See the next few pages for a Thesis code .... in ASAP & part in Fortran

ASAP Utilities

user guide

easy to use tools that save time
and speed up your work in Excel

author	Bastien Mensink
pages	375
latest revision	May 30, 2016
ASAP Utilities version	7.0

A Must in Every Office BV
Punterweg 20
8042 PB Zwolle
The Netherlands

www.amustineveryoffice.com
www.asap-utilities.com

Information in this document is subject to change without notice.

While every precaution has been taken in the preparation of this user guide, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained in herein. Microsoft, Microsoft Office, and Excel are registered trademarks of Microsoft Corporation.

```
      PROGRAM ASAP
C     DOUBLE PRECISION
C      PARAMETER (ISIZE=100)
C
C     CHANGE THE ISIZE^^^^ IN THE PARAMETER STATEMENT TO SCALE THE
C      COMPILED CODE FOR THE SIZE OF THE MAXIMUM PROBLEM THAT A
C      PARTICULAR COMPUTER/COMPILER WILL ALLOW
C
C       ASAP VERSION 3.2 FOR PC/MAINFRAME/WORKSTATION
C
C         V3.2D JULY 2004 ADDS LIST INPUT FOR NEAR-FIELD POINTS WITH NEW
C          KEYWORD 'NEAR' // CHANGED INPUT/OUTPUT FILENAME SO THAT IN.TXT
C           IS ALWAYS THE INPUT FILENAME AND OUT.TXT IS ALWAYS THE
C             OUTPUT FILENAME
C                FIXES OTHER THINGS LIKE THE SEGMENT AND NODE NUMBER PRINTOUT
C                TO HANDLE NUMBERS UP TO 6 DIGITS
C                 ALSO CHANGED THE FIXED FORMAT FLOATING POINT PRINT
C                 TO AN EXP TYPE PRINT SO THAT SMALL VALUES DON'T DISAPPEAR TO ZERO
C          FIXED VARIOUS DOUBLE PRECISION WARNING AND ERRORS THAT CAUSE PROBLEMS
C           WITH SOME COMPILERS
C
C         V3.1D CORRECTS A VARIBLE INITIALIZATION PROBLEM THAT CAUSES PROBLEMS
C              WITH INSULATED COATED WIRES AND LOSSY DIELECTRIC EXTERNAL MEDIA
C
C       ASAP VERSION 3.0 FOR PC/MAINFRAME/WORKSTATION
C         V3.0D CORRECTS SEVERAL MINOR BUGS MOSTLY THOSE THAT MAKE THE
C              SOURCE CODE COMPILER SENSITIVE;
C              ALSO FIXED BUG WHICH REPORT LESS GAIN THAN EXPECTED FOR
C              GROUNDPLANE CASES - JANUARY 1998
C
C         V2.0 ADDED THE LIST FORM OF GEOMETRY INPUT AND
C              NODE CONNECTION LIST AND CORRECTED MINOR BUGS
C              ADDED THE PARAMETER STATEMENT TO MAKE THE SIZE OF
C              PROBLEM HANDLED BY PROGRAM SCALEABLE  - 6 JULY 1989
C
C         V1.0 FIXED SOURCE CODE TO COMPILE ON ALMOST ANY FORTRAN COMPILER
C              (FIRST PC VERSION)
C              FIXED SEVERAL PROBLEMS RELATED TO VARIABLES, INDEXING AND
C              GROUND PLANE CONNECTIONS - ?? 1986-1989 ??
C
C         V0.0 THE ORIGINAL IBM 360 VERSION - 1974
C
C
C  *** NOTE ON DOUBLE PRECISION VERSION ***
C
C       MODIFIED TO DP 26JULY97 RAY L. CROSS
C
C     CONVERSION FROM SINGLE TO DOUBLE PRECISION OCCURED AS
C     FOLLOWS:
C       ALL "COMPLEX" DECLARATIONS WERE CHANGED TO "COMPLEX*16"
C       ALL LOCAL IMPLICIT REAL*4 WERE FOUND WITH COMPILER LISTING
C         AND EXPLICITLY DECLARED REAL*8
C       ALL IMPLICIT REAL*4 SUBROUTINE PARAMETERS CHANGED TO REAL*8
C
C     ALL "DOUBLE PRECISION" DECLARATIONS WERE ALREADY CONTAINED IN THE
C      SINGLE PRECISION VERSION OF THE CODE.  THERE WERE NO REAL*8 OR
```

```
C      COMPLEX*16 STATEMENTS IN THE ORIGINAL CODE.
C
C     ALL IMPLICIT INTEGERS HAVE BEEN LEFT IMPLICIT
C
C  *** END NOTE ON DOUBLE PRECISION ***
C
      REAL*8 AM, ABAP, ABAT, ABIP, ABIT, ACSP, ACST, AFFT, AFFP
      REAL*8 BM, CMM, CTHET, CPHI
      REAL*8 D, DATY1, DATY2, DATY3, DATY4
      REAL*8 E0, ECST, EFF, EPMAG, ETMAG, ECSP, ER2, ER3, ER4
      REAL*8 FHZ, FMC, GG, GPP, GTT, HGT, OMEGA, PI, PH, PHAF
      REAL*8 PHAI, PHII, PHIF, PHSF, PHSI, PHSPH, PHSTH, SCSP, SCST
      REAL*8 SIG2, SIG3, SIG4, SPPM, SPTM, STEP, STPM, STTM, STTMTP
      REAL*8 TD2, TD3, TH, THAF, THAI, THII, THIF, THSF, THSI, TP
      REAL*8 U0, X, XG, XP, XNP, Y, YG, YP, YNP, Z, ZG, ZP, ZNP, ZMIN
      DIMENSION X(ISIZE), Y(ISIZE), Z(ISIZE), XG(ISIZE)
      DIMENSION YG(ISIZE), ZG(ISIZE)
      DIMENSION I1(ISIZE), I2(ISIZE), I3(ISIZE), JA(ISIZE)
      DIMENSION JB(ISIZE), KFLAG(30)
      DIMENSION CPHI(500), CTHET(500)
      DIMENSION DATY1(360), DATY2(360), DATY3(360), DATY4(360)
      DIMENSION D(ISIZE), IA(ISIZE), IB(ISIZE), ISC(ISIZE)
      DIMENSION MD(ISIZE,4), ND(ISIZE), LZD(ISIZE), KGEN(ISIZE)
      COMMON IWL
      DIMENSION XNP(ISIZE), YNP(ISIZE), ZNP(ISIZE)
      COMPLEX*16 C(ISIZE*ISIZE/2+ISIZE/2)
      COMPLEX*16 CDAT1(360),CDAT2(360),CDAT3(360),CDAT4(360)
      COMPLEX*16 CJ(ISIZE),EP(ISIZE),EPP(ISIZE),ET(ISIZE),ETT(ISIZE)
      COMPLEX*16 CGD(ISIZE),SGD(ISIZE),CG(ISIZE*2),VG(ISIZE*2)
      COMPLEX*16 ZLD(ISIZE*2)
      COMPLEX*16 VOLT(ISIZE),ZLLD(ISIZE)
      COMPLEX*16 EPPS,EPTS,ETPS,ETTS,EX,EY,EZ
      COMPLEX*16 EP2,EP3,EP4,ERR,ETA,GAM,Y11,Z11,ZS
      DATA PI,TP/3.141592653589793,6.283185307179586/
      DATA E0,U0/8.854E-12,1.2566E-6/
C
C      OPEN STATEMENTS MOVED TO THE BEGINNING BEFORE STATEMENT LABEL 1
C      SO THAT END CARD WILL NOT CAUSE PROBLEMS - JAN 1998
C
C     OPEN STATEMENTS MAY HAVE TO BE CHANGED TO CORRESPOND TO USER'S PREFERENCE
C      OR TO WORK WITH A PARTICULAR COMPILER
C
      OPEN(5,FILE='in.txt')
      OPEN(6,FILE='out.txt ')
C
C        ** STATEMENT 1 WAS MADE A CONTINUE - JAN 1998
    1 CONTINUE
      NGEN = -1
      IGRD = -1
      LOAD = -1
      BM = -1
      ICARD = 0
      AM = -1
      IFLAG = 0
      VOLT(1) = (1.,0.)
      HGT = 0.
      NM = 0
      NP = 0
      MSG = 0
C
C     FOLLOWING 3 VARIBLES INITIALIZATIONS CHANGED TO =0 TO SOLVE SOME
C      INSULATION AND EXTERIOR MEDIA INPUT DATA PROBLEMS MARCH 1998
C
C     SIG2 = -1.
C     TD2 = -1.
C     SIG3 = -1
C
      SIG2 = 0.
      TD2 = 0.
      SIG3 = 0
C
C
      ER3 = 1
      TD3 = 0.
      CMM = 50.
      ER2 = 1.
      FMC = 300.
      INM = ISIZE
      ICJ = ISIZE
C
C    MOVED THE FILE OPEN STATEMENTS FROM HERE TO THE BEGINNING OF THE PROGRAM -
JAN 1998
C
      WRITE (6,74)
      WRITE (6,740)
C
      DO 2 I=1,30
    2 KFLAG(I) = -1
C
C
      DO 3 J=1,INM
      ISC(J) = 0
      VG(J) = (.0,.0)
      ZLD(J) = (.0,.0)
      JJ = J+INM
      VG(JJ) = (.0,.0)
    3 ZLD(JJ) = (.0,.0)
C
    4 NFFP = 0
      NBIP = 0
      NBAP = 0
      AFFP = 1000.
      AFFT = 1000.
      ABIP = 1000.
      ABIT = 1000.
      ABAP = 1000.
      ABAT = 1000.
      STEP = 1.
      KNM = 0
      CALL READD(IA,IB,IBISC,ICARD,IGAIN,IGRD,INEAR,INT,ISCAT,IWR,IFLAG,
     1KFLAG,KGEN,LOAD,LZD,MSG,NBAP,NBIP,NFFP,NGEN,NM,NP,ABAP,ABAT,AFFP,A
     2FFT,ABIP,ABIT,AM,BM,CMM,ER2,ER3,ER4,FMC,HGT,PHAF,PHAI,PHIF,PHII,PH
     3SF,PHSI,THAF,THAI,THIF,THII,THSF,THSI,SIG2,SIG3,SIG4,TD2,TD3,VOLT,
     4X,XNP,Y,YNP,Z,ZLLD,ZNP,STEP)
      WRITE (6,56)
      IF (MSG.LT.1) GO TO 5
      IF (MSG.EQ.1) WRITE (6,70) KFLAG(30)
      IF (IFLAG.EQ.4) GO TO 1
    5 IF (IFLAG.EQ.5) STOP
      IF (AM.LT.0) WRITE (6,127)
      IF (AM.LT.0) GO TO 6
      IF ((NM.GT.0).AND.(NP.GT.0)) GO TO 7
      WRITE (6,116)
    6 IF (IFLAG.EQ.1) GO TO 1
      MSG = 2
      GO TO 4
    7 WRITE (6,114)
      WRITE (6,113)
      WRITE (6,112)
      IF (KFLAG(1).EQ.1) WRITE (6,83) FMC
      IF (KFLAG(2).EQ.1) WRITE (6,84) AM
      IF (KFLAG(3).EQ.1) WRITE (6,85) CMM
      IF (KFLAG(20).NE.1) WRITE (6,87)
      IF (KFLAG(4).EQ.1) WRITE (6,86)
      IF (KFLAG(4).EQ.1) WRITE (6,88) BM
      IF (KFLAG(5).EQ.1) WRITE (6,89) SIG2
      IF (KFLAG(6).EQ.1) WRITE (6,90) ER2
      IF (KFLAG(7).EQ.1) WRITE (6,91) TD2
      IF (KFLAG(8).NE.1) WRITE (6,92)
      IF (KFLAG(9).EQ.1) WRITE (6,93) SIG3
      IF (KFLAG(10).EQ.1) WRITE (6,94) ER3
      IF (KFLAG(11).EQ.1) WRITE (6,95) TD3
      IF (KFLAG(26).NE.1) WRITE (6,122)
      IF ((IGRD.GT.1).AND.(KFLAG(25).EQ.1)) WRITE (6,123)
      IF ((IGRD.EQ.1).AND.(KFLAG(25).EQ.1)) WRITE (6,125)
      IF ((IGRD.GT.1).AND.(KFLAG(25).EQ.1)) WRITE (6,124) ER4,SIG4
      IF ((IGRD.GT.0).AND.(KFLAG(25).EQ.1)) WRITE (6,126) HGT
      IF (KFLAG(21).EQ.1) WRITE (6,121) INT
      WRITE (6,111)
      IF (KFLAG(12).EQ.1) WRITE (6,96) (I,IA(I),X(IA(I)),Y(IA(I)),Z(IA(I
     1)),IB(I),X(IB(I)),Y(IB(I)),Z(IB(I)),I=1,NM)
      WRITE (6,111)
      IF (KFLAG(24).GT.0) WRITE (6,119) (LZD(I),ZLLD(I),I=1,LOAD)
      IF (KFLAG(14).GT.0) WRITE (6,118) (LZD(I),ZLLD(I),I=1,LOAD)
      WRITE (6,111)
      IF (KFLAG(23).GT.0) WRITE (6,120) (KGEN(I),VOLT(I),I=1,NGEN)
      IF (KFLAG(13).GT.0) WRITE (6,97) (KGEN(I),VOLT(I),I=1,NGEN)
      WRITE (6,111)
      WRITE (6,114)
      WRITE (6,98)
      WRITE (6,112)
      IF (KFLAG(22).NE.1) WRITE (6,110)
      IF (KFLAG(15).EQ.1) WRITE (6,99)
      IF (KFLAG(16).EQ.1) WRITE (6,100) PHAI,PHAF,THAI,THAF,STEP
      IF (KFLAG(17).EQ.1) WRITE (6,101) PHII,PHIF,THII,THIF,STEP
      IF (KFLAG(18).EQ.1) WRITE (6,102) PHSI,PHSF,THSI,THSF,STEP
C
C     V3.2D CHANGE LOGIC ON KFLAG(19) SO THAT NEW 'NEAR' KEYWORD FLAG VALUE
```

```
C     WILL STILL GENERATE PRINT STATEMENTS FOR THE NEARFIELD LIST
C        RAY L. CROSS 18 JULY 2004
C
      IF (KFLAG(19).GT.0) WRITE(6,103)
      IF (KFLAG(19).GT.0) WRITE(6,130)(XNP(I),YNP(I),ZNP(I),I=1,INEAR)
C
      IF (AFFP.LT.500.) WRITE (6,105) AFFP
      IF (AFFT.LT.500.) WRITE (6,104) AFFT
      IF (ABAP.LT.500.) WRITE (6,109) ABAP
      IF (ABAT.LT.500.) WRITE (6,108) ABAT
      IF (ABIP.LT.500.) WRITE (6,107) ABIP
      IF (ABIT.LT.500.) WRITE (6,106) ABIT
      IF ((IBISC.GT.0).AND.(ISCAT.LT.0)) WRITE (6,73)
      IF (KFLAG(4).LT.1) GO TO 129
      DO 128 I=1,INM
  128 ISC(I)=1
  129 FHZ=FMC*1.E6
      OMEGA = TP*FHZ
C
C      THE FOLLOWING 4 LINES HAVE LOGIC CHANGED TO ACCOUNT FOR CHANGE IN
C       SIG2, SIG3, TD2, CHANGE TO BE INITIALIZED TO BE =0   -- MARCH 1998
C
C     IF (SIG2.LT.0.) EP2=ER2*E0*CMPLX(1.,-TD2)
C     IF (TD2.LT.0.) EP2 = CMPLX(ER2*E0,-SIG2/OMEGA)
C     IF (SIG3.LT.0.) EP3=ER3*E0*CMPLX(1.,-TD3)
C     IF (TD3.LT.0.) EP3 = CMPLX(ER3*E0,-SIG3/OMEGA)
C
C       LOGIC CHANGED FROM ORIGINAL TO ACCOUNT FOR THE INITIALIZATION TO ZERO
C
      EP2=ER2*E0*DCMPLX(DBLE(1.0),DBLE(-TD2))
C
C      **** V3.2D ABOVE LINE CHANGED FROM  EP2=ER2*E0*CMPLX(1.,-TD2)
C
      IF (SIG2.GT.0.) EP2 = CMPLX(ER2*E0,-SIG2/OMEGA)
C
      EP3=ER3*E0*DCMPLX(DBLE(1.0),DBLE(-TD3))
C
C      **** V3.2D ABOVE LINE CHANGED FROM  EP3=ER3*E0*CMPLX(1.,-TD3)
C
      IF (SIG3.GT.0.) EP3 = CMPLX(ER3*E0,-SIG3/OMEGA)
C
C
      IF (IGRD.GT.1) EP4 = CMPLX(ER4*E0,-SIG4/OMEGA)
      IF (IGRD.GT.1) ERR = EP4/EP3
      IF (KFLAG(21).GT.0) WRITE (6,121) INT
C
C       **** V3.2D FOLLOWING 2 LINES FIXED FOR DOUBLE COMPLEX
      ETA = CDSQRT(U0/EP3)
      GAM = OMEGA*CDSQRT(-U0*EP3)
      IF (KFLAG(12).NE.1) GO TO 9
      NPG = NP
      NMG = NM
C
      DO 8 I=1,NPG
      XG(I) = X(I)
      YG(I) = Y(I)
```

```
    8 ZG(I) = Z(I)
C
C
    9 DO 10 I=1,NPG
      X(I) = XG(I)
      Y(I) = YG(I)
   10 Z(I) = ZG(I)
C
      NP = NPG
      NM = NMG
      IWL = 0
      IF (IGRD.LE.0) GO TO 15
C     SET UP IMAGE FOR GROUND PLANE
      ZMIN = Z(1)+HGT
      K = 0
C
      DO 11 I=1,NP
      Z(I) = Z(I)+HGT
      IF (Z(I).LT.ZMIN) ZMIN=Z(I)
      IF (Z(I).GT.1.E-30) GO TO 11
      IWL = IWL+1
   11 CONTINUE
C
      IF (ZMIN.GE.0.0) GO TO 12
      WRITE (6,117)
      IF (IFLAG.EQ.1) GO TO 1
      IF (IFLAG.EQ.2) STOP
      MSG = 2
      GO TO 4
C
   12 DO 13 J=1,NM
      K = J+NM
      IA(K) = IA(J)
      IF (IA(J).GT.IWL) IA(K)=IA(J)+NP-IWL
   13 IB(K) = IB(J)+NP-IWL
C
      IWLP = IWL+1
C
      DO 14 I=IWLP,NP
      J = I+NP-IWL
      X(J) = X(I)
      Y(J) = Y(I)
   14 Z(J) = -Z(I)
C
      KNM = NM+1
      NM = 2*NM
      NP = 2*NP-IWL
   15 CALL SORT (IA,IB,I1,I2,I3,JA,JB,MD,ND,NM,NP,N,MAX,MIN,ICJ,INM)
      IF (MAX.LE.4) GO TO 16
      WRITE (6,71)
      IF (IFLAG.EQ.1) GO TO 1
      IF (IFLAG.EQ.2) STOP
      MSG = 2
      GO TO 4
   16 IF (MIN.GE.1) GO TO 17
      WRITE (6,72)
