" Characterization of Waveguide Longitudinal Slots Covered by a Dielectric Layer"

LIST OF PROGRAMS

Pisti B. Katehi

April 1989

```
Here is an index of your tape which was written:

99 records of "arrange_mutual.ftn" written to tape file #1.
     44 records of "data_g2_diel" written to tape file #2.
     53 records of "data_wave_k0" written to tape file #3.
     60 records of "data_wave_mutual" written to tape file #4.
    771 records of "g2_diel.ftn" written to tape file #5.
    356 records of "generate_k0.ftn" written to tape file #6.
    177 records of "inv_wave_k0.ftn" written to tape file #7.
    144 records of "main_wave_k0.ftn" written to tape file #8.
    193 records of "mutual_slot.ftn" written to tape file #9.
    370 records of "out_g2_diel" written to tape file #10.
    445 records of "out_wave_k0" written to tape file #11.
    367 records of "out_wave_mutual" written to tape file #12.
    190 records of "poles.ftn" written to tape file #13.
    191 records of "poles_mutual.ftn" written to tape file #14.
      2 records of "run_ericsson" written to tape file #15.
      1 record of "run_ericsson.bak" written to tape file #16.
      3 records of "run_k0" written to tape file #17.
      2 records of "run_k0.bak" written to tape file #18.
      1 record of "run_mutual" written to tape file #19.
    838 records of "slot_design.ftn" written to tape file #20.
   2047 records of "yij_diel_k0.ftn" written to tape file #21.
   2084 records of "yij_diel_mutual.ftn" written to tape file #22.
    271 records of "yij_wave_k0.ftn" written to tape file #23.
    367 records of "yij_wave_mutual.ftn" written to tape file #24.
```

PROGRAM I

This program evaluates the coupling term G in accordance to Elliott's definition in its paper " An Improved Design Procedure for Small Arrays of Shunt Slots", IEEE Trans. on Antennas and Propagation, Vol. AP-31, No. 1, Jan. 1983.

The files which consist this program are:

RUN_ERICSSON:	This program links all the subroutines.
DATA_G2_DIEL:	Input File
OUT_G2_DIEL:	Output File
G2_DIEL.FTN :	Main Program Subroutine G2_DIEL Subroutine F_EER Subroutine NORM Subroutine CUBSPL
POLES_MUTUAL.FTN :	Subroutine SPOLES
YIJ_DIEL_MUTUAL.FTN:	Subroutine YIJ_DIEL Subroutine LIMIT Subroutine GREEN Function GXXM Function GZXM Function HZXE

Subroutine FUNCT

Subroutine GREI

Subroutine ARIS

Subroutine ADONIS

Subroutine BESS1

Subroutine TAIL

Subroutine BESS2

Subroutine BSJ0

Subroutine F

Subroutine DATA_SLOT

ARRANGE_MUTUAL.FTN: Subroutine ARRANGE_MUTUAL

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

```
rrrrr   u    u  n    n          eeeeee  rrrrr     i    cccc    ssss    ssss    oooo   n    n
r    r  u    u  nn   n          e       r    r    i   c    c  s       s       o    o  nn   n
r    r  u    u  n n  n          eeeee   r    r    i   c        ssss    ssss   o    o  n n  n
rrrrr   u    u  n  n n          e       rrrrr     i   c            s       s  o    o  n  n n
r   r   u    u  n   nn          e       r   r     i   c    c  s    s  s    s  o    o  n   nn
r    r   uuuu   n    n  ______  eeeeee  r    r    i    cccc    ssss    ssss    oooo   n    n
```

//tera/users/katehi/tape/run_ericsson

LAST MODIFIED ON: 89/04/24 10:54 AM
FILE PRINTED: 89/04/24 11:00 AM

```
BIND G2_DIEL.BIN MUTUAL_SLOT_ERICSSON.BIN POLES_MUTUAL.BIN
     YIJ_DIEL_MUTUAL.BIN  ARRANGE_MUTUAL.BIN -B SLOT_ERICSSON
```

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

```
                                                        22222
ddddd      aa     ttttt     aa                 gggg   2     2         ddddd      i    eeeeee  l
d    d    a  a      t      a  a               g    g        2         d    d     i    e       l
d    d   a    a     t     a    a              g         22222         d    d     i    eeeee   l
d    d   aaaaaa     t     aaaaaa              g  ggg  2               d    d     i    e       l
d    d   a    a     t     a    a              g    g  2               d    d     i    e       l
ddddd    a    a     t     a    a  _______      gggg   2222222 _______ ddddd      i    eeeeee  llllll
```

//tera/users/katehi/tape/data_g2_diel

```
C
C     ---- Dielectric constant ---
C
      2.62
C
C     ---- Substrate Thickness ---
C
      0.050
C
C     ---- Conductor Thickness ---
C
      0.00001
C
C     ---- Dimensions of the Waveguide ----
C
      0.6858
      0.3048
C
C     ---- Half lengths of the slots
C
      0.17595
      0.16595
C
C     ----  Transverse offsets of the slots ----
C
      0.24765
      0.43815
C
C     ---- Longitudinal offets of the slot ----
C
      0.73053
C
C     ---- Slot widths ----
C
      0.047625
      0.047625
C
C     ---- Lower Limit of the Tail Contribution ----
C
      100.0
C
C     ---- Error in the evaluation of the series ----
C
      1.D-6
```

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

```
                                      22222
 oooo   u    u   ttttt          gggg  2     2           ddddd     i     eeeeee  l
o    o  u    u     t           g    g       2           d    d    i     e       l
o    o  u    u     t           g         22222          d    d    i     eeeee   l
o    o  u    u     t           g   ggg  2               d    d    i     e       l
o    o  u    u     t           g    g   2               d    d    i     e       l
 oooo    uuuu      t   _______  gggg  2222222  _______  ddddd     i     eeeeee  llllll
```

//tera/users/katehi/tape/out_g2_diel

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


          Substrate Thickness
           0.5000000E-01


          Conductor Thickness
           0.1000000E-04


          Dimensions of the Waveguide
          AW= 0.6858000E+00
          BW= 0.3048000E+00

  Half lengths of the slots

          SLOT_L1= 0.1759500E+00
          SLOT_L2= 0.1659500E+00




  Longitudinal offset of the slots

          SLOT_DX= 0.7305300E+00



          Slot Widths
          SLOT_W1= 0.4762500E-01
          SLOT_W2= 0.4762500E-01



          Lower Limit of Tail Contribution
           0.1000000E+03


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


          Normalization Constant
           0.1000000E+01

          L=   1  RCUR= 0.0000000E+00
          L=   2  RCUR= 0.9801714E-01
          L=   3  RCUR= 0.1950903E+00
          L=   4  RCUR= 0.2902847E+00
          L=   5  RCUR= 0.3826834E+00
          L=   6  RCUR= 0.4713967E+00
          L=   7  RCUR= 0.5555702E+00
          L=   8  RCUR= 0.6343933E+00
          L=   9  RCUR= 0.7071068E+00
          L=  10  RCUR= 0.7730104E+00
          L=  11  RCUR= 0.8314696E+00
          L=  12  RCUR= 0.8819213E+00
          L=  13  RCUR= 0.9238795E+00
          L=  14  RCUR= 0.9569404E+00
          L=  15  RCUR= 0.9807853E+00
          L=  16  RCUR= 0.9951847E+00
          L=  17  RCUR= 0.1000000E+01
```

```
          L=  18  RCUR= 0.9951847E+00
          L=  19  RCUR= 0.9807853E+00
          L=  20  RCUR= 0.9569404E+00
          L=  21  RCUR= 0.9238795E+00
          L=  22  RCUR= 0.8819213E+00
          L=  23  RCUR= 0.8314696E+00
          L=  24  RCUR= 0.7730104E+00
          L=  25  RCUR= 0.7071068E+00
          L=  26  RCUR= 0.6343933E+00
          L=  27  RCUR= 0.5555702E+00
          L=  28  RCUR= 0.4713967E+00
          L=  29  RCUR= 0.3826834E+00
          L=  30  RCUR= 0.2902847E+00
          L=  31  RCUR= 0.1950903E+00
          L=  32  RCUR= 0.9801714E-01
          L=  33  RCUR=-0.4102069E-09
  L=   1  AICUR= 0.0000000E+00
  L=   2  AICUR= 0.0000000E+00
  L=   3  AICUR= 0.0000000E+00
  L=   4  AICUR= 0.0000000E+00
  L=   5  AICUR= 0.0000000E+00
  L=   6  AICUR= 0.0000000E+00
  L=   7  AICUR= 0.0000000E+00
  L=   8  AICUR= 0.0000000E+00
  L=   9  AICUR= 0.0000000E+00
  L=  10  AICUR= 0.0000000E+00
  L=  11  AICUR= 0.0000000E+00
  L=  12  AICUR= 0.0000000E+00
  L=  13  AICUR= 0.0000000E+00
  L=  14  AICUR= 0.0000000E+00
  L=  15  AICUR= 0.0000000E+00
  L=  16  AICUR= 0.0000000E+00
  L=  17  AICUR= 0.0000000E+00
  L=  18  AICUR= 0.0000000E+00
  L=  19  AICUR= 0.0000000E+00
  L=  20  AICUR= 0.0000000E+00
  L=  21  AICUR= 0.0000000E+00
  L=  22  AICUR= 0.0000000E+00
  L=  23  AICUR= 0.0000000E+00
  L=  24  AICUR= 0.0000000E+00
  L=  25  AICUR= 0.0000000E+00
  L=  26  AICUR= 0.0000000E+00
  L=  27  AICUR= 0.0000000E+00
  L=  28  AICUR= 0.0000000E+00
  L=  29  AICUR= 0.0000000E+00
  L=  30  AICUR= 0.0000000E+00
  L=  31  AICUR= 0.0000000E+00
  L=  32  AICUR= 0.0000000E+00
  L=  33  AICUR= 0.0000000E+00
    N_SLOT=   1  L=   1  CUR= 0.0000000E+00    0.0000000E+00
    N_SLOT=   1  L=   2  CUR= 0.9801714E-01    0.0000000E+00
    N_SLOT=   1  L=   3  CUR= 0.1950903E+00    0.0000000E+00
    N_SLOT=   1  L=   4  CUR= 0.2902847E+00    0.0000000E+00
    N_SLOT=   1  L=   5  CUR= 0.3826834E+00    0.0000000E+00
    N_SLOT=   1  L=   6  CUR= 0.4713967E+00    0.0000000E+00
    N_SLOT=   1  L=   7  CUR= 0.5555702E+00    0.0000000E+00
    N_SLOT=   1  L=   8  CUR= 0.6343933E+00    0.0000000E+00
    N_SLOT=   1  L=   9  CUR= 0.7071068E+00    0.0000000E+00
    N_SLOT=   1  L=  10  CUR= 0.7730104E+00    0.0000000E+00
    N_SLOT=   1  L=  11  CUR= 0.8314696E+00    0.0000000E+00
    N_SLOT=   1  L=  12  CUR= 0.8819213E+00    0.0000000E+00
    N_SLOT=   1  L=  13  CUR= 0.9238795E+00    0.0000000E+00
    N_SLOT=   1  L=  14  CUR= 0.9569404E+00    0.0000000E+00
    N_SLOT=   1  L=  15  CUR= 0.9807853E+00    0.0000000E+00
    N_SLOT=   1  L=  16  CUR= 0.9951847E+00    0.0000000E+00
    N_SLOT=   1  L=  17  CUR= 0.1000000E+01    0.0000000E+00
```

```
    N_SLOT=   1  L=  18  CUR= 0.9951847E+00   0.0000000E+00
    N_SLOT=   1  L=  19  CUR= 0.9807853E+00   0.0000000E+00
    N_SLOT=   1  L=  20  CUR= 0.9569404E+00   0.0000000E+00
    N_SLOT=   1  L=  21  CUR= 0.9238795E+00   0.0000000E+00
    N_SLOT=   1  L=  22  CUR= 0.8819213E+00   0.0000000E+00
    N_SLOT=   1  L=  23  CUR= 0.8314696E+00   0.0000000E+00
    N_SLOT=   1  L=  24  CUR= 0.7730104E+00   0.0000000E+00
    N_SLOT=   1  L=  25  CUR= 0.7071068E+00   0.0000000E+00
    N_SLOT=   1  L=  26  CUR= 0.6343933E+00   0.0000000E+00
    N_SLOT=   1  L=  27  CUR= 0.5555702E+00   0.0000000E+00
    N_SLOT=   1  L=  28  CUR= 0.4713967E+00   0.0000000E+00
    N_SLOT=   1  L=  29  CUR= 0.3826834E+00   0.0000000E+00
    N_SLOT=   1  L=  30  CUR= 0.2902847E+00   0.0000000E+00
    N_SLOT=   1  L=  31  CUR= 0.1950903E+00   0.0000000E+00
    N_SLOT=   1  L=  32  CUR= 0.9801714E-01   0.0000000E+00
    N_SLOT=   1  L=  33  CUR=-0.4102069E-09   0.0000000E+00
 L=   1  RCUR(L)= 0.0000000E+00  AICUR(L)= 0.0000000E+00
 L=   2  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
 L=   3  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
 L=   4  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
 L=   5  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
 L=   6  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
 L=   7  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
 L=   8  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
 L=   9  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
 L=  10  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
 L=  11  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
 L=  12  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
 L=  13  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
 L=  14  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
 L=  15  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
 L=  16  RCUR(L)= 0.1000000E+01  AICUR(L)= 0.0000000E+00
 L=  17  RCUR(L)= 0.9945219E+00  AICUR(L)= 0.0000000E+00
 L=  18  RCUR(L)= 0.9781476E+00  AICUR(L)= 0.0000000E+00
 L=  19  RCUR(L)= 0.9510565E+00  AICUR(L)= 0.0000000E+00
 L=  20  RCUR(L)= 0.9135454E+00  AICUR(L)= 0.0000000E+00
 L=  21  RCUR(L)= 0.8660254E+00  AICUR(L)= 0.0000000E+00
 L=  22  RCUR(L)= 0.8090170E+00  AICUR(L)= 0.0000000E+00
 L=  23  RCUR(L)= 0.7431448E+00  AICUR(L)= 0.0000000E+00
 L=  24  RCUR(L)= 0.6691306E+00  AICUR(L)= 0.0000000E+00
 L=  25  RCUR(L)= 0.5877852E+00  AICUR(L)= 0.0000000E+00
 L=  26  RCUR(L)= 0.5000000E+00  AICUR(L)= 0.0000000E+00
 L=  27  RCUR(L)= 0.4067366E+00  AICUR(L)= 0.0000000E+00
 L=  28  RCUR(L)= 0.3090170E+00  AICUR(L)= 0.0000000E+00
 L=  29  RCUR(L)= 0.2079117E+00  AICUR(L)= 0.0000000E+00
 L=  30  RCUR(L)= 0.1045285E+00  AICUR(L)= 0.0000000E+00
 L=  31  RCUR(L)= 0.7932658E-12  AICUR(L)= 0.0000000E+00


       Number of elements to be evaluated for the mutual interactions

       I=   1  J=   2  NSSL=  99 0.1905000E+00


       Offsets for the dielectric layer and number of corresponding elements

       I=   1  OFFSET= 0.1905000E+00  NOFFS=  99
       I=   2  OFFSET= 0.0000000E+00  NOFFS=   0


       SLOTS and corresponding offsets in the dielectric

       I=   1  J=   2  INSS=   1  OFFSET= 0.1905000E+00
```

```
Max number of offsets in the dielectric
NOFF=    1




No TE waves excited in the substrate




There are    1 TM waves excited in the substrate

   1    0.640756827E+01


Contribution to admittance from the dielectric



OFFSET #    1


Interactions between slots 1 and  2

IJ=    1 YSD=-0.1336066E-05   0.2025418E-05
IJ=    2 YSD=-0.1335278E-05   0.1999338E-05
IJ=    3 YSD=-0.1332916E-05   0.1923216E-05
IJ=    4 YSD=-0.1328984E-05   0.1803263E-05
IJ=    5 YSD=-0.1323492E-05   0.1649357E-05
IJ=    6 YSD=-0.1316454E-05   0.1474246E-05
IJ=    7 YSD=-0.1307886E-05   0.1292181E-05
IJ=    8 YSD=-0.1297808E-05   0.1116486E-05
IJ=    9 YSD=-0.1286245E-05   0.9561593E-06
IJ=   10 YSD=-0.1273222E-05   0.8126619E-06
IJ=   11 YSD=-0.1258771E-05   0.6794269E-06
IJ=   12 YSD=-0.1242926E-05   0.5463025E-06
IJ=   13 YSD=-0.1225724E-05   0.4085598E-06
IJ=   14 YSD=-0.1207205E-05   0.2753354E-06
IJ=   15 YSD=-0.1187414E-05   0.1698209E-06
IJ=   16 YSD=-0.1166393E-05   0.1167152E-06
IJ=   17 YSD=-0.1144194E-05   0.1220969E-06
IJ=   18 YSD=-0.1120868E-05   0.1606052E-06
IJ=   19 YSD=-0.1096468E-05   0.1842461E-06
IJ=   20 YSD=-0.1071049E-05   0.1532349E-06
IJ=   21 YSD=-0.1044671E-05   0.6887217E-07
IJ=   22 YSD=-0.1017393E-05  -0.1981857E-07
IJ=   23 YSD=-0.9892756E-06  -0.4837193E-07
IJ=   24 YSD=-0.9603837E-06   0.1134345E-07
IJ=   25 YSD=-0.9307815E-06   0.1213307E-06
IJ=   26 YSD=-0.9005344E-06   0.2001468E-06
```

```
IJ=  27 YSD=-0.8697093E-06   0.1864835E-06
IJ=  28 YSD=-0.8383743E-06   0.9305222E-07
IJ=  29 YSD=-0.8065973E-06   0.1381068E-08
IJ=  30 YSD=-0.7744471E-06  -0.4461526E-08
IJ=  31 YSD=-0.7419927E-06   0.8831194E-07
IJ=  32 YSD=-0.7093030E-06   0.2060933E-06
IJ=  33 YSD=-0.6764468E-06   0.2505578E-06
IJ=  34 YSD=-0.6434921E-06   0.1858709E-06
IJ=  35 YSD=-0.6105077E-06   0.7272820E-07
IJ=  36 YSD=-0.5775607E-06   0.1523836E-07
IJ=  37 YSD=-0.5447172E-06   0.6629375E-07
IJ=  38 YSD=-0.5120475E-06   0.1782646E-06
IJ=  39 YSD=-0.4796019E-06   0.2464069E-06
IJ=  40 YSD=-0.4474570E-06   0.2064180E-06
IJ=  41 YSD=-0.4156614E-06   0.9473410E-07
IJ=  42 YSD=-0.3842918E-06   0.1393801E-07
IJ=  43 YSD=-0.3533924E-06   0.3517073E-07
IJ=  44 YSD=-0.3230218E-06   0.1307341E-06
IJ=  45 YSD=-0.2932330E-06   0.2017249E-06
IJ=  46 YSD=-0.2640776E-06   0.1740779E-06
IJ=  47 YSD=-0.2356037E-06   0.6977564E-07
IJ=  48 YSD=-0.2078578E-06  -0.1595174E-07
IJ=  49 YSD=-0.1808828E-06  -0.8506504E-08
IJ=  50 YSD=-0.1547202E-06   0.7367623E-07
IJ=  51 YSD=-0.1294074E-06   0.1391508E-06
IJ=  52 YSD=-0.1049797E-06   0.1150288E-06
IJ=  53 YSD=-0.8146906E-07   0.1856938E-07
IJ=  54 YSD=-0.5890498E-07  -0.6189339E-07
IJ=  55 YSD=-0.3731316E-07  -0.5629693E-07
IJ=  56 YSD=-0.1671714E-07   0.1789738E-07
IJ=  57 YSD= 0.2863389E-08   0.7537130E-07
IJ=  58 YSD= 0.2141152E-07   0.5016955E-07
IJ=  59 YSD= 0.3891360E-07  -0.3854086E-07
IJ=  60 YSD= 0.5535861E-07  -0.1079326E-06
IJ=  61 YSD= 0.7073939E-07  -0.9671453E-07
IJ=  62 YSD= 0.8505119E-07  -0.2648051E-07
IJ=  63 YSD= 0.9829265E-07   0.2246270E-07
IJ=  64 YSD= 0.1104652E-06  -0.5730499E-08
IJ=  65 YSD= 0.1215734E-06  -0.8626091E-07
IJ=  66 YSD= 0.1316245E-06  -0.1411647E-06
IJ=  67 YSD= 0.1406285E-06  -0.1203939E-06
IJ=  68 YSD= 0.1485986E-06  -0.5222273E-07
IJ=  69 YSD= 0.1555499E-06  -0.1206536E-07
IJ=  70 YSD= 0.1615008E-06  -0.4372259E-07
IJ=  71 YSD= 0.1664716E-06  -0.1152164E-06
IJ=  72 YSD= 0.1704851E-06  -0.1538414E-06
IJ=  73 YSD= 0.1735664E-06  -0.1222995E-06
IJ=  74 YSD= 0.1757428E-06  -0.5623792E-07
IJ=  75 YSD= 0.1770434E-06  -0.2534352E-07
IJ=  76 YSD= 0.1774991E-06  -0.6022003E-07
IJ=  77 YSD= 0.1771428E-06  -0.1217431E-06
IJ=  78 YSD= 0.1760088E-06  -0.1436529E-06
IJ=  79 YSD= 0.1741329E-06  -0.1022115E-06
IJ=  80 YSD= 0.1715522E-06  -0.3976521E-07
IJ=  81 YSD= 0.1683049E-06  -0.1881904E-07
IJ=  82 YSD= 0.1644305E-06  -0.5628800E-07
IJ=  83 YSD= 0.1599688E-06  -0.1071111E-06
IJ=  84 YSD= 0.1549609E-06  -0.1130817E-06
IJ=  85 YSD= 0.1494484E-06  -0.6415928E-07
IJ=  86 YSD= 0.1434729E-06  -0.7725475E-08
IJ=  87 YSD= 0.1370767E-06   0.2612182E-08
IJ=  88 YSD= 0.1303021E-06  -0.3658249E-07
IJ=  89 YSD= 0.1231914E-06  -0.7633645E-07
IJ=  90 YSD= 0.1157868E-06  -0.6819261E-07
IJ=  91 YSD= 0.1081300E-06  -0.1523171E-07
IJ=  92 YSD= 0.1002625E-06   0.3248329E-07
```

```
 IJ=   93 YSD= 0.9222526E-07   0.3186113E-07
 IJ=   94 YSD= 0.8405846E-07  -0.7976951E-08
 IJ=   95 YSD= 0.7580152E-07  -0.3671732E-07
 IJ=   96 YSD= 0.6749293E-07  -0.1710458E-07
 IJ=   97 YSD= 0.5917015E-07   0.3596153E-07
 IJ=   98 YSD= 0.5086946E-07   0.7247149E-07
 IJ=   99 YSD= 0.4262614E-07   0.6110565E-07
          Interactions between slots 2 and  2

          NOEL1=  33
          NOEL2=  31
          NS12=  66

          DIST= 0.7301800E+00  G_SLOT= 0.1971653E+00   0.3682245E+00  CMC_EXT=-0.2111288E+00

          NOEL1=  33
          NOEL2=  31
          NS12=  67

          DIST= 0.7412433E+00  G_SLOT= 0.2134013E+00   0.4087589E+00  CMC_EXT=-0.2343699E+00

          G_SLOT= 0.1976789E+00   0.3695068E+00
          C_MC=-0.2118640E+00   0.1133431E+00
```

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

```
       22222
 gggg  2     2          ddddd      i     eeeeee  l              ffffff   ttttt  n    n
g    g       2          d    d     i     e       l              f          t    nn   n
g        22222          d    d     i     eeeee   l              fffff      t    n n  n
g  ggg  2               d    d     i     e       l         ...  f          t    n  n n
g    g  2               d    d     i     e       l         ...  f          t    n   nn
 gggg   2222222 _______ ddddd      i     eeeeee  llllll    ...  f          t    n    n
```

//tera/users/katehi/tape/g2_diel.ftn

```
C*****************************************************************************
C     This program calls  G2_DIEL.FTN
C*****************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C..................................................................
      OPEN(UNIT=05,FILE='DATA_G2_DIEL',STATUS='OLD')
      OPEN(UNIT=06,FILE='OUT_G2_DIEL',STATUS='OLD')
      OPEN(UNIT=07,FILE='PLOT_G2_DIEL',STATUS='APPEND')
C.................................................................
C     Read the values of the geometrical parameters
C
      PI=3.141592653589D0
C
C
C     ---- Dielectric constant ---
      READ (5,1) DIEL_ER
  1   FORMAT (///6X,D16.9)
      WRITE (6,2) DIEL_ER
  2   FORMAT(10X,'Dielectric Constant of the Substrate'/10X,E14.7//)
C
C     ---- Substrate Thickness ---
C
      READ (5,1) DIEL_H
      WRITE (6,3) DIEL_H
  3   FORMAT(10X,'Substrate Thickness'/10X,E14.7//)
C
C     ---- Conductor Thickness ---
C
      READ (5,1) T
      WRITE (6,4) T
  4   FORMAT(10X,'Conductor Thickness'/10X,E14.7//)
C
C     ---- Dimensions of the Waveguide ----
C
      READ (5,1) AW
      READ (5,10) BW
 10   FORMAT(6X,D16.9)
      WRITE (6,5) AW,BW
  5   FORMAT(10X,'Dimensions of the Waveguide'/10X,'AW=',E14.7/
     *10X,'BW=',E14.7//)
C
C     ---- Limit for offsets: Small Offset< OFFLIM ----
C                             Large Offset> OFFLIM
C
      OFFLIM=0.1
C
C     ---- Half lengths of the slots
C
      READ(5,1) SLOT_L1
      READ(5,10) SLOT_L2
      WRITE (6,29) SLOT_L1,SLOT_L2
 29   FORMAT(2X,'Half lengths of the slots'//
     *10X,'SLOT_L1=',E14.7/10X,'SLOT_L2=',E14.7//)
C
C     ---- Transverse Offsets of the Slots in the Waveguide ----
C
      READ(5,1) SLOT_Y1
      READ(5,10) SLOT_Y2
  7   FORMAT(10X,'Transverse Offsets of the Slots'/10X,
     *'SLOT_Y1=',E14.7/10X,'SLOT_Y2=',E14.7)
      WRITE(6,60)
```

```
C
C     ---- Longitudinal Offsets of the Slots ----
C
      READ(5,1) SLOT_DX
      WRITE (6,11) SLOT_DX
   11 FORMAT(2X,'Longitudinal offset of the slots'//
     *10X,'SLOT_DX=',E14.7)
      WRITE(6,60)
   60 FORMAT(10X,//)
C
C     ---- Slot Widths ----
C
      READ(5,1) SLOT_W1
      WRITE(6,14) SLOT_W1
   14 FORMAT(10X,'Slot Widths'/10X,'SLOT_W1=',E14.7)
      READ(5,10) SLOT_W2
      WRITE (6,16) SLOT_W2
   16 FORMAT(10X,'SLOT_W2=',E14.7)
      WRITE (6,60)
C
C
C     ---- Lower Limit of the Tail Contribution ----
C
      READ (5,1) A
      WRITE (6,22) A
   22 FORMAT(10X,'Lower Limit of Tail Contribution'/10X,E14.7//)
C
C     ---- Error in the evaluation of the series ----
C
      READ (5,1) ERROR
      WRITE (6,27) ERROR
   27 FORMAT(10X,'Error in the evaluation of the series'/
     *10X,'ERROR=',E14.7//)
C-----------------------------------------------------------------
C     Initialize OFFSET( ) to 0
C
      DO 37 I=1,7
            OFFSET(I)=0.D0
   37 CONTINUE
C
C     Initialize NOFF to 1
C
      NOFF=1
C
      IS_OFF=1
C
C---------------------------------------------------------------
      CALL G2_DIEL(DIEL_ER,DIEL_H,SLOT_L1,SLOT_L2,SLOT_W1,
     *SLOT_W2,SLOT_DX,SLOT_Y1,SLOT_Y2,SLOT_DRX,SLOT_DRY,IS_OFF,G_SLOT)
C
      WRITE (6,100) DIEL_ER,DIEL_H,SLOT_L1,SLOT_L2,SLOT_W1,
     *SLOT_W2,SLOT_DX,SLOT_DY,G_SLOT
  100 FORMAT(//'*****************************************'/
     *2X,'ER=',E14.7,4X,'H=',E14.7//7X,'SLOT_1',14X,'SLOT_2'/
     *2X,'L1=',E14.7,2X,'L2=',E14.7/2X,'W1=',E14.7,2X,'W2=',
     *E14.7//10X,'Xoffset=',E14.7/10X,'Yoffset=',E14.7///10X,
     *'Mutual Coupling=',E14.7//)
      STOP
      END
C*****************************************************************************************
C                              The name of this program is:
C                                    G2_DIEL
C*****************************************************************************************
C      Calculates the mutual coupling between two longitudinal slots
C      All dimensions are normalized with respect to free space wavelength
C*****************************************************************************************
```

```
C      Subroutines and functions needed:
C
C                              YIJ_DIEL_MUTUAL
C                              POLES_MUTUAL
C                              ARRANGE_MUTUAL
C
C
C     **********INPUT*************
C     PI
C     DIEL_ER
C     DIEL_H
C     SLOT_L1
C     SLOT_L2
C     SLOT_W1
C     SLOT_W2
C     SLOT_DX
C     SLOT_DY
C
C     IS_OFF=1  Evaluate Mutual coupling between two slots spaced
C               half a waveguide wavelength apart
C     IS_OFF=2  Evaluate mutual coupling vs. slot distance
C
C     **********OUTPUT************
C
C     G_SLOT
C
C
C     COMMENTS: All lengths are in wavelengths in free space
C
C*****************************************************************************************
      SUBROUTINE G2_DIEL(DIEL_ER,DIEL_H,SLOT_L1,SLOT_L2,SLOT_W1,
     *SLOT_W2,SLOT_DX,SLOT_Y1,SLOT_Y2,SLOT_DRX,SLOT_DRY,IS_OFF,G_SLOT)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 RCUR,AICUR,CINC,ABS_CF
      REAL*4 RG_SLOT1,RG_SLOT2,RCMC_EXT1,RCMC_EXT2
      REAL*4 AIG_SLOT1,AIG_SLOT2,AICMC_EXT1,AICMC_EXT2
      COMPLEX YS,YS1S2,CI,SUM_MD,CUR_SLOT
      COMPLEX YS_ADM,CONSTN,CONSTM,Z12_MD
      COMPLEX Y12_MD,CF,G_SLOT,CMC_EXT
      EXTERNAL F_EER
C.................................................................
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
      COMMON/MAN/IBMATR(260,260)
C
      COMMON/WAY_OUT/RS10(7,7,200),XS10(7,7,200),SGMN(7,7,200),
     *RIJ(7,7,200)
C
      COMMON/B01/BJ0,BJ1
C
      COMMON/RES/CUR_SLOT(30,60),DLX_SLOT(30)
C
C
      PI=3.141592653589D0
C
C
C     ICUR=1    we assume a form for the resonant field
```

```
C               distribution
      ICUR=1
C
C     NSLOTS=2  we evaluate the mutual coupling between two slots
C
      NSLOTS=2
C
C     NSL     includes also the end points
C
      NSL(1)=31
      NSL(2)=31
C
      DLX_SLOT(1)=2.D0*SLOT_L1/(NSL(1)-1)
      DLX_SLOT(2)=2.D0*SLOT_L2/(NSL(2)-1)
C
      ER=DIEL_ER
      H=DIEL_H
      CALL F_EER
      CI=(0.0,1.0)
C
      YOFF(1)=SLOT_Y1
      YOFF(2)=SLOT_Y2
C
      WS(1)=SLOT_W1
      WS(2)=SLOT_W2
      WSDELTA(1)=0.0
      WSDELTA(2)=0.0
C
C
C.............................................................
C
C     Find minimum subsection length
C
      DLX=DLX_SLOT(1)
      IF (DLX_SLOT(2).LT.DLX) THEN
          DLX=DLX_SLOT(2)
      END IF
C
      NXOFF(1)=0
      NXOFF(2)=SLOT_DX/DLX+1
C
C     Interpolate the current on the slots
C
      DO 1 N_SLOT=1,NSLOTS
           DLX_DIF=DABS(DLX_SLOT(N_SLOT)-DLX)
           IF (DLX_DIF.GT.1.D-5) THEN
                CALL CUBSPL(ICUR,DLX,1,N_SLOT,1)
                CALL CUBSPL(ICUR,DLX,1,N_SLOT,2)
                DLX_SLOT(N_SLOT)=DLX
                L_MAX=NSL(N_SLOT)
                DO 7 L=1,L_MAX
                     CUR_SLOT(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
                     WRITE (6,77) N_SLOT,L,CUR_SLOT(N_SLOT,L)
 77                  FORMAT(5X,'N_SLOT=',I4,2X,'L=',I4,2X,'CUR=',
     *               E14.7,2X,E14.7)
  7             CONTINUE
           ELSE
                L_MAX=NSL(N_SLOT)
                DO 502 L=1,L_MAX
                       RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                       AICUR(L)=0.0
                       WRITE (6,601) L,RCUR(L),AICUR(L)
 601                   FORMAT(2X,'L=',I4,2X,'RCUR(L)=',E14.7,2X,
     *                 'AICUR(L)=',E14.7)
 502            CONTINUE
                DO 505 L=1,L_MAX
```

