LAMPIRAN A

Program Lengkap Solusi Invers Tiga Dimensi dari Benda Bersimetri Sumbu di Ruang Tak Hingga dengan Menggunakan Elemen Batas Dengan Regularisasi Tikhonov

LAMPIRAN B

Program Lengkap Solusi Invers Tiga Dimensi dari Benda Bersimetri Sumbu di Ruang Tak Hingga dengan Menggunakan Elemen Batas Dengan Regularisasi GCV

```
C*********************************************************************
C**              INVERSE ACOUSTIC THREE DIMENSION             **]
C**              MAXIMUM NUMBER OF NODE (NMAX) = 150          **]
C**              MAXIMUM NUMBER OF ELEMEN (MMAX) = 69         **]
C*********************************************************************
C- ONE DIMENSION
      DIMENSION NIT(14),BTN(150),FPP(200),PFP(200),PHI(150),DPHI(150),
     1          EPHI(160),EPHIF(200)
C- TWO DIMENSION
      DIMENSION X(3,150),XCP(3,10),XFP(3,200),NODE(8,69),E1(8,112),
     1     E2(8,112),WFPNQQ(8,144),WFPNQT(8,112),A(160,150),B(160,150),
     2     C(200,150),D(200,150),AIB(300,300),AP(300,320)
C- THREE DIMENSION
      DIMENSION SHPQ(8,8,144),SHPQ1(8,8,144),SHPQ2(8,8,144),
     1          SHPT(8,8,112),SHPT1(8,8,112),SHPT2(8,8,112),
     2          WFPQQ(8,3,100),WFPQT(8,2,100)
C- FOUR DIMENSION
      DIMENSION SHAPQ(8,3,8,100),SHAPQ1(8,3,8,100),SHAPQ2(8,3,8,100),
     1          SHAPT(8,2,8,100),SHAPT1(8,2,8,100),SHAPT2(8,2,8,100)
C-
      INTEGER*2 NDSKP(69,150),NDSKPC(69,10),NDSKPF(69,200),NDSS(2,69),
     1          KKK(8,69),KKMAX(6)
      INTEGER PSYM
      COMPLEX A,B,BTN,C,D,PFP,FPP,EPHI,EPHIF,PHI,DPHI
      COMMON/OMEGA/WN
      COMMON/MAT/EPS,SVDQ,MSVD
      COMMON/IDEN/IDD
      COMMON/SYM/PSYM
      COMMON/SCAT/ISC,AF,BE,GA
      COMMON/CGAUS/Y(12,12),W(12,12),NGTHET,NGRHO
C=====================================================================
C     MAIN PROGRAM
C=====================================================================
      NMAX = 150
      MMAX = 69
      OPEN (5,FILE='INPUT.TXT',STATUS='OLD')
      OPEN (6,FILE='AKHIR.TXT',STATUS='NEW')
C--
cc input : nmax,mmax
cc output: N,M,X,NODE,XCP,XFP,NCP,NFP,FPP,NPCH,ISEVCP dan ISEVFP
      CALL INPDAT (NMAX,N,MMAX,M,X,NODE,XCP,XFP,NCP,NFP,FPP,
     1            NPCH,ISEVCP,ISEVFP)
C--
cc input nggak ada
cc output Y(12,12) dan W(12,12)
C--
      CALL GAUS (Y,W)
C--
cc input : nggak ada
cc output: E1(8,112),E2(8,112),WFPNQT(8,112),NIT(14)
C--
      CALL FAIR(E1,E2,WFPNQT,NIT)
C--
cc input : M,N,NCP,NFP,NODE,X,XCP,XFP,ISEVCP dan ISEVFP
cc output: NDSKP,NDSKPC,NDSKPF,NDSS,KKK dan KKMAX
C--
      CALL DEGSEV (M,N,NCP,NFP,NDSKP,NDSKPC,NDSKPF,NDSS,NODE,KKK,
     1     KKMAX,X,XCP,XFP,ISEVCP,ISEVFP)
C--
cc input : NIT(14),E1,E2
cc output: SHPQ,SHPQ1,SHPQ2,SHPT,SHPT1,SHPT2,WFPNQQ
C--
               CALL CHAPE(SHPQ,SHPQ1,SHPQ2,SHPT,SHPT1,SHPT2,NIT,E1,E2,WFPNQQ)
C--
cc input : N,NPCH,M,X,XCP,XFP,NCP,NFP,NODE,SHPQ,SHPQ1,SHPQ2,
cc         SHPT,SHPT1,SHPT2,WFPNQQ,NIT,SHAPQ,SHAPQ1,SHAPQ2,
cc         SHAPT,SHAPT1,SHAPT2,WFPQQ,WFPQT,WFPNQT,NDSKP,
cc         NDSKPC,NDSKPF,KKK,KKMAX,FPP,EPHI,EPHIF
cc output: A,B,BTN,C,D,PFP
C--
      CALL COEF(N,NPCH,M,A,B,BTN,C,D,PFP,X,XCP,XFP,NCP,NFP,NODE,SHPQ,
```

```
     1        SHPQ1,SHPQ2,SHPT,SHPT1,SHPT2,NIT,SHAPQ,SHAPQ1,SHAPQ2,
     2        SHAPT,SHAPT1,SHAPT2,WFPQQ,WFPQT,WFPNQQ,WFPNQT,NDSKP,
     3        NDSKPC,NDSKPF,KKK,KKMAX,FPP,EPHI,EPHIF)
C--
      NPCT2 = NPCH * 2
      NFPT2 = NFP * 2
      NT2 = N * 2
C--
cc input : N,NPCH,NFPT2,NPCT2,NT2,A,B,BTN,C,D,PFP
cc output: PHI,DPHI,AIB,AP
C--
      CALL SOLVE (A,B,BTN,C,D,PFP,NFP,NPCH,N,NFPT2,NPCT2,NT2,
     1           PHI,DPHI,AIB,AP)
C--
      END
C=======================================================================
      SUBROUTINE INPDAT (NMAX,N,MMAX,M,X,NODE,XCP,XFP,NCP,NFP,FPP,
     1                   NPCH,ISEVCP,ISEVFP)
C=======================================================================
C-----
C-- TO READ AND OUTPUT THE DATA INPUT FILE
C-----
C- ONE DIMENSION
      DIMENSION XD(3),NODER(8),FPP(200)
C- TWO DIMENSION
      DIMENSION X(3,NMAX),XCP(3,10),XFP(3,200),NODE(8,MMAX)
      CHARACTER*4 DATMOD
      CHARACTER*40 TITLE
      INTEGER PSYM
      COMPLEX FPP
      COMMON/OMEGA/WN
      COMMON/MAT/EPS,SVDQ,MSVD
      COMMON/IDEN/IDD
      COMMON/SYM/PSYM
      COMMON/SCAT/ISC,AF,BE,GA
      WRITE (6,50)
   50 FORMAT (/72(1H=)/12X,'I N P U T   D A T A   F I L E',/
     1       72(1H=))
  100 READ (5,110) DATMOD
  110 FORMAT (A4,15X)
      IF (DATMOD.EQ.'TITL') GO TO 1000
      IF (DATMOD.EQ.'EXTE'.OR.DATMOD.EQ.'INTE') GO TO 1100
      IF (DATMOD.EQ.'PLAN'.OR.DATMOD.EQ.'POIN') GO TO 1200
      IF (DATMOD.EQ.'PARA') GO TO 1250
      IF (DATMOD.EQ.'FREQ') GO TO 1300
      IF (DATMOD.EQ.'PROP') GO TO 1400
      IF (DATMOD.EQ.'NODE') GO TO 1500
C     IF (DATMOD.EQ.'TRUE') GO TO 1600
      IF (DATMOD.EQ.'ELEM') GO TO 1700
      IF (DATMOD.EQ.'CHIE') GO TO 1800
      IF (DATMOD.EQ.'FIEL') GO TO 1900
      IF (DATMOD.EQ.'END') GO TO 2000
C-----
C- READ TITLE
C-----
 1000 READ (5,1010) TITLE
 1010 FORMAT (A40)
      WRITE (6,1020) TITLE
 1020 FORMAT (//2X,10(1H-),' TITLE ',10(1H-)/2X,A40)
      GO TO 100
C-----
C- IDENTIFY THE PROBLEM
C-----
 1100 ISC = 0
      PSYM = 0
      IF (DATMOD.EQ.'INTE') GO TO 1120
      WRITE (6,1110)
 1110 FORMAT (//2X,'***** EXTERIOR PROBLEM *****')
      GO TO 1140
 1120 WRITE (6,1130)
 1130 FORMAT (//2X,'***** INTERIOR PROBLEM *****')
```

```
 1140 IF (PSYM.EQ.0) GO TO 1180
      IF (PSYM.GT.0.AND.PSYM.LE.3) GO TO 1160
      WRITE (6,1150)
 1150 FORMAT (//2X,'!!!!! ERROR --- PSYM > 3 !!!!!',/2X)
      STOP
 1160 WRITE (6,1170) PSYM
 1170 FORMAT (2X,'USING SIMETRY',/2X,
     1      'PLANE OF SIMETRY PERPENDICULAR TO X(',I1,') AXIS')
 1180 GO TO 100
C-----
C- SPECIFY THE INCIDENT WAVE IN SCATTERING PROBLEM
C- ISC = 1 : PLANE WAVE; AF,BE,GA : THE DIRECTION COSINES
C- ISC = 2 : SPHERICAL WAVE; AF,BE,GA : LOCATION OF THE POINT SOURCE
C-----
 1200 ISC = 1
      IF (DATMOD.EQ.'POIN') ISC = 2
      READ (5,*) AF,BE,GA
      GO TO 100
C------
C-- TO READ PARAMETER
C-----
 1250 READ (5,*) EPS,SVDQ,MSVD
      GO TO 100
C-----
C- PRESCRIBE   THE FREQUENCY RANGE
C-----
 1300 READ (5,*) F,IDD
      IF (IDD.EQ.1) GO TO 1340
      IF (ISC.EQ.1.OR.ISC.EQ.2) GO TO 1320
      WRITE (6,1310)
 1310 FORMAT (/2X'***** RADIATION PROBLEM *****')
      GO TO 1340
 1320 WRITE (6,1330)
 1330 FORMAT (/2X'***** SCATTERING PROBLEM *****')
 1340 WRITE (6,1350) F
 1350 FORMAT (/2X,'FREQUENCY         : ',F10.4)
      GO TO 100
C-----
C- SPECIFY THE PROPERTY OF AIR
C-----
 1400 READ (5,*) SPEED, DENS, PREF
      WRITE (6,1410) SPEED
 1410 FORMAT (2X,'SPEED OF SOUND    : ',F10.4)
      PI = 3.141593
      WN = 2 * PI * F / SPEED
      WRITE (6,1420) WN
 1420 FORMAT (2X,'WAVE NUMBER       : ',F10.5)
      WRITE (6,1430) DENS
 1430 FORMAT (2X,'DENSITY           : ',F10.4)
      WRITE (6,1440) PREF
 1440 FORMAT (2X,'REFERENCE PRESSURE: ',F10.5)
      GO TO 100
C-----
C- READ THE NODAL COORDINATES
C-----
 1500 WRITE (6,1510)
 1510 FORMAT (//2X,10(1H-),' NODAL COORDINATES ',10(1H-),
     1       //2X,'NODE',7X,'X(1)',9X,'X(2)',9X,'X(3)')
      I = 0
 1520 I = I + 1
      READ (5,*) IN, (XD(J),J=1,3)
      IF (IN.EQ.0) GO TO 1560
      IF (IN.LE.NMAX.AND.IN.EQ.I) GO TO 1540
      WRITE (6,1530) I,IN
 1530 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'NODE NUMBER (',
     1       I3,') IMPROPERLY S[ECIFIED (',I3,')')
      STOP
 1540 X(1,IN) = XD(1)
      RMAX = 0
      R = ABS(X(1,IN))
      IF (R.GT.RMAX) RMAX = R
```

```
      X(2,IN) = XD(2)
      RMAX = 0
      R = ABS(X(2,IN))
      IF (R.GT.RMAX) RMAX = R
      X(3,IN) = XD(3)
      RMAX = 0
      R = ABS(X(3,IN))
      IF (R.GT.RMAX) RMAX = R
 1550 GO TO 1520
      RRMAX = 1./RMAX
 1560 N = I - 1
      DO 1590 IN = 1,N
      DO 1570 J = 1,3
      IF (ABS(X(J,IN))/RMAX.LT.1.E-4) X(J,IN) = 0
 1570 CONTINUE
      WRITE (6,1580) IN, (X(J,IN),J=1,3)
 1580 FORMAT (3X,I3,1X,3(3X,F10.5))
 1590 CONTINUE
      GO TO 100
C-----
