

NASA Technical Memorandum 101614

CWG - A Fortran Program for Mutual Coupling in a Planar Array of Circular Waveguide-fed Apertures

M. C. Bailey

June 1989

(NASA-TM-101614) CWG: A FORTRAN PROGRAM FOR
MUTUAL COUPLING IN A PLANAR ARRAY OF
CIRCULAR WAVEGUIDE-FED APERTURES (NASA,
Langley Research Center) 46 p CSCL 171

N89-27063

G3/32 Unclass
0224584



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Introduction

Circular waveguide-fed apertures or conical horns are often used as elements of planar array antennas. The purpose of this present paper is to document a computer code which calculates the mutual coupling between circular apertures in a conductive plane. The program is quite general in that the apertures do not have to be the same sizes, nor do they have to be polarized in the same direction. In addition, several waveguide modes (TE and/or TM) may be specified in the apertures and the mutual coupling between all combinations of apertures and modes will be calculated. The program also allows multiple layers of homogeneous dielectrics to be placed over the aperture array. Outside the layered region, one can specify either a homogeneous half-space or a perfect reflecting surface. The program is a revision of an earlier one (ref.1) in an attempt to overcome some difficulties encountered with the earlier version and to expand its versatility.

Symbols

$a_{i_{th}}$	Radius of i_{th} aperture.
$a_{j_{th}}$	Radius of j_{th} aperture.
d_n	Thickness of n_{th} homogeneous dielectric layer.
$E_{i_{th}}$	i_{th} aperture electric field polarization vector.
$E_{j_{th}}$	j_{th} aperture electric field polarization vector.
$J_m(z)$	Bessel function of the first kind of order m and argument z .
$J'_m(z)$	Derivative of $J_m(z)$ with respect to z .
j	$\sqrt{-1}$
k_0	Wave propagation constant in free space, $2\pi/\lambda$.
k_n	Wave propagation constant in n_{th} layer.
k_{N+1}	Wave propagation constant in homogeneous half-space.
$m_{i_{th}}$	First index of waveguide mode in i_{th} aperture.
$m_{j_{th}}$	First index of waveguide mode in j_{th} aperture.
$n_{i_{th}}$	Second index of waveguide mode in i_{th} aperture.
$n_{j_{th}}$	Second index of waveguide mode in j_{th} aperture.
N	Number of homogeneous layers outside array plane.
R	Center-to-center spacing between apertures.
$R_{\parallel}(\beta)$	Reflection coefficient for plane wave with electric field polarized parallel to plane of incidence.
$R_{\perp}(\beta)$	Reflection coefficient for plane wave with electric field polarized perpendicular to plane of incidence.
$R_{\parallel}^P(\beta)$	Plane wave reflection coefficient for parallel polarization for exterior scattering problem.
$R_{\perp}^P(\beta)$	Plane wave reflection coefficient for perpendicular polarization for exterior scattering problem.
S_{ij}	Mutual coupling between i_{th} and j_{th} apertures for one mode in each aperture.
$[S]$	Complex square matrix whose elements are S_{ij} .
TE	Denotes transverse (to z) electric field.
TM	Denotes transverse (to z) magnetic field.
x, y, z	Cartesian coordinate variables.

x_i	x-coordinate of i_{th} aperture center.
x_j	x-coordinate of j_{th} aperture center.
Y_i	y-coordinate of i_{th} aperture center.
Y_j	y-coordinate of j_{th} aperture center.
Y_{ij}	Mutual admittance between i_{th} and j_{th} apertures for one mode in each aperture.
$[Y]$	Complex square matrix whose elements are Y_{ij} .
$[Y_0]$	Complex diagonal matrix whose elements are the waveguide modal characteristic admittances.
β	Normalized radial propagation constant in cylindrical coordinate system.
δ	Distance from outer surface of N_{th} dielectric layer to exterior scattering boundary.
ϵ_0	Permittivity of free space.
ϵ_1	Permittivity of region just outside aperture plane.
ϵ_n	Permittivity of n_{th} homogeneous layer.
ϵ_{N+1}	Permittivity of homogeneous half-space outside of N_{th} layer.
ϵ_p	Equivalent permittivity just inside of exterior scattering boundary.
λ	Wavelength in free space.
μ_0	Permeability of free space.
μ_1	Permeability of region just outside of aperture.
μ_n	Permeability of n_{th} homogeneous layer.
μ_{N+1}	Permeability of homogeneous half-space outside of N_{th} layer.
μ_p	Equivalent permeability just inside of exterior scattering boundary.
ϕ_i	Polarization of i_{th} aperture with respect to y-axis.
ϕ_j	Polarization of j_{th} aperture with respect to y-axis.
ϕ_p	$\phi_j - \phi_i$.
χ_{mn}	n_{th} zero of $J_m(z)$.
χ'_{mn}	n_{th} zero of $J'_m(z)$.

Theory

The analytical formulation for the electromagnetic interaction between apertures in a planar array has been developed earlier (ref.1) and will not be repeated here.

The mutual interaction between the various modes among all the apertures in the array is obtained from the appropriate coefficients of the scattering matrix, [S], as follows:

$$[S] = \left[[Y_0] - [Y] \right] \left[[Y_0] + [Y] \right]^{-1} \quad (1)$$

Where [Y₀] is a diagonal matrix whose elements are the characteristic admittances of the waveguide modes and [Y] is the admittance matrix whose elements are the mutual admittances between each waveguide mode and all modes in each waveguide aperture. Therefore, for example, if the array consisted of 5 apertures and each aperture field was assumed to be the superposition of 7 waveguide modes (TE and/or TM), then (1) would consist of square complex matrices each of size 35 by 35. The sizes of the matrices can expand very rapidly if one does not use some "engineering judgment" in selecting the modes to approximate the aperture fields. The expressions for calculating the elements of the admittance matrix (i.e., the complex mutual admittance between pairs of waveguide modes) are derived in reference 1.

$$Y_{ij} = -\sqrt{\epsilon_0/\mu_0} \sqrt{\gamma_{m_i} \gamma_{m_j}} \int_0^\infty \left\{ W_1(\beta) \xi_1(\beta) \xi_j(\beta) U_{ij}(\beta) - W_2(\beta) \zeta_1(\beta) \zeta_j(\beta) V_{ij}(\beta) \right\} \beta d\beta \quad (2)$$

where,

$$\begin{aligned} \gamma_{m_i} &= 1 && \text{for } m_i = 0 \\ &= 2 && \text{otherwise.} \end{aligned}$$

$W_1(\beta)$ and $W_2(\beta)$ are defined as

$$W_1(\beta) = \left(\frac{1 - R_{\parallel}(\beta)}{1 + R_{\parallel}(\beta)} \right) / \sqrt{1 - \beta^2} \quad (3)$$

$$W_2(\beta) = \left(\frac{1 - R_{\perp}(\beta)}{1 + R_{\perp}(\beta)} \right) \sqrt{1 - \beta^2} \quad (4)$$

where (for real angles) β can be interpreted as the sine of the angle of incidence for a plane wave with $R_{\parallel}(\beta)$ and $R_{\perp}(\beta)$ being the reflection coefficients in the outward direction evaluated at the aperture boundary for parallel and perpendicular polarization at an air-dielectric interface. Notice from equation 2 that the mutual admittance is a weighted sum over all angles of incidence, including real space ($0 \leq \beta \leq 1$) and invisible space ($\beta > 1$). An alternate form for $W_1(\beta)$ and $W_2(\beta)$ is

$$W_1(\beta) = \left\{ \frac{-jk_0 \epsilon_1}{\epsilon_0} \right\} \left(\frac{g_1(\beta)}{g'_1(\beta)} \right) \quad (5)$$

$$W_2(\beta) = \left\{ \frac{\mu_0}{-jk_0 \mu_1} \right\} \left(\frac{f'_1(\beta)}{f_1(\beta)} \right) \quad (6)$$

where $g_1(\beta)$ and $f_1(\beta)$ are the solutions to the wave equations in the external region evaluated at the aperture plane, and $g'_1(\beta)$ and $f'_1(\beta)$ are the corresponding derivatives in the direction normal to the aperture plane. ϵ_1 and μ_1 are the permittivity and permeability of the external medium evaluated just outside the aperture plane.

By applying the boundary conditions of continuity of tangential fields at the interface of each homogeneous dielectric layer, one can evaluate the mutual admittance between aperture modal fields with a multilayer dielectric medium outside (as shown in figure 1).

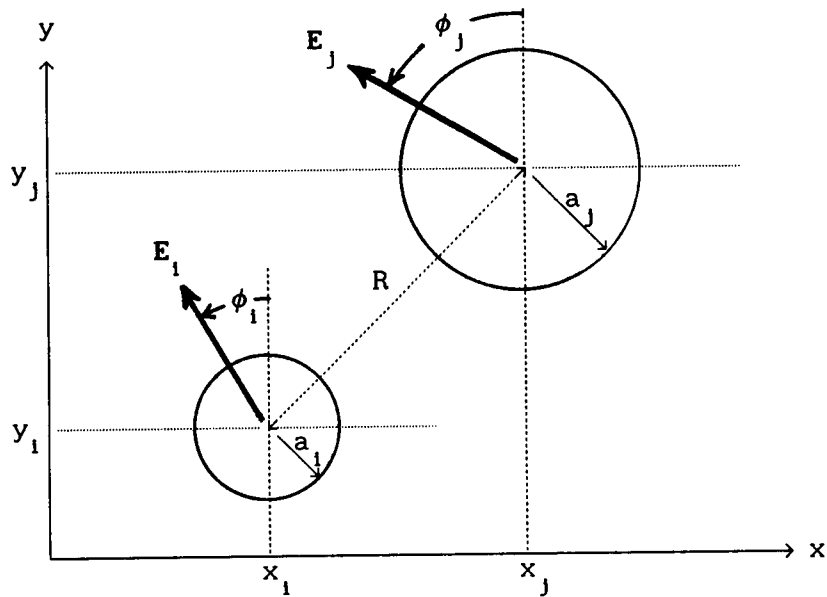


Figure 1a. Coordinate geometry for the i_{th} and j_{th} elements of a planar array of circular waveguide-fed apertures.