```
                     CUR_SLOT(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
 505           CONTINUE
            END IF
  1    CONTINUE
C
C
C      Call MUTUAL_SLOT to find external mutual coupling between the
C      two slots
C
       CALL MUTUAL_SLOT(N_SLOT)
C
C
       I_MIN=1
       I_MAX=2
       DO 11 I=I_MIN,I_MAX
             J_MIN=I_MIN+1
             J_MAX=I_MAX
             DO 12 J=J_MIN,J_MAX
                   IJMAX=NSSL(I,J)
                   WRITE (6,13) I,J
  13               FORMAT(10X,'Interactions between slots',I2,' and ',
     *                   I2//)
                   DO 14 IJ=1,IJMAX
                         WRITE (6,15) IJ,YS_ADM(I,J,IJ)
  15                     FORMAT(1X,'IJ=',I4,1X,'YSD=',E14.7,2X,E14.7)
  14               CONTINUE
  12         CONTINUE
  11   CONTINUE
C
       DLG=1.D0/DSQRT(1.D0-1.D0/(2.D0*AW)**2)
C
C      IS_OFF=2  Evaluate the mutual coupling between two slots vs.
C                slot separation distance
C
C      IS_OFF=1  Evaluate the mutual coupling between two slots half
C                waveguide wavelength appart
C
       IF (IS_OFF.EQ.2) THEN
          IZ_MIN=NSL(1)
          IZ_MAX=NXOFF(2)
          IZ_STEP=5
       ELSE IF (IS_OFF.EQ.1) THEN
          IZ_MIN=SLOT_DX/DLX
          IZ_MAX=IZ_MIN+1
          IZ_STEP=1
       END IF
C
       DO 108 IZ=IZ_MIN,IZ_MAX,IZ_STEP
              NXOFF(2)=IZ
              CALL ARRANGE_MUTUAL
C
C      Find the center of 1st slot
C
              NC0=(NSL(1)+1)/2
C
C      Find the center of 2nd slot
C
              NCI=(NSL(2)+1)/2
C
C      Find the corresponding row for IBMATR
C
              I_ROW=0
C
C      Find the corresponding collumn for IBMATR
C
              I_COL=NSL(1)
```

```
C
C     Find the Mutual coupling term
C
              SUM_MD=(0.0,0.0)
C
C     ICUR = 1    : We assume a form for the current
C
                      IN_MIN=1
                      IN_MAX=NSL(1)
                      DO 4 IN=IN_MIN,IN_MAX
                           CONSTN=CUR_SLOT(1,IN)/CUR_SLOT(1,NC0)
C
C                          WRITE (6,88) IN,CONSTN
C   88                     FORMAT(2X,'N=1',2X,'IN=',I4,5X,'CONSTN=',
C     *                    E14.7,2X,E14.7//)
C
                           IM_MIN=1
                           IM_MAX=NSL(2)
                           DO 5 IM=IM_MIN,IM_MAX
                                CONSTM=CUR_SLOT(2,IM)/CUR_SLOT(2,NCI)
                                CON=CONSTN*CONSTM
                                IJ=I_ROW+IN
                                KJ=I_COL+IM
                                IK=IBMATR(IJ,KJ)
C
C                         WRITE (6,89) IM,CONSTM,IJ,KJ,IK
C   89                    FORMAT(10X,'IM=',I4,2X,'CONSTM=',E14.7,2X,E14.7/
C     *                   10X,'IJ=',I4,2X,'KJ=',I4,2X,'IK=',I4)
C
                               SUM_MD=SUM_MD+SNGL(CON)*YS_ADM(1,2,IK)
  5                        CONTINUE
  4                   CONTINUE
  3             CONTINUE
                DIST_X=(NXOFF(2)-NXOFF(1))*DLX
                DIST_LG=DIST_X/DLG
                G0=(1.D0/(120.D0*PI))*DSQRT(1.D0-1.D0/(2.D0*AW)**2)
                Y12_MD=-SUM_MD/SNGL(G0)
C
C
C
C     Evaluation of the coupling term  Mc
C
C
C
                B01=DSQRT(1.D0-(0.5D0/AW)**2)
                B012=B01*B01
                ARGY0=PI*YOFF(1)/AW
                ARGP=PI*DLX*(B01+1.D0)
                ARGM=PI*DLX*(B01-1.D0)
                ARG0=AK0*DLX
C
                WRITE (*,*) ARGY0,ARG0
C
                ARG=B01*2.D0*PI*DLX
                CALL BSJ0(PI*WS(1)/(2.D0*AW))
                DINC=(1.D0/(2*PI*AW))**2*(1.D0/(AW*BW))*(1.D0/DSIN(ARG0))
                DINC=DINC*DCOS(ARGY0)*BJ0*DSIN(ARGM)*DSIN(ARGP)/(B01*
     *              (1.D0-B012))
                CINC=SNGL(DINC)
                CF=(0.0,0.0)
                JQMAX=NSL(1)
                DO 71 JQ=1,JQMAX
                      ARGX=ARG*FLOAT(JQ-1)
                      EC=DCOS(ARGX)
                      ES=DSIN(ARGX)
                      CF=CF+SNGL(DSIN((JQ-1)*PI/(NSL(1)-1)))
```

```
     *                  *(SNGL(EC)+CI*SNGL(ES))
  71        CONTINUE
            CF=CINC*CF
            ABS_CF=CABS(CF)
C
            S_LENG=(NSL(1)-1)*DLX
            AS_LENG=1.D0/(2.D0*S_LENG)
            FN=AS_LENG*DCOS(B01*PI*S_LENG)*DSIN(ARGY0)/
     *         (AS_LENG**2-B012)
            CMP=FN**2/(B01*(2.D0*PI*BW)**2*(2.D0*PI*AW)*4.D0*AW**5)
C
            G_SLOT=-CI*SNGL(CMP)*Y12_MD/ABS_CF**2
            CMC_EXT=Y12_MD/(SNGL(32.D0*AW**2*PI**2*AW*BW)*ABS_CF**2)
C
            WRITE (6,66) DIST_X,G_SLOT,CMC_EXT
  66        FORMAT(//10X,'DIST=',E14.7,2X,'G_SLOT=',E14.7,2X,E14.7,
     *      2X,'CMC_EXT=',E14.7,2X,E14.7//)
C
            IF (IS_OFF.EQ.2) THEN
                  WRITE (6,52) DIST_X
  52              FORMAT(///2X,'LONGITUDINAL DISTANCE IN',
     *            ' WAVELENGTHS IN FREE SPACE=',E14.7/)
                  WRITE (6,53) DIST_LG
  53              FORMAT(///2X,'LONGITUDINAL DISTANCE IN',
     *            ' WAVELENGTHS IN WAVEGUIDE=',E14.7/)
                  WRITE (6,60) SUM_MD,
 60               FORMAT (/10X,'SUM_MD=',E14.7,5X,E14.7//)
                  WRITE (6,62) Y12_MD
  62              FORMAT(/10X,'Y12_MD=',E14.7,2X,E14.7//)
                  WRITE (6,82) G_SLOT
 82               FORMAT (10X,'G_SLOT=',E14.7,2X,E14.7)
                  WRITE (6,83) CMC_EXT
 83               FORMAT (10X,'C_MC=',E14.7,2X,E14.7)
                  WRITE (7,707) DIST_X,Y12_MD,G_SLOT,CMC_EXT
707               FORMAT(E14.7,2X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7)
            ELSE IF (IS_OFF.EQ.1) THEN
                  IF (IZ.EQ.IZ_MIN) THEN
                      RG_SLOT1=REAL(G_SLOT)
                      AIG_SLOT1=AIMAG(G_SLOT)
                      RCMC_EXT1=REAL(CMC_EXT)
                      AICMC_EXT1=AIMAG(CMC_EXT)
                  ELSE IF (IZ.EQ.IZ_MAX) THEN
                      RG_SLOT2=REAL(G_SLOT)
                      AIG_SLOT2=AIMAG(G_SLOT)
                      RCMC_EXT2=REAL(CMC_EXT)
                      AICMC_EXT2=AIMAG(CMC_EXT)
                  END IF
            END IF
 108  CONTINUE
      IF (IS_OFF.EQ.1) THEN
           SC1=(IZ_MAX*DLX-SLOT_DX)/DLX
           SC2=(SLOT_DX-IZ_MIN*DLX)/DLX
           G_SLOT=(RG_SLOT1+CI*AIG_SLOT1)*SNGL(SC1)+
     *               (RG_SLOT2+CI*AIG_SLOT2)*SNGL(SC2)
           WRITE (6,82) G_SLOT
           CMC_EXT=(RCMC_EXT1+CI*AICMC_EXT1)*SNGL(SC1)+
     *                (RCMC_EXT2+CI*AICMC_EXT2)*SNGL(SC2)
           WRITE (6,83) CMC_EXT
      END IF
 1000 CONTINUE
      STOP
      END
C**********************************************************************
C     THIS FUNCTION EVALUTES EER
C**********************************************************************
      SUBROUTINE F_EER
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---- Normalization Constant ----
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C      EER=ER+(1.D0-ER)*(W/H)/(1.D0+W/H)
C
      EER=1.0
      WRITE (6,100) EER
 100  FORMAT(10X,'Normalization Constant'/10X,E14.7/)
      RETURN
      END
C*********************************************************************
C      NORMALIZATION SUBROUTINE
C
C      THIS SUBROUTINE DENORMALIZES WITH RESPECT TO CNORM_OLD
C      AND NORMALIZES AGAIN WITH RESPECT TO CNORM_NEW
C*********************************************************************
      SUBROUTINE NORM(CNORM_OLD,CNORM_NEW)
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      CNORM=CNORM_OLD/CNORM_NEW
C
      PI=3.141592654
C
      AK0=2.D0*PI*CNORM_NEW
      AKK=2.D0*PI
      AK=AK0*DSQRT(ER)
C
      H=H*CNORM
      AW=AW*CNORM
      BW=BW*CNORM
      T=T*CNORM
      DLX=DLX*CNORM
      OFFLIM=OFFLIM*CNORM
C
      YOFF(1)=YOFF(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 8 I=2,NSLOTS
                  YOFF(I)=YOFF(I)*CNORM
  8          CONTINUE
      END IF
C
      WS(1)=WS(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 15 I=2,NSLOTS
                  WS(I)=WS(I)*CNORM
  15         CONTINUE
      END IF
C
      WSDELTA(1)=WSDELTA(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 18 I=2,NSLOTS
```

```
                  WSDELTA(I)=WSDELTA(I)*CNORM
  18         CONTINUE
      END IF
      RETURN
      END
C**************************************************************
C....................... Spline Interpolation ...................
C**************************************************************
      SUBROUTINE  CUBSPL(ICUR,DLX,IEND,N_SLOT,IRX)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX CURRENT,CUR_SLOT,CC
      REAL*4 RCUR,AICUR,REAL_CUR,AIMAG_CUR
      DIMENSION S(260),A(260,4),X(260),Y(260),AI(260),BI(260),
     *CI(260),DI(260)
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/RES/CUR_SLOT(30,60),DLX_SLOT(30)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
C     This routine computes the matrix for finding the coefficients of a
C     cubic spline through a set of data.
C     The  system is then solved to obtain the second derivative values,
C     and the coefficients of the cubic spline between each pair of points.
C      ------------------------------------------------------------------
C      Parameters are
C      X,Y       Arrays of X and Y values to be fitted
C
C      DLX        Subsection length (if all points have same spacing)
C
C      S         Array of second derivative values at the points
C
C      N         Number of points
C
C      IEND      Type of end condition to be used
C                    IEND=1, Linear ends, S(1)=S(N)=0
C                    IEND=2, Parabolic ends, S(1)=S(2), S(N)=S(N-1)
C                    IEND=3, Cubic ends S(1),S(N) are extrapolated
C
C      A          Augmented matrix of coefficients and R.H.S. for finding S
C
C      IRX       1  : Interpolate the real part of the current
C                2  : Interpolate the imaginary part of the current
C
C      ICUR     =0  resonant field derived from GENERATE
C      ICUR     =1  we assume a form for the resonant field
C      ------------------------------------------------------------------
       PI=3.141592654
       N=NSL(N_SLOT)
       CC=(0.0,1.0)
C
C      Computation of matrices X,Y
C
       NC0_OLD=(NSL(N_SLOT)+1)/2
       NSLOT_NEW=2*NINT((NSL(N_SLOT)-1)*DLX_SLOT(N_SLOT)/(2.0*DLX))+1
       NC0_NEW=(NSLOT_NEW+1)/2
       ITEST=(NSLOT_NEW+1)-NC0_NEW*2
       I_CUR=(NSLOT_NEW+1)/2
       I_MIN=1
       I_MAX=NSL(N_SLOT)
       L_MAX=NSLOT_NEW
C
       WRITE (*,*) L_MAX
C
       IF (ICUR.EQ.1) GO TO 500
```

```
      DO 1 I=I_MIN,I_MAX
           X(I)=DLX_SLOT(N_SLOT)*FLOAT(I-1)
           REAL_CUR=REAL(CUR_SLOT(N_SLOT,I))
           CURRENT=-CC*CUR_SLOT(N_SLOT,I)
           AIMAG_CUR=REAL(CURRENT)
           IF (IRX.EQ.1) Y(I)=DBLE(REAL_CUR)
           IF (IRX.EQ.2) Y(I)=DBLE(AIMAG_CUR)
           WRITE (6,67) I,X(I),Y(I)
 67        FORMAT(10X,'I=',I4,2X,'X=',E14.7,2X,'Y=',E14.7)
  1   CONTINUE
C    ---------------------------------------------------------------------------------
C    | Compute the N-2 rows |
C    ------------------------
C
      NM2=N-2
      NM1=N-1
      DX1=X(2)-X(1)
      DY1=(Y(2)-Y(1))/DX1*6.D0
      DO 10 I=1,NM2
        DX2=X(I+2)-X(I+1)
        DY2=(Y(I+2)-Y(I+1))/DX2*6.D0
        A(I,1)=DX1
        A(I,2)=2.D0*(DX1+DX2)
        A(I,3)=DX2
        A(I,4)=DY2-DY1
        DX1=DX2
        DY1=DY2
10    CONTINUE
C
C   Adjust first and last rows to end condition
C
      GO TO  (20,50,80), IEND
C
C   for IEND = 1 no change is needed
C
20    GO TO 100
C
C   for IEND = 2, S(1)=S(2), S(N)=S(N-1), parabolic ends.
C
50    A(1,2)=A(I,2)+X(2)-X(1)
      A(NM2,2)=A(NM2,2)+X(N)-X(NM1)
      GO TO 100
C
C   for IEND = 3, cubic ends, S(1), S(N) are extrapolated.
C
80    DX1=X(2)-X(1)
      DX2=X(3)-X(2)
      A(1,2)=(DX1+DX2)*(DX1+2.D0*DX2)/DX2
      A(1,3)=(DX2*DX2-DX1*DX1)/DX2
      DXN2=X(NM1)-X(NM2)
      DXN1=X(N)-X(NM1)
      A(NM2,1)=(DXN2*DXN2-DXN1*DXN1)/DXN2
      A(NM2,2)=(DXN1+DXN2)*(DXN1+2.D0*DXN2)/DXN2
      GO TO 100
C
C   Now we solve the tridiagonal system.  First reduce
C
100   DO 110 I=2,NM2
        A(I,2)=A(I,2)-A(I,1)/A(I-1,2)*A(I-1,3)
        A(I,4)=A(I,4)-A(I,1)/A(I-1,2)*A(I-1,4)
110   CONTINUE
C
C   Back substitution
C
      A(NM2,4)=A(NM2,4)/A(NM2,2)
      DO 120 I=2,NM2
```

```
          J=NM1-I
          A(J,4)=(A(J,4)-A(J,3)*A(J+1,4))/A(J,2)
120     CONTINUE
C
C     Place values in S-vector
C
        DO 130 I=1,NM2
          S(I+1)=A(I,4)
130     CONTINUE
C
C     Set S(1) and S(N) according to end conditions
C
        GO TO (150,160,170), IEND
C
C     Linear ends
C
150     S(1)=0.
        S(N)=0.
        GO TO 200
C
C     Parabolic ends
C
160     S(1)=S(2)
        S(N)=S(N-1)
        GO TO 200
C
C     For cubic ends
C
170     S(1)=((DX1+DX2)*S(2)+DX1*S(3))/DX2
        S(N)=((DXN2+DXN1)*S(NM1)-DXN1*S(NM2))/DXN2
C
C     Find spline fit coefficients
C
C
C     Evaluation of the coefficients ai,bi,ci,di - Store into AI,BI
C     CI,DI
C
200      DO 210 I=1,NM1
           AI(I)=(S(I+1)-S(I))/(6.D0*DLX_SLOT(N_SLOT))
           BI(I)=S(I)/2.D0
           CI(I)=(Y(I+1)-Y(I))/DLX_SLOT(N_SLOT)-(2.D0*S(I)+S(I+1))
     *             *DLX_SLOT(N_SLOT)/6.D0
210        DI(I)=Y(I)
C
C      Re-evaluate nsl(n_slot) and cur_res(n_slot)
C
      DO 2 I=1,I_CUR
           IF (ITEST.EQ.0) NCP=NC0_NEW+I-1
           IF (ITEST.EQ.1) NCP=NC0_NEW+I
           NCM=NC0_NEW-I+1
           DISTP=(NCP-1)*DLX
           DISTM=(NCM-1)*DLX
           RIP=DISTP/DLX_SLOT(N_SLOT)
           IP=INT(RIP)
           IF ((RIP-IP).GT.0.999) IP=IP+1
           IF(IP.EQ.NSLOT_NEW) IP=IP-1
           RIM=DISTM/DLX_SLOT(N_SLOT)
           IM=INT(RIM)
           IF((RIM-IM).GT.0.999) THEN
                IM=IM+1
           END IF
           DIFP=DISTP-FLOAT(IP)*DLX_SLOT(N_SLOT)
           DIFM=DISTM-FLOAT(IM)*DLX_SLOT(N_SLOT)
           DIFP2=DIFP*DIFP
           DIFM2=DIFM*DIFM
           DIFP3=DIFP2*DIFP
```

```
          DIFM3=DIFM2*DIFM
          IF (IRX.EQ.1) THEN
             IP=IP+1
             IM=IM+1
             RCUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                 CI(IP)*DIFP+DI(IP))
             RCUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                 CI(IM)*DIFM+DI(IM))
             WRITE (6,666) NCP,IP,NCM,IM,RCUR(NCP),RCUR(NCM)
 666         FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *       I4/30X,'RCUR(NCP)=',E14.7,2X,'RCUR(NCM)=',E14.7)
          END IF
          IF (IRX.EQ.2) THEN
             IP=IP+1
             IM=IM+1
             AICUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                  CI(IP)*DIFP+DI(IP))
             AICUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                  CI(IM)*DIFM+DI(IM))
             WRITE (6,777) NCP,IP,NCM,IM,AICUR(NCP),AICUR(NCM)
 777         FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *       I4/10X,'AICUR(NCP)=',E14.7,2X,'AICUR(NCM)=',E14.7)
          END IF
  2   CONTINUE
C
      IF (IRX.EQ.2) NSL(N_SLOT)=NSLOT_NEW
      RETURN
C
 500  CONTINUE
      DO 502 L=1,L_MAX
            IF (IRX.EQ.1) THEN
                 RCUR(L)=SIN(PI*(L-1)/(NSLOT_NEW-1))
                 WRITE (6,601) L,RCUR(L)
 601             FORMAT(10X,'L=',I4,2X,'RCUR=',E14.7)
            ELSE IF (IRX.EQ.2) THEN
                 AICUR(L)=0.0
                 WRITE (6,602) L,AICUR(L)
 602             FORMAT(2X,'L=',I4,2X,'AICUR=',E14.7)
            END IF
 502     CONTINUE
C
      IF (IRX.EQ.2) NSL(N_SLOT)=NSLOT_NEW
C
      RETURN
      END
```

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

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

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

LAST MODIFIED ON: 89/04/24 10:36 AM
FILE PRINTED: 89/04/24 11:00 AM

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

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

LAST MODIFIED ON: 89/04/24 10:37 AM
FILE PRINTED: 89/04/24 11:07 AM

apollo domain

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:38 AM
FILE PRINTED: 89/04/24 10:47 AM

PROGRAM II

This program evaluates the resonant length of an isolated dielectric covered waveguide longitudinal slot as a function of the cover's relative dielectric constant, cover thickness, slot width and slot offset.

The files which consist this program are:

RUN_K0:	This program links all the subroutines.
DATA_WAVE_K0:	Input File
OUT_WAVE_K0:	Output File
GENERATE_K0.FTN :	Main Program Subroutine DATA Subroutine F_EER
MAIN_WAVE_K0.FTN:	Subroutine MAIN_WAVE Subroutine NORM
POLES.FTN :	Subroutine SPOLES
YIJ_DIEL_K0.FTN:	Subroutine YIJ_DIEL Subroutine LIMIT Subroutine GREEN Function GXXM Function GZXM Function HZXE Subroutine FUNCT Subroutine GREI Subroutine ARIS Subroutine ADONIS

	Subroutine BESS1
	Subroutine TAIL
	Subroutine BESS2
	Subroutine BSJ0
	Subroutine F
	Subroutine DATA_SLOT
YIJ_WAVE_K0.FTN:	Subroutine YIJ_WAVE
	Subroutine S14
	Subroutine VBJ0
INV_WAVE_K0.FTN	Subroutine INV_WAVE
	Subroutine MINVCD
	Subroutine SUBMCD

a p o l l o   d o m a i n

CAEN/Apollo

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

```
                                           000
rrrrr  u     u  n     n          k     k  0   0
r    r u     u  nn    n          k   k   0     0
r    r u     u  n n   n          kkkk    0     0
rrrrr  u     u  n  n  n          k  k    0     0
r   r  u     u  n    nn          k   k    0   0
r    r  uuuu    n     n _______  k     k   000
```

//tera/users/katehi/tape/run_k0

LAST MODIFIED ON: 89/04/24 10:55 AM
FILE PRINTED: 89/04/24 11:01 AM

```
BIND GENERATE_K0.BIN MAIN_WAVE_K0.BIN POLES.BIN YIJ_DIEL_K0.BIN
     YIJ_WAVE_K0.BIN   INV_WAVE_K0.BIN -B SLOT_K0
```

apollo domain

CAEN/Apollo

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

```
ddddd     aa      ttttt    aa          w     w    aa    v     v  eeeeee             ooo
d    d   a  a       t     a  a         w     w   a  a   v     v  e           k    k  o   o
d    d  a    a      t    a    a        w     w  a    a  v     v  eeeee       k   k  o     o
d    d  aaaaaa      t    aaaaaa        w ww w   aaaaaa  v     v  e           kkkk   o     o
d    d  a    a      t    a    a        ww  ww   a    a   v   v   e           k   k  o     o
ddddd   a    a      t    a    a ______ w     w  a    a    vv     eeeeee ______ k    k  o   o
                                                                             k     k  ooo
```

//tera/users/katehi/tape/data_wave_k0

LAST MODIFIED ON: 89/04/24 10:41 AM
FILE PRINTED: 89/04/24 10:49 AM

```
C
C     ---- Dielectric constant ---
C
      1.00001
C
C     ---- Substrate Thickness ---
C
      0.0185208
C
C     ---- Conductor Thickness ---
C
      0.000001
C
C     ---- Dimensions of the Waveguide ----
C
      0.6858
      0.3048
C
C     ---- Offset of the slot in the waveguide ----
C
      0.24765
C
C     ---- Slot width ----
C
      0.037041656
C
C     ---- Slot Excess Width ----
C
      0.0
C
C     ---- Subsection Length ----
C
      0.01061861
C
C     ---- Lower Limit of the Tail Contribution ----
C
      100.0
C
C     ---- Number of Points on the Slot ----
C
      29
C
C     ---- Number of Offsets for the dielectric ----
C
      1
C
C     ---- Offsets Between the Slots ----
C
      0.0
C
C     ---- Error in the evaluation of the series ----
C
      1.D-6
```

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

```
                                                                     000
 oooo    u     u    ttttt         w     w    aa     v     v   eeeeee         k     k  0     0
o     o  u     u      t           w     w   a  a    v     v   e              k   k   0       0
o     o  u     u      t           w     w  a    a   v     v   eeeee          kkkk    0       0
o     o  u     u      t           w ww w   aaaaaa   v     v   e              k  k    0       0
o     o  u     u      t           ww   ww  a    a    v   v    e              k   k    0     0
 oooo     uuuu        t   _______ w     w  a    a     vv      eeeeee _______ k    k    000
```

//tera/users/katehi/tape/out_wave_k0

LAST MODIFIED ON: 89/04/24 10:42 AM
FILE PRINTED: 89/04/24 10:57 AM

```
Dielectric Constant of the Substrate
 0.1000010E+01

Substrate Thickness
 0.1852080E-01

Conductor Thickness
 0.1000000E-05

Dimensions of the Waveguide
AW= 0.6858000E+00
BW= 0.3048000E+00

Offset of the slot
Y0= 0.2476500E+00

Slot Width
 0.3704166E-01

Slot Excess Width
 0.0000000E+00

Subsection Length
 0.1061861E-01

Lower Limit of Tail Contribution
 0.1000000E+03

Number of Points on the Slot
NS1=  29
Number of Offsets
   1

Offset
 0.0000000E+00

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

Normalization Constant
 0.1000000E+01




No TE waves excited in the substrate




There are   1 TM waves excited in the substrate


   1   0.628318532E+01
CONST=-0.2598055E+00




Contribution to admittance from the dielectric
```

```
            29
 K=    1
  YS= 0.1002559E-02   0.1239588E-05  GS=-0.2516107E-01  FSD=-0.3825282E-02

 K=    2
  YS= 0.7594118E-03   0.1239039E-05  GS= 0.5752941E-02  FSD=-0.2894459E-02

 K=    3
  YS= 0.1823966E-03   0.1237388E-05  GS= 0.3901441E-02  FSD=-0.6858614E-03

 K=    4
  YS=-0.3795267E-03   0.1234641E-05  GS= 0.1392694E-02  FSD= 0.1463932E-02

 K=    5
  YS=-0.6229520E-03   0.1230803E-05  GS= 0.6719672E-03  FSD= 0.2393130E-02

 K=    6
  YS=-0.4838263E-03   0.1225879E-05  GS= 0.3718589E-03  FSD= 0.1857090E-02

 K=    7
  YS=-0.1460914E-03   0.1219882E-05  GS= 0.2273453E-03  FSD= 0.5611756E-03

 K=    8
  YS= 0.1231841E-03   0.1212821E-05  GS= 0.1498514E-03  FSD=-0.4711988E-03

 K=    9
  YS= 0.1715970E-03   0.1204710E-05  GS= 0.1046965E-03  FSD=-0.6564383E-03

 K=   10
  YS= 0.4242733E-04   0.1195565E-05  GS= 0.7660745E-04  FSD=-0.1609708E-03

 K=   11
  YS=-0.1029601E-03   0.1185401E-05  GS= 0.5819261E-04  FSD= 0.3964225E-03

 K=   12
  YS=-0.1354520E-03   0.1174240E-05  GS= 0.4559021E-04  FSD= 0.5208779E-03

 K=   13
  YS=-0.5194168E-04   0.1162101E-05  GS= 0.3665173E-04  FSD= 0.2006697E-03

 K=   14
  YS= 0.5109497E-04   0.1149008E-05  GS= 0.3011761E-04  FSD=-0.1942577E-03

 K=   15
  YS= 0.8030815E-04   0.1134985E-05  GS= 0.2521611E-04  FSD=-0.3060789E-03

 K=   16
  YS= 0.2531187E-04   0.1120058E-05  GS= 0.2145622E-04  FSD=-0.9508056E-04

 K=   17
  YS=-0.4774246E-04   0.1104256E-05  GS= 0.1851521E-04  FSD= 0.1850742E-03

 K=   18
  YS=-0.6836429E-04   0.1087607E-05  GS= 0.1617484E-04  FSD= 0.2641650E-03

 K=   19
  YS=-0.2560176E-04   0.1070143E-05  GS= 0.1428383E-04  FSD= 0.1002649E-03

 K=   20
  YS= 0.3169442E-04   0.1051895E-05  GS= 0.1273495E-04  FSD=-0.1193086E-03

 K=   21
  YS= 0.4874380E-04   0.1032897E-05  GS= 0.1145070E-04  FSD=-0.1845544E-03

 K=   22
```

```
 YS= 0.1612558E-04   0.1013185E-05  GS= 0.1037405E-04  FSD=-0.5939474E-04

K=  23
 YS=-0.2803458E-04   0.9927938E-06  GS= 0.9462366E-05  FSD= 0.1099803E-03

K=  24
 YS=-0.4038068E-04   0.9717603E-06  GS= 0.8683310E-05  FSD= 0.1573622E-03

K=  25
 YS=-0.1405806E-04   0.9501232E-06  GS= 0.8012028E-05  FSD= 0.5649807E-04

K=  26
 YS= 0.1923609E-04   0.9279212E-06  GS= 0.7429186E-05  FSD=-0.7107564E-04

K=  27
 YS= 0.2434937E-04   0.9051940E-06  GS= 0.6919574E-05  FSD=-0.9059981E-04

K=  28
 YS=-0.2341931E-05   0.8819817E-06  GS= 0.6471105E-05  FSD= 0.1179821E-04

K=  29
 YS=-0.3149200E-04   0.8583254E-06  GS= 0.6074082E-05  FSD= 0.1236017E-03

         -0.1239588E-05  0.6545712E-02
         -0.1239039E-05 -0.1487230E-02
         -0.1237388E-05 -0.1009410E-02
         -0.1234641E-05 -0.3610188E-03
         -0.1230803E-05 -0.1757844E-03
         -0.1225879E-05 -0.9795505E-04
         -0.1219882E-05 -0.5936047E-04
         -0.1212821E-05 -0.3816813E-04
         -0.1204710E-05 -0.2614994E-04
         -0.1195565E-05 -0.1929681E-04
         -0.1185401E-05 -0.1508615E-04
         -0.1174240E-05 -0.1196968E-04
         -0.1162101E-05 -0.9328924E-05
         -0.1149008E-05 -0.7198956E-05
         -0.1134985E-05 -0.5764108E-05
         -0.1120058E-05 -0.4965028E-05
         -0.1104256E-05 -0.4469533E-05
         -0.1087607E-05 -0.3935100E-05
         -0.1070143E-05 -0.3263409E-05
         -0.1051895E-05 -0.2611225E-05
         -0.1032897E-05 -0.2179382E-05
         -0.1013185E-05 -0.2000734E-05
         -0.9927938E-06 -0.1919463E-05
         -0.9717603E-06 -0.1753102E-05
         -0.9501232E-06 -0.1461122E-05
         -0.9279212E-06 -0.1159895E-05
         -0.9051940E-06 -0.9866981E-06
         -0.8819817E-06 -0.9579209E-06
         -0.8583254E-06 -0.9576725E-06
 J=   1  YS=-0.1981945E-06   0.6591158E-02
 J=   2  YS=-0.1979879E-06  -0.1507198E-02
 J=   3  YS=-0.1973685E-06  -0.1012458E-02
 J=   4  YS=-0.1963376E-06  -0.3604004E-03
 J=   5  YS=-0.1948974E-06  -0.1740809E-03
 J=   6  YS=-0.1930509E-06  -0.9551602E-04
 J=   7  YS=-0.1908018E-06  -0.5822075E-04
 J=   8  YS=-0.1881550E-06  -0.3800676E-04
 J=   9  YS=-0.1851158E-06  -0.2627964E-04
 J=  10  YS=-0.1816908E-06  -0.1893080E-04
 J=  11  YS=-0.1778869E-06  -0.1416586E-04
 J=  12  YS=-0.1737122E-06  -0.1084505E-04
 J=  13  YS=-0.1691752E-06  -0.8533298E-05
 J=  14  YS=-0.1642856E-06  -0.6821965E-05
```

```
J=  15  YS=-0.1590534E-06  -0.5537389E-05
J=  16  YS=-0.1534897E-06  -0.4554695E-05
J=  17  YS=-0.1476059E-06  -0.3788376E-05
J=  18  YS=-0.1414144E-06  -0.3175634E-05
J=  19  YS=-0.1349281E-06  -0.2682359E-05
J=  20  YS=-0.1281604E-06  -0.2279050E-05
J=  21  YS=-0.1211256E-06  -0.1946268E-05
J=  22  YS=-0.1138382E-06  -0.1667089E-05
J=  23  YS=-0.1063135E-06  -0.1433393E-05
J=  24  YS=-0.9856713E-07  -0.1234178E-05
J=  25  YS=-0.9061527E-07  -0.1064086E-05
J=  26  YS=-0.8247447E-07  -0.9178405E-06
J=  27  YS=-0.7416173E-07  -0.7913422E-06
J=  28  YS=-0.6569437E-07  -0.6815385E-06
J=  29  YS=-0.5709005E-07  -0.5857473E-06
```

Total Admittance Matrix-No cav_slot contribution