C- READ TRUE VALUES OF NODAL PHI AND DPHI
C-----
C-
C-
C-----
C- READ THE ELEMENT DATA
C-----
 1700 WRITE (6,1710)
 1710 FORMAT (//2X,10(1H-),' ELEMENT DATA  ',10(1H-),//2X/
     1   'ELEMENT',3X,'NODE1',2X,'NODE2',2X,'NODE3',2X,
     2   'NODE4',2X,'NODE5',2X,'NODE6',2X,'NODE7',2X,'NODE8')
      J = 0
 1720 J = J + 1
      READ (5,*) K, (NODER(I),I=1,8)
      IF(K.EQ.0) GO TO 1750
      IF(K.LE.MMAX.AND.K.EQ.J) GO TO 1740
      WRITE (6,1730) J,K
 1730 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,
     1 'ELEMENT NUMBER (',I3,') IMPROPERLY SPECIFIED (',I3,')')
      STOP
 1740 NODE(1,K) = NODER(1)
      NODE(2,K) = NODER(2)
      NODE(3,K) = NODER(3)
      NODE(4,K) = NODER(4)
      NODE(5,K) = NODER(5)
      NODE(6,K) = NODER(6)
      NODE(7,K) = NODER(7)
      NODE(8,K) = NODER(8)
 1745 GO TO 1720
 1750 M = J - 1
      DO 1790 K = 1,M
      DO 1770 I = 1,8
      IF (NODE(I,K).GT.0.AND.NODE(I,K).LE.N) GO TO 1770
      WRITE (6,1760) I,K,NODE(I,K)
 1760 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,
     1 'NODE(',I1,',',I3,') IMPROPERLY SPECIFIED  : ',I5)
      STOP
 1770 CONTINUE
      WRITE (6,1780) K, (NODE(I,K),I=1,8)
 1780 FORMAT (4X,I3,1X,8(4X,I3))
 1790 CONTINUE
      GO TO 100
C-----
C- READ THE CHIEF POINT COORDINATES
C-----
 1800 WRITE (6,1810)
 1810 FORMAT (//2X,10(1H-),' CHIEF POINT ',10(1H-),//2X,'POINT',
     1  6X,'X(1)',9X,'X(2)',9X,'X(3)')
      NCMAX = 10
      I = 0
 1820 I = I + 1
```

```
      READ (5,*) IC,(XD(J),J=1,3)
      IF (IC.EQ.0) GO TO 1860
      IF (IC.LE.NCMAX.AND.IC.EQ.I) GO TO 1840
      WRITE (6,1830) I,IC
 1830 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'CHIEF POINT',
     1  ' NUMBER (',I3,') IMPROPERLY SPECIFIED (',I3,')')
      STOP
 1840 XCP(1,IC) = XD(1)
      XCP(2,IC) = XD(2)
      XCP(3,IC) = XD(3)
 1850 GO TO 1820
 1860 NCP = I - 1
      ISEVCP = XD(1)
      DO 1880 IC = 1,NCP
      WRITE (6,1870) IC,(XCP(J,IC),J=1,3)
 1870 FORMAT(3X,I3,1X,3(3X,F10.5))
 1880 CONTINUE
      IF (ISEVCP.EQ.0) GO TO 1894
      IF (ISEVCP.GT.0.AND.ISEVCP.LE.8) GO TO 1890
      WRITE (6,1882) ISEVCP
 1882 FORMAT(//12X,'!!!!! E R R O R !!!!!',/3X,'IMPROPER DEGREE',
     1       ' OF SEVERITY SPECIFIED FOR CHIEF POINT  : ',I2)
      STOP
 1890 WRITE (6,1892) ISEVCP
 1892 FORMAT(/2X,'USING NUMBER OF SEVERITY : ',I2,
     1       ' FOR ALL CHIEF POINT')
 1894 GO TO 100
C-----
C- READ THE FIELD POINT COORDINATES AND THE KNOWN PRESSURE
C-----
 1900 NPCH = N + NCP
      WRITE (6,1910)
 1910 FORMAT (//2X,10(1H-),' FIELD POINT ',10(1H-),//63X,'PRESSURE',
     1        /2X,'POINT',9X,'X(1)',10X,'X(2)',10X,'X(3)',10X,'REAL',
     2        7X,'IMAGINER')
      NFMAX = 200
      I = 0
 1920 I = I + 1
      READ (5,*) IF,(XD(J),J=1,3),PRE,PIM
      IF (IF.EQ.0) GO TO 1960
      IF (IF.LE.NFMAX.AND.IF.EQ.I) GO TO 1940
      WRITE (6,1930) I,IF
 1930 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'FIELD POINT',
     1  ' NUMBER (',I3,') IMPROPERLY SPECIFIED (',I3,')')
      STOP
 1940 XFP(1,IF) = XD(1)
      XFP(2,IF) = XD(2)
      XFP(3,IF) = XD(3)
 1950 FPP(IF) = CMPLX(PRE,PIM)
      GO TO 1920
 1960 NFP = I - 1
      ISEVFP = XD(1)
      DO 1980 IF = 1,NFP
      WRITE (6,1970) IF, (XFP(J,IF),J=1,3), FPP(IF)
 1970 FORMAT (3X,I3,1X,5(3X,F11.5))
 1980 CONTINUE
      IF (ISEVFP.EQ.0) GO TO 1994
      IF (ISEVFP.GT.0.AND.ISEVFP.LE.8) GO TO 1990
      WRITE (6,1982) ISEVFP
 1982 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'IMPROPER DEGREE',
     1       ' OF SEVERITY SPECIFIED FOR FIELD POINT : ',I2)
      STOP
 1990 WRITE (6,1992) ISEVFP
 1992 FORMAT(/2X,'USING DEGREE OF SEVERITY : ',I2,
     1       ' FOR ALL FIELD POINT')
 1994 IF (NFP.GE.N) GO TO 1998
      WRITE (6,1997)
 1997 FORMAT (//2X,'!!!!! ERROR --- ',
     1       ' TOO SEDIKIT FIELD POINT SPECIFIED !!!!!',/2X)
      STOP
 1998 GO TO 100
```

```
C-----
C- BACK TO MAIN PROGRAM
C-----
 2000 WRITE (6,2010)
 2010 FORMAT (/72(1H=)/12X,'END OF INPUT DATA FILE',/
     1       72(1H=))
      RETURN
      END
C=======================================================================
      SUBROUTINE GAUS (Y,W)
C=======================================================================
C-----
C-- TO READ THE GAUSSIAN QUADRATURE INTEGRATION POINT &
C-- THE WEIGHTING FUNCTION.
C-----
      DIMENSION Y(12,12),W(12,12)
      Y(2,1) = -.5773503
      Y(2,2) = -Y(2,1)
      W(2,1) = .5
      W(2,2) = .5
      Y(3,1) = W(2,1)
      Y(3,2) = 0
      Y(3,3) = -Y(3,1)
      W(3,1) = .55555556/2
      W(3,2) = .88888889/2
      W(3,3) = W(3,1)
      Y(4,1) = -.8611363
      Y(4,2) = -.3399810
      Y(4,3) = -Y(4,2)
      Y(4,4) = -Y(4,1)
      W(4,1) = .1739274
      W(4,2) = .3260726
      W(4,3) = W(4,2)
      W(4,4) = W(4,1)
      Y(5,1) = -.9061798
      Y(5,2) = -.5384693
      Y(5,3) = 0
      Y(5,4) = -Y(5,2)
      Y(5,5) = -Y(5,1)
      W(5,1) = .1184634
      W(5,2) = .2393143
      W(5,3) = .2844444
      W(5,4) = W(5,2)
      W(5,5) = W(5,1)
      Y(6,1) = -.9324695
      Y(6,2) = -.6612094
      Y(6,3) = -.2386192
      Y(6,4) = -Y(6,3)
      Y(6,5) = -Y(6,2)
      Y(6,6) = -Y(6,1)
      W(6,1) = .17132449/2
      W(6,2) = .36076157/2
      W(6,3) = .46791393/2
      W(6,4) = W(6,3)
      W(6,5) = W(6,2)
      W(6,6) = W(6,1)
      Y(8,1) = 0.5*(Y(4,1) - 1.0)
      Y(8,2) = 0.5*(Y(4,2) - 1.0)
      Y(8,3) = 0.5*(Y(4,3) - 1.0)
      Y(8,4) = 0.5*(Y(4,4) - 1.0)
      Y(8,5) = -Y(8,4)
      Y(8,6) = -Y(8,3)
      Y(8,7) = -Y(8,2)
      Y(8,8) = -Y(8,1)
      W(8,1) = W(4,1)*0.5
      W(8,2) = W(4,2)*0.5
      W(8,3) = W(4,3)*0.5
      W(8,4) = W(4,4)*0.5
      W(8,5) = W(8,4)
      W(8,6) = W(8,3)
      W(8,7) = W(8,2)
```

```
      W(8,8) = W(8,1)
      Y(10,1) = 0.5*(Y(5,1) - 1.0)
      Y(10,2) = 0.5*(Y(5,2) - 1.0)
      Y(10,3) = 0.5*(Y(5,3) - 1.0)
      Y(10,4) = 0.5*(Y(5,4) - 1.0)
      Y(10,5) = 0.5*(Y(5,5) - 1.0)
      Y(10,6) = -Y(10,5)
      Y(10,7) = -Y(10,4)
      Y(10,8) = -Y(10,3)
      Y(10,9) = -Y(10,2)
      Y(10,10) = -Y(10,1)
      W(10,1) = W(5,1)*0.5
      W(10,2) = W(5,2)*0.5
      W(10,3) = W(5,3)*0.5
      W(10,4) = W(5,4)*0.5
      W(10,5) = W(5,5)*0.5
      W(10,6) = W(10,5)
      W(10,7) = W(10,4)
      W(10,8) = W(10,3)
      W(10,9) = W(10,2)
      W(10,10) = W(10,1)
      Y(12,1) = 0.5*(Y(6,1) - 1.0)
      Y(12,2) = 0.5*(Y(6,2) - 1.0)
      Y(12,3) = 0.5*(Y(6,3) - 1.0)
      Y(12,4) = 0.5*(Y(6,4) - 1.0)
      Y(12,5) = 0.5*(Y(6,5) - 1.0)
      Y(12,6) = 0.5*(Y(6,6) - 1.0)
      Y(12,7) = -Y(12,6)
      Y(12,8) = -Y(12,5)
      Y(12,9) = -Y(12,4)
      Y(12,10) = -Y(12,3)
      Y(12,11) = -Y(12,2)
      Y(12,12) = -Y(12,1)
      W(12,1) = W(6,1)*0.5
      W(12,2) = W(6,2)*0.5
      W(12,3) = W(6,3)*0.5
      W(12,4) = W(6,4)*0.5
      W(12,5) = W(6,5)*0.5
      W(12,6) = W(6,6)*0.5
      W(12,7) = W(12,6)
      W(12,8) = W(12,5)
      W(12,9) = W(12,4)
      W(12,10) = W(12,3)
      W(12,11) = W(12,2)
      W(12,12) = W(12,1)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE FAIR(E1,E2,WFPNQT,NIT)
C=======================================================================
C-----
C-- TO READ THE GAUSSIAN QUADRATURE INTEGRATION POINT &
C-- THE WEIGHTING FUNCTION FOR TRIANGULAR ELEMENT.
C-----
      DIMENSION E1(8,112),E2(8,112),WFPNQT(8,112),NIT(14)
C-----
     1 NIT(1)=0
     2 NIT(2)=0
     3 NIT(3)=0
     4 NIT(4)=6
       E1(1,1)=.9157621/10.
       E2(1,1)=E1(1,1)
       E1(1,2)=E1(1,1)
       E2(1,2)=.8168476
       E1(1,3)=E2(1,2)
       E2(1,3)=E1(1,1)
       E1(1,4)=.4459485
       E2(1,4)=E1(1,4)
       E1(1,5)=E1(1,4)
       E2(1,5)=.1081030
```

```
   E1(1,6)=E2(1,5)
   E2(1,6)=E1(1,4)
   DO 41 IC=1,3
41 WFPNQT(1,IC)=.6348067/10
   DO 42 IC=4,6
42 WFPNQT(1,IC)=.1289694
 5 NIT(5)=7
   E1(2,1)=.3333333
   E2(2,1)=E1(2,1)
   E1(2,2)=.1012865
   E2(2,2)=E1(2,1)
   E1(2,3)=E1(2,2)
   E2(2,3)=.7974270
   E1(2,4)=E2(2,3)
   E2(2,4)=E1(2,2)
   E1(2,5)=.4701421
   E2(2,5)=E1(2,5)
   E1(2,6)=E1(2,5)
   E2(2,6)=.5971587/10.