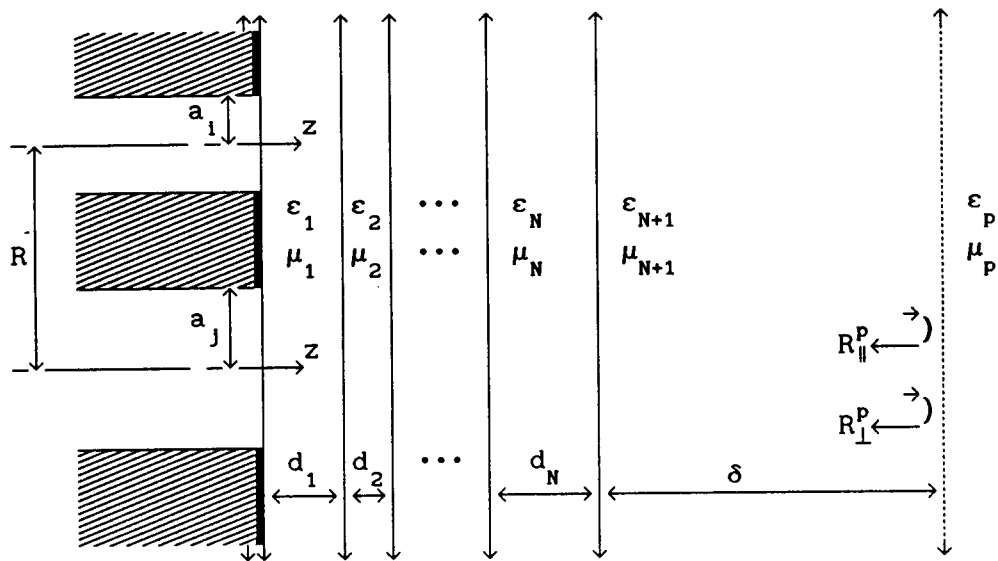


Figure 1b. Cross-section of i_{th} and j_{th} circular waveguide-fed apertures radiating into a multi-layer dielectric with an exterior arbitrary reflecting boundary.

Applying boundary conditions at layer interfaces, results in the following recursive equations,

$$\begin{pmatrix} f'_n(\beta) \\ f_n(\beta) \end{pmatrix} = \left\{ \frac{(k_z) \left\{ \sin(k_z d_n) + \left(\frac{\mu_n}{k_z \mu_{n+1}} \right) \left(\frac{f'_{n+1}(\beta)}{f_{n+1}(\beta)} \right) \cos(k_z d_n) \right\}}{\left\{ \cos(k_z d_n) - \left(\frac{\mu_n}{k_z \mu_{n+1}} \right) \left(\frac{f'_{n+1}(\beta)}{f_{n+1}(\beta)} \right) \sin(k_z d_n) \right\}} \right\} \quad (7)$$

$$\begin{pmatrix} g'_n(\beta) \\ g_n(\beta) \end{pmatrix} = \left\{ \frac{(k_z) \left\{ \sin(k_z d_n) + \left(\frac{\epsilon_n}{k_z \epsilon_{n+1}} \right) \left(\frac{g'_{n+1}(\beta)}{g_{n+1}(\beta)} \right) \cos(k_z d_n) \right\}}{\left\{ \cos(k_z d_n) - \left(\frac{\epsilon_n}{k_z \epsilon_{n+1}} \right) \left(\frac{g'_{n+1}(\beta)}{g_{n+1}(\beta)} \right) \sin(k_z d_n) \right\}} \right\} \quad (8)$$

for $n = N, (N-1), (N-2), \dots, 3, 2, 1$. where k_z is defined as

$$k_z = \begin{cases} \sqrt{k_n^2 - k_0^2 \beta^2} & ; \beta \leq k_n/k_0 \\ -j \sqrt{k_0^2 \beta^2 - k_n^2} & ; \beta > k_n/k_0 \end{cases}$$

Then starting with

$$\begin{pmatrix} f'_{N+1}(\beta) \\ f_{N+1}(\beta) \end{pmatrix} = \begin{pmatrix} g'_{N+1}(\beta) \\ g_{N+1}(\beta) \end{pmatrix} = \begin{cases} -j \sqrt{k_{N+1}^2 - k_0^2 \beta^2} & ; k_0 \beta \leq k_{N+1} \\ -\sqrt{k_0^2 \beta^2 - k_{N+1}^2} & ; k_0 \beta > k_{N+1} \end{cases} \quad (9)$$

equations 7 and 8 are evaluated successively from the outermost homogeneous layer to the layer just outside of the aperture plane.

In the special case of a perfect electric conductor of infinite extent on the outer surface of the N_{th} layer, $f_{N+1}(\beta)=0$ and $g'_{N+1}(\beta)=0$; therefore, the starting conditions for equations 7 and 8 become

$$\left(\frac{f'_N(\beta)}{f_N(\beta)} \right) = -k_z \left\{ \frac{\cos(k_z d_N)}{\sin(k_z d_N)} \right\} \quad (10a)$$

$$\left(\frac{g'_N(\beta)}{g_N(\beta)} \right) = k_z \left\{ \frac{\sin(k_z d_N)}{\cos(k_z d_N)} \right\} \quad (10b)$$

If the medium outside of the N_{th} dielectric layer is not a homogeneous half-space, the starting conditions (equation 9) are replaced by

$$\left(\frac{f'_{N+1}(\beta)}{f_{N+1}(\beta)} \right) = -j K(\beta) \left(\frac{\mu_p}{\mu_{N+1}} \right) \left\{ \frac{1 - R_{\perp}^p(\beta) \exp(-j2\delta K(\beta))}{1 + R_{\perp}^p(\beta) \exp(-j2\delta K(\beta))} \right\} \quad (11a)$$

$$\left(\frac{g'_{N+1}(\beta)}{g_{N+1}(\beta)} \right) = -j K(\beta) \left(\frac{\epsilon_p}{\epsilon_{N+1}} \right) \left\{ \frac{1 + R_{\parallel}^p(\beta) \exp(-j2\delta K(\beta))}{1 - R_{\parallel}^p(\beta) \exp(-j2\delta K(\beta))} \right\} \quad (11b)$$

where,

$$K(\beta) = \begin{cases} \sqrt{k_{N+1}^2 - k_0^2 \beta^2} & ; k_0 \beta \leq k_{N+1} \\ -j \sqrt{k_0^2 \beta^2 - k_{N+1}^2} & ; k_0 \beta > k_{N+1} \end{cases}$$

$R_{\perp}^p(\beta)$ and $R_{\parallel}^p(\beta)$ are the plane wave reflection coefficients, evaluated at the external scattering boundary (shown dashed in figure 1b) for perpendicular and parallel polarization of the electric field vector with respect to the plane of incidence. ϵ_p and μ_p are the equivalent permittivity and equivalent permeability just inside of the external scattering boundary. The starting

conditions in equation 11 allow for future modifications of the computer code to include problems for which the plane wave reflection is known from solutions to exterior scattering problems, such as objects in the vicinity of the array or an inhomogeneous ionized plasma.

The other quantities in equation 2 are related to the Fourier transforms of the aperture modal fields as derived in reference 1.

For the TE modes in the i_{th} aperture:

$$\zeta_1^{TE}(\beta) = \frac{(\chi'_{m_1 n_1})^2 (k_0 a_1) J'_{m_1}(k_0 a_1 \beta)}{\left\{ (\chi'_{m_1 n_1})^2 - (k_0 a_1 \beta)^2 \right\} \sqrt{(\chi'_{m_1 n_1})^2 - m_1^2}} \quad (12)$$

$$\xi_1^{TE}(\beta) = \frac{(k_0 a_1) m_1 J_{m_1}(k_0 a_1 \beta) / (k_0 a_1 \beta)}{\sqrt{(\chi'_{m_1 n_1})^2 - m_1^2}} \quad (13)$$

For the TM modes in the i_{th} aperture:

$$\zeta_1^{TM}(\beta) = 0 \quad (14)$$

$$\xi_1^{TM}(\beta) = \frac{(k_0 a_1) (k_0 a_1 \beta) J_{m_1}(k_0 a_1 \beta)}{(\chi_{m_1 n_1})^2 - (k_0 a_1 \beta)^2} \quad (15)$$

For the j_{th} aperture, the i subscripts are replaced with j in equations 12-15.

If $R=0$, then the i_{th} and j_{th} apertures are coincident and equation 2 is the mutual admittance between modal fields in the same aperture. When $R=0$, the orthogonality of the modal functions results in $U_{ij}(\beta)=V_{ij}(\beta)=0$ except when $m_i=m_j$. When $R=0$ and $m_i=m_j$; $U_{ij}(\beta)$ and $V_{ij}(\beta)$ are as given in Table I.

Table I
(R=0; m₁ = m_j)

Modes	U _{1j} (β)	V _{1j} (β)
TE ₁ TE _j	$-(\gamma_{m_1} - 1) \cos(m_1 \phi_p)$	$(2/\gamma_{m_1}) \cos(m_1 \phi_p)$
TM ₁ TM _j	$-(2/\gamma_{m_1}) \cos(m_1 \phi_p)$	0
TE ₁ TM _j	$(\gamma_{m_1} - 1) \sin(m_1 \phi_p)$	0
TM ₁ TE _j	$(\gamma_{m_1} - 1) \sin(m_1 \phi_p)$	0

Defining the following terms:

$$R = \sqrt{(x_j - x_1)^2 + (y_j - y_1)^2} \quad (16)$$

$$\phi = \arctan \left(\frac{y_j - y_1}{x_j - x_1} \right) - \phi_1 \quad (17)$$

$$B_{\pm}(\beta) = J_{m_j \pm m_1}(k_0 \beta R) \quad (18)$$

$$C_{\pm} = \cos((m_j \pm m_1)\phi - m_j \phi_p) \quad (19)$$

$$S_{\pm} = \sin((m_j \pm m_1)\phi - m_j \phi_p) \quad (20)$$

U_{1j}(β) and V_{1j}(β) for R>0 are given in Table II.