```
I=    1 YS=-0.1437783E-05   0.1313687E-01
I=    2 YS=-0.1437027E-05  -0.2994427E-02
I=    3 YS=-0.1434757E-05  -0.2021868E-02
I=    4 YS=-0.1430979E-05  -0.7214192E-03
I=    5 YS=-0.1425700E-05  -0.3498653E-03
I=    6 YS=-0.1418930E-05  -0.1934711E-03
I=    7 YS=-0.1410684E-05  -0.1175812E-03
I=    8 YS=-0.1400976E-05  -0.7617490E-04
I=    9 YS=-0.1389826E-05  -0.5242958E-04
I=   10 YS=-0.1377255E-05  -0.3822761E-04
I=   11 YS=-0.1363288E-05  -0.2925200E-04
I=   12 YS=-0.1347952E-05  -0.2281473E-04
I=   13 YS=-0.1331276E-05  -0.1786222E-04
I=   14 YS=-0.1313294E-05  -0.1402092E-04
I=   15 YS=-0.1294038E-05  -0.1130150E-04
I=   16 YS=-0.1273548E-05  -0.9519723E-05
I=   17 YS=-0.1251862E-05  -0.8257910E-05
I=   18 YS=-0.1229021E-05  -0.7110734E-05
I=   19 YS=-0.1205071E-05  -0.5945768E-05
I=   20 YS=-0.1180055E-05  -0.4890274E-05
I=   21 YS=-0.1154023E-05  -0.4125650E-05
I=   22 YS=-0.1127024E-05  -0.3667823E-05
I=   23 YS=-0.1099107E-05  -0.3352856E-05
I=   24 YS=-0.1070327E-05  -0.2987280E-05
I=   25 YS=-0.1040738E-05  -0.2525208E-05
I=   26 YS=-0.1010396E-05  -0.2077735E-05
I=   27 YS=-0.9793557E-06  -0.1778040E-05
I=   28 YS=-0.9476761E-06  -0.1639459E-05
I=   29 YS=-0.9154155E-06  -0.1543420E-05
```

I=	Electric field on the slot		Amplitude	Phase
1	0.1614435E+00	-0.7345520E+01	0.7347293E+01	-0.8874093E+02
2	0.8273641E-01	-0.1036104E+02	0.1036137E+02	-0.8954248E+02
3	-0.5025021E-01	-0.1336159E+02	0.1336168E+02	-0.9021548E+02
4	-0.2428600E+00	-0.1597132E+02	0.1597316E+02	-0.9087117E+02
5	-0.4815925E+00	-0.1832820E+02	0.1833453E+02	-0.9150516E+02
6	-0.7586125E+00	-0.2044400E+02	0.2045807E+02	-0.9212509E+02
7	-0.1066368E+01	-0.2233228E+02	0.2235773E+02	-0.9273380E+02
8	-0.1397839E+01	-0.2399779E+02	0.2403847E+02	-0.9333364E+02

9	-0.1746198E+01	-0.2544281E+02	0.2550266E+02	-0.9392618E+02
10	-0.2104734E+01	-0.2666779E+02	0.2675071E+02	-0.9451267E+02
11	-0.2466826E+01	-0.2767223E+02	0.2778196E+02	-0.9509414E+02
12	-0.2825941E+01	-0.2845498E+02	0.2859496E+02	-0.9567160E+02
13	-0.3175624E+01	-0.2901487E+02	0.2918814E+02	-0.9624606E+02
14	-0.3509470E+01	-0.2935096E+02	0.2956003E+02	-0.9681844E+02
15	-0.3821083E+01	-0.2946276E+02	0.2970951E+02	-0.9738956E+02
16	-0.4104057E+01	-0.2935008E+02	0.2963563E+02	-0.9796012E+02
17	-0.4352017E+01	-0.2901312E+02	0.2933771E+02	-0.9853086E+02
18	-0.4558642E+01	-0.2845240E+02	0.2881528E+02	-0.9910256E+02
19	-0.4717673E+01	-0.2766889E+02	0.2806820E+02	-0.9967614E+02
20	-0.4822832E+01	-0.2666376E+02	0.2709641E+02	-0.1002526E+03
21	-0.4867701E+01	-0.2543817E+02	0.2589971E+02	-0.1008328E+03
22	-0.4845575E+01	-0.2399267E+02	0.2447709E+02	-0.1014179E+03
23	-0.4749359E+01	-0.2232681E+02	0.2282636E+02	-0.1020090E+03
24	-0.4571324E+01	-0.2043834E+02	0.2094332E+02	-0.1026075E+03
25	-0.4302834E+01	-0.1832252E+02	0.1882097E+02	-0.1032158E+03
26	-0.3932360E+01	-0.1596583E+02	0.1644297E+02	-0.1038365E+03
27	-0.3448498E+01	-0.1335653E+02	0.1379453E+02	-0.1044769E+03
28	-0.2800800E+01	-0.1035675E+02	0.1072878E+02	-0.1051327E+03
29	-0.2093092E+01	-0.7342163E+01	0.7634684E+01	-0.1059117E+03

Back-scattering Coefficient =-0.2175354E-02 -0.1632286E-01

Forward-scattering Coefficient =-0.2176553E-02 -0.1712974E-01

****

ER= 0.1000010E+01 H= 0.1852080E-01 Y0= 0.2476500E+00 W= 0.3704166E-01

Guessing initial values: DLX= 0.1061861E-01 RIM=-0.2175354E-02 AIM=-0.1632286E-01
LENGTH= 0.3185583E+00 Z_SELF= 0.3823972E-02 0.3277946E-01

No TE waves excited in the substrate

There are 1 TM waves excited in the substrate

1 0.628318532E+01
CONST=-0.2351324E+00

Contribution to admittance from the dielectric

29
K= 1

```
 YS= 0.1097995E-02   0.1369962E-05  GS=-0.2779911E-01  FSD=-0.4628884E-02

K=   2
 YS= 0.8062614E-03   0.1369290E-05  GS= 0.6432111E-02  FSD=-0.3394885E-02

K=   3
 YS= 0.1313041E-03   0.1367275E-05  GS= 0.4333653E-02  FSD=-0.5403978E-03

K=   4
 YS=-0.4819159E-03   0.1363920E-05  GS= 0.1524415E-02  FSD= 0.2051492E-02

K=   5
 YS=-0.6805859E-03   0.1359233E-05  GS= 0.7263372E-03  FSD= 0.2888161E-02

K=   6
 YS=-0.4453604E-03   0.1353225E-05  GS= 0.3994324E-03  FSD= 0.1888789E-02

K=   7
 YS=-0.5296727E-04   0.1345908E-05  GS= 0.2435742E-03  FSD= 0.2256135E-03

K=   8
 YS= 0.1806018E-03   0.1337297E-05  GS= 0.1604687E-03  FSD=-0.7635965E-03

K=   9
 YS= 0.1435202E-03   0.1327409E-05  GS= 0.1121980E-03  FSD=-0.6060590E-03

K=  10
 YS=-0.3497005E-04   0.1316267E-05  GS= 0.8222182E-04  FSD= 0.1502286E-03

K=  11
 YS=-0.1487566E-03   0.1303892E-05  GS= 0.6258459E-04  FSD= 0.6321469E-03

K=  12
 YS=-0.1065498E-03   0.1290310E-05  GS= 0.4914715E-04  FSD= 0.4532373E-03

K=  13
 YS= 0.1709776E-04   0.1275551E-05  GS= 0.3961316E-04  FSD=-0.7049714E-04

K=  14
 YS= 0.8805848E-04   0.1259644E-05  GS= 0.3263904E-04  FSD=-0.3708923E-03

K=  15
 YS= 0.5039743E-04   0.1242624E-05  GS= 0.2740266E-04  FSD=-0.2111408E-03

K=  16
 YS=-0.3625177E-04   0.1224523E-05  GS= 0.2338141E-04  FSD= 0.1560399E-03

K=  17
 YS=-0.7593658E-04   0.1205380E-05  GS= 0.2023198E-04  FSD= 0.3241964E-03

K=  18
 YS=-0.3634243E-04   0.1185234E-05  GS= 0.1772229E-04  FSD= 0.1565214E-03

K=  19
 YS= 0.3032945E-04   0.1164127E-05  GS= 0.1569149E-04  FSD=-0.1257995E-03

K=  20
 YS= 0.5310360E-04   0.1142101E-05  GS= 0.1402555E-04  FSD=-0.2221231E-03

K=  21
 YS= 0.1632496E-04   0.1119200E-05  GS= 0.1264203E-04  FSD=-0.6618762E-04

K=  22
 YS=-0.3293042E-04   0.1095472E-05  GS= 0.1148027E-04  FSD= 0.1425556E-03

K=  23
```

```
 YS=-0.4210184E-04   0.1070962E-05  GS= 0.1049487E-04  FSD= 0.1814694E-03

K=  24
 YS=-0.8197770E-05   0.1045722E-05  GS= 0.9651423E-05  FSD= 0.3791817E-04

K=  25
 YS= 0.2547904E-04   0.1019799E-05  GS= 0.8923433E-05  FSD=-0.1046469E-03

K=  26
 YS= 0.2069454E-04   0.9932468E-06  GS= 0.8290291E-05  FSD=-0.8427927E-04

K=  27
 YS=-0.1486701E-04   0.9661160E-06  GS= 0.7735774E-05  FSD= 0.6644313E-04

K=  28
 YS=-0.3893746E-04   0.9384598E-06  GS= 0.7246977E-05  FSD= 0.1684616E-03

K=  29
 YS=-0.2181683E-04   0.9103319E-06  GS= 0.6813542E-05  FSD= 0.9598280E-04

        -0.1369962E-05  0.6546066E-02
        -0.1369290E-05 -0.1504384E-02
        -0.1367275E-05 -0.1014743E-02
        -0.1363920E-05 -0.3579829E-03
        -0.1359233E-05 -0.1722710E-03
        -0.1353225E-05 -0.9516435E-04
        -0.1345908E-05 -0.5719043E-04
        -0.1337297E-05 -0.3667588E-04
        -0.1327409E-05 -0.2536525E-04
        -0.1316267E-05 -0.1897947E-04
        -0.1303892E-05 -0.1483403E-04
        -0.1290310E-05 -0.1153514E-04
        -0.1275551E-05 -0.8792733E-05
        -0.1259644E-05 -0.6824797E-05
        -0.1242624E-05 -0.5691862E-05
        -0.1224523E-05 -0.5059468E-05
        -0.1205380E-05 -0.4464702E-05
        -0.1185234E-05 -0.3706260E-05
        -0.1164127E-05 -0.2939676E-05
        -0.1142101E-05 -0.2422599E-05
        -0.1119200E-05 -0.2210443E-05
        -0.1095472E-05 -0.2110370E-05
        -0.1070962E-05 -0.1900193E-05
        -0.1045722E-05 -0.1551344E-05
        -0.1019799E-05 -0.1225026E-05
        -0.9932468E-06 -0.1071558E-05
        -0.9661160E-06 -0.1063012E-05
        -0.9384598E-06 -0.1030694E-05
        -0.9103319E-06 -0.8502520E-06
```

apollo domain

CAEN/Apollo

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

```
                                                                000
 gggg   eeeeee  n    n  eeeeee  rrrrr     aa     ttttt  eeeeee          k    k  0   0          ffffff  ttttt  n    n
g    g  e       nn   n  e       r    r   a  a      t    e               k   k  0     0         f         t    nn   n
g       eeeee   n n  n  eeeee   r    r  a    a     t    eeeee           kkkk   0     0         fffff     t    n n  n
g  ggg  e       n  n n  e       rrrrr   aaaaaa     t    e               k  k   0     0   ...   f         t    n  n n
g    g  e       n   nn  e       r   r   a    a     t    e               k   k   0   0    ...   f         t    n   nn
 gggg   eeeeee  n    n  eeeeee  r    r  a    a     t    eeeeee ______   k    k   000     ...   f         t    n    n
```

//tera/users/katehi/tape/generate_k0.ftn

LAST MODIFIED ON: 89/04/24 10:42 AM
FILE PRINTED: 89/04/24 10:52 AM

```
C.......................................................................................
C      The name of this file is:              GENERATE_K0.FTN
C
C      This program evaluates the resonant length of dielectric covered waveguide
C      slots.
C.......................................................................................
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL F_EER
      REAL SLOT_V,R_SELF
      COMPLEX BACK_SCAT,FORW_SCAT,Z_SELF
C
      COMMON/SCAT_COEF/BACK_SCAT,FORW_SCAT
C
      COMMON/SLOT_VOLTAGE/SLOT_V
C----------------------------------------------------------------------
C      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
C     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
C      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
C     *CUR_RES(30,60)
C----------------------------------------------------------------------
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/IOFF/INS,INS1S2
C
      OPEN(UNIT=05,FILE='DATA_WAVE_K0',STATUS='OLD')
      OPEN(UNIT=06,FILE='OUT_WAVE_K0',STATUS='OLD')
      OPEN(UNIT=07,FILE='PLOT_RES',STATUS='APPEND')
C
C     Subroutine DATA reads the values of the geometrical
C     parameters
C
      CALL DATA
C
C     Number of desired iterations
C
      NLIM=15
C
C     Tolerance value for the imaginary part of BACK_SCAT
C
      AIM_TOL=1.D-3
C
      ER0=ER
      Y00=Y0
      W0=W
      H0=H
C
      DY=0.03429
      DER=1.0
      DW=0.015
      DH=0.02
C
      IY_VALUES=1
      IER_VALUES=12
      IW_VALUES=1
      IH_VALUES=1
      DLX0=DLX
C
      DO 1 IW=1,IW_VALUES
           W=W0+(IW-1)*DW
           DO 2 IY=1,IY_VALUES
                Y0=Y00+(IY-1)*DY
                DO 3 IER=1,IER_VALUES
```

```
                  ER=ER0+(IER-1)*DER
                  DO 4 IH=1,IH_VALUES
                       H=H0+(IH-1)*DH
                       CALL F_EER
                       SIG_ADD=1.0
                       AMPL_ADD=1.0
                       DLX1=DLX0
                       DLX=DLX1
                       CALL MAIN_WAVE
C----------------------------------------------------------------------
                       WRITE (*,300) ER,H,Y0,W
                       WRITE (6,300) ER,H,Y0,W
 300                   FORMAT(//'***********************************'
     *                 //10X,'ER=',E14.7,1X,'H=',E14.7,
     *                 1X,'Y0=',E14.7,1X,'W=',E14.7//)
                       AIM1=AIMAG(BACK_SCAT)
                       RIM1=REAL(BACK_SCAT)
C----------------------------------------------------------------------
                       WRITE (6,340) DLX,RIM1,AIM1
                       WRITE (*,340) DLX,RIM1,AIM1
 340                   FORMAT('Guessing initial values:',3X,
     *                 'DLX=',E14.7,2X,'RIM=',E14.7,2X,
     *                 'AIM=',E14.7)
C----------------------------------------------------------------------
                       S_LENGTH=(NS1+1)*DLX
                       Z_SELF=-2.0*BACK_SCAT/(1.0+BACK_SCAT)
C----------------------------------------------------------------------
                       WRITE(6,640) S_LENGTH,Z_SELF
                       WRITE(*,640) S_LENGTH,Z_SELF
 640                   FORMAT(20X,'LENGTH=',E14.7,2X,
     *                 'Z_SELF=',E14.7,1X,E14.7)
C----------------------------------------------------------------------
                       IF (ABS(AIM1).LT.AIM_TOL) THEN
                             AIM=AIM1
                             N=0
                             GO TO 100
                       END IF
                       ADD=-AIM1/30
 10                    DLX2=DLX+ADD
                       DLX=DLX2
                       IF (DLX.GT.0.03) GO TO 500
                       CALL MAIN_WAVE
                       AIM2=AIMAG(BACK_SCAT)
                       RIM2=REAL(BACK_SCAT)
                       IF (ABS(AIM2).LT.AIM_TOL) THEN
                             AIM=AIM2
                             N=0
                             GO TO 100
                       END IF
C----------------------------------------------------------------------
                       WRITE(6,340) DLX2,RIM2,AIM2
                       WRITE(*,340) DLX2,RIM2,AIM2
C----------------------------------------------------------------------
                       S_LENGTH=(NS1+1)*DLX
                       Z_SELF=-2.0*BACK_SCAT/(1.0+BACK_SCAT)
C----------------------------------------------------------------------
                       WRITE(6,640) S_LENGTH,Z_SELF
                       WRITE(*,640) S_LENGTH,Z_SELF
C----------------------------------------------------------------------
                       IF(AIM1*AIM2.GT.0) THEN
                             DIF=(AIM2-AIM1)/AIM1
                             IF (DIF.LT.0) THEN
                                 AIM1=AIM2
                                 DLX1=DLX2
                             END IF
                             GO TO 10
```

```
                         END IF
C-----------------------------------------------------------------------
                         WRITE (*,12) DLX1,AIM1,DLX2,AIM2
  12                     FORMAT(10X,'Initial Values:','dlx1=',E14.7,2X,
     *                   'aim1=',E14.7/10X,15X,'dlx2=',E14.7,2X,'aim2='
     *                   ,E14.7/)
C-----------------------------------------------------------------------
                         IF (AIM1.LT.0) THEN
                              AIMR=AIM1
                              DLXR=DLX1
                              DELTA_DLX=DLX2-DLX1
                         ELSE
                              AIMR=AIM2
                              DLXR=DLX2
                              DELTA_DLX=DLX1-DLX2
                         ENDIF
                         DO 20 N=1,NLIM
                               DELTA_DLX=DELTA_DLX*0.5D0
                               DLX=DLXR+DELTA_DLX
                               CALL MAIN_WAVE
                               RIM=REAL(BACK_SCAT)
                               AIM=AIMAG(BACK_SCAT)
C-----------------------------------------------------------------------
                               WRITE (*,13) N,DLX,RIM,AIM
  13                           FORMAT(10X,'Iteration #',I4,'dlx=',
     *                         E14.7,2X,'rim=',E14.7,1X,'aim=',E14.7)
C-----------------------------------------------------------------------
                               S_LENGTH=(NS1+1)*DLX
                               Z_SELF=-2.0*BACK_SCAT/(1.0+BACK_SCAT)
C-----------------------------------------------------------------------
                               WRITE(6,640) S_LENGTH,Z_SELF
                               WRITE(*,640) S_LENGTH,Z_SELF
C-----------------------------------------------------------------------
                               IF (ABS(AIM).LT.AIM_TOL) GO TO 100
                               IF (AIM.LT.0) THEN
                                   AIMR=AIM
                                   DLXR=DLX
                               ENDIF
  20                     CONTINUE
C-----------------------------------------------------------------------
                         WRITE(*,15) NLIM,DLXR,AIMR
                         WRITE(6,15) NLIM,DLXR,AIMR
  15                     FORMAT(//10X,'Tolerance not met after',I4,
     *                   'iterations'/10X,'dlxr=',E14.7,2X,'aimr=',
     *                   E14.7/)
C-----------------------------------------------------------------------
                         GO TO 600
C-----------------------------------------------------------------------
 500                     WRITE(6,16) DLX1,AIM1,DLX2,AIM2
                         WRITE(*,16) DLX1,AIM1,DLX2,AIM2
  16                     FORMAT(//10X,'The initial values do not have'
     *                  ,' opposite sign'/10X,'dlx1=',E14.7,2X,'aim1='
     *                   ,E14.7/10X,'dlx2=',E14.7,2X,'aim2=',E14.7/)
C-----------------------------------------------------------------------
                         GO TO 600
 100                     AIMR=AIM
                         RIMR=RIM
                         DLXR=DLX
C-----------------------------------------------------------------------
                         WRITE(*,17) N,DLXR,RIMR,AIMR
                         WRITE(6,17) N,DLXR,RIMR,AIMR
  17                     FORMAT(//10X,'Tolerance met after',I4,2X,
     *                   'iterations.'/10X,'dlxr=',E14.7,2X,'rimr='
     *                   ,E14.7,2X,'aimr=',E14.7)
C-----------------------------------------------------------------------
                         S_LENGTH=(NS1+1)*DLX
```

```
                              Z_SELF=-2.0*BACK_SCAT/(1.0+BACK_SCAT)
                              R_SELF=REAL(Z_SELF)
C--------------------------------------------------------------------------
                              WRITE(6,640) S_LENGTH,R_SELF
                              WRITE(*,640) S_LENGTH,R_SELF
                              WRITE(7,707) ER,H,S_LENGTH,R_SELF,SLOT_V
C---------------------------------------------------------------------------
 707                          FORMAT(E14.7,2X,E14.7,2X,E14.7,2X,E14.7,
     *                        2X,E14.7)
C----------------------------------------------------------------------
  19                          FORMAT(2X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7,
     *                        2X,E14.7)
C----------------------------------------------------------------------
  4                    CONTINUE
  3               CONTINUE
  2          CONTINUE
  1     CONTINUE
 600    CONTINUE
 3000   CONTINUE
        STOP
        END
C.............................................................
C       The name of this subroutine is        DATA
C       and gives all the data used by the main program and the other
C       subroutines.
C..............................................................
        SUBROUTINE DATA
        IMPLICIT REAL*8 (A-H,O-Z)
C
        COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
       *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
        COMMON/IOFF/INS,INS1S2
C
C       ******** ALL LENGTHS ARE NORMALIZED WITH RESPECT TO
C                FREE-SPACE WAVELENGTH *********
C
        PI=3.141592653589D0
C
C       ---- Dielectric constant ---
C
        READ (5,1) ER
  1     FORMAT (///6X,D16.9)
        WRITE (6,2) ER
  2     FORMAT(10X,'Dielectric Constant of the Substrate'/10X,E14.7/)
C
C       ---- Substrate Thickness ---
C
        READ (5,1) H
        WRITE (6,3) H
  3     FORMAT(10X,'Substrate Thickness'/10X,E14.7/)
C
C       ---- Conductor Thickness ---
C
        READ (5,1) T
        WRITE (6,4) T
  4     FORMAT(10X,'Conductor Thickness'/10X,E14.7/)
C
C       ---- Dimensions of the Waveguide ----
C
        READ (5,1) AW
        READ (5,10) BW
  10    FORMAT(6X,D14.7)
        WRITE (6,5) AW,BW
  5     FORMAT(10X,'Dimensions of the Waveguide'/10X,'AW=',E14.7/
       *10X,'BW=',E14.7/)
```

```
C
C     ---- Limit for offsets: Small Offset< OFFLIM ----
C                             Large Offset> OFFLIM
C
      OFFLIM=0.1
C
C     ---- Offset of the slot in the waveguide ----
C
      READ (5,1) Y0
      WRITE (6,6) Y0
    6 FORMAT(10X,'Offset of the slot'/10X,'Y0=',E14.7/)
C
C     ---- Slot width ----
C
      READ (5,1) W
      WRITE (6,7) W
    7 FORMAT(10X,'Slot Width'/10X,E14.7/)
C
C     ---- Slot excess width ----
C
      READ (5,1) WDELTA
      WRITE (6,8) WDELTA
    8 FORMAT(10X,'Slot Excess Width'/10X,E14.7/)
C
C
C     ---- Subsection Length ----
C
      READ (5,1) DLX
      WRITE (6,9) DLX
    9 FORMAT(10X,'Subsection Length'/10X,E14.7/)
C
C
C     ---- Lower Limit of the Tail Contribution ----
C
      READ (5,1) A
      WRITE (6,11) A
   11 FORMAT(10X,'Lower Limit of Tail Contribution'/10X,E14.7/)
C
C     ---- Number of Points on Each Slot ----
C
      READ (5,20) NS1
   20 FORMAT(///6X,I4)
      WRITE (6,12) NS1
   12 FORMAT (10X,'Number of Points on the Slot'/10X,'NS1=',I4)
C
C     ---- Number of Offsets for the dielectric ----
C
      READ (5,20) NOFF
      WRITE (6,15) NOFF
   15 FORMAT(10X,'Number of Offsets'/10X,I4/)
C
C     ---- Offsets Between the Slots ----
C
      READ (5,1) OFFSET(1)
      WRITE (6,13) OFFSET(1)
   13 FORMAT (10X,'Offset'/10X,E14.7/)
C
C     ---- Order of Offsets ----
C
      INS=1
      INS1S2=2
C
C
C     ---- Error in the evaluation of the series ----
C
      READ (5,1) ERROR
```

```
      WRITE (6,16) ERROR
   16 FORMAT(10X,'Error in the evaluation of the series'/
     *10X,'ERROR=',E14.7/)
      RETURN
      END
C-----------------------------------------------------------------------
C     THIS FUNCTION EVALUTES EER
C-----------------------------------------------------------------------
      SUBROUTINE F_EER
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---- Normalization Constant ----
C
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
C      EER=ER+(1.D0-ER)*(W/H)/(1.D0+W/H)
C
      EER=1.0
      WRITE (6,100) EER
      WRITE(*,100) EER
  100 FORMAT(10X,'Normalization Constant'/10X,E14.7/)
      RETURN
      END
```

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

//tera/users/katehi/tape/main_wave_k0.ftn

LAST MODIFIED ON: 89/04/24 10:43 AM
FILE PRINTED: 89/04/24 10:55 AM

```
C.......................................................................
C                              MAIN_WAVE_K0.FTN
C       This program solves the problem of a dielectric covered waveguide
C                                   slot
C.......................................................................
      SUBROUTINE MAIN_WAVE
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*4 CPU_SECONDS
      INTEGER*2 TIMEDATE_REC(6)
      COMPLEX YS(250),CI,CUR,BACK_SCAT,FORW_SCAT
      COMPLEX YSD,YSW
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/OUT/GS(250)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(51),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INS,INS1S2
C
      COMMON/B01/BJ0,BJ1
C
      COMMON/MAN/BMATR(260,260),IA(260),IB(260)
C
      COMMON/INV/CUR(260),NOR
C
      COMMON/SCAT_COEF/BACK_SCAT,FORW_SCAT
C
      COMMON/TEST/FSD(250)
C
      CI=(0.0,1.0)
C     Initialize YS to zero values
C
      DO 5 IYS=1,250
           YSD(IYS)=(0.0,0.0)
           YSW(IYS)=(0.0,0.0)
           FSD(IYS)=0.0
  5   CONTINUE
C
C     Subroutine YIJ_DIEL evaluates the contribution to the elements
C     of the admittance matrix coming from the dielectric substrate
C
      CNORM_OLD=1.D0
      CNORM_NEW=1.D0/DSQRT(EER)
      CALL NORM(CNORM_OLD,CNORM_NEW)
      CALL YIJ_DIEL
C
C     Subroutine YIJ_WAVE evaluates the contribution coming from the
C     waveguide
```

```
C
      CNORM_OLD=1.D0/DSQRT(EER)
      CNORM_NEW=1.D0
      CALL NORM(CNORM_OLD,CNORM_NEW)
      CALL YIJ_WAVE
C----------------------------------------------------------
      WRITE (6,12)
  12  FORMAT(///10X,'Total Admittance Matrix-No cav_slot contribution'
     *///)
      DO 10 I=1,NS1
            YS(I)=YSD(I)+YSW(I)
            WRITE (6,11) I,YS(I)
  11        FORMAT(1X,'I=',I4,1X,'YS=',E14.7,2X,E14.7)
  10  CONTINUE
C------------------------------------------------------------
C
      CALL INV_WAVE(YS)
C
C     Write the electric field on the slot
C
      WRITE (6,3)
   3  FORMAT(///6X,'I=',6X,'Electric field on the slot',15X,
     *'Amplitude',10X,'Phase'///)
      DO 1 I=1,NOR
           CUR_RE=REAL(CUR(I))
           CUR_IM=AIMAG(CUR(I))
           AMPL=CABS(CUR(I))
           PHASE=180.D0*ATAN2(CUR_IM,CUR_RE)/PI
           WRITE (6,2) I,CUR(I),AMPL,PHASE
  2        FORMAT (5X,I4,5X,(E14.7,1X,E14.7),5X,E14.7,5X,E14.7)
  1   CONTINUE
C
C     Write the scattering coefficients
C
      WRITE (6,4) BACK_SCAT,FORW_SCAT
   4  FORMAT(///6X,'Back-scattering Coefficient =',E14.7,2X,E14.7
     *       //6X,'Forward-scattering Coefficient =',E14.7,2X,
     *       E14.7///)
      RETURN
      END
C---------------------------------------------------------------
C      NORMALIZATION SUBROUTINE
C
C      THIS SUBROUTINE DENORMALIZES WITH RESPECT TO CNORM_OLD
C      AND NORMALIZES AGAIN WITH RESPECT TO CNORM_NEW
C---------------------------------------------------------------
      SUBROUTINE NORM(CNORM_OLD,CNORM_NEW)
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/IOFF/INS,INS1S2
C
      CNORM=CNORM_OLD/CNORM_NEW
C
      PI=3.141592654
C
      AK0=2.D0*PI*CNORM_NEW
      AKK=2.D0*PI
      AK=AK0*DSQRT(ER)
C
      H=H*CNORM
      AW=AW*CNORM
      BW=BW*CNORM
      T=T*CNORM
```

```
      Y0=Y0*CNORM
      DLX=DLX*CNORM
      OFFLIM=OFFLIM*CNORM
      W=W*CNORM
      WDELTA=WDELTA*CNORM
C
      DO 1 I=1,NOFF
           OFFSET(I)=OFFSET(I)*CNORM
  1   CONTINUE
      RETURN
      END
```

a p o l l o   d o m a i n

CAEN/Apollo

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

```
ppppp   oooo   l       eeeeee   ssss          ffffff   ttttt  n    n
p    p o    o  l       e       s              f          t    nn   n
p    p o    o  l       eeeee    ssss          fffff      t    n n  n
ppppp  o    o  l       e            s   ...   f          t    n  n n
p      o    o  l       e       s    s   ...   f          t    n   nn
p       oooo   llllll  eeeeee   ssss    ...   f          t    n    n
```

//tera/users/katehi/tape/poles.ftn

LAST MODIFIED ON: 89/04/24 10:43 AM
FILE PRINTED: 89/04/24 10:59 AM

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

a p o l l o   d o m a i n

CAEN/Apollo

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

```
y   y    i      j      w   w   aa   v   v  eeeeee     m   m  u   u  ttttt  u   u   aa   l          ffffff
 y y     i      j      w   w  a  a  v   v  e          mm mm  u   u    t    u   u  a  a  l          f
  y      i      j      w   w  a   a v   v  eeeee      m mm m u   u    t    u   u  a   a l          fffff
  y      i      j      w ww w aaaaaa v   v  e          m   m  u   u    t    u   u  aaaaaa l    ...  f
  y      i   j  j      ww  ww a   a  v v   e          m   m  u   u    t    u   u  a   a l    ...  f
  y      i    jjjj ____ w   w a   a   vv   eeeeee ____ m   m   uuuu    t     uuuu  a   a llllll ... f
```

//tera/users/katehi/tape/yij_wave_mutual.ftn

LAST MODIFIED ON: 89/04/24 10:38 AM
FILE PRINTED: 89/04/24 3:21 PM

```
C...............................................................................
C          The name of this file is YIJ_WAVE_MUTUAL.FTN
C...............................................................................
C     This subroutine evaluates the contribution to the admittance matrix
C     which comes from the waveguide
C.........................................................................
      SUBROUTINE YIJ_WAVE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YS,YS1S2,YSW,YS_ADM,YSW_ADM,COEF,CI
      DIMENSION ARG(200),R10(200),X10(200),AC(3),AS(3),ARG1(7),
     *ARG2(7),R00(2)
C
      COMMON/WAY_OUT/RS10(7,7,200),XS10(7,7,200),SGMN(7,7,200),
     *RIJ(7,7,200)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/SERIES/SUM(7,7,1)
C
      COMMON/BESSEL/BJ0(7,4000)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      ARGC=PI/AW
      DO 1 I=1,NSLOTS
           ARG2(I)=ARGC*WS(I)/2.D0
           ARG1(I)=ARGC*YOFF(I)
  1   CONTINUE
      CALL VBJ0(ARG2)
      CALL S14(ARG1)
C
      CI=(0.0,1.0)
      AKK2=AKK*AKK
      AK02=AK0*AK0
C
C     Evaluation of vector ARG
C
      JMAX=NSSL(1,1)
      DO 2 I=1,NSLOTS
           DO 3 J=I,NSLOTS
                IF (JMAX.LT.NSSL(I,J)) JMAX=NSSL(I,J)
  3        CONTINUE
  2   CONTINUE
      DO 4 J=1,JMAX
           ARG(J)=(J-1)*DLX
  4   CONTINUE
C
C     Evaluation of vectors R10,X10
C
      C3=1.D0
      B01=DSQRT(C3-1.D0/(2.D0*AW)**2)
      B012=B01*B01
      SCOEF=0.5D0*(C3-B012)/((2.D0*PI*(1.D0-B012))**2*B01)
      ARGK=AKK*DLX
      ARGB=ARGK*B01
      COSK=DCOS(ARGK)
      COSB=DCOS(ARGB)
      COS2K=DCOS(2.D0*ARGK)
      COS2B=DCOS(2.D0*ARGB)
      COS2=COSK*COSK
      SINK=DSIN(ARGK)
```

