Offset Between the Two Slots ---
0.0

--- Longitudinal Displacement Between the Two Slots ---
0

--- Theta angle ---
0.0

--- Phi angle ---
This program evaluates the reflection coefficient of a cavity-backed slot as a function of the characteristics of the cavity and the incident wave.

```
IMPLICIT REAL*8 (A-H,O-Z)

C          MAIN SLOT.FTN
C          -----------------------------------------------
C          This program evaluates the reflection coefficient of
C          cavity-backed slot as a function of the characteristics
C          of the cavity and the incident wave
C          -----------------------------------------------

C          INCLUDE '/SYS/INS/CAL.INS.FTN'
C          INTEGER*4 CPU_SECONDS
C          INTEGER*2 TIMEDATE_REC(6)
C          COMPLEX R0,REFL,R_RATIO,CUR,BMATR,ZS,ZS1S2
C          CHARACTER*10 PAR1,PAR2
C          LOGICAL THFLAG1,FIFLAG1,XCFLAG1,YCFLAG1,ZCFLAG1,ERCFLAG1
C          LOGICAL THFLAG2,FIFLAG2,XCFLAG2,YCFLAG2,ZCFLAG2,ERCFLAG2
C          COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
C          *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C          COMMON/COMP/ZS(50),ZS1S2(350),NS,NS1S2
C          COMMON/OUT/GS(50),GS1S2(350)
C          COMMON/MAT/PLI,AL,TI,V(3),IV
C          COMMON/PUT/SSJO(250,7),SAJ0(250,7),Y SIN,YCOS
C          COMMON/ADON/DIST(250,7,15),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
C          *DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C          COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,XO,YO,A,TPI,TPIZ,PI,W,E1,
C          *E2,ER,AKO,AK,AKK,FA,OFFSET(7),WDELTA,OFFLIM,ERROR,THI,FI,
C          *NS1,NS2,NS3,NOFF
C          COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLT(20),
C          *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
C          *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C          COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,FLX,FLZ
C          COMMON/IOFF/INS,INS1S2
C          COMMON/MAN/BMATR(260,260),IA(260),IB(260)
C          COMMON/PAT/CUR(260)
C          COMMON/B01/BJ0,BJ1
C          COMMON/LOSS/TLOS_E,TLOS_M
C          OPEN(UNIT=01,FILE='OUT.PLOT',STATUS='OLD')
C          OPEN(UNIT=05,FILE='DATA.SLOT',STATUS='OLD')
C          OPEN(UNIT=06,FILE='OUT.SLOT',STATUS='OLD')
C          CALL PROC1_SGET_CPUT(CLOCK)
C          CPU_SECONDS=CALL_CWLOCK_TO_SEC(CLOCK)
C          THFLAG1=.FALSE.
C          FIFLAG1=.FALSE.
C          XCFLAG1=.FALSE.
C          YCFLAG1=.FALSE.
C          ZCFLAG1=.FALSE.
C          ERCFLAG1=.FALSE.
```
THFLAG2=.FALSE.
FIFLAG2=.FALSE.
XCFLAG2=.FALSE.
YCFLAG2=.FALSE.
ZCFLAG2=.FALSE.
ERCFLAG2=.FALSE.

C
PRINT *, 'You can plot the reflection coeff. as a function of:'
PRINT *, 'THETA,FI,XC,YC,ZC,ERC'
PRINT *, 'The reflection coefficient will vary as a function of:'
PRINT *, 'First parameter:'
READ (*,1) PAR1
PRINT *, 'Second parameter:'
READ (*,1) PAR2

1 FORMAT(A10)

C
PRINT *, 'What is the maximum value of ', PAR1
READ (*,2) PAR1MX

2 FORMAT(E14.7)
PRINT *, 'What is the minimum value of ', PAR1
READ (*,2) PAR1MN
PRINT *, 'How many points should be considered'
PRINT *, '(number of points >=1)'
READ (*,23) NPAR1
PRINT *, 'What is the maximum value of ', PAR2
READ (*,2) PAR2MX
PRINT *, 'What is the minimum value of ', PAR2
READ (*,2) PAR2MN
PRINT *, 'How many points should be considered'
PRINT *, '(number of points >=1)'
READ (*,23) NPAR2

23 FORMAT (I5)

Subroutine DATA reads the given values for different variables

CALL DATA

PRINT *, 'Subroutine DATA run'

Subroutine slot evaluates the parts of the elements of the impedance matrix which are coming from the dielectric substrate

WREAL=W
W=W*(1.0D0+2.0D0*WDELTA/W)

CALL SLOT

PRINT *, 'Subroutine SLOT run'

DELTA1=0.0D0
DELTA2=0.0D0
IF (NPAR1.GT.1) DELTA1=(PAR1MX-PAR1MN)/FLOAT(NPAR1-1)
IF (NPAR2.GT.1) DELTA2=(PAR2MX-PAR2MN)/FLOAT(NPAR2-1)

NTHI=-1
NEI=-1
NXC=-1
NYC=-1
NZC=-1
NERC=-1

IF ((PAR1.EQ.'THETA').OR.(PAR1.EQ.'theta')) THFLAG1=.TRUE.
IF ((PAR2.EQ.'THETA').OR.(PAR2.EQ.'theta')) THFLAG2=.TRUE.
IF ((PAR1.EQ.'FI').OR.(PAR1.EQ.'fi')) FIFLAG1=.TRUE.
IF ((PAR2.EQ.'FI') .OR. (PAR2.EQ.'fi')) FIFLAG2=.TRUE.
IF ((PAR1.EQ.'XC') .OR. (PAR1.EQ.'xc')) XCFLAG1=.TRUE.
IF ((PAR2.EQ.'XC') .OR. (PAR2.EQ.'xc')) XCFLAG2=.TRUE.
IF ((PAR1.EQ.'YC') .OR. (PAR1.EQ.'yc')) YCFLAG1=.TRUE.
IF ((PAR2.EQ.'YC') .OR. (PAR2.EQ.'yc')) YCFLAG2=.TRUE.
IF ((PAR1.EQ.'ZC') .OR. (PAR1.EQ.'zc')) ZCFLAG1=.TRUE.
IF ((PAR2.EQ.'ZC') .OR. (PAR2.EQ.'zc')) ZCFLAG2=.TRUE.
IF ((PAR1.EQ.'ERC') .OR. (PAR1.EQ.'erc')) ERCFLAG1=.TRUE.
IF ((PAR2.EQ.'ERC') .OR. (PAR2.EQ.'erc')) ERCFLAG2=.TRUE.

IF (THFLAG1) THEN
  THIMX=PAR1MX*PI/180.D0
  THIMN=PAR1MN*PI/180.D0
  DELTHI=DELTA1*PI/180.D0
  NTII=NPAR1
  ID_THI=1
END IF

IF (THFLAG2) THEN
  THIMX=PAR2MX*PI/180.D0
  THIMN=PAR2MN*PI/180.D0
  DELTHI=DELTA2*PI/180.D0
  NTII=NPAR2
  ID_ THI=2
END IF

IF (FIFLAG1) THEN
  FIMX=PAR1MX*PI/180.D0
  FIMN=PAR1MN*PI/180.D0
  DELFI=DELTA1*PI/180.D0
  NFI=NPAR1
  ID_FI=1
END IF

IF (FIFLAG2) THEN
  FIMX=PAR2MX*PI/180.D0
  FIMN=PAR2MN*PI/180.D0
  DELFI=DELTA2*PI/180.D0
  NFI=NPAR2
  ID_ FI=2
END IF

IF (XCFLAG1) THEN
  XCMX=PAR1MX
  XCMN=PAR1MN
  DELXC=DELTA1
  NXC=NPAR1
  ID_XC=1
END IF

IF (XCFLAG2) THEN
  XCMX=PAR2MX
  XCMN=PAR2MN
  DELXC=DELTA2
  NXC=NPAR2
  ID_XC=2
END IF

IF (YCFLAG1) THEN
  YCMX=PAR1MX
  YCMN=PAR1MN
  DELYC=DELTA1
  NYC=NPAR1
  ID_YC=1
END IF

IF (YCFLAG2) THEN
  YCMX=PAR2MX
  YCMN=PAR2MN
  DELYC=DELTA2
  NYC=NPAR2
  ID_YC=2
END IF
IF (ZCFLAG1) THEN
  ZCMX=PAR1MX
  ZCMN=PAR1MN
  DELZC=DELTA1
  NZC=NPAR1
  ID_ZC=1
END IF
IF (ZCFLAG2) THEN
  ZCMX=PAR2MX
  ZCMN=PAR2MN
  DELZC=DELTA2
  NZC=NPAR2
  ID_ZC=2
END IF
IF (ERCFLAG1) THEN
  ERCMX=PAR1MX
  ERCMN=PAR1MN
  DELERC=DELTA1
  NERC=NPAR1
  ID_ERC=1
END IF
IF (ERCFLAG2) THEN
  ERCMX=PAR2MX
  ERCMN=PAR2MN
  DELERC=DELTA2
  NERC=NPAR2
  ID_ERC=2
END IF
C
WRITE (1,100) PAR1,PAR2
100 FORMAT(5X,A10,5X,A10,5X,'Rel.Refl.Coefficient-Real',5X,**Rel.Refl.Coefficient-Imaginary'/)
C
SLENG=(NS1+1)*DLX
IF (NXC.EQ.-1) THEN
  XCMN=XN
  NZC=1
END IF
DO 3 IXC=1,NXC
  XC=XCMN+FLOAT(IXC-1)*DELC
  X0=(XC-SLENG)/2.D0
  IF (ID_XC.EQ.1) WPAR1=XC
  IF (ID_XC.EQ.2) WPAR2=XC
  IF (NYC.EQ.-1) THEN
    YCMN=YC
    NYC=1
  END IF
3 CONTINUE
DO 4 IYC=1,NYC
  YC=YCMN+FLOAT(IYC-1)*DELYC
  Y0=YC/2.D0
  IF (ID_YC.EQ.1) WPAR1=YC
  IF (ID_YC.EQ.2) WPAR2=YC
  IF (NZC.EQ.-1) THEN
    ZCMN=ZC
    NZC=1
  END IF
4 CONTINUE
DO 5 IZC=1,NZC
  ZC=ZCMN+FLOAT(IZC-1)*DELZC
  IF (ID_ZC.EQ.1) WPAR1=ZC
  IF (ID_ZC.EQ.2) WPAR2=ZC
  IF (NERC.EQ.-1) THEN
    ERCMN=ERC
    NERC=1
  END IF
5 CONTINUE
DO 6 IERC=1,NERC
  ERC=ERCMN+FLOAT(IERC-1)*DELERC
6 CONTINUE
IF(ID_ERC.EQ.1) WPAR1=ERC
IF(ID_ERC.EQ.2) WPAR2=ERC
CALL CAV_INV(NOR)