Table II
(R > 0)

Modes	$U_{ij}(\beta)$	$V_{ij}(\beta)$
TE_i, TE_j	$(-1)^{m_j} (C_+ B_+(\beta) - (-1)^{m_i} C_- B_-(\beta))$	$(-1)^{m_j} (C_+ B_+(\beta) + (-1)^{m_i} C_- B_-(\beta))$
TM_i, TM_j	$-(-1)^{m_j} (C_+ B_+(\beta) + (-1)^{m_i} C_- B_-(\beta))$	0
TE_i, TM_j	$(-1)^{m_j} (S_+ B_+(\beta) - (-1)^{m_i} S_- B_-(\beta))$	0
TM_i, TE_j	$(-1)^{m_j} (S_+ B_+(\beta) - (-1)^{m_i} S_- B_-(\beta))$	0

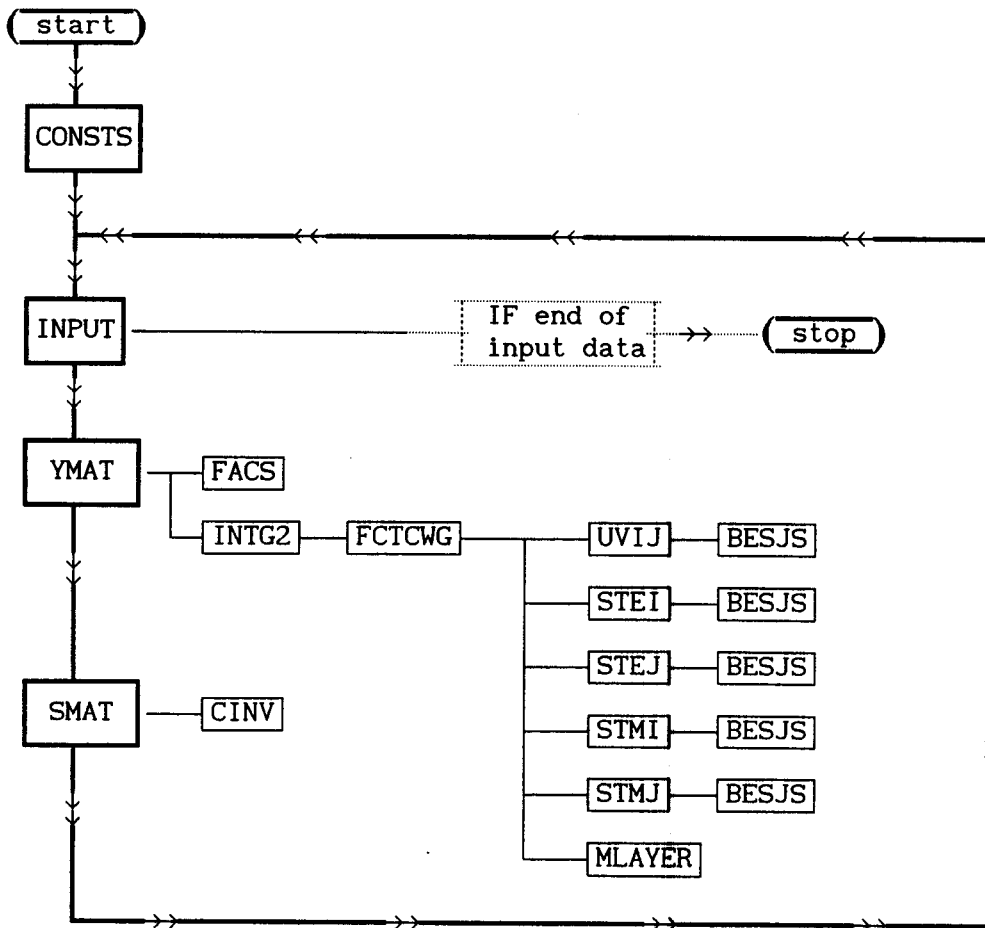
References

1. M. C. Bailey: "Analysis of Finite-size Phased Arrays of Circular Waveguide Elements", NASA Technical Report, R-408, April 1974.

Appendix
Computer Program

The computer code is written in FORTRAN IV and uses double precision variables throughout, except for integers. It was found that double precision was necessary in order to obtain sufficient accuracy on a VAX computer. The input data is read from logical unit 5. Results are output to logical unit 6. A flow chart indicating the subroutine calls is given and also a description of the input parameters with sample input and output data. The computer code has been verified against results in reference 1.

Subroutine Flowchart



Definition of Input Data

NHOLE Number of apertures in array.
NMODE Number of modes per aperture.
NUMTE Number of TE-modes per aperture.
NUMTM Number of TM-modes per aperture.
 (NOTE: same modes assumed for all apertures).

MIJ(I) First index of I-th TE mode.
NIJ(I) Second index of I-th TE mode.

MIJP(I) First index of I-th TM mode.
NIJP(I) Second index of I-th TM mode.

AIJ(I) Radius of I-th aperture.
XIJ(I) x-coordinate of center of I-th aperture.
YIJ(I) y-coordinate of center of I-th aperture.
PHIJP(I) Polarization of I-th aperture (degrees CCW).

F Frequency (Hertz).
FCON Factor for converting input to centimeters.
ER Dielectric constant of waveguide interior.

CEP Relative epsilon outside of layered region.
CUP Relative mu outside of layered region.

NLAY Number of homogeneous dielectric layers.

D(I) Thickness of I-th homogeneous layer.
CE(I) Relative epsilon of I-th layer.
CU(I) Relative mu of I-th layer.

Input Data

- (1) Read NHOLE, NMODE, NUMTE, NUMTM
Check for end-of-file on unit 5.
- (2) If NUMTE=0, skip to (3).
Read MIJ(I), NIJ(I) ; I=1 To NUMTE
- (3) If NUMTM=0, skip to (4).
Read MIJP(I), NIJP(I) ; I=1 To NUMTM
- (4) Read AIJ(I), XIJ(I), YIJ(I), PHIJP(I) ; I=1 To NHOLE
- (5) Read F, FCON, ER
- (6) Read CEP, CUP
- (7) Read NLAY
- (8) If NLAY=0, skip to (9).
Read D(I), CE(I), CU(I) ; I=1 To NLAY
- (9) Perform calculations.
Read next set of input data (1).

Sample Input Data

2 1 1 0
11
0.75 0.0 0.0 0.0
0.75 0.0 2.5 0.0
6.0D09 2.54 1.0
(1.0,0.0) (1.0,0.0)
1
0.18 (2.6,-0.0156) (1.0,0.0)

Sample Output Data

MUTUAL COUPLING OF CIRCULAR APERTURES
RADIATING INTO MULTI-LAYERS
NUMBER OF LAYERS = 1

LAYER	THICKNESS	COMPLEX EPSILON	COMPLEX MU
1	0.1800	(2.60000, -0.01560)	(1.00, 0.00)

OUTSIDE HALF-SPACE EPSILON =	1.00000	0.00000
OUTSIDE HALF-SPACE MU =	1.00000	0.00000

**** INPUT DIMENSIONS IN INCHES ****
**** FREQUENCY = .600000E+10 HERTZ ****

***** APERTURE ARRAY GEOMETRY *****

HOLE	RADIUS	X	Y	POL.
1	0.7500	0.000	0.000	0.000
2	0.7500	0.000	2.500	0.000

MODE 1 = TE-11

CY(1, 1)=(0.3415E-02, 0.1691E-02) BB= 17.00 IH= 1 IM= 1 JH= 1 JM= 1
CY(1, 2)=(0.3443E-04,-0.3158E-03) BB= 15.00 IH= 1 IM= 1 JH= 2 JM= 1

YMN(1)=(0.1695E-02, 0.0000E+00) IHOLE= 1 IMODE= 1
YMN(2)=(0.1695E-02, 0.0000E+00) IHOLE= 2 IMODE= 1

SCATTERING MATRIX

S(1, 1)	(-0.4036E+00, -0.1964E+00)	-6.9570 DB	-154.0525 DEG
S(1, 2)	(0.1871E-01, 0.3199E-01)	-28.6224 DB	59.6795 DEG
S(2, 1)	(0.1871E-01, 0.3199E-01)	-28.6224 DB	59.6795 DEG
S(2, 2)	(-0.4036E+00, -0.1964E+00)	-6.9570 DB	-154.0525 DEG

```

C***** CWG *****
C Mutual admittance and scattering matrix calculation for *
C circular waveguide array with TE-MN and TM-MN modes. *
C NOTE: Field equations of Marcuvitz show *
C TM modes polarized 90 degrees to TE modes. *
C program modified to rotate polarization of TM modes *
C by PI/2 such that TE-11 and TM-11 are polarized the same.*
C *
C Program can handle multiple homogeneous layers over apertures. *
C (INPUT on unit #5) (OUTPUT on unit #6) *
C *
C PROGRAM: M.C. Bailey *
C Hampton, VA (1989) *
C*****
102 FORMAT(1X,79(1H+))
WRITE(6,102)
CALL CONSTS
100 WRITE(6,102)
CALL INPUT
CALL YMAT
CALL SMAT
GO TO 100
END

```

SUBROUTINE CONSTS

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

IMPLICIT COMPLEX*16 (C)

COMMON /ZEROS/ XMNP(9,6),XMN(9,6)