```
      SINB=DSIN(ARGB)
      SIN2K=2.D0*SINK*COSK
      SIN2B=2.D0*SINB*COSB
      SIN2=SINK*SINK
C
      R00(1)=(-DLX+SIN2K/(2.D0*AK0))/(4.D0*PI)
      R00(2)=(DLX*COSK-SINK/AK0)/(8.D0*PI)
C
      R10(1)=(8.D0*COSK*SINB-2.D0*SIN2B-2.D0*B01*SIN2K)*SCOEF
      C1=(COSK-COSB)**2
      X10(1)=-4.D0*C1*SCOEF
      R10(2)=2.D0*SINB*(-2.D0*C1-1.D0+B01*SINK/SINB)*SCOEF
      X10(2)=-4.D0*COSB*C1*SCOEF
      DO 5 J=3,JMAX
           R10(J)=-4.D0*DSIN(AK0*B01*ARG(J))*C1*SCOEF
           X10(J)=-4.D0*DCOS(AK0*B01*ARG(J))*C1*SCOEF
  5   CONTINUE
      DO 6 I=1,NSLOTS
           DO 7 J=I,NSLOTS
                COSI=DCOS(ARG1(I))
                COSJ=DCOS(ARG1(J))
                SFACT=COSI*COSJ*BJ0(I,1)
                DO 8 IJ=1,JMAX
                     RTEST=R10(IJ)*SFACT
                     RS10(I,J,IJ)=RTEST
                     XTEST=X10(IJ)*SFACT
                     XS10(I,J,IJ)=XTEST
  8             CONTINUE
  7        CONTINUE
  6   CONTINUE
C
C      DO 200 I=1,NSLOTS
C             DO 201 J=1,NSLOTS
C                    WRITE (6,204) I,J
C  204               FORMAT(5X,'Interaction between slots',I4,' and ',
C    *               I4//)
C                    DO 202 IJ=1,JMAX
C                          WRITE (6,203) IJ,RS10(I,J,IJ),XS10(I,J,IJ)
C  203                     FORMAT(10X,'IJ=',I4,2X,'RS10=',E14.7,2X,
C    *                     'XS10=',E14.7)
C  202               CONTINUE
C  201        CONTINUE
C  200 CONTINUE
C
C     Evaluation of vectors AC(A) and AS(a)
C
      AS(1)=2.D0
      AS(2)=-1.D0
      AS(3)=0.D0
C
      AC(1)=2.D0*(1.D0+2.D0*COS2)
      AC(2)=-4.D0*COSK
      AC(3)=1.D0
C
C     Evaluation of vector SGMN
C
      DO 9 I=1,NSLOTS
           DO 10 J=I,NSLOTS
                 KMAX=NSSL(I,J)+2
                 DO 11 K=1,KMAX
                       SGMN(I,J,K)=0.D0
                       NTEST=0
                       INDEXN=-1
 12                    INDEXN=INDEXN+1
                           EN=0.5D0
                           IF(INDEXN.GT.0) EN=1.D0
```

```
                              C1=(INDEXN/(2.D0*AW))**2
                              SUMM=0.D0
                              INDEXM=0
                              IF (INDEXN.GE.2) INDEXM=-1
                              ITEST=0
  13                          INDEXM=INDEXM+1
                                  EM=0.5D0
                                  IF (INDEXM.GT.0) EM=1.D0
                                  C2=(INDEXM/(2.D0*BW))**2
                                  GMN2=C1+C2-C3
                                  GMN=DSQRT(GMN2)
                                  ITEST=ITEST+1
                                  D1=AK0*(K-1)*GMN*DLX
                                  D2=0.D0
                                  IF (D1.LT.40.D0) D2=DEXP(-D1)
                                  TERM=EM*(C3+GMN2)*D2/(GMN*(1.D0+GMN2)**2)
                                  SUMM=SUMM+TERM
                                  RATIO=0.D0
                                  IF (SUMM.GT.1.D-40) RATIO=DABS(TERM/SUMM)
                                  ERRORM=ERROR
                                  IF (K.LE.3) ERRORM=1.D-9
                                  IF (RATIO.GT.ERRORM) ITEST=0
                                  IF (ITEST.LT.5) GO TO 13
                              NTEST=NTEST+1
                              CBJ0=1.D0
                              IF (INDEXN.GT.0) CBJ0=BJ0(I,INDEXN)
                              ARGNI=INDEXN*ARG1(I)
                              ARGNJ=INDEXN*ARG1(J)
                              COSI=DCOS(ARGNI)
                              COSJ=DCOS(ARGNJ)
                              TERM=EN*COSI*COSJ*CBJ0*SUMM
                              SGMN(I,J,K)=SGMN(I,J,K)+TERM
                              RATIO=DABS(TERM/SGMN(I,J,K))
                              IF (RATIO.GT.ERROR) NTEST=0
                              IF (NTEST.LT.4) GO TO 12
                          SGMN(I,J,K)=SGMN(I,J,K)/(2.D0*PI)**2
C
C                         WRITE (6,14) K,INDEXN,SGMN(I,J,K)
C 14                      FORMAT(10X,'K=',I4,2X,'INDEXN=',I4,2X,'SGMN=',
C    *                    E14.7)
C
  11                CONTINUE
  10          CONTINUE
   9    CONTINUE
C
C       Evaluation of vector RIJ
C
        DO 15 I=1,NSLOTS
              DO 16 J=I,NSLOTS
                    RIJ(I,J,1)=AC(1)*SGMN(I,J,1)+2.D0*AC(2)*SGMN(I,J,2)
     *                        +2.D0*SGMN(I,J,3)
                    RIJ(I,J,2)=AC(2)*SGMN(I,J,1)+(1.D0+AC(1))*SGMN(I,J,2)
     *                        +AC(2)*SGMN(I,J,3)+SGMN(I,J,4)
                    JMAX=NSSL(I,J)
                    DO 17 JK=3,JMAX
                          RIJ(I,J,JK)=SGMN(I,J,JK-2)+AC(2)*SGMN(I,J,JK-1)
     *                               +AC(1)*SGMN(I,J,JK)+AC(2)*
     *                               SGMN(I,J,JK+1)+SGMN(I,J,JK+2)
  17                CONTINUE
  16          CONTINUE
  15    CONTINUE
C
C        DO 19 I=1,NSLOTS
C              DO 190 J=I,NSLOTS
C                     WRITE (6,204) I,J
C                     DO 191 JK=1,JMAX
```

```
C                            WRITE (6,18) JK,RIJ(I,J,JK)
C  18                        FORMAT(2X,'JK=',I4,2X,'RIJ=',E14.7)
C  191                CONTINUE
C  190         CONTINUE
C  19  CONTINUE
C
C     Evaluation of this part of the elements of the admittance
C     which comes from the waveguide
C
      WRITE (6,20)
  20  FORMAT(///10X,'Waveguide Admittance Matrix'///)
      COEF=CI*SNGL(-2.D0/(120.D0*PI*AW*BW*SIN2))
      DO 21 I=1,NSLOTS
            DO 22 J=I,NSLOTS
C
C                    WRITE (6,25) I,J
C  25                FORMAT(10X,'Interactions between the slots',I2,
C     *                    ' and ',I2/)
C
                  JMAX=NSSL(I,J)
                  DO 23 IJ=1,JMAX
                        IF (IJ.LE.2) THEN
                            SINA=DSIN(AS(IJ)*AK0*DLX)
                            YR1=-SINA*SUM(I,J,1)
                            R_IJ=R00(IJ)+RS10(I,J,IJ)+YR1+RIJ(I,J,IJ)
                            X_IJ=XS10(I,J,IJ)
                        ELSE
                            R_IJ=RS10(I,J,IJ)+RIJ(I,J,IJ)
                            X_IJ=XS10(I,J,IJ)
                        END IF
                        YSW=COEF*(SNGL(R_IJ)+CI*SNGL(X_IJ))
C
C                        WRITE (6,24) IJ,YSW
C  24                    FORMAT(5X,'IJ=',I4,2X,'YSW=',E14.7,2X,E14.7)
C
                        YSW_ADM(I,J,IJ)=YSW
  23              CONTINUE
  22        CONTINUE
  21  CONTINUE
      RETURN
      END
C.......................................................................
C     This subroutine evaluates the single and double series S1,S2,S3,S4
C     which are common to all Yij elements
C.......................................................................
      SUBROUTINE S14(ARG)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ARG(7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/SERIES/SUM(7,7,1)
C
      COMMON/BESSEL/BJ0(7,4000)
C
C     Evaluation of the single integral
C
      COST=AW*BW/PI
      ARGC=PI*BW/AW
      DO 1 I=1,NSLOTS
           DO 2 J=I,NSLOTS
                INDEX=1
                ITEST=0
```

```
                COSI=DCOS(ARG(I))
                COSJ=DCOS(ARG(J))
                COTH=1.D0/DTANH(ARGC)
                CI=COSI*COSJ*BJ0(I,1)
                SUM(I,J,1)=(BW**2/6.D0)+CI*(ARGC*COTH-1.D0)*(AW/PI)**2
  3             INDEX=INDEX+1
                    ARGNI=INDEX*ARG(I)
                    ARGNJ=INDEX*ARG(J)
                    COTH=1.D0/DTANH(INDEX*ARGC)
                    COSI=DCOS(ARGNI)
                    COSJ=DCOS(ARGNJ)
                    CI=COSI*COSJ*BJ0(I,INDEX)
                    TERM=CI*COST*COTH/INDEX
                    SUM(I,J,1)=SUM(I,J,1)+TERM
                    RATIO=DABS(TERM/SUM(I,J,1))
                    IF (RATIO.LT.ERROR) GO TO 4
                           ITEST=0
                           GO TO 3
  4                 ITEST=ITEST+1
                    IF (ITEST.LT.6) GO TO 3
C
C                WRITE (6,10) I,J,SUM(I,J,1)
C  10            FORMAT(10X,'I=',I2,1X,'J=',I2,3X,'SUM(I,J,1)=',E14.7)
C
  2          CONTINUE
  1     CONTINUE
        RETURN
        END
C*********************************************************************
C     This function evaluates the zeroth order first kind Bessel
C                               Function J0
C*********************************************************************
      SUBROUTINE VBJ0(ARG)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ARG(7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/BESSEL/BJ0(7,4000)
C
      PI=3.141592653589D0
      I=1
 1    W=WS(I)
      DO 2 M=1,4000
           X=FLOAT(M)*ARG(I)
           IF (X.GT.0.001D0) GO TO 10
               X3=X/3.D0
               X32=X3*X3
               X34=X32*X32
               X36=X34*X32
               BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *              *X36
               BJ0(I,M)=BSJ0
               GO TO 2
  10       IF (X.GT.3.D0) GO TO 12
               X3=X/3.D0
               X32=X3*X3
               X34=X32*X32
               X36=X34*X32
               X38=X36*X32
               X310=X38*X32
               X312=X310*X32
               BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
```

```
     *             *X36+0.0444479D0*X38-0.0039444D0*X310+0.00021000
     *             D0*X312
                BJ0(I,M)=BSJ0
                GO TO 2
 12         CONTINUE
            X3=3.D0/X
            X32=X3*X3
            X33=X32*X3
            X34=X33*X3
            X35=X34*X3
            X36=X35*X3
            FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.0000
     *          9512D0*X33+0.00137237D0*X34-0.00072805D0*X35+0.00014
     *          476D0*X36
            TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00
     *          262573D0*X33-0.00054125D0*X34-0.00029333D0*X35+0.000
     *          13558D0*X36
            WCON=DSQRT(1.D0/X)
            BSJ0=WCON*FJ0*DCOS(TJ0)
            BJ0(I,M)=BSJ0
  2    CONTINUE
       IF (I.EQ.NSLOTS) GO TO 100
  5    I=I+1
       LIMAX=I-1
       DO 3 LI=1,LIMAX
            IF(WS(I).EQ.WS(LI)) THEN
                     DO 4 M=1,4000
                          BJ0(I,M)=BJ0(LI,M)
  4                  CONTINUE
                     IF (I.EQ.NSLOTS) GO TO 100
                     GO TO 5
            END IF
  3    CONTINUE
       GO TO 1
100    CONTINUE
       RETURN
       END
```

a p o l l o   d o m a i n

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:38 AM
FILE PRINTED: 89/04/24 10:47 AM

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

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

```
C*******************************************************************
C     The name of this file is .......... POLES.FTN ...............
C*******************************************************************
      SUBROUTINE SPOLES
      IMPLICIT REAL*8(A-H,O-Z)
C..................................................................
C
C     ER     :....Dielectric constant
C
C     H      :....Height of the dielectric substrate
C
C     NE     :....Number of TE surface waves
C
C     NM     :....Number of tm surface waves
C
C     XS     :....Matrix of poles contributing to TE surface waves
C
C     XR     :....Matrix of poles contributing to TM surface waves
C
C     ERR    :....Error in the computation of the poles
C
C..................................................................
      DIMENSION XS(40),XR(40),LOR(40)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,
     *AK0_GENER,AK_GENER,AKK_GENER,FA,OFFSET(7),ALONG(7),WDELTA,
     *OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),TMP(20),TEP(20),
     *AM(41),DM(41),TPO(40),VXXM(20),VZXM(20),VZXE(20),BPOINT(10),
     *BCOAL(10),MPOINT,NPOINT,NK0,MA,NM,NE,NK0K,IFIRST
C
      AER=DSQRT(EER)
      ER2=ER*ER
      PI2=PI*PI
      MAXE=5
      ERR=0.0000001D0
      DP=H/AER
C
C     PART I : TE MODES
C
      AK0=2.D0*PI
      AK=DSQRT(ER)*AK0
      X0=DP*DSQRT(AK**2-AK0**2)
C-----------------------------------------------------------
C     WRITE (6,300) AK0,AK,X0,PI
C 300 FORMAT(10X,'AK0=',E14.7,2X,'AK=',E14.7,2X,'X0=',E14.7,
C    *2X,'PI=',E14.7/)
C-----------------------------------------------------------
      AN=X0/PI+0.5D0
      NE=AN
      IF (NE.EQ.0) GO TO 310
      DO 2 I=1,NE
           IF (X0-(2.D0*FLOAT(I)+1.D0)*PI/2.D0) 3,3,4
   4       XS0=(2.D0*FLOAT(I)-1.D0)*PI/2.D0+ERR
           XS1=(2.D0*FLOAT(I)+1.D0)*PI/2.D0-ERR
           GO TO 5
   3       XS0=(2.D0*FLOAT(I)-1.D0)*PI/2.D0+ERR
           XS1=X0
   5       CONTINUE
           IF (DABS(XS0-XS1)-ERR) 22,7,7
   7       XSA=(XS0+XS1)/2.D0
           Y=-DTAN(XSA)*DSQRT(X0**2-XSA**2)-XSA
           IF (Y) 8,9,10
   9       XS(I)=XSA
           GO TO 222
```

```
  8        XS1=XSA
           GO TO 5
 10        XS0=XSA
           GO TO 5
 22        XS(I)=(XS0+XS1)/2.D0
222        XS(I)=DSQRT(AK**2-XS(I)**2/DP**2)
  2   CON 'INUE
C-------- -----------------------------------------------
C     WRI E (6,301) ER,H
C301  FOR AT(//10X,' Dielectric Constant=',D16.9/10X,'Substrate '
C    *,'T ickness',D16.9///)
C-------- -----------------------------------------------
310   IF  NE.EQ.0) WRITE (6,304)
304   FORM T(/////10X,'No TE waves excited in the substrate'//)
      IF ( E.EQ.0) GO TO 312
      IF ( E.GT.0) WRITE (6,305) NE
305   FORM T(///10X,'There are',I4,
     *' TE waves excited in the substrate'//)
      DO 3 2 I=1,NE
             TEP(I)=XS(I)/AER
             IF (I.GT.1) THEN
                 I_MAX=I-1
                 DO 502 I_I=1,I_MAX
                    TEP_MIN=TEP(I_I)
                    IF (TEP(I).LT.TEP(I_I)) THEN
                        TEP(I_I)=TEP(I)
                        TEP(I)=TEP_MIN
                    END IF
502              CONTINUE
             END IF
302   CONTIN E
      DO 503 I=1,NE
             WRITE (6,303) I,TEP(I)
303          FORMAT (10X,I4,2X,D16.9)
503   CONTINU
312   CONTINU
C
C     END OF  ART I
C
C
C     PART II   TM MODES
C
      AN=X0/PI 1.D0
      NM=AN
      DO 13 I= ,NM
            IF (X0-(2.D0*FLOAT(I)+1.D0)*PI/2.D0) 14,14,15
 15         XS. =FLOAT(I)*PI-PI/3.D0-0.01D0
            GO TO 16
 14         XS1 =X0
 16         XSC  FLOAT(I-1)*PI+ERR
 17         CON INUE
            IF  DABS(XS0-XS1)-ERR) 113,19,19
 19         XRA  (XS0+XS1)/2.D0
C------------- ---------------------------------------------
C           WRI E (6,301) XRA
C 301       FORM T(10X,'XRA=',E14.7/)
C------------- ---------------------------------------------
            Y=DS RT(ER)**2*(1.D0/DTAN(XRA))*DSQRT(X0**2-XRA**2)-XRA
            IF ( ) 20,21,24
 21         XR(I =XRA
            GO T  333
 20         XS1=X A
            GO TC 17
 24         XS0=X A
            GO TO 17
113         XR(I)  (XS0+XS1)/2.D0
```

```
 333        XR(I)=DSQRT(AK**2-XR(I)**2/DP**2)
  13  CONTINUE
      WRITE (6,307) NM
 307  FORMAT(///10X,'There are',I4,' TM waves excited in the substrate'/
     */)
      DO 308 I=1,NM
             TMP(I)=XR(I)/AER
             IF (I.GT.1) THEN
                 I_MAX=I-1
                 DO 508 I_I=1,I_MAX
                    TMP_MIN=TMP(I_I)
                    IF (TMP(I).LT.TMP(I_I)) THEN
                        TMP(I_I)=TMP(I)
                        TMP(I)=TMP_MIN
                    END IF
 508             CONTINUE
             END IF
 308  CONTINUE
      DO 506 I=1,NM
             WRITE (6,306) I,TMP(I)
 306         FORMAT (10X,I4,2X,D16.9)
 506  CONTINUE
 322  CONTINUE
C
      NK=NE+NM
      IF (NE.EQ.0) GO TO 350
      DO 411 IQW=1,NE
             TPO(IQW)=TEP(IQW)
             LOR(IQW)=1
  411 CONTINUE
  350 CONTINUE
      DO 412 IQW=1,NM
             TPO(NE+IQW)=TMP(IQW)
             LOR(NE+IQW)=0
  412 CONTINUE
C
      IF (NK.EQ.1) GO TO 416
      NNK=NK-1
      DO 415 IIP=1,NNK
             IK=IIP+1
             DO 413 IIF=IK,NK
                    QWR=TPO(IIP)
                    IIW=LOR(IIP)
                    IF (TPO(IIP).LT.TPO(IIF)) GO TO 413
                    TPO(IIP)=TPO(IIF)
                    LOR(IIP)=LOR(IIF)
                    TPO(IIF)=QWR
                    LOR(IIF)=IIW
  413        CONTINUE
  415 CONTINUE
      IF (LOR(1).EQ.0) IFIRST=0
      IF (LOR(1).EQ.1) IFIRST=1
      GO TO 417
C
  416 IFIRST=2
  417 CONTINUE
      RETURN
      END
```

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

```
                                                                          000
y   y    i       j        ddddd    i   eeeeee  l          k    k  0   0          ffffff  ttttt  n    n
 y y     i       j        d    d   i   e       l          k   k  0     0         f         t    nn   n
  y      i       j        d    d   i   eeeee   l          kkkk   0     0         fffff     t    n n  n
  y      i       j        d    d   i   e       l          k  k   0     0         f         t    n  n n
  y      i   j   j        d    d   i   e       l          k   k   0   0    ...   f         t    n   nn
  y      i    jjjj ______ ddddd    i   eeeeee  llllll ______ k    k  000    ...   f         t    n    n
                                                                            ...
```

//tera/users/katehi/tape/yij_diel_k0.ftn

LAST MODIFIED ON: 89/04/24 10:44 AM
FILE PRINTED: 89/04/24 11:03 AM

```
C     ....................  YIJ_DIEL_K0.FTN   ...............................
C
C     In this program AKK=AK0
C
C*********************************************************************
      SUBROUTINE YIJ_DIEL
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 CONST,GSK,FSK,GS1S2K
      COMPLEX YSD,YSW,CI
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/OUT/GS(250)
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INS,INS1S2
C
      COMMON/TEST/FSD(250)
C
      COMMON/B01/BJ0,BJ1
C
      WREAL=W
      W=WREAL*(1.D0+2.D0*WDELTA/WREAL)
C
C
C     Subroutine POLES evaluates the poles of the Green's function
C     and orders them according to their magnitude
C
C     IFIRST=0  :dominant mode is a TM wave
C            1  :dominant mode is a TE wave
C            2  :only one TM wave
C
      CALL SPOLES
C
C     This subroutines gives data for the numerical integration
C
      CALL DATA_SLOT
C
      CI=(0.00,1.00)
C
      NS=NS1
      IF (NS1.LT.NS2) NS=NS2
      MS=NS
      IF (NOFF.EQ.1) GO TO 50
          NS1S2=NS2+NSS2-1
          MS1S2=NS1S2
          IF (NS1S2.GT.200) NS1S2=200
C
```

```
   50  CONTINUE
       IF(NMAX(INS).LE.(NS+2)) NMAX(INS)=NS+2
       IF (NOFF.EQ.1) GO TO 51
           IF(NMAX(INS1S2).LE.(NS1S2+2)) NMAX(INS1S2)=NS1S2+2
C
C
   51  ADL=AKK*DLX
       YSIN=DSIN(ADL)
       YCOS=DCOS(ADL)
C
C      For the normalization of the current along the y axis
C
                   CVON=W*PI/2.D0
C
C      Computation of lamda-integration limits between 0 and A
C
C
       CALL LIMIT
C
C      Evaluation of the Green's function at different points
C      in the interval [0,A]. The Bessel function has been excluded
C
       CALL GREEN
C
C      Evaluation of the tail contribution (from a to infinity)
C
       CALL TAIL
C
       CONST=-(1.D0/CVON)*DSQRT(EER)/(480.D0*(PI**3)*YSIN*YSIN)
C
       WRITE (6,99) CONST
  99   FORMAT(10X,'CONST=',E14.7//)
       WRITE(6,9)
   9   FORMAT(///10X,'Contribution to admittance from the dielectric'///)
       WRITE(6,10) MS
   10  FORMAT(11X,I4)
       DO 11 K=1,MS
             YSD(K)=YSD(K)*CONST
             GSK=REAL(GS(K)-FSD(K))*CONST
             WRITE (6,30) K,YSD(K),GS(K),FSD(K)
   30        FORMAT(1X,'K=',I4/2X,'YS=',E14.7,2X,E14.7,2X,
      *      'GS=',E14.7,2X,'FSD=',E14.7/)
             YSD(K)=(YSD(K)+GSK)*CI
   11  CONTINUE
       DO 20 K=1,MS
             WRITE (6,12) YSD(K)
   12        FORMAT(10X,E14.7,1X,E14.7)
   20  CONTINUE
C
 1000  CONTINUE
       RETURN
       END
C...........................................................
C...........................................................
C      This subroutine evaluates the limits of integration in
C      the interval [0,A].
C      Specifically:
C           1) It divides the interval [0,k0] to 10 equal
C              subsections and then apply fixed-point Gaussian
C              Quadrature
C           2) It divides the interval [k0,k] into so many
C              subsections as the number of poles and in
C              such a way that each subsection includes one
C              pole only away from the ends of the subsection
C           3) It divides the interval [k,A] into 20 equal
C              subsections and then apply fixed-point Gaussian
```

```
C             Quadrature
C.......................................................................
      SUBROUTINE LIMIT
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL WSPE,WTPE,WSPM
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
C----------------------------------------------+
C     Step 1 : Evaluation of vector CN         |
C              it gives the end points of the  |
C              intervals considered in (0,k0)  |
C----------------------------------------------+
      DELTA=AK0/FLOAT(NK0)
      CN(1)=0.D0
      DO 1 I=1,NK0
           CN(I+1)=DELTA*FLOAT(I)
  1   CONTINUE
C----------------------------------------------+
C     Step 2 : Evaluation of vector BM         |
C              it gives the end points of the  |
C              intervals considered in (k,A)   |
C----------------------------------------------+
      DELTA=(A/DSQRT(EER)-AK)/FLOAT(MA)
      BM(1)=AK
      DO 2 I=1,MA
           BM(I+1)=DELTA*FLOAT(I)+AK
  2   CONTINUE
C----------------------------------------------+
C     Step 3 : Evaluation of the vectors AM,DM |
C              "AM" gives the end points around|
C                   the TM poles               |
C              "DM" gives the end points around|
C                   the TE poles               |
C                                              |
C     IFIRST=   2   only one TM pole           |
C               1   TE0<TM0                    |
C               0   TM0<TE0                    |
C----------------------------------------------+
      AM(1)=AK0
      DM(1)=AK0
      NMAX=NTE+NTM-1
      IF (IFIRST.EQ.2) GO TO 3
      DO 4 I=1,NMAX
           AM(I+1)=(POLES(I+1)+POLES(I))/2.D0
           DM(I+1)=AM(I+1)
   4  CONTINUE
      AM(NMAX+2)=AK
      DM(NMAX+2)=AK
      IF (IFIRST.EQ.1) GO TO 5
          DM(NMAX+1)=AM(NMAX+2)
          DO 6 I=1,NMAX
               DM(NMAX-I+1)=AM(NMAX-I+2)
  6       CONTINUE
          GO TO 7
  5       AM(NMAX+1)=DM(NMAX+2)
          DO 8 I=1,NMAX
               AM(NMAX-I+1)=DM(NMAX-I+2)
  8       CONTINUE
          GO TO 7
C
```

```
  3        DELTA=(AK-AK0)/FLOAT(NK0K)
           AM(1)=AK0
           DO 9 I=1,NK0K
                AM(I+1)=DELTA*FLOAT(NK0K)+AK0
  9        CONTINUE
  7        CONTINUE
C---------------------------------------------+
C     Step 4 : evaluation of vectors VZXE     |
C---------------------------------------------+
      IF (IFIRST.EQ.2) GO TO 10
          DO 11 I=1,NTE
                ARG=POLTE(I)
                VZXE(I)=HZXE(ARG)
  11       CONTINUE
  10  CONTINUE
 ---------------------------------------------+
      Step 5 : evaluation of vector VXXM,VZXM |
 ---------------------------------------------+
      DO 12 I=1,NTM
            ARG=POLTM(I)
            VXXM(I)=GXXM(ARG)
            VZXM(I)=GZXM(ARG)
  12  CONTINUE
      RETURN
      END
C ..........................................................
C ..........................................................
C     This subroutine evaluates the values of the integrand of
C     the Green's function at different points in the interval
C     [0,A]. Then it evaluetes the space integrals of the Bessel
C     function at the same points and multiply these values with
C     the corresponding values of the Green's function.
C     Finally , it multiplies these products with known coeffic.
C     and it adds them up. This way, the moments'-method
C     space integrals of the first part of the Green's function are
C     evaluated and are stored in the complex vectors ZS,ZS1S2
C.........................................................
C.. ......................................................
      SUBROUTINE GREEN
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YSD,YSW,CI
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),SERA(5),
     *DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
      AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
      POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
      BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INS,INS1S2
C
C----- ----------------------------------------+
C     Evaluation of the coefficients for the  |
C            FF's functions                   |
C------ ---------------------------------------+
```

```
      F1X=1.0
      F1Z=2.D0*(1.D0-ER)/((1.D0+ER)*(1.D0+E2)*(1.D0+0.5D0*E1))
      IF ((ER-1.D0).LT.0.005) F1Z=0.D0
C
      CALL ARIS
C
      DO 1 I=1,NPOINT
            INCON=I
            IY=I
            AI=COAL(I)
            TI=POINT(I)
C
C     evaluation of intervals 1 and 2
C
            IAD=1
            DO 2 N=1,NK0
                 AUP=CN(N+1)
                 ALOW=CN(N)
                 CALL FUNCT(IAD,AUP,ALOW,N,INCON)
   2        CONTINUE
C
C     evaluation of intervals 3 and 4
C
            NTTM=NTM
            IF (IFIRST.EQ.2) NTTM=NK0K
            DO 3 IAD=3,4
                 IFD=0
                 DO 4 N=1,NTTM
                      IFD=IFD+1
                      AUP=AM(IFD+1)
                      ALOW=AM(IFD)
                      CALL FUNCT(IAD,AUP,ALOW,N,INCON)
                      IFD=IFD+1
   4             CONTINUE
   3        CONTINUE
            IF (IFIRST.EQ.2) GO TO 9
C
C     evaluation of the intervals 5 and 6,9,11
C
                 DO 5 IAD=5,6
                 IFD=0
                 DO 6 N=1,NTE
                      IFD=IFD+1
                      AUP=DM(IFD+1)
                      ALOW=DM(IFD)
                      CALL FUNCT(IAD,AUP,ALOW,N,INCON)
                      IFD=IFD+1
   6             CONTINUE
   5        CONTINUE
   9        CONTINUE
C
C     evaluation of the interval 7
C
            IAD=7
            DO 7 N=1,MA
                 AUP=BM(N+1)
                 ALOW=BM(N)
                 CALL FUNCT(IAD,AUP,ALOW,N,INCON)
  7         CONTINUE
  1   CONTINUE
C
C
C     evaluation of the intervals 8,10
C
      IAD=8
      IFD=0
```

```
      DO 8 N=1,NTM
           IFD=IFD+1
           AUP=AM(IFD+1)
           ALOW=AM(IFD)
           CALL FUNCT(IAD,AUP,ALOW,N,INCON)
           IFD=IFD+1
    8 CONTINUE
      RETURN
      END
C.............................................................
C             Functions :  GXXM,GZXM,HZXE
C
C     These functions evaluate the residues from the different poles
C.............................................................
      FUNCTION GXXM(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RM0H=RM0*H
      RMT=RM*(-H+T)
      SXN=RM*DCOS(RMT)-ER*RM0*DSIN(RMT)
      SXD=(ER+RM0H)*(RM/RM0)*DCOS(RMH)+(1.D0+ER*RM0H)*DSIN(RMH)
      GXXM=SXN/SXD
      RETURN
      END
C
C..............................................................
C
      FUNCTION GZXM(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RM0H=RM0*H
      RMT=RM*T
      CST=DCOS(RMT)
      CSH=DCOS(RMH)
      SNH=DSIN(RMH)
      SXN=RM*CST
      SXD=(RM*CSH+RM0*SNH)*((ER+RM0H)*CSH/RM0+(1.D0+ER*RM0H)*SNH/RM)
      GZXM=SXN/SXD
      RETURN
      END
C
C
C............................................................
C
      FUNCTION HZXE(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
```

```
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      X2=X*X
      AK02=AK0*AK0
      AK2=AK*AK
      RM=DSQRT(AK2-X2)
      RM0=DSQRT(X2-AK02)
      RMH=RM*H
      RMT=RM*T
      RM0H=RM0*H
      CSH=DCOS(RMH)
      CST=DCOS(RMT)
      SNH=DSIN(RMH)
      SXN=RM*CST
      SXD=(ER*RM0*CSH-RM*SNH)*(1.D0+RM0H)*(SNH/RM0-CSH/RM)
      HZXE=SXN/SXD
      RETURN
      END
C.............................................................
C     1) This subroutine evaluates the integrand of the Green's
C        function at different points (subroutine Grei).
C     2) It evaluates the space integrals comming from the
C        application of moments' method (subroutine adonis)
C     3) Multiply these two valueswith appropriate weighting
C        coefficients and it adds them upZXX2*SAJ0(K)
C.............................................................
      SUBROUTINE FUNCT(IAD,AUP,ALOW,N,INCON)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 S1,S2
      COMPLEX YSD,YSW,CI
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/IOFF/INS,INS1S2
C
      COMMON/TEST/FSD(250)
C
      CI=(0.0,1.0)
      NCON=0
      X=AUP-ALOW
      Y=AUP+ALOW
      AK02=AK0*AK0
      AK2=AK*AK
      AKK2=AKK*AKK
      ER1=1.D0-ER
      IF (IAD.GT.2) GO TO 1
         ALI=0.5D0*(TI*X+Y)
         GCONX=AI*X*0.5D0
         FCONX=GCONX
         GCONZ=GCONX*ER1
```

```
         IF(DABS(ER1).LT.0.005) GCONZ=0.D0
         FCONZ=FCONX
         AIMA=1.D0
         CALL GREI(ALI,0.D0,0.D0,IAD,0.D0)
         GO TO 10
 1    IF (IAD.NE.3) GO TO 2
         ALI=0.5D0*(TI*X+Y)
         XTM=POLTM(N)
         TMTM=(2.D0*XTM-Y)/X
         GCONX=AI/(TI-TMTM)
         GCONZ=GCONX*ER1
         FCONX=AI*X*0.5D0
         FCONZ=FCONX
         AIMA=0.D0
         IF (DABS(ER1).LT.0.005) THEN
                GCONX=0.0
                GCONZ=0.0
                FCONX=0.0
                FCONZ=0.0
         END IF
         CALL GREI(ALI,XTM,0.D0,IAD,0.D0)
         GO TO 10
 2    IF (IAD.NE.4) GO TO 3
         ALI=POLTM(N)
         TM=(2.D0*ALI-Y)/X
         GCONX=-AI/(TI-TM)
         GCONZ=GCONX*ER1
         FCONX=0.D0
         FCONZ=0.D0
         AIMA=0.D0
         RX=VXXM(N)
         RZ=VZXM(N)
         IF (DABS(ER1).LT.0.005) THEN
                GCONX=0.0
                GCONZ=0.0
                FCONX=0.0
                FCONZ=0.0
         END IF
         GO TO 10
 3    IF (IFIRST.EQ.2) GO TO 5
      IF (IAD.NE.5) GO TO 4
         ALI=0.5D0*(TI*X+Y)
         XTE=POLTE(N)
         TMTE=(2.D0*XTE-Y)/X
         GCONX=AI*X*0.5D0
         GCONZ=AI*ER1/(TI-TMTE)
         FCONX=GCONX
         FCONZ=FCONX
         AIMA=0.D0
         IF (DABS(ER1).LT.0.005) THEN
                GCONX=0.0
                GCONZ=0.0
                FCONX=0.0
                FCONZ=0.0
         END IF
         CALL GREI(ALI,0.D0,XTE,IAD,TMTE)
         GO TO 10
 4    IF (IAD.NE.6) GO TO 5
         NCON=6
         ALI=POLTE(N)
         TM=(2.D0*ALI-Y)/X
         GCONX=0.D0
         GCONZ=-AI*ER1/(TI-TM)
         FCONX=0.D0
         FCONZ=0.D0
         AIMA=0.D0
```