PRINT*, 'Subroutine CAV_INV run'

IF (NTHI.EQ.-1) THEN
 THIMN=THI
 NTHI=1
END IF
DO 7 ITHI=1,NTHI
 THI=THIMN+FLOAT(ITHI-1)*DELTHI
IF(ID_THI.EQ.1) WPAR1=THI*180.D0/PI
IF(ID_THI.EQ.2) WPAR2=THI*180.D0/PI
IF (NFI.EQ.-1) THEN
 FINM=FI
 NFI=1
END IF
DO 8 IFI=1,NFI
 FI=FINM+FLOAT(IFI-1)*DELFI
IF(ID_FI.EQ.1) WPAR1=FI*180.D0/PI
IF(ID_FI.EQ.2) WPAR2=FI*180.D0/PI
CALL MULT_EXC(NOR,R0)

PRINT *, 'Subroutine MULT_EXC run'
CALL REFLC(WREAL,R0,REFL)
PRINT *, 'Subroutine REFLC run'
R_RATIO=REFL/R0

WRITE (1,10) WPAR1, WPAR2, R_RATIO

10 *

8 CONTINUE
7 CONTINUE
6 CONTINUE
5 CONTINUE
4 CONTINUE
3 CONTINUE

WRITE (6,9000) CPU_SECONDS
9000 FORMAT(1X,'CPU TIME=',I10,'SECS')
STOP
END

The name of this subroutine is DATA
and gives all the data used by the main program and the other
subroutines.

SUBROUTINE DATA
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI, W, E1,
* E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, FI,
* NS1, NS2, NSS2, NOFF
COMMON/IOFF/INS, INS1S2
COMMON/LOSS/TLOS_E, TLOS_M
PI=3.1415926535899D0

---- Dielectric constant ----
READ (5,1) ER
1 FORMAT (16X,D16.9)
   EER=ER
   AER=DRT(ER)
   WRITE (6,2) ER
2 FORMAT(10X,'Dielectric Constant of the Substrate'/10X,E14.7/)

    ---- Substrate Thickness ----

READ (5,1) H
WRITE (6,3) H
3 FORMAT(10X,'Substrate Thickness'/10X,E14.7/)

    ---- Conductor Thickness ----

READ (5,1) T
WRITE (6,4) T
4 FORMAT(10X,'Conductor Thickness'/10X,E14.7/)
   T=T*AER

    ---- Dimensions of the Cavity ----

READ (5,1) XC
READ (5,10) YC
READ (5,10) ZC
10 FORMAT(6X,D14.7)
   WRITE (6,5) XC,YC,ZC

5 FORMAT(10X,'Dimensions of the Cavity'/10X,'XC=',E14.7/)
   *10X,'YC=',E14.7/10X,'ZC=',E14.7/)
      XC=XC*AER
      YC=YC*AER
      ZC=ZC*AER

      ---- Relative Permittivity and Permeability of the Cavity ----

READ (5,1) ERC
READ (5,10) RMC
READ (5,10) TLOS_E
READ (5,10) TLOS_M
   WRITE (6,6) ERC,RMC,TLOS_E,TLOS_M

6 FORMAT(10X,'Dielectric of the Cavity'/10X,'ERC=',E14.7,2X,'
   'RMC=',E14.7,2X,'
      TLOS_E=',E14.7,2X,'
      TLOS_M=',E14.7)

    ---- Slot width ----

READ (5,1) W
WRITE (6,7) W
7 FORMAT(10X,'Slot Width'/10X,E14.7/)
   W=W*AER

      ---- Position of the Center of the Slot ----

READ (5,1) X0
READ (5,10) Y0
WRITE (6,8) X0,Y0
8 FORMAT(10X,'Position of the Center of the Slot'/10X,'
   '*E14.7/10X,'
      'Y0=',E14.7/)
      X0=X0*AER
      Y0=Y0*AER

      ---- Subsection Length ----

READ (5,1) DLX
WRITE (6,9) DLX
9 FORMAT(10X,'Subsection Length'/10X,E14.7/)
---- Lower Limit of the Tail Contribution ----

READ (5,1) A
WRITE (6,11) A

11 FORMAT(1X,'Lower Limit of Tail Contribution'/10X,E14.7/)

---- Number of Points on Each Slot ----

READ (5,20) NS1
READ (5,21) NS2

20 FORMAT(//'6X,I4)
21 FORMAT(6X,I4)
WRITE (6,12) NS1,NS2

12 FORMAT(10X,'Number of Points on Each Slot'/10X,'NS1=',I4/*10X,'NS2=',I4)

---- Number of Offsets ----

READ (5,20) NOFF
WRITE (6,15) NOFF

15 FORMAT(10X,'Number of Offsets'/10X,I4/)

---- Offset Between the Two Slots ----

READ (5,1) OFFSET(1)
WRITE (6,13) OFFSET(1)

13 FORMAT(10X,'Offset Between the Two Slots'/10X,E14.7/)
OFFSET(1)=OFFSET(1)*AER

DO 16 I=2,NOFF
   READ (5,10) OFFSET(I)
   WRITE (6,17) OFFSET(I)
17 FORMAT(10X,E14.7/)
   OFFSET(I)=OFFSET(I)*AER
16 CONTINUE

---- Order of Offsets ----

INS=1
INS1S2=2

---- Longitudinal Displacement Between the Two Slots ----

READ (5,20) NSS2
WRITE (6,14) NSS2

14 FORMAT(10X,'Longitudinal Displacement Between the Two Slots'/10X,I4/)

---- Theta angle ----

READ (5,1) THI
WRITE (6,23) THI

23 FORMAT(10X,'Theta Angle'/10X,E14.7/)

---- Fi Angle ----

READ (5,1) FI
WRITE (6,22) FI

22 FORMAT(10X,'Fi Angle'/10X,E14.7/)
RETURN
END
COMMON/IOFF/INS,INS1S2

C
CI=(0.0,1.0)
NCON=0
X=AUP-ALOW
Y=AUP+ALOW
AK0Z=AK0*AK0
AK2=AK*AK
ER1=1.0D0-ER
NPOL=N
IF (IFIRST.EQ.2) NPOL=1
IF (IAD.GT.2) GO TO 1
   ALI=0.5D0*(TI*X+Y)
   GCONX=AI*X*0.5D0
   FCONX=GCONX
   GCONZ=GCONX*ER1
   FCONZ=FCONX
   AIMA=1.0D0
   CALL GREI(ALI,0.0D0,0.0D0,IAD,0.0D0)
   GO TO 10
1 IF (IAD.NE.3) GO TO 2
   ALI=0.5D0*(TI*X+Y)
   XTM=POLTM(NPOL)
   TMTM=(2.0D0*XTM-Y)/X
   GCONX=AI/(TI-TTM)
   GCONZ=GCONX*ER1
   FCONX=GCONX
   FCONZ=FCONX
   AIMA=0.0D0
   CALL GREI(ALI,XTM,0.0D0,IAD,0.0D0)
   GO TO 10
2 IF (IAD.NE.4) GO TO 3
   ALI=POLTM(N)
   TM=(2.0D0*ALI-Y)/X
   GCONX=AI/(TI-TM)
   GCONZ=GCONX*ER1
   FCONX=0.0D0
   FCONZ=0.0D0
   AIMA=0.0D0
   RX=VXXM(N)
   RZ=VXXM(N)
   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=(2.0D0*XTE-Y)/X
   GCONX=AI*X*0.5D0
   GCONZ=AI*ER1/(TI-TMTE)
   FCONX=GCONX
   FCONZ=FCONX
   AIMA=0.0D0
   CALL GREI(ALI,0.0D0,XTE,IAD,TMTE)
   GO TO 10
4 IF (IAD.NE.6) GO TO 5
   NCON=6
   ALI=POLTE(N)
   TM=(2.0D0*ALI-Y)/X
   GCONX=0.0D0
   GCONZ=AI*ER1/(TI-TM)
   FCONX=0.0D0
   FCONZ=0.0D0
   AIMA=0.0D0
   RZ=VXXE(N)
   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
FCONX=GCONX
FCONZ=FCONX
AIMA=0.0
CALL GREI(ALI,0.0,0.0,0.0)
GO TO 10

6 NCON=8
ALI=POLT(N)
TM=(2.0*ALI-Y)/X
FCONX=0.0
FCONZ=0.0
AIMA=0.0
RX=VX(N)
RZ=VZ(N)
GO TO 28

10 CONTINUE
GXXR=GCONX*RX-FCONX*FRX
GXXA=AIMA*GCONX*XX
GXXR=GCONZ*RZ-FCONZ*FRZ
GXXA=AIMA*GCONZ*XZ

27 CONTINUE
VARX=AK2*GXXR
VARZ=AK2*(GXXR-GXXR)
GXXR=VARX
GXXR=VARZ
VARX=AK2*GXXX
VARZ=AK2*(GXXX-GXXX)
GXXX=VARX
GXXX=VARZ
PLI=ALI

CALL ADONI
DO 13 K=1,NS
S1=REAL(GXXR*SSJ0(K,INS)+GXXR*SAJ0(K,INS))
S2=REAL(GXXX*SSJ0(K,INS)+GXXX*SAJ0(K,INS))
ZS(K)=ZS(K)+S1-CI*S2

13 CONTINUE
DO 20 K=1,NS1S2
S1=REAL(GXXR*SSJ0(K,INS1S2)+GXXR*SAJ0(K,INS1S2))
S2=REAL(GXXX*SSJ0(K,INS1S2)+GXXX*SAJ0(K,INS1S2))
ZS1S2(K)=ZS1S2(K)+S1-CI*S2

20 CONTINUE
28 IF (NCON.EQ.0) GO TO 24
GCONX1=0.0
GCONX2=0.0
GCONZ1=ER1*DLOG((1.0-TM)/(1.0+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
GXXR=GCONZ1*RX
GXXX=GCONZ2*RX

25 CONTINUE
NCON=0
GO TO 27

24 CONTINUE
RETURN
END

This subroutine evaluates the integrand of the green's
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
   CONTINUE
7 CONTINUE