COMMON /CONST/ CJ,PI,TWOPI,DTOR,RTOD,BIG,SMALL,BMAX,YO,MAX,LAYM

DATA XMNP(1,1),XMNP(1,2),XMNP(1,3)/3.831706,7.0155867,10.173468/

DATA XMNP(2,1),XMNP(2,2),XMNP(2,3)/1.84118,5.33144,8.53632/

DATA XMNP(3,1),XMNP(3,2),XMNP(3,3)/3.05424,6.70613,9.96947/

DATA XMNP(4,1),XMNP(4,2),XMNP(4,3)/4.20119,8.01524,11.34592/

DATA XMNP(5,1),XMNP(5,2),XMNP(5,3)/5.31755,9.28240,12.68191/

DATA XMNP(6,1),XMNP(6,2),XMNP(6,3)/6.41562,10.51986,13.98719/

DATA XMNP(7,1),XMNP(7,2),XMNP(7,3)/7.50127,11.73494,15.26818/

DATA XMNP(8,1),XMNP(8,2),XMNP(8,3)/8.57784,12.93239,16.52937/

DATA XMNP(9,1),XMNP(9,2),XMNP(9,3)/9.64742,14.11552,17.77401/

DATA XMNP(1,4),XMNP(1,5),XMNP(1,6)/13.323692,16.47063,19.615859/

DATA XMNP(2,4),XMNP(2,5),XMNP(2,6)/11.70600,14.86359,18.01553/

DATA XMNP(3,4),XMNP(3,5),XMNP(3,6)/13.17037,16.34752,19.51291/

DATA XMNP(4,4),XMNP(4,5),XMNP(4,6)/14.58585,17.78875,20.97248/

DATA XMNP(5,4),XMNP(5,5),XMNP(5,6)/15.96711,19.19603,22.40103/

DATA XMNP(6,4),XMNP(6,5),XMNP(6,6)/17.31284,20.57551,23.80358/

DATA XMNP(7,4),XMNP(7,5),XMNP(7,6)/18.63744,21.93172,25.18393/

DATA XMNP(8,4),XMNP(8,5),XMNP(8,6)/19.94185,23.26805,26.54503/

DATA XMNP(9,4),XMNP(9,5),XMNP(9,6)/21.22906,24.58720,27.88927/

DATA XMN(1,1),XMN(1,2),XMN(1,3)/2.4048256,5.5200781,8.6537279/

DATA XMN(2,1),XMN(2,2),XMN(2,3)/3.8317060,7.0155867,10.1734681/

DATA XMN(3,1),XMN(3,2),XMN(3,3)/5.1356223,8.4172441,11.6198412/

DATA XMN(4,1),XMN(4,2),XMN(4,3)/6.3801619,9.7610231,13.0152007/

DATA XMN(5,1),XMN(5,2),XMN(5,3)/7.5883427,11.0647095,14.3725367/

DATA XMN(6,1),XMN(6,2),XMN(6,3)/8.7714838,12.3386042,15.7001741/

DATA XMN(7,1),XMN(7,2),XMN(7,3)/9.93611,13.58929,17.00382/

DATA XMN(8,1),XMN(8,2),XMN(8,3)/11.08637,14.82127,18.28758/

DATA XMN(9,1),XMN(9,2),XMN(9,3)/12.22509,16.03777,19.55454/

DATA XMN(1,4),XMN(1,5),XMN(1,6)/11.7915344,14.9309177,18.071064/

DATA XMN(2,4),XMN(2,5),XMN(2,6)/13.32369,16.47063,19.61586/

DATA XMN(3,4),XMN(3,5),XMN(3,6)/14.79595,17.95982,21.11700/

DATA XMN(4,4),XMN(4,5),XMN(4,6)/16.22347,19.40942,22.58273/

DATA XMN(5,4),XMN(5,5),XMN(5,6)/17.61597,20.82693,24.01902/

DATA XMN(6,4),XMN(6,5),XMN(6,6)/18.98013,22.21780,25.43034/

DATA XMN(7,4),XMN(7,5),XMN(7,6)/20.32079,23.58608,26.82015/

DATA XMN(8,4),XMN(8,5),XMN(8,6)/21.64154,24.93493,28.19119/

DATA XMN(9,4),XMN(9,5),XMN(9,6)/22.94517,26.26681,29.54566/

DATA CJ,BIG,SMALL/(0.0D0,1.0D0),1.0D+38,1.0D-38/

DATA BMAX,MAX,LAYM/200.0,40,600/

PI=2.0*DASIN(1.0D0)

YO=1.0/(120.0*PI)

DTOR=PI/180.0

RTOD=180.0/PI

TWOPI=2.0*PI

RETURN

END

SUBROUTINE INPUT

```

C.....
c: Reads input data
C.....
    IMPLICIT REAL*8 (A,B,D-H,O-Z)
    IMPLICIT COMPLEX*16 (C)
    LOGICAL LPOL, LSIZE, LSAME, LCON, LPLAZ
    COMMON DIA, MMM, NHOLE, NMODE, NUMTE, NUMTM, F, FCON, ER, LPOL, LSIZE, LSAME
    COMMON /LAYER/ CEP, CUP, NLAY, D(601), CE(601), CU(601), CEUP, LCON, LPLAZ
    COMMON /CONST/ CJ, PI, TWOPI, DTOR, RTOD, BIG, SMALL, BMAX, YO, MAX, LAYM
    COMMON /ARRYS/ MIJ(20), NIJ(20), MIJP(20), NIJP(20),
    A          CY(40,40), CA(40,40), CB(40,40), IPIV(40), INDX(40,2),
    B          AIJ(40), XIJ(40), YIJ(40), PHIJP(40)
101  FORMAT(1X)
102  FORMAT(1X,79(1H+))
C***** INPUT DATA *****
C*****
C    NHOLE = Number of apertures in array.
C    NMODE = Number of modes per aperture.
C    NUMTE = Number of TE-modes per aperture.
C    NUMTM = Number of TM-modes per aperture.
C          NOTE: (NUMTE+NUMTM)=NMODE
C          The same modes are assumed in all apertures.
C*****
    READ(5,*,END=9999)NHOLE,NMODE,NUMTE,NUMTM
    MMM=NHOLE*NMODE
    IF(MMM.LE.MAX) GO TO 120
    WRITE(6,110)
110  FORMAT(1X'(NHOLE*NMODE) EXCEEDS DIMENSION OF CY')
    STOP
120  WRITE(6,130)
130  FORMAT(1X'MUTUAL COUPLING OF CIRCULAR APERTURES')
    IF(NUMTE.GT.20) GO TO 500
    IF(NUMTM.GT.20) GO TO 500
    IF(NUMTE.GT.NMODE) GO TO 500
    IF(NUMTM.GT.NMODE) GO TO 500
    IF(NMODE.GT.40) GO TO 500
    IF((NUMTE+NUMTM).NE.NMODE) GO TO 500
    IF(NUMTE.EQ.0) GO TO 150
C*****
C    MIJ(I),NIJ(I) = M,N indices of I-th TE mode.
C*****
    READ(5,140) ((MIJ(I),NIJ(I)),I=1,NUMTE)
140  FORMAT(20(2I1,1X))
150  IF(NUMTM.EQ.0) GO TO 170
C*****
C    MIJP(I),NIJP(I) = M,N indices of I-th TM mode.
C*****
    READ(5,160) ((MIJP(I),NIJP(I)),I=1,NUMTM)
160  FORMAT(20(2I1,1X))

```

```

170 CONTINUE
C*****
C      AIJ(I)  = Radius of I-th aperture.                *
C      XIJ(I)  = X-coordinate of center of I-th aperutre. *
C      YIJ(I)  = Y-coordinate of center of I-th aperture. *
C      PHIJP(I) = Angular rotation of XI-axis WRT X-axis (degrees CCW). *
C+++++
      READ(5,*) ((AIJ(I), XIJ(I), YIJ(I), PHIJP(I)), I=1, NHOLE)
C*****
C      F      = Frequency (Hertz).                        *
C      FCON   = Factor for converting input dimensions to centimeters. *
C              (2.54 for input in inches) (1.0 for input in centimeters). *
C      ER     = Relative dielectric constant of material filling waveguides *
C.....
C              For input dimensions in wavelengths set: *
C              F=3.0E10 *
C              FCON=1.0 *
C+++++
      READ(5,*)F, FCON, ER
C*****
C      CEP = Complex (relative) Epsilon of half-space outside of layers. *
C      CUP = Complex (relative) Mu of half-space outside of layers. *
C.....
C      Input format is: (A, -B) (C, -D) *
C      For free-space outside: (1.0, 0.0) (1.0, 0.0) *
C      For a metal sheet approximation: (-1.0E10, 0.0) (1.0, 0.0) *
C              or CABS(CEP*CUP) greater than 1.0E09 *
C+++++
      READ(5,*)CEP, CUP
C*****
C      NLAY = Number of homogeneous layers over apertures. *
C.....
C              NLAY=0 (half-space). *
C+++++
      READ(5,*)NLAY
      IF(NLAY.GT. LAYM)STOP 4444
      IF(NLAY.EQ.0) GO TO 210
C*****
C      NOTE: If NLAY=0, do not inter the following data. *
C.....
C      D(I) = Thickness of I-th layer. *
C      CE(I) = Complex Epsilon of I-th layer (relative to free space). *
C      CU(I) = Complex Mu of I-th layer (relative to free space). *
C.....
C      NOTE: Values of CE(I) must be input with format: (A, -B) *
C              Values of CU(I) must be input with format: (C, -D) *
C              where dielectric loss tangent = B/A *
C.....

```

```

C      CAUTION:
C      An input value of 0.0 for B could produce erroneous results.
C      A value of 0.0001 for the dielectric loss tangent should yield
C      results which approximate a lossless dielectric in most cases.
C      (Suggest varying B to determine sensitivity to small values)
C+++++
DO 180 I=1,NLAY
  READ(5,*)D(I),CE(I),CU(I)
180  CONTINUE
  WRITE(6,190)NLAY
190  FORMAT(1X'RADIATING INTO MULTI-LAYERS'/1X'NUMBER OF LAYERS = 'I4/)
  WRITE(6,200)
200  FORMAT(1X'LAYER'3X'THICKNESS'5X'COMPLEX EPSILON'10X'COMPLEX MU')
210  IF(DABS(ER).LT.1.0D-04)ER=1.0
  IF(DABS(FCON).LT.1.0D-04)FCON=1.0
  XLAM=(30./FCON)/(F*1.0D-09)
  IF(DABS(XLAM-1.0).LT.0.0001)XLAM=1.0
  OMEGA=TWOPI*F
  IF(NLAY.EQ.0)GO TO 280
  DO 270 I=1,NLAY
  WRITE(6,220)I,D(I),CE(I),CU(I)
220  FORMAT(1X'I4,2XF9.4,3X('F10.5','F9.5')'3X('F6.2','F6.2'))
  D(I)=D(I)/XLAM
  EPR=DREAL(CE(I))
  EPI=DIMAG(CE(I))
  IF(EPI.LE.-0.0001*EPR)GO TO 270
  IF(EPI.LT.0.0)GO TO 240
  EPI=-EPI
  WRITE(6,230)I
230  FORMAT(1X'NOTE: The imaginary part of the dielectric constant'/
  A8X'for the layered region must be negative.'/
  B8X'This has been corrected for layer 'I2/)
240  IF(EPI.LE.-0.0001*EPR)GO TO 260
  TANL=DABS(EPI/EPR)
  WRITE(6,250)I,TANL
250  FORMAT(1X'CAUTION: The dielectric loss tangent for the layered'/
  A11X'region must not be zero. Erroneous results could also be'/
  B11X'obtained for very small values of dielectric loss.'/
  C11X'The loss tangent for layer 'I2' is 'E11.4/)
260  CE(I)=DCMPLX(EPR,EPI)
270  CONTINUE
  WRITE(6,101)
280  CONTINUE
290  LCON=.FALSE.
  CEUP=CEP*CUP
  IF(CDABS(CEUP).GE.1.0D+9) LCON=.TRUE.
  ERC=DREAL(CEP)
  EIC=DIMAG(CEP)
  NSIGN=1
  IF(ERC.LT.0.0)NSIGN=-1