```
      RZ=VZXE(N)
      IF (DABS(ER1).LT.0.005) THEN
             GCONX=0.0
             GCONZ=0.0
             FCONX=0.0
             FCONZ=0.0
      END IF
      GO TO 10
 5    IF (IAD.NE.7) GO TO 6
      ALI=0.5D0*(TI*X+Y)
      GCONX=AI*X*0.5D0
      GCONZ=GCONX*ER1
      IF(DABS(ER1).LT.0.005) GCONZ=0.0
      FCONX=GCONX
      FCONZ=FCONX
      AIMA=0.D0
      CALL GREI(ALI,0.D0,0.D0,IAD,0.D0)
      GO TO 10
 6    NCON=8
      ALI=POLTM(N)
      TM=(2.D0*ALI-Y)/X
      FCONX=0.D0
      FCONZ=0.D0
      AIMA=0.D0
      RX=VXXM(N)
      RZ=VZXM(N)
      GO TO 28
C
 10   CONTINUE
      GXXR=GCONX*RX
      FXXR=FCONX*FRX
      GXXX=AIMA*GCONX*XX
      GZXR=GCONZ*RZ
      FZXR=FCONZ*FRZ
      GZXX=AIMA*GCONZ*XZ
 27   CONTINUE
      VARX=(AK2-AKK2)*GXXR+AKK2*GZXR
      FARX=(AK2-AKK2)*FXXR+AKK2*FZXR
      VARZ=AKK*(GXXR-GZXR)
      FARZ=AKK*(FXXR-FZXR)
      GXXR=VARX
      FXXR=FARX
      GZXR=VARZ
      FZXR=FARZ
      VARX=(AK2-AKK2)*GXXX+AKK2*GZXX
      VARZ=AKK*(GXXX-GZXX)
      GXXX=VARX
      GZXX=VARZ
      PLI=ALI
C
      CALL ADONIS
      DO 13 K=1,NS
         S1=REAL(GXXR*SSJ0(K,INS)+GZXR*SAJ0(K,INS))
         FS1=REAL(FXXR*SSJ0(K,INS)+FZXR*SAJ0(K,INS))
         S2=REAL(GXXX*SSJ0(K,INS)+GZXX*SAJ0(K,INS))
         YSD(K)=YSD(K)+S1-CI*S2
         FSD(K)=FSD(K)+FS1
C
C          IF (K.EQ.1) THEN
C              WRITE (6,966) NCON,IAD,ALI
C 966          FORMAT(10X,'NCON=',I4,2X,'IAD=',I4,2X,
C     *        'ALI=',E14.7/)
C              WRITE (6,866) FXXR,FZXR,FS1
C 866          FORMAT(10X,'FXXR=',E14.7,2X,'FZXR=',E14.7,
C     *        2X,'FS1=',E14.7/)
C              WRITE (6,766) YSD(K),FSD(K)
```

```
C 766             FORMAT(10X,'YSD=',E14.7,1X,E14.7,2X,'FSD=',E14.7//)
C           END IF
C
   13  CONTINUE
   28  IF (NCON.EQ.0) GO TO 24
       IF(INCON.LT.NPOINT) GO TO 24
           GCONX1=0.0
           GCONX2=0.0
           GCONZ1=ER1*DLOG((1.D0-TM)/(1.D0+TM))
           GCONZ2=ER1*PI
           IF (NCON.EQ.6) GO TO 29
               GCONX1=DLOG((1.D0-TM)/(1.D0+TM))
               GCONX2=PI
   29      CONTINUE
               GXXR=GCONX1*RX
               GXXX=GCONX2*RX
               GZXR=GCONZ1*RZ
               GZXX=GCONZ2*RZ
               FXXR=0.0
               FZXR=0.0
           IF (DABS(ER1).LT.0.005) THEN
                 GXXR=0.0
                 GXXX=0.0
                 GZXR=0.0
                 GZXX=0.0
           END IF
   25      CONTINUE
           NCON=0
           GO TO 27
   24      CONTINUE
           RETURN
           END
C...............................................................
C     This subroutine evaluates the integrand of the green's
C     function at different points
C...............................................................
      SUBROUTINE GREI(X,XFM,XFE,IAD,TM)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      X2=X*X
      AK2=AK*AK
      AK02=AK0*AK0
      RM=DSQRT(DABS(AK2-X2))
      RM0=DSQRT(DABS(X2-AK02))
      RMH=RM*H
      RMT=RM*T
      RMHT=RM*(-H+T)
C
      CSH=DCOS(RMH)
      SNH=DSIN(RMH)
      CST=DCOS(RMT)
      SNT=DSIN(RMT)
      CSHT=DCOS(RMHT)
      SNHT=DSIN(RMHT)
C
      RM2=RM*RM
      RM02=RM0*RM0
      CSH2=CSH*CSH
      ERM0=ER*RM0
      ERM02=ERM0*ERM0
C
```

```
      EXX=DEXP(-X*T/FA)/FA
      EXZ=DEXP(-X*2.D0*H/FA)/FA
      IF (IAD.NE.7) GO TO 100
          EX=DEXP(RMH)
          TANH=(EX-1.D0/EX)/(EX+1.D0/EX)
          CSHH=(EX+1.D0/EX)/2.D0
          EX=DEXP(RMT)
          CSHT=0.5D0*(EX+1.D0/EX)
          SNHT=0.5D0*(EX-1.D0/EX)
          TANT=SNHT/CSHT
          EX=DEXP(RMHT)
          CSHHT=0.5D0*(EX+1.D0/EX)
          SNHHT=0.5D0*(EX-1.D0/EX)
          TANHT=SNHHT/CSHHT
C
  100 IF (IAD.NE.1) GO TO 1
          DEN=RM2+(ERM02-RM2)*CSH2
          RNOM=-RM2*SNT+(RM2-ERM02)*CSH*SNHT
          XNOM=ER*RM*RM0*CST
          C1=X/RM
C
          RX=C1*RNOM/DEN
          IF((ER-1.D0).LT.0.005) RX=0.D0
          XX=C1*XNOM/DEN
          FRX=F1X*EXX
C
          DEN=DEN*(RM02+AK02*(ER-1.D0)*CSH2)
          RNOM=-CST*(RM2+ER*RM02)*CSH*SNH
          XNOM=CST*RM*RM0*(-1.D0+(1.D0+ER)*CSH2)
          C1=X*RM
          RZ=-C1*RNOM/DEN
          XZ=C1*XNOM/DEN
          FRZ=F1Z*EXZ
          RETURN
  1   IF (IAD.NE.3) GO TO 2
          C1=X-XFM
          IF (DABS(AK-X).LT.1.D-6) GO TO 10
              DEN=ERM0*CSH-RM*SNH
              RNOM=(RM*CSHT-ERM0*SNHT)
              C2=X/RM
              RX=C1*C2*RNOM/DEN
C
              DEN=DEN*(RM*CSH+RM0*SNH)
              RNOM=CST
              C3=X*RM
              RZ=C1*C3*RNOM/DEN
C
              FRX=F1X*EXX
              FRZ=F1Z*EXZ
              RETURN
C
   10     RNOM=1.D0-ERM0*(-H+T)
          RX=C1*X*RNOM/ERM0
          FRX=F1X*EXX
C
          RZ=X*C1/(ERM0*(1.D0+RM0*H))
          FRZ=F1Z*EXZ
          RETURN
  2   IF (IAD.NE.5) GO TO 4
          C1=X-XFE
          IF (DABS(AK-X).LT.1.D-6) GO TO 13
              RNOM=RM*CSHT-ERM0*SNHT
              DEN=ERM0*CSH-RM*SNH
              RX=(X/RM)*RNOM/DEN
              FRX=F1X*EXX
C
```

```
            RNOM=RM*CST
            DEN=DEN*(RM*CSH+RM0*SNH)
            RZ=X*C1*RNOM/DEN
            FRZ=F1Z*EXZ
            RETURN
 13      RX=X*(1.D0-ERM0*(-H+T))/ERM0
         FRX=F1X*EXX
         RZ=X*C1/(ERM0*(1.D0+RM0*H))
         FRZ=F1Z*EXZ
         RETURN
  4   IF (IAD.NE.7) GO TO 6
         IF (DABS(X-AK).LT.1.D-6) GO TO 15
            DEN=ERM0+RM*TANH
            RNOM=(RM+ERM0*TANH)*CSHT-DEN*SNHT
            RX=(X/RM)*RNOM/DEN
            FRX=F1X*EXX
C
            RNOM=X*(RM*CSHT)/(CSHH*CSHH)
            DEN=DEN*(RM+RM0*TANH)
            RZ=RNOM/DEN
            FRZ=F1Z*EXZ
            RETURN
 15   RX=X*(1.D0-ERM0*(-H+T))/ERM0
      FRX=F1X*EXX
      RZ=(X/ERM0)/(1.D0+RM0*H)
      FRZ=F1Z*EXZ
  6   CONTINUE
      RETURN
      END
C.................................................................
C                               ARIS
C.................................................................
      SUBROUTINE ARIS
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
C     +------------------------------------------+
C     | Formation of the matrices:  DIST,        |
C     |     DARG,RCOE                            |
C     |                                          |
C     +------------------------------------------+
      W2=W/2.D0
      U=WREAL/W
      THMIN=DATAN(DSQRT(1.D0/(U*U)-1.D0))
      THMAX=PI-THMIN
      AX=(THMAX-THMIN)/2.D0
      BX=(THMAX+THMIN)/2.D0
      X=PI/4.D0
      DO 1 J=1,NOFF
           MAX=NMAX(J)
           LPOINT=MPOINT
           IF (OFFSET(J).LE.OFFLIM) LPOINT=NPOINT
           DO 2 I=1,LPOINT
                POIN=BPOINT(I)
                IF (OFFSET(J).LE.OFFLIM) POIN=POINT(I)
```

```
                FI=X*(POIN+1.D0)
                THETA=AX*POIN+BX
                AS=DSIN(FI)
                AC=DCOS(FI)
                DARG(I,1)=W2*AC
                DARG(I,2)=AC
                DARG(I,3)=AS
                DARG(I,4)=X
                DO 3 N=1,MAX
                     AXN=FLOAT(N-2)*DLX
                     IF (OFFSET(J).GT.OFFLIM) GO TO 4
                        DIST(N,J,I)=AXN*AS
                        GO TO 5
  4                  AXN2=AXN*AXN
                     BXN=OFFSET(J)-W*DCOS(THETA)/2.D0
                     BXN2=BXN*BXN
                     DIST(N,J,I)=DSQRT(AXN2+BXN2)
                     SIG=DIST(N,J,I)
                     SIG2=SIG*SIG
                     SIG3=SIG2*SIG
                DSIG=DABS(AXN)/SIG
                DSIG2=BXN2/SIG3
                DSIG3=-3.D0*DSIG*DSIG2/SIG
                DSIG4=-3.D0*DSIG2*(DSIG2-4.D0*DSIG**2/SIG)/SIG
                DSIG5=-3.D0*(-15.D0*DSIG2**2*DSIG+(20.D0/SIG)*
     *                DSIG2*DSIG**3)/SIG2
                DSIG6=-3.D0*(-15.D0*DSIG2**3+(180.D0/SIG)*DSIG2
     *                **2*DSIG**2-(120.D0/SIG2)*DSIG2*DSIG**4)/
     *                SIG2
                DSIG7=-3.D0*(525.D0*DSIG2**3*DSIG-(2100.D0/SIG)*
     *                DSIG2**2*DSIG**3+(840.D0/SIG2)*DSIG2*DSIG
     *                **5)/SIG3
                DSIG8=-3.D0*(525.D0*DSIG2**4-(12600.D0/SIG)*DSIG2
     *                **3*DSIG**2+(25200.D0/SIG2)*DSIG2**2*DSIG**4
     *                -(6720.D0/SIG3)*DSIG2*DSIG**6)/SIG3
C
C           Evaluation of the coefficients Gij
C
                G21=DSIG2
                G22=DSIG**2
C----------------
                G41=DSIG4
                G42=4.D0*DSIG3*DSIG+3.D0*DSIG2**2
                G43=6.D0*DSIG2*DSIG**2
                G44=DSIG**4
C----------------
                G61=DSIG6
                G62=6.D0*DSIG5*DSIG+15.D0*DSIG4*DSIG2+10.D0*DSIG3**2
                G63=15.D0*DSIG4*DSIG**2+60.D0*DSIG3*DSIG2*DSIG+15.D0
     *              *DSIG2**3
                G64=20.D0*DSIG3*DSIG**3+45.D0*DSIG2**2*DSIG**2
                G65=15.D0*DSIG2*DSIG**4
                G66=DSIG**6
C----------------
                G81=DSIG8
                G82=8.D0*DSIG7*DSIG+28.D0*DSIG6*DSIG2+56.D0*DSIG5
     *              *DSIG3+35.D0*DSIG4**2
                G83=28.D0*DSIG6*DSIG**2+168.D0*DSIG5*DSIG2*DSIG+
     *              280.D0*DSIG4*DSIG3*DSIG+210.D0*DSIG4*DSIG2**2+
     *              280.D0*DSIG3**2*DSIG2
                G84=56.D0*DSIG5*DSIG**3+420.D0*DSIG4*DSIG2*DSIG**2
     *              +280.D0*DSIG3**2*DSIG**2+840.D0*DSIG3*DSIG2**2
     *              *DSIG+105.D0*DSIG2**4
                G85=70.D0*DSIG4*DSIG**4+560.D0*DSIG3*DSIG2*DSIG**3
     *              +420.D0*DSIG2**3*DSIG**2
                G86=56.D0*DSIG3*DSIG**5+210.D0*DSIG2**2*DSIG**4
```

```
                G87=28.D0*DSIG2*DSIG**6
                G88=DSIG**8
C----------------
                RCOE(2,N,J,I)=-0.5D0*(G22+SIG*G21)
                RCOE(1,N,J,I)=0.5D0*(G22-SIG*G21)
C----------------
                SX=0.5D0*SIG*(G42-SIG*G41)
                S30=-0.5D0*SIG*(G42+SIG*G41)
                S31=0.25D0*(SX+3.D0*G43)
                S33=0.25D0*(SX-G43)
                RCOE(3,N,J,I)=0.5D0*(SIG*S33/3.D0+G44/4.D0)
                RCOE(4,N,J,I)=0.5D0*(SIG*S31+SIG*S33/3.D0-G44)
                RCOE(5,N,J,I)=0.5D0*(SIG*S31+3.D0*G44/4.D0)
                RCOE(6,N,J,I)=SIG*S30
C----------------
                SX=SIG*S33/3.D0+G64/4.D0
                ST=SIG*S31+SIG*S33/3.D0-G64
                S5M3=SIG2*S30
                S5M1=0.5D0*SIG*(SIG*S31+3.D0*G64/4.D0)
                S51=0.25D0*(0.5D0*SIG*ST-5.D0*G65/2.D0)
                S53=0.25D0*(0.5D0*SIG*ST+0.25D0*SIG*SX+0.5D0*G65/
     *              4.D0)
                S55=0.125D0*(0.5D0*SIG*SX-0.5*G65)
                RCOE(7,N,J,I)=0.5D0*(SIG*S55/5.D0+G66/16.D0)
                RCOE(8,N,J,I)=0.5D0*(SIG*S53/3.D0+SIG*S55/5.D0-
     *                      6.D0*G66/16.D0)
                RCOE(9,N,J,I)=0.5D0*(SIG*S51+SIG*S53/3.D0+15.D0*
     *                      G66/16.D0)
                RCOE(10,N,J,I)=0.5D0*(SIG*S51-10.D0*G66/16.D0)
                RCOE(11,N,J,I)=SIG*S5M1
                RCOE(12,N,J,I)=SIG*S5M3
C----------------
                S7M5=SIG2*S5M3
                S7M3=SIG2*S5M1
                S7M1=0.5D0*SIG*(SIG*S51-10.D0*G86/16.D0)
                S71=0.5D0*(0.25D0*SIG*(SIG*S51+SIG*S53/3.D0+
     *              15.D0*G86/16.D0)+35.D0*G87/32.D0)
                S73=0.5D0*(0.25D0*SIG*(SIG*S51+SIG*S53/3.D0+15.D0
     *              *G86/16.D0)+0.125D0*SIG*(SIG*S53/3.D0+SIG*
     *              S55/5.D0-6.D0*G86/16.D0)-21.D0*G87/32.D0)
                S75=0.5D0*(0.125D0*SIG*(SIG*S53/3.D0+SIG*S55/5.D0-
     *              6.D0*G86/16.D0)+(SIG/12.D0)*(SIG*S55/5.D0+
     *              G86/16.D0)+7.D0*G87/32.D0)
                S77=0.5D0*((SIG/12.D0)*(SIG*S55/5.D0+G86/16.D0)-
     *              G87/32.D0)
                RCOE(13,N,J,I)=0.5D0*(SIG*S77/7.D0+G88/64.D0)
                RCOE(14,N,J,I)=0.5D0*(SIG*S75/5.D0+S77*SIG/7.D0
     *                      -8.D0*G88/64.D0)
                RCOE(15,N,J,I)=0.5D0*(SIG*S73/3.D0+SIG*S75/5.D0
     *                      +28.D0*G88/64.D0)
                RCOE(16,N,J,I)=0.5D0*(SIG*S71+SIG*S73/3.D0-56.D0
     *                      *G88/64.D0)
                RCOE(17,N,J,I)=0.5D0*(SIG*S71+35.D0*G88/64.D0)
                RCOE(18,N,J,I)=SIG*S7M1
                RCOE(19,N,J,I)=SIG*S7M3
                RCOE(20,N,J,I)=SIG*S7M5
    5               CONTINUE
    3           CONTINUE
    2       CONTINUE
    1   CONTINUE
C
C             Formation of the series s(dlx) . Storage in
C                 vectors SERS(5),SERA(5)
C
        U1=2.D0*THMIN/FLOAT(NSER)
        DO 6 JN=1,NSER
```

```
            S2=(2.D0*FLOAT(JN)-1.D0)
            S2=S2/(2.D0*FLOAT(NSER))
            S3=DCOS(S2*THMIN)
            S(JN,2)=S3*W/2.D0
            S(JN,1)=U1
  6   CONTINUE
      ADL=AKK*DLX
      ADL2=ADL*ADL
      ADL3=ADL2*ADL
      ADL4=ADL3*ADL
      ADL5=ADL4*ADL
      ADL6=ADL5*ADL
      YSIN=DSIN(ADL)
      YCOS=DCOS(ADL)
C
      SER1=(1.D0-YCOS)*2.D0/AKK
C
      SER2=-YSIN/3.D0+ADL*YCOS/4.D0+ADL2*YSIN/10.D0-ADL3*YCOS/36.D0
     *     -ADL4*YSIN/168.D0+ADL5*YCOS/960.D0+ADL6*YSIN/6480.D0
C
      SER3=YSIN/60.D0-ADL*5.D0*YCOS/360.D0-ADL2*YSIN/168.D0+ADL3
     *     *YCOS/560.D0+ADL4*YSIN/2592.D0-ADL5*YCOS/12960.D0-ADL6
     *     *YSIN/95040.D0
C
      SER4=-YSIN/2520.D0+ADL*YCOS/2880.D0+ADL2*YSIN/6480.D0-ADL3
     *     *YCOS/21600.D0-ADL4*YSIN/95040.D0+ADL5*YCOS/518400.D0
C
      SER5=YSIN/181440.D0-ADL*YCOS/201600.D0-ADL2*YSIN/443520.D0+
     *     ADL3*YCOS/1442775.9D0
C
      SERS(1)=SER1*SER1
      SERS(2)=DLX*2.D0*SER1*SER2
      SERS(3)=DLX*(DLX*SER2*SER2+2.D0*SER1*SER3)
      SERS(4)=DLX*(2.D0*SER1*SER4+2.D0*DLX*SER2*SER3)
      SERS(5)=DLX*(DLX*SER3*SER3+2.D0*DLX*SER2*SER4)
C
      SERA(1)=SER1
      SERA(2)=DLX*SER2
      SERA(3)=DLX*SER3
      SERA(4)=DLX*SER4
      SERA(5)=DLX*SER5
  111 CONTINUE
      RETURN
      END
C.....................................................................
C                         ADONIS
C     This subroutine evaluates the space integarls of the bessel
C         function
C
C..............................................................
      SUBROUTINE ADONIS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2),DERIV(9,3)
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/PUT/SSJ0(250,7),SAJ0(250,7),YSIN,YCOS
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
```

```
      COMMON/BSS/ARG(10),AARG
C
      COMMON/MAT/PLI,AI,TI,V(3),IY
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      ARX=W*AX/2.D0
      W1=2.D0*YCOS
      PR1=PLI*DLX
      PR2=PR1*PR1
      PR4=PR2*PR2
      PR6=PR4*PR2
      PR8=PR6*PR2
      DO 1 J=1,NOFF
           MAX=NMAX(J)
           DO 2 N=1,MAX
                SSJ0(N,J)=0.D0
                SAJ0(N,J)=0.D0
  2        CONTINUE
  1   CONTINUE
C
      DO 11 J=1,NOFF
            LPOINT=MPOINT
            IF (OFFSET(J).GT.OFFLIM) GO TO 12
                LPOINT=NPOINT
                DO 13 I=1,NPOINT
                      ARG(I)=PLI*DARG(I,1)
 13             CONTINUE
                CALL BESS1(BJ)
 12         DO 14 I=1,LPOINT
                  DO 17 NK=1,5
                        DERIV(NK,1)=0.D0
                        DERIV(NK,2)=0.D0
 17               CONTINUE
                  ASIN=ARX*BCOAL(I)
                  IF (OFFSET(J).GT.OFFLIM) GO TO 15
                        ASIN=W*DARG(I,4)*COAL(I)
                        AROF=PLI*OFFSET(J)*DARG(I,2)
                        COFF=DCOS(AROF)
                        SSUM=0.D0
                        DO 16 JN=1,NSER
                              ARAF=PLI*S(JN,2)*DARG(I,2)
                              CAFF=DCOS(ARAF)
                              SSUM=SSUM+S(JN,1)*CAFF
 16                     CONTINUE
 15               CONTINUE
                  KMAX=NMAX(J)
                  DO 18 K=1,KMAX
                        DO 20 NK=1,5
                              DERIV(NK,1)=DERIV(NK,2)
                              DERIV(NK,2)=DERIV(NK,3)
 20                     CONTINUE
                        IF (OFFSET(J).GT.OFFLIM) GO TO 21
                              SIN1=DARG(I,3)
                              SIN2=SIN1*SIN1
                              COS1=DCOS(PLI*DIST(K,J,I))
                              TERM=COFF*(BJ(I,1)-SSUM/PI)*COS1
                              DERIV(1,3)=TERM
                              SIN1=SIN2
                              DERIV(2,3)=-TERM*SIN1
                              SIN1=SIN1*SIN2
                              DERIV(3,3)=TERM*SIN1
                              SIN1=SIN1*SIN2
                              DERIV(4,3)=-TERM*SIN1
                              SIN1=SIN1*SIN2
                              DERIV(5,3)=TERM*SIN1
```

```
                          GO TO 22
  21                AARG=PLI*DIST(K,J,I)
                    ARG2=AARG*AARG
                    ARG4=ARG2*ARG2
                    ARG6=ARG4*ARG2
                    CALL BESS2(BJ)
                    DERIV(1,3)=BJ(1,2)
                    DERIV(2,3)=RCOE(1,K,J,I)*BJ(3,2)+
     *                      RCOE(2,K,J,I)*BJ(1,2)
                    DERIV(3,3)=RCOE(3,K,J,I)*BJ(5,2)+
     *                      RCOE(4,K,J,I)*BJ(3,2)+(RCOE(5,K,J,I)
     *                      +RCOE(6,K,J,I)/ARG2)*BJ(1,2)
                    DERIV(4,3)=RCOE(7,K,J,I)*BJ(7,2)+
     *                      RCOE(8,K,J,I)*BJ(5,2)+RCOE(9,K,J,I)*
     *                      BJ(3,2)+(RCOE(10,K,J,I)+RCOE(11,K,
     *                      J,I)/ARG2+RCOE(12,K,J,I)/ARG4)*
     *                      BJ(1,2)
                    DERIV(5,3)=RCOE(13,K,J,I)*BJ(9,2)+
     *                      RCOE(14,K,J,I)*BJ(7,2)+RCOE(15,K,J,
     *                      I)*BJ(5,2)+RCOE(16,K,J,I)*BJ(3,2)+
     *                      (RCOE(17,K,J,I)+RCOE(18,K,J,I)/ARG2
     *                      +RCOE(19,K,J,I)/ARG4+RCOE(20,K,J,I)
     *                      /ARG6)*BJ(1,2)
  22                IF (K.LT.3) GO TO 18
                    SUMS=SERS(1)*DERIV(1,2)-PR2*SERS(2)*DERIV(2,2)
     *                   +PR4*SERS(3)*DERIV(3,2)-PR6*SERS(4)*DERIV
     *                   (4,2)+PR8*SERS(5)*DERIV(5,2)
C
                    CH1=SERA(1)*(DERIV(1,1)+DERIV(1,3)-W1*DERIV
     *                 (1,2))
                    CH2=SERA(2)*(DERIV(2,1)+DERIV(2,3)-W1*DERIV
     *                 (2,2))*PR2
                    CH3=SERA(3)*(DERIV(3,1)+DERIV(3,3)-W1*DERIV
     *                 (3,2))*PR4
                    CH4=SERA(4)*(DERIV(4,1)+DERIV(4,3)-W1*DERIV
     *                 (4,2))*PR6
                    CH5=SERA(5)*(DERIV(5,1)+DERIV(5,3)-W1*DERIV
     *                 (5,2))*PR8
                    SUMA=CH1-CH2+CH3-CH4+CH5
                    KJ=K-2
                    SSJ0(KJ,J)=SSJ0(KJ,J)+ASIN*SUMS
                    SAJ0(KJ,J)=SAJ0(KJ,J)+ASIN*SUMA
CCCC
C                   IF (KJ.EQ.1)WRITE (6,665) KJ,J,SSJ0(KJ,J),
C    *              SUMS,SAJ0(KJ,J),SUMA
C665                FORMAT(10X,'KJ=',I4,2X,'J=',I4/10X,'SSJ0=',
C    *              E14.7,2X,'SUMS=',E14.7/10X,'SAJ0=',E14.7,
C    *              2X,'SUMA=',E14.7/)
CCCC
  18            CONTINUE
  14        CONTINUE
  11  CONTINUE
      RETURN
      END
C.....................................................
C                          BESS1
C     This subroutine gives values for the zeroth order
C     Bessel functions. It is used for small offsets
C.....................................................
      SUBROUTINE BESS1(BJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2)
C
      COMMON/COEF/RX,XX,RZ,XZ,FRX,FRZ,F1X,F1Z
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
```

```
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/BSS/ARG(10),AARG
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      PI=3.141592653589D0
      DO 1 IJ=1,NPOINT
           X=ARG(IJ)
           IF (X.GT.0.001D0) GO TO 10
              X3=X/3.D0
              X32=X3*X3
              X34=X32*X32
              X36=X34*X32
              BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *            *X36
              BJ(IJ,1)=BJ0
              GO TO 1
  10       IF (X.GT.3.D0) GO TO 12
              X3=X/3.D0
              X32=X3*X3
              X34=X32*X32
              X36=X34*X32
              X38=X36*X32
              X310=X38*X32
              X312=X310*X32
              BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *            *X36+0.0444479D0*X38-0.0039444D0*X310+0.00021000
     *            D0*X312
              BJ(IJ,1)=BJ0
              GO TO 1
   12      CONTINUE
           X3=3.D0/X
           X32=X3*X3
           X33=X32*X3
           X34=X33*X3
           X35=X34*X3
           X36=X35*X3
           FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.0000
     *         9512D0*X33+0.00137237D0*X34-0.00072805D0*X35+0.00014
     *         476D0*X36
           TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00
     *         262573D0*X33-0.00054125D0*X34-0.00029333D0*X35+0.000
     *         13558D0*X36
           WCON=DSQRT(1.D0/X)
           BJ(IJ,1)=WCON*FJ0*DCOS(TJ0)
   1       CONTINUE
      RETURN
      END
C.......................................................................
C                                TAIL
C     This subroutine evaluates the tail contribution
C.......................................................................
      SUBROUTINE TAIL
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YSD,YSW
      DIMENSION MAX(8,2)
C
      COMMON/CTAIL/S1(4,205,7),D1(4,205,7),D2(4,205,7),
     *T1(4,205,7),T2(4,205,7),T3(4,205,7),T4(4,205,7)
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
```

```
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XNT(40,3),
     *CNT(40),NDP,NTP,NSP
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/OUT/GS(250)
C
      COMMON/IOFF/INS,INS1S2
C
C     This vector contains the values of t in the integrals h0
C
      Z1=T
      Z2=2.D0*H
C
C     This vector contains the values of the coefficient C in
C          the integrals h0
C
      C1=FA
      WRITE (*,111) FA
 111  FORMAT(///10X,'FA=',E14.7///)
C
C      This vector contains the number of elements of the
C      matrices ZS,ZS1S2,....
C
      MAX(1,1)=NS
      MAX(2,1)=NS1S2
C
      MAX(1,2)=INS
      MAX(2,2)=INS1S2
C
C
C     This vector contains the values of the coefficient A in
C              the integrals h0
C
      AK2=AK*AK
      AKK2=AKK*AKK
      AK02=AK0*AK0
      W2=W/2.D0
      THMIN=WREAL/W
      THMIN=DATAN(DSQRT(1.D0/THMIN**2-1.D0))
      THMAX=PI-THMIN
      PI2=PI/2.D0
      PI4=PI/4.D0
      DLX2=DLX/2.D0
      DLX4=DLX2*DLX2
C
      YCOS=DCOS(AKK*DLX)
      CCS=DCOS(2.D0*AKK*DLX)
      YSIN=DSIN(AKK*DLX)
      SSN=DSIN(2.D0*AKK*DLX)
C
C     +---------------------------------+
C     | Evaluation of S1,S2,S3,S4,S5,S6 |
C     |      (Single Integrals)         |
C     +---------------------------------+
C
C
      DO 201 J=1,7
             DO 202 K=1,205
                    DO 203 JK=1,4
                           S1(JK,K,J)=0.D0
                           D1(JK,K,J)=0.D0
                           D2(JK,K,J)=0.D0
```

```
                              T1(JK,K,J)=0.D0
                              T2(JK,K,J)=0.D0
                              T3(JK,K,J)=0.D0
                              T4(JK,K,J)=0.D0
  203                 CONTINUE
  202          CONTINUE
  201 CONTINUE
C
      ZP1=Z1/C1
      ZP2=Z2/C1
C
      ZP12=ZP1*ZP1
      ZP22=ZP2*ZP2
      DO 1 J=1,NOFF
           KMAX=NMAX(J)+2
           IF (OFFSET(J).LT.1.D-6) THMAX=PI
           DSP=(THMAX-THMIN)/4.D0
           DDP=DSP*DLX2
           DTP=DSP*DLX4
           COEF1=(THMAX-THMIN)/2.D0
           IF (OFFSET(J).LT.1.D-6) COEF1=(PI/2.D0-THMIN)/2.D0
           COEF2=(THMAX+THMIN)/2.D0
           IF (OFFSET(J).LT.1.D-6) COEF2=(PI/2.D0+THMIN)/2.D0
           DO 10 I=1,NSP
                 THI=COEF1*XNS(I)+COEF2
                 C1=DCOS(THI)
                 C2=W2*C1
                 C2=OFFSET(J)-C2
                 CW=C2*C2
                 AASIN=CNS(I)*DSP
                 DO 11 K=1,KMAX
                       XN=(FLOAT(K-3)*DLX)
                       RAD2=XN*XN+CW
                       TRAD1=DSQRT(RAD2+ZP12)
                       TRAD2=DSQRT(RAD2+ZP22)
                       S1(1,K,J)=S1(1,K,J)+DLOG(2.D0*(TRAD1+XN))*AASIN
                       S1(2,K,J)=S1(2,K,J)+DLOG(2.D0*(TRAD2+XN))*AASIN
  11             CONTINUE
  10       CONTINUE
C
C     +---------------------------------------------------------------+
C     | EVALUATION OF D1,D2,D4,D5                                    1
C     +---------------------------------------------------------------+
           DO 20 I=1,NDP
                 THI=COEF1*XND(I,1)+COEF2
                 XI=DLX2*(XND(I,2)+1.D0)
                 C1=DCOS(THI)
                 C2=W2*C1
                 C2=OFFSET(J)-C2
                 CW=C2*C2
                 AASIN=CND(I)*DDP
                 SV1=DSIN(AKK*(DLX-XI))
                 SV2=-SV1
                 SV4=DSIN(AKK*XI)
                 C2=DCOS(AKK*(DLX-XI))
                 DO 21 K=1,KMAX
                      XNP=(XI+FLOAT(K-2)*DLX)
                      XNM=(-XI+FLOAT(K-2)*DLX)
                      RADP2=XNP*XNP+CW
                      RADM2=XNM*XNM+CW
                      TRAP1=DSQRT(RADP2+ZP12)
                      TRAP2=DSQRT(RADP2+ZP22)
C
                      TRAM1=DSQRT(RADM2+ZP12)
                      TRAM2=DSQRT(RADM2+ZP22)
C
```