      IF (IFLAG.EQ.1) GO TO 1
      IF (IFLAG.EQ.2) STOP
      MSG = 2
      GO TO 4
   17 WRITE (6,56)
      IF (MAX.GT.4.OR.MIN.LT.1.OR.N.GT.ICJ) GO TO 54
      I12 = 1
      IF (LOAD.GT.0) GO TO 19
C
      DO 18 I=1,NM
   18 ZLD(I) = (0.,0.)
C
   19 IF (NGEN.GT.0) GO TO 21
C
      DO 20 I=1,NM
   20 VG(I) = (0.,0.)
C
   21 KN = NM
      IF (IGRD.GT.0) KN = NM/2
      J = 1
C     ANTENNA CALCULATIONS
      IF (LOAD.LE.0) GO TO 24
      KNAA = KN
      IF (KFLAG(24).GT.0) KNAA = 1
C
      DO 23 J=1,KNAA
C
   22 CONTINUE
      DO 23 I=1,LOAD
      K = LZD(I)
      IF ((IA(J).EQ.K).AND.(KFLAG(14).GT.0)) ZLD(J)=ZLLD(I)
      IF (KFLAG(24).GT.0) ZLD(K)=ZLLD(I)
      IF ((KFLAG(14).GT.0).AND.(IGRD.GT.0)) ZLD(J+KN)=ZLD(J)
      IF ((KFLAG(24).GT.0).AND.(IGRD.GT.0)) ZLD(K+KN)=ZLD(K)
   23 CONTINUE
C
   24 IF (NGEN.LT.0) GO TO 27
      KN = NM
      IF (IGRD.GT.0) KN = NM/2
      KNAA = KN
      IF (KFLAG(23).GT.0) KNAA = 1
C
      DO 26 J=1,KNAA
C
   25 CONTINUE
      DO 26 I=1,NGEN
      K = KGEN(I)
      IF ((IA(J).EQ.K).AND.(KFLAG(13).GT.0)) VG(J)=VOLT(I)
      IF (KFLAG(23).GT.0) VG(K)=VOLT(I)
      IF ((KFLAG(13).GT.0).AND.(IGRD.GT.0)) VG(J+KN)=-VG(J)
      IF ((IGRD.GT.0).AND.(KFLAG(23).GT.0))VG(K+KN)=-VG(K)
   26 CONTINUE
C
   27 CALL SGANT (IA,IB,INM,INT,ISC,I1,I2,I3,JA,JB,MD,N,ND,NM,NP,AM,BM,C
     1,CGD,CMM,D,EP2,EP3,ETA,FHZ,GAM,SGD,X,Y,Z,ZLD,ZS,ERR,IGRD)
      IF (N.GT.0) GO TO 28
      IF (IFLAG.EQ.2) STOP
      MSG = 2
      IF (IFLAG.EQ.1) GO TO 1
      GO TO 4
   28 IF (NGEN.LE.0) GO TO 36
      WRITE (6,75)
      WRITE (6,76)
      WRITE (6,77)
      WRITE (6,82)
      CALL GANT1 (IA,IB,INM,IWR,I1,I2,I3,I12,JA,JB,MD,N,ND,NM,AM,C,CJ,CG
     1,CMM,D,EFF,GAM,GG,CGD,SGD,VG,Y11,Z11,ZLD,ZS,IGRD)
C
C     ** LINE ADDED JAN 1998 TO FIX REPORTED POWER INPUT FOR GROUNDPLANE CASES
      IF (IGRD.GT.0) GG=GG/2.0
C
C
      WRITE (6,57) EFF,GG,Z11
C
C     NEAR FIELD
      IF (INEAR.LE.0) GO TO 30
      WRITE (6,75)
      WRITE (6,78)
      WRITE (6,77)
C
C
C     V3.2D ADD NEW LIST TYPE OUTPUT FOR NEAR FIELD POINTS IF THE 'NEAR' KEYWORD IS
USED
C      INSTEAD OF THE NEAR INSIDE OF AN OUTPUT STATEMENT - RAY L. CROSS 18 JULY 2004
C
      IF (KFLAG(19).EQ.2) WRITE (6,140)
C
      DO 29 I=1,INEAR
      XP = XNP(I)
      YP = YNP(I)
      ZP = ZNP(I)
      CALL GNFLD (IA,IB,INM,I1,I2,I3,MD,N,ND,NM,AM,CGD,SGD,ETA,GAM,CJ,D,
     1X,Y,Z,XP,YP,ZP,EX,EY,EZ,IGRD,ERR)
C
C     V3.2D CHANGE LOGIC SO THAT ALTERNATE LIST OF NEAR FIELD POINTS CAN
C      BE WRITTEN IN THE NEW 'NEAR' KEYWORD IS USED
C         RAY L CROSS 18 JULY 2004
C
      IF (KFLAG(19).EQ.2) THEN
      WRITE (6,143) XP,YP,ZP,EX,EY,EZ
      ELSE
      WRITE (6,58) XP,YP,ZP
      WRITE (6,59) EX,EY,EZ
      END IF
   29 CONTINUE
C
C     FAR FIELD
   30 IF (IGAIN.LE.0) GO TO 36
C
      DO 31 I=1,360
      DATY1(I) = 0
      DATY2(I) = 0
      DATY3(I) = 0
   31 DATY4(I) = 0
C
      WRITE (6,75)
      WRITE (6,79)
      WRITE (6,77)
      WRITE (6,82)
      INC = 0
      NPL = -1
      IF (KFLAG(16).EQ.1) WRITE (6,69)
      IF (NFFP.EQ.1) GO TO 32
      NPHA = (PHAF-PHAI)/STEP+1
      NTHA = (THAF-THAI)/STEP+1
      GO TO 34
   32 IF (AFFT.GT.500.) GO TO 33
      NPL = 1
      NPHA = 360
      NTHA = 1
      PHAI = 0.
      THAI = AFFT
      STEP = 1.
      GO TO 34
   33 NPL = 2
      NPHA = 1
      NTHA = 360
      PHAI = AFFP
      THAI = 0.
      STEP = 1.
   34 PH = PHAI-STEP
      DO 35 K=1,NPHA
      PH = PH+STEP
      TH = THAI-STEP
      DO 35 I=1,NTHA
      PHSPH = 0.
      PHSTH = 0.
      TH = TH+STEP
      IF ((IGRD.GT.0).AND.((TH.GT.90).AND.(TH.LT.270))) GO TO 35
      CALL GFFLD (IA,IB,INC,INM,IWR,I1,I2,I3,I12,MD,N,ND,NM,AM,ACSP,ACST
     1,C,CGD,CG,CJ,CMM,D,ECSP,ECST,EP,ET,EPP,ETT,EPPS,EPTS,ETPS,ETTS,GG,
     2GPP,GTT,PH,SGD,SCSP,SCST,SPPM,SPTM,STPM,STTM,TH,X,Y,Z,ZLD,ZS,ETA,G
     3AM,ERR,IGRD)
C
C        *** V3.2D FOLLOWING 2 LINES FIXED FOR DOUBLE COMPLEX
      ETMAG = CDABS(ETTS)
      EPMAG = CDABS(EPPS)
C
C        *** V3.2D FOLLOWING 4 LINES FIXED FOR DOUBLE COMPLEX
      IF(ETMAG.GT.1.E-32) PHSTH=57.29577951308232 *
     1 DATAN2(DIMAG(ETTS),DBLE(ETTS))
      IF(EPMAG.GT.1.E-32) PHSPH=57.29577951308232 *
     1 DATAN2(DIMAG(EPPS),DBLE(EPPS))
      IF (NPL.EQ.1) DATY1(K)=EPMAG
      IF (NPL.EQ.1) DATY2(K)=ETMAG
      IF (NPL.EQ.2) DATY1(I)=EPMAG
      IF (NPL.EQ.2) DATY2(I)=ETMAG
      IF (KFLAG(16).NE.1) GO TO 35
      WRITE (6,60) TH,PH,GTT,GPP,ETTS,ETMAG,PHSTH,EPPS,EPMAG,PHSPH
   35 CONTINUE
C
      WRITE (6,56)
      IF (NPL.LE.0) GO TO 36
      CALL POLPRT (1,DATY1)
      CALL POLPRT (2,DATY2)
C
C     BACK SCATTERING
   36 IF (ISCAT.LE.0) GO TO 54
      WRITE (6,75)
      WRITE (6,80)
      WRITE (6,77)
      WRITE (6,82)
      L = 0
      NPL = -1
      INC = 1
      IF (NBAP.EQ.1) GO TO 37
      NPHI = (PHIF-PHII)/STEP+1
      NTHI = (THIF-THII)/STEP+1
      IF (IWR.LE.0) WRITE (6,62)
      GO TO 39
   37 IF (ABAT.GT.500.) GO TO 38
      NPL = 1
      NPHI = 360
      NTHI = 1
      PHII = 0.
      THII = ABAT
      STEP = 1.
      GO TO 39
   38 NPL = 2
      NPHI = 1
      NTHI = 360
      PHII = ABAP
      THII = 0.
      STEP = 1.
   39 PH = PHII-STEP
C
      DO 42 K=1,NPHI
      PH = PH+STEP
      TH = THII-STEP
C
      DO 42 I=1,NTHI
      TH = TH+STEP
      IF ((IGRD.GT.0).AND.((TH.GT.90).AND.(TH.LT.270))) GO TO 42
      L = L+1
      CALL GFFLD (IA,IB,INC,INM,IWR,I1,I2,I3,I12,MD,N,ND,NM,AM,ACSP,ACST
     1,C,CGD,CG,CJ,CMM,D,ECSP,ECST,EP,ET,EPP,ETT,EPPS,EPTS,ETPS,ETTS,GG,
     2GPP,GTT,PH,SGD,SCSP,SCST,SPPM,SPTM,STPM,STTM,TH,X,Y,Z,ZLD,ZS,ETA,G
     3AM,ERR,IGRD)
      IF (IWR.GT.0) GO TO 40
      IF (NPL.LT.0) WRITE (6,63) PH,TH,SPPM,SPTM,STPM,STTM,ACSP,ACST,ECS
     1P,ECST,SCSP,SCST
   40 CPHI(L) = PH
      CTHET(L) = TH
      CDAT1(L) = EPPS
      CDAT2(L) = EPTS
      CDAT3(L) = ETPS
      CDAT4(L) = ETTS
      IF (NPL.NE.1) GO TO 41
C
C       *** V3.2D FOLLOWING 4 LINES FIXED FOR DOUBLE COMPLEX
      DATY1(K) = CDABS(EPPS)
      DATY2(K) = CDABS(EPTS)
      DATY3(K) = CDABS(ETPS)
      DATY4(K) = CDABS(ETTS)
      GO TO 42
C
C       *** V3.2D FOLLOWING 4 LINES FIXED FOR DOUBLE COMPLEX
   41 DATY1(I) = CDABS(EPPS)
      DATY2(I) = CDABS(EPTS)
      DATY3(I) = CDABS(ETPS)
      DATY4(I) = CDABS(ETTS)
   42 CONTINUE
C
      WRITE (6,82)
      IF (NPL.LE.0) GO TO 43
      CALL POLPRT (7,DATY1)
      CALL POLPRT (8,DATY2)
      CALL POLPRT (9,DATY3)
      CALL POLPRT (10,DATY4)
      IF (KFLAG(17).NE.1) GO TO 45
   43 WRITE (6,64)
C
      DO 44 I=1,L
   44 WRITE (6,65) CPHI(I),CTHET(I),CDAT1(I),CDAT2(I),CDAT3(I),CDAT4(I)
C
C     BISTATIC SCATTERING
   45 IF (IBISC.LE.0) GO TO 54
      WRITE (6,75)
      WRITE (6,81)
      WRITE (6,77)
      WRITE (6,82)
      WRITE (6,61) CPHI(L),CTHET(L)
      WRITE (6,82)
      L = 0
      INC = 2
      NPL = -1
      IF (NBIP.EQ.1) GO TO 46
      NPHS = (PHSF-PHSI)/STEP+1
      NTHS = (THSF-THSI)/STEP+1
      IF (IWR.LE.0) WRITE (6,67)
      GO TO 48
   46 IF (ABIT.GT.500.) GO TO 47
      NPL = 1
      NPHS = 360
      NTHS = 1
      PHSI = 0.
      THSI = ABIT
      STEP = 1.
      GO TO 48
   47 NPL = 2
```

```
      NPHS = 1
      NTHS = 360
      PHSI = ABIP
      THSI = 0.
      STEP = 1.
   48 PH = PHSI-STEP
C
      DO 511 K=1,NPHS
      PH = PH+STEP
      TH = THSI-STEP
      IF ((IGRD.GT.0).AND.((TH.GT.90).AND.(TH.LT.270))) GO TO 511
C
      DO 51 I=1,NTHS
      TH = TH+STEP
      L = L+1
      CALL GFFLD (IA,IB,INC,INM,IWR,I1,I2,I3,I12,MD,N,ND,NM,AM,ACSP,ACST
     1,C,CGD,CG,CJ,CMM,D,ECSP,ECST,EP,ET,EPP,ETT,EPPS,EPTS,ETPS,ETTS,GG,
     2GPP,GTT,PH,SGD,SCSP,SCST,SPPM,SPTM,STPM,STTM,TH,X,Y,Z,ZLD,ZS,ETA,G
     3AM,ERR,IGRD)
      IF (IWR.GT.0) GO TO 49
      IF (NPL.LT.0) WRITE (6,63) PH,TH,SPPM,SPTM,STPM,STTM
   49 CPHI(L) = PH
      CTHET(L) = TH
      CDAT1(L) = EPPS
      CDAT2(L) = EPTS
      CDAT3(L) = ETPS
      CDAT4(L) = ETTS
      IF (NPL.NE.1) GO TO 50
C
C        *** V3.2D FOLLOWING 4 LINES FIXED FOR DOUBLE COMPLEX
      DATY1(K) = CDABS(EPPS)
      DATY2(K) = CDABS(EPTS)
      DATY3(K) = CDABS(ETPS)
      DATY4(K) = CDABS(ETTS)
   50 IF (NPL.NE.2) GO TO 51
C
C        *** V3.2D FOLLOWING 4 LINES FIXED FOR DOUBLE COMPLEX
      DATY1(I) = CDABS(EPPS)
      DATY2(I) = CDABS(EPTS)
      DATY3(I) = CDABS(ETPS)
      DATY4(I) = CDABS(ETTS)
   51 CONTINUE
  511 CONTINUE
C
      WRITE (6,82)
      IF (NPL.LE.0) GO TO 52
      CALL POLPRT (3,DATY1)
      CALL POLPRT (4,DATY2)
      CALL POLPRT (5,DATY3)
      CALL POLPRT (6,DATY4)
      IF (KFLAG(18).NE.1) GO TO 54
   52 WRITE (6,66)
C
      DO 53 I=1,L
   53 WRITE (6,65) CPHI(I),CTHET(I),CDAT1(I),CDAT2(I),CDAT3(I),CDAT4(I)
C
```

```
   54 IF (IFLAG.EQ.1) GO TO 1
      IF (IFLAG.EQ.2) STOP
C
      KKFLAG=0
      KJFLAG=0
      KMFLAG=0
      KNFLAG=0
      IF (KFLAG(13).GT.0) KKFLAG=1
      IF (KFLAG(23).GT.0) KJFLAG=1
      IF (KFLAG(14).GT.0) KMFLAG=1
      IF (KFLAG(24).GT.0) KNFLAG=1
      DO 55 I=1,30
   55 KFLAG(I) = -1
C
      KFLAG(8) = 1
      KFLAG(20) = 1
      KFLAG(26) = 1
      IF (KKFLAG.GT.0) KFLAG(13)=1
      IF (KJFLAG.GT.0) KFLAG(23)=1
      IF (KMFLAG.GT.0) KFLAG(14)=1
      IF (KNFLAG.GT.0) KFLAG(24)=1
      IF (IFLAG.EQ.3) WRITE (6,68)
      IF (IFLAG.EQ.6) WRITE (6,115)
      GO TO 4
C
   56 FORMAT (1H0)
C
C        POWER INPUT CHANGED TO E11.5 FORMAT
   57 FORMAT (10X,'THE RADIATION EFFICIENCY (PERCENT) IS ',F15.7//10X,'T
     1HE TIME-AVERAGE POWER INPUT IS ',E11.5//10X,'THE ANTENNA IMPEDANCE
     2 IS ',F15.7,' +J',F15.7//)
C
C    V3.2D CHANGED FORMAT OF NEAR FIELD REPORTING TO E11.5 FROM F15.7
C      RAY L. CROSS 18 JULY 2004
C
   58 FORMAT (2X,'THE NEAR-FIELD ELECTRIC FIELD INTENSITY AT THE OBSERV
     1ATION POINT ',E11.5,', ',E11.5,', ',E11.5,' (X,Y,Z RESPECTIVELY)
     2IS:'//)
   59 FORMAT (20X,'EX= ',E11.5,' +J ',E11.5/20X,'EY= ',E11.5,' +J ',
     1E11.5/20X,'EZ= ',E11.5,' +J ',E11.5////)
C
   60 FORMAT (3X,F5.1,2X,F5.1,3X,E10.4,2X,E10.4,2(3X,3(E10.4,2X)),F6.1,1X
     1))
   61 FORMAT (T41,'FOR BISTATIC SCATTERING THE INCIDENT'/T41,'PLANE WAVE
     1 IS PHI=',F5.1,' THETA=',F5.1///)
   62 FORMAT (' INCIDENT',T27,'ECHO AREA SIGMA',T66,'ABSORPTION',T90,'EX
     1TINCTION',T114,'SCATTERING'/'  PLANE',T25,'(INCIDENT-SCATTERED)',1
     24X,3(5X,'CROSS SECTION',6X)/'  WAVE ',52X,3(10X,'FOR',11X)/'  PHI
     3 THETA',3X,'PHI-PHI',3X,'PHI-THETA',4X,'THETA-PHI',2X,'THETA-THETA
     4',3(5X,'PHI',7X,'THETA',4X))
   63 FORMAT (1X,2(F5.1,1X),10(E10.4,2X))
   64 FORMAT (T54,'BACKSCATTERING'/' INCIDENT',T37,'ELECTRIC FIELD POLAR
     1IZATION SCATTERING MATRIX'/'  PLANE',T49,'(INCIDENT-SCATTERED)'/3X
     2,'WAVE',T23,'PHI-PHI',T49,'PHI-THETA',T75,'THETA-PHI',T102,'THETA-
     3THETA'/'  PHI  THETA',3X,4(3X,'REAL',8X,'IMAG',8X))
   65 FORMAT (1X,2(F5.1,1X),2X,4(E11.5,2X,E11.5,3X))
```

```
 66 FORMAT (T54,'BISTATIC'/T37,'ELECTRIC FIELD POLARIZATION SCATTERING
  1 MATRIX'/' OBSERVATION',T50,'(INCIDENT-SCATTERED)'/'   POINT',14X,
  2 'PHI-PHI',T49,'PHI-THETA',T76,'THETA-PHI',T101,'THETA-THETA'/'  P
  3HI  THETA',4X,4(3X,'REAL',8X,'IMAG',8X))
 67 FORMAT (' OBERSVATION',T27,'ECHO AREA SIGMA'/'   POINT',T25,'(INCI
  1DENT-SCATTERED)'/'  PHI  THETA',T14,'PHI-PHI',T24,'PHI-THETA',T37,
  2  'THETA-PHI',T48,'THETA-THETA')
 68 FORMAT (1H1,5X,'CONTINUE EXECUTION WITH THE FOLLOWING ADDITIONS AN
  1D/OR CHANGES'//)
 69 FORMAT (54X,'ELECTRIC FIELD INTENSITY'/5X,'DEGREES',11X,'POWER GAI
  1N',28X,'THETA',42X,'PHI'/3X,'THETA',3X,'PHI',7X,'THETA',8X,'PHI',1
  2X,2(8X,'REAL',8X,'IMAG',8X,'MAGN',5X,'PHASE'))
 70 FORMAT (10X,'*****ERROR IN DATA CARD NUMBER ',I2,'  EXECUTION STOP
  1PED*******')
 71 FORMAT (40X,'*     A WIRE SEGMENT MAYNOT BE SHARED BY MORE THAN FO
  1UR    *'/40X,'*     DIPOLE MODES---------CHECK DESCRIPTION DATA CA
  2RD     *'/40X,'*              EXECUTION STOPPED
  3        *')
 72 FORMAT (40X,'*     AN ISOLATED WIRE MUST HAVE AT LEAST TWO SEGMENT
  1S     *'/40X,'*     AND THREE POINTS-----CHECK DESCRIPTION DATA CA
  2RD     *'/40X,'*              EXECUTION STOPPED
  3        *')
 73 FORMAT (30X,'A BACKSCATTERING CALL MUST BE INCLUDED FOR A BISTATIC
  1 CALL'//50X,'REQUEST IGNORED'/////)
 74 FORMAT ('1',T50,37('*')/T50,'*',T86,'*'/
  1 T50,'*                                   *'/
  2 T50,'*      OHIO STATE UNIVERSITY        *'/
  3 T50,'*    ANTENNA ANALYSIS PROGRAM       *'/
  4 T50,'*       MODIFIED FOR USE AT         *'/
  5 T50,'*     NAVAL POSTGRADUATE SCHOOL     *'/
  6 T50,'*         17 JULY 1974              *'/ )
740 FORMAT (' ',T50,37(' ')/T50,'*',T86,'*'/
  2 T50,'*                                   *'/
  3 T50,'*    FURTHER MODIFIED JULY 1989     *'/
  4 T50,'*           FOR USE ON              *'/
  5 T50,'*  PC * WORKSTATIONS * MAINFRAMES   *'/
  6 T50,'*                                   *'/
  7 T50,'*  VERSION 3.2D DOUBLE PRECISION    *'/
  8 T50,'*           (JULY 2004)             *'/
  9 T50,'*',T86,'*'/T50,37('*'))
 75 FORMAT ('1',T50,29('*')/T50,'*',T78,'*')
 76 FORMAT (T50,'*',11X,'ANTENNA',T78,'*')
 77 FORMAT (T50,'*',8X,'CALCULATIONS',T78,'*'/T50,'*',T78,'*'/T50,29('
  1*'))
 78 FORMAT (T50,'*',9X,'NEAR FIELD',T78,'*')
 79 FORMAT (T50,'*',9X,'FAR FIELD',T78,'*')
 80 FORMAT (T50,'*',7X,'BACKSCATTERING',T78,'*')
 81 FORMAT (T50,'*',4X,'BISTATIC SCATTERING',T78,'*')
 82 FORMAT (////)
 83 FORMAT (T30,'FREQUENCY (MHZ)',T81,E11.5)
 84 FORMAT (T30,'WIRE RADIUS (METERS)',T81,E11.5)
 85 FORMAT (T30,'WIRE CONDUCTIVITY (MEGAMHOS/METER)',T81,E11.5)
 86 FORMAT (T30,'WIRE INSULATED (NO/YES)',T85,'YES')
 87 FORMAT (T30,'WIRE INSULATED (NO/YES)',T85,'NO')
 88 FORMAT (T30,'INSULATION RADIUS (METERS)',T81,E11.5)
 89 FORMAT (T30,'INSULATION CONDUCTIVITY (MHOS/METER)',T81,E11.5)
```

```
   90 FORMAT (T30,'INSULATION DIELECTRIC CONSTANT (RELATIVE)',T81,E11.5)
   91 FORMAT (T30,'INSULATION LOSS TANGENT',T81,E11.5)
   92 FORMAT (T30,'EXTERIOR MEDIUM',T81,'FREE SPACE')
   93 FORMAT (T30,'EXTERIOR MEDIUM CONDUCTIVITY (MHOS/METER)',T81,E11.5)
   94 FORMAT (T30,'EXTERIOR MEDIUM DIELECTRIC CONSTANT (RELATIVE)',T81,
     1 E11.5)
   95 FORMAT (T30,'EXTERIOR MEDIUM LOSS TANGENT',T81,E11.5)
C
C     V3.2D REPLACE ORIGINAL FORMAT WITH SOMETHING THAT ALLOWS HIGHER NODE
C      AND SEGMENT NUMBERS RAY L CROSS 18 JULY 2004
C
   96 FORMAT (T50,'WIRE STRUCTURE'//T8,'SEG',8X,2('NODE',19X,'LOCATION'
     1,24X)/T9,'NO.',3X,2('     NO.',11X,'X',13X,'Y',13X,'Z',7X)/(T9,I6
     2,5X,2(I6,5X,E11.5,4X,E11.5,4X,E11.5,2X)))
   97 FORMAT (T54,'ANTENNA FEEDS'/T44,'NODE',16X,'VOLTS'/T45,'NO.',12X,
     1 'REAL',7X,'IMAGINARY'/(T41,I6,6X,2(4X,E11.5)))
C
C     THE ORIGINAL FORMAT LINES
C
C   96 FORMAT (T50,'WIRE STRUCTURE'//T20,'SEG',4X,2('NODE',19X,'LOCATION'
C     1,18X)/T21,'NO.',3X,2(' NO.',9X,'X',13X,'Y',13X,'Z',7X)/(T21,I2,5X,
C    22(I2,5X,E11.5,4X,E11.5,4X,E11.5,1X)))
C   97 FORMAT (T50,'ANTENNA FEEDS'/T40,'NODE',16X,'VOLTS'/T41,'NO.',12X,
C     1 'REAL',7X,'IMAGINARY'/(T41,I2,6X,2(4X,E11.5)))
C
   98 FORMAT (T50,'*', 6X,'OUTPUT REQUESTED',T78,'*')
   99 FORMAT (T30,'STRUCTURE CURRENTS')
  100 FORMAT (T30,'FAR FIELDS FOR PHI VARYING FROM',1X,F5.1,' TO ',F5.1,
     1 'AND THETA VARYING FROM ',F5.1,' TO ',F5.1/
     2T50,'IN STEPS OF ',F5.1,' DEGREES.')
