AD-A059 813 ARIZONA UNIV TUCSON F/G 20/14

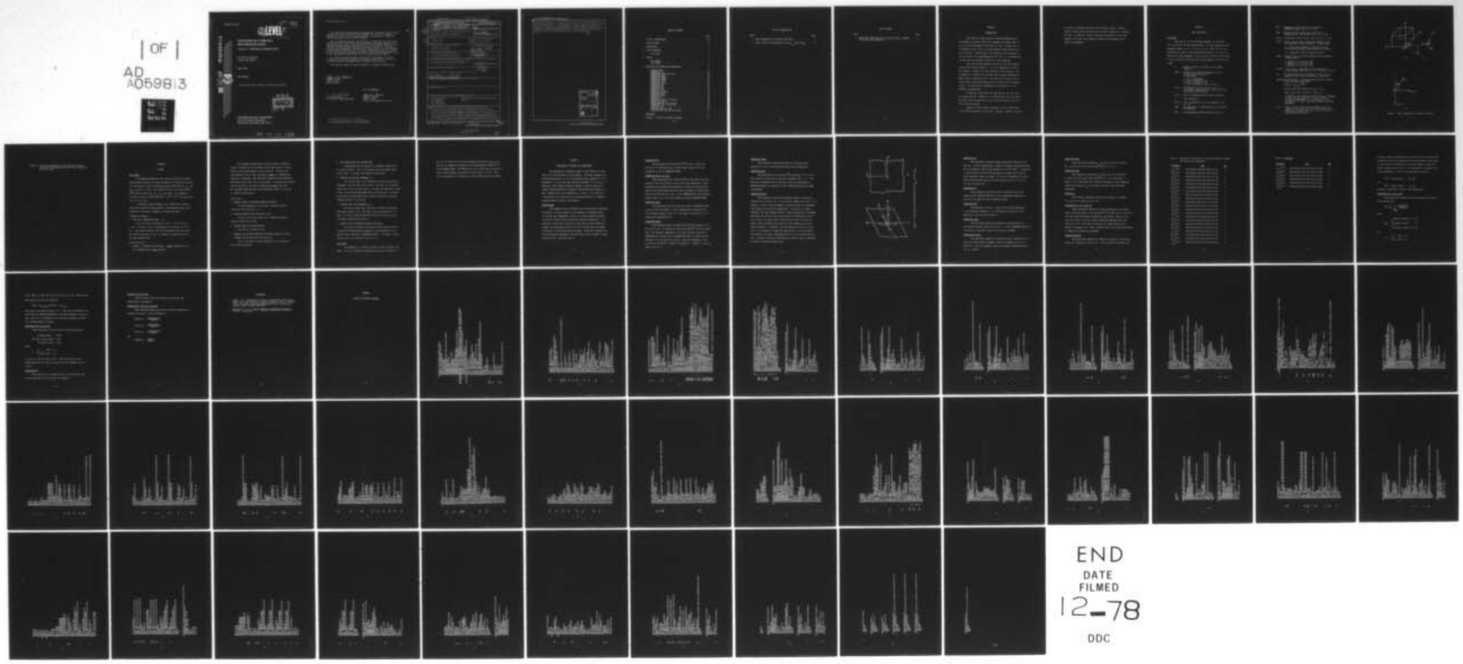
EXCITATION OF A WIRE IN A RECTANGULAR CAVITY. VOLUME II. COMPUT--ETC(U)

MAY 78 D B SEIDEL AFOSR-76-3009

UNCLASSIFIED AFWL-TR-77-221-VOL-2 NL

1 OF 1

AD A059813



END

DATE FILMED

12-78

DDC

AFWL-TR-77-221

AFWL-TR
77-221

EXCITATION OF A WIRE IN A RECTANGULAR CAVITY

Volume II: COMPUTER DOCUMENTATION

University of Arizona
Tucson, AZ 85724

May 1978

Final Report

Approved for public release; distribution unlimited.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

This final report was prepared by the University of Arizona, Tucson, Arizona, through AFOSR Grant 76-3009, Job Order 12090525 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Capt Howard G. Hudson (ELT) was the Laboratory Project Officer-in-Charge.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Information (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Howard G. Hudson

HOWARD G. HUDSON
Captain, USAF
Project Officer

FOR THE COMMANDER

J. Philip Castillo

J. PHILIP CASTILLO
Acting Chief, Technology Branch

Donald A. Dowler

DONALD A. DOWLER
Colonel, USAF
Chief, Electromagnetics Division

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-E200 165

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWL-TR-77-221, Vol II - VOL-2	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): EXCITATION OF A WIRE IN A RECTANGULAR CAVITY. Volume II. Computer Documentation.	5. TYPE OF REPORT & PERIOD COVERED: Final Report,
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): D. B. Seidel	8. CONTRACT OR GRANT NUMBER(s): AFOSR-76-3009
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Arizona, Tucson, Arizona 85724	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 64747F, 12090525
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Weapons Laboratory (ELT), Kirtland Air Force Base, NM 87117	12. REPORT DATE: May 1978
	13. NUMBER OF PAGES: 58
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Air Force Office of Scientific Research (NP), Bolling Air Force Base, Wash D C 20332	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(6) (9) (10) (11) (12) 64p. (12) 95 (15) (16) (18) AFWL, SBIE (19)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Rectangular Cavity
Small Aperture
Wire Scatterer
Shielding Effects
Dyadic Green's Functions for Rectangular Cavity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (UNCLASSIFIED)

In this work, the problem of determining the currents excited on a wire enclosed within a rectangular cavity is considered. The wire and cavity interior are excited by electromagnetic sources exterior to the cavity which couple to the cavity interior through a small aperture in the cavity wall. It is assumed that the wire is thin, straight and oriented perpendicular to one of the cavity walls. An integral equation is formulated for the problem in the frequency domain using equivalent dipole moments to approximate the

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

033 800

LB

effects of the aperture. This integral equation is then solved numerically by the method of moments. The dyadic Green's functions for this problem are difficult to compute numerically; consequently, extensive numerical analysis is necessary to render the solution tractable. Sample numerical results are presented for representative configurations of cavity, wire and aperture, and suggestions for future extensions of this work are discussed.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This report has been written to provide documentation to the computer code which numerically implements the theory found in [1] for the electromagnetic excitation of a thin, straight wire in a rectangular cavity which is excited through a small aperture in the cavity wall. Because many of the equations and discussions in [1] are crucial to an understanding of this code, it is assumed that the user has that document available for ready reference.

This code has been prepared for use on a CDC 6400 computer using the RUN fortran compiler. It is also compatable with CDC's FTN compiler. However, for later versions of this compiler it may be necessary to suppress the non-fatal error message "ARGUMENT TOO SMALL" which is detected in EXP. This error will naturally occur during normal program execution and has no effect upon the program output. This error can be suppressed by an appropriate call to SYSTEMC in program START.

In addition, every effort has been made to write the code in standard fortran. However, it is expected that a few minor modifications might be needed for its use on non-CDC machines and with other fortran compilers.

Chapter 2 of this document provides a list and description of all input parameters for this code. Similarly, chapter 3 outlines the various information which the code returns as output. These two chapters should provide the user with the needed information to operate the code. In addition, chapter 4 discusses the function of each sub-program in the code, and a complete listing of all programs can be found in the appendix.

CHAPTER 2

INPUT PARAMETERS

File INPUT

Note that all of the following parameters are read from the file INPUT in the main program START. All input parameters that represent lengths (A, B, C, XC, YC, R, ZL, ZU, LBDA, XPP, YPP, RA, DELFIX) may be input in any system of units desired, so long as the same unit is used throughout. Also note that if LBDA is set to unity, it has the effect of normalizing all other lengths in units of wavelength.

N	--	Number of pulses to be used in wire current expansion
IGRD	--	Integer which indicates connection of wire to one or both cavity walls. If IGRD equals: 0, wire is unattached -1, wire is attached at z=0 1, wire is attached at z=c 2, wire is attached at both z=0 and z=c
A,B,C	--	The dimensions of the cavity in the x, y, z directions, respectively (note that cavity must be oriented such that $A \leq B$)
XC,YC	--	The x-y coordinates of the center of the wire
R	--	The wire radius
ZL,ZU	--	The z coordinates of the wire endpoints, with ZL < ZU
LBDA	--	The wavelength (λ) associated with the frequency of operation
NSM	--	Maximum number of terms used for the sum S3

MAX -- Maximum value of m permitted in the sum of SUBROUTINE BADSUM (See [1], p. 57-58)

MAXX -- Maximum value of m permitted in the sums of subroutines GM1, GM2, and GM3 (See [1], p. 57-58)

CS,LS -- Convergence criteria for all sums (See [1], p. 57-59)

ADYAD -- Logical variable which controls the method in which the components of $\bar{\bar{G}}_e$ are computed. If ADYAD equals:

(.T.) first-order difference techniques are used to calculate $\bar{\bar{G}}_e$ from $\bar{\bar{G}}_A$ using equation (2.2) from [1].

(.F.) components of $\bar{\bar{G}}_e$ are computed directly

NWALL -- Indicates the cavity wall perforated by the aperture. If NWALL equals:

1, aperture is in x=0 wall ($\hat{n}=\hat{x}$)
2, aperture is in y=0 wall ($\hat{n}=\hat{y}$)
3, aperture is in z=0 wall ($\hat{n}=\hat{z}$)

XPP,YPP -- The x_1 and x_2 coordinates of the centerpoint of of the aperture, such that $\hat{a}_1 \times \hat{a}_2 = \hat{n}$ (see [1], p. 23)

RA -- An array containing the semi-axes of the elliptical aperture in the x_1 and x_2 dimensions, respectively

EMAG,PHASAP,THE,PHI,ANG -- Parameters controlling the incident plane wave excitation, where

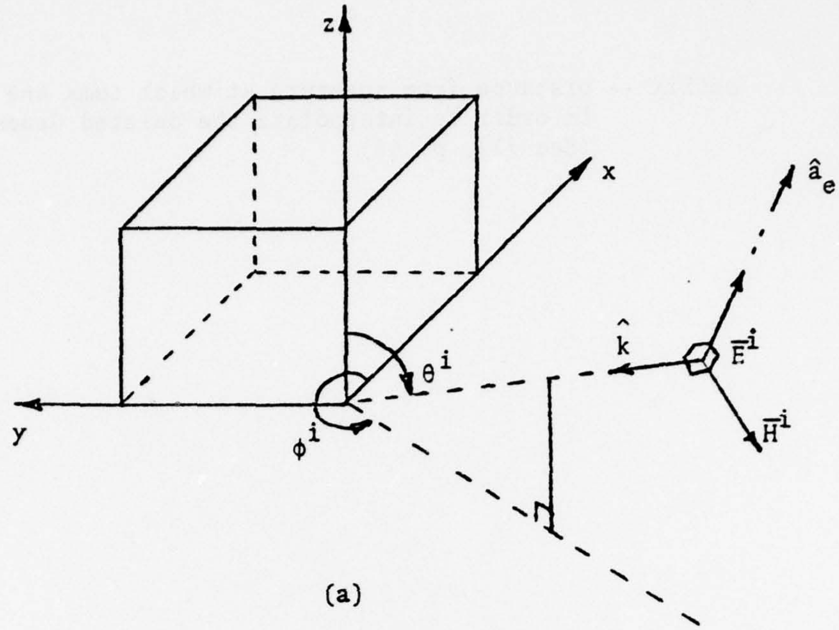
$$\bar{E}^{inc}(\bar{r}) = \hat{a}_e |E_o| e^{-j\hat{k}\cdot\bar{R}}$$

where $\hat{a}_e$ and $\hat{k}$ are defined in Figure 1 with

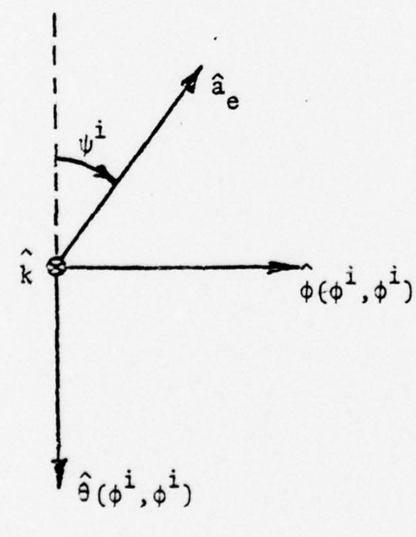
$$EMAG = |E_o|, \; THE = \theta^i, \; PHI = \phi^i, \; ANG = \psi^i.$$

If PHASAP = (.F.), phase of incident field is referenced to the coordinate origin ($\bar{R} = \bar{r}$) and if PHASAP = (.T.), the phase is referenced to the centerpoint of the aperture ($\bar{R} = \bar{r} - \bar{r}_a$). Note that all angles are input in degrees.