```

```

    ERCP=NSIGN*ERC
    IF(ERCP.GT.1.0D+10)ERCP=1.0D+10
    ERC=NSIGN*ERCP
    CEP=DCMLX(ERC,EIC)
    IF(NLAY.EQ.0.AND.(.NOT.LCON).AND.(.NOT.LPLAZ))WRITE(6,300)
300  FORMAT(1X'RADIATING INTO HALF-SPACE'/)
    IF(LCON)WRITE(6,320)
310  CONTINUE
320  FORMAT(1X'**** PERFECT CONDUCTOR OUTSIDE ****'/)
    IF(.NOT.LCON)WRITE(6,330)CEP,CUP
330  FORMAT(1X'OUTSIDE HALF-SPACE EPSILON = '2F18.5/
+      1X'OUTSIDE HALF-SPACE MU      = '2F18.5/')
    IF(FCON.EQ.2.54)WRITE(6,340)
340  FORMAT(1X'**** INPUT DIMENSIONS IN INCHES ****')
    IF((FCON.EQ.1.0).AND.(F.EQ.3.0E10))WRITE(6,350)
350  FORMAT(1X'**** INPUT DIMENSIONS IN WAVELENGTHS ****')
    IF((FCON.EQ.1.0).AND.(F.NE.3.0E10))WRITE(6,360)
360  FORMAT(1X'**** INPUT DIMENSIONS IN CENTIMETERS ****')
    IF((FCON.NE.1.0).AND.(FCON.NE.2.54))WRITE(6,370)FCON
370  FORMAT(1X'**** INPUT CONVERSION FACTOR = 'F10.5' ****')
    IF((FCON.NE.1.0).OR.(F.NE.3.0E10))WRITE(6,380)F
380  FORMAT(1X'**** FREQUENCY = 'E11.6' HERTZ ****')
    WRITE(6,390)
390  FORMAT(/1X'+++++++ APERTURE ARRAY GEOMETRY +++++++'/
A1X'HOLE',3X'RADIUS',6X,'X',8X,'Y',7X,'POL. ')
    LPOL=.TRUE.
    LSIZE=.TRUE.
    DIA=2.0*AIJ(1)
    DO 410 I=1,NHOLE
    WRITE(6,400)I,AIJ(I),XIJ(I),YIJ(I),PHIJP(I)
400  FORMAT(1X,I3,1X,F9.4,1X,F8.3,1X,F8.3,1X,F8.3)
    AIJ(I)=AIJ(I)/XLAM
    XIJ(I)=XIJ(I)/XLAM
    YIJ(I)=YIJ(I)/XLAM
    PHIJP(I)=PHIJP(I)*DTOR
    IF(DABS(AIJ(I)-AIJ(1)).GT.0.0001)LSIZE=.FALSE.
    IF(DABS(PHIJP(I)-PHIJP(1)).GT.0.0001)LPOL=.FALSE.
410  CONTINUE
    LSAME=.FALSE.
    IF(LSIZE.AND.LPOL)LSAME=.TRUE.
    WRITE(6,101)
    IF(NUMTE.EQ.0)GO TO 440
    DO 430 I=1,NUMTE
    WRITE(6,420)I,MIJ(I),NIJ(I)
420  FORMAT(1X'MODE' I2' = TE-' I1, I1)
    IF((1+MIJ(I)).GT.9)GO TO 480
    IF(NIJ(I).GT.6)GO TO 480
    IF(MIJ(I).LT.0)GO TO 480
    IF(NIJ(I).LT.1)GO TO 480
430  CONTINUE

```

```

440  IF(NUMTM.EQ.0) GO TO 470
      DO 460 I=1, NUMTM
        IP=I+NUMTE
        WRITE(6, 450) IP, MIJP(I), NIJP(I)
450  FORMAT(1X'MODE' I2' = TM-' I1, I1)
        IF((1+MIJP(I)).GT.9) GO TO 480
        IF(NIJP(I).GT.6) GO TO 480
        IF(MIJP(I).LT.0) GO TO 480
        IF(NIJP(I).LT.1) GO TO 480
460  CONTINUE
470  WRITE(6, 101)
      RETURN
480  WRITE(6, 490)
490  FORMAT(1X'RANGE OF MODE INDICES EXCEEDED' /
A1X' (M,N) VALUES ALLOWED: (0,1) THROUGH (8,6)')
      STOP
500  WRITE(6, 510) NHOLE, NMODE, NUMTE, NUMTM
510  FORMAT(1X'NUMBER OF MODES EXCEEDS ARRAY DIMENSIONS' //
A1X' NHOLE=' I3, 5X' NMODE=' I3, 5X' NUMTE=' I3, 5X' NUMTM=' I3)
      STOP
9999 WRITE(6, 102)
      STOP
      END

```

SUBROUTINE YMAT

```

C.....
c:  Calculates coefficients CY(II,JJ) of complex admittance matrix [Y] :
C.....
      IMPLICIT REAL*8 (A,B,D-H,O-Z)
      IMPLICIT COMPLEX*16 (C)
      LOGICAL TEI, TEJ, TMI, TMJ, LPOL, LSIZE, LSAME, LCON, LPLAZ
      DIMENSION Y(2), YA(2)
      COMMON DIA, MMM, NHOLE, NMODE, NUMTE, NUMTM, F, FCON, ER, LPOL, LSIZE, LSAME
      COMMON /APERS/ IT, MI, NI, MJ, NJ, AI, AJ, TEI, TEJ, TMI, TMJ, PHI, PHIP, R
      COMMON /CONST/ CJ, PI, TWOPI, DTOR, RTOD, BIG, SMALL, BMAX, YO, MAX, LAYM
      COMMON /LAYER/ CEP, CUP, NLAY, D(601), CE(601), CU(601), CEUP, LCON, LPLAZ
      COMMON /ARRYS/ MIJ(20), NIJ(20), MIJP(20), NIJP(20),
A          CY(40, 40), CA(40, 40), CB(40, 40), IPIV(40), INDX(40, 2),
B          AIJ(40), XIJ(40), YIJ(40), PHIJP(40)
      EXTERNAL FCTCWG
101  FORMAT(1X)
      DO 1000 IHOLE=1, NHOLE
      DO 1000 JHOLE=1, NHOLE
      DO 1000 IMODE=1, NMODE
      DO 1000 JMODE=1, NMODE
      II=(IHOLE-1)*NMODE+IMODE
      JJ=(JHOLE-1)*NMODE+JMODE
      IF(II.LE.JJ) GO TO 100
      CY(II, JJ)=CY(JJ, II)
      GO TO 1000
100  CONTINUE
      IF(IHOLE.EQ.1) GO TO 110
      IF(IHOLE.NE.JHOLE) GO TO 110
      IF(.NOT.LSIZE) GO TO 110
      CY(II, JJ)=CY(IMODE, JMODE)
      GO TO 1000
110  TEI=. TRUE.
      TEJ=. TRUE.
      TMI=. FALSE.
      TMJ=. FALSE.
      IF(IMODE.LE.NUMTE) GO TO 120
      TEI=. FALSE.
      TMI=. TRUE.
120  IF(JMODE.LE.NUMTE) GO TO 130
      TEJ=. FALSE.
      TMJ=. TRUE.
130  CONTINUE
      IT=0
      IF(TEI.AND.TEJ) IT=1
      IF(TMI.AND.TMJ) IT=2
      IF(TEI.AND.TMJ) IT=3
      IF(TMI.AND.TEJ) IT=4
      AI=AIJ(IHOLE)
      AJ=AIJ(JHOLE)

```



```

PHPI=PHIJP(IHOLE)
PHPJ=PHIJP(JHOLE)
IF(TMI)PHPI=PHPI-0.5*PI
IF(TMJ)PHPJ=PHPJ-0.5*PI
PHIP=PHPJ-PHPI
XJI=XIJ(JHOLE)-XIJ(IHOLE)
YJI=YIJ(JHOLE)-YIJ(IHOLE)
R=DSQRT(XJI*XJI+YJI*YJI)
IF(IHOLE.EQ.JHOLE)R=0.0
IF(DABS(R).LT.1.0D-20)R=0.0
IF(DABS(PHIP).LT.1.0D-04)PHIP=0.0
PHI=0.0
IF(R.LT.1.0D-04)GO TO 160
IF(DABS(XJI).LT.1.0D-04)GO TO 140
PHI=DATAN2(YJI,XJI)-PHPI
GO TO 150
140 PHI=0.5*PI
IF(YJI.LT.0.0)PHI=PHI+PI
PHI=PHI-PHPI
150 CONTINUE
160 IF(TMI)GO TO 170
IF(.NOT.TEI)STOP 1111
MI=MIJ(IMODE)
NI=NIJ(IMODE)
GO TO 180
170 IDEM=IMODE-NUMTE
MI=MIJP(IDEM)
NI=NIJP(IDEM)
180 IF(TMJ)GO TO 190
IF(.NOT.TEJ)STOP 2222
MJ=MIJ(JMODE)
NJ=NIJ(JMODE)
GO TO 200
190 JDEM=JMODE-NUMTE
MJ=MIJP(JDEM)
NJ=NIJP(JDEM)
200 CONTINUE
IF(IHOLE.NE.JHOLE)GO TO 210
IF(MJ.EQ.MI)GO TO 210
CY(II,JJ)=(0.0,0.0)
GO TO 1000
210 CONTINUE
IF(IHOLE.EQ.JHOLE)GO TO 230
IF(IHOLE.EQ.1)GO TO 230
IF(.NOT.LSIZE)GO TO 230
IF(.NOT.LPOL)GO TO 230
IHM1=IHOLE-1
JHM1=JHOLE-1
DO 220 IKH=1,IHM1
DO 220 JKH=1,JHM1

```

```

IF(DABS(AI-AIJ(IKH)).GT.0.001) GO TO 220
IF(DABS(AJ-AIJ(JKH)).GT.0.001) GO TO 220
TX=DABS(DABS(XJI)-DABS(XIJ(IKH)-XIJ(JKH)))
IF(TX.GT.0.001) GO TO 220
TY=DABS(DABS(YJI)-DABS(YIJ(IKH)-YIJ(JKH)))
IF(TY.GT.0.001) GO TO 220
PT=DABS(DABS(PHIP)-DABS(PHIJP(IKH)-PHIJP(JKH)))
IF(PT.GT.0.001) GO TO 220
IK=(IKH-1)*NMODE+IMODE
JK=(JKH-1)*NMODE+JMODE
CY(II, JJ)=CY(IK, JK)
GO TO 1000
220 CONTINUE
230 CONTINUE
EMI=2.0
EMJ=2.0
IF(MI.EQ.0)EMI=1.0
IF(MJ.EQ.0)EMJ=1.0
CALL FACS(MI, NI, MJ, NJ, AI, AJ, R, PHI, PHIP, EMI)
YTEST=0.0001
Y(1)=0.0
Y(2)=0.0
NDEL=50
BDEL=0.25
AA=0.0001
BB=BDEL
TOL=0.0001
240 CALL INTG2(AA, BB, FCTCWG, NDEL, TOL, YA, Y)
Y(1)=Y(1)+YA(1)
Y(2)=Y(2)+YA(2)
IF(NLAY.EQ.0) GO TO 270
DO 250 N=1, NLAY
EUREAL=DREAL(CE(N)*CU(N))
IF(EUREAL.LT.0.0)EUREAL=-EUREAL
IF((BB*BB).LE.EUREAL) GO TO 260
250 CONTINUE
GO TO 270
260 CONTINUE
IF(DABS(BB-1.0).LT.0.01)BB=1.0
AA=BB
BB=BB+BDEL
IF(DABS(AA-1.0).LT.0.01)AA=1.0001
IF(DABS(BB-1.0).LT.0.01)BB=0.9999
GO TO 240
270 IF(BB.LT.2.0) GO TO 260
BDEL=1.0
NBDEL=0
AA=BB
BB=BB+BDEL
280 CALL INTG2(AA, BB, FCTCWG, NDEL, TOL, YA, Y)