   E1(2,7)=E2(2,6)
   E2(2,7)=E1(2,5)
   WFPNQT(2,1)=.1299038
   DO 51 IC=2,4
51 WFPNQT(2,IC)=.7271102/10.
   DO 52 IC=5,7
52 WFPNQT(2,IC)=.7643780/10.
 6 NIT(6)=0
 7 NIT(7)=0
 8 NIT(8)=16
   E1(3,1)=.3333333
   E2(3,1)=E1(3,1)
   E1(3,2)=.4592926
   E2(3,2)=E1(3,2)
   E1(3,3)=E1(3,2)
   E2(3,3)=.8141482/10.
   E1(3,4)=E2(3,3)
   E2(3,4)=E1(3,2)
   E1(3,5)=.5054723/10.
   E2(3,5)=E1(3,5)
   E1(3,6)=E1(3,5)
   E2(3,6)=.8989055
   E1(3,7)=E2(3,6)
   E2(3,7)=E1(3,5)
   E1(3,8)=.1705693
   E2(3,8)=E1(3,8)
   E1(3,9)=E1(3,8)
   E2(3,9)=.6588614
   E1(3,10)=E2(3,9)
   E2(3,10)=E1(3,8)
   E1(3,11)=.7284924
   E2(3,11)=.2631128
   E1(3,12)=E2(3,11)
   E2(3,12)=E1(3,11)
   E1(3,13)=.8394777/100.
   E2(3,13)=E2(3,11)
   E1(3,14)=E2(3,11)
   E2(3,14)=E1(3,13)
   E1(3,15)=E1(3,13)
   E2(3,15)=E1(3,11)
   E1(3,16)=E1(3,11)
   E2(3,16)=E1(3,13)
   WFPNQT(3,1)=.8332066/10.
   DO 81 IC=2,4
81 WFPNQT(3,IC)=.5490118/10.
   DO 82 IC=5,7
82 WFPNQT(3,IC)=.1873992/10.
   DO 83 IC=8,10
83 WFPNQT(3,IC)=.5959258/10.
   DO 84 IC=11,16
84 WFPNQT(3,IC)=.1572143/10
 9 NIT(9)=19
```

```
      E1(4,1)=.3333333
      E2(4,1)=.3333333
      E1(4,2)=.4896825
      E2(4,2)=.4896825
      E1(4,3)=.4896825
      E2(4,3)=.02063496
      E1(4,4)=.02063496
      E2(4,4)=E1(4,2)
      E1(4,5)=.4370896
      E2(4,5)=.4370896
      E1(4,6)=.4370896
      E2(4,6)=.1258208
      E1(4,7)=.1258208
      E2(4,7)=E1(4,5)
      E1(4,8)=.1882035
      E2(4,8)=.1882035
      E1(4,9)=.1882035
      E2(4,9)=.6235929
      E1(4,10)=E2(4,9)
      E2(4,10)=E1(4,9)
      E1(4,11)=.04472951
      E2(4,11)=.04472951
      E1(4,12)=.04472951
      E2(4,12)=.9105410
      E1(4,13)=.9105410
      E2(4,13)=E1(4,11)
      E1(4,14)=.7411986
      E2(4,14)=.2219630
      E1(4,15)=.2219630
      E2(4,15)=E1(4,14)
      E1(4,16)=.03683841
      E2(4,16)=E2(4,14)
      E1(4,17)=E2(4,14)
      E2(4,17)=E1(4,16)
      E1(4,18)=E1(4,16)
      E2(4,18)=E1(4,14)
      E1(4,19)=E1(4,14)
      E2(4,19)=E1(4,16)
      WFPNQT(4,1)=.05608138
      DO 91 IC=2,4
   91 WFPNQT(4,IC)=.01809110
      DO 92 IC=5,7
   92 WFPNQT(4,IC)=.04493375
      DO 93 IC=8,10
   93 WFPNQT(4,IC)=.04598464
      DO 94 IC=11,13
   94 WFPNQT(4,IC)=.01476728
      DO 95 IC=14,19
   95 WFPNQT(4,IC)=.02498976
      NIT(10)=0
   11 NIT(11)=28
      E1(5,1)=.3333333
      E2(5,1)=.3333333
      E1(5,2)=.02598914
      E2(5,2)=.02598914
      E1(5,3)=.02598914
      E2(5,3)=.9480217
      E1(5,4)=.9480217
      E2(5,4)=E1(5,2)
      E1(5,5)=.09428750
      E2(5,5)=.09428750
      E1(5,6)=.09428752
      E2(5,6)=.8114250
      E1(5,7)=.8114250
      E2(5,7)=E1(5,5)
      E1(5,8)=.4946368
      E2(5,8)=.4946368
      E1(5,9)=.4946368
      E2(5,9)=.0107264
      E1(5,10)=E2(5,9)
      E2(5,10)=E1(5,8)
```

```
      E1(5,11)=.2073434
      E2(5,11)=.2073434
      E1(5,12)=.2073434
      E2(5,12)=.5853132
      E1(5,13)=.5853132
      E2(5,13)=E1(5,11)
      E1(5,14)=.4389078
      E2(5,14)=.4389078
      E1(5,15)=.4389078
      E2(5,15)=.1221844
      E1(5,16)=.1221844
      E2(5,16)=E1(5,14)
      E1(5,17)=.8588703
      E2(5,17)=.1411297
      E1(5,18)=.1411297
      E2(5,18)=E1(5,17)
      E1(5,19)=0.0
      E2(5,19)=E2(5,17)
      E1(5,20)=E2(5,17)
      E2(5,20)=0.0
      E1(5,21)=0.0
      E2(5,21)=E1(5,17)
      E1(5,22)=E1(5,17)
      E2(5,22)=0.0
      E1(5,23)=.6779377
      E2(5,23)=.2772206
      E1(5,24)=.2772206
      E2(5,24)=E1(5,23)
      E1(5,25)=.04484168
      E2(5,25)=E2(5,23)
      E1(5,26)=E2(5,23)
      E2(5,26)=E1(5,25)
      E1(5,27)=E1(5,25)
      E2(5,27)=E1(5,23)
      E1(5,28)=E1(5,23)
      E2(5,28)=E1(5,25)
      WFPNQT(5,1)=.05079372
      DO 111 IC=2,4
  111 WFPNQT(5,IC)=.005048531
      DO 112 IC=5,7
  112 WFPNQT(5,IC)=.02198641
      DO 113 IC=8,10
  113 WFPNQT(5,IC)=.01088620
      DO 114 IC=11,13
  114 WFPNQT(5,IC)=.04166142
      DO 115 IC=14,16
  115 WFPNQT(5,IC)=.04002720
      DO 116 IC=17,22
  116 WFPNQT(5,IC)=.004250674
      DO 117 IC=23,28
  117 WFPNQT(5,IC)=.02370388
      NIT(12) = 64
      NIT(13) = 76
      NIT(14) = 112

            DO 155 IJ=6,8

            II = IJ - 3
      IF (IJ.EQ.6) IK=16
      IF (IJ.EQ.7) IK=19
      IF (IJ.EQ.8) IK=28

            DO 120 IL=1,IK
      E1(IJ,IL) = 0.5*E1(II,IL)
      E2(IJ,IL) = 0.5*E2(II,IL)
      WFPNQT(IJ,IL) = WFPNQT(II,IL)/4.0
      E1(IJ,IL) = 0.5*E1(II,IL)
      E2(IJ,IL) = 0.5*E2(II,IL)
      WFPNQT(IJ,IL) = WFPNQT(II,IL)/4.0
  120 CONTINUE
```

```
            IK1 = IK + 1
      IK2 = IK*2

            DO 130 IL=IK1,IK2
      JJ = IL - IK
      E1(IJ,IL) = 0.5*(1.0 + E1(II,JJ))
      E2(IJ,IL) = 0.5*E2(II,JJ)
      WFPNQT(IJ,IL) = WFPNQT(II,JJ)/4.0
  130 CONTINUE

            IK3 = IK2 + 1
      IK4 = IK2 + IK

            DO 140 IL=IK3,IK4
      JJ = IL - IK2
      E1(IJ,IL) = 0.5*E1(II,JJ)
      E2(IJ,IL) = 0.5*(1.0 + E2(II,JJ))
      WFPNQT(IJ,IL) = WFPNQT(II,JJ)/4.0
  140 CONTINUE

            IK5 = IK4 + 1
      IK6 = IK4 + IK

            DO 150 IL=IK5,IK6
      JJ = IL - IK4
      E1(IJ,IL) = 0.5*(1.0 - E1(II,JJ))
      E2(IJ,IL) = 0.5*(1.0 - E2(II,JJ))
      WFPNQT(IJ,IL) = WFPNQT(II,JJ)/4.0
  150 CONTINUE

  155 CONTINUE
      RETURN
      END
C=======================================================================
      SUBROUTINE DEGSEV (M,N,NCP,NFP,NDSKP,NDSKPC,NDSKPF,NDSS,NODE,KKK,
     1                KKMAX,X,XCP,XFP,ISEVCP,ISEVFP)
C=======================================================================
C-----
C-- TO DETERMINE THE DEGREE OF SEVERITY
C-- THIS VERSIION OF DSEV IS COMPATIBLE WITH THE VERSION OF NMDS WHICH
C-- USES THE NEW ALGORITHM FOR SEVERITY.