FIX -- Logical variable which controls whether or not the effects of the cavity walls upon the aperture dipole moments are included (See [1], p. 54). If FIX = (.T.), such effects are included.



(a)



(b)

Figure 1. Input Parameters for Incident Plane Wave.

DELFIX -- Distance from aperture at which sums are computed in order to interpolate the deleted Green's functions (See [1], p. 68)

CHAPTER 3

OUTPUT

File OUTPUT

The program automatically will create the print file OUTPUT. During normal execution, the input information of the previous chapter will be printed, as well as the aperture polarizabilities α_e, $\alpha_{m_{11}}$ and $\alpha_{m_{22}}$ (see [1], p. 17-19). The solution for the wire current and the total aperture fields (E_{T_n}, H_{T_1}, H_{T_2}) are given. For comparison, the exterior short circuit fields (E_n^{sc-}, H_1^{sc-}, H_2^{sc-}) are also given (see [1], pp. 20-26).

In addition, several messages of an informative or warning nature may be printed during execution. These messages which can be divided into three basic categories, are described below:

Informative messages --

1. NOTE--W(I) INTERPOLATED THRU I = n

W(I) is an array containing P(α) for α = (I-1)Δ (see [1], p. 63). For small α, P(α) is interpolated as discussed on p.65 of [1]. This message indicates that this approximation has been made for the first n values of P(α). To reduce n, increase the size of the input parameter MAX.

Non-Fatal Errors --

1. WARNING -- DIMENSION SIZE EXCEEDED -- name1 TRUNCATED TO n. TO FIX, INCREASE SIZE OF name2 ARRAY(S).

This message indicates that the arrays used for temporary storage in BADSUM, GM1, GM2 and GM3 are not large enough to sum the series to the maximum number of terms specified. The particular input parameter (MAX or MAXX, specified by name1) is truncated and execution is continued. Note that this truncation could subsequently cause either fatal error 5 or 6 listed below. To rectify this problem, increase the size of the array(s) specified by name2, such that MS1 > max(MAX, MAXX) and MS2 > max(c_1MAX,c_2MAXX) where c_1 = B/A and c_2 = max(A,B,C)/min(A,B,C).

Fatal Errors --

1. WARNING--MATRIX SIZE EXCEEDS STORAGE ALLOCATION

The input parameter N is too large. Decrease N and/or increase MS such that MS $\geq$ N + 3.

2. WARNING--REORIENT CAVITY SUCH THAT A.LE.B

A must be less than or equal to B. Redefine coordinate system such that this is true.

3. WARNING--ERROR IN WIRE END POINTS

Note that ZL < ZU must be true

4. WARNING--ALL OR PART OF THE WIRE IS OUTSIDE THE CAVITY, OR UNATTACHED WIRE HAS END POINT ON CAVITY WALL

Check to see that the input parameters do not violate any of the above conditions.

5. NOT ENOUGH POINTS FOR INTERPOLATION

Interpolation of P (α) requires its numerical evaluation at at least two points. This is not possible with the present input value of MAX. To correct this problem, increase MAX.

6. WARNING--GMI SUM NOT CONVERGED c ℓ

The value of MAXX is not sufficient to obtain the specified convergence (CS,LS) in GM1, GM2, or GM3. Note that c is the convergence ratio at the last term and ℓ is the number of consecutive times, if any, the specified convergence ratio CS has been met. To rectify this problem, increase MAXX and/or reduce the severity of the convergence specified by CS and LS.

7. WARNING--SUM 3 NOT CONVERGED c ℓ

The value of NSM is not sufficient to obtain the specified convergence (CS,LS) in S3. Note that c and ℓ are defined as in 6 above. To recify, increase NSM and/or reduce the severity of the convergence specified by CS and LS.

8. WARNING--CAVITY DIMENSION TOO SMALL FOR DPLFIX

This error is caused by insufficient room inside the cavity to perform the interpolation necessary for the evaluation of the deleted Green's functions in SUBROUTINE DPLFIX. To correct, either decrease the size of DELFIX or let FIX = (.F.).

File PUNCH

The program will in certain situations create the punch file PUNCH. This file contains the information from the file INPUT, as well as the solution for the wire current and the fields (E_{T_n}, H_{T_1}, H_{T_2}). The file is formatted according to the PUNCH statements found at the end of PROGRAM START. The PUNCH file will be created only if execution has not been abnormally halted and if sense switch 1 is "on". This can be accomplished on CDC machines with the SCOPE control card ONSW(1).

CHAPTER 4

DESCRIPTION OF PROGRAM AND SUBPROGRAMS

The programs and subprograms listed in this chapter are those used in the code described in this document. Uniformly throughout the following description, it is to be understood that variables (X, Y, Z) represent $\bar{r}$ or (x,y,z) and that (XP,YP,ZP) represent $\bar{r}'$ or (x',y',z'). Similarly, other program variables (denoted in capital letters) obviously represent physical variables of the problem (for example, A, B and C represent the cavity dimensions a, b, and c). Only when such a connection is not obvious will it be explicitly noted. A complete program listing is given in the appendix.

PROGRAM START

This program's role is primarily that of input/output operations. In addition, it also provides for the checking of consistancy among the various input parameters. Finally, it directs program execution by calling the basic matrix filling and inverting subprograms. It is important to note that if the size of the arrays in blank COMMON are changed, the appropriate values of MS, MS1, and MS2 should be changed accordingly to insure proper error checking. Sufficient dimension size can be obtained by adhering to the algorithms listed in chapter 3 under non-fatal error 1 and fatal error 1.

SUBROUTINE FILL

This subroutine fills the matrix $\overline{\overline{Q}}^a$ (see [1], p.52,63) for the case of an unattached wire. The necessary values of P(β) are provided by a call to SUBROUTINE GETUM.

SUBROUTINES FILL1 and FILL2

These subroutines fill the matrix $\overline{\overline{Q}}^a$ for the cases of the wire attached to the cavity walls at one or both ends respectively. They incorporate the necessary modifications to account for non-zero half-pulse and the half-testing functions which are needed at the attached ends of the wire (see [1], p.50). As in SUBROUTINE FILL, the necessary values of P(β) ([1], p.63) are provided by calling SUBROUTINE GETUM.

SUBROUTINE GETUM

This subroutine fills the array W(I) with the necessary values of P(β) to fill the matrix. It will automatically interpolate P(β) for sufficiently small β (see [1], p.65) using a three-point fit to a second order polynominal in β.

SUBROUTINE MORFIL

This subroutine fills the matrices $\overline{\overline{Q}}^b$ and $\overline{\overline{Q}}^c$ according to (4.7) and (4.8) in [1]. In addition it also initializes $\overline{\overline{Q}}^d$ to $\overline{\overline{I}}$, the identity dyad. The necessary components of the dyads have been calculated in SUBROUTINE EAP (through calls to SUBROUTINES EDIPL and MDIPL) and are contained in the arrays FP, G1 and G2. Note that advantage is taken of the fact that $\overline{\overline{G}}_e(\overline{r},\overline{r}') = \tilde{\overline{\overline{G}}}_e(\overline{r}',\overline{r})$ and $\overline{\overline{g}}_e(\overline{r},\overline{r}') = -\tilde{\overline{\overline{G}}}_H(\overline{r}',\overline{r})$ {see [1], Table 2 and (2.7)}.

SUBROUTINE INVERS

This subroutine numerically solves the resulting matrix equation (4.5) in [1] using partial-pivoting Gauss Elimination.

SUBROUTINE FIXUP

This subroutine fills the matrix $\bar{\bar{Q}}^{d}$ according to (4.9) in [1]. This routine is called only if the input parameter FIX = (.T.). The values of the deleted Green's functions, which are calculated by SUBROUTINE DPLFIX, are supplied to this subprogram through the common block/WHYNOT/.

SUBROUTINE DPLFIX

This subroutine calculates the deleted Green's functions as outlined on pp. 67-70 of [1]. It is called by START only if FIX = (.T.). Figures 2 (a and b) show arrangement of points at which the components of $\bar{G}_A$ and $\bar{g}_F$ are calculated, respectively. Note that Δ in Figure 2 represents the input parameter DELFIX. These computations are somewhat complicated by the fact that if the aperture is sufficiently close to one or more of the side walls of the cavity the choice of points in Figure 2 must be modified such that all points remain on the cavity interior surface. In addition, when the aperture is close to a side wall, it is necessary to remove the image source in that wall before interpolation, and then to subsequently add it to the interpolated result. Because of these two complications, extensive logic is necessary in DPLFIX to adequately handle them.

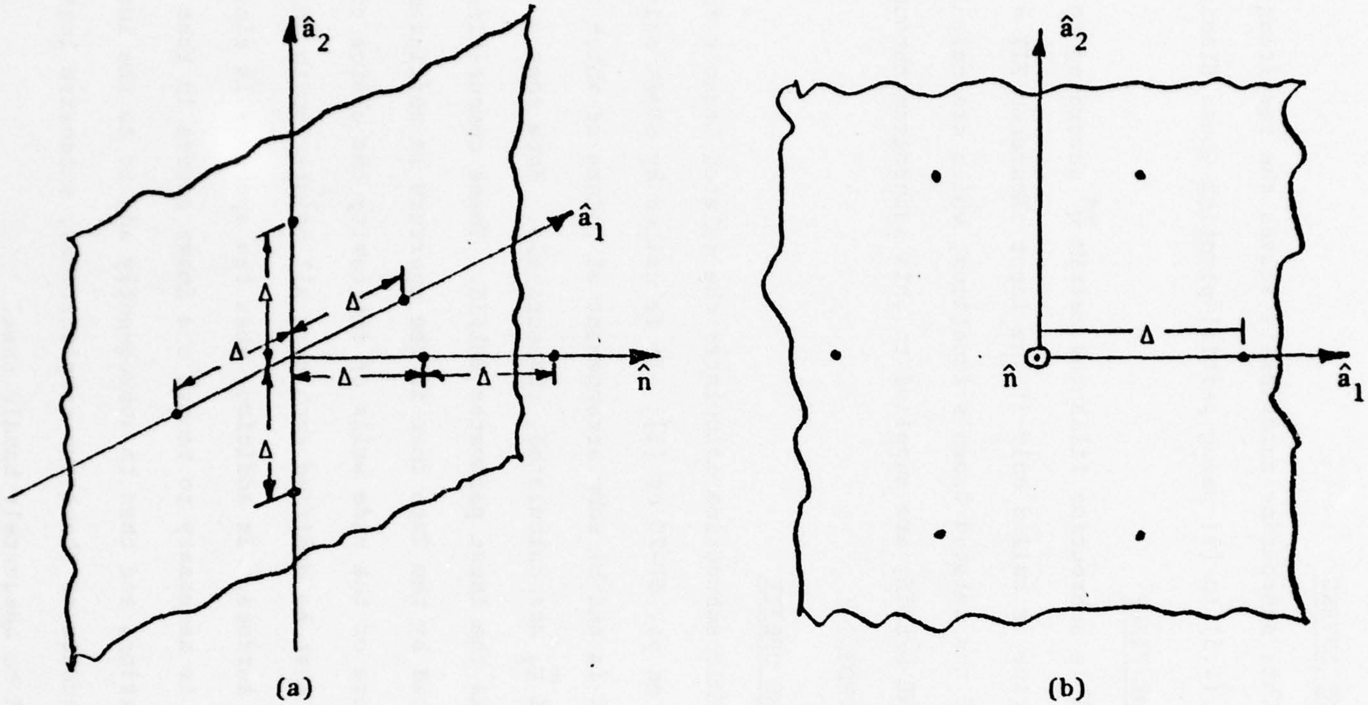


Figure 2. Points Used for Interpolation of (a) $\hat{G}_{A_{nn}}$, and (b) $\hat{g}_{F_{11}}$.