  101 FORMAT (T30,'BACKSCATTERING FOR PHI VARYING FROM ',F5.1,' TO ',F5.
     11,' AND THETA VARYING FROM ',F5.1,' TO ',F5.1/
     2T50,'IN STEPS OF ',F5.1,' DEGREES.')
  102 FORMAT (T30,'BISTATIC SCATTERING FOR PHI VARYING FROM ',F5.1,' TO
     1',F5.1,' AND THETA VARYING FROM ',F5.1,' TO ',F5.1/
     2T50,'IN STEPS OF ',F5.1,' DEGREES.')
  103 FORMAT (T30,'NEAR FIELDS FOR FOLLOWING POINTS (X,Y,Z)')
  104 FORMAT (T30,'PLOT FOR FAR FIELD THETA=',F5.1)
  105 FORMAT (T30,'PLOT FOR FAR FIELD PHI=',F5.1)
  106 FORMAT (T30,'PLOT FOR BISTATIC SCATTERING-FOR THETA=',F5.1)
  107 FORMAT (T30,'PLOT FOR BISTATIC SCATTERING FOR PHI=',F5.1)
  108 FORMAT (T30,'PLOT FOR BACKSCATTERING THETA=',F5.1)
  109 FORMAT (T30,'PLOT FOR BACKSCATTERING PHI=',F5.1)
  110 FORMAT (T30,'NO OUTPUT OR PLOTS REQUESTED')
  111 FORMAT (//)
  112 FORMAT (T50,'*',T78,'*'/T50,29('*'))
  113 FORMAT (T50,'*', 8X,'INPUT DATA ',T78,'*')
  114 FORMAT (T50,29('*')/T50,'*',T78,'*')
  115 FORMAT (10X,'SINCE THIS DATA BLOCK DOES NOT HAVE A TERMINATION CAR
     1D A CHANGE CARD IS ASSUMED')
  116 FORMAT (//10X,40('*')/10X,'THE DESCRIPTION AND THE GEOMETRY OF THE
     1 STRUCTURE'/10X,'MUST BE STATED IN THE FIRST DATA BLOCK.'/10X,'***
     2* EXECUTION STOPPED ***')
  117 FORMAT (//10X,'NO PART OF THE WIRE STRUCTURE CAN LIE BELOW THE GRO
     1 UND PLANE.'/10X,'****EXECUTION STOPPED****')
C
```

```
C     V3.2D FORMATS MODIFIED 18 JULY 2004 TO ALLOW LARGER NUMBER OF NODES AND
SEGMENTS
C
  118 FORMAT (T54,'STRUCTURE LOADS'/T40,'    NODE',16X,'OHMS'/T41,
     1'    NO.',12X ,'REAL',7X,'IMAGINARY'/(T41,I6,6X,2(4X,E11.5)))
  119 FORMAT (T54,'STRUCTURE LOADS'/T39,'    SEGMENT',14X,'OHMS'/T41,
     1'    NO',12X,'REAL',7X,'IMAGINARY'/(T41,I6,6X,2(4X,E11.5)))
  120 FORMAT (T54,'ANTENNA FEEDS'/T39,'    SEGMENT',14X,'VOLTS'/T41,
     1'    NO.',12X,'REAL',7X,'IMAGINARY'/(T41,I6,6X,2(4X,E11.5)))
C
C     ORIGINAL FORMAT STATMENTS
C
C  118 FORMAT (T50,'STRUCTURE LOADS'/T40,'NODE',16X,'OHMS'/T41,'NO.',12X
C     1 ,'REAL',7X,'IMAGINARY'/(T41,I2,6X,2(4X,E11.5)))
C  119 FORMAT (T50,'STRUCTURE LOADS'/T39,'SEGMENT',14X,'OHMS'/T41,'NO',12
C     1X,'REAL',7X,'IMAGINARY'/(T41,I2,6X,2(4X,E11.5)))
C  120 FORMAT (T50,'ANTENNA FEEDS'/T39,'SEGMENT',14X,'VOLTS'/T41,'NO.',12
C     1X,'REAL',7X,'IMAGINARY'/(T41,I2,6X,2(4X,E11.5)))
C
  121 FORMAT (//T30,'THE NUMBER OF INTERVALS FOR CALCULATING THE ELEMENT
     1S'/T30,'IN THE IMPEDANCE MATRIX WITH SIMPSONS-RULE INTEGRATION IS'
     2,/T30,I3,'.  IF CLOSED FORM INTEGRATION IS REQUIRED SET INT=0'///)
  122 FORMAT (T30,'GROUND PLANE (NO/YES)',T85,'NO')
  123 FORMAT (T30,'GROUND PLANE (NO/YES)',T85,'YES')
  124 FORMAT (T30,'GROUND DIELECTRIC CONSTANT (RELATIVE)',T81,E11.5/
     1 T30,'GROUND CONDUCTIVITY (MHOS/METER)',T81,E11.5)
  125 FORMAT (T30,'GROUND PLANE',T83,'PERFECT')
  126 FORMAT (T30,'ANTENNA HEIGHT (METERS)',T81,E11.5)
  127 FORMAT (//10X,40('*')/10X,'THE WIRE RADIUS MUST BE STATED'/10X,40(
     1'*'))
  130 FORMAT(T40,E11.5,5X,E11.5,5X,E11.5)
C
C     V3.2D FORMAT STATEMENTS ADDED FOR NEW LIST OUTPUT OF NEAR FIELD POINTS
C       18 JULY 2004 RAY L. CROSS
C
  140 FORMAT (2X,'THE NEAR-FIELD ELECTRIC FIELD INTENSITY AT THE OBSERVA
     1TION POINT LIST X,Y,Z FOR Ex Ey Ez IS:'//)
  143 FORMAT ('POINT_X_Y_Z ',E11.5,' ',E11.5,' ',E11.5,4X,'EX= ',E11.5,
     1' +J',E11.5,2X,'EY= ',E11.5,' +J',E11.5,2X,'EZ= ',E11.5,
     2' +J',E11.5)
C
      END
C
C
      SUBROUTINE BLNK (A)
      CHARACTER*1 A(80)
      CHARACTER*1 BLANK
C       CHANGED BLANK TO EXPLICIT ASSIGNMENT RATHER THAN DATA
C        TO BE COMPATIBLE WITH MORE COMPILERS 6 JAN 1998
      BLANK=' '
      K = 0
C
      DO 1 I=1,80
      J = I-K
      A(J) = A(I)
    1 IF (A(I).EQ.BLANK) K=K+1
```

```
C
      IF (K.EQ.0) RETURN
      A(81-K) = BLANK
      RETURN
      END
      SUBROUTINE CBES (Z,B01)
      REAL*8 PI, ERROR, Y, FACTOR
      COMPLEX*16 ARG,CC,CS,EX
      COMPLEX*16 B01,Z,TERMJ,TERMN,MZ24,JN(2)
      DATA PI/3.141592653589793/
C
C        *** V3.2D FOLLOWING LINE FIXED FOR DOUBLE COMPLEX
      IF (CDABS(Z).GE.12.0) GO TO 4
      FACTOR = 0.0
      TERMN = (0.,0.)
      MZ24 = -0.25*Z*Z
      TERMJ = (1.0,0.0)
C
      DO 3 NP=1,2
      N = NP-1
      JN(NP) = TERMJ
      M = 0
    1 M = M+1
      TERMJ = TERMJ*MZ24/FLOAT(M*(N+M))
      JN(NP) = JN(NP)+TERMJ
      IF (NP.NE.1) GO TO 2
      FACTOR = FACTOR+1.0/FLOAT(M)
      TERMN = TERMN+TERMJ*FACTOR
C
C        *** V3.2D FOLLOWING LINE FIXED FOR DOUBLE COMPLEX
    2 ERROR = CDABS(TERMJ)
      IF (ERROR.GT.1.0E-10) GO TO 1
    3 TERMJ = 0.5*Z
C
      B01 = JN(1)/JN(2)
      RETURN
C
C        *** V3.2D FOLLOWING LINE FIXED FOR DOUBLE COMPLEX
    4 Y = DIMAG(Z)
      IF (ABS(Y).GT.20.) GO TO 5
      ARG = (0.0,1.0)*Z
C
C        *** V3.2D FOLLOWING LINE FIXED FOR DOUBLE COMPLEX
      EX = CDEXP(ARG)
      CC = EX+1.0/EX
      CS = (.0,-1.)*(EX-1./EX)
      B01 = (CS+CC)/(CS-CC)
      RETURN
    5 B01 = (.0,-1.)
      IF (Y.LT.0.) B01 = (.0,1.)
      RETURN
      END
      SUBROUTINE DSHELL (AM,BM,DK,CGDS,SGDS,EP2,EP,ETA,GAM,P11,P12)
      REAL*8 PI, AM, BM, DK
      COMPLEX*16 CGDS,SGDS,EP2,EP,ETA,GAM,P11,P12,GD,CST
      DATA PI/3.141592653589793/
      GD = GAM*DK
C
C        *** V3.2D FOLLOWING LINE FIXED FOR DOUBLE
      CST = (EP2-EP)*ETA*DLOG(BM/AM)/(4.0*PI*EP2*SGDS*SGDS)
      P11 = -CST*(GD+SGDS*CGDS)
      P12 = CST*(GD*CGDS+SGDS)
      RETURN
      END
      SUBROUTINE EQUAL (N)
      CHARACTER*1 A, EQULS
      COMMON /A/ A(80)
      DATA EQULS/'='/
      K = N
C
      DO 1 I=K,80
      N = I+1
      IF (A(I).EQ.EQULS) GO TO 2
    1 CONTINUE
C
      N = 1
    2 RETURN
      END
      SUBROUTINE EXPJ (V1,V2,W12)
      REAL*8 V, W, AB, T3, T4, T5, CF, T6, T7, T8, T9
      REAL*8 D, X, Y, YA, TH, EX, E, XI, YS, T10
      COMPLEX*16 EC,E15,S,T,UC,VC,V1,V2,W12,Z
      DIMENSION V(21), W(21), D(16), E(16)
      DATA V/0.22284667E00,0.11889321E01,0.29927363E01,0.57751436E01,0.9
     18374674E01,0.15982874E02,0.93307812E-01,0.49269174E00,0.12155954E0
     21,0.22699495E01,0.36676227E01,0.54253366E01,0.75659162E01,0.101202
     328E02,0.13130282E02,0.16654408E02,0.20776479E02,0.25623894E02,0.31
     4407519E02,0.38530683E02,0.48026086E02/
      DATA W/0.45896460E00,0.41700083E00,0.11337338E00,0.10399197E-01,0.
     126101720E-03,0.89854791E-06,0.21823487E00,0.34221017E00,0.26302758
     2E00,0.12642582E00,0.40206865E-01,0.85638778E-02,0.12124361E-02,0.1
     31167440E-03,0.64599267E-05,0.22263169E-06,0.42274304E-08,0.3921897
     43E-10,0.14565152E-12,0.14830270E-15,0.16005949E-19/
      DATA D/0.22495842E02,0.74411568E02,-0.41431576E03,-0.78754339E02,0
     1.11254744E02,0.16021761E03,-0.23862195E03,-0.50094687E03,-0.684878
     254E02,0.12254778E02,-0.10161976E02,-0.47219591E01,0.79729681E01,-0
     3.21069574E02,0.22046490E01,0.89728244E01/
      DATA E/0.21103107E02,-0.37959787E03,-0.97489220E02,0.12900672E03,0
     1.17949226E02,-0.12910931E03,-0.55705574E03,0.13524801E02,0.1469672
     221E03,0.17949528E02,-0.32981014E00,0.31028836E02,0.81657657E01,0.22
     3236961E02,0.39124892E02,0.81636799E01/
      Z = V1
C
      DO 12 JIM=1,2
      X = DBLE(Z)
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE
      Y = DIMAG(Z)
      E15 = (0.0,0.0)
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      AB = CDABS(Z)
      IF (AB.EQ.0.) GO TO 11
      IF (X.GE.0..AND.AB.GT.10.) GO TO 10
      YA = ABS(Y)
      IF (X.LE.0..AND.YA.GT.10.) GO TO 10
      IF (YA-X.GE.17.5.OR.YA.GE.6.5.OR.X+YA.GE.5.5.OR.X.GE.3.) GO TO 2
      IF (X.LE.-9.) GO TO 6
      IF (YA-X.GE.2.5) GO TO 7
      IF (X+YA.GE.1.5) GO TO 3
      N = 6.+3.*AB
      E15 = 1./(N-1.)-Z/N**2
    1 N = N-1
      E15 = 1./(N-1.)-Z*E15/N
      IF (N.GE.3) GO TO 1
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE
      E15 = Z*E15-CMPLX(0.577216+DLOG(AB),ATAN2(Y,X))
      GO TO 11
    2 J1 = 1
      J2 = 6
      GO TO 4
    3 J1 = 7
      J2 = 21
    4 S = (.0,.0)
      YS = Y*Y
C
      DO 5 I=J1,J2
      XI = V(I)+X
      CF = W(I)/(XI*XI+YS)
    5 S = S+CMPLX(XI*CF,-YA*CF)
C
      GO TO 9
    6 T3 = X*X-Y*Y
      T4 = 2.*X*YA
      T5 = X*T3-YA*T4
      T6 = X*T4+YA*T3
      UC = CMPLX(D(11)+D(12)*X+D(13)*T3+T5-E(12)*YA-E(13)*T4,E(11)+E(12)
     1*X+E(13)*T3+T6+D(12)*YA+D(13)*T4)
      VC = CMPLX(D(14)+D(15)*X+D(16)*T3+T5-E(15)*YA-E(16)*T4,E(14)+E(15)
     1*X+E(16)*T3+T6+D(15)*YA+D(16)*T4)
      GO TO 8
    7 T3 = X*X-Y*Y
      T4 = 2.*X*YA
      T5 = X*T3-YA*T4
      T6 = X*T4+YA*T3
      T7 = X*T5-YA*T6
      T8 = X*T6+YA*T5
      T9 = X*T7-YA*T8
      T10 = X*T8+YA*T7
      UC = CMPLX(D(1)+D(2)*X+D(3)*T3+D(4)*T5+D(5)*T7+T9-(E(2)*YA+E(3)*T4
     1+E(4)*T6+E(5)*T8),E(1)+E(2)*X+E(3)*T3+E(4)*T5+E(5)*T7+T10+(D(2)*YA
     2+D(3)*T4+D(4)*T6+D(5)*T8))
      VC = CMPLX(D(6)+D(7)*X+D(8)*T3+D(9)*T5+D(10)*T7+T9-(E(7)*YA+E(8)*T
     14+E(9)*T6+E(10)*T8),E(6)+E(7)*X+E(8)*T3+E(9)*T5+E(10)*T7+T10+(D(7)
     2*YA+D(8)*T4+D(9)*T6+D(10)*T8))
    8 EC = UC/VC
      S = EC/CMPLX(X,YA)
    9 EX = EXP(-X)
      T = EX*CMPLX(COS(YA),-SIN(YA))
      E15 = S*T
      IF (Y.LT.0.) E15 = CONJG(E15)
      GO TO 11
   10 E15 = .409319/(Z+.193044)+.421831/(Z+1.02666)+.147126/(Z+2.56788)+
     1.206335E-1/(Z+4.90035)+.107401E-2/(Z+8.18215)+.158654E-4/(Z+12.734
     22)+.317031E-7/(Z+19.3957)
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      E15 = E15*CDEXP(-Z)
   11 IF (JIM.EQ.1) W12 = E15
   12 Z = V2
C
      Z = V2/V1
C
C        **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      TH = DATAN2(DIMAG(Z),DBLE(Z))-DATAN2(DIMAG(V2),DBLE(V2)) +
     1 DATAN2(DIMAG(V1),DBLE(V1))
      AB = ABS(TH)
      IF (AB.LT.1.) TH = .0
      IF (TH.GT.1.) TH = 6.2831853
      IF (TH.LT.-1.) TH = -6.2831853
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      W12 = W12-E15+DCMPLX(DBLE(0.0),DBLE(TH))
      RETURN
      END
      SUBROUTINE GANT1 (IA,IB,INM,IWR,I1,I2,I3,I12,JA,JB,MD,N,ND,NM,AM,C
     1,CJ,CG,CMM,D,EFF,GAM,GG,CGD,SGD,VG,Y11,Z11,ZLD,ZS,IGRD)
      REAL*8 AM, CMM, D, EFF, GC, FI, PRAD, PIN, DISS, GG
      COMPLEX*16 YY,CGEN
      COMPLEX*16 C(1),CJ(1),CGD(1),SGD(1),VG(1),ZLD(1),CG(1)
      COMPLEX*16 Y11,Z11,ZS,GAM
      DIMENSION D(1), IA(1), IB(1), JA(1), JB(1)
      DIMENSION I1(1), I2(1), I3(1), MD(INM,4), ND(1)
      COMMON IWL
C
      DO 3 I=1,N
      CJ(I) = (.0,.0)
      K = JA(I)
C
C
      DO 2 KK=1,2
      KA = IA(K)
      KB = IB(K)
      JJ = K
      FI = 1.