```
                    XA1=AKK*XNP
                    XA2=AKK*XNM
                    XAP=DSIN(XA1)
                    XAM=DSIN(XA2)
C
                    SANP1=XAP*DLOG(2.D0*(TRAP1+XNP))
                    SANP2=XAP*DLOG(2.D0*(TRAP2+XNP))
C
                    SANM1=XAM*DLOG(2.D0*(TRAM1+XNM))
                    SANM2=XAM*DLOG(2.D0*(TRAM2+XNM))
C
                    XAP=DSIN(XA1/2.D0)
                    XAM=DSIN(XA2/2.D0)
                    SONP1=XAP/TRAP1
                    SONP2=XAP/TRAP2
C
                    SONM1=XAM/TRAM1
                    SONM2=XAM/TRAM2
C
                    Y1=-XNM/2.D0-DLX
                    Y2=-XNP/2.D0+DLX
                    CY1=DCOS(AKK*Y1)
                    CY2=DCOS(AKK*Y2)
                    SY1=DSIN(AKK*Y1)
                    SY2=DSIN(AKK*Y2)
C
                    D1(1,K,J)=D1(1,K,J)+(SANP1+SANM1)*SV2*AASIN
                    D2(1,K,J)=D2(1,K,J)+(CY1*SONP1-CY2*SONM1)*AASIN
                    D1(2,K,J)=D1(2,K,J)+(SANP2+SANM2)*SV2*AASIN
                    D2(2,K,J)=D2(2,K,J)+(CY1*SONP2-CY2*SONM2)*AASIN
  21            CONTINUE
  20        CONTINUE
C
C       evaluation of T1,T2,T3,T4
C
            DO 30 I=1,NTP
                THI=COEF1*XNT(I,1)+COEF2
                XI=DLX2*(XNT(I,2)+1.D0)
                XIP=DLX2*(XNT(I,3)+1.D0)
                C1=DCOS(THI)
                C2=W2*C1
                C2=OFFSET(J)-C2
                CW=C2*C2
                SV1=DSIN(AKK*(DLX-XI))
                SV2=-SV1
                SV3=DSIN(AKK*(DLX-XIP))
                AASIN=DTP*CNT(I)
                DO 31 K=1,KMAX
                    XNPP=(XI+XIP)+FLOAT(K-1)*DLX
                    XNPM=(XI-XIP)+FLOAT(K-1)*DLX
                    XNMP=(-XI+XIP)+FLOAT(K-1)*DLX
                    XNMM=(-XI-XIP)+FLOAT(K-1)*DLX
                    RADPP2=XNPP*XNPP+CW
                    RADPM2=XNPM*XNPM+CW
                    RADMP2=XNMP*XNMP+CW
                    RADMM2=XNMM*XNMM+CW
                    TAPP1=DSQRT(RADPP2+ZP12)
                    TAPP2=DSQRT(RADPP2+ZP22)
                    TAPM1=DSQRT(RADPM2+ZP12)
                    TAPM2=DSQRT(RADPM2+ZP22)
                    TAMP1=DSQRT(RADMP2+ZP12)
                    TAMP2=DSQRT(RADMP2+ZP22)
                    TAMM1=DSQRT(RADMM2+ZP12)
                    TAMM2=DSQRT(RADMM2+ZP22)
                    CST1=DCOS(AKK*(XNPM/2.D0+DLX))*DSIN(AKK*XNPP
     *                  /2.D0)
```

```
                     CST2=DCOS(AKK*(-XNMP/2.D0+DLX))*DSIN(AKK*XNMM
     *                   /2.D0)
                     CST3=DCOS(AKK*(XNMM/2.D0+DLX))*DSIN(AKK*XNMP
     *                   /2.D0)
                     CST4=DCOS(AKK*(-XNPP/2.D0+DLX))*DSIN(AKK*XNPM
     *                   /2.D0)
                     T1(1,K,J)=T1(1,K,J)+SV2*AASIN*CST1/TAPP1
                     T2(1,K,J)=T2(1,K,J)+SV1*AASIN*CST2/TAMM1
                     T3(1,K,J)=T3(1,K,J)+SV1*AASIN*CST3/TAMP1
                     T4(1,K,J)=T4(1,K,J)+SV2*AASIN*CST4/TAPM1
                     T1(2,K,J)=T1(2,K,J)+SV2*AASIN*CST1/TAPP2
                     T2(2,K,J)=T2(2,K,J)+SV1*AASIN*CST2/TAMM2
                     T3(2,K,J)=T3(2,K,J)+SV1*AASIN*CST3/TAMP2
                     T4(2,K,J)=T4(2,K,J)+SV2*AASIN*CST4/TAPM2
   31             CONTINUE
   30        CONTINUE
    1 CONTINUE
C
C
C     Evaluation of GS,GS1S2
C
C
      CZX=2.D0*(1.D0-ER)/((1.D0+ER)*(1.D0+E2)*(1.D0+0.5D0*E1))
      IF((ER-1.D0).LT.0.005) CZX=0.D0
      CXX=1.D0
      CSX=(AK2-AKK2)*CXX/FA
      CSZ=AKK2*CZX/FA
      CAX=AKK*CXX/FA
      CAZ=AKK*CZX/FA
      DO 4 JM=1,NOFF
           NJMAX=MAX(JM,1)
           J=MAX(JM,2)
           DO 62 N=1,NJMAX
                 NP1=N+2
                 N0=N+1
                 NM1=N
                 STX=-D1(1,NP1,J)+2.D0*YCOS*D1(1,N0,J)-D1(1,NM1,J)
     *              +2.D0*(T1(1,N,J)+T2(1,N,J)-T3(1,N,J)-T4(1,N,J))
                 STZ=-D1(2,NP1,J)+2.D0*YCOS*D1(2,N0,J)-D1(2,NM1,J)
     *              +2.D0*(T1(2,N,J)+T2(2,N,J)-T3(2,N,J)-T4(2,N,J))
                 MP2=N+4
                 MP1=N+3
                 M0=N+2
                 MM1=N+1
                 MM2=N
                 SINP2=DSIN(AKK*FLOAT(N+1)*DLX)
                 SINP1=DSIN(AKK*FLOAT(N)*DLX)
                 SIN0=DSIN(AKK*FLOAT(N-1)*DLX)
                 SINM1=DSIN(AKK*FLOAT(N-2)*DLX)
                 SINM2=DSIN(AKK*FLOAT(N-3)*DLX)
                 ATX=SINP2*S1(1,MP2,J)-4.D0*YCOS*SINP1*S1(1,MP1,J)
     *               +2.D0*(2.D0+CCS)*SIN0*S1(1,M0,J)-4.D0*YCOS
     *               *SINM1*S1(1,MM1,J)+SINM2*S1(1,MM2,J)
                 ATZ=SINP2*S1(2,MP2,J)-4.D0*YCOS*SINP1*S1(2,MP1,J)
     *               +2.D0*(2.D0+CCS)*SIN0*S1(2,M0,J)-4.D0*YCOS
     *               *SINM1*S1(2,MM1,J)+SINM2*S1(2,MM2,J)
                 AAX=-2.D0*(D2(1,NP1,J)-2.D0*YCOS*D2(1,N0,J)
     *               +D2(1,NM1,J))
                 AAZ=-2.D0*(D2(2,NP1,J)-2.D0*YCOS*D2(2,N0,J)
     *               +D2(2,NM1,J))
                 AX=ATX+AAX
                 AZ=ATZ+AAZ
                 ZW=W*(CSX*STX+CSZ*STZ+CAX*AX-CAZ*AZ)
                 GS(N)=ZW
   62      CONTINUE
    4 CONTINUE
```

```
      RETURN
      END
C..................................................................
C   This subroutine evaluates the higher order bessel functions using
C   the ascenting series expression or hankel's expansion.
C..................................................................
      SUBROUTINE BESS2 (BJ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BJ(10,2),U(4),RBJ(50,2)
      COMMON/B01/BJ0,BJ1
      COMMON/BSS/ARG(10),X
C
      PI=3.141592653589
C
C     Evaluation of J0,J1
C
      CALL BSJ0(X)
      RBJ(1,2)=BJ0
      RBJ(2,2)=BJ1
C
      NCON=1
      N=IDINT(2.4D0*X)
      IF (N.LT.10) N=10
      IF (X.LT.3.D0) GO TO 10
C
C     EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS UP TO
C     ORDER LESS THEN THE ARGUMENT
C
      NIMAX=IDINT(X)-1
      IF (NIMAX.GT.9) NIMAX=9
      DO 1 I=2,NIMAX
           NJ1=I
           NJ2=I-1
           NB=I+1
           RBJ(NB,2)=FLOAT(2*NJ2)*RBJ(NJ1,2)/X-RBJ(NJ2,2)
  1   CONTINUE
      IF (NIMAX.EQ.9) GO TO 20
      NCON=NIMAX
C
C     DEBYE'S ASYMPTOTIC EXPANSION-EVALUATION OF JN
C
  10  DO 11 J=1,2
            JN=N-J+1
            XA=X/FLOAT(JN)
            XA=1.D0/XA
            XE=XA+DSQRT(XA*XA-1.D0)
            A=DLOG(XE)
            CTH=(XE+1.D0/XE)/(XE-1.D0/XE)
            CALL F(CTH,U)
            TNH=1.D0/CTH
            R1=DEXP(FLOAT(JN)*(TNH-A))
            R2=DSQRT(2.D0*PI*FLOAT(JN)*TNH)
            BN1=JN
            BN2=JN*JN
            BN3=BN2*JN
            BN4=BN3*JN
            RBJ(JN+1,2)=(R1/R2)*(1.D0+U(1)/BN1+U(2)/BN2+U(3)/BN3+
     *                  U(4)/BN4)
  11  CONTINUE
C
C     EVALUATION OF HIGHER ORDER BESSEL FUNCTIONS WHEN X<10
C
      NJMAX=N-2-NCON
      DO 2 I=1,NJMAX
           NJB=N-I
           NJB1=NJB+1
```

```
            NJB2=NJB1+1
            RBJ(NJB,2)=2.D0*FLOAT(NJB)*RBJ(NJB1,2)/X-RBJ(NJB2,2)
  2   CONTINUE
  20  CONTINUE
      DO 3 I=1,9
            BJ(I,2)=RBJ(I,2)
  3   CONTINUE
      RETURN
      END
C.........................................................................
C.........................................................................
      SUBROUTINE BSJ0(X)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/B01/BJ0,BJ1
C
C
C     Evaluation of J0 using the series expansion given in
C     Abramowitz.
C
      PI=3.141592653589D0
      IF (X.GT.3.D0) GO TO 20
      X3=X/3.D0
      X32=X3*X3
      X34=X32*X32
      X36=X32*X34
      X38=X32*X36
      X310=X38*X32
      X312=X310*X32
      BJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0*X36+
     *    0.0444479D0*X38-0.0039444D0*X310+0.00021000D0*X312
      BJ1=X*(0.5D0-0.56249985D0*X32+0.21093573D0*X34-0.03954289D0
     *    *X36+0.00443319D0*X38-0.00031761D0*X310+0.00001109D0
     *    *X312)
      GO TO 21
C
   20 X3=3.D0/X
      X32=X3*X3
      X33=X32*X3
      X34=X33*X3
      X35=X34*X3
      X36=X35*X3
      FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.00009512D0
     *    *X33+0.00137237D0*X34-0.00072805D0*X35+0.00014476D0*X36
      FJ1=0.79788456D0+0.00000156D0*X3+0.01659667D0*X32+0.00017105D0
     *    *X33-0.00249511D0*X34+0.00113653D0*X35-0.00020033D0*X36
      TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00262573D0
     *    *X33-0.00054125D0*X34-0.00029333D0*X35+0.00013558D0*X36
      TJ1=X-2.35619449D0+0.12499612D0*X3+0.00005650D0*X32-0.00637879D0
     *    *X33+0.00074348D0*X34+0.00079824D0*X35-0.00029166D0*X36
      WCON=DSQRT(1.D0/X)
      BJ0=WCON*FJ0*DCOS(TJ0)
      BJ1=WCON*FJ1*DCOS(TJ1)
  21  CONTINUE
      RETURN
      END
C...............................................................................
C.....................................................................
      SUBROUTINE F(X,U)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(4)
      X2=X*X
      X3=X2*X
      X4=X3*X
      X5=X4*X
      X6=X5*X
      X7=X6*X
```

```
      X8=X7*X
      X9=X8*X
      X10=X9*X
      X11=X10*X
      X12=X11*X
C
      U(1)=(3.D0*X-5.D0*X3)/24.D0
      U(2)=(81.D0*X2-462.D0*X4+385.D0*X6)/1152.D0
      U(3)=(30375.D0*X3-369603.D0*X5+765765.D0*X7-425425.D0*X9)/
     *      414720.D0
      U(4)=(4465125.D0*X4-94121676.D0*X6+349922430.D0*X8-446185740.D0*
     *      X10+185910725.D0*X12)/39813120.D0
      RETURN
      END
C.....................................................................
C              SUBROUTINE DATA_SLOT
C     This subroutine gives all the data for integration used in
C     subroutine SLOT.FTN
C.....................................................................
      SUBROUTINE DATA_SLOT
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/DATT/COAL(20),POINT(20),CN(51),BM(151),POLTM(20),
     *POLTE(20),AM(41),DM(41),POLES(40),VXXM(20),VZXM(20),VZXE(20),
     *BPOINT(10),BCOAL(10),MPOINT,NPOINT,NK0,MA,NTM,NTE,NK0K,IFIRST
C
      COMMON/INT/XNS(40),CNS(40),XND(20,2),CND(20),XNT(40,3),
     *CNT(40),NDP,NTP,NSP
C
      COMMON/ADON/DIST(250,7,10),RCOE(20,250,7,10),AX,SERS(5),
     *SERA(5),DARG(10,4),S(10,2),WREAL,NSER,NMAX(7)
C
      COMMON/IOFF/INS,INS1S2
C
      PI=3.141592653589D0
C
C
      TPI=2.D0*PI
      TPI2=TPI*TPI
C     +------------------+
C     | ERROR FUNCTIONS  |
C     +------------------+
C
      A1=A*A/ER-TPI2
      A2=TPI2-TPI2/ER
      E1=0.5D0*A2/A1
      E2=ER*E1/(1.D0+ER)
      FA=DSQRT(1.D0+TPI2/A1)
C     +-----------------------------------------------------+
C     |              Data for the poles                     |
C     |   IFIRST= 0 : dominant mode is TM wave (many poles) |
C     |           1 : dominant mode is TE wave (many poles) |
C     |           2 : only one TM surface wave              |
C     +-----------------------------------------------------+
C   +----------------------------+
C   |  Data for the Integration  |
C   +----------------------------+
      NK0=20
      NK0K=1
      MA=40
      NSER=10
C
      NPOINT=10
```

```
C-----------
C     Vector COAL
C-----------
      COAL(1)=0.0666713443D0
      COAL(2)=0.14945134915D0
      COAL(3)=0.21908636251D0
      COAL(4)=0.26926671931D0
      COAL(5)=0.29552422471D0
      COAL(6)=COAL(5)
      COAL(7)=COAL(4)
      COAL(8)=COAL(3)
      COAL(9)=COAL(2)
      COAL(10)=COAL(1)
C     ------------
C     Vector POINT
C     ------------
      POINT(1)=0.973906528517D0
      POINT(2)=0.865063366688D0
      POINT(3)=0.679409568299D0
      POINT(4)=0.433395394129D0
      POINT(5)=0.148874338981D0
      POINT(6)=-POINT(5)
      POINT(7)=-POINT(4)
      POINT(8)=-POINT(3)
      POINT(9)=-POINT(2)
      POINT(10)=-POINT(1)
C
      MPOINT=5
C     ------------
C     Vector BCOAL
C     ------------
      BCOAL(1)=0.2369268851D0
      BCOAL(2)=0.4786286705D0
      BCOAL(3)=0.5688888888D0
      BCOAL(4)=BCOAL(2)
      BCOAL(5)=BCOAL(1)
C     -------------
C     Vector BPOINT
C     -------------
      BPOINT(1)=0.9061798459D0
      BPOINT(2)=0.5384693101D0
      BPOINT(3)=0.D0
      BPOINT(4)=-BPOINT(2)
      BPOINT(5)=-BPOINT(1)
C     ------------------
C     Single integration
C     ------------------
C
      NSP=31
      RS1=0.99708748181D0
      RS2=0.98468590966D0
      RS3=0.96250392509D0
      RS4=0.93075699789D0
      RS5=0.88976002994D0
      RS6=0.83992032014D0
      RS7=0.78173314841D0
      RS8=0.71577678458D0
      RS9=0.64270672292D0
      RS10=0.56324916140D0
      RS11=0.47819378204D0
      RS12=0.38838590160D0
      RS13=0.29471806998D0
      RS14=0.19812119933D0
      RS15=0.09955531215D0
      RS16=0.D0
C
```

```
      XNS(1)=RS1
      XNS(2)=RS2
      XNS(3)=RS3
      XNS(4)=RS4
      XNS(5)=RS5
      XNS(6)=RS6
      XNS(7)=RS7
      XNS(8)=RS8
      XNS(9)=RS9
      XNS(10)=RS10
      XNS(11)=RS11
      XNS(12)=RS12
      XNS(13)=RS13
      XNS(14)=RS14
      XNS(15)=RS15
      XNS(16)=RS16
      XNS(17)=-RS15
      XNS(18)=-RS14
      XNS(19)=-RS13
      XNS(20)=-RS12
      XNS(21)=-RS11
      XNS(22)=-RS10
      XNS(23)=-RS9
      XNS(24)=-RS8
      XNS(25)=-RS7
      XNS(26)=-RS6
      XNS(27)=-RS5
      XNS(28)=-RS4
      XNS(29)=-RS3
      XNS(30)=-RS2
      XNS(31)=-RS1
C
      CNS(1)=0.0074708315792D0
      CNS(2)=0.0173186207903D0
      CNS(3)=0.0270090191849D0
      CNS(4)=0.0364322739123D0
      CNS(5)=0.0454937075272D0
      CNS(6)=0.0541030824249D0
      CNS(7)=0.0621747865610D0
      CNS(8)=0.0696285832354D0
      CNS(9)=0.0763903865987D0
      CNS(10)=0.0823929917615D0
      CNS(11)=0.0875767406084D0
      CNS(12)=0.0918901138936D0
      CNS(13)=0.0952902429123D0
      CNS(14)=0.0977433353863D0
      CNS(15)=0.0992250112266D0
      CNS(16)=0.0997205447934D0
      CNS(17)=CNS(15)
      CNS(18)=CNS(14)
      CNS(19)=CNS(13)
      CNS(20)=CNS(12)
      CNS(21)=CNS(11)
      CNS(22)=CNS(10)
      CNS(23)=CNS(9)
      CNS(24)=CNS(8)
      CNS(25)=CNS(7)
      CNS(26)=CNS(6)
      CNS(27)=CNS(5)
      CNS(28)=CNS(4)
      CNS(29)=CNS(3)
      CNS(30)=CNS(2)
      CNS(31)=CNS(1)
C
C
C
```

```
C     2)  Double Integration
C         ------------------
C
      NDP=16
      R1=DSQRT((15.D0-2.D0*DSQRT(30.D0))/35.D0)
      R2=-R1
      S1=DSQRT((15.D0+2.D0*DSQRT(30.D0))/35.D0)
      S2=-S1
      A1=4.D0*(59.D0+6.D0*DSQRT(30.D0))/864.D0
      A2=4.D0*(59.D0-6.D0*DSQRT(30.D0))/864.D0
      A3=4.D0*49.D0/864.D0
C
      XND(1,1)=R1
      XND(1,2)=R1
      CND(1)=A1
C
      XND(2,1)=R2
      XND(2,2)=R1
      CND(2)=A1
C
      XND(3,1)=R1
      XND(3,2)=R2
      CND(3)=A1
C
      XND(4,1)=R2
      XND(4,2)=R2
      CND(4)=A1
C
      XND(5,1)=S1
      XND(5,2)=S1
      CND(5)=A2
C
      XND(6,1)=S1
      XND(6,2)=S2
      CND(6)=A2
C
      XND(7,1)=S2
      XND(7,2)=S1
      CND(7)=A2
C
      XND(8,1)=S2
      XND(8,2)=S2
      CND(8)=A2
C
      XND(9,1)=R1
      XND(9,2)=S1
      CND(9)=A3
C
      XND(10,1)=R1
      XND(10,2)=S2
      CND(10)=A3
C
      XND(11,1)=S1
      XND(11,2)=R1
      CND(11)=A3
C
      XND(12,1)=S2
      XND(12,2)=R1
      CND(12)=A3
C
      XND(13,1)=R2
      XND(13,2)=S1
      CND(13)=A3
C
      XND(14,1)=R2
      XND(14,2)=S2
```

```
      CND(14)=A3
C
      XND(15,1)=S1
      XND(15,2)=R2
      CND(15)=A3
C
      XND(16,1)=S2
      XND(16,2)=R2
      CND(16)=A3
C
C     3)  Triple Integration
C         ------------------
C
      NTP=34
      RS1=0.9317380000D0
      RS2=-RS1
      UU1=0.9167441779D0
      UU2=-UU1
      SS1=0.4086003800D0
      SS2=-SS1
      TT1=0.7398529500D0
      TT2=-TT1
      B1=8.D0*0.03558180896D0
      B2=8.D0*0.01247892770D0
      B3=8.D0*0.05286772991D0
      B4=8.D0*0.02672752182D0
C
      XNT(1,1)=RS1
      XNT(1,2)=0.D0
      XNT(1,3)=0.D0
      CNT(1)=B1
C
      XNT(2,1)=RS2
      XNT(2,2)=0.D0
      XNT(2,3)=0.D0
      CNT(2)=B1
C
      XNT(3,1)=0.D0
      XNT(3,2)=RS1
      XNT(3,3)=0.D0
      CNT(3)=B1
C
      XNT(4,1)=0.D0
      XNT(4,2)=RS2
      XNT(4,3)=0.D0
      CNT(4)=B1
C
      XNT(5,1)=0.D0
      XNT(5,2)=0.D0
      XNT(5,3)=RS1
      CNT(5)=B1
C
      XNT(6,1)=0.D0
      XNT(6,2)=0.D0
      XNT(6,3)=RS2
      CNT(6)=B1
C
      XNT(7,1)=UU1
      XNT(7,2)=UU1
      XNT(7,3)=0.D0
      CNT(7)=B2
C
      XNT(8,1)=UU2
      XNT(8,2)=UU1
      XNT(8,3)=0.D0
      CNT(8)=B2
```

```
C
      XNT(9,1)=UU1
      XNT(9,2)=UU2
      XNT(9,3)=0.D0
      CNT(9)=B2
C
      XNT(10,1)=UU2
      XNT(10,2)=UU2
      XNT(10,3)=0.D0
      CNT(10)=B2
C
      XNT(11,1)=UU1
      XNT(11,2)=0.D0
      XNT(11,3)=UU1
      CNT(11)=B2
C
      XNT(12,1)=UU1
      XNT(12,2)=0.D0
      XNT(12,3)=UU2
      CNT(12)=B2
C
      XNT(13,1)=UU2
      XNT(13,2)=0.D0
      XNT(13,3)=UU1
      CNT(13)=B2
C
      XNT(14,1)=UU2
      XNT(14,2)=0.D0
      XNT(14,3)=UU2
      CNT(14)=B2
C
      XNT(15,1)=0.D0
      XNT(15,2)=UU1
      XNT(15,3)=UU1
      CNT(15)=B2
C
      XNT(16,1)=0.D0
      XNT(16,2)=UU1
      XNT(16,3)=UU2
      CNT(16)=B2
C
      XNT(17,1)=0.D0
      XNT(17,2)=UU2
      XNT(17,3)=UU1
      CNT(17)=B2
C
      XNT(18,1)=0.D0
      XNT(18,2)=UU2
      XNT(18,3)=UU2
      CNT(18)=B2
C
      XNT(19,1)=SS1
      XNT(19,2)=SS1
      XNT(19,3)=SS1
      CNT(19)=B3
C
      XNT(20,1)=SS1
      XNT(20,2)=SS1
      XNT(20,3)=SS2
      CNT(20)=B3
C
      XNT(21,1)=SS1
      XNT(21,2)=SS2
      XNT(21,3)=SS1
      CNT(21)=B3
C
```

```
      XNT(22,1)=SS1
      XNT(22,2)=SS2
      XNT(22,3)=SS2
      CNT(22)=B3
C
      XNT(23,1)=SS2
      XNT(23,2)=SS1
      XNT(23,3)=SS1
      CNT(23)=B3
C
      XNT(24,1)=SS2
      XNT(24,2)=SS1
      XNT(24,3)=SS2
      CNT(24)=B3
C
      XNT(25,1)=SS2
      XNT(25,2)=SS2
      XNT(25,3)=SS1
      CNT(25)=B3
C
      XNT(26,1)=SS2
      XNT(26,2)=SS2
      XNT(26,3)=SS2
      CNT(26)=B3
C
      XNT(27,1)=TT1
      XNT(27,2)=TT1
      XNT(27,3)=TT1
      CNT(27)=B4
C
      XNT(28,1)=TT1
      XNT(28,2)=TT1
      XNT(28,3)=TT2
      CNT(28)=B4
C
      XNT(29,1)=TT1
      XNT(29,2)=TT2
      XNT(29,3)=TT1
      CNT(29)=B4
C
      XNT(30,1)=TT1
      XNT(30,2)=TT2
      XNT(30,3)=TT2
      CNT(30)=B4
C
      XNT(31,1)=TT2
      XNT(31,2)=TT1
      XNT(31,3)=TT1
      CNT(31)=B4
C
      XNT(32,1)=TT2
      XNT(32,2)=TT1
      XNT(32,3)=TT2
      CNT(32)=B4
C
      XNT(33,1)=TT2
      XNT(33,2)=TT2
      XNT(33,3)=TT1
      CNT(33)=B4
C
      XNT(34,1)=TT2
      XNT(34,2)=TT2
      XNT(34,3)=TT2
      CNT(34)=B4
C
      RETURN
```

```
      END
```

################ apollo domain ################

CAEN/Apollo

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

```
                                                                     000
y   y    i          j       w     w    aa     v     v  eeeeee       k     k  0   0        ffffff  ttttt  n    n
 y y     i          j       w     w   a  a    v     v  e            k    k   0     0      f         t    nn   n
  y      i          j       w     w  a    a   v     v  eeeee        kkkk     0     0      fffff     t    n n  n
  y      i          j       w ww  w  aaaaaa   v     v  e            k  k     0     0  ... f         t    n  n n
  y      i      j   j       ww  ww  a    a     v   v   e            k   k     0   0   ... f         t    n   nn
  y      i       jjjj ______ w    w  a    a      vv    eeeeee ______ k    k    000    ... f         t    n    n
```

//tera/users/katehi/tape/yij_wave_k0.ftn

LAST MODIFIED ON: 89/04/24 10:44 AM
FILE PRINTED: 89/04/24 3:20 PM

```
C..............................................................................
C          The name of this file is YIJ_WAVE_K0.FTN
C
C     In this program the current has akk=ak0
C
C     Also SUM3=0.0
C..............................................................................
C     This subroutine evaluates the contribution to the admittance matrix
C     which comes from the waveguide
C..............................................................................
      SUBROUTINE YIJ_WAVE
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX YSD,YSW,COEF,CI
      DIMENSION ARG(250),R10(250),X10(250),SGMN(250),AC(3),AS(3),
     *RIJ(250),R00(2)
C
      COMMON/SERIES/SUM1
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/BESSEL/BJ0(6000)
C
      ARG1=PI*Y0/AW
      ARG2=PI*W/(2.D0*AW)
      CALL VBJ0(ARG1,ARG2)
      CALL S14
C
      CI=(0.0,1.0)
      AK02=AK0*AK0
C
C     Evaluation of vector ARG
C
      JMAX=NS1
      I=1
      DO 1 J=1,JMAX
           ARG(J)=(J-I)*DLX
  1   CONTINUE
C
C     Evaluation of vectors R00,R10,X10
C
      C3=1.D0
      B01=DSQRT(C3-1.D0/(2.D0*AW)**2)
      B012=B01*B01
      SCOEF=0.5D0*(C3-B012)*BJ0(1)/((2.D0*PI*(1.D0-B012))**2*B01)
      ARGK=AK0*DLX
      ARGB=ARGK*B01
      COSK=DCOS(ARGK)
      COSB=DCOS(ARGB)
      COS2K=DCOS(2.D0*ARGK)
      COS2B=DCOS(2.D0*ARGB)
      COS2=COSK*COSK
      SINK=DSIN(ARGK)
      SINB=DSIN(ARGB)
      SIN2K=2.D0*SINK*COSK
      SIN2B=2.D0*SINB*COSB
      SIN3B=DSIN(3.D0*ARGB)
      SIN2=SINK*SINK
C
      R00(1)=(-DLX+SIN2K/(2.D0*AK0))/(4.D0*PI)
      R00(2)=(DLX*COSK-SINK/AK0)/(8.D0*PI)
C
      R10(1)=(8.D0*COSK*SINB-2.D0*SIN2B-2.D0*B01*SIN2K)*SCOEF
      C1=(COSK-COSB)**2
```

```
      X10(1)=-4.D0*C1*SCOEF
      R10(2)=2.D0*SINB*(-2.D0*C1-1.D0+B01*SINK/SINB)*SCOEF
      X10(2)=-4.D0*COSB*C1*SCOEF
      DO 2 J=3,JMAX
           R10(J)=-4.D0*DSIN(AK0*B01*ARG(J))*C1*SCOEF
           X10(J)=-4.D0*DCOS(AK0*B01*ARG(J))*C1*SCOEF
  2   CONTINUE
C-----------------------------------------------------------------
C      DO 200 IG=1,JMAX
C             WRITE (6,201) IG,R10(IG),X10(IG)
C  201        FORMAT(10X,'IG=',I4,2X,'R10=',E14.7,2X,'X10=',E14.7)
C  200 CONTINUE
C-----------------------------------------------------------------
C     Evaluation of vectors AC(A) and AS(a)
C
      AS(1)=2.D0
      AS(2)=-1.D0
      AS(3)=0.D0
C
      AC(1)=2.D0*(1.D0+2.D0*COS2)
      AC(2)=-4.D0*COSK
      AC(3)=1.D0
C
C     Evaluation of vector SGMN
C
      KMAX=JMAX+2
      DO 5 K=1,KMAX
           SGMN(K)=0.D0
           NTEST=0
           INDEXN=-1
 3         INDEXN=INDEXN+1
                 EN=0.5D0
                 IF (INDEXN.GT.0) EN=1.D0
                 C1=(INDEXN/(2.D0*AW))**2
                 SUMM=0.D0
                 INDEXM=0
                 IF (INDEXN.GE.2) INDEXM=-1
                 ITEST=0
                 ID_M=1
 4               INDEXM=INDEXM+1
                       EM=0.5D0
                       IF (INDEXM.GT.0) EM=1.D0
                       C2=(INDEXM/(2.D0*BW))**2
                       GMN2=C1+C2-C3
                       GMN=DSQRT(GMN2)
                       ITEST=ITEST+1
                       D1=AK0*(K-1)*GMN*DLX
                       D2=0.D0
                       IF (D1.LT.40.D0) D2=DEXP(-D1)
                       TERM=EM*(C3+GMN2)*D2/(GMN*(1.D0+GMN2)**2)
                       SUMM=SUMM+TERM
                       RATIO=0.D0
                       IF (SUMM.GT.1.D-40) RATIO=DABS(TERM/SUMM)
                       ERRORM=ERROR
                       IF (K.LE.3) ERRORM=1.D-9
                       IF (RATIO.GT.ERRORM) ITEST=0
                       IF (ITEST.LT.5) GO TO 4
                 NTEST=NTEST+1
                 CBJ0=1.D0
                 IF (INDEXN.GT.0) CBJ0=BJ0(INDEXN)
                 TERM=EN*CBJ0*SUMM
                 SGMN(K)=SGMN(K)+TERM
                 RATIO=DABS(TERM/SGMN(K))
                 IF (RATIO.GT.ERROR) NTEST=0
                 IF (NTEST.LT.4) GO TO 3
           SGMN(K)=SGMN(K)/(2.D0*PI)**2
```

```
C------------------------------------------------------------
C            WRITE (6,11) K,INDEXN,SGMN(K)
C  11        FORMAT(10X,'K=',I4,2X,'INDEXN=',I4,2X,'SGMN=',E14.7)
C------------------------------------------------------------
  5   CONTINUE
C
C     Evaluation of vector RIJ
C
      RIJ(1)=AC(1)*SGMN(1)+2.D0*AC(2)*SGMN(2)+2.D0*SGMN(3)
      RIJ(2)=AC(2)*SGMN(1)+(1.D0+AC(1))*SGMN(2)+AC(2)*SGMN(3)+SGMN(4)
      DO 6 J=3,JMAX
           RIJ(J)=SGMN(J-2)+AC(2)*SGMN(J-1)+AC(1)*SGMN(J)
     *              +AC(2)*SGMN(J+1)+SGMN(J+2)
  6   CONTINUE
C------------------------------------------------------------
C      DO 19 JK=1,JMAX
C            WRITE (6,18) JK,RIJ(JK)
C  18        FORMAT(2X,'JK=',I4,2X,'SIJ=',E14.7)
C  19  CONTINUE
C------------------------------------------------------------
C     Evaluation of this part of the elements of the admittance
C     which comes from the waveguide
C
C      WRITE (6,50)
  50  FORMAT(///10X,'Waveguide Admittance Matrix'///)
      COEF=CI*SNGL(-2.D0/(120.D0*PI*AW*BW*SIN2))
      DO 7 J=1,2
           SINA=DSIN(AS(J)*AK0*DLX)
           YR1=-SINA*SUM1
           R_UWIJ=R00(J)+R10(J)+YR1+RIJ(J)
           X_UWIJ=X10(J)
           YSW(J)=COEF*(SNGL(R_UWIJ)+CI*SNGL(X_UWIJ))
C--------------------------------------------------------
           WRITE (6,20) J,YSW(J)
  20       FORMAT(2X,'J=',I4,2X,'YS=',E14.7,2X,E14.7)
C--------------------------------------------------------
  7   CONTINUE
      DO 8 J=3,JMAX
           R_UWIJ=R10(J)+RIJ(J)
           X_UWIJ=X10(J)
           YSW(J)=COEF*(SNGL(R_UWIJ)+CI*SNGL(X_UWIJ))
C--------------------------------------------------------
           WRITE (6,21) J,YSW(J)
  21       FORMAT(2X,'J=',I4,2X,'YS=',E14.7,2X,E14.7)
C--------------------------------------------------------
   8  CONTINUE
      RETURN
      END
C...................................................................................
C     This subroutine evaluates the single and double series S1,S2,S3,S4
C     which are common to all Yij elements
C...................................................................................
      SUBROUTINE S14
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BESSEL/BJ0(6000)
C
      COMMON/SERIES/SUM1
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
C     Evaluation of the single integral
C
      INDEX=1
      ITEST=0
      ARG=PI*BW/AW
```