SUBROUTINE EAP

This subroutine calculates several quantities relating to the aperture. It calls subroutines to compute the aperture polarizabilities, and the components of $\overline{G}_h$ and $\overline{g}_e$ necessary to fill $\overrightarrow{Q}^b$ and $\overrightarrow{Q}^c$. In addition it calls routines which convert the exterior incident field from the input parameters (shown in Figure 1) to the x,y,z components of E and H. These values are then used to fill the excitation vector in (4.5) of [1] into the array E(I).

SUBROUTINE INC

This subroutine converts the input specifications for the incident field (EMAG,ANG,THE,PHI) to the corresponding components of E and H in the spherical polar coordinate system.

SUBROUTINE CONV

This subroutine converts a vector from its polar components to its cartesian components. It is assumed that the vector is perpendicular to the polar unit vector $\hat{r}$.

SUBROUTINE ALPHA

This subroutine computes the aperture polarizabilities for the elliptic aperture using (2.15) of [1]. It calls SUBROUTINE ELLIPTIC to provide the necessary values of the elliptic integrals.

SUBROUTINE ELLIPTIC

This subroutine computes the complete elliptic integrals of the first and second kinds of argument X which are denoted EK and E, respectively. They are computed using the polynominal approximations of [2], pp. 591-592.

SUBROUTINE MDIPL

This subroutine calculates $G_{h_{Jz}}(\overline{r}_a,\overline{r}_p)$ for use in the matrix $\overline{\overline{Q}}^c$. It fills the array E with these values for p = 1, N.

SUBROUTINE EDIPL

This subroutine calculates $G_{e_{Jz}}(\overline{r}_a,\overline{r}_p)$ for use in matrix $\overline{\overline{Q}}^c$. It fills the array E with these values for p = 1, N. Note that if ADYAD = (.T.), these values are computed indirectly by first computing appropriate values of $G_{A_{zz}}$ and using first-order finite difference techniques.

FUNCTION S3

This function numerically computes the function S_3, defined by (3.2c) in [1], using (3.16) in [1].

SUBROUTINES GA, GF, GM and GE

These subroutines compute the various components of the dyadic Green's functions needed in the matrices $\overline{\overline{Q}}^b$, $\overline{\overline{Q}}^c$ and $\overline{\overline{Q}}^d$ (see [1], pp.52-54). Note that these subroutines automatically analytically reduce the sum that will result in the most rapidly converging remaining double sum (see [1], p. 55). Note that IN, SX and SY are dummy arrays and the answer is returned as SD. Table 1 shows the way to call these subroutines to compute any desired dyad component.

SUBROUTINE BADSUM

This subroutine computes the indefinite integral of the reduced kernel P(β) defined by (4.22) in [1]. The m-n plane is subdivided into

Table 1. Appropriate subroutine call and value of GEZ to compute the various dyad components.

To Compute	CALL	GEZ
$G_{A_{xx}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,Y,YP,Z,ZP,X,XP,B,C,A,SD)	F
$G_{A_{yy}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,Z,ZP,X,XP,Y,YP,C,A,B,SD)	F
$G_{A_{zz}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)	F
$g_{F_{xx}}(\bar{r},\bar{r}')$	GF(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)	F
$g_{F_{yy}}(\bar{r},\bar{r}')$	GF(IN,SX,SY,Y,YP,Z,ZP,X,XP,B,C,A,SD)	F
$g_{F_{zz}}(\bar{r},\bar{r}')$	GF(IN,SX,SY,Z,ZP,X,XP,Y,YP,C,A,B,SD)	F
$G_{e_{xx}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,Y,YP,Z,ZP,X,XP,B,C,A,SD)	T
$G_{e_{xy}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,XP,X,ZP,Z,YP,Y,A,C,B,SD)	F
$G_{e_{xz}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,XP,X,YP,Y,ZP,Z,A,B,C,SD)	F
$G_{e_{yx}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,X,XP,Z,ZP,Y,YP,A,C,B,SD)	F
$G_{e_{yy}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,Z,ZP,X,XP,Y,YP,C,A,B,SD)	T
$G_{e_{yz}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,YP,Y,XP,X,ZP,Z,B,A,C,SD)	F
$G_{e_{zx}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)	F
$G_{e_{zy}}(\bar{r},\bar{r}')$	GE(IN,SX,SY,Y,YP,X,XP,Z,ZP,B,A,C,SD)	F
$G_{e_{zz}}(\bar{r},\bar{r}')$	GA(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)	T
$-G_{h_{xy}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,XP,X,ZP,Z,YP,Y,A,C,B,SD)	F
$G_{h_{xz}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,XP,X,YP,Y,ZP,Z,A,B,C,SD)	F
$G_{h_{yx}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,YP,Y,ZP,Z,XP,X,B,C,A,SD)	F
$-G_{h_{yz}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,YP,Y,XP,X,ZP,Z,B,A,C,SD)	F
$-G_{h_{zx}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,ZP,Z,YP,Y,XP,X,C,B,A,SD)	F
$G_{h_{zy}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,ZP,Z,XP,X,YP,Y,C,A,B,SD)	F

Table 1.--Continued

To Compute	CALL	GEZ
$-g_{e_{xy}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,Y,YP,Z,ZP,X,XP,B,C,A,SD)	F
$g_{e_{xz}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,Z,ZP,Y,YP,X,XP,C,A,B,SD)	F
$g_{e_{yx}}(\bar{r},\bar{r}')$	GM,IN,SX,SY,X,XP,Z,ZP,Y,YP,A,C,B,SD)	F
$-g_{e_{yz}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,Z,ZP,X,XP,Y,YP,C,A,B,SD)	F
$-g_{e_{zx}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)	F
$g_{e_{zy}}(\bar{r},\bar{r}')$	GM(IN,SX,SY,Y,YP,X,XP,Z,ZP,B,A,C,SD)	F

elliptical bands as discussed on pp. 55-59 of [1] and the double sum is evaluated using the convergence criteria discussed on those pages. If the sum does not converge in the allotted number of terms, the logical variable CIND is set to (.T.) and control is returned to the calling subprogram. SX and SY are input arrays defined as

$$SX(I) = \sin k_x x \sin k_x x' , \quad k_x = \frac{\pi}{a} I$$

and

$$SY(I) = \sin k_y y \sin k_y y' , \quad k_y = \frac{\pi}{b} I$$

The input X represents β and P(β) is returned as SD.

SUBROUTINES GM1, GM2 and GM3

These subroutines numerically sum two dimensional series of the following form:

$$GMI = \frac{1}{ab} \sum_{\substack{m=m_o \\ n=n_o}}^{\infty} \frac{\varepsilon_m \varepsilon_n u_m v_n w_{mn}}{\gamma_c \sinh \gamma_c c}$$

where

$$w_{mn} = \begin{cases} \sinh\gamma_c z_< \sinh\gamma_c (c-z_>) , & I=1 \\ \gamma_c \sinh\gamma_c z \cosh\gamma_c z' , & I=2 \\ \Gamma_1 \cosh\gamma_c z_< \cosh\gamma_c (c-z_>) , & I=3 \end{cases}$$

and

$$\Gamma_1 = \begin{cases} 1 , & GEZ = (.F.) \\ k_c^2 , & GEZ = (.T.) \end{cases}$$

for $k_x = \frac{m\pi}{a}$, $k_y = \frac{n\pi}{b}$, $k_c^2 = k_x^2 + k_y^2$ and $\gamma_c^2 = k_c^2 - k^2$. Note that the input arrays SX and SY are defined as

$$SX(I) = u_{(I-1+m_o)} \text{ and } SY(I) = v_{(I-1+n_o)}$$

and m_o and n_o can each be either 0 or 1. This sum is performed in the same fashion as SUBROUTINE BADSUM and the same convergence criteria is used. Note that if convergence is not achieved, execution is halted and a warning message is printed.

SUBROUTINES NSS, NCC and NSC

These subroutines fill the array D in the following ways:

$$D(I) = \begin{cases} \sin k_x x \sin k_x x' , & (NSS) \\ \Gamma_2 \cos k_y y \cos k_y y', & (NCC) \\ k_z \sin k_z z \cos k_z z' , & (NSC) \end{cases}$$

where

$$\Gamma_2 = \begin{cases} 1 & , GEZ = (.F.) \\ k^2 - k_y^2 & , GEZ = (.T.) \end{cases}$$

$k_x = \frac{\pi}{a} I$, $k_y = \frac{\pi}{b} (I-1)$ and $k_z = \frac{\pi}{c} I$. This subroutine is used to appropriately fill the SX and SY arrays for use in BADSUM, GM1, GM2 or GM3.

SUBROUTINE NDE

This subroutine is a dummy routine to call either NSS, NCC or NSC, depending upon the value of the integer L.

FUNCTIONS COSH and SINH

These functions compute the hyperbolic cosine and sine, respectively, of argument X.

FUNCTIONS RSS, RCC, RCS and RSINH

These functions compute the ratios of various combinations of hyperbolic functions. They are defined by

$$\mathrm{RSS}(X,Y,Z) = \frac{\sinh X \sinh Y}{\sinh Z}$$

$$\mathrm{RCC}(X,Y,Z) = \frac{\cosh X \cosh Y}{\sinh Z}$$

$$\mathrm{RCS}(X,Y,Z) = \frac{\cosh X \sinh Y}{\sinh Z}$$

and

$$\mathrm{RSINH}(X,Y) = \frac{\sinh X}{\sinh Y} .$$

REFERENCES

1. Seidel, D. B., "Excitation of a Wire in a Rectangular Cavity, Part I: Theory," Air Force Office of Scientific Research, Grant No. AFOSR76-3009, Final Report, Engineering Experiment Station, University of Arizona, Tucson, Arizona, June 1977.