      IF (KB.EQ.I2(I)) GO TO 1
      IF (KB.EQ.I1(I)) FI=-1.
      CJ(I) = CJ(I)+FI*VG(JJ)
      GO TO 2
    1 IF (KA.EQ.I3(I)) FI=-1.
      JJ = K+NM
      CJ(I) = CJ(I)+FI*VG(JJ)
    2 K = JB(I)
```

```
C
C
    3 CONTINUE
C
C
C
C
      DO 4 I=1,N
    4 CG(I) = CJ(I)
C
C
      CALL SQROT (C,CJ,0,I12,N)
      I12 = 2
      Y11 = (.0,.0)
      NNN = N
      IF (IGRD.GT.0) NNN=(N+IWL)/2
C
C
      DO 6 I=1,NNN
      NN = IA(JB(I))
      CGEN=CG(I)
      IF (I.LE.IWL) CGEN=CGEN/2.
      YY=CJ(I)*CONJG(CGEN)
C
C        **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE
      IF (CDABS(YY).LT.1.E-20) GO TO 5
      Z11=(1./YY)*(CDABS(CGEN)**2)
      WRITE (6,8) NN,Z11
    5 Y11 = Y11+YY
    6 CONTINUE
C
C
      IF (IWR.GT.0) WRITE (6,7)
      CALL RITE (IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,CJ,CG,IGRD)
      GG = DBLE(Y11)
      IF (IGRD.GT.0) GG=2.*DBLE(Y11)
      PIN = GG
      CALL GDISS (AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS)
C     FOLLOWING LINE MODIFIED TO FIX EFFICIENCY BUG - JAN 1998
      PRAD = PIN-ABS(DISS)
      EFF = 100.*PRAD/PIN
      RETURN
C
C
    7 FORMAT (50X,'ANTENNA BRANCH CURRENTS')
C
C    V3.2D FORMAT MODIFIED TO PERMIT LARGER NODE NUMBERS RAY L CROSS 18 JULY 2004
C
    8 FORMAT (10X,'THE INPUT IMPEDANCE AT NODE ',I6,' IS',F15.7,
     1' + J ',F15.7//)
C
      END
      SUBROUTINE GDISS (AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS)
      REAL*8 AM, CMM, D, DISS, PI, FA, FB, CAD, CBD, EAD, DK, DEN, SAD
      REAL*8 SBD, RH, BETA, ALPH
      COMPLEX*16 CG(1),SGD(1),ZLD(1),CJA,CJB,GAM,ZS
      DIMENSION D(1)
      DATA PI/3.141592653589793/
      DISS = .0
      IF (CMM.LE.0.) GO TO 2
C
C          **** V3.2D FIXED FOLLOWING 3 LINES FOR DOUBLE COMPLEX
      ALPH = DBLE(GAM)
      BETA = DIMAG(GAM)
      RH = DBLE(ZS)/(4.*PI*AM)
C
      DO 1 K=1,NM
      DK = D(K)
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      DEN = CDABS(SGD(K))**2
      EAD = EXP(ALPH*DK)
      CAD = (EAD+1.0/EAD)/2.0
      CBD = COS(BETA*DK)
      SAD = DK
      IF (ALPH.NE.0.) SAD=(EAD-1./EAD)/(2.*ALPH)
      SBD = DK
      IF (BETA.NE.0.) SBD=SIN(BETA*DK)/BETA
      FA = RH*(SAD*CAD-SBD*CBD)/DEN
      FB = 2.*RH*(CAD*SBD-SAD*CBD)/DEN
      CJA = CG(K)
      L = K+NM
      CJB = CG(L)
C
C     FOLLOWING TWO LINES REPLACED TO CORRECT DISSIPATION BUG - JAN 1998
C    1 DISS = DISS+FA*(CABS(CJA)**2+CABS(CJB)**2)+FB*(REAL(CJA)*REAL(CJB)
C     1+AIMAG(CJA)*AIMAG(CJB))
C
C
C          **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE
    1 DISS = DISS+DABS(FA*(CDABS(CJA)**2+CDABS(CJB)**2))+
     1DABS(FB*(DBLE(CJA)*DBLE(CJB)+DIMAG(CJA)*DIMAG(CJB)))
C
C
    2 DO 3 J=1,NM
      K = J+NM
C
C          **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
    3 DISS = DISS+DBLE(ZLD(J))*(CDABS(CG(J))**2)+DBLE(ZLD(K))*
     1(CDABS(CG(K))**2)
C
      RETURN
      END
      SUBROUTINE GFF (XA,YA,ZA,XB,YB,ZB,D,CGD,SGD,CTH,STH,CPH,SPH,GAM,ET
     1A,ET1,ET2,EP1,EP2,IGRD,ERR)
      REAL*8 XA,YA,ZA,XB,YB,ZB,D,CTH,STH,CPH,SPH,ET
      REAL*8 CA, CB, FA, G, FB, CG, P, GK, T, FP, XAB, YAB, ZAB
      COMPLEX*16 ERR,RV,RH,RR,EX,EY,EZ,EE
      COMPLEX*16 ET1,ET2,EP1,EP2,GAM,ETA
      COMPLEX*16 GD,CGD,SGD,EGD
      COMPLEX*16 EGFA,EGFB,EGGD,ESA,ESB
      COMPLEX*16 CST
      FP = 12.56637
      XAB = XB-XA
      YAB = YB-YA
      ZAB = ZB-ZA
      CA = XAB/D
      CB = YAB/D
      CG = ZAB/D
      G = (CA*CPH+CB*SPH)*STH+CG*CTH
      GK = 1.-G*G
      ET1 = (.0,.0)
      ET2 = (.0,.0)
      EP1 = (.0,.0)
      EP2 = (.0,.0)
      IF (GK.LT..001) GO TO 3
      FA = (XA*CPH+YA*SPH)*STH+ZA*CTH
      FB = (XB*CPH+YB*SPH)*STH+ZB*CTH
C
C        **** V3.2D FIXED FOLLOWING 3 LINES FOR DOUBLE COMPLEX
      EGFA = CDEXP(GAM*FA)
      EGFB = CDEXP(GAM*FB)
      EGGD = CDEXP(GAM*G*D)
      CST = ETA/(GK*SGD*FP)
      ESA = CST*EGFA*(EGGD-G*SGD-CGD)
      ESB = CST*EGFB*(1./EGGD+G*SGD-CGD)
      IF (IGRD.LE.0) GO TO 2
      RV = (-1.,0.)
      RH = (-1.,0.)
      IF (IGRD.EQ.1) GO TO 1
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR = CDSQRT(ERR-STH*STH)
      RV = -(ERR*CTH-RR)/(ERR*CTH+RR)
      RH = (CTH-RR)/(CTH+RR)
    1 EX = CA*ESA
      EY = CB*ESA
      EZ = CG*ESA
      EE = (EX*SPH-EY*CPH)*(RH-RV)
      EX = EX*RV+EE*SPH
      EY = EY*RV-EE*CPH
      EZ = -EZ*RV
      ESA= -EX*CA-EY*CB+EZ*CG
      EX = CA*ESB
      EY = CB*ESB
      EZ = CG*ESB
      EE = (EX*SPH-EY*CPH)*(RH-RV)
      EX = EX*RV+EE*SPH
      EY = EY*RV-EE*CPH
      EZ = -EZ*RV
      ESB=-EX*CA-EY*CB+EZ*CG
    2 T = (CA*CPH+CB*SPH)*CTH-CG*STH
      P = -CA*SPH+CB*CPH
      ET1 = T*ESA
      ET2 = T*ESB
      EP1 = P*ESA
      EP2 = P*ESB
    3 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE GFFLD (IA,IB,INC,INM,IWR,I1,I2,I3,I12,MD,N,ND,NM,AM,ACS
     1P,ACST,C,CGD,CG,CJ,CMM,D,ECSP,ECST,EP,ET,EPP,ETT,EPPS,EPTS,ETPS,ET
     2TS,GG,GPP,GTT,PH,SGD,SCSP,SCST,SPPM,SPTM,STPM,STTM,TH,X,Y,Z,ZLD,ZS
     3,ETA,GAM,ERR,IGRD)
      REAL*8 AM,ACSP,ACST,CMM,D,ECSP,ECST
      REAL*8 GG,GPP,GTT,PH,SCSP,SCST,SPPM,SPTM,STPM
      REAL*8 STTM,TH,X,Y,Z
      REAL*8 PI, TP, FI, GGG, CPH, CTH, APP, PIN, ATT, PHR, TIN
      REAL*8 SPH, THR, STH, PDIS, TDIS
      COMPLEX*16 ERR
      COMPLEX*16 CJI,ET1,ET2,EP1,EP2,EPPS,ETTS,EPTS,ETPS,ZS,VP,VT
      COMPLEX*16 C(1),CJ(1),EP(1),ET(1),EPP(1),ETT(1),ZLD(1)
      COMPLEX*16 ETA,GAM,CGD(1),SGD(1),CG(1)
      DIMENSION IA(1), IB(1), I1(1), I2(1), I3(1), ND(1), MD(INM,4)
      DIMENSION D(1), X(1), Y(1), Z(1)
      DATA PI,TP/3.141592653589793,6.283185307179586/
      CJI = -4.*PI/(ETA*GAM)
      GGG = DBLE(1./ETA)
      THR = .0174533*TH
      CTH = COS(THR)
      STH = SIN(THR)
      PHR = .0174533*PH
      CPH = COS(PHR)
      SPH = SIN(PHR)
C
      DO 1 I=1,N
      ETT(I) = (.0,.0)
    1 EPP(I) = (.0,.0)
C
C
      DO 3 K=1,NM
      KA = IA(K)
      KB = IB(K)
      NGRD = IGRD
      IF (K.LE.NM/2) IGRD=-1
      CALL GFF (X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),D(K),CGD(K),SGD(K),C
     1TH,STH,CPH,SPH,GAM,ETA,ET1,ET2,EP1,EP2,IGRD,ERR)
      IGRD = NGRD
      NDK = ND(K)
C
      DO 3 II=1,NDK
      I = MD(K,II)
      FI = 1.
      IF (KB.EQ.I2(I)) GO TO 2
      IF (KB.EQ.I1(I)) FI=-1.
      EPP(I) = EPP(I)+FI*EP1
      ETT(I) = ETT(I)+FI*ET1
      GO TO 3
    2 IF (KA.EQ.I3(I)) FI=-1.
      EPP(I) = EPP(I)+FI*EP2
      ETT(I) = ETT(I)+FI*ET2
    3 CONTINUE
C
      EPPS = (.0,.0)
      ETTS = (.0,.0)
      IF (INC.EQ.0) GO TO 8
      IF (INC.EQ.2) GO TO 6
C
      DO 4 I=1,N
      ET(I) = ETT(I)*CJI
    4 EP(I) = EPP(I)*CJI
C
      CALL SQROT (C,EP,0,I12,N)
      I12 = 2
      CALL SQROT (C,ET,0,I12,N)
      IF (IWR.GT.0) WRITE (6,10) PH,TH
      IF (IWR.GT.0) WRITE (6,11)
      CALL RITE (IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,EP,CG,IGRD)
      CALL GDISS (AM,CG,CMM,D,PDIS,GAM,NM,SGD,ZLD,ZS)
      IF (IWR.GT.0) WRITE (6,12)
      CALL RITE (IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,ET,CG,IGRD)
      CALL GDISS (AM,CG,CMM,D,TDIS,GAM,NM,SGD,ZLD,ZS)
      ACSP = PDIS/GGG
      ACST = TDIS/GGG
      PIN = .0
      TIN = .0
C
      DO 5 I=1,N
      VP = CJI*EPP(I)
      VT = CJI*ETT(I)
      PIN = PIN+DBLE(VP*CONJG(EP(I)))
    5 TIN = TIN+DBLE(VT*CONJG(ET(I)))
C
      ECSP = PIN/GGG
      ECST = TIN/GGG
      SCSP = ECSP-ACSP
      SCST = ECST-ACST
    6 EPTS = (.0,.0)
      ETPS = (.0,.0)
C
      DO 7 I=1,N
      EPPS = EPPS+EP(I)*EPP(I)
      EPTS = EPTS+EP(I)*ETT(I)
      ETTS = ETTS+ET(I)*ETT(I)
    7 ETPS = ETPS+ET(I)*EPP(I)
C
C
C        **** V3.2D FIXED FOLLOWING 4 LINES FOR DOUBLE COMPLEX
      SPPM = 2.*TP*(CDABS(EPPS)**2)
      SPTM = 2.*TP*(CDABS(EPTS)**2)
      STPM = 2.*TP*(CDABS(ETPS)**2)
      STTM = 2.*TP*(CDABS(ETTS)**2)
      RETURN
C
    8 DO 9 I=1,N
      ETTS = ETTS+CJ(I)*ETT(I)
    9 EPPS = EPPS+CJ(I)*EPP(I)
C
C
C        **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
```

```
      APP = CDABS(EPPS)
      ATT = CDABS(ETTS)
      GPP = 4.*PI*APP*APP*GGG/GG
      GTT = 4.*PI*ATT*ATT*GGG/GG
      RETURN
C
   10 FORMAT (10X,'BRANCH CURRENTS ASSOCIATED WITH PLANE-WAVE SCATTERING
     1 FOR THE INCIDENT ANGLES, PHI=',F5.1,' AND THETA=',F5.1//)
   11 FORMAT (44X,'CURRENTS INDUCED BY THE PHI POLARIZED WAVE')
   12 FORMAT (44X,'CURRENTS INDUCED BY THE THETA POLARIZED WAVE')
      END
      SUBROUTINE GGS (XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,DS,CGDS,SGD
     1S,DT,SGDT,INT,ETA,GAM,P11,P12,P21,P22,ERR,IGRD)
      REAL*8 XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,DS
      REAL*8 DT, DK, ZERO, ONE
      REAL*8 FP, P1, C, R1, S1, R2, CA, T1, CB, CC, DR1
      REAL*8 CAD, DR2, CBD, RG1, FAC, XXZ, CG
      REAL*8 DG, RG2, DDD, YYZ, CGD, ZZZ, D, CTH1, XG1, CTH2
      REAL*8 XG2, YG1, YG2, ZG1, ZG2, CAP, T
      REAL*8 CBP, CAS, CBS, XP1, RG, TT1, CGP, YP1, TT2, ZP1
      REAL*8 CPH, CGS, SZ1, SZ2, AMS, RS
      REAL*8 ZZ1, SS, ZZ2, SGN, DELT, SPH, SZ, DSZ, XZ, YZ
      REAL*8 SSTH1, ZZ, SSTH2, SSPH
      COMPLEX*16 EX1,EY1,EX2,EY2,EZ1,EZ2
      COMPLEX*16 P11,P12,P21,P22,EJA,EJB,EJ1,EJ2,ETA,GAM,C1,C2,CST
      COMPLEX*16 EGD,CGDS,SGDS,SGDT,ER1,ER2,ET1,ET2
      COMPLEX*16 ERR
      COMPLEX*16 EE,EXX,EYY
      COMPLEX*16 PP,PX,PY,PZ
      COMPLEX*16 RR1,RR2,RR3,RR4,RH1,RV1,RH2,RV2,RH3,RV3,RH4,RV4
      DATA FP/12.56637/
      DATA ZERO/0.0000/
      DATA ONE/1.0000/
      CA = (X2-X1)/DT
      CB = (Y2-Y1)/DT
      CG = (Z2-Z1)/DT
      CAS = (XB-XA)/DS
      CBS = (YB-YA)/DS
      CGS = (ZB-ZA)/DS
      CC = CA*CAS+CB*CBS+CG*CGS
      IF ((CG.LE..003).AND.(CGS.LE..003).AND.(IGRD.GT.0)) GO TO 1
      IF (ABS(CC).GT..997) GO TO 6
    1 SZ = (X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      IF (INT.LE.0) GO TO 7
      INS = 2*(INT/2)
      IF (INS.LT.2) INS = 2
      IP = INS+1
      DELT = DT/INS
      T = .0
      DSZ = CC*DELT
      P11 = (.0,.0)
      P12 = (.0,.0)
      P21 = (.0,.0)
      P22 = (.0,.0)
      AMS = AM*AM
      SGN = -1.
```

```
C
C
      DO 5 IN=1,IP
      ZZ1 = SZ
      ZZ2 = SZ-DS
      XXZ = X1+T*CA-XA-SZ*CAS
      YYZ = Y1+T*CB-YA-SZ*CBS
      ZZZ = Z1+T*CG-ZA-SZ*CGS
      RS = XXZ**2+YYZ**2+ZZZ**2
      R1 = SQRT(RS+ZZ1**2)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EJA = CDEXP(-GAM*R1)
      EJ1 = EJA/R1
      R2 = SQRT(RS+ZZ2**2)
C
C         **** V3.2D FIXED FOLLOWING 3 LINES FOR DOUBLE COMPLEX
      EJB = CDEXP(-GAM*R2)
      EJ2 = EJB/R2
      ER1 = EJA*SGDS+ZZ1*EJ1*CGDS-ZZ2*EJ2
      ER2 = -EJB*SGDS+ZZ2*EJ2*CGDS-ZZ1*EJ1
      FAC = .0
      IF (RS.GT.AMS) FAC = (CA*XXZ+CB*YYZ+CG*ZZZ)/RS
      ET1 = CC*(EJ2-EJ1*CGDS)+FAC*ER1
      ET2 = CC*(EJ1-EJ2*CGDS)+FAC*ER2
      IF (IGRD.LT.0) GO TO 4
      RV1 = (-1.,0.)
      RH1 = (-1.,0.)
      RV2 = (-1.,0.)
      RH2 = (-1.,0.)
      IF (IGRD.EQ.1) GO TO 2
      XG1 = X1+T*CA-XA
      YG1 = Y1+T*CB-YA
      ZG1 = Z1+T*CG-ZA
      XG2 = X1+T*CA-XB
      YG2 = Y1+T*CB-YB
      ZG2 = Z1+T*CG-ZB
      RG1 = SQRT(XG1*XG1+YG1*YG1)
      RG2 = SQRT(XG2*XG2+YG2*YG2)
      TT1 = ATAN(RG1/ZG1)
      TT2 = ATAN(RG2/ZG2)
      CTH1 = COS(TT1)
      SSTH1 = SIN(TT1)*SIN(TT1)
      CTH2 = COS(TT2)
      SSTH2 = SIN(TT2)*SIN(TT2)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SSTH1)
      RH1 = (CTH1-RR1)/(CTH1+RR1)
      RV1 = -(ERR*CTH1-RR1)/(ERR*CTH1+RR1)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR2 = CDSQRT(ERR-SSTH2)
      RH2 = (CTH2-RR2)/(CTH2+RR2)
      RV2 = -(ERR*CTH2-RR2)/(ERR*CTH2+RR2)
    2 RG = SQRT((XB-XA)*(XB-XA)+(YB-YA)*(YB-YA))
```

```
      CPH = 0
      SPH = 0
      IF (RG.LT.1.E-32) GO TO 3
      CPH = (XB-XA)/RG
      SPH = (YB-YA)/RG
    3 EXX = ET1*CAS
      EYY = ET1*CBS
      EE = (EXX*SPH-EYY*CPH)*(RH1-RV1)
      EX1 = EXX*RV1+EE*SPH
      EY1 = EYY*RV1-EE*CPH
      EZ1 = -ET1*RV1*CGS
      ET1=-EX1*CAS-EY1*CBS+EZ1*CGS
      EXX = ET2*CAS
      EYY = ET2*CBS
      EE = (EXX*SPH-EYY*CPH)*(RH2-RV2)
      EX2 = EXX*RV2+EE*SPH
      EY2 = EYY*RV2-EE*CPH
      EZ2 = -ET2*CGS*RV2
      ET2=-EX2*CAS-EY2*CBS+EZ2*CGS
    4 C = 3.+SGN
      IF (IN.EQ.1.OR.IN.EQ.IP) C=1.