```

```

Y(1)=Y(1)+YA(1)
Y(2)=Y(2)+YA(2)
IF(DABS(Y(1)).LE.YTEST) GO TO 290
IF(DABS(YA(1)/Y(1)).GT.YTEST) GO TO 310
290 IF(DABS(Y(2)).LE.YTEST) GO TO 300
IF(DABS(YA(2)/Y(2)).GT.YTEST) GO TO 310
300 NBDEL=NBDEL+1
IF(NBDEL.GE.5) GO TO 330
GO TO 320
310 NBDEL=0
320 IF(BB.GE.BMAX) GO TO 330
IF(BB.GE.20.0)BDEL=5.0
AA=BB
BB=BB+BDEL
GO TO 280
330 CYIJ=-Y0*DSQRT(EMI*EMJ)*DCMPLX(Y(1),Y(2))
CY(II, JJ)=CYIJ
WRITE(6, 340) II, JJ, CY(II, JJ), BB, IHOLE, IMODE, JHOLE, JMODE
340 FORMAT(1X' CY(' I2', ' I2')=( ' E11. 4', ' E11. 4' )' 3X' BB=' F6. 2,
A4X' IH=' I2, 2X' IM=' I2, 3X' JH=' I2, 2X' JM=' I2)
1000 CONTINUE
WRITE(6, 101)
RETURN
END

```

SUBROUTINE SMAT

C.....
 c: Calculates coefficients of complex scattering matrix [S] (see eq.1):
 C.....

```

    IMPLICIT REAL*8 (A,B,D-H,O-Z)
    IMPLICIT COMPLEX*16 (C)
    LOGICAL TEI, TEJ, TMI, TMJ
    LOGICAL LPOL, LSIZE, LSAME
    COMMON DIA, MMM, NHOLE, NMODE, NUMTE, NUMTM, F, FCON, ER, LPOL, LSIZE, LSAME
    COMMON /APERS/ IT, MI, NI, MJ, NJ, AI, AJ, TEI, TEJ, TMI, TMJ, PHI, PHIP, R
    COMMON /ZEROS/ XMNP(9,6), XMN(9,6)
    COMMON /CONST/ CJ, PI, TWOPI, DTOR, RTOD, BIG, SMALL, BMAX, YO, MAX, LAYM
    COMMON /ARRYS/ MIJ(20), NIJ(20), MIJP(20), NIJP(20),
  A          CY(40,40), CA(40,40), CB(40,40), IPIV(40), INDX(40,2),
  B          AIJ(40), XIJ(40), YIJ(40), PHIJP(40)
    DO 130 IHOLE=1, NHOLE
    DO 130 JHOLE=1, NHOLE
    DO 130 IMODE=1, NMODE
    DO 130 JMODE=1, NMODE
    TMI=.FALSE.
    IF(IMODE.GT.NUMTE)TMI=.TRUE.
    II=(IHOLE-1)*NMODE+IMODE
    JJ=(JHOLE-1)*NMODE+JMODE
    AKOI=AIJ(IHOLE)*TWOPI
    IF(TMI)GO TO 100
    NI=NIJ(IMODE)
    MIP=MIJ(IMODE)+1
    FSQI=(XMNP(MIP,NI)/AKOI)**2
    FTSQ=ER-FSQI
    FCON=DABS(FTSQ)
    IF(FCON.LT.SMALL)FCON=0.0
    IF(FCON.GT.BIG)FCON=BIG
    IF(FTSQ.GE.0.0)CYMNO=YO*DSQRT(FCON)
    IF(FTSQ.LT.0.0)CYMNO=-CJ*YO*DSQRT(FCON)
    GO TO 110
  100  IDEM=IMODE-NUMTE
    NI=NIJP(IDEM)
    MIP=MIJP(IDEM)+1
    FSQI=(XMN(MIP,NI)/AKOI)**2
    FTSQ=ER-FSQI
    FCON=DABS(FTSQ)
    IF(FCON.LT.SMALL)FCON=SMALL
    IF(FCON.GT.BIG)FCON=BIG
    IF(FTSQ.GE.0.0)CYMNO=YO/DSQRT(FCON)
    IF(FTSQ.LT.0.0)CYMNO=YO/(-CJ*DSQRT(FCON))
  110  CONTINUE
    CA(II, JJ)=CY(II, JJ)
    CB(II, JJ)=-CY(II, JJ)
    IF(II.EQ.JJ)CA(II, JJ)=CYMNO+CY(II, JJ)
    IF(II.EQ.JJ)CB(II, JJ)=CYMNO-CY(II, JJ)
  
```

```

CY(II, JJ)=CB(II, JJ)
IF(II.EQ. JJ)WRITE(6, 120)II, CYMNO, IHOLE, IMODE
120  FORMAT(1X' YMN(' I2' )=(' E11.4' , ' E11.4' )' 3X' IHOLE=' I2, 3X' IMODE=' I2)
130  CONTINUE
WRITE(6, 140)
140  FORMAT(/10X' SCATTERING MATRIX' )
CALL CINV(CA, MMM, IPIV, INDX, MAX)
DO 160 I=1, MMM
DO 160 J=1, MMM
CB(I, J)=(0.0, 0.0)
DO 150 K=1, MMM
CB(I, J)=CB(I, J)+CY(I, K)*CA(K, J)
150  CONTINUE
160  CONTINUE
DO 200 I=1, MMM
DO 200 J=1, MMM
XISOL=-300.0
XDX=CDABS(CB(I, J))
IF(XDX.LT. 1.0D-15)GO TO 180
XISOL=20. *DLOG10(XDX)
XX1=DIMAG(CB(I, J))
XX2=DREAL(CB(I, J))
PHASE=RTOD*DATAN2(XX1, XX2)
WRITE(6, 170)I, J, CB(I, J), XISOL, PHASE
170  FORMAT(1X'S(' I2' , ' I2' )=(' E11.4' , ' E11.4' )' 3XF9.4' DB' 3XF9.4' DEG' )
GO TO 200
180  WRITE(6, 190)I, J, CB(I, J)
190  FORMAT(1X'S(' I2' , ' I2' )=(' E11.4' , ' E11.4' )' 3X' BELOW -300 DB' )
200  CONTINUE
RETURN
END

```

SUBROUTINE CINV(CA, N, IPIV, INDX, MAX)

```

C.....
c: Performs matrix inversion of (N by N) complex matrix CA.      :
c: On return, CA contains inverted matrix. CA must be dimensioned at :
c: least N by N in calling routine. IPIV and INDX are used internally:
c: by CINV and must be dimensioned at least N and N by 2 in calling  :
c: routine. MAX is row dimension of CA and INDX in calling routine.  :
C.....
      IMPLICIT REAL*8 (A, B, D-H, O-Z)
      IMPLICIT COMPLEX*16 (C)
      DIMENSION CA(MAX, N), IPIV(N), INDX(MAX, 2)
      DATA CO, C1 /(.0D0, 0.0D0), (1.0D0, 0.0D0)/
      DATA BIG, SMALL /1.0D+38, 1.0D-38/
      ISCALE=0
      RL=BIG
      RS=SMALL
      CDET=C1
      ADM=1.0
100   DO 100 J=1, N
      IPIV(J)=0
      DO 220 I=1, N
      AVM=0.0
      DO 130 J=1, N
      IF(IPIV(J).EQ.1) GO TO 130
      DO 120 K=1, N
      IF(IPIV(K)-1)110, 120, 260
110   CONTINUE
      AVA=CDABS(CA(J, K))
      IF(AVM.GE.AVA) GO TO 120
      IROW=J
      ICOL=K
      AVM=AVA
120   CONTINUE
130   CONTINUE
      IF(AVM.EQ.0.0) GO TO 250
      IPIV(ICOL)=IPIV(ICOL)+1
      IF(IROW.EQ.ICOL) GO TO 150
      CDET=-CDET
      DO 140 L=1, N
      CSWAP=CA(IROW, L)
      CA(IROW, L)=CA(ICOL, L)
      CA(ICOL, L)=CSWAP
140   CONTINUE
150   CONTINUE
      INDX(I, 1)=IROW
      INDX(I, 2)=ICOL
      CPIV=CA(ICOL, ICOL)
      APV=CDABS(CPIV)
      IF(APV.EQ.0.0) GO TO 250
      CPIVI=CPIV

```

```

ADM=CDABS(CDET)
IF(ADM.LT.RL) GO TO 160
CDET=CDET/RL
ADM=CDABS(CDET)
ISCALE=ISCALE+1
IF(ADM.LT.RL) GO TO 170
CDET=CDET/RL
ISCALE=ISCALE+1
GO TO 170
160 CONTINUE
IF(ADM.GT.RS) GO TO 170
CDET=CDET*RL
ADM=CDABS(CDET)
ISCALE=ISCALE-1
IF(ADM.GT.RS) GO TO 170
CDET=CDET*RL
ISCALE=ISCALE-1
170 CONTINUE
APV=CDABS(CPIVI)
IF(APV.LT.RL) GO TO 180
CPIVI=CPIVI/RL
APV=CDABS(CPIVI)
ISCALE=ISCALE+1
IF(APV.LT.RL) GO TO 190
CPIVI=CPIVI/RL
ISCALE=ISCALE+1
GO TO 190
180 CONTINUE
IF(APV.GT.RS) GO TO 190
CPIVI=CPIVI*RL
APV=CDABS(CPIVI)
ISCALE=ISCALE-1
IF(APV.GT.RS) GO TO 190
CPIVI=CPIVI*RL
ISCALE=ISCALE-1
190 CONTINUE
CDET=CDET*CPIVI
CA(ICOL, ICOL)=C1
DO 200 L=1, N
200 CA(ICOL, L)=CA(ICOL, L)/CPIV
DO 220 L1=1, N
IF(L1.EQ. ICOL) GO TO 220
CSWAP=CA(L1, ICOL)
CA(L1, ICOL)=C0
DO 210 L=1, N
210 CA(L1, L)=CA(L1, L)-CA(ICOL, L)*CSWAP
220 CONTINUE
DO 240 I=1, N
L=N+1-I
IF(INDX(L, 1).EQ. INDX(L, 2)) GO TO 240

```

```
IROW=INDX(L,1)
ICOL=INDX(L,2)
DO 230 K=1,N
CSWAP=CA(K,IROW)
CA(K,IROW)=CA(K,ICOL)
CA(K,ICOL)=CSWAP
230 CONTINUE
240 CONTINUE
GO TO 260
250 CDET=CO
ISCALE=0
260 RETURN
END
```


SUBROUTINE INTG2(AA, BB, FX, NT, TOL, SUM, Y)