;-----------------------------------------------
; Step 4 : evaluation of vectors VZX,E
;-----------------------------------------------
IF (IFIRST.EQ.2) GO TO 10
DO 11 I = 1, NTE
   ARG = POLTE(I)
   VZX(E)(I) = HZX(E)(ARG)
11 CONTINUE
10 CONTINUE

;-----------------------------------------------
; Step 5 : evaluation of vector VXXM,VZXM
;-----------------------------------------------
DO 12 I = 1, NTM
   ARG = POLTM(I)
   VXXM(I) = GXXM(ARG)
   VZXM(I) = GZXM(ARG)
12 CONTINUE
RETURN
END

This subroutine evaluates the values of the integrand of the
Green’s function at different points in the interval
[0, Α]. Then it evaluates the space integrals of the Bessel
function at the same points and multiply these values with
the corresponding values of the Green’s function.
Finally, it multiplies these products with known coeffic.
and it adds them up. This way, the momenta’–method
space integrals of the first part of the Green’s function are
evaluated and are stored in the complex vectors ZS,ZS1S2

SUBROUTINE GREEN
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX ZS,ZS1S2,CI
COMMON/COMP/ZS(50), ZS1S2(350), NS, NS1S2
COMMON/MAT/PLI, AI, TI, V(3), IY
COMMON/PUT/SSJO(250, 7), SAJ0(250, 7), YSIN, YCOS
COMMON/ADON/DIST(250, 7, 10), ROE(20, 250, 7, 10), AX, SERS(5), SERA(5),
  *DARG(10, 4), S(10, 2), WREAL, NSER, NMAX(7)
COMMON/DAET/ER,H,T,DLX,XY,YC,YZ,ERC,RMC,X0,Y0,A,TPI,TPI2,FI,W,F1,
*E2,ER,AK0,AK,AKK,FA,OFFSET(7),WDelta,OFFLIM,ERROR,THI,FI,
*NS1,NS2,NSS2,NOFF

COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
*POLTE(20),AM(41),DM(41),POLES(40),VXNM(20),VZM(20),VZXE(20),
*BPOINT(10),BCOAL(10),NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST

COMMON/IOFF/INS,INS1S2

Evaluation of the coefficients for the | FF's functions |
+------------------------------------------+-------------------------------------+
F1X=1.D0
F1Z=(1.D0-ER)/((1.D0+ER)*(1.D0+E2)*(1.D0+0.5D0*E1))

CALL ARIS

DO 1 I=1,NPOINT
   IY=I
   AI=COAL(I)
   TI=POINT(I)
   evaluation of intervals 1 and 2
   IAD=1
   DO 2 N=1,NK0
      AUP=CN(N+1)
      ALOW=CN(N)
      CALL FUNCT(IAD,AUP,ALOW,N)
   CONTINUE
   evaluation of intervals 3 and 4
   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)
         IFD=IFD+1
      CONTINUE
   3 CONTINUE
   IF (IFIRST.EQ.2) GO TO 9
   evaluation of the intervals 5 and 6,9,11
   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)
         IFD=IFD+1
      CONTINUE
   5 CONTINUE
   6 CONTINUE
   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)
   CONTINUE
7 CONTINUE

evaluation of the intervals 8,10

   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)
   IFD=IFD+1
8 CONTINUE
RETURN
END

Functions : GXXM,GZXM,H2XE

These functions evaluate the residues from the different poles

FUNCTION GXXM(X)
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,A,TPI,TPI2,PI,W,E1,
*E2,ER,AK0,AK,AKK,FA,OFFSET(7),WDELTA,OFFLIM,ERROR,THI,FI,
*NS1,NS2,NSS2,NOFF

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+RM0)*(RM/RM0)*DCOS(RMH)+(1.0+ER*RM0)*DSIN(RMH)
GXXM=SXN/SXD
RETURN
END

FUNCTION GZXM(X)
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,A,TPI,TPI2,PI,W,E1,
*E2,ER,AK0,AK,AKK,FA,OFFSET(7),WDELTA,OFFLIM,ERROR,THI,FI,
*NS1,NS2,NSS2,NOFF

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)
FUNCTION HZXE(X)
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,A,TPI,TPI2,PI,W,E1,
*E2,ER,EK0,A,K,AK,FA,OFFSET(7),WDELT,OFFLIM,ERROR,THI,FI,
*N1,N2,NSS2,NOFF

X2=X*X
AK0=A*K0
AK2=A*K
RM=DQSRT(AK2-X2)
RM0=DQSRT(X2-AK02)
RMH=RM*H
RMT=RM*T
RM0H=RM0*H
COSH=DQSRT(RMH)
CST=DQSRT(RMT)
SNH=DQSIN(RMH)
SXN=RM*CST
SXD=(ER*RM0*COSH-RM*SNH)*(1.0+RM0H)*(SNH/RM0-COSH/RM)
HZXE=SXD
RETURN
END

---

1) This subroutine evaluates the integrand of the Green's function at different points (subroutine Green).
2) It evaluates the space integrals coming from the application of moments' method (subroutine adonis).
3) Multiply these two values with appropriate weighting coefficients and sum them upZXX2*SAJ0(K)

SUBROUTINE FUNCT(IAD, AUP, ALOW, N)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 S1,S2
COMPLEX ZS,ZS1S2,ZI

COMMON/COMP/ZS(50),ZS1S2(350),NS,NS1S2

COMMON/MAT/PLI,AI,TI,V(3),IY

COMMON/FUT/SSJ0(250,7),SAJ0(250,7),YSIN,YGOS

COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AK,SERS(5),
*SER(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)

COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,A,TPI,TPI2,PI,W,E1,
*E2,ER,EK0,A,K,AK,FA,OFFSET(7),WDELT,OFFLIM,ERROR,THI,FI,
*N1,N2,NSS2,NOFF

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),BEOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST

COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C function at different points

SUBROUTINE GREI(X, XF, XE, IAD, TM)
IMPLICIT REAL*8 (A-H, O-Z)

COMMON/DAE/ER, H, T, DLX, XIC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPIZ, PI, W, W1,
    *E2, EER, AKD, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, FI,
    *NS1, NS2, NSS2, NOFF

COMMON/COEF/RX, XX, RZ, XZ, FRX, FRZ, F1X, F1Z

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)

CSH=DCOS(RMH)
SNH=DSIN(RMH)
CST=DCOS(RMT)
SNT=DSIN(RMT)
CSHT=DCOS(RMHT)
SNHT=DSIN(RMHT)

RM2=RM*RM
RM02=RM0*RM0
CSH2=CSH*CSH
ER0=ER*RM0
ER02=ER0*ERM0

EXX=DEXP(-X*T/FA)/FA
EXZ=DEXP(-X*(2.D0*H-T)/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)
CSHH=0.5D0*(EX+1.D0/EX)
SNHT=0.5D0*(EX-1.D0/EX)
TANHT=SNHT/CSHT

100 IF (IAD, NE, 1) GO TO 1
DEN=RM2+(ER02-RM2)*CSH2
RMNO=-RM2*SNT+(RM2-ER02)*CSH*SNHT
XNO=ER*RM0*CST
C1=X/RM

RX=C1*RNOM/DEN
XX=C1*XNOM/DEN
FRX=F1X*EXX

DEN=DEN*(RM02+AK02*(ER-1.D0)*CSH2)
RMNO=CST*(RM2+ER*RM02)*CSH*SNH
XNO=CST*RM0*RM0*(1.D0-(1.D0+ER)*CSH2)
C1=X/RM
R2=-C1*RNOM/DEN
X2=C1*XNOM/DEN
FRZ=F1Z*EXZ
RETURN
IF (IAD.NE.3) GO TO 2
C1=X*XFM
IF (DABS(AK-X).LT.1.D-6) GO TO 10
DEN=ERM0*CSHT-ERM0*SNH
RNom=(RM*CSHT-ERM0*SNH)
C2=X/RM
RX=C1*C2*RNOM/DEN

DEN=DEN*(RM*CSHT+RM0*SNH)
RNom=CST
C3=X*RM
RZ=C1*C3*RNOM/DEN

FRX=F1X*EXX
FRZ=F1Z*EXZ
RETURN

10 RNOM=1.D0-ERM0*(-H+T)
RX=C1*X*RNOM/ERM0
FRX=F1X*EXX

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*CSHT-RM*SNH
RX=(X/RM)*RNOM/DEN
FRX=F1X*EXX

RNOM=RM*CST
DEN=DEN*(RM*CSHT+RM0*SNH)
RZ=X*C1*RNOM/DEN
FRZ=F1Z*EXZ
RETURN

13 RX=X*(1.D0-ERM0*(-H+T))/ERM0
FRX=F1X*EXX

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-TANH*SNHT)
RX=(X/RM)*RNOM/DEN
FRX=F1X*EXX

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
CONTINUE
RETURN
END
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DAT/R, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TFIP, PI, W, E1,
* E2, ERR, AK0, AK, AKK, FA, OFFSET(7), WDELT, OFFLM, ERROR, THI, FFI,
* NS1, NS2, NSS2, NOFF

COMMON/DATT/C, COAL(20), P20, CN(51), BM(51), DTM(20),
* POLT(20), AM(41), DM(41), POLES(40), VXMM(20), VZXM(20), VZXE(20),
* BPOINT(10), BCALD(10), MVPNT, NPOIN, NK0, MA, NTM, NTE, NK0, IFE

COMMON/ADON(DIST(250,7,10), RCCE(20,250,7,10), AX, SERS(5),
* SERA(5), DARG(10,4), S(10,2), WREAL, NSER, NMAX(7)

COMMON/COEF/RX, XX, RX, ZX, FXX, FRZ, FX, FLX, F1Z

+-----------------------------------------------+
| Formation of the matrices: DIST, |
| DARG, RCCE |
| +-----------------------------------------------+