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EGD = CDEXP(GAM*(DT-T))
      C1 = C*(EGD-1./EGD)/2.
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EGD = CDEXP(GAM*T)
      C2 = C*(EGD-1./EGD)/2.
      P11 = P11+ET1*C1
      P12 = P12+ET1*C2
      P21 = P21+ET2*C1
      P22 = P22+ET2*C2
      T = T+DELT
      SZ = SZ+DSZ
    5 SGN = -SGN
C
C
      CST = -ETA*DELT/(3.*FP*SGDS*SGDT)
      P11 = CST*P11
      P12 = CST*P12
      P21 = CST*P21
      P22 = CST*P22
      RETURN
    6 SZ1 = (X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      DR1 = SQRT((X1-XA-SZ1*CAS)**2+(Y1-YA-SZ1*CBS)**2+(Z1-ZA-SZ1*CGS)**
     12)
      SZ2 = SZ1+DT*CC
      DR2 = SQRT((X2-XA-SZ2*CAS)**2+(Y2-YA-SZ2*CBS)**2+(Z2-ZA-SZ2*CGS)**
     12)
      DDD = (DR1+DR2)/2.
      IF (DDD.GT.20.*AM.AND.INT.GT.0) GO TO 1
      IF (DDD.LT.AM) DDD = AM
      CALL GGMM (ZERO,DS,SZ1,SZ2,DDD,CGDS,SGDS,SGDT,ONE,ETA,GAM,P11,P12,
     1P21,P22)
      IF (IGRD.LE.1) RETURN
      IF (IGRD.GT.1) GO TO 8
C
    7 SS = SQRT(1.-CC*CC)
      CAD = (CGS*CB-CBS*CG)/SS
      CBD = (CAS*CG-CGS*CA)/SS
      CGD = (CBS*CA-CAS*CB)/SS
      DK = (X1-XA)*CAD+(Y1-YA)*CBD+(Z1-ZA)*CGD
      DK = ABS(DK)
      IF (DK.LT.AM) DK = AM
      XZ = XA+SZ*CAS
      YZ = YA+SZ*CBS
      ZZ = ZA+SZ*CGS
      XP1 = X1-DK*CAD
      YP1 = Y1-DK*CBD
      ZP1 = Z1-DK*CGD
      CAP = CBS*CGD-CGS*CBD
      CBP = CGS*CAD-CAS*CGD
      CGP = CAS*CBD-CBS*CAD
      P1 = CAP*(XP1-XZ)+CBP*(YP1-YZ)+CGP*(ZP1-ZZ)
      T1 = P1/SS
      S1 = T1*CC-SZ
      CALL GGMM (S1,S1+DS,T1,T1+DT,DK,CGDS,SGDS,SGDT,CC,ETA,GAM,P11,P12,
     1P21,P22)
      RETURN
C
    8 AMS = AM*AM
      RG = (X1-XA)*(X1-XA)+(Y1-YA)*(Y1-YA)
      IF (RG.LT.AMS) RG = AMS
      DG = SQRT((Z1-ZA)*(Z1-ZA)+RG)
      CPH = ABS(Z1-ZA)/DG
      SSPH=RG/(DG*DG)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SSPH)
      RV1 = -(ERR*CPH-RR1)/(ERR*CPH+RR1)
      P11=-P11*RV1
      RG = (X1-XB)*(X1-XB)+(Y1-YB)*(Y1-YB)
      IF (RG.LT.AMS) RG = AMS
      DG = SQRT((Z1-ZB)*(Z1-ZB)+RG)
      CPH = ABS(Z1-ZB)/DG
      SSPH=RG/(DG*DG)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SSPH)
      RV1 = -(ERR*CPH-RR1)/(ERR*CPH+RR1)
      P12=-P12*RV1
      RG = (X2-XA)*(X2-XA)+(Y2-YA)*(Y2-YA)
      IF (RG.LT.AMS) RG = AMS
      DG = SQRT((Z2-ZA)*(Z2-ZA)+RG)
      CPH = ABS(Z2-ZA)/DG
      SSPH=RG/(DG*DG)
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SSPH)
      RV1 = -(ERR*CPH-RR1)/(ERR*CPH+RR1)
      P21=-P21*RV1
      RG = (X2-XB)*(X2-XB)+(Y2-YB)*(Y2-YB)
      IF (RG.LT.AMS) RG = AMS
      DG = SQRT((Z2-ZB)*(Z2-ZB)+RG)
      CPH = ABS(Z2-ZB)/DG
      SSPH=RG/(DG*DG)
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SSPH)
      RV1 = -(ERR*CPH-RR1)/(ERR*CPH+RR1)
      P22=-P22*RV1
      RETURN
      END
      SUBROUTINE GGMM (S1,S2,T1,T2,D,CGDS,SGD1,SGD2,CPSI,ETA,GAM,P11,P12
     1,P21,P22)
      REAL*8 S1,S2,T1,T2,D, CPSI
      REAL*8 PI, B, C, V1, W1
      REAL*8 FI, FK, R
      REAL*8 FL, TA, RR1, V, TB, RR2, W, SI, ZC, TJ, DSQ, ZIJ, XX
      DOUBLE PRECISION R1,R2,DPQ,SIS,TS1,TS2,ST1,ST2,CD,BD,CPSS,SK,TL1,T
     1L2,TD1,TD2,SDI,DPSI,DD,ZD
      COMPLEX*16 CGDS,SGDS,SGDT,SGD1,SGD2,ETA,GAM,P11,P12,P21,P22
      COMPLEX*16 CST,EB,EC,EK,EL,EKL,EGZI,ES1,ES2,ET1,ET2,EXPA,EXPB
      COMPLEX*16 E(2,2),F(2,2)
      COMPLEX*16 EGZ(2,2),GM(2),GP(2)
      DATA PI/3.141592653589793/
      DSQ = D*D
      SGDS = SGD1
      IF (S2.LT.S1) SGDS = -SGD1
      SGDT = SGD2
      IF (T2.LT.T1) SGDT = -SGD2
      IF (ABS(CPSI).GT..997) GO TO 5
C
C        **** V3.2D FIXED FOLLOWING 4 LINES FOR DOUBLE COMPLEX
      ES1 = CDEXP(GAM*S1)
      ES2 = CDEXP(GAM*S2)
      ET1 = CDEXP(GAM*T1)
      ET2 = CDEXP(GAM*T2)
      DD = D
      DPSI = CPSI
      TD1 = T1
      TD2 = T2
      CPSS = DPSI*DPSI
      CD = DD/DSQRT(1.D0-CPSS)
      C = CD
      BD = CD*DPSI
      B = BD
C
C        **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      EB = CDEXP(GAM*DCMPLX(DBLE(0.0),B))
      EC = CDEXP(GAM*DCMPLX(DBLE(0.0),C))
C
      DO 1 K=1,2
C
      DO 1 L=1,2
    1 E(K,L) = (.0,.0)
C
```

```
      TS1 = TD1*TD1
      TS2 = TD2*TD2
      DPQ = DD*DD
      SI = S1
C
      DO 4 I=1,2
      FI = (-1)**I
      SDI = SI
      SIS = SDI*SDI
      ST1 = 2.*SDI*TD1*DPSI
      ST2 = 2.*SDI*TD2*DPSI
      R1 = DSQRT(DPQ+SIS+TS1-ST1)
      R2 = DSQRT(DPQ+SIS+TS2-ST2)
      EK = EB
C
      DO 3 K=1,2
      FK = (-1)**K
      SK = FK*SDI
      EL = EC
C
      DO 2 L=1,2
      FL = (-1)**L
      EKL = EK*EL
      XX = FK*BD+FL*CD
      TL1 = FL*TD1
      TL2 = FL*TD2
      RR1 = R1+SK+TL1
      RR2 = R2+SK+TL2
      CALL EXPJ (GAM*CMPLX(RR1,-XX),GAM*CMPLX(RR2,-XX),EXPA)
      CALL EXPJ (GAM*CMPLX(RR1,XX),GAM*CMPLX(RR2,XX),EXPB)
      E(K,L) = E(K,L)+FI*(EXPA*EKL+EXPB/EKL)
    2 EL = 1./EC
C
    3 EK = 1./EB
C
      ZD = SDI*DPSI
      ZC = ZD
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EGZI = CDEXP(GAM*ZC)
      RR1 = R1+ZD-TD1
      RR2 = R2+ZD-TD2
      CALL EXPJ (GAM*RR1,GAM*RR2,EXPB)
      RR1 = R1-ZD+TD1
      RR2 = R2-ZD+TD2
      CALL EXPJ (GAM*RR1,GAM*RR2,EXPA)
      F(I,1) = 2.*SGDS*EXPA/EGZI
      F(I,2) = 2.*SGDS*EXPB*EGZI
    4 SI = S2
C
      CST = ETA/(16.*PI*SGDS*SGDT)
      P11 = CST*((F(1,1)+E(2,2)*ES2-E(1,2)/ES2)*ET2+(-F(1,2)-E(2,1)*ES2+
     1E(1,1)/ES2)/ET2)
      P12 = CST*((-F(1,1)-E(2,2)*ES2+E(1,2)/ES2)*ET1+(F(1,2)+E(2,1)*ES2-
     1E(1,1)/ES2)/ET1)
      P21 = CST*((-F(2,1)-E(2,2)*ES1+E(1,2)/ES1)*ET2+(F(2,2)+E(2,1)*ES1-
     1E(1,1)/ES1)/ET2)
      P22 = CST*((F(2,1)+E(2,2)*ES1-E(1,2)/ES1)*ET1+(-F(2,2)-E(2,1)*ES1+
     1E(1,1)/ES1)/ET1)
      RETURN
    5 IF (CPSI.LT.0.) GO TO 6
      TA = T1
      TB = T2
      GO TO 7
    6 TA = -T1
      TB = -T2
      SGDT = -SGDT
    7 SI = S1
C
      DO 9 I=1,2
      TJ = TA
C
      DO 8 J=1,2
      ZIJ = TJ-SI
      R = SQRT(DSQ+ZIJ*ZIJ)
      W = R+ZIJ
      IF (ZIJ.LT.0.) W = DSQ/(R-ZIJ)
      V = R-ZIJ
      IF (ZIJ.GT.0.) V = DSQ/(R+ZIJ)
      IF (J.EQ.1) V1 = V
      IF (J.EQ.1) W1 = W
C
C         **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EGZ(I,J) = CDEXP(GAM*ZIJ)
    8 TJ = TB
C
      CALL EXPJ (GAM*V1,GAM*V,GP(I))
      CALL EXPJ (GAM*W1,GAM*W,GM(I))
    9 SI = S2
C
      CST = -ETA/(8.*PI*SGDS*SGDT)
      P11 = CST*(GM(2)*EGZ(2,2)+GP(2)/EGZ(2,2)-CGDS*(GM(1)*EGZ(1,2)+GP(1
     1)/EGZ(1,2)))
      P12 = CST*(-GM(2)*EGZ(2,1)-GP(2)/EGZ(2,1)+CGDS*(GM(1)*EGZ(1,1)+GP(
     11)/EGZ(1,1)))
      P21 = CST*(GM(1)*EGZ(1,2)+GP(1)/EGZ(1,2)-CGDS*(GM(2)*EGZ(2,2)+GP(2
     1)/EGZ(2,2)))
      P22 = CST*(-GM(1)*EGZ(1,1)-GP(1)/EGZ(1,1)+CGDS*(GM(2)*EGZ(2,1)+GP(
     12)/EGZ(2,1)))
      RETURN
      END
      SUBROUTINE GNF (XA,YA,ZA,XB,YB,ZB,X,Y,Z,AM,DS,CGDS,SGDS,ETA,GAM,EX
     11,EY1,EZ1,EX2,EY2,EZ2,IGRD,ERR)
      REAL*8 XA,YA,ZA,XB,YB,ZB,X,Y,Z,AM,DS
      REAL*8 PI, R1, R2, XXZ, YYZ, TH1, TH2, ZZZ, CTH1, CTH2, CAS
      REAL*8 CBS, RG, CPH, CGS, AMS, RS, ZZ1, ZZ2, SPH, SZ
      COMPLEX*16 ERR,RV1,RH1,RV2,RH2,RR1,RR2,EE
      COMPLEX*16 EJA,EJB,EJ1,EJ2,ER1,ER2,ES1,ES2,SGDS,GAM,CST,CGDS,ETA
      COMPLEX*16 EX1,EY1,EZ1,EX2,EY2,EZ2
      DATA PI/3.141592653589793/
      CAS = (XB-XA)/DS
      CBS = (YB-YA)/DS
      CGS = (ZB-ZA)/DS
      SZ = (X-XA)*CAS+(Y-YA)*CBS+(Z-ZA)*CGS
      ZZ1 = SZ
      ZZ2 = SZ-DS
      XXZ = X-XA-SZ*CAS
      YYZ = Y-YA-SZ*CBS
      ZZZ = Z-ZA-SZ*CGS
      RS = XXZ**2+YYZ**2+ZZZ**2
      R1 = SQRT(RS+ZZ1**2)
C
C     **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EJA = CDEXP(-GAM*R1)
      EJ1 = EJA/R1
      R2 = SQRT(RS+ZZ2**2)
C
C     **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EJB = CDEXP(-GAM*R2)
      EJ2 = EJB/R2
      ES1 = EJ2-EJ1*CGDS
      ES2 = EJ1-EJ2*CGDS
      ER1 = (.0,.0)
      ER2 = (.0,.0)
      AMS = AM*AM
      IF (RS.LT.AMS) GO TO 1
      CTH1 = ZZ1/R1
      CTH2 = ZZ2/R2
      ER1 = (EJA*SGDS+EJA*CGDS*CTH1-EJB*CTH2)/RS
      ER2 = (-EJB*SGDS+EJB*CGDS*CTH2-EJA*CTH1)/RS
    1 CST = ETA/(4.*PI*SGDS)
      EX1 = CST*(ES1*CAS+ER1*XXZ)
      EY1 = CST*(ES1*CBS+ER1*YYZ)
      EZ1 = CST*(ES1*CGS+ER1*ZZZ)
      EX2 = CST*(ES2*CAS+ER2*XXZ)
      EY2 = CST*(ES2*CBS+ER2*YYZ)
      EZ2 = CST*(ES2*CGS+ER2*ZZZ)
      IF (IGRD.LE.0) RETURN
      RV1 = (-1.,0.)
      RH1 = (-1.,0.)
      RV2 = (-1.,0.)
      RH2 = (-1.,0.)
      IF (IGRD.EQ.1) GO TO 2
      R1 = SQRT((XA-X)*(XA-X)+(YA-Y)*(YA-Y))
      R2 = SQRT((XB-X)*(XB-X)+(YB-Y)*(YB-Y))
      TH1 = ATAN(R1/(ZA-Z))
      TH2 = ATAN(R2/(ZB-Z))
C
C     **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      RR1 = CDSQRT(ERR-SIN(TH1)*SIN(TH1))
      RR2 = CDSQRT(ERR-SIN(TH2)*SIN(TH2))
      RV1 = -(ERR*COS(TH1)-RR1)/(ERR*COS(TH1)+RR1)
      RH1 = (COS(TH1)-RR1)/(COS(TH1)+RR1)
      RV2 = -(ERR*COS(TH2)-RR2)/(ERR*COS(TH2)+RR2)
      RH2 = (COS(TH2)-RR2)/(COS(TH2)+RR2)
    2 RG = SQRT((XA-XB)*(XA-XB)+(YA-YB)*(YA-YB))
      CPH = 0
      SPH = 0
      IF (RG.LT.1.E-32) GO TO 3
      CPH = (XB-XA)/RG
      SPH = (YB-YA)/RG
    3 EE = (EX1*SPH-EY1*CPH)*(RH1-RV1)
      EX1=-EX1*RV1+EE*SPH
      EY1=-EY1*RV1-EE*CPH
      EZ1 = EZ1*(-RV1)
      EE = (EX2*SPH-EY2*CPH)*(RH2-RV2)
      EX2=-EX2*RV2+EE*SPH
      EY2=-EY2*RV2-EE*CPH
      EZ2 = EZ2*(-RV2)
      RETURN
      END
      SUBROUTINE GNFLD (IA,IB,INM,I1,I2,I3,MD,N,ND,NM,AM,CGD,SGD,ETA,GAM
     1,CJ,D,X,Y,Z,XP,YP,ZP,EX,EY,EZ,IGRD,ERR)
      REAL*8 AM,D,X,Y,Z,XP,YP,ZP
      REAL*8 PI, TP, FI
      COMPLEX*16 EX,EY,EZ,EX1,EY1,EZ1,EX2,EY2,EZ2,ETA,GAM
      COMPLEX*16 ERR
      COMPLEX*16 CJ(1),CGD(1),SGD(1)
      DIMENSION IA(1), IB(1), I1(1), I2(1), I3(1), D(1), X(1), Y(1), Z(1
     1)
      DIMENSION MD(INM,4), ND(1)
      DATA PI,TP/3.141592653589793,6.283185307179586/
      EX = (.0,.0)
      EY = (.0,.0)
      EZ = (.0,.0)
C
      DO 2 K=1,NM
      KA = IA(K)
      KB = IB(K)
      NGRD = IGRD
      IF (K.LE.NM/2) IGRD=-1
      CALL GNF (X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),XP,YP,ZP,AM,D(K),CGD
     1(K),SGD(K),ETA,GAM,EX1,EY1,EZ1,EX2,EY2,EZ2,IGRD,ERR)
      IGRD = NGRD
      NDK = ND(K)
C
      DO 2 II=1,NDK
      I = MD(K,II)
      FI = 1.
      IF (KB.EQ.I2(I)) GO TO 1
      IF (KB.EQ.I1(I)) FI=-1.