2. Abramowitz, M. and I. Stegun, Handbook of Mathematical Functions, New York: Dover, 1965.

APPENDIX

LISTING OF COMPUTER PROGRAMS

```
      PROGRAM START(INPUT,OUTPUT,PUNCH)
      LOGICAL GEZ,ADYAD,FIX,PHASAP
      COMPLEX Q,D,S,ESC(3)
      REAL K,KK,LBDA
      DIMENSION RA(2)
      COMMON/LAST2/RA,EMAG,ANG,THE,PHI,PHASAP
      COMMON/SPEC/GEZ
      COMMON/TYPE/ADYAD
      COMMON/WN/K
      COMMON/FSUM/A,B,C,MAX
      COMMON/GM/MAXX
      COMMON/GRD/IGRD
      COMMON/SUMS/KK,NSM,CS,LS
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      COMMON/LAST/XPP,YPP,NWALL
C*******WARNING--DO NOT CHANGE DIMENSIONING OF ARRAYS IN BLANK COMMON
C*******WITHOUT CHANGING APPROPRIATE VALUES OF MS,MS1 AND/OR MS2!!!
      COMMON Q(36,36),D(36),S(36),W(36),FP(36)
     1 ,IN(86),SX(86),SY(285)
      DATA MS,MS1,MS2/36,86,285/
C*****************************************
      DATA ETA/376.7/
      DATA GEZ/.F./
      DATA PI/3.1415927/
      DATA XAM,XXAM/4H-MAX,4HMAXX/
      CALL SSWTCH(1,IPUNCH)
      READ 100,N,IGRD
      READ 101,A,B,C,XC,YC,R,ZL,ZU
      READ 101,LBDA
      READ 100,NSM,MAX,MAXX,LS,CS,ADYAD
      READ 800,PHASAP,NWALL,XPP,YPP,RA,EMAG,ANG,THE,PHI
      READ 202,FIX,DELFIX
      N3=N+3
      PRINT 99
      PRINT 300
      IF(IGRD)19,20,21
19    PRINT 129
30    ZL=0.
      GOTO 23
21    ZU=C
      IF(IGRD.EQ.1)GOTO 31
      PRINT 132
      GOTO 30
31    PRINT 131
      GOTO 23
20    PRINT 130
      IF(ZL.EQ.0..OR.ZU.EQ.C)GOTO 22
```

```
23    PRINT 301,A,B,C,LBDA
      PRINT 302,R,ZL,ZU,XC,YC
      PRINT 303,N
      PRINT 304,NSM,MAX,MAXX,CS,LS,ADYAD
      IF(ZL.LT.O..OR.ZU.GT.C)GOTO 22
      IF(ZU.GT.ZL)GOTO 17
      PRINT 116
      STOP
17    IF(XC.LT.A.A.XC.GT.O..A.YC.LT.B.A.YC.GT.O.)GOTO 18
22    PRINT 117
      STOP
18    IF(A.LE.B)GOTO 11
      PRINT 110
      STOP
11    IF(MS.GE.N3)GOTO 12
      PRINT 111
      STOP
12    IF(MS1.GT.MAX)GOTO 13
      MAX=MS1-1
      PRINT 112, XAM,MAX
13    CON=B/A
      NAX=CON*MAX
      IF(MS2.GT.NAX)GOTO 14
      MAX=(MS2-1)/CON
      PRINT 113,XAM,MAX
14    IF(MS1.GT.MAXX)GOTO 15
      MAXX=MS1-1
      PRINT 112,XXAM,MAXX
15    CON=AMAX1(A,B,C)/AMIN1(A,B,C)
      NAX=CON*MAXX
      IF(MS2.GT.NAX)GOTO 16
      MAXX=(MS2-1)/CON
      PRINT 113,XXAM,MAXX
16    NAX=B/A*MAX
      K=2.*PI/LBDA
      KK=K*K
      DEL=(ZU-ZL)/(N+1-IABS(IGRD))
      HDEL=.5*DEL
      CKD=2.*COS(K*DEL)
      CON=2.-CKD
      DQ=-.5*S3(YC,YC,XC,XC+R,B,A)
      CALL NSS(SX,XC,XC+R,A,MAX)
      CALL NSS(SY,YC,YC,B,NAX)
      IF(IGRD)1,2,3
3     IF(IGRD.EQ.1)GOTO 1
      CALL FILL2(IN,SX,SY,S,Q,MS,DQ)
      GOTO 4
```

```
1         CALL FILL1(IN,SX,SY,S,Q,MS,DQ)
          GOTO 4
2         CALL FILL(IN,SX,SY,D,S,Q,MS,DQ,W)
4         CALL EAP(IN,SX,SY,D,N,FP,W,S,CON)
          CALL MORFIL(Q,MS,FP,W,S,CON)
          IF(.N.FIX)GOTO 6
          CALL DPLFIX(DELFIX,IN,SX,SY,XPP,YPP,NWALL)
          CALL FIXUP(Q,MS)
          PRINT 305, DELFIX
6         DO 5 I=1,3
5         ESC(I)=D(I+N)
          CALL INVERS(Q,MS,MS,N3,D,S)
          PRINT 200
          Z=ZL
          IF(IGRD.EQ.-1.OR.IGRD.EQ.2)Z=Z-CEL
          DO 7 I=1,N
          Z=Z+DEL
7         PRINT 201,Z,S(I)
          PRINT 203,(S(I+N),ESC(I),I=1,3)
          IF(IPUNCH.NE.1)STOP
          PUNCH 100,N,IGRD
          PUNCH 101,A,B,C,XC,YC,R,ZL,ZU
          PUNCH 101,LBOA
          PUNCH 100,NSM,MAX,MAXX,LS,CS,ADYAD
          PUNCH 800,PHASAP,NWALL,XPP,YPP,RA,EMAG,ANG,THE,PHI
          PUNCH 202,FIX,DELFIX
          PUNCH 101,(S(I),I=1,N3)
99        FORMAT(1H1)
100       FORMAT(4I5,E10.3,L1)
101       FORMAT(8E10.3)
110       FORMAT(2X*WARNING--REORIENT CAVITY SUCH THAT A.LE.B*)
111       FORMAT(2X*WARNING--MATRIX SIZE EXCEEDS STORAGE ALLOCATION*)
112       FORMAT(2X*WARNING--DIMENSION SIZE EXCEEDED--*A4* TRUNCATED TO *
     1    I3*. TO FIX, INCREASE SIZE OF IN AND SX ARRAYS.*/)
113       FORMAT(2X*WARNING--DIMENSION SIZE EXCEEDED--*A4* TRUNCATED TO *
     1    I3*. TO FIX, INCREASE SIZE OF SY ARRAY.*/)
116       FORMAT(2X*WARNING--ERROR IN WIRE END POINTS*)
117       FORMAT(2X*WARNING--ALL OR PART OF THE WIRE IS OUTSIDE THE *
     1    *CAVITY, OR UNATTACHED WIRE HAS END POINT ON CAVITY WALL.*)
129       FORMAT(25X*WIRE CONNECTED TO LOWER (Z=0) WALL*///)
130       FORMAT(28X*WIRE NOT CONNECTED TO CAVITY WALLS*///)
131       FORMAT(25X*WIRE CONNECTED TO UPPER (Z=C) WALL*///)
132       FORMAT(25X*WIRE CONNECTED TO WALL AT BOTH ENDS*///)
200       FORMAT(5X*WIRE CURRENTS--I(Z)*//16X*Z*17X*REAL*13X*IMAGINARY*/)
201       FORMAT(2X,3E20.5)
202       FORMAT(L1,E10.3)
203       FORMAT(1X/5X*SHORT CIRCUIT APERTURE FIELDS*/35X*TOTAL*38X
```

```
     1   *EXTERIOR*/25X*REAL*14X*IMAGINARY*18X*REAL*14X*IMAGINARY*
     2   //10X*EN:*2E20.5,5X,2E20.5/10X*H1:*2E20.5,5X,2E20.5/10X
     3   *H2:*2E20.5,5X,2E20.5)
300      FORMAT(20X*APERTURE-EXCITED WIRE IN A RECTANGULAR CAVITY*//)
301      FORMAT(5X*BOX DIMENSIONS:  A = *E10.3,5X*B = *E10.3,5X
     1   *C = *E10.3,5X*LAMBDA = *E10.3/)
302      FORMAT(5X*WIRE SIZE AND LOCATION:*/10X*RADIUS = *E10.3,5X
     1   *ZL = *E10.3,5X*ZU = *E10.3,5X*XC = *E10.3,5X*YC = *E10.3/)
303      FORMAT(5X*NUMBER OF PULSES USED IN CURRENT EXPANSION:  N = *I3/)
304      FORMAT(5X*CONVERGENCE CRITERION, ETC.*/10X*MAXIMUM TERM FOR *
     1   *SUM 3:  NSM = *I4/10X*MAXIMUM TERM FOR BAD SUM:  *
     1   *MAX = *I4/10X*MAXIMUM TERM FOR GMI SUM:  *
     2   *MAXX = *I4/10X*CONVERGENCE RATIO:  CS = *E10.3/10X
     3   *REPETITION FACTOR:  LS = *I2/1CX*FINITE DIFFERENCE METHOD *
     4   *USED IN CALCULATING INCIDENT FIELD (T/F):  ADYAD = *L1/)
305      FORMAT(10X*NOTE--DIPOLE APPROXIMATIONS ACCOUNT FOR WALL RE*
     1   *FLECTIONS (FIX=.T.) WITH DELFIX=*E10.3/)
800      FORMAT(L1,I1,8E9.2)
         STOP
         END

      SUBROUTINE FILL(IN,SX,SY,D,S,Q,MS,DQ,W)
         DIMENSION IN(1),SX(1),SY(1),D(1),S(1),Q(MS,1)
     1   ,W(1)
         COMPLEX Q,CE
         REAL K
         COMMON/WN/K
         COMMON/FSUM/A,B,C,MAX
         COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
         DATA CE/(0.,-376.7)/
         NP=N+1
         NNP=NP+NP
         X=2.*ZL+HDEL
         CALL GETUM(D,DQ,X,NNP,IN,SX,SY)
         DO 2 I=2,NNP
         J=I-1
2        S(J)=D(I)-D(J)
         X=HDEL
         CALL GETUM(W,DQ,X,NP,IN,SX,SY)
         D(1)=2.*W(1)
         DO 1 I=2,NP
1        D(I)=W(I)-W(I-1)
         DZ=2.*D(2)-CKD*D(1)
         DO 3 I=1,N
         IP=I+1
```

```
      IJ=I+I
      Q(I,I)=(DZ+S(IJ-1)+S(IJ+1)-CKD*S(IJ))*CE
      IF(I.EQ.N)GOTO3
      DO 4 J=IP,N
      IJ=I+J
      JI=J-I
      Q(J,I)=(D(JI)+D(JI+2)+S(IJ-1)+S(IJ+1)
     1 -CKD*(D(JI+1)+S(IJ)))*CE
4     Q(I,J)=Q(J,I)
3     CONTINUE
      RETURN
      END

      SUBROUTINE FILL1(IN,SX,SY,S,Q,MS,DQ)
      DIMENSION IN(1),SX(1),SY(1),S(1),Q(MS,1)
      COMPLEX Q,CE
      REAL K
      COMMON/WN/K
      COMMON/FSUM/A,B,C,MAX
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      LOGICAL HI
      COMMON/GRD/IGRD
      DATA CE/(0.,-376.7)/
      HI=IGRD.EQ.1
      NN=N+N
      X=HDEL
      NNP=NN+2
      NP=N+1