C-----
      DIMENSION NODE(8,M),X(3,N),XCP(3,NCP),XFP(3,NFP)
      INTEGER*2 NDSKP(M,N),NDSKPC(M,NCP),NDSKPF(M,NFP),NDSS(2,M),
     1         KKK(8,M),KKMAX(6)
C-----
C-- NDSKP(K,L)       DEGREE OF SEVERITY ON SEGMENT K WITH P AT NODE L
C-- NDSKPC(K,LC)  DEGREE OF SEVERITY ON SEGMENT K WITH P AT CHIEF
C--              POINT LC
C-- NDSKPF(K,LF)  DEGREE OF SEVERITY ON SEGMENT K WITH P AT FIELD
C--              POINT LF
C-- NDSS(2,K)     (IF NOT ZERO) DEGREE OF SEVERITY ON SEGMENT K
C--              FOR ALL P OF THE P.EQ.Q CASES
C-----
C--

      CALL NMDS (M,N,NCP,NFP,NDSKP,NDSKPC,NDSKPF,NDSS,X,XCP,XFP,
     1          ISEVCP,ISEVFP,NODE)
C--
      WRITE(6,1011)
 1011 FORMAT (//2X,10(1H-),' DEGREE OF SEVERITY ',10(1H-),//)
      WRITE(6,1021) (J,J=1,5)
 1021 FORMAT(16X,5(9X,I1))
      WRITE(6,1025)
 1025 FORMAT(16X,'1234567890123456789012345678901234567890123456789012345678901234567890')
      WRITE(6,1030)
 1030 FORMAT (/2X,'ELEMENT',2X,'P=Q',10X,
     1        'P NODE NUMBER, P.NE.Q INTEGRATION',8X/)

            KK1 =0
      KK2 =0
```

```
      KK3 =0
      KK4 =0
      KK5 =0
      KK6 =0

      DO 90 K=1,M
      MDS =NDSS(2,K)
      MD = MDS - 2
      GO TO (5501,5502,5503,5504,5505,5506),MD
 5501 KK1 = KK1 + 1
      KKK(MDS,KK1) = K
      GO TO 53
 5502 KK2 = KK2 + 1
      KKK(MDS,KK2) = K
      GO TO 53
 5503 KK3 = KK3 + 1
      KKK(MDS,KK3) = K
      GO TO 53
 5504 KK4 = KK4 + 1
      KKK(MDS,KK4) = K
      GO TO 53
 5505 KK5 = KK5 + 1
      KKK(MDS,KK5) = K
      GO TO 53
 5506 KK6 = KK6 + 1
      KKK(MDS,KK6) = K
   53 CONTINUE
C-----
C-- PRINT DEGREES OF SEVERITY FOR NODAL POINT
C-----
      LN = INT (N/50)
      IF (LN.LT.1) GO TO 64
      DO 63 J = 1,LN
      IF (J.NE.1) GO TO 62
      WRITE (6,1040) K,MDS,(NDSKP(K,L),L=1,50)
 1040 FORMAT (4X,I3,5X,I1,3X,50I1)
      GO TO 63
   62 WRITE(6,1045) (NDSKP(K,L),L=(J-1)*50+1,J*50)
 1045 FORMAT(16X,50I1)
   63 CONTINUE
      IF ((N-LN*50).NE.0) WRITE(6,1045) (NDSKP(K,L),L=LN*50+1,N)
      GO TO 90
   64 WRITE(6,1040) K,MDS,(NDSKP(K,L),L=1,N)
   90 CONTINUE
C-----
C-- END OF K LOOP : ELEMENT LOOP
C-----
      KKMAX(1) = KK1
      KKMAX(2) = KK2
      KKMAX(3) = KK3
      KKMAX(4) = KK4
      KKMAX(5) = KK5
      KKMAX(6) = KK6
C-----
C-- PRINT DEGREES OF SEVERITY FOR CHIEF POINT
C-----
      IF (NCP.EQ.0.OR.ISEVCP.NE.0) GO TO 200
      WRITE (6,1050)
 1050 FORMAT(/31X,'P CHIEF POINT NUMBER',15X/)
      DO 190 K=1,M
      LNC = INT (NCP/50)
      IF (LNC.LT.1) GO TO 164
      DO 163 J = 1,LNC
      IF (J.NE.1) GO TO 162
      WRITE (6,1042) K,(NDSKPC(K,LC),LC=1,50)
 1042 FORMAT(4X,I3,9X,50I1)
      GO TO 163
  162 WRITE(6,1045) (NDSKPC(K,LC),LC=(J-1)*50+1,J*50)
c 1045 FORMAT(16X,50I1)
  163 CONTINUE
      IF((NCP-LNC*50).NE.0) WRITE(6,1045) (NDSKPC(K,LC),LC=LNC*50+1,NCP)
```

```
      GO TO 190
  164 WRITE(6,1042) K, (NDSKPC(K,LC),LC=1,NCP)
  190 CONTINUE
C-----
C-- 190 END OF K LOOP : ELEMENT LOOP
C-----
C
C-----
C-- PRINT DEGREES OF SEVERITY FOR FIELD POINT
C-----
  200 IF (ISEVFP.NE.0) GO TO 300
      WRITE (6,2050)
 2050 FORMAT(/31X,'P FIELD POINT NUMBER',15X/)
      DO 290 K = 1,M
      LNF = INT (NFP/50)
      IF (LNF.LT.1) GO TO 264
      DO 263 J = 1,LNF
      IF (J.NE.1) GO TO 262
      WRITE (6,1042) K,(NDSKPF(K,LF),LF=1,50)
C1042 FORMAT (4X,I3,9X,50I1)
      GO TO 263
  262 WRITE(6,1045) (NDSKPF(K,LF),LF=(J-1)*50+1,J*50)
C1045 FORMAT(16X,50I1)
  263 CONTINUE
      IF((NFP-LNF*50).NE.0) WRITE(6,1045) (NDSKPF(K,LF),LF=LNF*50+1,NFP)
      GO TO 290
  264 WRITE (6,1042) K,(NDSKPF(K,LF),LF=1,NFP)
  290 CONTINUE
C-----
C-- 290 END OF K LOOP : ELEMENT LOOP
C-----
  300 WRITE (6,1060)
 1060 FORMAT (2X)
      WRITE (6,1025)
              WRITE(6,1021) (J,J=1,5)
      RETURN
      END


C==1=================================================================================
      SUBROUTINE NMDS (M,N,NCP,NFP,NDSKP,NDSKPC,NDSKPF,NDSS,X,XCP,XFP,
     1               ISEVCP,ISEVFP,NODE)
C=====================================================================================
C-----
C-- TO SPECIFY DEGREES OF SEVERITY FOR EACH SEGMENT
C-- USES NEW ALGORITHM : SEV = (3.35 + 0.6*COS(THETA))*(L/D)
C-----
      DIMENSION NODE(8,M),SMP(3),V1(3),V2(3),SMP1(3),SMP2(3),
     1         SNOR(3),D(3),PQD(8),X(3,N),XCP(3,NCP),XFP(3,NFP)
      INTEGER NSEV
              REAL LAMDA
              INTEGER*2 NDSKP(M,N),NDSKPC(M,NCP),NDSKPF(M,NFP),NDSS(2,M)
      COMMON/OMEGA/WN
      COMMON/NGTR/NGRH(8,2,2),NGTH(8,2,2)
      PI = 3.141593
      LAMDA = 2.0*PI/WN

C-----
C-- MODIFICATIONS TO ALLOW DIFFERENT VALUES OF NGRHO AND NGTHET
C-- KQT = 1 -- QUADRILATERAL ELEMENT; KQT = 2 -- TRIANGULAR ELEMENT
C-- KLN = 1 -- CORNER NODE; KLN = 2 -- MIDSIDE NODE
C-----
C
C-----
C-- KSS LOOP : TO GIVE THE VALUES OF NGRHO & NGTHET FOR DEGREE
C--           OF SEVERITY -- KSS, FROM 3 TO 8
C-----
      DO 69 KSS = 3,8
      IF (KSS.LE.5) KSS1=KSS+1
      IF (KSS.GT.5) KSS1=KSS1+2
      KSS2 = KSS - 1
```

```
      IF (KSS.EQ.3.OR.KSS.EQ.8) KSS2=KSS
      KSS3 = KSS
      IF (KSS.GT.6.) KSS3=KSS2+2
      DO 69 KQT=1,2
      DO 69 KLN=1,2
      IF (KLN.EQ.2) GO TO 35
      NGRHO = KSS1
      NGTHET = KSS2
      GO TO 36
   35 IF (KQT.EQ.1) GO TO 37
      NGRHO = KSS2
      NGTHET = KSS1
      GO TO 36
   37 NGRHO = KSS3
      NGTHET = KSS3
   36 IF (NGRHO.LT.4) NGRHO=4
      IF (NGTHET.LT.4) NGTHET=4
      NGRH(KSS,KQT,KLN) = NGRHO
      NGTH(KSS,KQT,KLN) = NGTHET
   69 CONTINUE
C-----
C-- END OF KSS LOOP
C-----
      NDMAX = 0
      DO 300 K=1,M
      NLN = 8
      IF(NODE(2,K).EQ.NODE(3,K)) NLN=6
C-----
C-- DETERMINE THE SEGMENT MID POINT, ITS SLOPE IN DIRECTION 1 AND 2, AND
C-- THEN CROSS MULTIPY THE TWO SLOPE VECTORS TO OBTAIN THE NORMAL.
C-----
      DO 40 I=1,3
      SMP1(I)=0.
      SMP2(I)=0.
      DO 30 J=1,4
      SMP1(I)=SMP1(I)+X(I,NODE(J,K))
      SMP2(I)=SMP2(I)+X(I,NODE(J+4,K))
   30 CONTINUE
      SMP(I)= -0.25 * SMP1(I) + 0.5 * SMP2(I)
      V1(I)=(X(I,NODE(5,K))-X(I,NODE(7,K)))/2.
      V2(I)=(X(I,NODE(6,K))-X(I,NODE(8,K)))/2.
   40 CONTINUE
C-----
C-- SMP IS THE SEGMENT MID POINT
C-- V1 AND V2 ARE THE SLOPES IN RESPECTIVE DIRECTIONS
C-----
      SNOR(1)=V1(2)*V2(3)-V1(3)*V2(2)
      SNOR(2)=V1(3)*V2(1)-V1(1)*V2(3)
      SNOR(3)=V1(1)*V2(2)-V1(2)*V2(1)
      ABSN=(SNOR(1)**2+SNOR(2)**2+SNOR(3)**2)**0.5
C-----
C-- SNOR ARE THE COMPONENT OF NORMAL VECTOR.
C-- DETERMINE THE LONGEST DIAGONAL.
C-----
      IK = 2
      IF (NLN.EQ.6) IK=3
      B2M=0.
      DO 60 J=1,IK
      II = J + 2
      JJ = J
      IF(J.NE.3) GO TO 45
      II = 4
      JJ = 1
   45 B2=0.
      DO 50 I=1,3
      NODII = NODE(II,K)
      NODJJ = NODE(JJ,K)
   50 B2=B2+(X(I,NODII)-X(I,NODJJ))**2
      IF(B2.GT.B2M) B2M=B2
   60 CONTINUE
      BM = SQRT(B2M)
```

```
      BMAX=BM*0.7071
C-----
C-- DETERMINE THE SEVERITY FOR THE P.EQ.Q PART OF THE CALCULATIONS
C-- NOTE THAT THE LEAST NUMBER OF SEVERITY IS 3
C-----
      RATIO = BM/LAMDA
      IF (RATIO.GE.3.0) GO TO 65
      IF (RATIO.LT.1.5) RAT1 = RATIO + 3
      IF (RATIO.GE.1.5) RAT1 = RATIO + 4
      IF (RATIO.GE.2.5) RAT1 = RATIO + 5
      IRAT1 = INT(RAT1)
      GO TO 66
   65 IRAT1 = 8
   66 NDSS(2,K) = IRAT1
C-----
C-- DETERMINE THE SEVERITY FOR P.NE.Q PART OF THE CALCULATION
C-- L IS THE GLOBAL NODE NUMBER
C-----
      DO 85 L=1,N
C-----
C-- SKIP TO THE NEXT NODE IF NODE L LIES ON SEGMENT K
C-----
      DO 70 J=1,8
      IF (L.EQ.NODE(J,K)) NDSKP(K,L) = 1
      IF (L.EQ.NODE(J,K)) GO TO 85
   70 CONTINUE
C-----
C-- DETERMINE THE VECTOR D AND ITS ABSOLUTE VALUE FROM POINT P TO THE
C-- SEGMENT MID POINT AND THE DOT PRODUCT OF D AND SNOR TO EVALUATE THE
C-- COSINE OF ANGLE 'C' BETWEEN THEM, THEN CALCULATE THE SEVERITY USING
C-- THE NEW ALGORITHM.
C-----
      D2=0.
      DDOTN=0.
      DO 80 I=1,3
      D(I)=X(I,L)-SMP(I)
      DDOTN=DDOTN+D(I)*SNOR(I)
   80 D2=D2+D(I)**2
      ABSD=SQRT(D2)
      RAT=BMAX/ABSD
      C=ABS(DDOTN/(ABSD*ABSN))
      SEV = ( 3.35 + 0.6*C ) * RAT
      NSEV = INT (SEV)
      DIFF = SEV - NSEV
      IF (DIFF.GE.0.5) NSEV = NSEV + 1
      IF (NSEV.GT.5) NSEV = 5
      IF (NSEV.LT.1) NSEV = 1
C-----
C-- EVALUATE THE SEVERITY BASED ON THE RATIO DELTAR(I.E.,PMMAX - PMIN))
C-- TO WAVELENGTH LAMDA ANND COMPARE TO NSEV
C-----
      DO 75 J=1,8
      PQD2 =0.
      DO 74 IC=1,3
      PD = X(IC,NODE(J,K)) - X(IC,L)
   74 PQD2 = PQD2 + PD**2
   75 PQD(J) = SQRT(PQD2)
      PMAX = AMAX1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     *           PQD(8))
      PMIN = AMIN1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     *           PQD(8))
      DELR = PMAX  - PMIN
      RATIO = DELR/LAMDA
      IF (NSEV.EQ.5) GO TO 1004
      IF (RATIO.LT.0.5) GO TO 1001
      IF (RATIO.GE.3.0) GO TO 1002
      IF (RATIO.LT.1.5) RAT1 = RATIO + 3.0
      IF (RATIO.GE.1.5) RAT1 = RATIO + 4.0
      IF (RATIO.GE.2.5) RAT1 = RATIO + 5.0
cc 84->1004   83->1003   82->1002   81->1001
```

```
              MNSEV = INT(RAT1)
      GO TO 1003
 1001 MNSEV = 2
      GO TO 1003
 1002 MNSEV = 8
 1003 IF (MNSEV.GT.NSEV) NSEV = MNSEV
      IF (NSEV.GT.NDMAX) NDMAX=NSEV
      GO TO 86
 1004 IRAT = INT(RATIO)
      NSEV = NSEV + IRAT
      IF (NSEV.GT.8) NSEV=8
   86 NDSKP(K,L) = NSEV
   85 CONTINUE
C-----
C-- DETERMINE THE SEVERITY FOR CHIEF POINT P
C-- LC IS THE CHIEF POINT NUMBER
C-----
      IF (NCP.EQ.0.OR.ISEVCP.NE.0) GO TO 200
      DO 185 LC=1,NCP
C-----
C-- DETERMINE THE VECTOR D AND ITS ABSOLUTE VALUE FROM POINT P TO THE
C-- SEGMENT MID POINT AND THE DOT PRODUCT OF D AND SNOR TO EVALUATE THE
C-- COSINE OF ANGLE 'C' BETWEEN THEM, THEN CALCULATE THE SEVERITY USING
C-- THE NEW ALGORITHM.
C-----
      D2=0.
      DDOTN=0.