      EX = EX+FI*EX1*CJ(I)
      EY = EY+FI*EY1*CJ(I)
      EZ = EZ+FI*EZ1*CJ(I)
      GO TO 2
    1 IF (KA.EQ.I3(I)) FI=-1.
      EX = EX+FI*EX2*CJ(I)
      EY = EY+FI*EY2*CJ(I)
      EZ = EZ+FI*EZ2*CJ(I)
    2 CONTINUE
C
      RETURN
      END
      SUBROUTINE LEFT (N)
      CHARACTER*1 A
      COMMON /A/ A(80)
      CHARACTER*1 PLEFT/'('/
      K = N
C
      DO 1 I=K,80
      N = I+1
      IF (A(I).EQ.PLEFT) GO TO 2
    1 CONTINUE
C
      N = 1
    2 RETURN
      END
      SUBROUTINE LINECK (X,Y)
C
C     THIS SUBROUTINE INSURES ALL GRID CHARACTORS LIE ON THE POLAR GRID
C
      REAL*8 X, Z
      CHARACTER*1 ISYM,LINE
      COMMON /PLOT/ ISYM(14),LINE(130)
      INTEGER Y
      IF (Y.EQ.0) GO TO 3
      K = 0
      IF (X.LT.10.0) GO TO 5
C
C     SET UP AREAS OF "PERIOD" POLAR GRID POINT CHARACTERS
C
      I = INT(X)
      I = IABS(I)
      Z = ABS(X)
      IF ((Z-I).GT.0.5) I=I+1
    1 IF (Z.LT.10.0.OR.Z.GT.111.0) GO TO 2
      LINE(I) = ISYM(2)
      LINE(60) = ISYM(3)
      LINE(62) = ISYM(3)
      K = K+1
      IF (K.EQ.2) GO TO 2
      I = 122-I
      GO TO 1
    2 LINE(61) = ISYM(2)
      IF (Y.NE.0) GO TO 5
C
    3 DO 4 K=11,111
      LINE(K) = ISYM(2)
    4 CONTINUE
C
C
C     FILL IN GRID NUMBER LABELS ON HORIZONTAL AXIS
C
      LINE(11) = ISYM(7)
      LINE(20) = ISYM(10)
      LINE(21) = ISYM(5)
      LINE(22) = ISYM(11)
      LINE(30) = ISYM(9)
      LINE(31) = ISYM(5)
      LINE(32) = ISYM(11)
      LINE(40) = ISYM(8)
      LINE(41) = ISYM(5)
      LINE(42) = ISYM(11)
      LINE(50) = ISYM(7)
      LINE(51) = ISYM(5)
      LINE(52) = ISYM(11)
      LINE(61) = ISYM(1)
      LINE(70) = ISYM(7)
      LINE(71) = ISYM(5)
      LINE(72) = ISYM(11)
      LINE(80) = ISYM(8)
      LINE(81) = ISYM(5)
      LINE(82) = ISYM(11)
      LINE(90) = ISYM(9)
      LINE(91) = ISYM(5)
      LINE(92) = ISYM(11)
      LINE(100) = ISYM(10)
      LINE(101) = ISYM(5)
      LINE(102) = ISYM(11)
      LINE(111) = ISYM(7)
    5 CONTINUE
      RETURN
      END
      SUBROUTINE NUMB (Y)
C
C     THIS SUBROUTINE PUTS DEGREE NUMBERS ON POLAR GRID
C
      CHARACTER*1 ISYM, LINE
      COMMON /PLOT/ ISYM(14),LINE(130)
      INTEGER Y
      IF (Y.NE.37) GO TO 1
      LINE(33) = ISYM(7)
      LINE(34) = ISYM(8)
      LINE(35) = ISYM(6)
      LINE(87) = ISYM(6)
      LINE(88) = ISYM(12)
      LINE(89) = ISYM(6)
    1 IF (Y.NE.21) GO TO 2
      LINE(12) = ISYM(7)
      LINE(13) = ISYM(11)
      LINE(14) = ISYM(6)
      LINE(108) = ISYM(6)
      LINE(109) = ISYM(9)
      LINE(110) = ISYM(6)
    2 IF (Y.NE.0) GO TO 3
      LINE(7) = ISYM(7)
      LINE(8) = ISYM(13)
      LINE(9) = ISYM(6)
      LINE(113) = ISYM(6)
      LINE(114) = ISYM(6)
      LINE(115) = ISYM(6)
    3 IF (Y.NE.-21) GO TO 4
      LINE(12) = ISYM(8)
      LINE(13) = ISYM(7)
      LINE(14) = ISYM(6)
      LINE(108) = ISYM(9)
      LINE(109) = ISYM(9)
      LINE(110) = ISYM(6)
    4 IF (Y.NE.-37) GO TO 5
      LINE(33) = ISYM(8)
      LINE(34) = ISYM(10)
      LINE(35) = ISYM(6)
      LINE(87) = ISYM(9)
      LINE(88) = ISYM(6)
      LINE(89) = ISYM(6)
    5 CONTINUE
      RETURN
      END
      SUBROUTINE NUMBER (N1,N2,X,IX)
      REAL*8 Y, X
      CHARACTER*1 A
      CHARACTER*1 AMNUS,PLUS,POINT,AK,AM,AU
      COMMON /A/ A(80)
      CHARACTER*1 B(10)
      DATA B/'0','1','2','3','4','5','6','7','8','9'/
      DATA AMNUS,PLUS,POINT/'-','+','.'/
      DATA AK,AM,AU/'K','M','U'/
      N = N1
      NSIGN = 0
      II = -1
      IX = 0
      ISET = 0
      IF (A(N).EQ.PLUS) N=N+1
      IF (A(N).NE.AMNUS) GO TO 1
      NSIGN = 1
      N = N+1
C
    1 DO 6 I=N,80
      IF (A(I).NE.POINT) GO TO 2
      ISET = 1
      GO TO 6
    2 IF (ISET.EQ.1) II = II+1
C
      DO 3 K=1,10
      IF (A(I).EQ.B(K)) GO TO 4
    3 CONTINUE
C
      GO TO 7
C
    4 DO 5 K=1,10
      KK = K-1
      IF (A(I).EQ.B(K)) NUMB=KK
    5 CONTINUE
C
      IX = NUMB+10*IX
      N2 = I+1
    6 CONTINUE
C
    7 IF (NSIGN.EQ.1) IX = -IX
      Y = IX
      IF (II.LT.0) II = 0
      X = Y/(10**II)
      IF (A(N2).EQ.POINT) N2=N2+1
      IF (A(N2).EQ.AK) X = X*1000.
      IF (A(N2).EQ.AM) X = X*0.001
      IF (A(N2).EQ.AU) X = X*0.000001
      IF((A(N2).EQ.AK).OR.(A(N2).EQ.AM).OR.(A(N2).EQ.AU)) N2=N2+1
      N1 = N2
      RETURN
      END
      SUBROUTINE POLPRT (NAME,Y)
      REAL*8 X, Y, D, S, AMAG, BIN, OK, DIM, UL, ULL, FACTOR
      REAL*8 DATAX, DATAY
      CHARACTER*1 ISYM,LINE
      COMMON /PLOT/ ISYM(14),LINE(130)
      DIMENSION X(360), Y(360), DATAX(360), DATAY(360)
      CHARACTER*4 TITLA(2), TITL1, TITL2(2)
      DATA TITLA/'PHI ','THET'/
      DATA DATAX/360*0.0/,DATAY/360*0.0/
      N = 360
      DIM = 1.0
      NST = 1
      KST = 1
C
C     S IS SCALE FACTOR OF PRINTER:
C     ABSCISSA CHAR. PER INCH / ORDINATE CHAR. PER INCH
C
      S = 10.0/8.0
C
C     ZERO DATAX AND DATAY
C
C
      DO 1 IA=1,N
      D = IA-1
    1 X(IA) = D*3.1415927/180.0
C
C
C     FACTOR IS THE NORMALIZING DIVISOR
C
      FACTOR = Y(1)
C
      DO 2 IA=2,N
    2 IF (FACTOR.LT.Y(IA)) FACTOR=Y(IA)
C
C
      IF (NAME.EQ.1) TITL1=TITLA(1)
      IF (NAME.EQ.2) TITL1=TITLA(2)
      IF ((NAME.EQ.3).OR.(NAME.EQ.4).OR.(NAME.EQ.7).OR.(NAME.EQ.8)) TITL
     12(1)=TITLA(1)
      IF ((NAME.EQ.5).OR.(NAME.EQ.6).OR.(NAME.EQ.9).OR.(NAME.EQ.10)) TIT
     1L2(1)=TITLA(2)
      IF ((NAME.EQ.3).OR.(NAME.EQ.5).OR.(NAME.EQ.7).OR.(NAME.EQ.9)) TITL
     12(2)=TITLA(1)
      IF ((NAME.EQ.4).OR.(NAME.EQ.6).OR.(NAME.EQ.8).OR.(NAME.EQ.10)) TIT
     1L2(2)=TITLA(2)
      IF (FACTOR.GT.1.E-32) GO TO 3
      IF (NAME.LE.2) WRITE (6,9) TITL1
      IF (NAME.GE.3) WRITE (6,10) TITL2
      RETURN
C
C     NORMALIZE DATA TO ONE
C
C
    3 DO 4 IA=1,N
    4 Y(IA) = Y(IA)/FACTOR
C
C
      IF (NAME.LE.2) WRITE (6,11) TITL1,FACTOR
      IF ((NAME.GE.3).AND.(NAME.LE.6)) WRITE (6,13) TITL2,FACTOR
      IF (NAME.GE.7) WRITE (6,12) TITL2,FACTOR
C     FILL DATAX AND DATAY ARRAY FROM X AND Y ARRAY
C
C
      DO 5 IA=1,N
      DATAX(IA) = Y(IA)*COS(X(IA))
    5 DATAY(IA)= Y(IA)*SIN(X(IA))
C
C
C     SORT DATA BY ORDINATE MAGNITUDE
C
      CALL SART (DATAX,DATAY,N)
C
C     DATAX AND DATAY ARE SORTED BY DESENDING MAGNITUDE ON THE DATAY VAL
C     SET UP FOR PLOTTING POLAR GRID WITH DATA
C
C
      DO 8 IYY=1,81
C
      CALL PTPLOT (IYY,S)
C
C     LINE IS RETURNED WITH POLAR GRID INFORMATION
C
C     SET UP 'Y' BIN SIZE UPPER AND LOWER LIMITS
C     ULL IS THE LOWER BIN LIMIT
C     UL IS THE UPPER BIN LIMIT
C
      BIN = DIM/80.0
      ULL = DIM-(2*IYY-1)*BIN
      UL = ULL+2*BIN
C
C
C     CYCLE THROUGH DATA TO FIND WHICH ONES FALL IN 'Y' BINS
C
C
      IF (NST.GT.N) GO TO 7
C
      DO 6 JJ=NST,N
      IF (DATAY(JJ).LT.ULL) GO TO 7
      KST = JJ
      AMAG = SQRT(DATAX(JJ)*DATAX(JJ)+DATAY(JJ)*DATAY(JJ))
C
C     CHECK THAT MAGNITUDE IS NOT OVER DIM
C
      IF (AMAG.GT.DIM) GO TO 6
```

```
C
C     OK IS THE FINAL LINE POSITION FOR THE '*'
C
      OK = DATAX(JJ)*S*40.0/DIM+61.0
      IF (OK.LT.10.0) GO TO 6
      K = INT(OK)
      K = IABS(K)
      OK = ABS(OK)
      IF ((OK-K).GT.0.5) K=K+1
      IF (OK.LT.10.0.OR.OK.GT.111.0) GO TO 6
      LINE(K) = ISYM(4)
    6 CONTINUE
C
    7 CONTINUE
      NST = KST+1
C
C     PRINT OUT ONE LINE OF PLOT
C
      WRITE (6,14) LINE
    8 CONTINUE
C
      RETURN
C
    9 FORMAT (10X,1A4,' COMPONENT OF THE ELECTRIC FIELD IS LESS'/10X,
     1 'THAN 1.E-64, THEREFORE THIS FIELD WAS NOT '/10X,'PLOTTED.   EXEC
     2UTION WILL CONTINUE AS NORMAL.'//)
   10 FORMAT (10X,'THE MAXIMUM VALUE OF THE BISTATIC PATTERN FOR '/
     1 10X,1A4,'-',1A4,' (INCIDENT-SCATTERED) IS LESS THAN '/
     2 10X, ' 1.E-30.)   POLAR PLOT NOT CALLED.'///)
   11 FORMAT ('1',1A4,' ELECTRIC FIELD ANTENNA PATTERN FOR SPECIFIED PLA
     1NE.',9X,'NORMALIZING FACTOR= ',E10.5)
   12 FORMAT ('1BISTATIC SCATTERING PATTERN FOR',1A4,'-',1A4,'(INCIDENT-
     1SCATTERED) POLARIZATION.',9X,'NORMALIZING FACTOR=',E10.5)
   13 FORMAT ('1BACKSCATTERING PATTERN FOR',1A4,'-',1A4,'(INCIDENT-SCATT
     1ERED) POLARIZATION.',9X,'NORMALIZING FACTOR=',E10.5)
   14 FORMAT (1X,130A1)
      END
      SUBROUTINE PTPLOT (IYY,S)
C
C     THIS SUBROUTINE SETS UP POLAR GRID INFORMATION
C
      REAL*8 X, Z, S
      CHARACTER*1 LINE, ISYM, ISYN(14)
      COMMON /PLOT/ ISYM(14),LINE(130)
      INTEGER Y,YY,W
      DATA ISYN/1H+,1H.,1H ,1H*,1H/,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H8,1H9/
C
C     SET UP ISYM FROM ISYN FOR COMMON
C
C
      DO 1 K=1,14
      ISYM(K) = ISYN(K)
    1 CONTINUE
C
C
C     CLEAR LINE AND SET TO BLANK
```

```
C
C
      DO 2 I=1,130
    2 LINE(I) = ISYM(3)
C
      Y = 41-IYY
      IF (Y.EQ.0) GO TO 7
C
C     SET UP EQUATIONS FOR CONCENTRIC CIRCLES
C
      YY = Y*Y
      Z = (YY*2.5/2)*S
      X = 61.0+SQRT(2500.0-Z)
      CALL LINECK (X,Y)
      IF (Y.GT.32.OR.Y.LT.-32) GO TO 3
      X = 61.0+SQRT(1600.0-Z)
      CALL LINECK (X,Y)
    3 IF (Y.GT.24.OR.Y.LT.-24) GO TO 4
      X = 61.0+SQRT(900.0-Z)
      CALL LINECK (X,Y)
    4 IF (Y.GT.16.OR.Y.LT.-16) GO TO 5
      X = 61.0+SQRT(400.0-Z)
      CALL LINECK (X,Y)
    5 IF (Y.GT.8.OR.Y.LT.-8) GO TO 6
      X = 61.0+SQRT(100-Z)
      CALL LINECK (X,Y)
C     SET UP EQUATIONS FOR MULTIPLES OF 30 DEGREES
    6 X = 61.0+1.732051*Y*S
      CALL LINECK (X,Y)
      X = 61.0+Y*S/1.732051
    7 CALL LINECK (X,Y)
C
C     PUT IN POLAR PLOT NUMBER LABELS
C
      CALL NUMB (Y)
      W = IABS(Y)
C
C     FILL IN POLAR PLOT AT 000, 090, 180, AND 270
C
      IF (W.NE.40) GO TO 8
      LINE(55) = ISYM(2)
      LINE(57) = ISYM(2)
      LINE(59) = ISYM(2)
      LINE(63) = ISYM(2)
      LINE(65) = ISYM(2)
      LINE(67) = ISYM(2)
    8 IF (W.NE.32) GO TO 9
      LINE(56) = ISYM(2)
      LINE(58) = ISYM(2)
      LINE(60) = ISYM(2)
      LINE(62) = ISYM(2)
      LINE(64) = ISYM(2)
      LINE(66) = ISYM(2)
    9 IF (W.NE.24) GO TO 10
      LINE(57) = ISYM(2)
      LINE(59) = ISYM(2)
```

```
  LINE(60) = ISYM(2)
  LINE(62) = ISYM(2)
  LINE(63) = ISYM(2)
  LINE(65) = ISYM(2)
10 IF (W.NE.16) GO TO 11
  LINE(58) = ISYM(2)
  LINE(60) = ISYM(2)
  LINE(62) = ISYM(2)
  LINE(64) = ISYM(2)
11 IF (W.NE.08) GO TO 12
  LINE(59) = ISYM(2)
  LINE(63) = ISYM(2)
12 CONTINUE
  RETURN
  END
  SUBROUTINE READD(IA,IB,IBISC,ICARD,IGAIN,IGRD,INEAR,INT,ISCAT,IWR,
 1IFLAG,KFLAG,KGEN,LOAD,LZD,MSG,NBAP,NBIP,NFFP,NGEN,NM,NP,ABAP,ABAT,
 2AFFP,AFFT,ABIP,ABIT,AM,BM,CMM,ER2,ER3,ER4,FMC,HGT,PHAF,PHAI,PHIF,P
 3HII,PHSF,PHSI,THAF,THAI,THIF,THII,THSF,THSI,SIG2,SIG3,SIG4,TD2,TD3
 4,VOLT,X,XNP,Y,YNP,Z,ZLLD,ZNP,STEP)
  REAL*8 ABAP,ABAT,AFFP,AFFT,ABIP,ABIT,AM,BM,CMM,ER2,ER3
  REAL*8 ER4,FMC,HGT,PHAF,PHAI,PHIF,PHII,PHSF,PHSI,THAF,THAI
  REAL*8 THIF,THII,THSF,THSI,SIG2,SIG3,SIG4,TD2,TD3
  REAL*8 X,XNP,Y,YNP,Z,ZNP,STEP
  REAL*8 XXX, X1, YYY, ZZZ, RAD, RDEG, VDEG, RMAG, VMAG
  REAL*8 RIMAG, VIMAG, RREAL, VREAL
  CHARACTER*1 A
  CHARACTER*1  BLANK,COMMA,MINUS,PLEFT,POINT,RIGHT,SLANT
  CHARACTER*1 AA,AB,AC,AD,AE,AF,AG,AH,AI,AK,AL,AMA,AN,AO,AP,AQ,AR,
 1AS,AT,AU,AW,AX
  COMMON /A/ A(80)
  CHARACTER*1 B(80)
  COMPLEX*16 VOLT(1),ZLLD(1)
  DIMENSION IA(1), IB(1), X(1), Y(1), Z(1), KGEN(1), KFLAG(30)
  DIMENSION XNP(1), YNP(1), ZNP(1), LZD(1)
  DATA AA,AB,AC,AD,AE,AF,AG,AH,AI,AK,AL,AMA,AN,AO,AP,AQ,AR,AS,AT,AU,
 1AW,AX/'A','B','C','D','E','F','G','H','I','K','L','M','N','O','P',
 2'Q','R','S','T','U','W','X'/
  DATA BLANK,COMMA,MINUS,PLEFT,POINT,RIGHT,SLANT/' ',',','-','(','.'