W2=W/2.D0
U=WREAL/W
THMIN=DATAN(DSQR(T(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=MPOINTER
IF (OFFSET(J).LE.OFFLIM) LPOINTER=NPOINTER
DO 2 I=1,LPOINTER
POIN=BPOINTER(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(I,1)=W2*AC
DARG(I,2)=AC
DARG(I,3)=AS
DARG(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
AXN=AXN*AXN
BXN=OFFSET(J)-W*DCOS(THETA)/2.D0
BXN=BXN*BXN
DIST(N,J,I)=DSQR(T(AXN2+BXN2)
SIG=DSQR(T(AXN2+BXM2)
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)*SIG**2/SIG
DSIG5=-3.D0*(-15.D0)*SIG**2*SIG+(20.D0/SIG)**3/SIG*/
DSIG2*SIG**3)/SIG2
DSIG6=-3.D0*(-15.D0)*SIG**2*(180.D0/SIG)*DSIG2
**2*DSIG**2-(120.D0/SIG2)*DSIG2*SIG**4)/
SIG2
DSIG7=-3.D0*(525.D0)*SIG**2*SIG-(2100.D0/SIG)**3/SIG*
SIG2**2*SIG**3+(840.D0/SIG2)*SIG2*SIG*
**5)/SIG3
DSIG8=-3.D0*(525.D0*DSIG2**4-(12600.D0/SIG)*DSIG2
* 3*DSIG**2+(25200.D0/DSIG2)*DSIG2**2*DSIG**4
* -(6720.D0/DSIG3)*DSIG2*DSIG**6)/SIG3

Evaluation of the coefficients Gij

G21=DSIG2
G22=DSIG**2

G41=DSIG4
G42=4.D0*DSIG3*DSIG+3.D0*DSIG2**2
G43=6.D0*DSIG2*DSIG**2
G44=DSIG**4

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

G81=8.D0*DSIG7*DSIG+28.D0*DSIG6*DSIG2+56.D0*DSIG5
* DSIG3+35.D0*DSIG4**2
G82=28.D0*DSIG6*DSIG**2+168.D0*DSIG5*DSIG2*DSIG+
  280.D0*DSIG4*DSIG3*DSIG2+210.D0*DSIG4*DSIG2**2*
  280.D0*DSIG3*DSIG2*DSIG2
G83=56.D0*DSIG5*DSIG**3+420.D0*DSIG4*DSIG2*DSIG**2
* +480.D0*DSIG3*DSIG**2+840.D0*DSIG3*DSIG2*DSIG**2
* DSIG105.D0*DSIG2**4
G84=70.D0*DSIG4*DSIG**4+560.D0*DSIG3*DSIG2*DSIG**3
* +420.D0*DSIG2*DSIG**3*DSIG**2
G85=56.D0*DSIG3*DSIG**5+210.D0*DSIG2**2*DSIG**4
G86=28.D0*DSIG2*DSIG**6
G88=DSIG**8

RCEO(2,N,J,I)=-0.5D0*(G22+SIG*G21)
RCEO(1,N,J,I)=-0.5D0*(G22-SIG*G21)

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)
RCEO(3,N,J,I)=0.5D0*(SIG*S31+SIG*S33/3.D0+G44/4.D0)
RCEO(4,N,J,I)=0.5D0*(SIG*S33+SIG*S31/3.D0-G44)
RCEO(5,N,J,I)=0.5D0*(SIG*S31+3.D0*G44/4.D0)
RCEO(6,N,J,I)=SIG*S30

SX=SIG*S31+SIG*S33/3.D0-G64
ST=SIG*S31+SIG*S33/3.D0-G64
S55=SIG2*SIG

S51=0.5D0*SIG*(SIG*S31+SIG*[3.D0+G64/4.D0)
S51=0.25D0*SIG*[0.5D0*SIG*ST-5.D0*G65/2.D0)
S53=0.25D0*SIG*[0.5D0*SIG*ST+0.25D0*SIG*STx+0.5D0*G65/4.D0)
S55=0.125D0*SIG*[0.5D0*SIG*STx-0.5*G65)
RCEO(7,N,J,I)=0.5D0*(SIG*S55/5.D0+G66/16.D0)
RCEO(8,N,J,I)=0.5D0*(SIG*S53/3.D0+SIG*[S55/5.D0-
   6.D0*G66/16.D0)
RCEO(9,N,J,I)=0.5D0*(SIG*S51+SIG*S53/3.D0+15.D0*
   G66/16.D0)
RCEO(10,N,J,I)=0.5D0*(SIG*S51-10.D0*G66/16.D0)
RCEO(11,N,J,I)=SIG*S51
RCEO(12,N,J,I)=SIG*S53
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* 
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)
RCEO(13,N,J,I)=0.5D0*(SIG*S77/7.D0+G88/64.D0)
RCEO(14,N,J,I)=0.5D0*(SIG*S75/5.D0+S77*SIG/7.D0 
  -8.D0*G88/64.D0)
RCEO(15,N,J,I)=0.5D0*(SIG*S73/3.D0+SIG*S75/5.D0 
  +28.D0*G88/64.D0)
RCEO(16,N,J,I)=0.5D0*(SIG*S71+SIG*S73/3.D0-56.D0 
  *G88/64.D0)
RCEO(17,N,J,I)=0.5D0*(SIG*S71+70.D0*G88/64.D0)
RCEO(18,N,J,I)=SIG*S7M1
RCEO(19,N,J,I)=SIG*S7M3
RCEO(20,N,J,I)=SIG*S7M5
CONTINUE

Formation of the series s(dlx) . Storage in vectors SERS(5), SERA(5)

U1=2.0*THMIN/FLOAT(NSER)
DO 6 JN=1,NSER
  S2=(2.0*FLOAT(JN)-1.0)
  S2=S2/(2.0*FLOAT(NSER))
  S3=DCOS(S2*THMIN)
  S(JN,2)=S3/W/2.0
  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)

SER1=(1.0-YCOS)*2.0/AKK
SER2=YSIN/3.0+ADL*YCOS/4.0+ADL2*YSIN/10.0-ADL3*YCOS/36.0
  *ADL4*YSIN/168.0+ADL5*YCOS/960.0+ADL6*YSIN/6480.0
SER3=YSIN/60.0-ADL*5.0*YCOS/360.0-ADL2*YSIN/168.0+ADL3
  *YCOS/560.0+ADL4*YSIN/2592.0-ADL5*YCOS/12960.0-ADL6
  *YSIN/95040.0
SER4=YSIN/2520.0+ADL*YCOS/2880.0+ADL2*YSIN/6480.0-ADL3
  *YCOS/21600.0-ADL4*YSIN/95040.0+ADL5*YCOS/518400.0
SER5=YSIN/181440.0-ADL*YCOS/201600.0-ADL2*YSIN/443520.0+ 
  ADL3*YCOS/1442775.9D0

SERS(1)=SER1*SER1
SERS(2)=DLX*2.0*SER1*SER2
SUBROUTINE ADONIS
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION BJ(10,2), DERIV(9, 3)
COMMON/ADON/DIST(250,7,10), RCOE(20,250,7,10), AX, SERS(5),
*SE(5), DARG(10,4), S(10,2), WREAL, NSER, NMAX(7)
COMMON/PUT/SSJ0(250,7), SAJ0(250,7), YSIN, YCOS
COMMON/DAT/ER,H,T,DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI, W, E,
*E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, PI,
*NS1, NS2, NS3, NOFF
COMMON/DATT/COAL(20), POINT(20), CN(51), BM(51), POLTM(20),
*POLTE(20), AM(41), DM(41), POLES(40), VXJM(20), VXJ(20), VXE(20),
*BPOINT(10), BCOAL(10), NPOINT, NPOINTER, NK0, MA, NTM, TNE, NK0K, IFIRST
COMMON/BSS/ARG(10), AARG
COMMON/MAT/PLI, AI, TI, V(3), IY
COMMON/COEF/RX, XX, RX, AX, F1X, F1Z

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

DO 11 J=1, NOFF
LPOINT=MPOINTER
DO 12 IF (OFFSET(J), GT, OFFLIM) GO TO 12
LPOINT=MPOINTER
DO 13 I=1, NPOINTER
ARG(I)=PLI*DARG(I, 1)
13 CONTINUE
CALL BESSL(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(I,4)*COAL(I)
AROF=PLI*OFFSET(J)*DARG(I,2)
COFF=DCOS(AROF)
SSUM=0.D0
DO 16 JN=1,NSER
   ARAF=PLI*S(JN,2)*DARG(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(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*SIN1
DERIV(3,3)=TERM*SIN1
SIN1=SIN1*SIN1
DERIV(4,3)=-TERM*SIN1
SIN1=SIN1*SIN1
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 BESSZ(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*SER(2)*DERIV(2,2)
   +PR4*SER(3)*DERIV(3,2)-PR6*SER(4)*DERIV
   (4,2)+PR8*SER(5)*DERIV(5,2)
   CH1=SEMA(1)*(DERIV(1,1)+DERIV(1,3)-W1*DERIV
   (1,2))
   CH2=SEMA(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 

CCCCC 
C 
IF (KJ.EQ.1) WRITE (6, 665) KJ, J, SSJ0(KJ, J), 
C 
SUMS, SAJ0(KJ, J), SUMA 
C66S 
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) 
CCCCC 
10 CONTINUE 
14 CONTINUE 
11 CONTINUE 
RETURN 
END 

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

BESS1 
This subroutine gives values for the zeroth order 
Bessel functions. It is used for small offsets 

SUBROUTINE BESS1(BJ) 
IMPLICIT REAL*8 (A-H,O-Z) 
DIMENSION BJ(10,2) 
COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z 
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) 
COMMON/BSS/ARG(10),AARG 
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), 
*BPOL(10),BGOA(10),MP0INT,NPOINT,NKO,MA,NMT,NTE,NKO0,KFIRST 
PI=3.14159265358979D0 
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 
*B36 
BJ(IJ,1)=BJ0 
X36+X36 
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 
*B36+0.0444444D0*X38-0.0039444D0*X310+0.00021D0 
D0*X312
BJ(IJ,1)=BJ0
GO TO 1

12 CONTINUE
X3=3.0D0/X
X32=X3*X3
X33=X32*X3
X34=X33*X3
X35=X34*X3
X36=X35*X3

FJ0= 0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.0000000014
* 8512D0*X33+0.00137237D0*X34-0.00072805D0*X35+0.00000014
* 476D0*X36

TJ0= X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32-0.0000000014
* 262573D0*X33-0.00054125D0*X34-0.00029333D0*X35+0.0000000014
* 13558D0*X36

WCON=DSQRT(1.0D0/X)
BJ(IJ,1)=WCON*FJ0*DCOS(TJ0)
CONTINUE

RETURN
END

This subroutine evaluates the tail contribution

SUBROUTINE TAIL
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX ZS,ZS1S2
DIMENSION MAX(8,2)

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)

COMMON/COMP/ZS(50),ZS1S2(350),NS,NS1S2

COMMON/DAT/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,TA,TP1,TP2,PI,W,E1,*E2,EER,AK0,AK,AKK,FA,OFFSET(7),WDELTA,OFFLIM,ERROR,THI,FI,*NS1,NS2,NS3,NOFF

COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XNT(40,3),*CNT(40),NDP,NTP,NSP

COMMON/ADON/DIST(250,7,10),RCE(20,250,7,10),AX,ERS(5),*ERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)

COMMON/OUT/GS(50),GS1S2(350)

COMMON/IOFF/INS,INS1S2

This vector contains the values of t in the integrals h0

Z1=T
Z2=2.0D0*t

This vector contains the values of the coefficient C in the integrals h0

C1=FA

This vector contains the number of elements of the matrices ZS,ZS1S2,...