      DO 180 IY=1,3
      D(IY) = XCP(IY,LC) - SMP(IY)
      DDOTN =DDOTN + D(IY)*SNOR(IY)
  180 D2 = D2 + D(IY)**2
      ABSD=SQRT(D2)
      RAT=BMAX/ABSD
      C=ABS(DDOTN/ABSD*ABSN)
      SEV = (3.35 + 0.6*C) * RAT
      NSEV = INT(SEV)
      DIFF = SEV - NSEV
      IF (DIFF.GE.0.5) NSEV = NSEV + 1
      IF (NSEV.GT.5) NSEV = 5
      IF (NSEV.LT.1) NSEV = 1
C-----
C-- EVALUATE THE SEVERITY BASED ON THE RATIO DELTAR(I.E.,PMAX - PMIN)
C-- TO WAVELENGTH LAMDA AND COMPARE TO NSEV
C-----
      DO 175 J=1,8
      PQD2 =0.
      DO 174 I=1,3
      PD = X(I,NODE(J,K)) - XCP(I,LC)
  174 PQD2 = PQD2 + PD**2
  175 PQD(J) = SQRT(PQD2)
      PMAX = AMAX1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     1           PQD(8))
      PMIN = AMIN1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     1           PQD(8))
      DELR = PMAX - PMIN
      RATIO = DELR/LAMDA
      IF (NSEV.EQ.5) GO TO 184
      IF (RATIO.LT.0.5) GO TO 181
      IF (RATIO.GE.3.0) GO TO 182
      IF (RATIO.LT.1.5) RAT1 = RATIO + 3.0
      IF (RATIO.GE.1.5) RAT1 = RATIO + 4.0
      IF (RATIO.GE.2.5) RAT1 = RATIO + 5.0
      MNSEV = INT(RAT1)
      GO TO 183
  181 MNSEV = 2
      GO TO 183
  182 MNSEV = 8
  183 IF (MNSEV.GT.NSEV) NSEV = MNSEV
      IF (NSEV.GT.NDMAX) NDMAX=NSEV
      GO TO 186
  184 IRAT = INT(RATIO)
```

```
      NSEV = NSEV + IRAT
      IF (NSEV.GT.8) NSEV=8
  186 NDSKPC(K,LC) = NSEV
  185 CONTINUE
C-----
C-- 185 END OF LC LOOP : CHIEF POINT LOOP
C-----
C
C-----
C-- DETERMINE TH ESEVERITY FOR FIELD POINT P
C-- LF IS THE FIELD POINT NUMBER
C-----
  200 IF (ISEVFP.NE.0) GO TO 300
      DO 285 LF=1,NFP
C-----
C-- DETERMINE THE VECTOR D AND ITS ABSOLUTE VALUE FROM POINT P TO THE
C-- SEGMENT MID POINT AND THE DOT PRODUCT OF D AND SNOR TO EVALUATE THE
C-- COSINE OF ANGLE 'C' BETWEEN THEM, THEN CALCULATE THE SEVERITY USING
C-- THE NEW ALGORITHM
C-----
      D2=0.
      DDOTN=0.
      DO 280 I=1,3
      D(I) = XFP(I,LF) - SMP(I)
      DDOTN = DDOTN + D(I)*SNOR(I)
  280 D2 = D2 + D(I)**2
      ABSD=SQRT(D2)
      RAT=BMAX/ABSD
      C=ABS(DDOTN/(ABSD*ABSN))
      SEV = (3.35 + 0.6*C) * RAT
      NSEV = INT(SEV)
      DIFF = SEV -NSEV
      IF (DIFF.GE.0.5) NSEV = NSEV + 1
      IF (NSEV.GT.5) NSEV = 5
      IF (NSEV.LT.1) NSEV = 1
C-----
C-- EVALUATE THE SEVERITY BASED ON THE RATIO DELTAR(I.E.,PMAX - PMIN)
C-- TO WAVELENGTH LAMDA AND COMPARE TOO NSEV
C-----
      DO 275 J=1,8
      PQD2 =0.
      DO 274 I=1,3
      PD = X(I,NODE(J,K)) - XFP(I,LF)
  274 PQD2 = PQD2 + PD**2
  275 PQD(J) = SQRT(PQD2)
      PMAX = AMAX1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     1           PQD(8))
      PMIN = AMIN1(PQD(1),PQD(2),PQD(3),PQD(4),PQD(5),PQD(6),PQD(7),
     1           PQD(8))
      DELR = PMAX - PMIN
      RATIO = DELR/LAMDA
      IF (NSEV.EQ.5) GO TO 284
      IF (RATIO.LT.0.5) GO TO 281
      IF (RATIO.GE.3.0) GO TO 282
      IF (RATIO.LT.1.5) RAT1 = RATIO + 3.0
      IF (RATIO.GE.1.5) RAT1 = RATIO + 4.0
      IF (RATIO.GE.2.5) RAT1 = RATIO + 5.0
      MNSEV = INT(RAT1)
      GO TO 283
  281 MNSEV = 2
      GO TO 283
  282 MNSEV = 8
  283 IF (MNSEV.GT.NSEV) NSEV = MNSEV
      IF (NSEV.GT.NDMAX) NDMAX=NSEV
      GO TO 286
  284 IRAT = INT(RATIO)
      NSEV = NSEV + IRAT
      IF (NSEV.GT.8) NSEV=8
  286 NDSKPF(K,LF) = NSEV
  285 CONTINUE
C-----
```

```
C-- 285 END OF LF LOOP : FIELD POINT LOOP
C-----
  300 CONTINUE
      DO 700 K=1,M
      DO 400 L=1,N
      IF (NDSKP(K,L).EQ.1) NDSKP(K,L)=NDMAX
  400 CONTINUE
      IF (ISEVCP.EQ.0) GO TO 550
      DO 500 JC = 1,NCP
      NDSKPC(K,JC) = ISEVCP
  500 CONTINUE
  550 IF (ISEVFP.EQ.0) GO TO 700
      DO 600 JF = 1,NFP
      NDSKPF(K,JF) = ISEVFP
  600 CONTINUE
  700 CONTINUE
C-----
C-- BACK TO SUBROUTINE DEGSEV
C-----
      RETURN
      END


C==1=================================================================
      SUBROUTINE CHAPE (SHPQ,SHPQ1,SHPQ2,SHPT,SHPT1,SHPT2,NIT,
     1                                                  E1,E2,WFPNQQ)
C=====================================================================
C-----
C-- TO EVALUATE SHAPE FUNCTIONS AND ITS DERIVATIVES
C-----
      DIMENSION SHPQ(8,8,144),SHPQ1(8,8,144),SHPQ2(8,8,144),
     1          SHPT(8,8,112),SHPT1(8,8,112),SHPT2(8,8,112),
     2          E1(8,112),E2(8,112),WFPNQQ(8,144),NIT(14)
      COMMON/CGAUS/Y(12,12),W(12,12),NGTHET,NGRHO

               DO 61 NID = 1,8
c-----
c-- INTERIOR OF -1,+1 SQUARE
C-----
      IF (NID.LE.5) NDQ=NID+1
      IF (NID.GT.5) NDQ=NDQ+2
      NFEQ = NDQ*NDQ
      IFE=0
      DO 10 IG1 = 1,NDQ
      DO 10 IG2 = 1,NDQ
      IFE=IFE+1
      Y1=Y(NDQ,IG1)
      Y2=Y(NDQ,IG2)
      SHPQ(1,NID,IFE) = .25*(Y1+1)*(Y2+1)*(Y1+Y2-1)
      SHPQ(2,NID,IFE) = .25*(Y1-1)*(Y2+1)*(Y1-Y2+1)
      SHPQ(3,NID,IFE) = .25*(-Y1+1)*(Y2-1)*(Y1+Y2+1)
      SHPQ(4,NID,IFE) = .25*(Y1+1)*(Y2-1)*(-Y1+Y2+1)
      SHPQ(5,NID,IFE) = .5*(Y1+1)*(1-Y2*Y2)
      SHPQ(6,NID,IFE) = .5*(Y2+1)*(1-Y1*Y1)
      SHPQ(7,NID,IFE) = .5*(Y1-1)*(-1+Y2*Y2)
      SHPQ(8,NID,IFE) = .5*(-Y2+1)*(1-Y1*Y1)
C- TURUNAN TERHADAP PSI(1)
      SHPQ1(1,NID,IFE) = .25*(Y2+1)*(2*Y1+Y2)
      SHPQ1(2,NID,IFE) = .25*(Y2+1)*(2*Y1-Y2)
      SHPQ1(3,NID,IFE) = .25*(Y2-1)*(-2*Y1-Y2)
      SHPQ1(4,NID,IFE) = .25*(Y2-1)*(-2*Y1+Y2)
      SHPQ1(5,NID,IFE) = .5*(1-Y2*Y2)
      SHPQ1(6,NID,IFE) = -Y1*(Y2+1)
      SHPQ1(7,NID,IFE) = .5*(-1+Y2*Y2)
      SHPQ1(8,NID,IFE) = -Y1*(-Y2+1)
C- TURUNAN TERHADAP PSI(2)
      SHPQ2(1,NID,IFE) = .25*(Y1+1)*(Y1+2*Y2)
      SHPQ2(2,NID,IFE) = .25*(Y1-1)*(Y1-2*Y2)
      SHPQ2(3,NID,IFE) = .25*(-Y1+1)*(Y1+2*Y2)
      SHPQ2(4,NID,IFE) = .25*(Y1+1)*(-Y1+2*Y2)
      SHPQ2(5,NID,IFE) = -Y2*(Y1+1)
```

```
      SHPQ2(6,NID,IFE) = .5*(1-Y1*Y1)
      SHPQ2(7,NID,IFE) = Y2*(Y1-1)
      SHPQ2(8,NID,IFE) = -.5*(1-Y1*Y1)
C-
      WFPNQQ(NID,IFE)=W(NDQ,IG1)*W(NDQ,IG2)
   10 CONTINUE
C-----
C-- INTERIOR OF 1,1,1 TRIANGLE
C-----
            IF (NID.EQ.1) NDI=4
      IF (NID.EQ.2) NDI=5
      IF (NID.EQ.3) NDI=8
      IF (NID.EQ.4) NDI=9
      IF (NID.EQ.5) NDI=11
      IF (NID.EQ.6) NDI=12
      IF (NID.EQ.7) NDI=13
      IF (NID.EQ.8) NDI=14
      NFE=NIT(NDI)

            DO 15 IFE=1,NFE
      Y1=E1(NID,IFE)
      Y2=E2(NID,IFE)
      Y3=1.-Y1-Y2

            SHPT(1,NID,IFE) = Y1*(2*Y1-1)
      SHPT(2,NID,IFE) = 0.0
      SHPT(3,NID,IFE) = Y2*(2*Y2-1)
      SHPT(4,NID,IFE) = Y3*(2*Y3-1)
      SHPT(5,NID,IFE) = 4*Y1*Y3
      SHPT(6,NID,IFE) = 4*Y1*Y2
      SHPT(7,NID,IFE) = 0.0
      SHPT(8,NID,IFE) = 4*Y2*Y3
      SHPT1(1,NID,IFE) = 4*Y1 - 1
      SHPT1(2,NID,IFE) = 0.0
      SHPT1(3,NID,IFE) = 0.0
      SHPT1(4,NID,IFE) = 1 - 4*Y3
      SHPT1(5,NID,IFE) = 4*(1-2*Y1-Y2)
      SHPT1(6,NID,IFE) = 4*Y2
      SHPT1(7,NID,IFE) = 0.0
      SHPT1(8,NID,IFE) = -4*Y2
      SHPT2(1,NID,IFE) = 0.0
      SHPT2(2,NID,IFE) = 0.0
      SHPT2(3,NID,IFE) = 4*Y2 - 1
      SHPT2(4,NID,IFE) = 1 - 4*Y3
      SHPT2(5,NID,IFE) = -4*Y1
      SHPT2(6,NID,IFE) = 4*Y1
      SHPT2(7,NID,IFE) = 0.0
      SHPT2(8,NID,IFE) = 4*(1-Y1-2*Y2)
   15 CONTINUE
   61 CONTINUE

ctambahan darwin buat ngecek
C-------------------------------------------------------------------------------
c     DO 1091 L = 1,8
c           IF (L.EQ.1) NDI=4
c     IF (L.EQ.2) NDI=5
c     IF (L.EQ.3) NDI=8
c     IF (L.EQ.4) NDI=9
c     IF (L.EQ.5) NDI=11
c     IF (L.EQ.6) NDI=12
c     IF (L.EQ.7) NDI=13
c     IF (L.EQ.8) NDI=14
c     NFE=NIT(NDI)
c                       WRITE(6,1098)
c 1098                  FORMAT(/1X,20(1H=),'NILAI VARIABEL SHPT(1-8,L,K)',20(1H=))
c                       WRITE(6,1097)
c 1097                  FORMAT(1X,'L',2X,'K',8X,'1',8X,'2',8X,'3',8X,'4',8X,'5',
c    1                    8X,'6',8X,'7',8X,'8')
c                       DO 1091 K = 1,NFE
c                       WRITE(6,1092) L,K,(SHPT(JA,L,K),JA=1,8)