 1,')','/'/
  RAD = 57.295779
  INT = 4
  IBISC = -1
  IGAIN = -1
  INEAR = -1
  ISCAT = -1
  IWR = -1
  IF (IFLAG.EQ.6) GO TO 2
  IF (MSG.NE.0) GO TO 4
 1 READ (5,76,END=72) A
 2 IF ((A(1).NE.AC).OR.(A(2).NE.BLANK).OR.(A(3).NE.BLANK).OR.(A(4).NE
 1.BLANK)) GO TO 3
  WRITE (6,74) A
  GO TO 1
 3 WRITE (6,75)
  GO TO 5
```

```
    4 READ (5,76,END=72) A
    5 ICARD = ICARD+1
      WRITE (6,77) ICARD,A
C
C       CHECK FOR KEYWORD - END
C
      IF ((MSG.NE.0).AND.((A(1).EQ.AE).AND.(A(2).EQ.AN).AND.(A(3).EQ.AD)
     1)) GO TO 70
C
C       CHECK FOR KEYWORD - STOP
C
      IF ((MSG.NE.0).AND.((A(1).EQ.AS).AND.(A(2).EQ.AT).AND.(A(3).EQ.AO)
     1.AND.(A(4).EQ.AP))) GO TO 69
C
C       CHECK FOR COMMENT LINE
C
      IF ((A(1).EQ.AC).AND.(A(2).EQ.BLANK).AND.(A(3).EQ.BLANK).AND.(A(4)
     1.EQ.BLANK)) GO TO 73
      IF (MSG.GT.0) GO TO 4
      CALL BLNK (A)
      N = 4
C
C       INSULATION
C
C       CHECK FOR KEYWORD - INSU  FOR INSULATION
C
      IF ((A(1).NE.AI).OR.(A(2).NE.AN).OR.(A(3).NE.AS).OR.(A(4).NE.AU))
     1GO TO 10
      KFLAG(20) = 1
      CALL LEFT (N)
C
C       CHECK FOR KEYWORD - RADI  FOR INSULATION RADIUS
C
    6 IF ((A(N).NE.AR).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AD).OR.(A(N+3).NE
     1.AI)) GO TO 7
      KFLAG(4) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      BM = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 6
C
C       CHECK FOR KEYWORD - DIEL  FOR INSULATION DIELECTRIC
C
    7 IF ((A(N).NE.AD).OR.(A(N+1).NE.AI).OR.(A(N+2).NE.AE).OR.(A(N+3).NE
     1.AL)) GO TO 8
      KFLAG(6) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      ER2 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 6
```

```
C
C     CHECK FOR KEYWORD - COND  FOR INSULATION CONDUCTIVITY
C
    8 IF ((A(N).NE.AC).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AN).OR.(A(N+3).NE
     1.AD)) GO TO 9
      KFLAG(5) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      SIG2 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 6
C
C     CHECK FOR KEYWORD - LOSS  FOR INSULATION LOSS
C
    9 IF ((A(N).NE.AL).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AS).OR.(A(N+3).NE
     1.AS)) GO TO 71
      KFLAG(7) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      TD2 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 6
C
C     WIRE
C
C     CHECK FOR KEYWORD - WIRE
C
   10 IF ((A(1).NE.AW).OR.(A(2).NE.AI).OR.(A(3).NE.AR).OR.(A(4).NE.AE))
     1GO TO 13
      CALL LEFT (N)
C
C     CHECK FOR KEYWORD - RADI   FOR WIRE RADIUS
C
   11 IF ((A(N).NE.AR).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AD).OR.(A(N+3).NE
     1.AI)) GO TO 12
      KFLAG(2) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      AM = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 11
C
C     CHECK FOR KEYWORD - COND   FOR WIRE CONDUCTIVITY
C
   12 IF ((A(N).NE.AC).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AN).OR.(A(N+3).NE
     1.AD)) GO TO 71
      KFLAG(3) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      CMM = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 11
C
C     EXTERNAL MEDIUM
C
C
C     CHECK FOR KEYWORD - EXTE   FOR EXTERNAL MEDIUM
C
   13 IF ((A(1).NE.AE).OR.(A(2).NE.AX).OR.(A(3).NE.AT).OR.(A(4).NE.AE))
     1GO TO 17
      KFLAG(8) = 1
      CALL LEFT (N)
C
C     CHECK FOR KEYWORD - COND   FOR EXTERNAL MEDIUM CONDUCTIVITY
C
   14 IF ((A(N).NE.AC).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AN).OR.(A(N+3).NE
     1.AD)) GO TO 15
      KFLAG(9) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      SIG3 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 14
C
C     CHECK FOR KEYWORD - DIEL   FOR EXTERNAL MEDIUM DIELECTRIC
C
   15 IF ((A(N).NE.AD).OR.(A(N+1).NE.AI).OR.(A(N+2).NE.AE).OR.(A(N+3).NE
     1.AL)) GO TO 16
      KFLAG(10) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      ER3 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 14
C
C     CHECK FOR KEYWORD - LOSS   FOR EXTERNAL MEDIUM LOSS
C
   16 IF ((A(N).NE.AL).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AS).OR.(A(N+3).NE
     1.AS)) GO TO 71
      KFLAG(11) = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      TD3 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 14
C
C
C     LOAD
```

```
C
C
C     CHECK FOR KEYWORD - LOAD   FOR ANTENNA LOADING
C
   17 IF ((A(1).NE.AL).OR.(A(2).NE.AO).OR.(A(3).NE.AA).OR.(A(4).NE.AD))
     1GO TO 18
      KFLAG(14) = 1
      GO TO 19
C
C     CHECK FOR KEYWORD - IMPE   FOR IMPEDANCE LOADING
C
   18 IF ((A(1).NE.AI).OR.(A(2).NE.AMA).OR.(A(3).NE.AP).OR.(A(4).NE.AE))
     1 GO TO 22
   19 I = 1
      IF(KFLAG(24).EQ.1) I=LOAD+1
      KFLAG(24) = 1
      CALL LEFT (N)
   20 CALL NUMBER (N,N2,X1,IX)
      IF (IX.LE.0) GO TO 21
      LZD(I) = IX
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      RMAG = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      RDEG = X1
      RREAL = RMAG*COS(RDEG/RAD)
      RIMAG = RMAG*SIN(RDEG/RAD)
      ZLLD(I) = CMPLX(RREAL,RIMAG)
      LOAD = I
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF(A(N2+1).EQ.PLEFT) GO TO 800
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      I=I+1
      GO TO 20
   21 KFLAG(24) = -1
      LOAD = -1
      GO TO 4
  800 READ(5,76) A
      ICARD=ICARD+1
      WRITE(6,77) ICARD,A
      N=1
      GOTO 20
C
C     FREQUENCY
C
C
C     CHECK FOR KEYWORD - FREQ   FOR FREQUENCY
C
   22 IF ((A(1).NE.AF).OR.(A(2).NE.AR).OR.(A(3).NE.AE).OR.(A(4).NE.AQ))
     1GO TO 23
      KFLAG(1) = 1
      CALL LEFT (N)
      CALL NUMBER (N,N2,X1,IX)
      FMC = X1
      GO TO 4
C
C      PLOT
C
C
C      CHECK FOR KEYWORD - PLOT
C
   23 IF ((A(1).NE.AP).OR.(A(2).NE.AL).OR.(A(3).NE.AO).OR.(A(4).NE.AT))
     1GO TO 31
      KFLAG(22) = 1
      CALL LEFT (N)
C
C      CHECK FOR KEYWORD - FARF   FOR PLOT FAR FIELD
C
   24 IF ((A(N).NE.AF).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AR).OR.(A(N+3).NE
     1.AF)) GO TO 25
      IGAIN = 1
      NFFP = 1
      GO TO 27
C
C      CHECK FOR KEYWORD - BIST   FOR PLOT BISTATIC
C
   25 IF ((A(N).NE.AB).OR.(A(N+1).NE.AI).OR.(A(N+2).NE.AS).OR.(A(N+3).NE
     1.AT)) GO TO 26
      IBISC = 1
      NBIP = 1
      GO TO 27
C
C      CHECK FOR KEYWORD - BACK   FOR PLOT BACKSCATTERING
C
   26 IF ((A(N).NE.AB).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AC).OR.(A(N+3).NE
     1.AK)) GO TO 71
      ISCAT = 1
      NBAP = 1
C
C
C
   27 DO 28 I=N,80
      K = I+1
      IF (A(I).EQ.SLANT) GO TO 29
   28 CONTINUE
C
C
C
      GO TO 71
   29 N = K
C
C      CHECK FOR KEYWORD - THET   FOR PLOT THETA ANGLES
C
      IF ((A(N).NE.AT).OR.(A(N+1).NE.AH).OR.(A(N+2).NE.AE).OR.(A(N+3).NE
     1.AT)) GO TO 30
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      IF (NFFP.EQ.1) AFFT=X1
      IF (NBIP.EQ.1) ABIT=X1
      IF (NBAP.EQ.1) ABAT=X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 24
C
C     CHECK FOR KEYWORD - PHI   FOR PLOT PHI ANGLES
C
   30 IF ((A(N).NE.AP).OR.(A(N+1).NE.AH).OR.(A(N+2).NE.AI)) GO TO 71
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      IF (NFFP.EQ.1) AFFP=X1
      IF (NBIP.EQ.1) ABIP=X1
      IF (NBAP.EQ.1) ABAP=X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 24
C
C     OUTPUT
C
C
C     CHECK FOR KEYWORD - OUTP   FOR OUTPUT
C
   31 IF ((A(1).NE.AO).OR.(A(2).NE.AU).OR.(A(3).NE.AT).OR.(A(4).NE.AP))
     1GO TO 44
      KFLAG(22) = 1
      CALL LEFT (N)
C
C     CHECK FOR KEYWORD - BIST   FOR OUTPUT BISTATIC
C
   32 IF ((A(N).NE.AB).OR.(A(N+1).NE.AI).OR.(A(N+2).NE.AS).OR.(A(N+3).NE
     1.AT)) GO TO 33
      KFLAG(18) = 1
      IBISC = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      PHSI = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      PHSF = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THSI = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THSF = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 32
C
C     CHECK FOR KEYWORD - FARF   FOR OUTPUT FAR FIELD
C
   33 IF ((A(N).NE.AF).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AR).OR.(A(N+3).NE
     1.AF)) GO TO 34
      KFLAG(16) = 1
      IGAIN = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      PHAI = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      PHAF = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THAI = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THAF = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 32
C
C     CHECK FOR KEYWORD - NEAR   FOR OUTPUT NEAR FIELD
C
C     THIS IS THE ORIGINAL 'NEAR' CONTAINED WITHIN THE OUTPUT CARD FIELD
C      THIS ORIGINAL NEAR CAN ONLY EXCEPT POINTS WITHIN ONE STATEMENT ON
C     ONE LINE. THE NEW 'NEAR' IS AT A HIGHER LEVEL AND CAN ACCEPT LISTS
C        - V3.2D THE NEW 'NEAR' MODIFICATION ADDED 17 JULY 2004 RAY L. CROSS
C
   34 IF ((A(N).NE.AN).OR.(A(N+1).NE.AE).OR.(A(N+2).NE.AA).OR.(A(N+3).NE
     1.AR)) GO TO 40
      KFLAG(19) = 1
      INEAR = 2
      CALL EQUAL (N)
C
C     IF THERE IS A LIST OF NEAR FIELD POINTS INCLOSED BY PARENS GO TO 35
C
      IF (A(N).EQ.PLEFT) GO TO 35
C
C     READ THE SINGLE NEAR FIELD POINT
C
      INEAR = 1
      I = 1
      CALL NUMBER (N,N2,X1,IX)
      XNP(I) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      YNP(I) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      ZNP(I) = X1
      GO TO 39
C
C     READ THE LIST OF NEAR FIELD POINTS INCLOSED BY PARENS
C      THIS IS LIMITED TO A SINGLE LINE WITH NO CONTINUATION ALLOWED
C
   35 CONTINUE
      DO 37 L=1,50
      I = L
      N = N+1
      CALL NUMBER (N,N2,X1,IX)
      XNP(I) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      YNP(I) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      ZNP(I) = X1
      INEAR = L+1
      IF (A(N2).EQ.RIGHT) GO TO 38
      N = N2
   37 CONTINUE
C
C
C
      GO TO 71
   38 N2 = N2+1
      INEAR = INEAR-1
   39 IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 32
C
C
C
C     CHECK FOR KEYWORD - BACK   FOR OUTPUT BACKSCATTERING
C
   40 IF ((A(N).NE.AB).OR.(A(N+1).NE.AA).OR.(A(N+2).NE.AC).OR.(A(N+3).NE
     1.AK)) GO TO 41
      KFLAG(17) = 1
      ISCAT = 1
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      PHII = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      PHIF = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THII = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      THIF = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 32
C
C     CHECK FOR KEYWORD - CURR   FOR OUTPUT ANTENNA STRUCTURE CURRENTS
C
   41 IF ((A(N).NE.AC).OR.(A(N+1).NE.AU).OR.(A(N+2).NE.AR).OR.(A(N+3).NE
     1.AR)) GO TO 43
      KFLAG(15) = 1
      IWR = 1
C
C
```

```
C
      NSPL = N
      DO 42 K=NSPL,80
      IF (A(K).EQ.RIGHT) GO TO 4
      N = K+1
      IF (A(K).EQ.SLANT) GO TO 32
   42 CONTINUE
C
      GO TO 71
C
C     CHECK FOR KEYWORD - STEP   FOR OUTPUT ANGLE STEP SIZE
C
   43 IF ((A(N).NE.AS).OR.(A(N+1).NE.AT).OR.(A(N+2).NE.AE).OR.(A(N+3).NE
     1.AP)) GO TO 71
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      STEP = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 32
C
C     FEED POINT
C
C
C     CHECK FOR KEYWORD - FEED   FOR FEED POINT
C
   44 IF ((A(1).NE.AF).OR.(A(2).NE.AE).OR.(A(3).NE.AE).OR.(A(4).NE.AD))
     1GO TO 45
      KFLAG(13) = 1
      GO TO 46
C
C     CHECK FOR KEYWORD - GENE   FOR GENERATOR SEGMENT
C
   45 IF ((A(1).NE.AG).OR.(A(2).NE.AE).OR.(A(3).NE.AN).OR.(A(4).NE.AE))
     1GO TO 49
      KFLAG(23) = 1
   46 NGEN = 0
      CALL LEFT (N)
   47 CALL NUMBER (N,N2,X1,IX)
      NGEN = NGEN+1
      KGEN(NGEN) = IX
      IF (A(N2).EQ.RIGHT) GO TO 4
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      VMAG = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      VDEG = X1
      VREAL = VMAG*COS(VDEG/RAD)
      VIMAG = VMAG*SIN(VDEG/RAD)
      VOLT(NGEN) = CMPLX(VREAL,VIMAG)
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      IF ((A(N2).EQ.SLANT).AND.(A(N2+1).EQ.BLANK)) GO TO 48
      N = N2+1
      GO TO 47
   48 READ (5,76) A
      ICARD = ICARD+1
      WRITE (6,77) ICARD,A
      N = 1
      CALL BLNK (A)
      GO TO 47
C
C     V3.2D NEW 'NEAR' TOP LEVEL KEYWORD TO PERMIT LIST INPUT OF NEAR FIELD INPUT
POINTS
C       THIS 'NEAR' IS AT A HIGHER LEVEL AND IS NOT THE SAME NEAR THAT IS READ INSIDE
C        THE OUTPUT CARD FIELD - MODIFICATION ADDED 17 JULY 2004 RAY L. CROSS
C
C     CHECK FOR KEYWORD - NEAR   FOR ALTERNATE INPUT LIST OF NEAR FIELD POINTS
C
   49 IF ((A(1).EQ.'N').AND.(A(2).EQ.'E').AND.(A(3).EQ.'A').AND.
     1  (A(4).EQ.'R')) GO TO 90
C
C
C     DESCRIPTION
C
C      *********** DESCRIPTION MODIFIED TO ACCEPT LIST INPUT ***********
C
C
C     CHECK FOR KEYWORD - DNOD   FOR ALTERNATE INPUT LIST DNODE
C
      IF ((A(1).EQ.'D').AND.(A(2).EQ.'N').AND.(A(3).EQ.'O').AND.
     1  (A(4).EQ.'D')) GO TO 85
C
C     CHECK FOR KEYWORD - DESC   FOR DESCRIPTION IN ORIGINAL FORMAT
C
      IF ((A(1).NE.AD).OR.(A(2).NE.AE).OR.(A(3).NE.AS).OR.(A(4).NE.AC))
     1GO TO 52
      KFLAG(12) = 1
      J = 0
      CALL LEFT (N)
   50 CALL NUMBER (N,N2,X1,IX)
      J = J+1
      NM = J
      IA(J) = IX
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      IB(J) = IX
      IF (A(N2).EQ.RIGHT) GO TO 4
C
C      LOOK FOR A CONTINUATION CARD
C
      IF (A(N2+1).EQ.PLEFT) GO TO 51
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 50
C
C      PROCESS CONTINUATION CARD
C
   51 READ (5,76) A
      ICARD = ICARD+1
      WRITE(6,77)ICARD,A
      CALL BLNK (A)
      N = 1
      GO TO 50
C
C     GEOMETRY
C
C     ***********GEOMETRY MODIFIED TO ACCEPT LIST INPUT***************
C
C
C     CHECK FOR KEYWORD - GXYZ   FOR ALTERNATE INPUT POINT LIST GXYZ
C
   52 IF ((A(1).EQ.'G').AND.(A(2).EQ.'X').AND.(A(3).EQ.'Y').AND.
     1  (A(4).EQ.'Z')) GO TO 80
C
C     CHECK FOR KEYWORD - GEOM   FOR GEOMETRY ORIGINAL INPUT FORMAT
C
      IF ((A(1).NE.AG).OR.(A(2).NE.AE).OR.(A(3).NE.AO).OR.(A(4).NE.AMA))
     1 GO TO 55
      KFLAG(12) = 1
      JJ = 0
      CALL LEFT (N)
   53 CALL NUMBER (N,N2,X1,IX)
      JJ = JJ+1
      NP = JJ
      X(JJ) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      Y(JJ) = X1
      N = N2+1
      CALL NUMBER (N,N2,X1,IX)
      Z(JJ) = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
C
C      LOOK FOR A CONTINUATION CARD
C
      IF (A(N2+1).EQ.PLEFT) GO TO 54
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 53
C
C      PROCESS CONTINUATION CARD
C
   54 READ (5,76) A
      ICARD = ICARD+1
      WRITE (6,77) ICARD,A
      CALL BLNK (A)
      N = 1
      GO TO 53
C
C
C
C     INTERVAL FOR CALCULATION
C
C
C     CHECK FOR KEYWORD - INTE   FOR INTERVAL OF CALCULATION
```

```
C
   55 IF ((A(1).NE.AI).OR.(A(2).NE.AN).OR.(A(3).NE.AT).OR.(A(4).NE.AE))
     1GO TO 56
      KFLAG(21) = 1
      CALL LEFT (N)
      CALL NUMBER (N,N2,X1,IX)
      INT = IX
      IF (A(N2).EQ.RIGHT) GO TO 4
      GO TO 71
C
C
C
C     GROUND
C
C
C     CHECK FOR KEYWORD - GROU   FOR GROUND
C
   56 IF ((A(1).NE.AG).OR.(A(2).NE.AR).OR.(A(3).NE.AO).OR.(A(4).NE.AU))
     1GO TO 66
      KFLAG(25) = 1
      KFLAG(26) = 1
      IGRD = 2
      CALL LEFT (N)
C
C     CHECK FOR KEYWORD - PERF   FOR PERFECT GROUND
C
   57 IF ((A(N).NE.AP).OR.(A(N+1).NE.AE).OR.(A(N+2).NE.AR).OR.(A(N+3).NE
     1.AF)) GO TO 58
      IGRD = 1
      GO TO 64
C
C     CHECK FOR KEYWORD - GOOD   FOR GOOD GROUND
C
   58 IF ((A(N).NE.AG).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AO).OR.(A(N+3).NE
     1.AD)) GO TO 59
      ER4 = 30.
      SIG4 = .02
      GO TO 64
C
C     CHECK FOR KEYWORD - POOR   FOR POOR GROUND
C
   59 IF ((A(N).NE.AP).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AO).OR.(A(N+3).NE
     1.AR)) GO TO 60
      ER4 = 4.
      SIG4 = .001
      GO TO 64
C
C     CHECK FOR KEYWORD - SEA   FOR SEA/OCEAN AS THE 'GROUND'
C
   60 IF ((A(N).NE.AS).OR.(A(N+1).NE.AE).OR.(A(N+2).NE.AA)) GO TO 61
      ER4 = 80.
      SIG4 = 4.
      GO TO 64
C
C     CHECK FOR KEYWORD - HEIG   FOR STRUCTURE HEIGHT ABOVE THE GROUND
C
```

```
   61 IF ((A(N).NE.AH).OR.(A(N+1).NE.AE).OR.(A(N+2).NE.AI).OR.(A(N+3).NE
     1.AG)) GO TO 62
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      HGT = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 57
C
C     CHECK FOR KEYWORD - COND   FOR GROUND CONDUCTIVITY
C
   62 IF ((A(N).NE.AC).OR.(A(N+1).NE.AO).OR.(A(N+2).NE.AN).OR.(A(N+3).NE
     1.AD)) GO TO 63
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      SIG4 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 57
C
C     CHECK FOR KEYWORD - DIEL   FOR GROUND DIELECTRIC
C
   63 IF ((A(N).NE.AD).OR.(A(N+1).NE.AI).OR.(A(N+2).NE.AE).OR.(A(N+3).NE
     1.AL)) GO TO 71
      CALL EQUAL (N)
      CALL NUMBER (N,N2,X1,IX)
      ER4 = X1
      IF (A(N2).EQ.RIGHT) GO TO 4
      IF (A(N2).NE.SLANT) GO TO 71
      N = N2+1
      GO TO 57
C
C
C
   64 NSPL = N
      DO 65 K=NSPL,80
      IF (A(K).EQ.RIGHT) GO TO 4
      N = K+1
      IF (A(K).EQ.SLANT) GO TO 57
   65 CONTINUE
C
C
C
      GO TO 71
C
C
C
C       ** STOP, CHANGE, END **
C
C
C     CHECK FOR KEYWORD - STOP
C
   66 IF ((A(1).NE.AS).OR.(A(2).NE.AT).OR.(A(3).NE.AO).OR.(A(4).NE.AP))
     1GO TO 67
```

```
      IFLAG = 2
      RETURN
C
C     CHECK FOR KEYWORD - CHAN   FOR CHANGE
C
   67 IF ((A(1).NE.AC).OR.(A(2).NE.AH).OR.(A(3).NE.AA).OR.(A(4).NE.AN))
     1GO TO 68
      IFLAG = 3
      RETURN
C
C     CHECK FOR KEYWORD - END
C
   68 IF ((A(1).NE.AE).OR.(A(2).NE.AN).OR.(A(3).NE.AD)) GO TO 71
      IFLAG = 1
      RETURN
   69 IFLAG = 5
      RETURN
   70 IFLAG = 4
      RETURN
   71 MSG = 1
      KFLAG(30) = ICARD
      GO TO 4
   72 IF (IFLAG.NE.5) WRITE (6,78)
      IFLAG = 5
      RETURN
C
   73 IFLAG = 6
      ICARD = ICARD-1
      RETURN
C
C     ***************INPUT MODIFIED TO ACCEPT LISTS******************
C
C     JUMP POINT FOR THE GXYZ LIST INPUT FOR GEOMETRY POINT LIST
C
   80 JJ = 0
      KFLAG(12) = 1
   83 READ(5,*,ERR=4) XXX,YYY,ZZZ
      JJ = JJ + 1
      NP = JJ
      X(JJ) = XXX
      Y(JJ) = YYY
      Z(JJ) = ZZZ
      GO TO 83
C
C     JUMP POINT FOR THE DNODE LIST INPUT FOR STRUCTURE DESCRIPTION LIST
C
   85 J = 0
      KFLAG(12) = 1
   87 READ(5,*,ERR=4) IAAA,IBBB
      J = J + 1
      NM = J
      IA(J) = IAAA
      IB(J) = IBBB
      GO TO 87
C
C     V3.2D JUMP POINT FOR THE NEW 'NEAR' KEYWORK NEAR FIELD LIST INPUT
```

```
C     ADDED BY RAY L. CROSS 17 JULY 2004
C
  90 JJJ = 0
    KFLAG(22) = 1
    KFLAG(19) = 2
C           THE KFLAG(19) VALUE OF 2 REPRESENTS THE LIST INPUT
    INEAR = 0
  93 READ(5,*,ERR=4) XXX,YYY,ZZZ
    JJJ = JJJ + 1
    INEAR = JJJ
    XNP(JJJ) = XXX
    YNP(JJJ) = YYY
    ZNP(JJJ) = ZZZ
    GO TO 93
C     ******************END OF INPUT MODIFICATION*********************
C
C
  74 FORMAT (5X,80A1)
  75 FORMAT (////5X,'DATA CARDS'//)
  76 FORMAT (80A1)
  77 FORMAT (6X,I2,2X,80A1)
  78 FORMAT ('    $$$$$ END CARD/STOP CARD MISSING****')
    END
C
C
C
    SUBROUTINE RITE (IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,CJ,CG,IGRD)
    REAL*8 ACJ, BCJ, FI, PA, CCJA, PB, CCJB, AMAX
    COMPLEX*16 CJ(1),CG(1),CJA,CJB
    DIMENSION IA(1), IB(1), I1(1), I2(1), I3(1), MD(INM,4), ND(1)
    AMAX = .0
C
C
    DO 3 K=1,NM
    KA = IA(K)
    KB = IB(K)
    CJA = (.0,.0)
    CJB = (.0,.0)
    NDK = ND(K)
C
C
    DO 2 II=1,NDK
    I = MD(K,II)
    FI = 1.