MAX(1,1)=NS
MAX(2,1)=NS1S2

MAX(1,2)=INS
MAX(2,2)=INS1S2
This vector contains the values of the coefficient A in the integrals h0

AKK=TP1
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

YCOS=DCOS(AKK*DLX)
CCS=DCOS(2.D0*AKK*DLX)
YSIN=DSIN(AKK*DLX)
SSN=DSIN(2.D0*AKK*DLX)

+----------------------------------+
| Evaluation of S1,S2,S3,S4,S5,S6 | |
| (Single Integrals)               | |
+----------------------------------+

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

ZP1=Z1/C1
ZP2=Z2/C1
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
   DTF=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)
      11 CONTINUE
   10 CONTINUE
TRAD2=D SQRT (RAD2+ZP22)
S1(1,K,J)=S1(1,K,J)+DLOG (2.0*(TRAD1+XN))*AASIN
S1(2,K,J)=S1(2,K,J)+DLOG (2.0*(TRAD2+XN))*AASIN

CONTINUE

DO 20 I=1,NDF
THI=COEF1*XND(I,1)+COEF2
XI=DLX2*(XND(I,2)+1.D0)
C1=DCOS (THI)
C2=W2*C1
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+F L O A T (K-2)*DLX
RADP2=XNP*XNP+CW
RADM2=XNM*XNM+CW
TRAP1=D SQRT (RADP2+ZP12)
TRAP2=D SQRT (RADP2+ZP22)

TRAM1=D SQRT (RADM2+ZP12)
TRAM2=D SQRT (RADM2+ZP22)

XA1=AKK*XNP
XA2=AKK*XNM
XAP=DSIN(XA1)
XAM=DSIN(XA2)

SANP1=XAP*DLOG (2.0*(TRAP1+XNP))
SANP2=XAP*DLOG (2.0*(TRAP2+XNP))

SANM1=XAM*DLOG (2.0*(TRAM1+XNM))
SANM2=XAM*DLOG (2.0*(TRAM2+XNM))

XAP=DSIN(XA1/2.D0)
XAM=DSIN(XA2/2.D0)
SONP1=XAP/TRAP1
SONP2=XAP/TRAP2

SONM1=XAM/TRAM1
SONM2=XAM/TRAM2

Y1=-XNM/2.0-DLX
Y2=-XNP/2.0+DLX
CY1=DCOS(AKK*Y1)
CY2=DCOS(AKK*Y2)
SY1=DSIN(AKK*Y1)
SY2=DSIN(AKK*Y2)

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

CONTINUE

evaluation of T1, T2, T3, T4
C

DO 30 I=1,NTP
  THI=COEF1*XNT(I,1)+COEF2
  XI=DLX2*(XNT(I,2)+1.0D0)
  XIP=DLX2*(XNT(I,3)+1.0D0)
  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+2*P12)
  TAPP2=DSQRT(RADPP2+2*P22)
  TAPM1=DSQRT(RADPM2+2*P12)
  TAPM2=DSQRT(RADPM2+2*P22)
  TAMP1=DSQRT(RADMP2+2*P12)
  TAMP2=DSQRT(RADMP2+2*P22)
  TAMM1=DSQRT(RADMM2+2*P12)
  TAMM2=DSQRT(RADMM2+2*P22)
  CST1=DCOS(AKK*(XNPM/2.0D0+DLX)) * DSIN(AKK*XNPP
  * /2.0D0)
  CST2=DCOS(AKK*(-XNMP/2.0D0+DLX)) * DSIN(AKK*XNMM
  * /2.0D0)
  CST3=DCOS(AKK*(XNMM/2.0D0+DLX)) * DSIN(AKK*XNMP
  * /2.0D0)
  CST4=DCOS(AKK*(-XNPP/2.0D0+DLX)) * DSIN(AKK*XNPM
  * /2.0D0)
  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
31 CONTINUE
30 CONTINUE
1 CONTINUE

Evaluation of GS,Gs1S2

CZX=(1.0D0-ER)/((1.0D0+ER)*(1.0D0+E2)*(1.0D0+0.5D0*E1))
CXX=1.0D0
CS=TP1*CZX/FA
CAX=TP1*CXX/FA
CAZ=TP1*CZX/FA
DO 4 JM=1,NOFF
  NJMAX=MAX(JM,1)
  J=MAX(JM,2)
DO 62 N=1,NJMAX
  NP1=N+2
  N0=N+1
  NM1=N
4 CONTINUE

Evaluation of GS,Gs1S2
ST=D1(2,NP1,J)+2.D0*Y*OS*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)
SIN1=DSIN(AKK*FLOAT(N-2)*DLX)
SIN2=DSIN(AKK*FLOAT(N-3)*DLX)
ATX=SINP2*S1(1,MP2,J)-4.D0*Y*OS*SINP1*S1(1,MP1,J)
    +2.D0*(2.D0+CCS)*SIN0*S1(1,M0,J)-4.D0*Y*OS
    *SINM1*S1(1,MM1,J)+SINM2*S1(1,MM2,J)
AT2=SINP2*S1(2,MP2,J)-4.D0*Y*OS*SINP1*S1(2,MP1,J)
    +2.D0*(2.D0+CCS)*SIN0*S1(2,M0,J)-4.D0*Y*OS
    *SINM1*S1(2,MM1,J)+SINM2*S1(2,MM2,J)
AAX=-2.D0*(D2(1,NP1,J)-2.D0*Y*OS*D2(1,N0,J)
    +D2(1,NM1,J))
AA2=-2.D0*(D2(2,NP1,J)-2.D0*Y*OS*D2(2,N0,J)
    +D2(2,NM1,J))
AX=ATX+AAX
AZ=AT2+AA2
ZW=W*(CS*ST+CA*AX-CA*AZ
IF (JM.EQ.1) GS(N)=ZW
IF (JM.EQ.2) GS1S2(N)=ZW

This subroutine evaluates the higher order bessel functions using
the ascending series expression or hankel’s expansion.

SUBROUTINE BESS2 (BJ)
IMPLICIT REAL*8 (A-H-O-Z)
DIMENSION BJ(10,2),U(4),RBJ(50,2)
COMMON/B01/B01,B0J
COMMON/BSS/ARG(10),X
PI=3.141592653589

Evaluation of J0, J1

CALL BSJ0(X)
RBJ(1,2)=BJ0
RBJ(2,2)=BJ1

NCON=1
N=IDINT(2.4D0*X)
IF (N.LT.10) N=10
IF (X.LT.3.0D0) GO TO 10

EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS UP TO ORDER LESS THAN THE ARGUMENT

NIMAX=IDINT(X)-1
IF (NMAX.GT.9) NMAX=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 (NMAX.EQ.9) GO TO 20
NCON=NIMAX

DEBYE'S ASYMPTOTIC EXPANSION-EVALUATION OF JN

10 DO 11 J=1,2
   JN=N-J+1
   XA=X/FLOAT(JN)
   XA=1./DN/XA
   XE=XA+SDRT(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

EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS WHEN X<10

NJMAX=N-2-NCON
DO 2 I=1,NJMAX
   NJB=N-I
   NJB=NJ+1
   RBJ(NJB,2)=2./D0*FLOAT(NJB)*RBJ(NJB,2)/X-RBJ(NJB2,2)
2 CONTINUE
20 CONTINUE
20 CONTINUE
DO 3 I=1,9
   BJ(I,2)=RBJ(I,2)
3 CONTINUE
RETURN
END

SUBROUTINE BSJ0(X)
IMPLICIT REAL*8(A-H,O-Z)
COMMON/B01,BJ0,BJ1

Evaluation of J0 using the series expansion given in
Abramowitz.

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./D0*2.2499997D0*X32+1.2656208D0*X34-0.3163866D0*X36+
* 0.0444479D0*X38-0.0039444D0*X310+0.000210000D0*X312
BJ1=X*(0.50D0-0.56249985D0*X32+0.21093573D0*X34-0.03954289D0
* X36+0.00443319D0*X38-0.00031761D0*X310+0.00001109D0
* X312)
GO TO 21
20 X3=3./D0/X
X32=X3*X3
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

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

This subroutine gives all the data for integration used in
subroutine SLOT.FTN

SUBROUTINE DATA SLOT
IMPLICIT REAL*8 (A-H,O-Z)

COMMON/DA7/ER,H,T,DLX,XC,YC,ZC,ERC,RMC,X0,Y0,A,TFI,TPI2,PI,W,E1,
  *E2,ERR,AKO,AK,AKK,FA,OFFSET(7),WDELTA,OFFLIM,ERROR,THI,FI,
  *NS1,NS2,NS3,NOFF

COMMON/DATT,COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
  *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZHM(20),VXHE(20),
  *BPOINT(10),BCOAL(10),NPOINT,NPOIN,NO,MA,NTM,MT,EK,IK,IFIRST

COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XT(40,3),
  *CNT(40),NDP,NTP,NSP

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)

COMMON/IOFF/INS,INS1S2
PI=3.141592653589D0
TPI=2.0*D0*PI
TPI2=TPI*TPI

<table>
<thead>
<tr>
<th>ERROR FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1=A<em>A</em>TPI2</td>
</tr>
<tr>
<td>A2=TPI2-TPI2/ER</td>
</tr>
<tr>
<td>E1=0.5<em>D0</em>A2/A1</td>
</tr>
<tr>
<td>E2=ER*E1/(1.0+ER)</td>
</tr>
<tr>
<td>FA=DSQRT(1.0+TPI2/A1)</td>
</tr>
<tr>
<td>AK=2.0*PI</td>
</tr>
<tr>
<td>AK=AKO*DSQRT(ER)</td>
</tr>
</tbody>
</table>

Data for the poles
IFIRST=
0: dominant mode is TM wave (many poles)
1: dominant mode is TE wave (many poles)
2: only one TM surface wave

Data for the Integration
NKO=20
NKOK=1
MA=20
NSER=10
NPOINT=10

Vector COAL

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)