c 1092 FORMAT (1X,I1,1X,I3,1X,8(1X,F8.5))
```

```
c 1091 CONTINUE
c
c              DO 1093 L = 1,8
c              IF (L.EQ.1) NDI=4
c     IF (L.EQ.2) NDI=5
c     IF (L.EQ.3) NDI=8
c     IF (L.EQ.4) NDI=9
c     IF (L.EQ.5) NDI=11
c     IF (L.EQ.6) NDI=12
c     IF (L.EQ.7) NDI=13
c     IF (L.EQ.8) NDI=14
c     NFE=NIT(NDI)
c                          WRITE(6,1099)
c 1099                     FORMAT(/1X,20(1H=),'NILAI VARIABEL SHPT1(1-8,L,K)',20(1H=))
c                          WRITE(6,1097)
c                          DO 1093 K = 1,NFE
c                            WRITE(6,1094) L,K,(SHPT1(JA,L,K),JA=1,8)
c 1094 FORMAT (1X,I1,1X,I3,1X,8(1X,F8.5))
c 1093 CONTINUE
c
c              DO 1095 L = 1,8
c              IF (L.EQ.1) NDI=4
c     IF (L.EQ.2) NDI=5
c     IF (L.EQ.3) NDI=8
c     IF (L.EQ.4) NDI=9
c     IF (L.EQ.5) NDI=11
c     IF (L.EQ.6) NDI=12
c     IF (L.EQ.7) NDI=13
c     IF (L.EQ.8) NDI=14
c     NFE=NIT(NDI)
c                          WRITE(6,1082)
c 1082                     FORMAT(/1X,20(1H=),'NILAI VARIABEL SHPT2(1-8,L,K)',20(1H=))
c                          WRITE(6,1097)
c                          DO 1095 K = 1,NFE
c                            WRITE(6,1096) L,K,(SHPT2(JA,L,K),JA=1,8)
c 1096 FORMAT (1X,I1,1X,I3,1X,8(1X,F8.5))
c 1095 CONTINUE
c
C-------------------------------------------------------------------------------
              RETURN
      END

C=======================================================================
      SUBROUTINE COEF(N,NPCH,M,A,B,BTN,C,D,PFP,X,XCP,XFP,NCP,NFP,NODE,
     1     SHPQ,SHPQ1,SHPQ2,SHPT,SHPT1,SHPT2,NIT,SHAPQ,SHAPQ1,SHAPQ2,
     2     SHAPT,SHAPT1,SHAPT2,WFPQQ,WFPQT,WFPNQQ,WFPNQT,NDSKP,NDSKPC,
     3     NDSKPF,KKK,KKMAX,FPP,EPHI,EPHIF)
C=======================================================================
C-----
C-- TO COMPUTE THE COEFFISIEN OF MATRIX A, B AND BTN IN EQUATION
C-- [A] [PHI] + [B] [DPHI/DN] = [BTN]
C-- [C] [PHI] + [D] [DPHI/DN] = [PFP]
C-----
      DIMENSION AA(3),BB(3),XQ(3),XP(3),NIT(14),EPHI(NPCH),EPHIF(NFP),
     1         DX(3),RGRAD(3),FPP(NFP),BTN(NPCH),PFP(NFP)
C-
      DIMENSION A(NPCH,N),B(NPCH,N),C(NFP,N),D(NFP,N),
     1         NODE(8,M),WFPNQQ(8,144),WFPNQT(8,112)
C-
      DIMENSION SHPQ(8,8,144),SHPQ1(8,8,144),SHPQ2(8,8,144),
     1         SHPT(8,8,112),SHPT1(8,8,112),SHPT2(8,8,112)
C-
      DIMENSION SHAPQ(8,3,8,100),SHAPQ1(8,3,8,100),SHAPQ2(8,3,8,100),
     1         SHAPT(8,2,8,100),SHAPT1(8,2,8,100),SHAPT2(8,2,8,100),
     2         WFPQQ(8,3,100),WFPQT(8,2,100)
C-
      INTEGER*2 NDSKP(M,N),III(8,150),JJMAX(8),KKK(8,M),KKMAX(6),
     1         NDSKPC(M,NCP),IIC(8,10),JCMAX(8),
     2         NDSKPF(M,NFP),IIF(8,200),JFMAX(8)
      INTEGER PSYM
      REAL X(3,N),XCP(3,NCP),XFP(3,NFP)
```

```
      REAL JAC,NSTAR(3)
      COMPLEX A,B,BTN,C,D,PFP,FPP
      COMPLEX UKERN1,UKERN2,ARG,ARGT,EPHI,EPHIF,IK
      LOGICAL IPS2,INTP
      COMMON /SYM/PSYM
      COMMON/CGAUS/Y(12,12),W(12,12),NGTHET,NGRHO
      COMMON/NGTR/NGRH(8,2,2),NGTH(8,2,2)/SEV/KSS
      COMMON/OMEGA/WN
      COMMON/IDEN/IDD
      COMMON/SCAT/ISC,AF,BE,GA
C-----
C-- EVALUATE CONSTANTS
C-----
      PI = 3.141593
      PIT4 = PI * 4.
      ROOT3 = SQRT(3.0)
      CCC = 1.0
      CCR = ROOT3/2.0
C-----
C-- INITIALIZE A,B AND BTN MATRIX
C-----
      DO 5 I=1,NPCH
      BTN(I) = CMPLX(0.,0.)
            DO 5 J=1,N
      A(I,J) = CMPLX(0.,0.)
    5 B(I,J) = CMPLX(0.,0.)
C-----
C-- INITIALIZE C,D AND PFP MATRIX
C-----
      IK = CMPLX(0.0,WN)
      DO 505 I=1,NFP
      PFP(I) = -PIT4*FPP(I)/(IK*415.0)
      DO 505 J=1,N
      C(I,J) = CMPLX(0.,0.)
  505 D(I,J) = CMPLX(0.,0.)
C-----
C-- INCREMENT SEGMENT SEVERITY NUMBER
C-----
      DO 650 KSS = 3,8
      KSSM2=KSS-2
      NKS = KKMAX(KSSM2)
      IF (NKS.EQ.0) GO TO 650
C--
      CALL PEQ(SHAPQ,SHAPQ1,SHAPQ2,SHAPT,SHAPT1,SHAPT2,WFPQQ,WFPQT,NGPQ)
C--
            DO 603 KSSC=1,NKS
            K = KKK(KSS,KSSC)
            NLN=8
            IF (NODE(2,K).EQ.NODE(3,K)) NLN=6
C-----
C-- INITIALIZE II VALUES TO ZERO AND SORT PER VALUE OF NDI
C-----
            DO 6 IC=1,8
            JJMAX(IC)=0
            DO 6 JC=1,N
    6       III(IC,JC) = 0
            JJ1 = 0
            JJ2 = 0
            JJ3 = 0
            JJ4 = 0
            JJ5 = 0
            JJ6 = 0
            JJ7 = 0
            JJ8 = 0
            DO 10 JC=1,N
            NID = NDSKP(K,JC)
   15       GO TO (1501,1502,1503,1504,1505,1506,1507,1508),NID
 1501       JJ1=JJ1+1
            III(NID,JJ1) = JC
            GO TO 10
 1502       JJ2=JJ2+1
```

```
            III(NID,JJ2) = JC
            GO TO 10
 1503       JJ3=JJ3+1
            III(NID,JJ3) = JC
            GO TO 10
 1504       JJ4=JJ4+1
            III(NID,JJ4) = JC
            GO TO 10
 1505       JJ5=JJ5+1
            III(NID,JJ5) = JC
            GO TO 10
 1506       JJ6=JJ6+1
            III(NID,JJ6) = JC
            GO TO 10
 1507       JJ7=JJ7+1
            III(NID,JJ7) = JC
            GO TO 10
 1508       JJ8=JJ8+1
            III(NID,JJ8) = JC
   10       CONTINUE
            JJMAX(1)=JJ1
            JJMAX(2)=JJ2
            JJMAX(3)=JJ3
            JJMAX(4)=JJ4
            JJMAX(5)=JJ5
            JJMAX(6)=JJ6
            JJMAX(7)=JJ7
            JJMAX(8)=JJ8
C-----
C-- P.NE.Q
C-----
            DO 110 NID=1,8
            IF (JJMAX(NID).EQ.0) GO TO 110
            IF (NLN.EQ.8) GO TO 20
            IF (NID.EQ.1) NDI=4
            IF (NID.EQ.2) NDI=5
            IF (NID.EQ.3) NDI=8
            IF (NID.EQ.4) NDI=9
            IF (NID.EQ.5) NDI=11
            IF (NID.EQ.6) NDI=12
            IF (NID.EQ.7) NDI=13
            IF (NID.EQ.8) NDI=14
            NFE=NIT(NDI)
            GO TO 25
   20       NX=NID+1
            NFE=NX*NX
            IF (NID.EQ.6) NFE=(NID+2)**2
            IF (NID.EQ.7) NFE=(NID+3)**2
            IF (NID.EQ.8) NFE=(NID+4)**2
C-----
C-- INCREMENT INDICES IDENTIFIYING POINTS FOR GAUSSIAN QUADRATURE
C-----
   25       DO 100 IFE=1,NFE
C-----
C-- EVALUATE JACOBIAN AND COMPONENTS OF MODIFIED UNIT NORMAL
C-- AA AND BB ARE VECTORS TANGENTIAL TO THE LOCAL COORDINATE
C-- Y1 AND Y2 RESPECTIVELY
C-- XQ IS THE GLOBAL COORDINATES OF (Y1,Y2) POINT OF AN ELEMENT
C-- ACCORDING TO IFE
C-----
            DO 30 I=1,3
            XQ(I) = 0.
            AA(I) = 0.
            BB(I) = 0.
            DO 30 JALPH = 1,8
            XIN = X(I,NODE(JALPH,K))
            IF (NLN.EQ.6) GO TO 31
            XQ(I) = XQ(I) + SHPQ (JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPQ1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPQ2(JALPH,NID,IFE)*XIN
            GO TO 30
```

```
  31        IF (JALPH.EQ.2.OR.JALPH.EQ.7) GO TO 30
            XQ(I) = XQ(I) + SHPT (JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPT1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPT2(JALPH,NID,IFE)*XIN
  30        CONTINUE
C-----
C-- NSTAR NORMAL VECTOR
C-- JAC   JACOBIAN OF TRANSFORMATION, .EQ. MAG OF NORMAL VECTOR
C-----
            NSTAR(1) = AA(2)*BB(3) - AA(3)*BB(2)
            NSTAR(2) = AA(3)*BB(1) - AA(1)*BB(3)
            NSTAR(3) = AA(1)*BB(2) - AA(2)*BB(1)
            JAC = SQRT(NSTAR(1)**2+NSTAR(2)**2+NSTAR(3)**2)
C-----
C-- UTILIZE SYMMETRY
C-----
            NJJ=JJMAX(NID)
            IPS2 = .FALSE.
            NPSYM = 1
            IF (PSYM.NE.0) NPSYM = 2
             DO 95 IPSYM = 1,NPSYM
                     IF (IPSYM.EQ.2) IPS2 = .TRUE.
C-----
C-- 90 INCREMENT P NODE NUMBER
C-----
            DO 90 JJ=1,NJJ
            IR=III(NID,JJ)
            INTP = .FALSE.
            IF (PSYM.EQ.0) GO TO 35
            IF (X(PSYM,IR).EQ.0.) INTP= .TRUE.
  35        CONTINUE
            IF (.NOT.IPS2) GO TO 38
            IF (.NOT.INTP) GO TO 42
  38        CONTINUE
            DO 40 JALPH = 1,8
            IF (X(1,NODE(JALPH,K)).EQ.X(1,IR).AND.X(2,NODE(JALPH,K)).EQ.