    IF (KB.EQ.I2(I)) GO TO 1
    IF (KB.EQ.I1(I)) FI=-1.
    CJA = CJA+FI*CJ(I)
    GO TO 2
   1 IF (KA.EQ.I3(I)) FI=-1.
    CJB = CJB+FI*CJ(I)
   2 CONTINUE
C
C
    CG(K) = CJA
    KK = K+NM
    CG(KK) = CJB
```

```
C
C         **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      ACJ = CDABS(CJA)
      BCJ = CDABS(CJB)
      IF (ACJ.GT.AMAX) AMAX=ACJ
      IF (BCJ.GT.AMAX) AMAX=BCJ
    3 CONTINUE
C
C
      IF (IWR.GT.0) GO TO 4
      RETURN
    4 IF (AMAX.LE.0.) AMAX=1.
      WRITE (6,8)
      NMG = NM
      IF (IGRD.GT.0) NMG = NM/2
C
      DO 5 K=1,NMG
      CJA = CG(K)
      KK = K+NM
      CJB = CG(KK)
C
C         **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      CCJA = CDABS(CJA)
      CCJB = CDABS(CJB)
      ACJ = CCJA/AMAX
      BCJ = CCJB/AMAX
      PA = .0
      PB = .0
C
C         **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      IF (ACJ.GT.0.) PA=57.29577951308232*DATAN2(DIMAG(CJA),DBLE(CJA))
      IF (BCJ.GT.0.) PB=57.29577951308232*DATAN2(DIMAG(CJB),DBLE(CJB))
    5 WRITE (6,7) K,IA(K),CJA,CCJA,ACJ,PA,IB(K),CJB,CCJB,BCJ,PB
C
C
      WRITE (6,6)
      RETURN
C
C
    6 FORMAT (1H0)
C
C    V3.2D FORMATS 7 AND 8 CHANGED 18 JULY 2004 RAY L. CROSS TO ALLOW LARGER
C     SEGMENT AND NODE NUMBERS
C
    7 FORMAT (2X,I6,2(2X,I6,2X,E11.5,1X,E11.5,1X,E11.5,1X,E11.5,1X,F6.1)
     1)
    8 FORMAT (/2(46X,'NORMALIZED',5X)/'     SEG',2('     NODE',4X,'REAL'
     1,6X,'IMAGINARY',3X,'MAGNITUDE',3X,'MAGNITUDE',3X,'PHASE'))
C
      END
C
C
C
      SUBROUTINE SART (DATAX,DATAY,N)
      REAL*8 STOR, DATAX, DATAY
      DIMENSION DATAX(500), DATAY(500)
```

```
C
C     THIS ROUTINE SORTS DATA IN DATAY BY MAGNITUDE
C
      NN = N-1
C
      DO 2 I=1,NN
      NM = I+1
C
      DO 1 J=NM,N
      IF (DATAY(I).GE.DATAY(J)) GO TO 1
      STOR = DATAY(I)
      DATAY(I) = DATAY(J)
      DATAY(J) = STOR
      STOR = DATAX(I)
      DATAX(I) = DATAX(J)
      DATAX(J) = STOR
    1 CONTINUE
C
    2 CONTINUE
C
      RETURN
      END
      SUBROUTINE SGANT (IA,IB,INM,INT,ISC,I1,I2,I3,JA,JB,MD,N,ND,NM,NP,A
     1M,BM,C,CGD,CMM,D,EP2,EP3,ETA,FHZ,GAM,SGD,X,Y,Z,ZLD,ZS,ERR,IGRD)
      REAL*8 E0, TP, U0, FI, DK, FJ, DL, SGN, DMIN, OMEGA, DMAX, CPSI
      REAL*8 AM,BM,CMM,D,FHZ,X,Y,Z, ZERO, ONE
      COMPLEX*16 ERR
      COMPLEX*16 ZG,ZH,ZS,EGD,GD,CGDS,SGDS,SGDT,B01
      COMPLEX*16 P11,P12,P21,P22,Q11,Q12,Q21,Q22,EP2,EP,ETA,GAM,EP3
      COMPLEX*16 EPSILA,CWEA,BETA,ZARG
      COMPLEX*16 P(2,2),Q(2,2),CGD(1),SGD(1),C(1),ZLD(1)
      DIMENSION X(1), Y(1), Z(1), D(1), IA(1), IB(1), MD(INM,4)
      DIMENSION I1(1), I2(1), I3(1), JA(1), JB(1), ND(1), ISC(1)
      DATA E0,TP,U0/8.854E-12,6.283185307179586,1.2566E-6/
      DATA ZERO/0.0000/
      DATA ONE/1.0000/
      EP = EP3
      ICC = (N*N+N)/2
C
      DO 1 I=1,ICC
    1 C(I) = (.0,.0)
C
      ZS = (.0,.0)
      IF (CMM.LE.0.) GO TO 2
      OMEGA = TP*FHZ
      EPSILA = CMPLX(E0,-CMM*1.E6/OMEGA)
      CWEA = (.0,1.)*OMEGA*EPSILA
C
C        **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      BETA = OMEGA*SQRT(U0)*CDSQRT(EPSILA-EP)
      ZARG = BETA*AM
      CALL CBES (ZARG,B01)
      ZS = BETA*B01/CWEA
    2 ZH = ZS/(TP*AM*GAM)
      DMIN = 1.E30
      DMAX = .0
```

```
C
      DO 3 J=1,NM
      K = IA(J)
      L = IB(J)
      D(J) = SQRT((X(K)-X(L))**2+(Y(K)-Y(L))**2+(Z(K)-Z(L))**2)
      IF (D(J).LT.DMIN) DMIN=D(J)
      IF (D(J).GT.DMAX) DMAX=D(J)
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      EGD = CDEXP(GAM*D(J))
      CGD(J) = (EGD+1./EGD)/2.
    3 SGD(J) = (EGD-1./EGD)/2.
C
      IF (DMIN.LT.2.*AM) GO TO 4
C
C          **** V3.2D FIXED FOLLOWING 2 LINES FOR DOUBLE COMPLEX
      IF (CDABS(GAM*AM).GT. 0.06) GO TO 4
      IF (CDABS(GAM*DMAX).GT. 3.0) GO TO 4
      IF (AM.GT.0.) GO TO 5
    4 CONTINUE
C      N=0
      WRITE (6,24) AM,DMAX,DMIN
      WRITE (6,25)
C
    5 DO 19 K=1,NM
      IFLAG = 0
      IF ((IGRD.GT.0).AND.(K.GT.NM/2)) IFLAG=1
      NDK = ND(K)
      KA = IA(K)
      KB = IB(K)
      DK = D(K)
      CGDS = CGD(K)
      SGDS = SGD(K)
C
      DO 19 L=1,NM
      JFLAG = 0
      IF ((IGRD.GT.0).AND.(L.GT.NM/2)) JFLAG=1
      NDL = ND(L)
      LA = IA(L)
      LB = IB(L)
      DL = D(L)
      SGDT = SGD(L)
      NIL = 0
C
      DO 19 II=1,NDK
      I = MD(K,II)
      MM = (I-1)*N-(I*I-I)/2
      FI = 1.
      IF (KB.EQ.I2(I)) GO TO 6
      IF (KB.EQ.I1(I)) FI=-1.
      IS = 1
      GO TO 7
    6 IF (KA.EQ.I3(I)) FI=-1.
      IS = 2
C
    7 DO 19 JJ=1,NDL
```

```
      J = MD(L,JJ)
      MMM = MM+J
      IF (I.GT.J) GO TO 19
      FJ = 1.
      IF (LB.EQ.I2(J)) GO TO 8
      IF (LB.EQ.I1(J)) FJ=-1.
      JS = 1
      GO TO 9
    8 IF (LA.EQ.I3(J)) FJ=-1.
      JS = 2
    9 IF (NIL.NE.0) GO TO 18
      NIL = 1
      IF (K.EQ.L) GO TO 14
C     ****The following line removed because results were only used to
C          Test for zero which caused an overflow under some conditions
C
C      IND = (LA-KA)*(LB-KA)*(LA-KB)*(LB-KB)
C
      NGRD = IGRD
      IF (IFLAG.EQ.JFLAG) IGRD=-1
C      ***** The test for IND changed *****
C
C      IF (IND.EQ.0) GO TO 10
C
C     SUBSTITUTE FOR CALCULATION TEST OF IND
      IF (LA*1.0 .EQ. KA*1.0) GO TO 10
      IF (LB*1.0 .EQ. KA*1.0) GO TO 10
      IF (LA*1.0 .EQ. KB*1.0) GO TO 10
      IF (LB*1.0 .EQ. KB*1.0) GO TO 10
C     SEGMENTS K AND L SHARE NO POINTS
      CALL GGS (X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),X(LA),Y(LA),Z(LA),X(
     1LB),Y(LB),Z(LB),AM,DK,CGDS,SGDS,DL,SGDT,INT,ETA,GAM,P(1,1),P(1,2),
     2P(2,1),P(2,2),ERR,IGRD)
      IGRD = NGRD
      GO TO 18
C     SEGMENTS K AND L SHARE ONE POINT (THEY INTERSECT)
   10 KG = 0
      JM = KB
      JC = KA
      KF = 1
C
C      SUBSTITUTE IND CALCULATION AND TEST
C      IND = (KB-LA)*(KB-LB)
C      IF (IND.NE.0) GO TO 11
      IF ( (KB*1.0 .NE. LA*1.0) .AND. (KB*1.0 .NE. LB*1.0) ) GO TO 11
      JC = KB
      KF = -1
      JM = KA
      KG = 3
   11 LG = 3
      JP = LA
      LF = -1
      IF (LB.EQ.JC) GO TO 12
      JP = LB
      LF = 1
      LG = 0
```

```
  12 SGN = KF*LF
    CPSI = ((X(JP)-X(JC))*(X(JM)-X(JC))+(Y(JP)-Y(JC))*(Y(JM)-Y(JC))+(Z
   1(JP)-Z(JC))*(Z(JM)-Z(JC)))/(DK*DL)
    CALL GGMM (ZERO,DK,ZERO,DL,AM,CGDS,SGDS,SGDT,CPSI,ETA,GAM,Q(1,1),
   1Q(1,2),Q(2,1),Q(2,2))
C
    DO 13 KK=1,2
    KP = IABS(KK-KG)
C
    DO 13 LL=1,2
    LP = IABS(LL-LG)
    P(KP,LP) = SGN*Q(KK,LL)
  13 CONTINUE
C
    IGRD=NGRD
    GO TO 18
C     K=L  (SELF REACTION OF SEGMENT K)
  14 Q11 = (.0,.0)
    Q12 = (.0,.0)
    IF (CMM.LE.0.) GO TO 15
    GD = GAM*DK
    ZG = ZH/(SGDS**2)
    Q11 = ZG*(SGDS*CGDS-GD)/2.
    Q12 = ZG*(GD*CGDS-SGDS)/2.
  15 ISCK = ISC(K)
    P11 = (.0,.0)
    P12 = (.0,.0)
    IF (ISCK.EQ.0) GO TO 16
    IF (BM.LE.AM) GO TO 16
    CALL DSHELL (AM,BM,DK,CGDS,SGDS,EP2,EP,ETA,GAM,P11,P12)
  16 Q11 = P11+Q11
    Q12 = P12+Q12
    CALL GGMM (ZERO,DK,ZERO,DK,AM,CGDS,SGDS,SGDS,ONE,ETA,GAM,P11,P12,
   1P21,P22)
    Q11 = P11+Q11
    Q12 = P12+Q12
    P(1,1) = Q11
    P(1,2) = Q12
    P(2,1) = Q12
    P(2,2) = Q11
    IF (KA.NE.LA) GO TO 17
    GO TO 18
  17 P(1,1) = -Q12
    P(1,2) = -Q11
    P(2,1) = -Q11
    P(2,2) = -Q12
  18 C(MMM) = C(MMM)+FI*FJ*P(IS,JS)
  19 CONTINUE
C
C
    DO 23 I=1,N
    MM = (I-1)*N-(I*I-I)/2
    IJ = MM+I
    JJA = JA(I)
    J1 = JJA
    II2 = I2(I)
```

```
      II1 = I1(I)
      IF (II2.EQ.IB(J1)) J1=J1+NM
      JJB = JB(I)
      J2 = JJB
      IF (II2.EQ.IB(J2)) J2=J2+NM
      C(IJ) = C(IJ)+ZLD(J1)+ZLD(J2)
      JJJ = JJA
C
      DO 22 K=1,2
      NDJ = ND(JJJ)
C
      DO 21 JJ=1,NDJ
      J = MD(JJJ,JJ)
      IF (J.EQ.I) GO TO 21
      IF (I2(J).NE.II2) GO TO 21
      IJ = MM+J
      FI = 1.
      IF (K.EQ.2) GO TO 20
      IF (I1(J).NE.II1) FI=-1.
      C(IJ) = C(IJ)+FI*ZLD(J1)
      GO TO 21
   20 IF (I3(J).NE.I3(I)) FI=-1.
      C(IJ) = C(IJ)+FI*ZLD(J2)
   21 CONTINUE
C
   22 JJJ = JJB
C
   23 CONTINUE
C
      RETURN
C
   24 FORMAT (3X,'AM = ',E10.3,3X,'DMAX = ',E10.3,3X,'DMIN = ',E10.3)
   25 FORMAT (' WARNING **********************************************'/
     1,' THIS PROBLEM EXCEED LIMIT OF THIN WIRE CONDITION, THE RESULTS
     2 ARE NOT CORRECT')
      END
      SUBROUTINE SORT (IA,IB,I1,I2,I3,JA,JB,MD,ND,NM,NP,N,MAX,MIN,ICJ,IN
     1M)
      DIMENSION JSP(20)
      DIMENSION I1(1), I2(1), I3(1), JA(1), JB(1)
      DIMENSION IA(1), IB(1), ND(1), MD(INM,4)
      I = 0
C
      DO 3 K=1,NP
      NJK = 0
C
      DO 1 J=1,NM
C
C     SUBSTITUTE IND CALCULATION AND TEST
C      IND = (IA(J)-K)*(IB(J)-K)
C      IF (IND.NE.0) GO TO 1
      IF ((IA(J)*1.0 .NE. K*1.0) .AND. (IB(J)*1.0 .NE. K*1.0)) GO TO 1
      NJK = NJK+1
      JSP(NJK) = J
    1 CONTINUE
C
```

```
   MOD = NJK-1
   IF (MOD.LE.0) GO TO 3
C
   DO 2 IMD=1,MOD
   I = I+1
   IF (I.GT.ICJ) GO TO 2
   IPD = IMD+1
   JAI = JSP(IMD)
   JA(I) = JAI
   JBI = JSP(IPD)
   JB(I) = JBI
   I1(I) = IA(JAI)
   IF (IA(JAI).EQ.K) I1(I)=IB(JAI)
   I2(I) = K
   I3(I) = IA(JBI)
   IF (IA(JBI).EQ.K) I3(I)=IB(JBI)
  2 CONTINUE
C
  3 CONTINUE
C
   N = I
C
   DO 4 J=1,NM
   ND(J) = 0
C
   DO 4 K=1,4
  4 MD(J,K) = 0
C
   III = N
   IF (N.GT.ICJ) III = ICJ
C
   DO 8 I=1,III
   J = JA(I)
C
   DO 7 L=1,2
   ND(J) = ND(J)+1
   K = 1
   M = 0
  5 MJK = MD(J,K)
   IF (MJK.NE.0) GO TO 6
   M = 1
   MD(J,K) = I
  6 K = K+1
   IF (K.GT.4) GO TO 7
   IF (M.EQ.0) GO TO 5
  7 J = JB(I)
C
  8 CONTINUE
C
   MIN = 100
   MAX = 0
C
   DO 9 J=1,NM
   NDJ = ND(J)
   IF (NDJ.GT.MAX) MAX=NDJ
  9 IF (NDJ.LT.MIN) MIN=NDJ
```

```
C
      RETURN
      END
      SUBROUTINE SQROT (C,S,IWR,I12,NEQ)
      REAL*8 SNOR, SA, PH, CNOR
      COMPLEX*16 C(1),S(1),SS
      N = NEQ
      IF (I12.EQ.2) GO TO 6
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      C(1) = CDSQRT(C(1))
C
      DO 1 K=2,N
    1 C(K) = C(K)/C(1)
C
C
      DO 5 I=2,N
      IMO = I-1
      IPO = I+1
      ID = (I-1)*N-(I*I-I)/2
      II = ID+I
C
      DO 2 L=1,IMO
      LI = (L-1)*N-(L*L-L)/2+I
    2 C(II) = C(II)-C(LI)*C(LI)
C
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      C(II) = CDSQRT(C(II))
      IF (IPO.GT.N) GO TO 5
C
      DO 4 J=IPO,N
      IJ = ID+J
C
      DO 3 M=1,IMO
      MD = (M-1)*N-(M*M-M)/2
      MI = MD+I
      MJ = MD+J
    3 C(IJ) = C(IJ)-C(MJ)*C(MI)
C
    4 C(IJ) = C(IJ)/C(II)
C
    5 CONTINUE
C
    6 S(1) = S(1)/C(1)
C
      DO 8 I=2,N
      IMO = I-1
C
      DO 7 L=1,IMO
      LI = (L-1)*N-(L*L-L)/2+I
    7 S(I) = S(I)-C(LI)*S(L)
C
      II = (I-1)*N-(I*I-I)/2+I
    8 S(I) = S(I)/C(II)
C
```

```
      NN = ((N+1)*N)/2
      S(N) = S(N)/C(NN)
      NMO = N-1
C
      DO 10 I=1,NMO
      K = N-I
      KPO = K+1
      KD = (K-1)*N-(K*K-K)/2
C
      DO 9 L=KPO,N
      KL = KD+L
    9 S(K) = S(K)-C(KL)*S(L)
C
      KK = KD+K
   10 S(K) = S(K)/C(KK)
C
      IF (IWR.LE.0) GO TO 13
      CNOR = .0
C
      DO 11 I=1,N
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      SA = CDABS(S(I))
   11 IF (SA.GT.CNOR) CNOR=SA
C
      IF (CNOR.LE.0.) CNOR=1.
C
      DO 12 I=1,N
      SS = S(I)
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      SA = CDABS(SS)
      SNOR = SA/CNOR
      PH = .0
C
C          **** V3.2D FIXED FOLLOWING LINE FOR DOUBLE COMPLEX
      IF (SA.GT.0.) PH=57.29577951308232*DATAN2(DIMAG(SS),DBLE(SS))
   12 WRITE (6,14) I,SNOR,SA,PH,SS
C
      WRITE (6,15)
   13 RETURN
C
   14 FORMAT (1X,1I5,1F10.3,1F15.7,1F10.0,2F15.6)
   15 FORMAT (1H0)
      END
```