Vector POINT

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)

MPOINT=5

Vector BCOAL

BCOAL(1)=0.2369268851D0
BCOAL(2)=0.4786286705D0
Print file "SLOT.FTN"

BCOAL(3) = 0.5688888888D0
BCOAL(4) = BCOAL(2)
BCOAL(5) = BCOAL(1)

Vector BPOINT

BPOINT(1) = 0.9061798459D0
BPOINT(2) = 0.5384693101D0
BPOINT(3) = 0.0D0
BPOINT(4) = -BPOINT(2)
BPOINT(5) = -BPOINT(1)

Single integration

NSP = 31
RS1 = 0.99708748181D0
RS2 = 0.38468590966D0
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 = 0.0D0

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

CNS(1) = 0.0074708315792D0
Print file "SLOT.FTN"

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) = 0.097743335863D0
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)

2) Double Integration

\[ NDP=16 \]
\[ R1=\text{DSQRT}((15.0-2.0*\text{DSQRT}(30.0))/35.0) \]
\[ R2=-R1 \]
\[ S1=\text{DSQRT}((15.0+2.0*\text{DSQRT}(30.0))/35.0) \]
\[ S2=-S1 \]
\[ A1=4.0*59.0+6.0*\text{DSQRT}(30.0)/864.0 \]
\[ A2=4.0*59.0-6.0*\text{DSQRT}(30.0)/864.0 \]
\[ A3=4.0*49.0/864.0 \]

\[ XND(1,1)=R1 \]
\[ XND(1,2)=R1 \]
\[ CND(1)=A1 \]

\[ XND(2,1)=R2 \]
\[ XND(2,2)=R1 \]
\[ CND(2)=A1 \]

\[ XND(3,1)=R1 \]
\[ XND(3,2)=R2 \]
\[ CND(3)=A1 \]

\[ XND(4,1)=R2 \]
\[ XND(4,2)=R2 \]
\[ CND(4)=A1 \]

\[ XND(5,1)=S1 \]
\[ XND(5,2)=S1 \]
\[ CND(5)=A2 \]

\[ XND(6,1)=S1 \]
Print file "SLOT.FTN"

```
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

3) Triple Integration
---------------------

NTP = 34
RS1 = 0.9317380000D0
RS2 = -RS1
UU1 = 0.9167441779D0
UU2 = -UU1
SS1 = 0.4066003800D0
SS2 = -SS1
TT1 = 0.7398529500D0
TT2 = -TT1
B1 = 8.0D0 * 0.03558180896D0
B2 = 8.0D0 * 0.01247892770D0
B3 = 8.0D0 * 0.05286772991D0
B4 = 8.0D0 * 0.02672752182D0

XNT (1, 1) = RS1
XNT (1, 2) = 0.0D0
XNT (1, 3) = 0.0D0
CNT (1) = B1

XNT (2, 1) = RS2
```
XNT(2,2) = 0.0
XNT(2,3) = 0.0
CNT(2) = B1

XNT(3,1) = 0.0
XNT(3,2) = RS1
XNT(3,3) = 0.0
CNT(3) = B1

XNT(4,1) = 0.0
XNT(4,2) = RS2
XNT(4,3) = 0.0
CNT(4) = B1

XNT(5,1) = 0.0
XNT(5,2) = 0.0
XNT(5,3) = RS1
CNT(5) = B1

XNT(6,1) = 0.0
XNT(6,2) = 0.0
XNT(6,3) = RS2
CNT(6) = B1

XNT(7,1) = UU1
XNT(7,2) = UU1
XNT(7,3) = 0.0
CNT(7) = B2

XNT(8,1) = UU2
XNT(8,2) = UU1
XNT(8,3) = 0.0
CNT(8) = B2

XNT(9,1) = UU1
XNT(9,2) = UU2
XNT(9,3) = 0.0
CNT(9) = B2

XNT(10,1) = UU2
XNT(10,2) = UU2
XNT(10,3) = 0.0
CNT(10) = B2

XNT(11,1) = UU1
XNT(11,2) = 0.0
XNT(11,3) = UU1
CNT(11) = B2

XNT(12,1) = UU1
XNT(12,2) = 0.0
XNT(12,3) = UU2
CNT(12) = B2

XNT(13,1) = UU2
XNT(13,2) = 0.0
XNT(13,3) = UU1
CNT(13) = B2

XNT(14,1) = UU2
XNT(14,2) = 0.0
XNT(14,3) = UU2
CNT(14) = B2

XNT(15,1) = 0.0
XNT(15,2) = UU1
XNT(15,3)=UU1
CNT(15)=B2

XNT(16,1)=0.0D0
XNT(16,2)=UU1
XNT(16,3)=UU2
CNT(16)=B2

XNT(17,1)=0.0D0
XNT(17,2)=UU2
XNT(17,3)=UU1
CNT(17)=B2

XNT(18,1)=0.0D0
XNT(18,2)=UU2
XNT(18,3)=UU2
CNT(18)=B2

XNT(19,1)=SS1
XNT(19,2)=SS1
XNT(19,3)=SS1
CNT(19)=B3

XNT(20,1)=SS1
XNT(20,2)=SS1
XNT(20,3)=SS2
CNT(20)=B3

XNT(21,1)=SS1
XNT(21,2)=SS2
XNT(21,3)=SS1
CNT(21)=B3

XNT(22,1)=SS1
XNT(22,2)=SS2
XNT(22,3)=SS2
CNT(22)=B3

XNT(23,1)=SS2
XNT(23,2)=SS1
XNT(23,3)=SS1
CNT(23)=B3

XNT(24,1)=SS2
XNT(24,2)=SS1
XNT(24,3)=SS2
CNT(24)=B3

XNT(25,1)=SS2
XNT(25,2)=SS2
XNT(25,3)=SS1
CNT(25)=B3

XNT(26,1)=SS2
XNT(26,2)=SS2
XNT(26,3)=SS2
CNT(26)=B3

XNT(27,1)=TT1
XNT(27,2)=TT1
XNT(27,3)=TT1
CNT(27)=B4

XNT(28,1)=TT1
XNT(28,2)=TT1
XNT(28,3)=TT2
Print file "SLOT.FTN"

\begin{verbatim}
CNT(28) = B4
XNT(29, 1) = TT1
XNT(29, 2) = TT2
XNT(29, 3) = TT1
CNT(29) = B4

XNT(30, 1) = TT1
XNT(30, 2) = TT2
XNT(30, 3) = TT2
CNT(30) = B4

XNT(31, 1) = TT2
XNT(31, 2) = TT1
XNT(31, 3) = TT1
CNT(31) = B4

XNT(32, 1) = TT2
XNT(32, 2) = TT1
XNT(32, 3) = TT2
CNT(32) = B4

XNT(33, 1) = TT2
XNT(33, 2) = TT2
XNT(33, 3) = TT1
CNT(33) = B4

XNT(34, 1) = TT2
XNT(34, 2) = TT2
XNT(34, 3) = TT2
CNT(34) = B4

RETURN
END
\end{verbatim}
PROGRAM POLES
IMPLICIT REAL*8(A-H,O-Z)

ER : Dielectric constant
H : Height of the dielectric substrate
NE : Number of TE surface waves
NM : Number of TM surface waves
XS : Matrix of poles contributing to TE surface waves
XR : Matrix of poles contributing to TM surface waves
ERR : Error in the computation of the poles

DIMENSION XS(40), XR(40), LOR(40)

COMMON/DAT/ER, H, T, DLX, XG, ZC, ERC, RMC, XQ, YQ, A, TPI, TPI2, PI, W, E1,
* E2, ER, BK0, BK, BKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, FI,
* NS1, NS2, NNS2, NFF

COMMON/DAT/COAL(20), POINT(20), CN(51), BM(51), TMP(20), TEP(20),
* AM(41), DM(41), TFO(40), VXXM(20), VZXM(20), VZXE(20), BPOINT(10),
* BCOAL(10), MPOINT, NPOINT, NKO, MA, NM, NE, NK0K, IFIRST

AER=DSQRT(ER)
ER2=ER*ER
PI2=PI*PI
MASE=5
ERR=0.0000001D0
DP=H

PART I : TE MODES

AK0=2.0D0*PI
AK=DSQRT(ER)*AK0
X0=DP*DSQRT(AK**2-AK0**2)

WRITE (6,300) AK0, AK, X0, PI
300 FORMAT(10X,'AK0=',E14.7,2X,'AK=',E14.7,2X,'X0=',E14.7,
*2X,'PI=',E14.7/)

AN=X0/PI+0.5D0
NE=AN
IF (NE.EQ.0) GO TO 310
DO 2 I=1,NE
2 IF (X0-2.0D0*FLOAT(I)+1.0D0)*PI/2.0D0 3,3,4
4 XS0=(2.0D0*FLOAT(I)-1.0D0)*PI/2.0D0+ERR
XS1=(2.0D0*FLOAT(I)+1.0D0)*PI/2.0D0-ERR
GO TO 5
3 XS0=(2.0D0*FLOAT(I)-1.0D0)*PI/2.0D0+ERR
XS1=X0
5 CONTINUE
IF (DABS(XS0-XS1)-ERR) 22,7,7
XSA=(XS0+XS1)/2.0D0
Y=DTAN(XSA)*DSQRT(XO**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
   WRITE (6,301) ER,H
C301   FORMAT(/'10X,'Dielectric Constant=',D16.9/10X,'Substrate'
C     *, 'Thickness', D16.9///)
C310   IF (NE.EQ.0) WRITE (6,304)
304   FORMAT(/'10X,'No TE waves excited in the substrate'///)
305     IF (NE.EQ.0) GO TO 312
305     IF (NE.GT.0) WRITE (6,305) NE
305     FORMAT(/'10X,'There are',I4,
* ' TE waves excited in the substrate'///)
302   DO 302 I=1,NE
303     TEP(I)=XS(I)
   WRITE (6,303) I,TEP(I)
   FORMAT (10X,I4,2X,D16.9)
   TEP (I)=TEP(I)/AER
302   CONTINUE
312   CONTINUE
   END OF PART I

PART II : TM MODES

AN=X0/PI+1.D0
NM=AN
DO 13 I=1,NM
   IF (X0-(2.D0*FLOAT(I)*PI/2.D0)*PI/2.D0) 14,14,15
15   XS1=FLOAT(I)*PI/3.D0+0.01D0
   GO TO 16
14   XS1=X0
16   XS0=FLOAT(I-1)*PI+ERR
17   CONTINUE
19   XRA=(XS0+XS1)/2.D0
   WRITE (6,301) XRA
   FORMAT(10X,'XRA=','E14.7/
301
   Y=DSQRT(ER)**2*(1.D0/DTAN(XRA))*DSQRT(X0**2-XRA**2)-XRA
302     IF (Y) 20,21,24
21   XR(I)=XRA
   GO TO 333
20   XR(I)=XRA
   GO TO 17
24   XR(I)=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 306 I=1,NM
306   TMP(I)=XR(I)
   WRITE (6,306) I,XR(I)
306   FORMAT (10X,I4,2X,D16.9)
   TMP (I)=TMP (I)/AER
308   CONTINUE
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

 IF (NK.EQ.1) GO TO 416
 NNK=NNK-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(IIF)=TPO(IIP)
  LOR(IIF)=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

416 IFIRST=2

417 CONTINUE
 RETURN
 END
The name of this file is CAV_INV.FTN. It finds the inverse matrix for the case of scattering one or two slots on the ground of a dielectric substrate.