C:
c: Integrates 2 real functions (may be real and imaginary parts of :
c: of a complex function) over limits of AA to BB by using a Simpson :
c: integration procedure with interval halving until relative error :
c: TOL is reached on each interval. NT is number of intervals. :
c: SUM is a 2 element array of the resultant integrals over AA to BB. :
c: Y is the cumulative values of SUM from previous calls to INTG2. :
c: Y must be set to zero prior to first call to INTG2 :
c: and is used when integrating FX over wide limits :
c: by successive calls to INTG2. FX(X,WK) is user : May 1989 :
c: supplied subroutine to evaluate integrands WK. : M.C.Bailey :
c: FX must be declared EXTERNAL in calling routine. : Hampton, VA :
C:

```

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SUM(2),Y(2),WK(2),SUMA(2),SUMB(2),ESUM(2),BASE(2)
      DIMENSION A(401,2),B(401,2)
      DATA NMAX,NIMAX/400,2/
      IF(NT.GT.NMAX) STOP
      DEL=(BB-AA)/NT
      DELO2=0.5DO*DEL
      DO 100 I=1,NT+1
      X=AA+(I-1)*DEL
      CALL FX(X,WK)
      A(I,1)=WK(1)
      A(I,2)=WK(2)
100  CONTINUE
      BASE(1)=DELO2*(A(1,1)+A(NT+1,1))
      BASE(2)=DELO2*(A(1,2)+A(NT+1,2))
      DO 110 I=2,NT
      BASE(1)=BASE(1)+DEL*A(I,1)
      BASE(2)=BASE(2)+DEL*A(I,2)
110  CONTINUE
      DENOM=1.D-28
      TEST=DABS(BASE(1)+Y(1))
      IF(TEST.GT.DENOM)DENOM=TEST
      TEST=DABS(BASE(2)+Y(2))
      IF(TEST.GT.DENOM)DENOM=TEST
      SUM(1)=0.0DO
      SUM(2)=0.0DO
      DO 300 K=1,NT
      ND=1
      FF=AA+(K-1)*DEL
      X=FF+DEL
      CALL FX(X,WK)
      A(2,1)=WK(1)
      A(2,2)=WK(2)
      SUMA(1)=DELO2*(A(1,1)+A(2,1))
      SUMA(2)=DELO2*(A(1,2)+A(2,2))
200  ND=ND+ND

```

```

IF(ND.GT.NMAX) GO TO 270
NB=ND/2
NA=NB+1
NP=ND+1
DELN=DEL/ND
IF(DELN.LT.1.D-6) GO TO 270
DO 210 J=1,NB
X=FF+DELN*(J+J-1)
CALL FX(X,WK)
B(J,1)=WK(1)
B(J,2)=WK(2)
210 CONTINUE
NAP=NA+1
NBP=NB+1
DO 220 J=1,NA
JL=NAP-J
JP=JL+JL-1
A(JP,1)=A(JL,1)
A(JP,2)=A(JL,2)
220 CONTINUE
DO 230 J=1,NB
JL=NBP-J
JP=JL+JL
A(JP,1)=B(JL,1)
A(JP,2)=B(JL,2)
230 CONTINUE
SUMB(1)=0.5DO*(A(1,1)+A(NP,1))
SUMB(2)=0.5DO*(A(1,2)+A(NP,2))
DO 240 NS=2,ND
SUMB(1)=SUMB(1)+A(NS,1)
SUMB(2)=SUMB(2)+A(NS,2)
240 CONTINUE
SUMB(1)=DELN*SUMB(1)
SUMB(2)=DELN*SUMB(2)
ESUM(1)=DABS(SUMB(1)-SUMA(1))/DENOM
ESUM(2)=DABS(SUMB(2)-SUMA(2))/DENOM
SUMA(1)=SUMB(1)
SUMA(2)=SUMB(2)
IF(ESUM(1).GT.TOL) GO TO 200
IF(ESUM(2).GT.TOL) GO TO 200
GO TO 280
270 ND=ND/2
280 CONTINUE
SUM(1)=SUM(1)+SUMA(1)
SUM(2)=SUM(2)+SUMA(2)
A(1,1)=A(NP,1)
A(1,2)=A(NP,2)
300 CONTINUE
RETURN
END

```

```

SUBROUTINE FCTCWG(BETA,FI)
C.....
c: Evaluates integrand (real and imaginary parts) of equation 2.      :
C.....
  IMPLICIT REAL*8 (A,B,D-H,O-Z)
  IMPLICIT COMPLEX*16 (C)
  LOGICAL TEI, TEJ, TMI, TMJ, LCON, LPLAZ
  DIMENSION FI(2)
  COMMON /APERS/ IT, MI, NI, MJ, NJ, AI, AJ, TEI, TEJ, TMI, TMJ, PHI, PHIP, R
  COMMON /LAYER/ CEP, CUP, NLAY, D(601), CE(601), CU(601), CEUP, LCON, LPLAZ
  CALL MLAYER(BETA, CW1, CW2)
  CALL UVIJ(BETA, UIJ, VIJ, IT)
  GO TO (10, 20, 30, 40) IT
  GO TO 900
10  CALL STEI(BETA, SSI, TTI)
    IF(MI.NE.MJ) GO TO 15
    IF(NI.NE.NJ) GO TO 15
    IF(DABS(AI-AJ).GT.1.D-4) GO TO 15
    SSJ=SSI
    TTJ=TTI
    GO TO 50
15  CALL STEJ(BETA, SSJ, TTJ)
    GO TO 50
20  CALL STMI(BETA, SSI, TTI)
    IF(MI.NE.MJ) GO TO 25
    IF(NI.NE.NJ) GO TO 25
    IF(DABS(AI-AJ).GT.1.D-4) GO TO 25
    SSJ=SSI
    TTJ=TTI
    GO TO 50
25  CALL STMJ(BETA, SSJ, TTJ)
    GO TO 50
30  CALL STEI(BETA, SSI, TTI)
    CALL STMJ(BETA, SSJ, TTJ)
    GO TO 50
40  CALL STMI(BETA, SSI, TTI)
    CALL STEJ(BETA, SSJ, TTJ)
50  F1=+SSI*SSJ*UIJ
    F2=-TTI*TTJ*VIJ
    CCW=(F1*CW1+F2*CW2)*BETA
    FI(1)=DREAL(CCW)
    FI(2)=DIMAG(CCW)
    RETURN
900 WRITE(6, 901) IT
901 FORMAT(1X'### ERRORS IN FCTCWG ###'
A/1X' (IT must be between 1 and 4)'
B/1X' IT=' I5)
  STOP
  END

```

SUBROUTINE MLAYER(BETA, CW1, CW2)

```

C.....
c:  Evaluates complex functions in equations 5 and 6.
C.....
      IMPLICIT REAL*8 (A,B,D-H,O-Z)
      IMPLICIT COMPLEX*16 (C)
      LOGICAL LCON, LPLAZ
      COMMON /LAYER/ CEP, CUP, NLAY, D(601), CE(601), CU(601), CEUP, LCON, LPLAZ
      DATA XK0, BIG, CSMAL/6.283185307179586D0, 1.0D+28, (1.0D-28, 0.0D0)/
      DATA CO, C1, CJ/(0.0D0, 0.0D0), (1.0D0, 0.0D0), (0.0D0, 1.0D0)/
      CU(NLAY+1)=CUP
      CE(NLAY+1)=CEP
      BSQ=BETA*BETA
      CPF=CO
      CGPOG=CO
      CFPOF=C1
      IF(LCON) GO TO 35
      CPF=C1
      CBSQP=CEUP-BSQ
      IF(LPLAZ)CBSQP=C1-BSQ
      IF(DREAL(CBSQP))20, 10, 10
10     CKZ=XK0*CDSQRT(CBSQP)
      GO TO 30
20     CKZ=-CJ*XK0*CDSQRT(-CBSQP)
30     CFPOF=-CJ*CKZ
      CGPOG=CFPOF
35     IF(NLAY.EQ.0) GO TO 200
      N=NLAY
40     CARG=CE(N)*CU(N)-BSQ
      IF(DREAL(CARG).LT.0.0) GO TO 50
      CKZ=XK0*CDSQRT(CARG)
      GO TO 60
50     CKZ=-CJ*XK0*CDSQRT(-CARG)
60     CKZD=CKZ*D(N)
      A=DREAL(CKZD)
      B=DIMAG(CKZD)
      DSINA=DSIN(A)
      DCOSA=DCOS(A)
      NSIGN=1
      IF(B.LT.0.0)NSIGN=-1
      BP=NSIGN*B
      IF(BP.GT.1.0D-10) GO TO 70
      CSINKD=DSINA
      CCOSKD=DCOSA
      GO TO 100
70     IF(BP.LT.65) GO TO 80
      DCOSHB=0.5*(1.6948892D+28)
      DSINHB=NSIGN*DCOSHB
      GO TO 90
80     EP=DEXP(BP)

```

```

EM=1.0D0/EP
DSINHB=0.5D0*(EP-EM)
DCOSHB=0.5D0*(EP+EM)
DSINHB=NSIGN*DSINHB
90  CSINKD=DSINA*DCOSHB+CJ*DCOSA*DSINHB
    CCOSKD=DCOSA*DCOSHB-CJ*DSINA*DSINHB
100 CONTINUE
    CCU=CFPOF*(CU(N)/CU(N+1))/CKZ
    CCE=CGPOG*(CE(N)/CE(N+1))/CKZ
    CFOF1=CPF*CCOSKD-CCU*CSINKD
    CGOG1=CCOSKD-CCE*CSINKD
    CFPOF1=CKZ*(CPF*CSINKD+CCU*CCOSKD)
    CGPOG1=CKZ*(CSINKD+CCE*CCOSKD)
    CFPOF=CFPOF1/CFOF1
    CGPOG=CGPOG1/CGOG1
    CPF=C1
    N=N-1
    IF(N.GT.0) GO TO 40
200 CW1=-CJ*XKO*CE(1)/CGPOG
    CW2=CJ*CFPOF/(XKO*CU(1))
    RETURN
    END

```



```
SFJ=MJ*FEJ
TFI=XESI*FEI
TFJ=XESJ*FEJ
TFEI=-0.5D0*MI*MI/(XEI*DEI)
TFEJ=-0.5D0*MJ*MJ/(XEJ*DEJ)
RETURN
END
```

SUBROUTINE UVIJ(BETA,UIJ,VIJ,IT)