```
      COTH=1.D0/DTANH(ARG)
      SUM1=(BW**2/6.D0)+BJ0(1)*(ARG*COTH-1.D0)*(AW/PI)**2
  1   INDEX=INDEX+1
          ARGN=INDEX*ARG
          COTH=1.D0/DTANH(ARGN)
          TERM=BJ0(INDEX)*(AW*BW/PI)*(COTH/INDEX)
          SUM1=SUM1+TERM
          RATIO=DABS(TERM/SUM1)
          IF (RATIO.LT.ERROR) GO TO 2
                    ITEST=0
                    GO TO 1
  2       ITEST=ITEST+1
          IF (ITEST.LT.6) GO TO 1
C-----------------------------------------------------
C          WRITE (6,10) INDEX,SUM1
C  10      FORMAT(2X,'INDEX=',I4,5X,'SUM1=',E14.7)
C-----------------------------------------------------
      RETURN
      END
C**********************************************************************
C     This function evaluates the zeroth order first kind Bessel
C                                Function J0
C**********************************************************************
      SUBROUTINE VBJ0(ARG1,ARG2)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BESSEL/BJ0(6000)
      PI=3.141592653589D0
      DO 1 M=1,6000
           X=FLOAT(M)*ARG2
           X1=FLOAT(M)*ARG1
           COS1=DCOS(X1)
           COS2=COS1*COS1
           IF (X.GT.0.001D0) GO TO 10
               X3=X/3.D0
               X32=X3*X3
               X34=X32*X32
               X36=X34*X32
               BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *              *X36
               BJ0(M)=BSJ0*COS2
               GO TO 1
  10       IF (X.GT.3.D0) GO TO 12
               X3=X/3.D0
               X32=X3*X3
               X34=X32*X32
               X36=X34*X32
               X38=X36*X32
               X310=X38*X32
               X312=X310*X32
               BSJ0=1.D0-2.2499997D0*X32+1.2656208D0*X34-0.3163866D0
     *              *X36+0.0444479D0*X38-0.0039444D0*X310+0.00021000
     *              D0*X312
               BJ0(M)=BSJ0*COS2
               GO TO 1
   12      CONTINUE
           X3=3.D0/X
           X32=X3*X3
           X33=X32*X3
           X34=X33*X3
           X35=X34*X3
           X36=X35*X3
           FJ0=0.79788456D0-0.00000077D0*X3-0.00552740D0*X32-0.0000
     *         9512D0*X33+0.00137237D0*X34-0.00072805D0*X35+0.00014
     *         476D0*X36
           TJ0=X-0.78539816D0-0.04166397D0*X3-0.00003954D0*X32+0.00
     *         262573D0*X33-0.00054125D0*X34-0.00029333D0*X35+0.000
```

```
     *         13558D0*X36
           WCON=DSQRT(1.D0/X)
           BSJ0=WCON*FJ0*DCOS(TJ0)
           BJ0(M)=BSJ0*COS2
    1   CONTINUE
        RETURN
        END
```

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

//tera/users/katehi/tape/inv_wave_k0.ftn

LAST MODIFIED ON: 89/04/24 10:45 AM
FILE PRINTED: 89/04/24 10:54 AM

```
C*************************************************************
C     The name of this file is ......... INV_WAVE_K0.FTN................
C     It finds the inverse matrix for the case of  a waveguide slot
C     covered by a dielectric substrate.
C*************************************************************
      SUBROUTINE INV_WAVE(YS)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL SLOT_V
      COMPLEX CUR,BMATR,SUMC,CI,CINC,CIN,BACK_SCAT,FORW_SCAT,B_S,
     *F_S
      COMPLEX YSD,YSW,YS(250)
C
      COMMON/AD_MAT/YSD(250),YSW(250),NS,NS1S2
C
      COMMON/DAT/ER,H,T,DLX,AW,BW,Y0,A,TPI,TPI2,PI,W,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),ALONG(7),WDELTA,OFFLIM,ERROR,NS1,NS2,NSS2,NOFF
C
      COMMON/MAN/BMATR(260,260),IA(260),IB(260)
C
      COMMON/SLOT/SNMIJ(150,2)
C
      COMMON/INV/CUR(260),NOR
C
      COMMON/SCAT_COEF/BACK_SCAT,FORW_SCAT
C
      COMMON/B01/BJ0,BJ1
C
      COMMON/SLOT_VOLTAGE/SLOT_V
C.............................................................
C                   DATA
C.............................................................
      CI=(0.0,1.0)
      NOEL1=NS1
      NS12=NS1S2
      NOR=NS1
C
C     .....First Diagonal Matrix.........
C
      IMIN=1
      IMAX=NOEL1
      DO 4 I=IMIN,IMAX
           KI=0
           DO 5 KJ=I,IMAX
                KI=KI+1
                BMATR(KI,KJ)=YS(I)
                BMATR(KJ,KI)=BMATR(KI,KJ)
  5        CONTINUE
  4   CONTINUE
      CALL MINVCD (NOR,NOR,DETA)
C
C     Evaluation of the magnetic current or the electric field
C     distribution
C
      B01=DSQRT(1.D0-(0.5D0/AW)**2)
      B012=B01*B01
      ARGY0=PI*Y0/AW
      ARGP=PI*DLX*(B01+1.D0)
      ARGM=PI*DLX*(B01-1.D0)
      ARG0=AK0*DLX
C
      CINC=-SNGL(DCOS(ARGY0)*DSIN(ARGP)*DSIN(ARGM)/(DSIN(ARG0)*
     *     (1.D0-B012)))
C
      ARG=B01*2.D0*PI*DLX
      ARGL=FLOAT(NOR+1)*DLX*B01*PI
      DO 70 IQ=1,NOR
```

```
            SUMC=(0.0,0.0)
  401       DO 170 JQ=1,NOR
                    ARGX=-ARG*FLOAT(JQ)+ARGL
                    EC=DCOS(ARGX)
                    ES=DSIN(ARGX)
                    CIN=(SNGL(EC)+CI*SNGL(ES))
                    SUMC=SUMC+BMATR(IQ,JQ)*CIN
  170       CONTINUE
            CUR(IQ)=SUMC*CINC
   70 CONTINUE
C
C     Evaluation of the scattering coefficients
C
      CALL BSJ0(PI*W/(2.D0*AW))
      DINC=(1.D0/(PI*AW))**2*(1.D0/(AW*BW))*(1.D0/DSIN(ARG0))
      DINC=DINC*DCOS(ARGY0)*BJ0*DSIN(ARGM)*DSIN(ARGP)/(B01*
     *      (1.D0-B012))/(120.D0*PI)
      CINC=-SNGL(DINC)
      B_S=(0.0,0.0)
      F_S=(0.0,0.0)
      DO 71 JQ=1,NOR
            ARGX=-ARG*FLOAT(JQ)+ARGL
            EC=DCOS(ARGX)
            ES=DSIN(ARGX)
            B_S=B_S+CUR(JQ)*(SNGL(EC)+CI*SNGL(ES))
            F_S=F_S+CUR(JQ)*(SNGL(EC)-CI*SNGL(ES))
   71 CONTINUE
      BACK_SCAT=CINC*B_S
      FORW_SCAT=CINC*F_S
      N_CENTER=(NOR+1)/2
      SLOT_V=CABS(CUR(N_CENTER))
      RETURN
      END
C**********************************************************************
C     THIS SUBROUTINE INVERTS A SQUARE COMPLEX MATRIX
C**********************************************************************
      SUBROUTINE MINVCD (IA,MA,DETA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX A,PIV,DETA,TEMP,PIV1
      COMMON/MAN/A(260,260),IR(260),IC(260)
      DO 1 I=1,MA
      IR(I)=0
    1 IC(I)=0
C     DETA=(1.00,0.00)
      S=0.00
      R=MA
    2 CALL SUBMCD(IA,IA,MA,MA,I,J)
      PIV=A(I,J)
C     DETA=PIV*DETA
      Y=CABS(PIV)
      IF (Y.EQ.0) GO TO 17
      IR(I)=J
      IC(J)=I
      PIV=(1.00,0.00)/PIV
      A(I,J)=PIV
      DO 5 K=1,MA
    5 IF (K.NE.J) A(I,K)=A(I,K)*PIV
      DO 9 K=1,MA
      IF (K.EQ.I) GO TO 9
      PIV1=A(K,J)
    6 DO 8 L=1,MA
    8 IF (L.NE.J) A(K,L)=A(K,L)-PIV1*A(I,L)
    9 CONTINUE
      DO 11 K=1,MA
   11 IF (K.NE.I) A(K,J)=-PIV*A(K,J)
      S=S+1.00
```

```
      IF (S.LT.R) GO TO 2
   12 DO 16 I=1,MA
      K=IC(I)
      M=IR(I)
      IF (K.EQ.I) GO TO 16
C     DETA=-DETA
      DO 14 L=1,MA
      TEMP=A(K,L)
      A(K,L)=A(I,L)
   14 A(I,L)=TEMP
      DO 15 L=1,MA
      TEMP=A(L,M)
      A(L,M)=A(L,I)
   15 A(L,I)=TEMP
      IC(M)=K
      IR(K)=M
   16 CONTINUE
      RETURN
   17 WRITE (6,18)I,J
   18 FORMAT (10X,'MATRIX IS SINGULAR'/10X,'I=',I4,5X,'J=',I4)
      RETURN
      END
C*********************************************************************
C....................................................................
C*********************************************************************
      SUBROUTINE SUBMCD(IA,JA,MA,NA,I,J)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX  A
      COMMON/MAN/A(260,260),IR(260),IC(260)
      I=0
      J=0
      TEST=0.00
      DO 5 K=1,MA
      IF (IR(K).NE.0) GO TO 5
      DO 4 L=1,NA
      IF (IC(L).NE.0) GO TO 4
      X=CABS(A(K,L))
      IF(X.LT.TEST) GO TO 4
      I=K
      J=L
      TEST=X
    4 CONTINUE
    5 CONTINUE
      RETURN
      END
```

PROGRAM III

This program evaluates the mutual coupling between two dielectric covered longitudinal slots as a function of their separation distance.

The files which consist this program are:

RUN_MUTUAL:	This program links all the subroutines.
DATA_WAVE_MUTUAL:	Input File
OUT_WAVE_MUTUAL:	Output File
SLOT_DESIGN.FTN :	Main Program Subroutine DATA Subroutine F_EER Subroutine NORM Subroutine CUBSPL
MUTUAL_SLOT.FTN	Subroutine MUTUAL_SLOT Subroutine DATA_MUTUAL_SLOT
POLES_MUTUAL.FTN :	Subroutine SPOLES
YIJ_DIEL_MUTUAL.FTN:	Subroutine YIJ_DIEL Subroutine LIMIT Subroutine GREEN Function GXXM Function GZXM Function HZXE Subroutine FUNCT Subroutine GREI Subroutine ARIS Subroutine ADONIS

	Subroutine BESS1
	Subroutine TAIL
	Subroutine BESS2
	Subroutine BSJ0
	Subroutine F
	Subroutine DATA_SLOT
YIJ_WAVE_MUTUAL.FTN:	Subroutine YIJ_WAVE
	Subroutine S14
	Subroutine VBJ0
ARRANGE_MUTUAL.FTN:	Subroutine ARRANGE_MUTUAL

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

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

//tera/users/katehi/tape/run_mutual

LAST MODIFIED ON: 89/04/24 10:40 AM
FILE PRINTED: 89/04/24 11:01 AM

```
BIND SLOT_DESIGN.BIN MUTUAL_SLOT.BIN POLES_MUTUAL.BIN YIJ_DIEL_MUTUAL.BIN YIJ_WAVE_MUTUAL.BIN A
```

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

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

//tera/users/katehi/tape/data_wave_mutual

LAST MODIFIED ON: 89/04/24 10:39 AM
FILE PRINTED: 89/04/24 10:50 AM

```
C
C     ---- Dielectric constant ---
C
      2.62
C
C     ---- Substrate Thickness ---
C
      0.050
C
C     ---- Conductor Thickness ---
C
      0.00001
C
C     ---- Dimensions of the Waveguide ----
C
      0.6858
      0.3048
C
C     ---- Number of Slots ----
C
      2
C
C     ----  Transverse offsets of the slots ----
C
      0.24765
      0.43815
C
C     ---- Longitudinal offets of the slot ----
C
      1
      100
C
C     ---- Slot widths ----
C
      0.047625
      0.047625
C
C     ---- Slot Excess Widths ----
C
      0.0
      0.0
C
C     ---- Subsection Length ----
C
      0.01173
      0.01173
C
C     ---- Lower Limit of the Tail Contribution ----
C
      100.0
C
C     ---- Number of Points on the Slots ----
C
      29
      29
C
C     ---- Error in the evaluation of the series ----
C
      1.D-6
```

KATEHI

out_wave_mutual

//tera/users/katehi/tape/out_wave_mutual

LAST MODIFIED ON: 89/04/24 10:40 AM
FILE PRINTED: 89/04/24 10:58 AM

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


Substrate Thickness
 0.5000000E-01


Conductor Thickness
 0.1000000E-04


Dimensions of the Waveguide
AW= 0.6858000E+00
BW= 0.3048000E+00


Number of Slots
NSLOTS=   2


Transverse Offsets of the Slots
YOFF(1)= 0.2476500E+00
YOFF( 2)= 0.4381500E+00


Longitudinal Offset of the Slots
NXOFF(1)=   1
NXOFF( 2)= 100


Slot Widths
WS(1)= 0.4762500E-01
WS( 2)= 0.4762500E-01


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


Subsection Length
 0.1173000E-01


DLX_RES( 2)= 0.1173000E-01


Lower Limit of Tail Contribution
 0.1000000E+03


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


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

```
          Normalization Constant
           0.1000000E+01

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

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

I=    1  J=    2  NSSL= 130 0.1905000E+00


Offsets for the dielectric layer and number of corresponding elements

I=    1  OFFSET= 0.1905000E+00  NOFFS= 130
I=    2  OFFSET= 0.0000000E+00  NOFFS=    0


SLOTS and corresponding offsets in the dielectric

I=    1  J=    2  INSS=    1  OFFSET= 0.1905000E+00


Max number of offsets in the dielectric
NOFF=    1


No TE waves excited in the substrate


There are    1 TM waves excited in the substrate

   1    0.640756827E+01

Contribution to admittance from the dielectric


OFFSET #    1


Waveguide Admittance Matrix

Interactions between slots 1 and  2
```

IJ=	1	YSD=-0.1502052E-05	0.2275877E-05	YSW= 0.2415901E-06	0.7609726E-06
IJ=	2	YSD=-0.1501056E-05	0.2242999E-05	YSW= 0.2412827E-06	0.1326728E-05
IJ=	3	YSD=-0.1498070E-05	0.2147365E-05	YSW= 0.2403616E-06	0.1069803E-05
IJ=	4	YSD=-0.1493103E-05	0.1997759E-05	YSW= 0.2388290E-06	0.9477048E-06
IJ=	5	YSD=-0.1486167E-05	0.1808023E-05	YSW= 0.2366887E-06	0.7447657E-06
IJ=	6	YSD=-0.1477282E-05	0.1595797E-05	YSW= 0.2339464E-06	0.6527574E-06
IJ=	7	YSD=-0.1466470E-05	0.1380082E-05	YSW= 0.2306088E-06	0.3855379E-06
IJ=	8	YSD=-0.1453760E-05	0.1177184E-05	YSW= 0.2266846E-06	0.2333452E-06
IJ=	9	YSD=-0.1439187E-05	0.9954610E-06	YSW= 0.2221837E-06	0.6199575E-07
IJ=	10	YSD=-0.1422788E-05	0.8318239E-06	YSW= 0.2171175E-06	-0.9202368E-07
IJ=	11	YSD=-0.1404607E-05	0.6741448E-06	YSW= 0.2114990E-06	-0.2308191E-06
IJ=	12	YSD=-0.1384692E-05	0.5118209E-06	YSW= 0.2053424E-06	-0.3394165E-06
IJ=	13	YSD=-0.1363095E-05	0.3496734E-06	YSW= 0.1986634E-06	-0.4326996E-06
IJ=	14	YSD=-0.1339874E-05	0.2135603E-06	YSW= 0.1914790E-06	-0.5042081E-06
IJ=	15	YSD=-0.1315088E-05	0.1374578E-06	YSW= 0.1838074E-06	-0.5603256E-06
IJ=	16	YSD=-0.1288803E-05	0.1352719E-06	YSW= 0.1756683E-06	-0.5984197E-06
IJ=	17	YSD=-0.1261085E-05	0.1786389E-06	YSW= 0.1670822E-06	-0.6262339E-06
IJ=	18	YSD=-0.1232009E-05	0.2057486E-06	YSW= 0.1580711E-06	-0.6422787E-06
IJ=	19	YSD=-0.1201648E-05	0.1648202E-06	YSW= 0.1486578E-06	-0.6503756E-06
IJ=	20	YSD=-0.1170079E-05	0.6159189E-07	YSW= 0.1388663E-06	-0.6508224E-06
IJ=	21	YSD=-0.1137385E-05	-0.3463379E-07	YSW= 0.1287216E-06	-0.6469030E-06
IJ=	22	YSD=-0.1103647E-05	-0.4370827E-07	YSW= 0.1182494E-06	-0.6383362E-06
IJ=	23	YSD=-0.1068951E-05	0.5150355E-07	YSW= 0.1074763E-06	-0.6268533E-06
IJ=	24	YSD=-0.1033385E-05	0.1783485E-06	YSW= 0.9642982E-07	-0.6131191E-06
IJ=	25	YSD=-0.9970355E-06	0.2299578E-06	YSW= 0.8513803E-07	-0.5978355E-06
IJ=	26	YSD=-0.9599943E-06	0.1619685E-06	YSW= 0.7362963E-07	-0.5814549E-06
IJ=	27	YSD=-0.9223518E-06	0.4019307E-07	YSW= 0.6193392E-07	-0.5644310E-06
IJ=	28	YSD=-0.8841999E-06	-0.1446665E-07	YSW= 0.5008064E-07	-0.5470432E-06
IJ=	29	YSD=-0.8456310E-06	0.5967399E-07	YSW= 0.3809995E-07	-0.5295273E-06
IJ=	30	YSD=-0.8067377E-06	0.2009203E-06	YSW= 0.2602233E-07	-0.5120179E-06
IJ=	31	YSD=-0.7676119E-06	0.2791403E-06	YSW= 0.1387852E-07	-0.4947112E-06
IJ=	32	YSD=-0.7283458E-06	0.2225402E-06	YSW= 0.1699396E-08	-0.4775658E-06
IJ=	33	YSD=-0.6890300E-06	0.9135283E-07	YSW=-0.1048405E-07	-0.4607260E-06
IJ=	34	YSD=-0.6497557E-06	0.1995795E-07	YSW=-0.2264083E-07	-0.4441717E-06
IJ=	35	YSD=-0.6106114E-06	0.8139568E-07	YSW=-0.3474000E-07	-0.4279175E-06
IJ=	36	YSD=-0.5716889E-06	0.2118529E-06	YSW=-0.4675080E-07	-0.4119574E-06
IJ=	37	YSD=-0.5330600E-06	0.2755494E-06	YSW=-0.5864266E-07	-0.3962759E-06
IJ=	38	YSD=-0.4948237E-06	0.2053935E-06	YSW=-0.7038532E-07	-0.3808515E-06
IJ=	39	YSD=-0.4570481E-06	0.7291942E-07	YSW=-0.8194893E-07	-0.3656590E-06
IJ=	40	YSD=-0.4198299E-06	0.1269200E-07	YSW=-0.9330405E-07	-0.3506709E-06
IJ=	41	YSD=-0.3832313E-06	0.8001712E-07	YSW=-0.1044218E-06	-0.3358593E-06
IJ=	42	YSD=-0.3473286E-06	0.1932922E-06	YSW=-0.1152739E-06	-0.3211962E-06
IJ=	43	YSD=-0.3121926E-06	0.2227584E-06	YSW=-0.1258327E-06	-0.3066554E-06
IJ=	44	YSD=-0.2778891E-06	0.1301198E-06	YSW=-0.1360714E-06	-0.2922119E-06
IJ=	45	YSD=-0.2444822E-06	0.8233656E-08	YSW=-0.1459640E-06	-0.2778434E-06
IJ=	46	YSD=-0.2120311E-06	-0.2138847E-07	YSW=-0.1554852E-06	-0.2635299E-06
IJ=	47	YSD=-0.1805917E-06	0.5935715E-07	YSW=-0.1646108E-06	-0.2492541E-06
IJ=	48	YSD=-0.1502157E-06	0.1478093E-06	YSW=-0.1733177E-06	-0.2350018E-06
IJ=	49	YSD=-0.1209510E-06	0.1354010E-06	YSW=-0.1815836E-06	-0.2207613E-06
IJ=	50	YSD=-0.9284119E-07	0.2678439E-07	YSW=-0.1893876E-06	-0.2065240E-06
IJ=	51	YSD=-0.6592566E-07	-0.6797063E-07	YSW=-0.1967097E-06	-0.1922841E-06
IJ=	52	YSD=-0.4023950E-07	-0.5725451E-07	YSW=-0.2035315E-06	-0.1780386E-06
IJ=	53	YSD=-0.1581337E-07	0.3140372E-07	YSW=-0.2098354E-06	-0.1637871E-06
IJ=	54	YSD= 0.7326277E-08	0.8512472E-07	YSW=-0.2156055E-06	-0.1495317E-06
IJ=	55	YSD= 0.2915747E-07	0.3358150E-07	YSW=-0.2208270E-06	-0.1352770E-06
IJ=	56	YSD= 0.4966289E-07	-0.7370772E-07	YSW=-0.2254868E-06	-0.1210298E-06
IJ=	57	YSD= 0.6882937E-07	-0.1260233E-06	YSW=-0.2295730E-06	-0.1067990E-06
IJ=	58	YSD= 0.8664801E-07	-0.7714630E-07	YSW=-0.2330751E-06	-0.9259556E-07
IJ=	59	YSD= 0.1031147E-06	0.4667868E-08	YSW=-0.2359842E-06	-0.7843211E-07
IJ=	60	YSD= 0.1182291E-06	0.1688500E-07	YSW=-0.2382930E-06	-0.6432303E-07
IJ=	61	YSD= 0.1319955E-06	-0.6060679E-07	YSW=-0.2399955E-06	-0.5028418E-07
IJ=	62	YSD= 0.1444220E-06	-0.1451612E-06	YSW=-0.2410875E-06	-0.3633278E-07
IJ=	63	YSD= 0.1555209E-06	-0.1479553E-06	YSW=-0.2415661E-06	-0.2248724E-07
IJ=	64	YSD= 0.1653083E-06	-0.7492224E-07	YSW=-0.2414303E-06	-0.8767000E-08
IJ=	65	YSD= 0.1738043E-06	-0.1716000E-07	YSW=-0.2406802E-06	0.4807561E-08
IJ=	66	YSD= 0.1810323E-06	-0.4463254E-07	YSW=-0.2393177E-06	0.1821532E-07

```
 IJ=  67 YSD= 0.1870195E-06  -0.1269922E-06  YSW=-0.2373465E-06   0.3143452E-07
 IJ=  68 YSD= 0.1917963E-06  -0.1712370E-06  YSW=-0.2347714E-06   0.4444292E-07
 IJ=  69 YSD= 0.1953963E-06  -0.1300020E-06  YSW=-0.2315990E-06   0.5721791E-07
 IJ=  70 YSD= 0.1978561E-06  -0.5448680E-07  YSW=-0.2278375E-06   0.6973664E-07
 IJ=  71 YSD= 0.1992151E-06  -0.3276671E-07  YSW=-0.2234963E-06   0.8197615E-07
 IJ=  72 YSD= 0.1995153E-06  -0.8747440E-07  YSW=-0.2185865E-06   0.9391350E-07
 IJ=  73 YSD= 0.1988012E-06  -0.1517164E-06  YSW=-0.2131207E-06   0.1055258E-06
 IJ=  74 YSD= 0.1971194E-06  -0.1488389E-06  YSW=-0.2071126E-06   0.1167906E-06
 IJ=  75 YSD= 0.1945186E-06  -0.8119605E-07  YSW=-0.2005777E-06   0.1276854E-06
 IJ=  76 YSD= 0.1910490E-06  -0.2514673E-07  YSW=-0.1935324E-06   0.1381884E-06
 IJ=  77 YSD= 0.1867627E-06  -0.4059746E-07  YSW=-0.1859948E-06   0.1482783E-06
 IJ=  78 YSD= 0.1817129E-06  -0.1021450E-06  YSW=-0.1779841E-06   0.1579345E-06
 IJ=  79 YSD= 0.1759542E-06  -0.1310470E-06  YSW=-0.1695205E-06   0.1671368E-06
 IJ=  80 YSD= 0.1695418E-06  -0.8821132E-07  YSW=-0.1606257E-06   0.1758661E-06
 IJ=  81 YSD= 0.1625316E-06  -0.1851879E-07  YSW=-0.1513222E-06   0.1841041E-06
 IJ=  82 YSD= 0.1549803E-06   0.3718924E-08  YSW=-0.1416337E-06   0.1918332E-06
 IJ=  83 YSD= 0.1469442E-06  -0.3794094E-07  YSW=-0.1315850E-06   0.1990371E-06
 IJ=  84 YSD= 0.1384802E-06  -0.8461234E-07  YSW=-0.1212014E-06   0.2057006E-06
 IJ=  85 YSD= 0.1296449E-06  -0.7293960E-07  YSW=-0.1105095E-06   0.2118093E-06
 IJ=  86 YSD= 0.1204940E-06  -0.9248197E-08  YSW=-0.9953651E-07   0.2173502E-06
 IJ=  87 YSD= 0.1110833E-06   0.3972809E-07  YSW=-0.8831027E-07   0.2223117E-06
 IJ=  88 YSD= 0.1014673E-06   0.2649978E-07  YSW=-0.7685936E-07   0.2266831E-06
 IJ=  89 YSD= 0.9169973E-07  -0.2218684E-07  YSW=-0.6521291E-07   0.2304555E-06
 IJ=  90 YSD= 0.8183300E-07  -0.3877182E-07  YSW=-0.5340056E-07   0.2336209E-06
 IJ=  91 YSD= 0.7191830E-07   0.4754270E-08  YSW=-0.4145236E-07   0.2361730E-06
 IJ=  92 YSD= 0.6200529E-07   0.6500414E-07  YSW=-0.2939869E-07   0.2381068E-06
 IJ=  93 YSD= 0.5214184E-07   0.7999154E-07  YSW=-0.1727024E-07   0.2394187E-06
 IJ=  94 YSD= 0.4237404E-07   0.4239320E-07  YSW=-0.5097850E-08   0.2401069E-06
 IJ=  95 YSD= 0.3274624E-07   0.6652215E-08  YSW= 0.7087508E-08   0.2401706E-06
 IJ=  96 YSD= 0.2330027E-07   0.2325351E-07  YSW= 0.1925484E-07   0.2396109E-06
 IJ=  97 YSD= 0.1407623E-07   0.7934773E-07  YSW= 0.3137318E-07   0.2384300E-06
 IJ=  98 YSD= 0.5111628E-08   0.1154926E-06  YSW= 0.4341171E-07   0.2366321E-06
 IJ=  99 YSD=-0.3558423E-08   0.9687795E-07  YSW= 0.5533979E-07   0.2342225E-06
 IJ= 100 YSD=-0.1190110E-07   0.5370288E-07  YSW= 0.6712710E-07   0.2312080E-06
 IJ= 101 YSD=-0.1988636E-07   0.4384140E-07  YSW= 0.7874361E-07   0.2275971E-06
 IJ= 102 YSD=-0.2748658E-07   0.8377850E-07  YSW= 0.9015981E-07   0.2233997E-06
 IJ= 103 YSD=-0.3467690E-07   0.1301444E-06  YSW= 0.1013466E-06   0.2186269E-06
 IJ= 104 YSD=-0.4143504E-07   0.1324581E-06  YSW= 0.1122756E-06   0.2132916E-06
 IJ= 105 YSD=-0.4774164E-07   0.9297958E-07  YSW= 0.1229190E-06   0.2074077E-06
 IJ= 106 YSD=-0.5357975E-07   0.6271296E-07  YSW= 0.1332496E-06   0.2009908E-06
 IJ= 107 YSD=-0.5893550E-07   0.7981566E-07  YSW= 0.1432413E-06   0.1940576E-06
 IJ= 108 YSD=-0.6379727E-07   0.1247026E-06  YSW= 0.1528685E-06   0.1866261E-06
 IJ= 109 YSD=-0.6815674E-07   0.1446568E-06  YSW= 0.1621069E-06   0.1787156E-06
 IJ= 110 YSD=-0.7200748E-07   0.1171176E-06  YSW= 0.1709328E-06   0.1703466E-06
 IJ= 111 YSD=-0.7534648E-07   0.7572305E-07  YSW= 0.1793239E-06   0.1615407E-06
 IJ= 112 YSD=-0.7817252E-07   0.6905992E-07  YSW= 0.1872587E-06   0.1523204E-06
 IJ= 113 YSD=-0.8048754E-07   0.1026431E-06  YSW= 0.1947172E-06   0.1427097E-06
 IJ= 114 YSD=-0.8229527E-07   0.1333889E-06  YSW= 0.2016803E-06   0.1327331E-06
 IJ= 115 YSD=-0.8360234E-07   0.1219574E-06  YSW= 0.2081303E-06   0.1224163E-06
 IJ= 116 YSD=-0.8441715E-07   0.7944448E-07  YSW= 0.2140508E-06   0.1117857E-06
 IJ= 117 YSD=-0.8475055E-07   0.5304804E-07  YSW= 0.2194267E-06   0.1008685E-06
 IJ= 118 YSD=-0.8461529E-07   0.6912677E-07  YSW= 0.2242444E-06   0.8969273E-07
 IJ= 119 YSD=-0.8402606E-07   0.1023341E-06  YSW= 0.2284917E-06   0.7828692E-07
 IJ= 120 YSD=-0.8299939E-07   0.1069955E-06  YSW= 0.2321576E-06   0.6668027E-07
 IJ= 121 YSD=-0.8155347E-07   0.7199077E-07  YSW= 0.2352329E-06   0.5490242E-07
 IJ= 122 YSD=-0.7970798E-07   0.3326932E-07  YSW= 0.2377098E-06   0.4298345E-07
 IJ= 123 YSD=-0.7748424E-07   0.3000378E-07  YSW= 0.2395819E-06   0.3095382E-07
 IJ= 124 YSD=-0.7490444E-07   0.5786035E-07  YSW= 0.2408445E-06   0.1884423E-07
 IJ= 125 YSD=-0.7199245E-07   0.7516525E-07  YSW= 0.2414945E-06   0.6685578E-08
 IJ= 126 YSD=-0.6877274E-07   0.5347459E-07  YSW= 0.2415300E-06  -0.5491114E-08
 IJ= 127 YSD=-0.6527063E-07   0.1127290E-07  YSW= 0.2409510E-06  -0.1765478E-07
 IJ= 128 YSD=-0.6151261E-07  -0.9049415E-08  YSW= 0.2397591E-06  -0.2977441E-07
 IJ= 129 YSD=-0.5752525E-07   0.7721951E-08  YSW= 0.2379572E-06  -0.4181910E-07
 IJ= 130 YSD=-0.5333570E-07   0.3207941E-07  YSW= 0.2355499E-06  -0.5375814E-07
         Interactions between slots 2 and  2
```

```
NOEL1=  31
NOEL2=  31
NS12=  61
```

```
LONGITUDINAL DISTANCE IN WAVELENGTHS IN FREE SPACE= 0.7155300E+00
```

```
LONGITUDINAL DISTANCE IN WAVELENGTHS IN WAVEGUIDE= 0.4897325E+00
```

```
SUM_MD= 0.3312877E-04      -0.1799215E-04
SUM_MW=-0.7872593E-04      -0.1516772E-04
SUM_M= 0.0000000E+00       0.0000000E+00


Z12_MD=-0.2354168E+01   0.1278543E+01
Z12_MW= 0.5594355E+01   0.1077835E+01


Y12_MD=-0.1824758E-01   0.9910215E-02
Y12_MW= 0.4336285E-01   0.8354495E-02

CMC_EXT=-0.2105943E+00   0.1143732E+00  CMC_INT= 0.5004480E+00   0.9641872E-01
```

apollo domain

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:39 AM
FILE PRINTED: 89/04/24 11:01 AM