   1        X(2,IR).AND.X(3,NODE(JALPH,K)).EQ.X(3,IR)) GO TO 90
  40        CONTINUE
  42        CONTINUE
            XP(1) = X(1,IR)
            XP(2) = X(2,IR)
            XP(3) = X(3,IR)
            IF (IPS2) XP(PSYM) = -XP(PSYM)

C-----
C--  R          R(P,Q) -- DISTANCE BETWEEN P AND Q
C--  RGRAD(I)    DR/DX(I)
C--  DRDN        DR/DN * JAC
C-----
C-
            DX(1) = XQ(1) - XP(1)
            DX(2) = XQ(2) - XP(2)
            DX(3) = XQ(3) - XP(3)
            R2 = DX(1)*DX(1) + DX(2)*DX(2) + DX(3)*DX(3)
            R = SQRT(R2)
            RGRAD(1) = DX(1)/R
            RGRAD(2) = DX(2)/R
            RGRAD(3) = DX(3)/R
            DRDN = RGRAD(1)*NSTAR(1) + RGRAD(2)*NSTAR(2) +RGRAD(3)*NSTAR(3)
C-----
C-- EVALUATE THE KERNELS (UKERN1 & UKERN2)
C-- IDD .EQ. 1 -- INTERIOR PROBLEM; IDD .NE. 1 -- EXTERIOR PROBLEM
C-- ARG     -IKR(P,Q)
C-- ARGT    (1+IKR)
C-- UKERN1 K1(P,Q) = DK2/DN = (DK2/DR) (DR/DN)
C-- UKERN2 K2(P,Q) = EXP (-IKR(P,Q))/R(P,Q)
C-- TTSTAR D(1/R)/DN = -(1/R2) (DR/DN)
C-----
            ARG = CMPLX(0.,-WN*R)
            UKERN2 = CEXP(ARG)/R
            ARGT = CMPLX(1.,WN*R)
```

```
            UKERN1 = -ARGT*CEXP(ARG)*DRDN/R2
            TTSTAR = -DRDN/R2
            IF (PSYM.EQ.0) GO TO 45
            IF (IPS2.AND.INTP) GO TO 90
 43         IF (IPS2.OR..NOT.INTP) GO TO 45
            UKERN1 = UKERN1 * 2.
            UKERN2 = UKERN2 * 2.
            TTSTAR = TTSTAR * 2.
 45         CONTINUE
            IF (NLN.EQ.6) WTIG=WFPNQT(NID,IFE)*CCR
            IF (NLN.EQ.8) WTIG=WFPNQQ(NID,IFE)*4.
            DO 80 JA = 1,8
            IF (NLN.EQ.6.AND.(JA.EQ.2.OR.JA.EQ.7)) GO TO 80
            IC = NODE(JA,K)
            IF (NLN.EQ.6) SHALPH = SHPT(JA,NID,IFE)
            IF (NLN.EQ.8) SHALPH = SHPQ(JA,NID,IFE)
            A(IR,IC) = A(IR,IC) + SHALPH * UKERN1 * WTIG
            B(IR,IC) = B(IR,IC) + SHALPH * UKERN2 * JAC * WTIG
 80         CONTINUE
            A(IR,IR) = A(IR,IR) - TTSTAR * WTIG
 90         CONTINUE
 95         CONTINUE
100         CONTINUE
110         CONTINUE
C-----
C--  90 END OF JJ LOOP
C--  95 END OF IPSYM LOOP
C-- 100 END OF IFE LOOP
C-- 110 END OF NID LOOP
C-----
C
C-----
C-- P.EQ.Q
C-- PLACE P AT EACH NODE OF THE SEGMENT
C-----
            CCC=1
            IF (NLN.EQ.6) CCC=CCR
            DO 230 JA = 1,8
            IF (NLN.EQ.6.AND.(JA.EQ.2.OR.JA.EQ.7)) GO TO 230
            RO1=(23-NLN)/8
            RO2=(3+JA)/4
            KQT=INT(RO1)
            KLN=INT(RO2)
C-----
C-- NGRHO   NUMBER OF EVALUATIONS IN RHO DIRECTION
C-- NGTHET  NUMBER OF EVALUATIONS IN THETA DIRECTION
C-- NGPQ    TOTAL NUMBER OF EVALUATIONS IN POLAR COORDINATE
C--         FOR P.EQ.Q
C-----
            NGRHO=NGRH(KSS,KQT,KLN)
            NGTHET=NGTH(KSS,KQT,KLN)
            NGPQ=NGRHO*NGTHET
            IR = NODE(JA,K)
            LMAX = 2
            IF (JA.GT.4) LMAX = 3
            IF (NLN.EQ.6) LMAX = 2
            INTP = .FALSE.
            IF (PSYM.EQ.0) GO TO 140
            IF (X(PSYM,IR).EQ.0.) INTP = .TRUE.
 140        CONTINUE
C-----
C-- L     INCREMENT TRIANGULAR SUBDIVISION OF MODIFIED UNIT NORMAL
C-- IPQ   INCREMENT NUMBER OF EVALUATION
C-----
            DO 220 L = 1,LMAX
            DO 200 IPQ=1,NGPQ
C-----
C-- EVALUATE JACOBIAN AND COMPONENTS OF MODIFIED UNIT NORMAL
C-- AA AND BB ARE VECTORS TANGENTIAL TO THE LOCAL COORDINATE
C-- XQ IS THE GLOBAL COORDINATES OF (RHO,THETA) POINT OF AN ELEMENT
C-- ACCORDING TO IPQ
```

```
C-----
  146       DO 150 I=1,3
            XQ(I) = 0
            AA(I) = 0
            BB(I) = 0
            DO 150 JALPH = 1,8
            XIN = X(I,NODE(JALPH,K))
            IF (NLN.EQ.6) GO TO 151
            XQ(I) = XQ(I) + SHAPQ(JALPH,L,JA,IPQ)*XIN
            AA(I) = AA(I) + SHAPQ1(JALPH,L,JA,IPQ)*XIN
            BB(I) = BB(I) + SHAPQ2(JALPH,L,JA,IPQ)*XIN
            GO TO 150
  151       IF (JALPH.EQ.2.OR.JALPH.EQ.7) GO TO 150
            XQ(I) = XQ(I) + SHAPT(JALPH,L,JA,IPQ)*XIN
            AA(I) = AA(I) + SHAPT1(JALPH,L,JA,IPQ)*XIN
            BB(I) = BB(I) + SHAPT2(JALPH,L,JA,IPQ)*XIN
  150       CONTINUE
C-----
C-- NSTAR   NORMAL VECTOR
C-- JAC     JACOBIAN OF TRANSFORMATION, .EQ. MAG OF NORMAL VECTOR
C-----
            NSTAR(1) = AA(2)*BB(3) - AA(3)*BB(2)
            NSTAR(2) = AA(3)*BB(1) - AA(1)*BB(3)
            NSTAR(3) = AA(1)*BB(2) - AA(2)*BB(1)
            JAC = SQRT(NSTAR(1)**2+NSTAR(2)**2+NSTAR(3)**2)
C-----
C-- R       R(P,Q) -- DISTANCE BETWEEN P AND Q
C-- RGRAD(I)  DR/XY(I)
C-- DRDN      DR/DN
C-----
            DX(1) = XQ(1) - X(1,IR)
            DX(2) = XQ(2) - X(2,IR)
            DX(3) = XQ(3) - X(3,IR)
            R2 = DX(1)*DX(1) + DX(2)*DX(2) + DX(3)*DX(3)
            R = SQRT(R2)
            RGRAD(1) = DX(1)/R
            RGRAD(2) = DX(2)/R
            RGRAD(3) = DX(3)/R
            DRDN = RGRAD(1)*NSTAR(1) + RGRAD(2)*NSTAR(2) + RGRAD(3)*NSTAR(3)
C-----
C-- EVALUATE THE KERNELS (UKERN1 & UKERN2)
C-- IDD .EQ. 1 -- INTERIOR PROBLEM; IDD .NE. 1 -- EXTERIOR PROBLEM
C-- ARG     -IKR(P,Q)
C-- ARGT    (1+IKR)
C-- UKERN1   K1(P,Q) = DK2/DR) (DR/DN)
C-- UKERN2   K2(P,Q) = EXP (-IKR (P,Q) / R(P,Q)
C-- TTSTAR   D(1/R)/DN = -(1/R2) (DR/DN)
C-----
            ARG = CMPLX(0.,-WN*R)
            UKERN2 = CEXP(ARG)/R
            ARGT = CMPLX(1.,WN*R)
            UKERN1 = -ARGT*CEXP(ARG)*DRDN/R2
            TTSTAR = -DRDN/R2
C-----
C-- UTILIZE SYMMETRI
C-----
            IF (.NOT.INTP) GO TO 155
            UKERN1 = UKERN1 * 2.
            UKERN2 = UKERN2 * 2.
            TTSTAR = TTSTAR * 2.
  155       CONTINUE
            IF (NLN.EQ.6) WTRDR=WFPQT(JA,L,IPQ)*CCC
            IF (NLN.EQ.8) WTRDR=WFPQQ(JA,L,IPQ)*CCC
            DO 190 J = 1,8
            IF (NLN.EQ.6.AND.(J.EQ.2.OR.J.EQ.7)) GO TO 190
            IC = NODE(J,K)
            IF (NLN.EQ.6) SHALPH=SHAPT(J,L,JA,IPQ)
            IF (NLN.EQ.8) SHALPH=SHAPQ(J,L,JA,IPQ)
            A(IR,IC) = A(IR,IC) + SHALPH * UKERN1 * WTRDR
            B(IR,IC) = B(IR,IC) + SHALPH * UKERN2 * JAC * WTRDR
            DO 185 I = 1,N
```

```
            IF (X(1,I).NE.X(1,IR).OR.X(2,I).NE.X(2,IR).OR.
     1      X(3,I).NE.X(3,IR)) GO TO 185
            IF (I.EQ.IR) GO TO 185
            A(I,IC) = A(I,IC) + SHALPH * UKERN1 * WTRDR
            B(I,IC) = B(I,IC) + SHALPH * UKERN2 * JAC * WTRDR
  185       CONTINUE
  190       CONTINUE
            A(IR,IR) = A(IR,IR) - TTSTAR * WTRDR
  200       CONTINUE
  220       CONTINUE
  230       CONTINUE
C-----
C-- 200 END OF IPQ LOOP
C-- 220 END OF L LOOP
C-- 230 END OF JA LOOP
C-----
C
C-----
C-- P CHIEF POINT
C-- INITIALIZE IIC VALUES TO ZERO AND SORT PER VALUE OF NDSKPC/NID
C-----
            IF (NCP.EQ.0) GO TO 500
            DO 407 IC = 1,8
            JCMAX(IC) = 0
            DO 407 JC = 1,NCP
  407       IIC(IC,JC) = 0
            JC1 = 0
            JC2 = 0
            JC3 = 0
            JC4 = 0
            JC5 = 0
            JC6 = 0
            JC7 = 0
            JC8 = 0
            DO 410 JC=1,NCP
            NID = NDSKPC(K,JC)
  415       GO TO (1601,1602,1603,1604,1605,1606,1607,1608),NID
 1601       JC1=JC1+1
            IIC(NID,JC1) = JC
            GO TO 410
 1602       JC2=JC2+1
            IIC(NID,JC2) = JC
            GO TO 410
 1603       JC3=JC3+1
            IIC(NID,JC3) = JC
            GO TO 410
 1604       JC4=JC4+1
            IIC(NID,JC4) = JC
            GO TO 410
 1605       JC5=JC5+1
            IIC(NID,JC5) = JC
            GO TO 410
 1606       JC6=JC6+1
            IIC(NID,JC6) = JC
            GO TO 410
 1607       JC7=JC7+1
            IIC(NID,JC7) = JC
            GO TO 410
 1608       JC8=JC8+1
            IIC(NID,JC8) = JC

  410       CONTINUE
            JCMAX(1) = JC1
            JCMAX(2) = JC2
            JCMAX(3) = JC3
            JCMAX(4) = JC4
            JCMAX(5) = JC5
            JCMAX(6) = JC6
            JCMAX(7) = JC7
            JCMAX(8) = JC8
C--
```

```
            DO 4110 NID=1,8
            IF (JCMAX(NID).EQ.0) GO TO 4110
            IF (NLN.EQ.8) GO TO 420
            IF (NID.EQ.1) NDI=4
            IF (NID.EQ.2) NDI=5
            IF (NID.EQ.3) NDI=8
            IF (NID.EQ.4) NDI=9
            IF (NID.EQ.5) NDI=11
            IF (NID.EQ.6) NDI=12
            IF (NID.EQ.7) NDI=13
            IF (NID.EQ.8) NDI=14
            NFE=NIT(NDI)
            GO TO 425
 420        NX=NID+1
            NFE=NX*NX
            IF (NID.EQ.6) NFE=(NID+2)**2
            IF (NID.EQ.7) NFE=(NID+3)**2
            IF (NID.EQ.8) NFE=(NID+4)**2
C-----
C-- INCREMENT INDICES IDENTIFYING POINTS FO RGAUSSIAN QUADRATURE
C-----
 425        DO 4100 IFE=1,NFE
C-----
C-- EVALUATE JACOBIAN AND COMPONENTS OF MODIFIED UNIT NORMAL
C-- AA AND BB ARE VECTORS TANGENTIAL TO THE LOCAL COORDINATE
C-- Y1 AND Y2 RESPECTIVELY
C-- XQ IS THE GLOBAL COORDINATES OF (Y1,Y2) POINT OF AN ELEMENT
C-- ACCORDING TO IFE
C-----
            DO 430 I = 1,3
            XQ(I) = 0
            AA(I) = 0
            BB(I) = 0
            DO 430 JALPH = 1,8
            XIN = X(I,NODE(JALPH,K))
            IF (NLN.EQ.6) GO TO 431
            XQ(I) = XQ(I) + SHPQ(JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPQ1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPQ2(JALPH,NID,IFE)*XIN
            GO TO 430
 431        IF (JALPH.EQ.2.OR.JALPH.EQ.7) GO TO 430
            XQ(I) = XQ(I) + SHPT(JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPT1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPT2(JALPH,NID,IFE)*XIN
 430        CONTINUE
C-----
C-- NSTAR   NORMAL VECTOR
C-- JAC     JACOBIAN OF TRANSFORMATION, .EQ. MAG OF NORMAL VECTOR
C-----
            NSTAR(1) = AA(2)*BB(3) - AA(3)*BB(2)
            NSTAR(2) = AA(3)*BB(1) - AA(1)*BB(3)
            NSTAR(3) = AA(1)*BB(2) - AA(2)*BB(1)
            JAC = SQRT(NSTAR(1)**2+NSTAR(2)**2+NSTAR(3)**2)
C-----
C-- UTILIZE SYMMETRI
C-----
            NJC=JCMAX(NID)
            IPS2 = .FALSE.