The slots have cavities at the back for minimization of the reflected field.

SUBROUTINE CAV_INV(NOR)
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX CUR, BMATR, CAV_MAT(60)
COMPLEX ZS, ZS1S2, C, CCAV, CAV, CINC, SUMC, DETA, CIN, RUN, CS1,
*CADD, SUMN

COMMON/COMP/ZS(50), ZS1S2(300), NS, NS1S2

COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI, W, E1,
*E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELT, OFFLIM, ERROR, THI, PI,
*NS1, NS2, NSS2, NOFF

COMMON/MAN/BMATR(260, 260), IA(260), IB(260)

COMMON/BESSEL/BJO(6000)

COMMON/CAV/RUN(1000), IRUN

COMMON/LOSS/TLOS_E, TLOS_M

DATA

DATA

CJ=(0.0, 1.0)
NOEL1=NS1
NOR=NS1
IDEL=20

....First Diagonal Matrix....

ARG1=PI*W/(2.0D0*YC)
ARG2=PI*Y0/YC
CALL VBJ0(ARG1, ARG2)
CALL CAVITY
SIN=DSSIN(2.0D0*PI*DLX)
CS1=DSINGL((16.0D0/120.0D0)*DSQRT(ER)/(PI**3*XC*YC*RMC*SIN*SIN)) /
*(1.0-CJ*DSINGL(TLOS_M))
CCAV=CJ*CS1
C1=PI/XC
IMIN=1
IMAX=NOEL1
DO 4 I=IMIN, IMAX
   IXN=0
   DO 5 KI=I, IMAX
      IXN=IXN+1
      XJ=(IXN*DLX+X0)*C1
      XI=(KI*DLX+X0)*C1
      SUMN=(0.0, 0.0)
      DO 200 IN=1, IRUN
         SINI=DSSIN((FLOAT(IN))*XI)
         SINJ=DSSIN((FLOAT(IN))*XJ)
         CADD=RUN(IN)*DSINGL(SINI*SINJ)
         SUMN=SUMN+CADD
      200 CONTINUE
   CAV=CCAV*SUMN
   IF (IXN.EQ.1) CAV_MAT(KI)=CAV
   BMATR(IXN, KI)=CS(I)+CAV
   BMATR(KI, IXN)=BMATR(IXN, KI)
4 CONTINUE
CONTINUE
300 CONTINUE
C
DO 400 I=1,NS1
WRITE (6,401) I,CAV_MAT(I),ZS(I)
401 FORMAT (10X,'I=',I4,2X,'CAV=',E14.7,2X,E14.7,2X,
       * 'ZS=',E14.7,2X,E14.7/)
400 CONTINUE
C
CALL MINVCD (NOR,NOR,DETA)
RETURN
END
C******************************************************************************
C THIS SUBROUTINE INVERTS A SQUARE COMPLEX MATRIX
C******************************************************************************
SUBROUTINE MINVCD (IA,MA,DETA)
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX A,PIV,DETA,TEMP,PIV1
COMMON/MAN/A(260,260),IR(260),IC(260)
DO 1 I=1,MA
1 IR(I)=0
1 IC(I)=0
C
DETA=(1.00,0.00)
S=0.00
R=MA
2 CALL SUBMCD (IA,IA,MA,MA,I,J)
P=PIV=0.00
C
DETA=PIV*DETA
Y=CABS(PIV)
IF (Y.EQ.0) GO TO 17
IR(I)=J
IC(J)=I
PIV=(1.00,0.00)/PIV
A(I,J)=PIV
DO 5 K=1,MA
5 IF (K.NE.J) A(I,K)=A(I,K)*PIV
DO 9 K=1,MA
9 IF (K.EQ.I) GO TO 9
PIV1=A(K,J)
DO 8 L=1,MA
8 IF (L.NE.J) A(K,L)=A(K,L)-PIV1*A(I,L)
9 CONTINUE
DO 11 K=1,MA
11 IF (K.NE.I) A(K,J)=-PIV*A(K,J)
S=S+1.00
IF (S.LT.R) GO TO 2
DO 12 I=1,MA
12 K=IC(I)
M=IR(I)
IF (K.EQ.I) GO TO 16
DETA=-DETA
DO 14 L=1,MA
TEMP=A(K,L)
A(K,L)=A(I,L)
14 A(I,L)=TEMP
DO 15 L=1,MA
TEMP=A(L,M)
A(L,M)=A(L,I)
15 A(L,I)=TEMP
IC(M)=K
IR(K)=M
16 CONTINUE
RETURN
17 WRITE (6,18)I,J
18 FORMAT (10X,'MATRIX IS SINGULAR'/10X,'I=',I4,5X,'J=',I4)
RETURN
SUBROUTINE SUBMCD(IA,JA,MA,NA,I,J)
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX A
COMMON/MAN/A(260,260),IR(260),IC(260)
I=0
J=0
TEST=0.00
DO 5 K=1,MA
   IF (IR(K).NE.0) GO TO 5
   DO 4 L=1,NA
      IF (IC(L).NE.0) GO TO 4
      X=CABS(A(K,L))
      IF(X.LT.TEST) GO TO 4
      I=K
      J=L
      TEST=X
   4 CONTINUE
5 CONTINUE
RETURN
END
SUBROUTINE CAVITY
IMPLICIT REAL*8 (A-H,O-Z)
COMPLEX RUN, CRNER, CK2, CI, COEF, SERM, CRNM, CSQR, CARG, CTERM, CCOT,
* SERN

COMMON/CAV/RUN(1000), IRUN

COMMON/BESSEL/BJ0(6000)

COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, XO, Y0, A, TPI, TPI2, P1, W, E1,
* E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, FI,
* NS1, NS2, NSS2, NOFF

COMMON/LOSS/TLOS_E, TLOS_M

CI=(0.0, 1.0)
ERROR=1.0-4
PI=3.14159265358979D0
SERM=(0.0, 0.0)
NTEST=0
DO 1 N=1,1000
EN=1.0D0
RN=FLOAT(N)/(2.0D0*XC)
RN2=RN*RN
RN1=1.0D0+RN
RNM=1.0D0-RN
RNPM=RN1*RNM
RNPM2=RNPM*RNPM
CK2=(1.0-CI*SINGL(TLOS_E))*(1.0-CI*SINGL(TLOS_M))*SINGL(ERC*RMC /
* ER)
CRNER=CK2-SINGL(RN2)
SIN1=DSIN(P1*DLX*RN1)
SIN2=DSIN(P1*DLX*RNM)
SINP1=SIN1*SIN2
SINP2=SINP1*SINP
COEF=CRNER*SINGL(SINP2*EN/RNPM2)
SERM=(0.0, 0.0)
NTEST=0
M1=0
M1=M1+1
M=M1-1
EM=0.5D0
IF (M.GT.0) EM=1.0D0
BJ=BJ0(M1)
RM=FLOAT(M)/(2.0D0*YC)
RM2=RM*RM
CRNM=CRNER-SINGL(RM2)
CABS1=CABS(CRNM)
IF (CABS1 LT 1.0E-8) GO TO 100
CSQR=CSQRT(CRNM)
CARG=SINGL(2.0D0*P1*ZC)*CSQR
IF (CABS1 LT 50.0) THEN
CTERM=SINGL(EM*BJ)*CCOS(CARG)/(CSQR*CSIN(CARG))
ELSE
RARG=REAL(CARG)
AARG=AIMAG(CARG)
COT=DCOS(RARG)/DSIN(RARG)
EX=EXP(-2.0D0*AARG)
COTH=(1.0D0+EX)/(1.0D0-EX)
CCOT=(-1.0-CI*SINGL(COT*COTH))/(SINGL(COT)-
* CI*SINGL(COTH))
CTERM=SINGL(EM*BJ)*CCOT/CSQR
ENDIF
SERM=SERM+CTERM
RATIO=CABS(CTERM/SERM)
IF (RATIO.LT.ERROR) GO TO 5
  ITEST=0
  GO TO 2
5
  ITEST=ITEST+1
  IF (ITEST.LT.4) GO TO 2

WRITE (6,10) N,M,SERM
10  FORMAT ('/6X,'Maximum M for a given N:',2X,'N=',I4,3X,'Mmax=',I4,2X,'Serr=',E14.7,1X,E14.7)

RUN(N)=COEF*SERM
WRITE (6,40) N,RUN(N)
40  FORMAT(10X,'N=',I4,3X,'RUN=',E14.7,2X,E14.7))
SESN=SESN+RUN(N)
RATIO=CABS(RUN(N)/SESN)
IF (RATIO.LT.ERROR) GO TO 8
  NTEST=0
  GO TO 1
8
  NTEST=NTEST+1
  IF (NTEST.LT.4) GO TO 1
  IRUN=N
  GO TO 101
1
  CONTINUE
  GO TO 101

100  WRITE (6,20)
20  FORMAT(/10X,'WARNING! WARNING!/10X,'The eigenvalue Kz'
  ','became equal to 0'/10X)

101  CONTINUE
  RETURN
END

******************************************************************************
;; This function evaluates the zeroth order first kind Bessel
;;
;; Function J0
******************************************************************************

SUBROUTINE VBJ0(ARG1,ARG2)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/BESEL/BJ0 (6000)
PI=3.1415926535899D0
DO 1 M=1,6000
  M=M+1
  X=FLOAT(M)*ARG1
  X1=FLOAT(M)*ARG2
  COS1=DCOS(X1)
  COS2=COS1*COS1
  IF (X.GT.0.001D0) GO TO 10
    X3=X/3.D0
    X32=X3*X3
    X34=X32*X32
    X36=X34*X32
    BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
    *X36
    BJ0(M1)=BSJ0*COS2
    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
    BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
Print file "CAVITY.FTN"