      CALL GETUM(S,DQ,X,NN,IN,SX,SY)
      DO 1 I=2,NN
      J=NNP-I
1     S(J)=S(J)-S(J-1)
      S(1)=2.*S(1)
      M=1
      IF(HI)M=N
      DZ=2.*S(2)-CKD*S(1)
      Q(M,M)=CE*DZ
      DO 6 J=2,N
      JJ=J
      IF(HI)JJ=NP-J
      Q(JJ,M)=(S(J-1)+S(J+1)-CKD*S(J))*CE
6     Q(M,JJ)=Q(JJ,M)*2.
      DO 2 I=2,N
      II=I
      IF(HI)II=NP-I
```

```
      DO 2 J=I,N
      JJ=J
      IF(HI)JJ=NP-J
      JI=I+J
      IF(I.EQ.J)GOTO10
      IJ=J-I
      Q(II,JJ)=(S(IJ+2)+S(JI-2)+S(IJ)+S(JI)-CKD*(S(IJ+1)+S(JI-1)))*CE
      Q(JJ,II)=Q(II,JJ)
3     CONTINUE
2     CONTINUE
      RETURN
10    Q(II,II)=(DZ+S(JI-2)+S(JI)-CKD*S(JI-1))*CE
      GOTO 3
      END

     SUBROUTINE FILL2(IN,SX,SY,S,Q,MS,DQ)
      DIMENSION IN(1),SX(1),SY(1),S(1),Q(MS,1)
      COMPLEX Q,CE
      REAL K
      COMMON/WN/K
      COMMON/FSUM/A,B,C,MAX
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      LOGICAL HI
      COMMON/GRD/IGRD
      DATA CE/(0.,-376.7)/
      NP=N+1
      NNP=NP+2
      X=HDEL
      NM=N-1
      CALL GETUM(S,DQ,X,NP,IN,SX,SY)
      DO 1 I=2,NP
      J=NNP-I
1     S(J)=S(J)-S(J-1)
      S(1)=2.*S(1)
      DZ=2.*S(2)-CKD*S(1)
      Q(1,1)=CE*DZ
      Q(N,N)=Q(1,1)
      Q(1,N)=CE*(2.*S(NM)-CKD*S(N))
      Q(N,1)=Q(1,N)
      DO 6 J=2,NM
      JJ=NP-J
      Q(J,1)=(S(J-1)+S(J+1)-CKD*S(J))*CE
      Q(JJ,N)=Q(J,1)
      Q(1,J)=2.*Q(J,1)
6     Q(N,JJ)=Q(1,J)
```

```
      DO 2 I=2,N
      II=NP-I
      IF(I.GT.II) GOTO 4
      DO 2 J=I,II
      JJ=NP-J
      JI=I+J
      IF(I.EQ.J)GOTO 10
      IJ=J-I
      Q(I,J)=(S(IJ+2)+S(JI-2)+S(IJ)+S(JI)-CKD*(S(IJ+1)+S(JI-1)))*CE
      Q(II,JJ)=Q(I,J)
      Q(J,I)=Q(I,J)
      Q(JJ,II)=Q(I,J)
3     CONTINUE
2     CONTINUE
4     CONTINUE
      RETURN
10    Q(I,I)=(DZ+S(JI-2)+S(JI)-CKD*S(JI-1))*CE
      Q(II,II)=Q(I,I)
      GOTO 3
      END

    SUBROUTINE GETUM(W,DQ,ST,NP,IN,SX,SY)
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      LOGICAL CIND
      COMMON/YES/CIND
      DIMENSION W(1),IN(1),SX(1),SY(1)
      DATA PI/3.1415927/
      X=NP*DEL+ST
      NP2=NP+1
      CIND=.F.
      DO 1 I=1,NP
      J=NP2-I
      X=X-DEL
      CALL BADSUM(IN,SX,SY,X,SD)
      IF(CIND)GOTO 10
1     W(J)=SD-DQ
      RETURN
10    PRINT 11,J
11    FORMAT(10X*NOTE--W(I) INTERPOLATED THRU I=*I2/)
      NS=J+1
      X=X+DEL
      J1=NS+1
      X1=X+DEL
      IF(J1.GT.NP)GOTO21
      CD=.25/PI
```

```
      XN=X/R
      CC1=W(NS)-CO*ALOG(XN+SQRT(XN*XN+1.))
      XN=X1/R
      CC2=W(J1)-CO*ALOG(XN+SQRT(XN*XN+1.))
      B=(CC2-X1*CC1/X)/X1/DEL
      A=(CC1-B*X*X)/X
      X=ST-DEL
      DO 6 I=1,J
      X=X+DEL
      XN=X/R
6     W(I)=CO*ALOG(XN+SQRT(XN*XN+1.))+X*(A+B*X)
      RETURN
21    PRINT 31
      STOP
31    FORMAT(10X*NOT ENOUGH POINTS FOR INTERPOLATION.*)
      END

    SUBROUTINE MORFIL(Q,MS,FP,G1,G2,CON)
      DIMENSION Q(MS,1),FP(1),G1(1),G2(1),ALPM(2)
      COMPLEX Q,CC,CE
      COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
      COMMON/GRD/IGRD
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      COMMON/NEXTRA/N1,N2,N3
      DATA CE/(0.,-376.7)/
      CC=CE*CON
      DO 1 I=1,N
      Q(I,N1)=CON*ALPE*FP(I)
      Q(I,N2)=-CC*ALPM(1)*G1(I)
      Q(I,N3)=-CC*ALPM(2)*G2(I)
      Q(N1,I)=CE*DEL*FP(I)
      Q(N2,I)=DEL*G1(I)
1     Q(N3,I)=DEL*G2(I)
      DO 2 I=N1,N3
      DO 3 J=N1,N3
3     Q(I,J)=(0.,0.)
2     Q(I,I)=(1.,0.)
      IF(IGRD.EQ.0.O.IGRD.EQ.1)GOTO 5
      DO 4 I=N1,N3
4     Q(I,1)=Q(I,1)*.5
5     IF(IGRD.LT.1)RETURN
      DO 6 I=N1,N3
6     Q(I,N)=Q(I,N)*.5
      RETURN
      END
```

```
      SUBROUTINE INVERS (AM,M1,M2,NPTS,C,B)
C   SUBROUTINE INVERS INVERTS THE EQUATION AM(I,J) B(J) = C(I) AND RETURN
      COMPLEX AM,B,C,TEMP,SUM,BETA
      DIMENSION AM(M1,M2),C(M1),B(M2)
      IF (NPTS .GT. M1 .OR. NPTS .GT. M2) GO TO 1
      NM1 = NPTS - 1
      DO 690 KK = 1,NM1
      KKP1 = KK + 1
      L = KK
      DO 600 I = KKP1,NPTS
      D=REAL(AM(I,KK))
      P=AIMAG(AM(I,KK))
      S=REAL(AM(L,KK))
      T=AIMAG(AM(L,KK))
600   IF(    (D*D+P*P).GT.(S*S+T*T)   ) L = I
      IF (  L.EQ.KK  ) GO TO 620
      DO 610 J = KK,NPTS
      TEMP=AM(KK,J)
      AM(KK,J)=AM(L,J)
610   AM(L,J)=TEMP
      TEMP=C(KK)
      C(KK)=C(L)
      C(L)=TEMP
620   DO 690 I = KKP1,NPTS
      BETA = AM(I,KK) / AM(KK,KK)
      DO 650 J = KKP1,NPTS
650   AM(I,J) = AM(I,J) -BETA*AM(KK,J)
690   C(I) = C(I) -BETA*C(KK)
      B(NPTS) = C(NPTS) /AM(NPTS,NPTS)
      I = NM1
710   IP1 = I + 1
      SUM = (0.0,0.0)
      DO 700 J = IP1,NPTS
700   SUM = SUM + AM(I,J)*B(J)
      B(I) = (   C(I) - SUM   ) / AM(I,I)
      I = I - 1
      IF(  I.GE.1  ) GO TO 710
      RETURN
1     PRINT 10
10    FORMAT (//*  WARNING - MATRIX SIZE EXCEEDS STORAGE ALLOCATION*
    1 //)
      RETURN
      END
```

```
      SUBROUTINE FIXUP(Q,MS)
      REAL K
      COMPLEX Q,CHGA,CHGF(2),CE,CK,COK
      DIMENSION Q(MS,1),DGX(2),DGYX(2),ALPM(2)
      COMMON/WHYNOT/CHGA,CHGF,DGAX,DGAY,DGX,DGYX
      COMMON/WN/K
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
      COMMON/NEXTRA/N1,N2,N3
      DATA CE/(0.,-376.7)/
      CK=CE*K
      COK=K/CE
      Q(N1,N1)=1.+ALPE*CHGA
      Q(N1,N2)=CK*ALPM(1)*DGX(1)
      Q(N1,N3)=-CK*ALPM(2)*DGX(2)
      Q(N2,N1)=COK*ALPE*DGAY
      Q(N3,N1)=-COK*ALPE*DGAX
      Q(N2,N2)=1.-ALPM(1)*CHGF(1)
      Q(N3,N3)=1.-ALPM(2)*CHGF(2)
      Q(N2,N3)=-ALPM(2)*DGYX(2)
      Q(N3,N2)=-ALPM(1)*DGYX(1)
      RETURN
      END


      SUBROUTINE DPLFIX(DEL,IN,SX,SY,XPF,YPP,NWALL)
      COMPLEX CQ,CHGA,CHGF(2)
      REAL K,KK
      DIMENSION QX(6),QY(6),QA(6),QB(6),IN(1),SX(1),SY(1),XA(1)
     1 ,DGX(2),DGYX(2)
      COMMON/WN/K
      COMMON/FSUM/A,B,C,MAX
      COMMON/SUMS/KK,NSM,CONV,LC
      COMMON/WHYNOT/CHGA,CHGF,DGAX,DGAY,DGX,DGYX
      DATA PI/3.1415927/
      CM=.5/PI
      CN=CM/DEL
      CQ=CMPLX(0.,K/PI/3.)
      XKD=K*DEL
      CD=CN*COS(XKD)
      CDD=.5*CN*COS(2.*XKD)
      ZP=0.
      XP=XPP
      YP=YPP
      IF(NWALL-2)1,2,3
    1 CC=A
```

```
      AA=B
      BB=C
      GOTO 4
2     CC=B
      AA=C
      BB=A
      GOTO 4
3     CC=C
      AA=A
      BB=B
4     Z=ZP+DEL
      IXL=0
      IXU=0
      IYL=0
      IYU=0
      IF(XP+DEL.GE.AA)IXU=1
      IF(YP+DEL.GE.BB)IYU=1
      IF(YP-DEL.LE.0.)IYL=1
      IF(XP-DEL.LE.0.)IXL=1
      IX=IXU+IXL
      IY=IYU+IYL
      IF(IX.EQ.2.O.IY.EQ.2)GOTO 77
      ISUM=IX+IY
      IF(ISUM.EQ.0)GOTO 29
      IF(IXU.EQ.0)GOTO 21
      XS=AA+XP
      YS=YP
21    IF(IXL.EQ.0)GOTO 22
      XS=-XP
      YS=YP
22    IF(ISUM.EQ.2)GOTO 24
      IF(IYU.EQ.0)GOTO 23
      YS=BB+YP
      XS=XP
23    IF(IYL.EQ.0)GOTO 29
      YS=-YP
      XS=XP
      GOTO 29
24    IF(IYU.EQ.0)GOTO 25
      YS1=YP+BB
      GOTO 29
25    YS1=-YP
29    CONTINUE
      CALL GA(IN,SX,SY,XP,XP,YP,YP,Z,ZP,AA,BB,CC,SA)
      SA=SA-CD
      Z1=Z+DEL
      CALL GA(IN,SX,SY,XP,XP,YP,YP,Z1,ZP,AA,BB,CC,SB)
```

```
      SB=SB-CDD
      IF(ISUM.EQ.0)GOTO 26
      ISE=-1
      X=XP
      Y=YP
      X1=X
      Y1=Y
      GOTO 70
26    DGNN=2.*(SB-SA)/DEL/DEL/3.
      GAA=(4.*SA-SB)/9.
      D1=DEL
      X=XP+DEL
      IF(X.LT.AA)GOTO 5
      D1=AA-XP