```
C.....................................................................
C                           SLOT_DESIGN.FTN
C      This program solves the problem of a dielectric covered waveguide
C                                slot
C.....................................................................
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 RCUR,AICUR,CINC,ABS_CF
      COMPLEX YS,YS1S2,CI,SUM_MD,SUM_MW,SUM_M,CUR_RES,Z_SELF_RES
      COMPLEX YS_ADM,YSW_ADM,CONSTN,CONSTM,Z12_MD,Z12_MW
      COMPLEX Y12_MD,Y12_MW,CF,CMC_EXT,CMC_INT
      EXTERNAL F_EER
C..............................................................
C     This common statement is the same in GENERATE and MUTUAL_SLOT
C..............................................................
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C................................................................
C     This common statement is the same in MUTUAL_SLOT
C................................................................
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C     NSL.............. includes also the end points ..............
C................................................................
C     This common statement is the same in MUTUAL_SLOT
C................................................................
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C..............................................................
C     This common statement is the same in GENERATE
C..............................................................
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
     *CUR_RES(30,60)
C..............................................................
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
      COMMON/MAN/IBMATR(260,260)
C
      COMMON/WAY_OUT/RS10(7,7,200),XS10(7,7,200),SGMN(7,7,200),
     *RIJ(7,7,200)
C
      COMMON/B01/BJ0,BJ1
C
      OPEN(UNIT=05,FILE='DATA_WAVE_MUTUAL',STATUS='OLD')
      OPEN(UNIT=06,FILE='OUT_WAVE_MUTUAL',STATUS='OLD')
      OPEN(UNIT=07,FILE='PLOT_MUTUAL',STATUS='APPEND')
C
C     Subroutine DATA reads the values of the geometrical
C     parameters
C
      CALL DATA
C
      CALL F_EER
      CI=(0.0,1.0)
      ICUR=1
C
C     ICUR=0  resonant field derived from GENERATE
C     ICUR=1  we assume a form for the resonant field
C
      IF (ICUR.EQ.0) THEN
C.....................................................
C     Call GENERATE to find the resonant lengths of various
C     slots
C..............................................
C      DO 1 I_SLOT=1,NSLOTS
C           CALL GENERATE(I_SLOT)
```

```
C 1    CONTINUE
C
C.............................................................
      END IF
      NSL(1)=NSL(1)+2
      NSL(2)=NSL(2)+2
C
C.............................................................
C
C     Find common subsection length
C
      N_SLOT=1
      IF (N_SLOT.LE.4) THEN
          I_MIN=1
      ELSE
          I_MIN=N_SLOT-3
      END IF
C
      IF (N_SLOT.GT.(NSLOTS-4)) THEN
          I_MAX=NSLOTS
      ELSE
          I_MAX=N_SLOT+3
      END IF
      DLX=DLX_RES(I_MIN)
      DO 2 I=I_MIN,I_MAX
           IF (DLX_RES(I).LT.DLX) THEN
               DLX=DLX_RES(I)
           END IF
  2   CONTINUE
C
      WRITE (*,*) DLX
C
C     Interpolate the current of n_slot
C
      DLX_DIF=DABS(DLX_RES(N_SLOT)-DLX)
      IF (DLX_DIF.GT.1.D-5) THEN
          CALL CUBSPL(ICUR,DLX,1,N_SLOT,1)
          CALL CUBSPL(ICUR,DLX,1,N_SLOT,2)
          DLX_RES(N_SLOT)=DLX
          L_MAX=NSL(N_SLOT)
          DO 7 L=1,L_MAX
               CUR_RES(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
               WRITE (6,77) N_SLOT,L,CUR_RES(N_SLOT,L)
 77            FORMAT(5X,'N_SLOT=',I4,2X,'L=',I4,2X,'CUR=',
     *         E14.7,2X,E14.7)
  7       CONTINUE
      ELSE  IF (ICUR.EQ.1) THEN
          L_MAX=NSL(N_SLOT)
          DO 502 L=1,L_MAX
                 RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                 AICUR(L)=0.0
                 WRITE (6,601) L,RCUR(L),AICUR(L)
 601             FORMAT(2X,'L=',I4,2X,'RCUR(L)=',E14.7,2X,
     *           'AICUR(L)=',E14.7)
 502     CONTINUE
         DO 505 L=1,L_MAX
                CUR_RES(N_SLOT,L)=RCUR(L)+CI*AICUR(L)
 505     CONTINUE
      END IF
C
C     Interpolate the current on the other slots
C
      DO 33  M=I_MIN,I_MAX
             IF (M.EQ.N_SLOT) GO TO 33
C
C          Interpolate the current of m slot
```

```
C
               DLX_DIF=DABS(DLX_RES(M)-DLX)
               IF (DLX_DIF.GT.1.D-5) THEN
                      CALL CUBSPL(ICUR,DLX,1,M,1)
                      CALL CUBSPL(ICUR,DLX,1,M,2)
                      DLX_RES(M)=DLX
                      L_MAX=NSL(M)
                      DO 34 L=1,L_MAX
                           CUR_RES(M,L)=RCUR(L)+CI*AICUR(L)
                           WRITE (6,77) M,L,CUR_RES(M,L)
  34                  CONTINUE
               ELSE IF (ICUR.EQ.1) THEN
                      L_MAX=NSL(M)
                      DO 503 L=1,L_MAX
                             RCUR(L)=SIN(PI*(L-1)/(NSL(N_SLOT)-1))
                             AICUR(L)=0.0
                             WRITE (6,601) L,RCUR(L),AICUR(L)
 503                  CONTINUE
                      DO 506 L=1,L_MAX
                             CUR_RES(M,L)=RCUR(L)+CI*AICUR(L)
 506                  CONTINUE
              END IF
  33   CONTINUE
C
C
C      Call MUTUAL_SLOT to find mutual coupling between slot
C      n_slot and the neighboring slots
C
       CALL MUTUAL_SLOT(N_SLOT)
C
C
       DO 11 I=I_MIN,I_MAX
             J_MIN=I_MIN+1
             J_MAX=I_MAX
             DO 12 J=J_MIN,J_MAX
                   IJMAX=NSSL(I,J)
                   WRITE (6,13) I,J
  13               FORMAT(10X,'Interactions between slots',I2,' and ',
     *                   I2//)
                   DO 14 IJ=1,IJMAX
                         WRITE (6,15) IJ,YS_ADM(I,J,IJ),YSW_ADM(I,J,IJ)
  15                     FORMAT(1X,'IJ=',I4,1X,'YSD=',E14.7,2X,E14.7,
     *                   2X,'YSW=',E14.7,2X,E14.7)
  14               CONTINUE
  12         CONTINUE
  11   CONTINUE
C
       DLG=1.D0/DSQRT(1.D0-1.D0/(2.D0*AW)**2)
       IZ_MAX=NXOFF(2)
C
C       DO 108 IZ=31,IZ_MAX,2
C
       IZ_FIX=0.5D0*DLG/DLX
       DO 108 IZ=IZ_FIX,IZ_FIX
C
              NXOFF(2)=IZ
              CALL ARRANGE_MUTUAL
C
C      Find the center of n_slot
C
              NC0=(NSL(N_SLOT)+1)/2
C
C      Find the corresponding row for IBMATR
C
              I_ROW=0
              DO 20 I=I_MIN,N_SLOT
```

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

                IF (M.GT.1) I_COL=I_COL+NSL(M-1)
C
C         ICUR = 0   : We derive the current from GENERATE
C         ICUR = 1   : We assume a form for the current
C
                IN_MIN=1
                IN_MAX=NSL(N_SLOT)
                DO 4 IN=IN_MIN,IN_MAX
                     CONSTN=CUR_RES(N_SLOT,IN)/CUR_RES(N_SLOT,NC0)
C
C                    WRITE (6,88) N_SLOT,IN,CONSTN
C 88                 FORMAT(2X,'N=',I4,2X,'IN=',I4,5X,'CONSTN=',
C    *               E14.7,2X,E14.7//)
C
                     IM_MIN=1
                     IM_MAX=NSL(M)
                     DO 5 IM=IM_MIN,IM_MAX
                          CONSTM=CUR_RES(M,IM)/CUR_RES(M,NCI)
                          CON=CONSTN*CONSTM
                          IJ=I_ROW+IN
                          KJ=I_COL+IM
                          IK=IBMATR(IJ,KJ)
C
C                   WRITE (6,89) IM,CONSTM,IJ,KJ,IK
C 89                FORMAT(10X,'IM=',I4,2X,'CONSTM=',E14.7,2X,E14.7/
C    *              10X,'IJ=',I4,2X,'KJ=',I4,2X,'IK=',I4)
C
                         SUM_MD=SUM_MD+SNGL(CON)*YS_ADM(N_SLOT,M,IK)
                         SUM_MW=SUM_MW+SNGL(CON)*YSW_ADM(N_SLOT,M,IK)
  5                  CONTINUE
  4             CONTINUE
  3        CONTINUE
           DIST_X=(NXOFF(2)-NXOFF(1))*DLX
C
           WRITE (6,52) DIST_X
  52       FORMAT(///2X,'LONGITUDINAL DISTANCE IN',
     *     ' WAVELENGTHS IN FREE SPACE=',E14.7/)
C
           DIST_X=DIST_X/DLG
C
           WRITE (6,53) DIST_X
  53       FORMAT(///2X,'LONGITUDINAL DISTANCE IN',
     *     ' WAVELENGTHS IN WAVEGUIDE=',E14.7/)

           WRITE (6,60) SUM_MD,SUM_MW,SUM_M
```

```
 60         FORMAT (/10X,'SUM_MD=',E14.7,5X,E14.7/
     *      10X,'SUM_MW=',E14.7,5X,E14.7/10X,'SUM_M=',
     *      E14.7,5X,E14.7//)
C
C
            Z12_MD=-(120.0*SNGL(PI))**2*SUM_MD/2.0
            Z12_MW=-(120.0*SNGL(PI))**2*SUM_MW/2.0
C
            WRITE (6,61) Z12_MD,Z12_MW
  61        FORMAT(/10X,'Z12_MD=',E14.7,2X,E14.7/
     *      10X,'Z12_MW=',E14.7,2X,E14.7//)
C
            G0=DSQRT(1.D0-1.D0/(2.D0*AW)**2)/(120.D0*PI)
            Y12_MD=-SUM_MD/SNGL(G0)
            Y12_MW=-SUM_MW/SNGL(G0)
C
            WRITE (6,62) Y12_MD,Y12_MW
  62        FORMAT(/10X,'Y12_MD=',E14.7,2X,E14.7/
     *      10X,'Y12_MW=',E14.7,2X,E14.7//)
C
C
C
C
C     Evaluation of the coupling term  Mc
C
C
C
            B01=DSQRT(1.D0-(0.5D0/AW)**2)
            B012=B01*B01
            ARGY0=PI*YOFF(N_SLOT)/AW
            ARGP=PI*DLX*(B01+1.D0)
            ARGM=PI*DLX*(B01-1.D0)
            ARG0=AK0*DLX
C
            WRITE (*,*) ARGY0,ARG0
C
            ARG=B01*2.D0*PI*DLX
            CALL BSJ0(PI*WS(N_SLOT)/(2.D0*AW))
            DINC=(1.D0/(2*PI*AW))**2*(1.D0/(AW*BW))*(1.D0/DSIN(ARG0))
            DINC=DINC*DCOS(ARGY0)*BJ0*DSIN(ARGM)*DSIN(ARGP)/(B01*
     *          (1.D0-B012))
C
            WRITE (*,*) DINC
C
            CINC=SNGL(DINC)
            CF=(0.0,0.0)
            JQMAX=NSL(N_SLOT)
            DO 71 JQ=1,JQMAX
                 ARGX=ARG*FLOAT(JQ-1)
                 EC=DCOS(ARGX)
                 ES=DSIN(ARGX)
                 IF (ICUR.EQ.1)   CF=CF+SNGL(DSIN((JQ-1)*PI/
     *           (NSL(N_SLOT)-1)))*(SNGL(EC)+CI*SNGL(ES))
                 IF (ICUR.EQ.0)   CF=CF+CUR_RES(N_SLOT,JQ)*
     *           (SNGL(EC)+CI*SNGL(ES))
  71        CONTINUE
            CF=CINC*CF
            ABS_CF=CABS(CF)
C
            WRITE (*,*) CF,ABS_CF
C
            CMC_EXT=Y12_MD/(SNGL(32.D0*AW**2*PI**2*AW*BW)*ABS_CF**2)
            CMC_INT=Y12_MW/(SNGL(32.D0*AW**2*PI**2*AW*BW)*ABS_CF**2)
C
            WRITE (6,83) CMC_EXT,CMC_INT
 83         FORMAT (10X,'CMC_EXT=',E14.7,2X,E14.7,2X,'CMC_INT=',
```

```
     *       E14.7,2X,E14.7)
C
C
             WRITE (7,707) Y12_MD,Y12_MW,CMC_EXT,CMC_INT
707          FORMAT(E14.7,2X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7,
     *       2X,E14.7,2X,E14.7,2X,E14.7)
C
 108  CONTINUE
 1000 CONTINUE
      STOP
      END
C..............................................................
C     The name of this subroutine is       DATA
C     and gives all the data used by the main program and the other
C     subroutines.
C..............................................................
      SUBROUTINE DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION WORK(7,7)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      COMMON/MAT_DIEL/YS(200),YS1S2(7,200),NOFFS(7)
C
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
     *CUR_RES(30,60)
C
      PI=3.141592653589D0
C
C
C     ---- Dielectric constant ---
      READ (5,1) ER
  1   FORMAT (///6X,D16.9)
      WRITE (6,2) ER
  2   FORMAT(10X,'Dielectric Constant of the Substrate'/10X,E14.7//)
C
C     ---- Substrate Thickness ---
C
      READ (5,1) H
      WRITE (6,3) H
  3   FORMAT(10X,'Substrate Thickness'/10X,E14.7//)
C
C     ---- Conductor Thickness ---
C
      READ (5,1) T
      WRITE (6,4) T
  4   FORMAT(10X,'Conductor Thickness'/10X,E14.7//)
C
C     ---- Dimensions of the Waveguide ----
C
      READ (5,1) AW
      READ (5,10) BW
  10  FORMAT(6X,D16.9)
      WRITE (6,5) AW,BW
  5   FORMAT(10X,'Dimensions of the Waveguide'/10X,'AW=',E14.7/
     *10X,'BW=',E14.7//)
C
C     ---- Number of Slots ----
C
```

```
      READ (5,20) NSLOTS
   20 FORMAT(///6X,I4)
      WRITE (6,6) NSLOTS
    6 FORMAT(10X,'Number of Slots'/10X,'NSLOTS=',I4//)
C
C     ---- Limit for offsets: Small Offset< OFFLIM ----
C                             Large Offset> OFFLIM
C
      OFFLIM=0.1
C
C     ---- Transverse Offsets of the Slots ----
C
      READ(5,1) YOFF(1)
      WRITE (6,7) YOFF(1)
    7 FORMAT(10X,'Transverse Offsets of the Slots'/10X,
     *'YOFF(1)=',E14.7)
      IF (NSLOTS.GT.1) THEN
             DO 8 I=2,NSLOTS
                  READ(5,10) YOFF(I)
                  WRITE (6,9) I,YOFF(I)
    9             FORMAT(10X,'YOFF(',I2,')=',E14.7)
    8        CONTINUE
      END IF
      WRITE(6,60)
   60 FORMAT(10X,//)
C
C     ---- Longitudinal Offsets of the Slots ( in dlx )
C
      READ (5,20) NXOFF(1)
      WRITE (6,11) NXOFF(1)
   11 FORMAT(10X,'Longitudinal Offset of the Slots'/
     *10X,'NXOFF(1)=',I4)
      IF (NSLOTS.GT.1) THEN
             DO 12 I=2,NSLOTS
                  READ(5,30) NXOFF(I)
   30             FORMAT(6X,I4)
                  WRITE (6,13) I,NXOFF(I)
   13             FORMAT(10X,'NXOFF(',I2,')=',I4)
   12        CONTINUE
      END IF
      WRITE(6,60)
C
C     ---- Slot Widths ----
C
      READ(5,1) WS(1)
      WRITE(6,14) WS(1)
   14 FORMAT(10X,'Slot Widths'/10X,'WS(1)=',E14.7)
      IF (NSLOTS.GT.1) THEN
             DO 15 I=2,NSLOTS
                  READ(5,10) WS(I)
                  WRITE (6,16) I,WS(I)
   16             FORMAT(10X,'WS(',I2,')=',E14.7)
   15        CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Slots Excess Widths ----
C
      READ(5,1) WSDELTA(1)
      WRITE (6,17) WSDELTA(1)
   17 FORMAT(10X,'Slots Excess Widths'/10X,'WSDELTA=',
     *E14.7)
      IF (NSLOTS.GT.1) THEN
             DO 18 I=2,NSLOTS
                  READ(5,10) WSDELTA(I)
                  WRITE(6,19) I,WSDELTA(I)
```

```
 19               FORMAT(10X,'WSDELTA(',I2,')=',E14.7)
 18         CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Subsection Length ----
C
      READ (5,1) DLX_RES(1)
      WRITE (6,21) DLX_RES(1)
 21   FORMAT(10X,'Subsection Length'/10X,E14.7//)
      IF (NSLOTS.GT.1) THEN
            DO 40 I=2,NSLOTS
                  READ(5,10) DLX_RES(I)
                  WRITE(6,46) I,DLX_RES(I)
 46               FORMAT(10X,'DLX_RES(',I2,')=',E14.7)
 40         CONTINUE
      END IF
      WRITE (6,60)
C
C
C     ---- Lower Limit of the Tail Contribution ----
C
      READ (5,1) A
      WRITE (6,22) A
 22   FORMAT(10X,'Lower Limit of Tail Contribution'/10X,E14.7//)
C
C     ---- Number of Points on Each Slot ----
C
      READ (5,20) NSL(1)
      WRITE (6,23) NSL(1)
 23   FORMAT(10X,'Number of Points on Each Slot including the ends',
     */10X,'NSL(1)=',I4)
      IF (NSLOTS.GT.1) THEN
            DO 24 I=2,NSLOTS
                  READ(5,25) NSL(I)
 25               FORMAT(6X,I4)
                  WRITE(6,26) I,NSL(I)
 26               FORMAT(10X,'NSL(',I2,')=',I4)
 24         CONTINUE
      END IF
      WRITE (6,60)
C
C     ---- Error in the evaluation of the series ----
C
      READ (5,1) ERROR
      WRITE (6,27) ERROR
 27   FORMAT(10X,'Error in the evaluation of the series'/
     *10X,'ERROR=',E14.7//)
C
C     Initialize OFFSET( ) to 0
C
      DO 37 I=1,7
            OFFSET(I)=0.D0
 37   CONTINUE
C
C     Initialize NOFF to 1
C
      NOFF=1
      RETURN
      END
C-----------------------------------------------------------------------
C     THIS FUNCTION EVALUTES EER
C-----------------------------------------------------------------------
      SUBROUTINE F_EER
      IMPLICIT REAL*8 (A-H,O-Z)
C
```

```
C      ---- Normalization Constant ----
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
C       EER=ER+(1.D0-ER)*(W/H)/(1.D0+W/H)
C
      EER=1.0
      WRITE (6,100) EER
      WRITE(*,100) EER
 100  FORMAT(10X,'Normalization Constant'/10X,E14.7/)
      RETURN
      END
C-------------------------------------------------------------
C       NORMALIZATION SUBROUTINE
C
C       THIS SUBROUTINE DENORMALIZES WITH RESPECT TO CNORM_OLD
C       AND NORMALIZES AGAIN WITH RESPECT TO CNORM_NEW
C-------------------------------------------------------------
      SUBROUTINE NORM(CNORM_OLD,CNORM_NEW)
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/DAT_SUB/ER,H,T,DLX,AW,BW,A,TPI,TPI2,PI,E1,E2,EER,AK0,AK,
     *AKK,FA,OFFSET(7),OFFLIM,ERROR,NOFF
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/MUTUAL_AD_MAT/YS_ADM(7,7,200),YSW_ADM(7,7,200)
C
      COMMON/IOFF/INSS(7,7),NSSL(7,7)
C
      CNORM=CNORM_OLD/CNORM_NEW
C
      PI=3.141592654
C
      AK0=2.D0*PI*CNORM_NEW
      AKK=2.D0*PI
      AK=AK0*DSQRT(ER)
C
      H=H*CNORM
      AW=AW*CNORM
      BW=BW*CNORM
      T=T*CNORM
      DLX=DLX*CNORM
      OFFLIM=OFFLIM*CNORM
C
      YOFF(1)=YOFF(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 8 I=2,NSLOTS
                  YOFF(I)=YOFF(I)*CNORM
  8          CONTINUE
      END IF
C
      WS(1)=WS(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 15 I=2,NSLOTS
                  WS(I)=WS(I)*CNORM
  15         CONTINUE
      END IF
C
      WSDELTA(1)=WSDELTA(1)*CNORM
      IF (NSLOTS.GT.1) THEN
             DO 18 I=2,NSLOTS
                  WSDELTA(I)=WSDELTA(I)*CNORM
```

```
 18         CONTINUE
      END IF
      RETURN
      END
C*******************************************************************
C........................ Spline Interpolation .....................
C*******************************************************************
      SUBROUTINE  CUBSPL(ICUR,DLX,IEND,N_SLOT,IRX)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX CURRENT,CUR_RES,Z_SELF_RES,CC
      REAL*4 RCUR,AICUR,REAL_CUR,AIMAG_CUR
      DIMENSION S(260),A(260,4),X(260),Y(260),AI(260),BI(260),
     *CI(260),DI(260)
C
      COMMON/SLOTS/YOFF(30),NXOFF(30),WS(30),WSDELTA(30),NSL(30),NSLOTS
C
      COMMON/RES/S_LENGTH(30),DLX_RES(30),Z_SELF_RES(30),
     *CUR_RES(30,60)
C
      COMMON/SPLINE/RCUR(60),AICUR(60)
C
C     This routine computes the matrix for finding the coefficients of a
C     cubic spline through a set of data.
C     The  system is then solved to obtain the second derivative values,
C     and the coefficients of the cubic spline between each pair of points.
C      -----------------------------------------------------------------------
C      Parameters are
C      X,Y        Arrays of X and Y values to be fitted
C
C      DLX         Subsection length (if all points have same spacing)
C
C      S          Array of second derivative values at the points
C
C      N          Number of points
C
C      IEND       Type of end condition to be used
C                     IEND=1, Linear ends, S(1)=S(N)=0
C                     IEND=2, Parabolic ends, S(1)=S(2), S(N)=S(N-1)
C                     IEND=3, Cubic ends S(1),S(N) are extrapolated
C
C      A           Augmented matrix of coefficients and R.H.S. for finding S
C
C      IRX        1  : Interpolate the real part of the current
C                 2  : Interpolate the imaginary part of the current
C
C      ICUR      =0  resonant field derived from GENERATE
C      ICUR      =1  we assume a form for the resonant field
C      -----------------------------------------------------------------------
       PI=3.141592654
       N=NSL(N_SLOT)
       CC=(0.0,1.0)
C
C      Computation of matrices X,Y
C
       NC0_OLD=(NSL(N_SLOT)+1)/2
       NSLOT_NEW=2*NINT((NSL(N_SLOT)-1)*DLX_RES(N_SLOT)/(2.0*DLX))+1
       NC0_NEW=(NSLOT_NEW+1)/2
       ITEST=(NSLOT_NEW+1)-NC0_NEW*2
       I_CUR=(NSLOT_NEW+1)/2
       I_MIN=1
       I_MAX=NSL(N_SLOT)
       L_MAX=NSLOT_NEW
C
       WRITE (*,*) L_MAX
C
       IF (ICUR.EQ.1) GO TO 500
```

```
      DO 1 I=I_MIN,I_MAX
           X(I)=DLX_RES(N_SLOT)*FLOAT(I-1)
           REAL_CUR=REAL(CUR_RES(N_SLOT,I))
           CURRENT=-CC*CUR_RES(N_SLOT,I)
           AIMAG_CUR=REAL(CURRENT)
           IF (IRX.EQ.1) Y(I)=DBLE(REAL_CUR)
           IF (IRX.EQ.2) Y(I)=DBLE(AIMAG_CUR)
           WRITE (6,67) I,X(I),Y(I)
 67        FORMAT(10X,'I=',I4,2X,'X=',E14.7,2X,'Y=',E14.7)
  1   CONTINUE
C     ---------------------------------------------------------------------------------
C     | Compute the N-2 rows |
C     ------------------------
C
      NM2=N-2
      NM1=N-1
      DX1=X(2)-X(1)
      DY1=(Y(2)-Y(1))/DX1*6.D0
      DO 10 I=1,NM2
        DX2=X(I+2)-X(I+1)
        DY2=(Y(I+2)-Y(I+1))/DX2*6.D0
        A(I,1)=DX1
        A(I,2)=2.D0*(DX1+DX2)
        A(I,3)=DX2
        A(I,4)=DY2-DY1
        DX1=DX2
        DY1=DY2
10    CONTINUE
C
C    Adjust first and last rows to end condition
C
      GO TO  (20,50,80), IEND
C
C    for IEND = 1 no change is needed
C
20    GO TO 100
C
C    for IEND = 2, S(1)=S(2), S(N)=S(N-1), parabolic ends.
C
50    A(1,2)=A(I,2)+X(2)-X(1)
      A(NM2,2)=A(NM2,2)+X(N)-X(NM1)
      GO TO 100
C
C    for IEND = 3, cubic ends, S(1), S(N) are extrapolated.
C
80    DX1=X(2)-X(1)
      DX2=X(3)-X(2)
      A(1,2)=(DX1+DX2)*(DX1+2.D0*DX2)/DX2
      A(1,3)=(DX2*DX2-DX1*DX1)/DX2
      DXN2=X(NM1)-X(NM2)
      DXN1=X(N)-X(NM1)
      A(NM2,1)=(DXN2*DXN2-DXN1*DXN1)/DXN2
      A(NM2,2)=(DXN1+DXN2)*(DXN1+2.D0*DXN2)/DXN2
      GO TO 100
C
C    Now we solve the tridiagonal system.  First reduce
C
100   DO 110 I=2,NM2
        A(I,2)=A(I,2)-A(I,1)/A(I-1,2)*A(I-1,3)
        A(I,4)=A(I,4)-A(I,1)/A(I-1,2)*A(I-1,4)
110   CONTINUE
C
C    Back substitution
C
      A(NM2,4)=A(NM2,4)/A(NM2,2)
      DO 120 I=2,NM2
```

```
          J=NM1-I
          A(J,4)=(A(J,4)-A(J,3)*A(J+1,4))/A(J,2)
120     CONTINUE
C
C     Place values in S-vector
C
        DO 130 I=1,NM2
          S(I+1)=A(I,4)
130     CONTINUE
C
C     Set S(1) and S(N) according to end conditions
C
        GO TO (150,160,170), IEND
C
C     Linear ends
C
150     S(1)=0.
        S(N)=0.
        GO TO 200
C
C     Parabolic ends
C
160     S(1)=S(2)
        S(N)=S(N-1)
        GO TO 200
C
C     For cubic ends
C
170     S(1)=((DX1+DX2)*S(2)+DX1*S(3))/DX2
        S(N)=((DXN2+DXN1)*S(NM1)-DXN1*S(NM2))/DXN2
C
C     Find spline fit coefficients
C
C
C     Evaluation of the coefficients ai,bi,ci,di - Store into AI,BI
C     CI,DI
C
200      DO 210 I=1,NM1
           AI(I)=(S(I+1)-S(I))/(6.D0*DLX_RES(N_SLOT))
           BI(I)=S(I)/2.D0
           CI(I)=(Y(I+1)-Y(I))/DLX_RES(N_SLOT)-(2.D0*S(I)+S(I+1))
     *               *DLX_RES(N_SLOT)/6.D0
210        DI(I)=Y(I)
C
C      Re-evaluate nsl(n_slot) and cur_res(n_slot)
C
      DO 2 I=1,I_CUR
           IF (ITEST.EQ.0) NCP=NC0_NEW+I-1
           IF (ITEST.EQ.1) NCP=NC0_NEW+I
           NCM=NC0_NEW-I+1
           DISTP=(NCP-1)*DLX
           DISTM=(NCM-1)*DLX
           RIP=DISTP/DLX_RES(N_SLOT)
           IP=INT(RIP)
           IF ((RIP-IP).GT.0.999) IP=IP+1
           IF(IP.EQ.NSLOT_NEW) IP=IP-1
           RIM=DISTM/DLX_RES(N_SLOT)
           IM=INT(RIM)
           IF((RIM-IM).GT.0.999) THEN
                IM=IM+1
           END IF
           DIFP=DISTP-FLOAT(IP)*DLX_RES(N_SLOT)
           DIFM=DISTM-FLOAT(IM)*DLX_RES(N_SLOT)
           DIFP2=DIFP*DIFP
           DIFM2=DIFM*DIFM
           DIFP3=DIFP2*DIFP
```

```
          DIFM3=DIFM2*DIFM
          IF (IRX.EQ.1) THEN
             IP=IP+1
             IM=IM+1
             RCUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                 CI(IP)*DIFP+DI(IP))
             RCUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                 CI(IM)*DIFM+DI(IM))
             WRITE (6,666) NCP,IP,NCM,IM,RCUR(NCP),RCUR(NCM)
 666         FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *       I4/30X,'RCUR(NCP)=',E14.7,2X,'RCUR(NCM)=',E14.7)
          END IF
          IF (IRX.EQ.2) THEN
             IP=IP+1
             IM=IM+1
             AICUR(NCP)=SNGL(AI(IP)*DIFP3+BI(IP)*DIFP2+
     *                  CI(IP)*DIFP+DI(IP))
             AICUR(NCM)=SNGL(AI(IM)*DIFM3+BI(IM)*DIFM2+
     *                  CI(IM)*DIFM+DI(IM))
             WRITE (6,777) NCP,IP,NCM,IM,AICUR(NCP),AICUR(NCM)
 777         FORMAT(2X,'NCP=',I4,2X,'IP=',I4,2X,'NCM=',I4,2X,'IM=',
     *       I4/10X,'AICUR(NCP)=',E14.7,2X,'AICUR(NCM)=',E14.7)
          END IF
  2   CONTINUE
C
      IF (IRX.EQ.2) NSL(N_SLOT)=NSLOT_NEW
      RETURN
C
 500  CONTINUE
      DO 502 L=1,L_MAX
            IF (IRX.EQ.1) THEN
                 RCUR(L)=SIN(PI*(L-1)/(NSLOT_NEW-1))
                 WRITE (6,601) L,RCUR(L)
 601             FORMAT(10X,'L=',I4,2X,'RCUR=',E14.7)
            ELSE IF (IRX.EQ.2) THEN
                 AICUR(L)=0.0
                 WRITE (6,602) L,AICUR(L)
 602             FORMAT(2X,'L=',I4,2X,'AICUR=',E14.7)
            END IF
 502     CONTINUE
C
      IF (IRX.EQ.2) NSL(N_SLOT)=NSLOT_NEW
C
      RETURN
      END
```

apollo domain

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:35 AM
FILE PRINTED: 89/04/24 10:55 AM

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

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

```
                  IMIN=I
                  JMIN=J
                  DO 33 L=I_MIN,I_MAX
                        K_MIN=L+1
                        K_MAX=I_MAX
                        DO 34 K=K_MIN,K_MAX
                              IF (TEST.GT.WORK(L,K)) THEN
                                  TEST=WORK(L,K)
                                  IMIN=L
                                  JMIN=K
                              END IF
   34                   CONTINUE
   33             CONTINUE
                  DO 35 N=1,IJ
                        IF (TEST.EQ.OFFSET(N)) THEN
                            INSS(IMIN,JMIN)=N
                            IF(NOFFS(N).LT.NSSL(IMIN,JMIN)) THEN
                                   NOFFS(N)=NSSL(IMIN,JMIN)
                            END IF
                            GO TO 36
                        END IF
   35             CONTINUE
                  IJ=IJ+1
                  OFFSET(IJ)=TEST
                  INSS(IMIN,JMIN)=IJ
                  NOFFS(IJ)=NSSL(IMIN,JMIN)
   36             WORK(IMIN,JMIN)=100
   32       CONTINUE
   31 CONTINUE
C
      WRITE (6,80)
  80  FORMAT(///10X,'Offsets for the dielectric layer and number',
     *' of corresponding elements'/)
      DO 51 I=I_MIN,I_MAX
            WRITE (6,52) I,OFFSET(I),NOFFS(I)
  52        FORMAT(10X,'I=',I4,2X,'OFFSET=',E14.7,2X,'NOFFS=',I4)
  51  CONTINUE
      WRITE (6,90)
  90  FORMAT (///10X,'SLOTS and corresponding offsets in the dielectric'
     *,/)
      DO 53 I=I_MIN,I_MAX
            J_MIN=I+1
            J_MAX=I_MAX
            DO 54 J=J_MIN,J_MAX
                  IJ=INSS(I,J)
                  WRITE(6,55) I,J,INSS(I,J),OFFSET(IJ)
  55              FORMAT(10X,'I=',I4,2X,'J=',I4,2X,'INSS=',I4,
     *            2X,'OFFSET=',E14.7/)
  54        CONTINUE
  53  CONTINUE
C
C     ---- Evaluation of the Max Number of Offsets for the Diel. ----
C
      NOFF=IJ
C
      WRITE (6,56) NOFF
  56  FORMAT(///10X,'Max number of offsets in the dielectric'/
     *10X,'NOFF=',I4//)
C
      RETURN
      END
```

a p o l l o   d o m a i n

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:36 AM
FILE PRINTED: 89/04/24 11:00 AM

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

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

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

a p o l l o   d o m a i n

CAEN/Apollo

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

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

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

LAST MODIFIED ON: 89/04/24 10:37 AM
FILE PRINTED: 89/04/24 11:07 AM

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

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

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

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

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

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

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

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

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