C.....
 c: Evaluates functions in table I or table II. :
 C.....

```

    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION BJ(21)
    COMMON /FACTS/ MI,MJ,MP,MM,MPP,MMP,MI1,MJ1,MXP,MPI,MPJ,MPPI,MPPJ,
  A      EM1,EM2,AIKO,AJKO,RKO,XEI,XEJ,XMI,XMJ,
  B      XESI,XESJ,XMSI,XMSJ,SFI,SFJ,TFI,TFJ,TFEI,TFEJ,
  C      XDS,XDC,XDSP,XDCP,XDSM,XDCM

    UIJ=0.0D0
    VIJ=0.0D0
    IF(RKO.GT.1.0D-04) GO TO 150
    IF(MI.NE.MJ) RETURN
    GO TO (110,120,130,140)IT
110   UIJ=-EM1*XDC
      VIJ=EM2*XDC
      RETURN
120   UIJ=-EM2*XDC
      RETURN
130   UIJ=EM1*XDS
      RETURN
140   UIJ=EM1*XDS
      RETURN
150   ARG=RKO*BETA
      CALL BESJS(ARG,BJ,MPP)
      GO TO (160,190,220,250)IT
160   T1=BJ(MPP)*XDCP
      IF(MI.GT.MJ) GO TO 170
      T2=MI1*BJ(MMP)*XDCM
      GO TO 180
170   T2=MJ1*BJ(MXP)*XDCM
180   UIJ=MJ1*(T1-T2)
      VIJ=MJ1*(T1+T2)
      RETURN
190   T1=BJ(MPP)*XDCP
      IF(MI.GT.MJ) GO TO 200
      T2=MI1*BJ(MMP)*XDCM
      GO TO 210
200   T2=MJ1*BJ(MXP)*XDCM
210   UIJ=-MJ1*(T1+T2)
      RETURN
220   T1=BJ(MPP)*XDSP
      IF(MI.GT.MJ) GO TO 230
      T2=MI1*BJ(MMP)*XDSM
      GO TO 240
230   T2=MJ1*BJ(MXP)*XDSM
240   UIJ=MJ1*(T1-T2)
      RETURN
250   T1=BJ(MPP)*XDSP
  
```



```
IF(MI.GT.MJ) GO TO 260
T2=MI1*BJ(MMP)*XDSM
GO TO 270
260 T2=MJ1*BJ(MXP)*XDSM
270 UIJ=MJ1*(T1-T2)
RETURN
END
```

SUBROUTINE STEI(BETA, SS, TT)

```

C.....
c: Evaluates equations 12 and 13 for i-th aperture.
C.....
  IMPLICIT REAL*8(A-H, O-Z)
  DIMENSION BJ(21)
  COMMON /FACTS/ MI, MJ, MP, MM, MPP, MMP, MI1, MJ1, MXP, MPI, MPJ, MPPI, MPPJ,
A          EM1, EM2, AIKO, AJKO, RKO, XEI, XEJ, XMI, XMJ,
B          XESI, XESJ, XMSI, XMSJ, SFI, SFJ, TFI, TFJ, TFEI, TFEJ,
C          XDS, XDC, XDSP, XDCP, XDSM, XDCM

  ARG=AIKO*BETA
  CALL BESJS(ARG, BJ, MPPI)
  BJA=BJ(MPI)/ARG
  SS=SFI*BJA
  DN=XESI-ARG*ARG
  IF(DABS(DN).GT. 1.0D-05) GO TO 10
  CALL BESJS(XEI, BJ, MPI)
  TT=TFEI*BJ(MPI)
  RETURN
10  TT=TFI*(MI*BJA-BJ(MPPI))/DN
  RETURN
  END

```

SUBROUTINE STMI(BETA, SS, TT)

```

C.....
c: Evaluates equations 14 and 15 for i-th aperture.
C.....
  IMPLICIT REAL*8(A-H, O-Z)
  DIMENSION BJ(21)
  COMMON /FACTS/ MI, MJ, MP, MM, MPP, MMP, MI1, MJ1, MXP, MPI, MPJ, MPPI, MPPJ,
A          EM1, EM2, AIKO, AJKO, RKO, XEI, XEJ, XMI, XMJ,
B          XESI, XESJ, XMSI, XMSJ, SFI, SFJ, TFI, TFJ, TFEI, TFEJ,
C          XDS, XDC, XDSP, XDCP, XDSM, XDCM

  TT=0.0D0
  ARG=AIKO*BETA
  DN=XMSI-ARG*ARG
  IF(DABS(DN).GT. 1.0D-05) GO TO 10
  CALL BESJS(XMI, BJ, MPPI)
  SS=AIKO*0.5D0*BJ(MPPI)
  RETURN
10  CALL BESJS(ARG, BJ, MPI)
  SS=AIKO*ARG*BJ(MPI)/DN
  RETURN
  END

```

SUBROUTINE STEJ(BETA,SS,TT)

```

C.....:
c: Evaluates equations 12 and 13 for j-th aperture.      :
C.....:
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION BJ(21)
  COMMON /FACTS/ MI,MJ,MP,MM,MPP,MMP,MI1,MJ1,MXP,MPI,MPJ,MPPI,MPPJ,
A          EM1,EM2,AIKO,AJKO,RKO,XEI,XEJ,XMI,XMJ,
B          XESI,XESJ,XMSI,XMSJ,SFI,SFJ,TFI,TFJ,TFEI,TFEJ,
C          XDS,XDC,XDSP,XDCP,XDSM,XDCM

  ARG=AJKO*BETA
  CALL BESJS(ARG,BJ,MPPJ)
  BJA=BJ(MPJ)/ARG
  SS=SFJ*BJA
  DN=XESJ-ARG*ARG
  IF(DABS(DN).GT.1.0D-05) GO TO 10
  CALL BESJS(XEJ,BJ,MPJ)
  TT=TFEJ*BJ(MPJ)
  RETURN
10  TT=TFJ*(MJ*BJA-BJ(MPPJ))/DN
  RETURN
  END

```

SUBROUTINE STMJ(BETA,SS,TT)

```

C.....:
c: Evaluates equations 14 and 15 for j-th aperture.      :
C.....:
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION BJ(21)
  COMMON /FACTS/ MI,MJ,MP,MM,MPP,MMP,MI1,MJ1,MXP,MPI,MPJ,MPPI,MPPJ,
A          EM1,EM2,AIKO,AJKO,RKO,XEI,XEJ,XMI,XMJ,
B          XESI,XESJ,XMSI,XMSJ,SFI,SFJ,TFI,TFJ,TFEI,TFEJ,
C          XDS,XDC,XDSP,XDCP,XDSM,XDCM

  TT=0.0D0
  ARG=AJKO*BETA
  DN=XMSJ-ARG*ARG
  IF(DABS(DN).GT.1.0D-05) GO TO 10
  CALL BESJS(XMJ,BJ,MPPJ)
  SS=AJKO*0.5D0*BJ(MPPJ)
  RETURN
10  CALL BESJS(ARG,BJ,MPJ)
  SS=AJKO*ARG*BJ(MPJ)/DN
  RETURN
  END

```

SUBROUTINE BESJS(XX,BJ,MP)

```

C.....
c: Evaluates Bessel function of first kind or orders 0 through MP-1 :
c: for argument XX, and stores values in BJ(1) through BJ(MP). :
c: BJ must be dimensioned at least (MP+1) in calling program. :
C.....
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(1),B(130)
      DATA PI,PI02 /3.14159265358979D0,1.57079632679490D0/
      DATA BIG,SMALL,PI04 /1.0D+38,1.0D-38,0.78539816339745D0/
      DATA TWOTRD /0.666666666666667D0/
      X=DABS(XX)
      BJ(1)=1.0D0
      MT=MP-1
      N=MT+1
      IF(X.LT.80.)GO TO 110
      DO 100 I=1,N
100    BJ(I)=DSQRT(2.0D0/(PI*X))*DCOS(X-PI04-PI02*(I-1))
      GO TO 310
110    CONTINUE
      DO 120 I=2,N
120    BJ(I)=0.0D0
      IF(X.LT.1.D-06)RETURN
      IF(X-15.)130,130,140
130    NTEST=20.0D0+10.0D0*X-X**TWOTRD
      GO TO 150
140    NTEST=90.0D0+0.5D0*X
150    IF(MT-NTEST)170,160,160
160    N=NTEST-1
      GO TO 180
170    N=MT
180    BPREV=0.0
      N1=N+1
      F=2.0D0/X
      D=1.0D-06
      IF(X-5.)190,200,200
190    MA=X+6.0D0
      GO TO 210
200    MA=1.4D0*X+60.0D0/X
210    IFIXX=X
      MB=N+IFIXX/4+2
      MZERO=MAX0(MA,MB)
      MMAX=NTEST
      DO 280 M=MZERO,MMAX,3
      FM1=SMALL
      FM=0.0D0
      ALPHA=0.0D0
      IF(M-(M/2)*2)230,220,230
220    JT=-1
      GO TO 240

```

```

230  JT=1
240  M2=M-2
      DO 260 K=1, M2
      MK=M-K
      B(MK)=F*MK*FM1-FM
      IF(B(MK)-BIG)250, 310, 310
250  FM=FM1
      FM1=B(MK)
      JT=-JT
      S=1+JT
260  ALPHA=ALPHA+B(MK)*S
      B(1)=F*FM1-FM
      ALPHA=ALPHA+B(1)
      BTEST=B(N1)/ALPHA
      IF(DABS(BTEST-BPREV)-DABS(D*BTEST))290, 290, 270
270  BPREV=BTEST
280  CONTINUE
290  DO 300 I=1, N1
300  BJ(I)=B(I)/ALPHA
310  IF(XX.LT.0.0D0)GO TO 320
      RETURN
320  N=MT+1
      DO 330 I=1, N
      NN=I-1
330  BJ(I)=BJ(I)*(-1.0)**NN
      RETURN
      END

```



Report Documentation Page

1. Report No. NASA TM-101614	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle CWG - A Fortran Program for Mutual Coupling in a Planar Array of Circular Waveguide-fed Apertures		5. Report Date June 1989	6. Performing Organization Code
		8. Performing Organization Report No.	
7. Author(s) M. C. Bailey		10. Work Unit No. 506-44-21-03	
		11. Contract or Grant No.	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225		13. Type of Report and Period Covered Technical Memorandum	
		14. Sponsoring Agency Code	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001			
15. Supplementary Notes			
16. Abstract <p>This paper is a documentation of a Fortran program which calculates the mutual coupling between circular apertures in a conductive plane. The program is quite general in that the apertures do not have to be the same sizes, nor do they have to be polarized in the same direction. In addition, several waveguide modes (TE and/or TM) may be specified in the apertures and the mutual coupling between all combinations of apertures and modes will be calculated. The program also allows multiple layers of homogeneous dielectrics to be placed over the aperture array. Outside the layered region, one can specify either a homogeneous half-space, or a perfect reflecting surface.</p>			
17. Key Words (Suggested by Author(s)) antenna apertures arrays coupling		18. Distribution Statement Unclassified - Unlimited Subject Category 32	
19. Security Classif. (of this report) unclassified	20. Security Classif. (of this page) unclassified	21. No. of pages 45	22. Price A03