      X=AA
      SA=0.
      GOTO 6
5     CALL GA(IN,SX,SY,X,XP,YP,YP,ZP,ZP,AA,BB,CC,SA)
6     SA=SA-CM*COS(K*D1)/D1
      D2=DEL
      X1=XP-DEL
      IF(X1.GT.0.)GOTO 7
      D2=XP
      X1=0.
      SB=0.
      GOTO 8
7     CALL GA(IN,SX,SY,X1,XP,YP,YP,ZP,ZP,AA,BB,CC,SB)
8     SB=SB-CM*COS(K*D2)/D2
      IF(ISUM.EQ.0)GOTO 27
      ISE=0
      Z=ZP
      Z1=Z
      GOTO 70
27    DGAX=(SA-SB)/(D1+D2)
      GAA=GAA+(SA+SB)/6.
      D1=DEL
      Y=YP+DEL
      IF(Y.LT.BB)GOTO 9
      Y=BB
      D1=BB-YP
      SA=0.
      GOTO10
9     CALLGA(IN,SX,SY,XP,XP,Y,YP,ZP,ZP,AA,BB,CC,SA)
10    SA=SA-CM*COS(K*D1)/D1
      D2=DEL
      Y1=YP-DEL
.     IF(Y1.GT.0.)GOTO 11
```

```
      D2=YP
      Y1=0.
      SB=0.
      GOTO 12
11    CALLGA(IN,SX,SY,XP,XP,Y1,YP,ZP,ZP,AA,BB,CC,SB)
12    SB=SB-CM*COS(K*D2)/D2
      IF(ISUM.EQ.0)GOTO 28
      ISE=1
      X=XP
      X1=X
      GOTO 70
28    DGAY=(SA-SB)/(D1+D2)
      GAA=GAA+(SA+SB)/6.
      IF(ISUM.NE.0)GOTO 72
79    CHGA=DGNN+KK*(GAA+CQ)
      DO 49 IQD=1,2
      CD=.5
      SD=.86602540378444
      SS=CD
      CS=SD
      DO 13 I=1,3
      DY=CS*DEL
      DX=SS*DEL
      DYY=DY*DY
      Y=YP+DY
      IF(Y.GT.BB)GOTO 14
      IF(Y.LT.0.)GOTO 15
16    X=XP+DX
      DA=DX
      IF(X.LT.AA)GOTO 17
      DA=AA-XP
      SA=0.
      GOTO 18
17    CALL GF(IN,SX,SY,X,XP,Y,YP,ZP,ZP,AA,BB,CC,SA)
18    DAA=DA*DA
      D1=SQRT(DAA+DYY)
      SA=SA-CM*COS(K*D1)/D1
      QX(I)=DA
      QY(I)=DY
      QB(I)=SA
      X=XP-DX
      DA=-DX
      IF(X.GT.0.)GOTO 19
      DA=-XP
      SA=0.
      GOTO 20
19    CALL GF(IN,SX,SY,X,XP,Y,YP,ZP,ZP,AA,BB,CC,SA)
```

```
20    DAA=DA*DA
      D1=SQRT(DAA+DYY)
      SA=SA-CM*COS(K*D1)/D1
      J=I+3
      QX(J)=DA
      QY(J)=DY
      QB(J)=SA
      SSS=SS
      SS=SS*CD+CS*SD
13    CS=CS*CD-SSS*SD
      ISUM=IXU+IXL+IYU+IYL
      IF(ISUM.EQ.0)GOTO 91
      FO=0.
      DFY=0.
      DFXX=0.
      DFXY=0.
      DO 50 I=1,6
50    QA(I)=0.
      IF(IQD.EQ.2)GOTO 75
      DX=AA-XP
      IF(DX.GE.XP)IXL=1
      IF(DX.LE.XP)IXU=1
      DX=BB-YP
      IF(DX.GE.YP)IYL=1
      IF(DX.LE.YP)IYU=1
75    IF(IXU.EQ.0)GOTO 92
      ISE=1
      XS=2.*AA-XP
      GOTO 30
92    IF(IXL.EQ.0)GOTO 93
      ISE=0
      XS=-XP
      GOTO 30
93    IF(IYU.EQ.0)GOTO 94
      ISE=1
      YS=2.*BB-YP
      GOTO 40
94    IF(IYL.EQ.0)GOTO 95
      ISE=0
      YS=-YP
      GOTO 40
95    IF(IXU+IYU.NE.2)GOTO 96
      ISE=-1
      XS=2.*AA-XP
      YS=2.*BB-YP
      GOTO 60
96    IF(IXU+IYL.NE.2)GOTO 97
```

```
      ISE=0
      XS=2.*AA-XP
      YS=-YP
      GOTO 60
97    IF(IXL+IYU.NE.2)GOTO 98
      ISE=1
      XS=-XP
      YS=2.*BB-YP
      GOTO60
98    IF(IXL+IYL.NE.2)GOTO 99
      ISE=2
      XS=-XP
      YS=-YP
      GOTO 60
99    DO 51 I=1,6
51    QB(I)=QB(I)-QA(I)
91    D1=QX(1)-QX(4)
      D2=QX(2)-QX(5)
      D3=QX(3)-QX(6)
      D4=QY(1)-QY(3)
      G7=(-QX(4)*QB(1)+QX(1)*QB(4))/D1
      G8=(-QX(6)*QB(3)+QX(3)*QB(6))/D3
      GO=.5*((-QX(5)*QB(2)+QX(2)*QB(5))/D2+(-QY(3)*G7+QY(1)*G8)/D4)
      DGY=(G7-G8)/D4
      DGXY=((QB(1)-QB(4))/D1 - (QB(3)-QB(6))/D3)/D4
      DGXX=2.*((QB(2)/QX(2)-QB(5)/QX(5))/D2+GO/QX(2)/QX(5))
      IF(ISUM.EQ.0)GOTO 53
      DO 52 I=1,6
52    QB(I)=QB(I)+QA(I)
      GO=GO+FO
      DGY=DGY+DFY
      DGXX=DGXX+DFXX
      DGXY=DGXY+DFXY
53    CHGF(IQD)=DGXX+KK*(GO+CQ)
      DGX(IQD)=DGY
      DGYX(IQD)=DGXY
      XP=YPP
      YP=XPP
      IT=IXU
      IXU=IYU
      IYU=IT
      IT=IXL
      IXL=IYL
      IYL=IT
      SSS=AA
      AA=BB
49    BB=SSS
```

```
      RETURN
14    Y=BB
      DY=Y-YP
      DX=DEL
      DYY=DY*DY
      GOTO 16
15    Y=0.
      DY=-YP
      DX=DEL
      DYY=DY*DY
      GOTO 16
30    DO 31 I=1,6
      DX=XP+QX(I)-XS
      DY=QY(I)
      R=SQRT(DX*DX+DY*DY)
31    QA(I)=QA(I)-CM/R
      R=ABS(XP-XS)
      R3=R*R*R
      FO=FO-CM/R
      DFXX=DFXX-2.*CM/R3
      IF(ISE)93,93,92
40    DO 41 I=1,6
      DY=YP+QY(I)-YS
      DX=QX(I)
      R=SQRT(DX*DX+DY*DY)
41    QA(I)=QA(I)+CM/R
      DY=YP-YS
      R=ABS(DY)
      R3=R*R*R
      FO=FO+CM/R
      DFY=DFY-CM*DY/R3
      DFXX=DFXX-CM/R3
      IF(ISE)95,95,94
60    DO 61 I=1,6
      DX=XP+QX(I)-XS
      DY=YP+QY(I)-YS
      R=SQRT(DX*DX+DY*DY)
61    QA(I)=QA(I)-CM/R
      DX=XP-XS
      DY=YP-YS
      DXX=DX*DX
      RR=DXX+DY*DY
      R=SQRT(RR)
      R3=R*RR
      R5=R3*RR
      FO=FO-CM/R
      DFY=DFY+CM*DY/R3
```

```
      DFXY=DFXY-3.*CM*DX*DY/R5
      DFXX=DFXX+CM*(1./R3-3.*DXX/R5)
      IF(ISE)96,97,62
62    IF(ISE.EQ.1)GOTO 98
      GOTO 99
77    PRINT 78
      STOP
78    FORMAT(1X*WARNING--CAVITY DIMENSION TOO SMALL FOR DPLFIX*)
70    DX=X-XS
      DY=Y-YS
      ZZ=Z*Z
      ZZ1=Z1*Z1
      R=SQRT(ZZ+DX*DX+DY*DY)
      SA=SA+CM/R
      DX=X1-XS
      DY=Y1-YS
      R=SQRT(ZZ1+DX*DX+DY*DY)
      SB=SB+CM/R
      IF(ISUM.EQ.1)GOTO 71
      DX=X-XP
      DY=Y-YS1
      R=SQRT(ZZ+DX*DX+DY*DY)
      SA=SA+CM/R
      DX=X1-XP
      DY=Y1-YS1
      R=SQRT(ZZ1+DX*DX+DY*DY)
      SB=SB+CM/R
      DX=X1-XS
      R=SQRT(ZZ1+DX*DX+DY*DY)
      SB=SB-CM/R
      DX=X-XS
      DY=Y-YS1
      R=SQRT(ZZ+DX*DX+DY*DY)
      SA=SA-CM/R
71    IF(ISE)26,27,28
72    DX=XP-XS
      DY=YP-YS
      RR=DX*DX+DY*DY
      R=SQRT(RR)
      R3=R*RR
      GAA=GAA-CM/R
      DGAX=DGAX+CM*DX/R3
      DGAY=DGAY+CM*DY/R3
      DGNN=DGNN+CM/R3
      IF(ISUM.EQ.1)GOTO 79
      DY=YP-YS1
      RR=DY*DY
```

```
      RR1=RR+DX*DX
      R=SQRT(RR1)
      R3=R*RR1
      GAA=GAA-CM/DY+CM/R
      DGAX=DGAX-CM*DX/R3
      DGAY=DGAY+CM/RR-CM*DY/R3
      DGNN=DGNN-CM/R3+CM/RR/DY
      GOTO 79
      END

      SUBROUTINE EAP(IN,SX,SY,E,N,FP,G1,G2,CON)
      LOGICAL PHASAP
      COMPLEX E,C2,CE
      DIMENSION E(1),FP(1),G1(1),G2(1),IN(1),SX(1),SY(1)
      DIMENSION LN(3),LT(2,3),XA(2),RA(2),ALPM(2),EC(3),HC(3)
      DIMENSION AN(3)
      COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
      COMMON/GRD/IGRD
      COMMON/LAST/XA,NWAL
      COMMON/LAST2/RA,EMAG,ANG,THE,PHI,PHASAP
      COMMON/NEXTRA/N1,N2,N3
      DATA LN/1,2,3/
      DATA (LT(I),I=1,6)/2,3,3,1,1,2/
      DATA AN/2HXX,2HYY,2HZZ/
      DATA CE/(0.,-376.7)/
      N1=N+1
      N2=N+2
      N3=N+3
      NWALL=NWAL
      WHERE=8HORIGIN
      IF(PHASAP)WHERE=8HAPERTURE
      I1=LT(1,NWALL)
      I2=LT(2,NWALL)
      RX=0.
      RY=0.
      RZ=0.
      NIN=NWALL-2
      IF(NIN)1,2,3
    1 XP=0.
      YP=XA(1)
      ZP=XA(2)
      RY=RA(1)
      RZ=RA(2)
      GOTO 4
    2 YP=0.
```

```
      XP=XA(2)
      ZP=XA(1)
      RX=RA(2)
      RZ=RA(1)
      GOTO 4
3     ZP=0.
      XP=XA(1)
      YP=XA(2)
      RX=RA(1)
      RY=RA(2)
4     CALL ALPHA (RA,ALPE,ALPM)
      PRINT 101,NWALL,XP,YP,ZP,RX,RY,RZ
      PRINT 103,ALPE,AN(I1),ALPM(1),AN(I2),ALPM(2)
      PRINT 102,THE,PHI,EMAG,ANG,WHERE
      J=LN(NWALL)
      CALL EDIPL(J,FP,IN,SX,SY,G1,G2,CON)
      DO 5 I=1,2
      J=LT(I,NWALL)
      IF(I.EQ.2) GOTO 6
      CALL MDIPL(J,G1,IN,SX,SY)
      GOTO 5
6     CALL MDIPL(J,G2,IN,SX,SY)
5     CONTINUE
      CALL INC(EMAG,ANG,THE,PHI,ET,EP,HT,HP,PHASE)
      CALL CONV(EC,ET,EP)
      CALL CONV(HC,HT,HP)