* X36+0.0444479D0*X38-0.0039444D0*X310+0.00021000
* D0*X312
BJ0 (M1) = BSJ0*COS2
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+0.0000
* 13558B0*X36
WCON=DSQRT (1.D0/X)
BSJ0=WCON*FJ0*DCOS (TJ0)
BJ0 (M1) = BSJ0*COS2
1 CONTINUE
RETURN
END

C******************************************************************************
*--------------------------------------------------------------------
* The name of this file is:  MULT_EXC.FTN
*--------------------------------------------------------------------
* This subroutine evaluates the excitation vector and multiplies it
* with the inverse of the impedance matrix to get the unknown field
* on the slot
*--------------------------------------------------------------------

SUBROUTINE MULT_EXC(NOR, R0)
IMPLICIT REAL*8 (A-H, O-2)
DIMENSION Y(260), PHASE(260)
COMPLEX CUR, BMATR, CJ, CCAV, CAV, CINC, SUMC, CIN, R0

COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI, W, E1,
* E2, EER, AKO, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIM, ERROR, THI, PI,
* NS1, NS2, NSS2, NOFF

COMMON/MAN/BMATR(260, 260), IA(260), IB(260)

COMMON/PAT/CUR(260)

DATA CJ=(0.0, 1.0)

RNO=120.0*D0*PI
RN=RNO/DSQRT ER
RK=2.0*PI
RK0=2.0/DSQRT ER
SINI=DSIN THI
COSI=DCOS THI
SINT2=SINT*SINT
SINT=DSQRT ER
SINT2=SINT2
COST=DSQRT 1.00
ARGT=ARGT
ARGH=ARGH
ARGL=ARGL
ARGL2=ARGL2*SINT
D1=4.0*COSI*DCOS (FI)*DCOS (ARGT) *(DCOS (ARGL2) -DCOS (ARGL))
* / (RK*(1.00 -SINT2))
DC=RN0*COST*DCOS (ARGH)
DS=RN0*COSI*DSIN (ARGH)
CINC=SNGL(D1)/(SNGL(DC) +CJ*SNGL (DS))
ARG=ARGH
DC=RN0*COST
DS=RN0*COSI*DTAN (ARG)
R0=-(SNGL(DC) -CJ*SNGL (DS))/(SNGL(DC) +CJ*SNGL (DS))
DO 70 IQ=1, NOR
SUMC=(0.0, 0.0)

401 DO 170 JQ=1, NOR
ARG=ARGL2*FLOAT(JQ) +X0
EC=DCOS (ARG)
ES=DSIN (ARG)
CINC=INC*(SNGL(EC) -CJ*SNGL (ES))
WRITE (6, 778) IQ, JQ, CIN, BMATR(IQ, JQ)
* 
SUMC=SUMC-BMATR(IQ, JQ) *CIN

CONTINUE
CUR(IQ)=SUMC

70 CONTINUE
WRITE (6, 71)
71 FORMAT (//10X,'Electric Field Distribution on the Slots',////)
IMIN=1
IMAX=NOR
DO 76 IQQ=IMIN,IMAX
  RECUR1=REAL(CUR(IQQ))
  Y(IQQ)=CABS(CUR(IQQ))
  AICUR1=AIMAG(CUR(IQQ))
  PHCUR1=ATAN2(AICUR1,RECUR1)
  PHASE(IQQ)=180.00*PHCUR1/PI
  IF (IQQ.EQ.1) WRITE (6,77)
  IF (IQQ.EQ.(NS1+1)) WRITE (6,78)
77 FORMAT(///10X,'Electric Field on the first Slot',///)
78 FORMAT(///10X,'Electric Filed on the Second Slot',///)
C
  WRITE (6,81) IQQ,CUR(IQQ),Y(IQQ),PHASE(IQQ)
81 FORMAT (1X,'C(',I4,')=(',E14.7,',',E14.7,')',2X,
              '(',E14.7,',',E14.7,')',/)
C
  WRITE (1,92) IQQ,CUR(IQQ),Y(IQQ),PHASE(IQQ)
92 FORMAT(I5,4E15.3)
C
76 CONTINUE
RETURN
END
The name of this file is: \textquote{REFLEC.FTN}

It computes the reflection coefficient for cavity-backed slots.

```
SUBROUTINE REFLEC(WREAL, R0, REFL)
  IMPLICIT REAL*8 (A-H, O-Z)
  COMPLEX ETH, EFI, CUR, REFL, R0, RS, RA

  COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI,
  *WIDTH, E1, E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIN, ERROR,
  *THI, FI, NS1, NS2, NSS2, NOFF

  COMMON/PAR/NSLOT, ALONG(7), S(20, 2), NSER

  COMMON/PAT/CUR(260)

  NSLOT=NOFF

  NSER=10
  U=WREAL/WIDTH
  U=DATAN(DSQR1(1.0D0/(U*U)-1.0D0))
  U1=U/FLOAT(NSER)
  DO 3 JN=1, NSER
       S2=2.0D0*FLOAT(JN)-1.0D0
       S2=S2/(2.0D0*FLOAT(NSER))
       S3=COS(S2*U)
       S(JN, 2)=S3*WIDTH/2.0D0
       S(JN, 1)=U1
  3 CONTINUE

  ATH1=WREAL/WIDTH
  ATH1=DATAN(DSQR1(1./((ATH1*ATH1)-1.0)))
  ATH2=PI-ATH1
  CURIN=1.0D0/PI)*((ATH2-ATH1)
  CALL EFIELD(ETH, EFI)
  RS=(ETH*SNGL(DCOS(THI)*DSIN(FI))+EFI*SNGL(DCOS(FI)))*
  *SNGL(CURIN)
  RA=2.0*RS/NSER(DSQR1(ER))
  REFL=R0+RA

  WRITE (6,100) THI, FI, R0, RA
  100 FORMAT(10X,'THI=', E14.7, 2X,'FI=', E14.7/10X,'R0=', E14.7, 1X, E14.7
  */10X,'RA=', E14.7, 2X, E14.7)

  RETURN
END
```

The name of this subroutine is: \textquote{EFIELD}

It evaluates the far field of a dipole.

```
SUBROUTINE EFIELD(ETH, EFI)
  IMPLICIT REAL*8 (A-H, O-Z)
  COMPLEX W, ETH, EFI, XNUM, XDEN, F1X, ZNUM, ZDEN, F1Z, PTH, PFI,
  *SUMC, WEXP, CUR, COEF

  COMMON/DAT/ER, H, T, DLX, XC, YC, ZC, ERC, RMC, X0, Y0, A, TPI, TPI2, PI,
  *WIDTH, E1, E2, EER, AK0, AK, AKK, FA, OFFSET(7), WDELTA, OFFLIN, ERROR,
  *THI, FI, NS1, NS2, NSS2, NOFF

  COMMON/PAR/NSLOT, ALONG(7), S(20, 2), NSER

  COMMON/PAT/CUR(260)

  COMMON/B01/BJ0, BJ1
```
CKK=2.0*D0*PI
CK0=CKK/DSQRT(ER)
SINFT=DSIN(FI)*DSIN(THI)
ARG=(CK0*WIDTH/2.00)*SINFT
CALL BS30(ARG)
SSUM=0.0
DO 5 JN=1,NSER
       ARAF=CK0*S(JN,2)*SINFT
       CAFF=DCOS(ARAF)
       SSSU=SSUM+S(JN,1)*CAFF
5 CONTINUE
TERM1=(BJ0-SSUM*2.00/PI)

C
ERTH=DSQRT(ER-DSIN(THI)**2)
ERH=CK0*ERTH*H
SINH=DSIN(ERH)
SINFT=DSIN(THI)
SINFI=DSIN(FI)
COSH=DCOS(ERH)
COSTH=DCOS(THI)
COSFI=DCOS(FI)
W=(0.0,1.0)
XNUM=SNGL(COSTH)
XDEN=SNGL(-ERH*SINH)+W*SNGL(ER*COSTH*COSH)
FIX=SNGL(ER/(4.00*PI))*XNUM/XDEN
ZNUM=SNGL(SINH*SINFT*COSTH)
ZDEN=XDEN*(-SNGL(COSTH*SINH)+W*SNGL(ERH*COSH))
FIZ=SNGL((1.00-ER)/(4.00*PI))*ZNUM/ZDEN
PTH=SNGL(COSFI)*(FIX*SNGL(COSTH)-FIZ*SNGL(SINH))
PFI=SNGL(SINFI)*FIX

For the single slot

SUMC=(0.0,0.0)
DO 1 I=1,NSLOT
       R8=CK0*(FLOAT(I)*DLX)*SINTH*COSFI
       SUMC=SUMC+CUR(I)*(SNGL(DCOS(R8))+SNGL(DSIN(R8)))*W
1 CONTINUE
R8=CK0*ALONG(I)*SINTH*COSFI
R9=CK0*OFFSET(I)*SINFT*SINTH
WEXP=(SNGL(DCOS(R8))+SNGL(DSIN(R8)))*W*(SNGL(DCOS(R8))+SNGL(DSIN(R8)))*W)
COEF=WEXP*SNGL(TERMI)
PTH=PTH*COEF
PFI=PFI*COEF
IF (ABS(FI).GT.1.E-4) GO TO 2
THER=ABS(ABS(THETA)-PI/2.0)
IF (THER.GE.1.E-4) GO TO 2
IF (ABS(ER-1.00).LT.1.E-6) GO TO 3
2 RNUM=DCOS(CK0*DLX*SINTH*COSFI)-DCOS(CK0*DLX)
RDEN=DSIN(CK0*DLX)*(1.00-(CK0*SINTH*COSFI/CKK)**2)
RATIO=RNUM/RDEN

ETH=PFI*SNGL(RATIO)
EFI=PTH*SNGL(RATIO)
ETH=ETH*SUMC
EFI=EFI*SUMC

WRITE (6,4) ETH, EFI
4 FORMAT (5X,'ETH=',(E14.8,1X,E14.8),5X,'EFI=',*(E14.8,1X,E14.8)/*)

RETURN
Print file "REFLEC.FTN"

3 RATIO=DLX*CK0/2.0
ETH=PFI*SNGL(RATIO)*SUMC
EFI=-PTH*SNGL(RATIO)*SUMC
RETURN
END