      J=LN(NWALL)
      C2=(1.,0.)
      IF(PHASAP)GOTO 8
      C2=C2*CEXP((0.,1.)*PHASE)
8     DO 7 I=1,N
7     E(I)=(0.,0.)
      E(N1)=2.*EC(J)*C2
      E(N2)=2.*HC(I1)*C2
      E(N3)=2.*HC(I2)*C2
      RETURN
101   FORMAT(5X*APERTURE LOCATION AND SIZE  (WALL*I2*)*/
     1 10X*XA = *E10.3,5X*YA = *E10.3,5X*ZA = *E10.3/
     2 10X*RX = *E10.3,5X*RY = *E10.3,5X*RZ = *E10.3/)
102   FORMAT(5X*INCIDENT PLANE WAVE IMPINGING FROM (THETA,*
     1 *PHI) = (*E10.3*,*E10.3*)*/10X*WITH E-FIELD MAGNITUDE = *
     2 E10.3/10X*AND E-FIELD POLARIZATION ANGLE = *E10.3/10X
     3 *PHASE OF INCIDENT FIELD REFERENCED TO *A8//)
103   FORMAT(5X*APERTURE POLARIZABILITIES:*/10X*ALPHA-E = *
     1 E10.3,5X*ALPHA-M*A2* = *E10.3,5X*ALPHA-M*A2* = *E10.3/)
      END
```

```
      SUBROUTINE INC(E,A,T,P,ET,EP,HT,HP,PH)
        DIMENSION ALPM(2)
        COMMON/WN/K
        COMMON/OVER/ST,CT,SP,CP
        COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
        REAL K
        DATA PI/3.1415927/
        RAD=PI/180.
        RA=RAD*A
        RT=RAD*T
        RP=RAD*P
        H=E/376.7
        ET=-E*COS(RA)
        EP=E*SIN(RA)
        HT=H*SIN(RA)
        HP=H*COS(RA)
        ST=SIN(RT)
        CT=COS(RT)
        SP=SIN(RP)
        CP=COS(RP)
        PH=K*(XP*ST*CP+YP*ST*SP+ZP*CT)
        RETURN
        END

      SUBROUTINE CONV(E,ET,EP)
        DIMENSION E(1)
        COMMON/OVER/ST,CT,SP,CP
        E(1)=ET*CT*CP-EP*SP
        E(2)=ET*CT*SP+EP*CP
        E(3)=-ET*ST
        RETURN
        END

      SUBROUTINE ALPHA(R,E,A)
        REAL L
        DIMENSION R(1),A(1)
        DATA PI/3.1415927/
        IF(R(2)-R(1))1,20,2
1       L=R(1)
        W=R(2)
        IBIG=1
        ISM=2
        GOTO 3
```

```
2     L=R(2)
      W=R(1)
      IBIG=2
      ISM=1
3     X=(L-W)/L
      IF(X.LT.1.E-3)GOTO 20
      RAT=W/L
      RR=RAT*RAT
      EE=1.-RR
      CALL ELIPTIC(EE,FK,FE)
      C=PI*L*L*L/3.
      CC=C*EE
      A(IBIG)=CC/(FK-FE)
      A(ISM)=CC/(FE/RR-FK)
      E=C*RR/FE
      RETURN
20    L=R(1)
      E=2.*L*L*L/3.
      A(2)=2.*E
      A(1)=A(2)
      RETURN
      END

     SUBROUTINE ELIPTIC(X,EK,E)
      DIMENSION A(4),B(4),C(4),D(4)
      DATA A/.09666344259,.03590092383,.03742563713,.01451196212/
      DATA B/.12498593597,.06880248576,.03328355346,.00441787012/
      DATA C/.44325141463,.06260601220,.04757383546,.01736506451/
      DATA D/.24998368310,.09200180037,.04069697526,.00526449639/
      Y=1.-X
      Z=ALOG(1./Y)
      W=1.
      P=1.38629436112
      Q=.5
      R=1.
      S=0.
      DO 1 I=1,4
      W=W*Y
      P=P+A(I)*W
      Q=Q+B(I)*W
      R=R+C(I)*W
1     S=S+D(I)*W
      EK=P+Q*Z
      E=R+S*Z
```

```
      RETURN
      END

      SUBROUTINE MDIPL(J,E,IN,SX,SY)
      DIMENSION E(1),IN(1),SX(1),SY(1),ALPM(2)
      REAL K
      COMMON/WN/K
      COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      COMMON/FSUM/A,B,C,MAX
      COMMON/GRD/IGRD
      SD=0.
      Z=ZL
      IF(IGRD.EQ.-1.OR.IGRD.EQ.2)Z=Z-DEL
      XO=XC+R
      DO 1 I=1,N
      Z=Z+DEL
      IF(J.EQ.3)GOTO 1
      IF(J.EQ.2)GOTO 2
      CALL GM(IN,SX,SY,XO,XP,YC,YP,Z,ZP,A,B,C,SD)
      GOTO1
2     CALL GM(IN,SX,SY,YC,YP,XO,XP,Z,ZP,B,A,C,SD)
      SD=-SD
1     E(I)=SD
      RETURN
      END

      SUBROUTINE EDIPL(J,E,IN,SX,SY,DUM,BUM,CON)
      DIMENSION E(1),IN(1),SX(1),SY(1),DUM(1),BUM(1),ALPM(2)
      REAL K
      LOGICAL GEZ,ADYAD
      COMMON/TYPE/ADYAD
      COMMON/SPEC/GEZ
      COMMON/WN/K
      COMMON/DIPOLE/XP,YP,ZP,ALPE,ALPM,NWALL
      COMMON/FORFILL/N,DEL,HDEL,ZL,XC,YC,R,CKD
      COMMON/FSUM/A,B,C,MAX
      COMMON/GRD/IGRD
      Z=ZL
      IF(IGRD.EQ.-1.OR.IGRD.EQ.2)Z=Z-DEL
      CC=1./K
      XO=XC+R
      IF(J.EQ.3)GOTO 30
```

```
      IF(ADYAD)GOTO 20
      DO 1 I=1,N
      Z=Z+DEL
      IF(J.EQ.1)CALL  GE(IN,SX,SY,XD,XP,YC,YP,Z,ZP,A,B,C,SD)
      IF(J.EQ.2)CALL  GE(IN,SX,SY,YC,YP,XD,XP,Z,ZP,B,A,C,SD)
1     E(I)=CC*SD
      RETURN
20    XD=XD+HDEL
      XM=XD-DEL
      YD=YC+HDEL
      YM=YD-DEL
      NP=N+1
      Z=ZL-HDEL
      CC=CC/DEL/DEL
      DO 2 I=1,NP
      Z=Z+DEL
      IF(J.EQ.2)GOTO 21
      CALL  GA(IN,SX,SY,YC,YP,Z,ZP,XD,XP,B,C,A,SD)
      CALL  GA(IN,SX,SY,YC,YP,Z,ZP,XM,XP,B,C,A,SM)
      GOTO 22
21    CALL  GA(IN,SX,SY,Z,ZP,XD,XP,YD,YP,C,A,B,SD)
      CALL  GA(IN,SX,SY,Z,ZP,XD,XP,YM,YP,C,A,B,SM)
22    DUM(I)=SD
2     BUM(I)=SM
      DO 3 I=1,N
      IP=I+1
3     E(I)=CC*(DUM(IP)-BUM(IP)-DUM(I)+BUM(I))
      RETURN
25    NP=N+2
      Z=ZL-DEL
      CC=K/CON
      DO 4 I=1,NP
      Z=Z+DEL
      CALL  GA(IN,SX,SY,XD,XP,YC,YP,Z,ZP,A,B,C,SD)
4     DUM(I)=SD
      DO 5 I=1,N
5     E(I)=CC*(DUM(I)+DUM(I+2)-CKD*DUM(I+1))
      RETURN
30    IF(ADYAD)GOTO 25
      GEZ=.T.
      DO 6 I=1,N
      Z=Z+DEL
      CALL  GA(IN,SX,SY,XD,XP,YC,YP,Z,ZP,A,B,C,SD)
6     E(I)=CC*SD
      GEZ=.F.
      RETURN
      END
```

```
      FUNCTION S3(Z,ZP,W,WP,C,D)
        COMMON/PEEK/I
        COMMON/SUMS/KK,N,CONV,LC
        REAL KK,KW,KC,KWW
        DATA PI/3.1415927/
        X=PI/D
        KW=0.
        ZL=AMIN1(Z,ZP)
        ZG=C-AMAX1(Z,ZP)
        ARG=X*(C-ZG-ZL)
        CX=COSH(ARG)
        SUM=
     1  .125*ALOG((CX-COS(X*(W+WP)))/(CX-COS(X*(W-WP))))/X
        L=0
        DO 1 I=1,N
        KW=KW+X
        KWW=KW*KW
        SKC=KK-KWW
        IF(KWW.GT.KK)GOTO 2
        KC=SQRT(SKC)
        T=SIN (KC*ZL)*SIN (KC*ZG)/SIN (KC*C)/KC
        GOTO 3
    2   KC=SQRT(-SKC)
        T=SINH(KC*ZL)*SINH(KC*ZG)/SINH(KC*C)/KC
    3   TT=SIN(KW*W)*SIN(KW*WP)*(T-.5*EXP(-I*ARG)/KW)
        SUM=SUM+TT
        L=L+1
        AB=ABS(TT/SUM)
        IF(AB.GT.CONV)L=0
        IF(L.EQ.LC) GOTO 4
    1   CONTINUE
        PRINT 5 ,AB,L
        STOP
    4   S3=SUM*2./D
    5   FORMAT(10X*WARNING--SUM 3 NOT CONVERGED*E15.5I10)
        RETURN
        END


      SUBROUTINE GM(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)
        DIMENSION IN(1),SX(1),SY(1)
        COMMON /NEUMAN/ MO,NO
        COMMON /GM/ MAX
        LGM=-1
        IF(Y.GT.YP)GOTO 5
        AS=Y
```

```
      SIG=-1.
      AC=B-YP
      GOTO 6
5     AS=B-Y
      AC=YP
      SIG=1.
      GOTO 6
    ENTRY GA
      LGM=0
      GOTO 6
    ENTRY GF
      LGM=1
6     NO=1
      MO=NO
      DX=X-XP
      DY=Y-YP
      DZ=Z-ZP
      RA=1./A/A
      RB=1./B/B
      RC=1./C/C
      ALP=(RB+RC)*DX*DX
      BET=(RA+RC)*DY*DY
      GAM=(RA+RB)*DZ*DZ
      IF(ALP.GT.BET)GOTO 10
      IF(BET.GT.GAM)GOTO 2
      GOTO 3
10    IF(ALP.LE.GAM)GOTO 3
1     IF(B.GT.C)GOTO 4
      NAX=C*MAX/B
      NG=0
      IF(LGM.EQ.1)MO=0
      CALL NDE(SX,Y,YP,B,MAX,LGM)
      CALL NCC(SY,Z,ZP,C,NAX)
      CALL GM1(IN,SX,SY,X,XP,B,C,A,SD)
      RETURN
4     NAX=B*MAX/C
      MO=0
      IF(LGM.EQ.1)NO=0
      CALL NDE(SY,Y,YP,B,NAX,LGM)
      CALL NCC(SX,Z,ZP,C,MAX)
      CALL GM1(IN,SX,SY,X,XP,C,B,A,SD)
      RETURN
2     IF(A.GT.C)GOTO 7
      NAX=C*MAX/A
      NO=0
      CALL NSS(SX,X,XP,A,MAX)
      CALL NCC(SY,Z,ZP,C,NAX)
```

```
      IF(LGM)12,11,21
11    CALL GM1(IN,SX,SY,Y,YP,A,C,B,SD)
      GOTO 13
21    CALL GM3(IN,SX,SY,Y,YP,A,C,B,SD)
      GOTO 13
12    CALL GM2(IN,SX,SY,AS,AC,A,C,B,SD)
      SD=SIG*SD
13    RETURN
7     NAX=A*MAX/C
      MO=0
      CALL NSS(SY,X,XP,A,NAX)
      CALL NCC(SX,Z,ZP,C,MAX)
      IF(LGM)15,14,22
14    CALL GM1(IN,SX,SY,Y,YP,C,A,B,SD)
      GOTO 13
22    CALL GM3(IN,SX,SY,Y,YP,C,A,B,SD)
      GOTO 13
15    CALL GM2(IN,SX,SY,AS,AC,C,A,B,SD)
      SD=SIG*SD
      RETURN
3     IF(A.GT.B)GOTO 8
      NAX=B*MAX/A
      IF(LGM.EQ.1)NO=0
      CALL NSS(SX,X,XP,A,MAX)
      CALL NDE(SY,Y,YP,B,NAX,LGM)
      CALL GM3(IN,SX,SY,Z,ZP,A,B,C,SD)
      RETURN
8     NAX=A*MAX/B
      IF(LGM.EQ.0)MO=0
      CALL NSS(SY,X,XP,A,NAX)
      CALL NDE(SX,Y,YP,B,MAX,LGM)
      CALL GM3(IN,SX,SY,Z,ZP,B,A,C,SD)
      RETURN
      END

      SUBROUTINE GE(IN,SX,SY,X,XP,Y,YP,Z,ZP,A,B,C,SD)
      DIMENSION IN(1),SX(1),SY(1)
      COMMON /NEUMAN/ MO,NO
      COMMON /GM/ MAX
      NO=1
      MO=1
      SIG=1.
      DX=X-XP
      DY=Y-YP
      DZ=Z-ZP
```

```
      RA=1./A/A
      RB=1./B/B
      RC=1./C/C
      ALP=(RB+RC)*DX*DX
      BET=(RA+RC)*DY*DY
      GAM=(RA+RB)*DZ*DZ
      IF(ALP.GT.BET)GOTO 10
      IF(BET.GT.GAM)GOTO 2
      GOTO 3
10    IF(ALP.LE.GAM)GOTO 3
1     IF(X.GT.XP)GOTO 12
      AC=A-XP
      AS=X
      SIG=-1.
      GOTO 11
12    AC=XP
      AS=A-X
11    IF(B.GT.C)GOTO 4
      NAX=C*MAX/B
      CALL NSS(SX,Y,YP,B,MAX)
      CALL NSC(SY,ZP,Z,C,NAX)
      CALL GM2(IN,SX,SY,AS,AC,B,C,A,SD)
      SD=SIG*SD
      GOTO 6
4     NAX=B*MAX/C
      CALL NSS(SY,Y,YP,B,NAX)
      CALL NSC(SX,ZP,Z,C,MAX)
      CALL GM2(IN,SX,SY,AS,AC,C,B,A,SD)
      SD=SIG*SD
      GOTO 6
2     IF(A.GT.C)GOTO 7
      NAX=C*MAX/A
      CALL NSC(SX,X,XP,A,MAX)
      CALL NSC(SY,ZP,Z,C,NAX)
      CALL GM1(IN,SX,SY,Y,YP,A,C,B,SD)
      GOTO 6
7     NAX=A*MAX/C
      CALL NSC(SY,X,XP,A,NAX)
      CALL NSC(SX,ZP,Z,C,MAX)
      CALL GM1(IN,SX,SY,Y,YP,C,A,B,SD)
      GOTO 6
3     IF(Z.GT.ZP)GOTO 5
      AC=Z
      AS=C-ZP
      GOTO 13
5     AC=C-Z
      AS=ZP
```

```
      SIG=-1.
13    IF(A.GT.B)GOTO 8
      NAX=B*MAX/A
      CALL NSC(SX,X,XP,A,MAX)
      CALL NSS(SY,Y,YP,B,NAX)
      CALL GM2(IN,SX,SY,AS,AC,A,B,C,SD)
      SD=SIG*SD
      GOTO 6
8     NAX=A*MAX/B
      CALL NSC(SY,X,XP,A,NAX)
      CALL NSS(SX,Y,YP,B,MAX)
      CALL GM2(IN,SX,SY,AS,AC,B,A,C,SD)
6     SD=-SD
      RETURN
      END

      SUBROUTINE BADSUM(IN,SX,SY,X,SD)
      REAL K,KK,KX
      DIMENSION IN(1),SX(1),PX(1),SY(1)
      LOGICAL CIND
      COMMON/YES/CIND
      COMMON/WN/K
      COMMON /PEEK/M
      COMMON/FSUM/A,B,C,MAX
      COMMON/SUMS/KK,NSM,CONV,LC
      DATA PI/3.1415927/
      AR=PI/A
      BR=PI/B
      L=0
      SBR=BR*BR
      CW=C-X
      DO 10 M=1,MAX
10    IN(M)=0
      BSUM=0.
      DO 1 M=1,MAX
      KX=M*AR
      SKX=KX*KX
      RR=SKX+SBR
      IN(M)=1
      SUM=0.
      I=M
      N=1
      GOTO 9
2     I=I-1
      IF(I.EQ.0)GOTO7
```

```
      KX=I*AR
      SKX=KX*KX
      NF=IN(I)+1
      IN(I)=SQRT(RR-SKX)/BR
      NL=IN(I)
      IF(NF.GT.NL)GOTO 2
      DO 3 N=NF,NL
      GOTO 9
6     CONTINUE
3     CONTINUE
      GOTO 2
7     BSUM=BSUM+SUM
      L=L+1
      AB=ABS(SUM/BSUM)
      IF(AB.GT.CONV)L=0
      IF(L.EQ.LC)GOTO 4
1     CONTINUE
      CIND=.T.
      RETURN
9     SKY=N*N*SBR
      SGC=SKY+SKX-KK
      IF(SGC.GT.0.)GOTO11
      GC=SQRT(-SGC)
      T=SIN(GC*CW)/SIN(GC*C)
      GOTO 12
11    GC=SQRT(SGC)
      T=RSINH(GC*CW,GC*C)
12    TT=T*SX(I)*SY(N)/SGC
      SUM=SUM+TT
      IF(I.EQ.M)GOTO 2
      GOTO 6
4     SD=-BSUM*2./A/B
      RETURN
      END

     SUBROUTINE GM1(IN,SX,SY,Z,ZP,A,B,C,SD)
      REAL K,KK,KX,KXX
      LOGICAL GEZ
      DIMENSION IN(1),SX(1),SY(1)
      COMMON/WN/K
      COMMON /PEEK/M
      COMMON/NEUMAN/MO,NO
      COMMON/GM/MAX
      COMMON/SUMS/KK,NSM,CONV,LC
      COMMON/SPEC/GEZ
```

```
      DATA PI/3.1415927/
      IND=1
      GOTO 55
     ENTRY GM3
      IND=3
55    ZL=AMIN1(Z,ZP)
      ZG=C-AMAX1(Z,ZP)
      GOTO 8
     ENTRY GM2
      IND=2
      AC=ZP
      AS=Z
8     INDM=IND-2
      AR=PI/A
      BR=PI/B
      L=0
      SBR=BR*BR
      DO 10 M=1,MAX
10    IN(M)=0
      BSUM=0.
      KXX=MO*AR
      NN=1-NO
      DO 1 M=1,MAX
      KX=KXX
      SKX=KX*KX
      RR=SKX+NO*SBR
      IN(M)=NO
      SUM=0.
      I=M
      N=NO
      GOTO 9
2     I=I-1
      IF(I.EQ.0)GOTO7
      KX=KX-AR
      SKX=KX*KX
      NF=IN(I)+1
      IN(I)=SQRT(RR-SKX)/BR
      NL=IN(I)
      IF(NF.GT.NL)GOTO 2
      DO 3 N=NF,NL
      GOTO 9
6     CONTINUE
3      CONTINUE
      GOTO 2
7     BSUM=BSUM+SUM
      L=L+1
      AB=ABS(SUM/BSUM)
```

```
      IF(AB.GT.CONV)L=0
      IF(L.EQ.LC)GOTO 4
1     KXX=KXX+AR
      PRINT 5,IND,AB,L
      STOP
9     SKY=N*N*SBR
      SKC=SKY+SKX
      SGC=SKC-KK
      IF(SGC.GT.0.)GOTO11
      GC=SQRT(-SGC)
      IF(INDM)13,14,15
13    T=SIN(GC*ZL)*SIN(GC*ZG)/SIN(GC*C)/GC
      GOTO 12
14    T=COS(GC*AC)*SIN(GC*AS)/SIN(GC*C)
      GOTO 12
15    T=-COS(GC*ZL)*COS(GC*ZG)/SIN(GC*C)/GC
      IF(GEZ)T=T*SKC
      GOTO 12
11    GC=SQRT(SGC)
      IF(INDM)16,17,18
16    T=RSS(GC*ZL,GC*ZG,GC*C)/GC
      GOTO 12
17    T=RCS(GC*AC,GC*AS,GC*C)
      GOTO 12
18    T=RCC(GC*ZL,GC*ZG,GC*C)/GC
      IF(GEZ)T=T*SKC
12    TT=T*SX(I)*SY(N+NN)
      IF(N.EQ.0) TT=.5*TT
      IF(MO+I.EQ.1) TT=.5*TT
      SUM=SUM+TT
      IF(I.EQ.M)GOTO 2
      GOTO 6
4     SD=4.*BSUM/A/B
5     FORMAT(10X*WARNING--GM*I1* SUM NOT CONVERGED*E15.5I10)
      RETURN
      END

     SUBROUTINE NSS(D,X,XP,A,M)
      DIMENSION D(1)
      DATA PI/3.1415927/
      AR=PI/A
      W=0.
      DO 1 I=1,M
      W=W+AR
1     D(I)=SIN(W*X)*SIN(W*XP)
```

```
      RETURN
      END

      SUBROUTINE NCC(D,Z,ZP,C,M)
      REAL KK
      LOGICAL GEZ
      COMMON/SUMS/KK,NSM,CONV,LC
      COMMON/SPEC/GEZ
      DIMENSION D(1)
      DATA PI/3.1415927/
      AR=PI/C
      D(1)=1.
      W=0.
      DO 1 I=1,M
      W=W+AR
      IP=I+1
      D(IP)=COS(W*Z)*COS(W*ZP)
      IF(GEZ) D(IP)=D(IP)*(KK-W*W)
    1 CONTINUE
      IF(GEZ) D(1)=KK
      RETURN
      END

      SUBROUTINE NSC(D,YS,YC,B,M)
      DIMENSION D(1)
      DATA PI/3.1415927/
      AR=PI/B
      W=0.
      DO 1 I=1,M
      W=W+AR
    1 D(I)=W*SIN(W*YS)*COS(W*YC)
      RETURN
      END

      SUBROUTINE NDE(D,Y,YP,B,M,L)
      DIMENSION D(1)
      IF(L)1,2,3
    1 CALL NSC(D,Y,YP,B,M)
      RETURN
    2 CALL NSS(D,Y,YP,B,M)
      RETURN
```

```
3     CALL NCC(D,Y,YP,B,M)
      RETURN
      END

      FUNCTION COSH(X)
      Y=EXP(X)
      COSH=.5*(Y+1./Y)
      RETURN
      END

      FUNCTION SINH(X)
      Y=EXP(X)
      SINH=.5*(Y-1./Y)
      RETURN
      END

      FUNCTION RSS(X,Y,Z)
      RSS=.5*(EXP(X+Y-Z)-EXP(X-Y-Z)-EXP(Y-X-Z)+EXP(-X-Y-Z))
     1 /(1.-EXP(-Z-Z))
      RETURN
      END

      FUNCTION RCC(X,Y,Z)
      RCC=.5*(EXP(X+Y-Z)+EXP(X-Y-Z)+EXP(Y-X-Z)+EXP(-X-Y-Z))
     1 /(1.-EXP(-Z-Z))
      RETURN
      END

      FUNCTION RCS(X,Y,Z)
      RCS=.5*(EXP(X+Y-Z)-EXP(X-Y-Z)+EXP(Y-X-Z)-EXP(-X-Y-Z))
     1 /(1.-EXP(-Z-Z))
      RETURN
      END
```

```
      FUNCTION RSINH(X,Y)
        RSINH=(EXP(X-Y)-EXP(-X-Y))/(1.-EXP(-Y-Y))
        RETURN
        END
```