            NPSYM = 1
            IF (PSYM.NE.0) NPSYM = 2
            DO 495 IPSYM = 1,NPSYM
            IF (IPSYM.EQ.2) IPS2 = .TRUE.
C-----
C-- 490 INCREMENT CHIEF POINT NUMBER
C-----
            DO 490 JC=1,NJC
            IR=IIC(NID,JC)
            INTP = .FALSE.
            IF (PSYM.EQ.0) GO TO 435
            GO TO 435
            IF (XCP(PSYM,IR).EQ.0) INTP = .TRUE.
```

```
 435      CONTINUE
          XP(1) = XCP(1,IR)
          XP(2) = XCP(2,IR)
          XP(3) = XCP(3,IR)
          IF (IPS2) XP(PSYM) = -XP(PSYM)
C-----
C-- R      R(P,Q) -- DISTANCE BETWEEN P AND Q
C-- RGRAD(I)  DR/XY(I)
C-- DRDN     DR/DN
C-----
          DX(1) = XQ(1) - XP(1)
          DX(2) = XQ(2) - XP(2)
          DX(3) = XQ(3) - XP(3)
          R2 = DX(1)*DX(1) + DX(2)*DX(2) + DX(3)*DX(3)
          R = SQRT(R2)
          RGRAD(1) = DX(1)/R
          RGRAD(2) = DX(2)/R
          RGRAD(3) = DX(3)/R
          DRDN = RGRAD(1)*NSTAR(1)+RGRAD(2)*NSTAR(2)+RGRAD(3)*NSTAR(3)
C-----
C-- EVALUATE THE KERNELS (UKERN1 & UKERN2)
C-- IDD .EQ. 1 -- INTERIOR PROBLEM; IDD .NE. 1 -- EXTERIOR PROBLEM
C-- ARG     -IKR(P,Q)
C-- ARGT    (1+IKR)
C-- UKERN1   K1(P,Q) = DK2/DN = (DK2/DR) (DR/DN)
C-- UKERN2   K2(P,Q) = EXP (-IKR(P,Q) ) / r(P,Q)
C-----
          ARG = CMPLX(0.,-WN*R)
          UKERN2 = CEXP(ARG)/R
          ARGT = CMPLX(1.,WN*R)
          UKERN1 = -ARGT*CEXP(ARG)*DRDN/R2
          IF (PSYM.EQ.0) GO TO 445
          IF (IPS2.AND.INTP) GO TO 490
 443      IF (IPS2.OR..NOT.INTP) GO TO 445
          UKERN1 = UKERN1 * 2.
          UKERN2 = UKERN2 * 2.
 445      CONTINUE
          IF (NLN.EQ.6) WTIG=WFPNQT(NID,IFE)*CCR
          IF (NLN.EQ.8) WTIG=WFPNQT(NID,IFE)*4.
          DO 480 JA = 1,8
          IF (NLN.EQ.6.AND.(JA.EQ.2.OR.JA.EQ.7)) GO TO 480
          IC = NODE(JA,K)
          IF (NLN.EQ.6) SHALPH = SHPT(JA,NID,IFE)
          IF (NLN.EQ.8) SHALPH = SHPQ(JA,NID,IFE)
          ICH = N + IR
          A(ICH,IC) = A(ICH,IC) + SHALPH * UKERN1 * WTIG
          B(ICH,IC) = B(ICH,IC) + SHALPH * UKERN2 * JAC * WTIG
 480      CONTINUE
 490      CONTINUE
 495      CONTINUE
4100      CONTINUE
4110      CONTINUE
C-----
C-- 490 END OF JF LOOP
C-- 495 END OF IPSYM LOOP
C-- 4100 END OF IFE LOOP
C-- 4110 END OF NIDO LOOP
C-----
C
C-----
C-- P FIELD POINT
C-- INITIALIZE IIF VALUES TO ZERO AND SORT PER VALUE OF NDSKPF/NID
C-----
 500      DO 506 IF1=1,8
          JFMAX(IF1) = 0
          DO 506 JF=1,NFP
 506      IIF(IF1,JF) = 0
          JF1 = 0
          JF2 = 0
          JF3 = 0
          JF4 = 0
```

```
            JF5 = 0
            JF6 = 0
            JF7 = 0
            JF8 = 0
            DO 510 JF=1,NFP
            NID = NDSKPF(K,JF)
 515        GO TO (1701,1702,1703,1704,1705,1706,1707,1708),NID
1701        JF1=JF1+1
            IIF(NID,JF1) = JF
            GO TO 510
1702        JF2=JF2+1
            IIF(NID,JF2) = JF
            GO TO 510
1703        JF3=JF3+1
            IIF(NID,JF3) = JF
            GO TO 510
1704        JF4=JF4+1
            IIF(NID,JF4) = JF
            GO TO 510
1705        JF5=JF5+1
            IIF(NID,JF5) = JF
            GO TO 510
1706        JF6=JF6+1
            IIF(NID,JF6) = JF
            GO TO 510
1707        JF7=JF7+1
            IIF(NID,JF7) = JF
            GO TO 510
1708        JF8=JF8+1
            IIF(NID,JF8) = JF
 510        CONTINUE
            JFMAX(1) = JF1
            JFMAX(2) = JF2
            JFMAX(3) = JF3
            JFMAX(4) = JF4
            JFMAX(5) = JF5
            JFMAX(6) = JF6
            JFMAX(7) = JF7
            JFMAX(8) = JF8
C-------------------------
            DO 5110 NID = 1,8
            IF (JFMAX(NID).EQ.0) GO TO 5110
            IF (NLN.EQ.8) GO TO 520
            IF (NID.EQ.1) NDI=4
            IF (NID.EQ.2) NDI=5
            IF (NID.EQ.3) NDI=8
            IF (NID.EQ.4) NDI=9
            IF (NID.EQ.5) NDI=11
            IF (NID.EQ.6) NDI=12
            IF (NID.EQ.7) NDI=13
            IF (NID.EQ.8) NDI=14
            NFE=NIT(NDI)
            GO TO 525
 520        NX=NID+1
            NFE=NX*NX
            IF (NID.EQ.6) NFE=(NID+2)**2
            IF (NID.EQ.7) NFE=(NID+3)**2
            IF (NID.EQ.8) NFE=(NID+4)**2
C-----
C-- INCREMENT INDICES IDENTIFYING POINTS FOR GAUSSIAN QUADRATURE
C-----
 525        DO 5100 IFE=1,NFE
C-----
C-- EVALUATE JACOBIAN AND COMPONENTS OF MODIFIED UNIT NORMAL
C-- AA AND BB ARE VECTORS TANGENTIAL TO THE LOCAL COORDINATE
C-- Y1 AND Y2 RESPECTIVELY
C-- XQ IS THE GLOBAL COORDINATES OF (Y1,Y2) POINT OF AN ELEMENT
C-- ACCORDING TO IFE
C-----
            DO 530 I = 1,3
            XQ(I) = 0.
```

```
            AA(I) = 0.
            BB(I) = 0.
            DO 530 JALPH = 1,8
            XIN = X(I,NODE(JALPH,K))
            IF (NLN.EQ.6) GO TO 531
            XQ(I) = XQ(I) + SHPQ(JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPQ1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPQ2(JALPH,NID,IFE)*XIN
            GO TO 530
 531        IF (JALPH.EQ.2.OR.JALPH.EQ.7) GO TO 530
            XQ(I) = XQ(I) + SHPT (JALPH,NID,IFE)*XIN
            AA(I) = AA(I) + SHPT1(JALPH,NID,IFE)*XIN
            BB(I) = BB(I) + SHPT2(JALPH,NID,IFE)*XIN
 530        CONTINUE
C-----
C-- NSTAR   NORMAL VECTOR
C-- JAC     JACOBIAN OF TRANSFORMATION, ..EQ. MAG OF NORMAL VECTOR
C-----
            NSTAR(1) = AA(2)*BB(3) - AA(3)*BB(2)
            NSTAR(2) = AA(3)*BB(1) - AA(1)*BB(3)
            NSTAR(3) = AA(1)*BB(2) - AA(2)*BB(1)
            JAC = SQRT(NSTAR(1)**2+NSTAR(2)**2+NSTAR(3)**2)
C-----
C-- UTILIZE SYMMETRI
C-----
            NJF=JFMAX(NID)
            IPS2 = .FALSE.
            NPSYM = 1
            IF (PSYM.NE.0) NPSYM = 2
            DO 595 IPSYM = 1,NPSYM
            IF (IPSYM.EQ.2) IPS2 = .TRUE.
C-----
C-- 590 INCREMENT P FIELD POINT NUMBER
C-----
            DO 590 JF=1,NJF
            IR=IIF(NID,JF)
            INTP = .FALSE.
            IF (PSYM.EQ.0) GO TO 535
            IF (XFP(PSYM,IR).EQ.0) INTP = .TRUE.
 535        CONTINUE
            XP(1) = XFP(1,IR)
            XP(2) = XFP(2,IR)
            XP(3) = XFP(3,IR)
            IF (IPS2) XP(PSYM) = -XP(PSYM)
C-----
C-- R       R(P,Q) -- DISTANCE BETWEEN P AND Q
C-- RGRAD(I) DR/DX(I)
C-- DRDN    DR/DN
C-----
            DX(1) = XQ(1) - XP(1)
            DX(2) = XQ(2) - XP(2)
            DX(3) = XQ(3) - XP(3)
            R2 = DX(1)*DX(1) + DX(2)*DX(2) + DX(3)*DX(3)
            R = SQRT(R2)
            RGRAD(1) = DX(1)/R
            RGRAD(2) = DX(2)/R
            RGRAD(3) = DX(3)/R
            DRDN = RGRAD(1)*NSTAR(1) + RGRAD(2)*NSTAR(2) +RGRAD(3)*NSTAR(3)
C-----
C-- EVALUATE THE KERNELS (UKERN1 & UKERN2)
c     IDD . EQ. 1 -- INTERIOR PROBLEM; IDD .NE. 1 -- EXTERIOR PROBLEM
c     ARG       -ikR(P,Q)
c     ARGT      (1+ikR)
c     UKERN1    K1(P,Q = DK2/DN = (DK2/DR) (DR/DN)
c     UKERN2    K2(P,Q) = EXP ( -ikR(P,Q) ) / R(P,Q)
C-----
            ARG = CMPLX(0.,-WN*R)
            UKERN2 = CEXP(ARG)/R
            ARGT = CMPLX(1.,WN*R)
            UKERN1 = -ARGT*CEXP(ARG)*DRDN/R2
            IF (PSYM.EQ.0) GO TO 545
```

```
              IF (IPS2.AND.INTP) GO TO 590
  543         IF (IPS2.OR..NOT.INTP) GO TO 545
              UKERN1 = UKERN1 * 2.
              UKERN2 = UKERN2 * 2.
  545         CONTINUE
              IF (NLN.EQ.6) WTIG=WFPNQT(NID,IFE)*CCR
              IF (NLN.EQ.8) WTIG=WFPNQT(NID,IFE)*4.
              DO 580 JA = 1,8
              IF (NLN.EQ.6.AND.(JA.EQ.2.OR.JA.EQ.7)) GO TO 580
              IC = NODE(JA,K)
              IF (NLN.EQ.6) SHALPH = SHPT(JA,NID,IFE)
              IF (NLN.EQ.8) SHALPH = SHPQ(JA,NID,IFE)
              C(IR,IC) = C(IR,IC) + SHALPH * UKERN1 * WTIG
              D(IR,IC) = D(IR,IC) - SHALPH * UKERN2 * JAC * WTIG
  580         CONTINUE
  590         CONTINUE
  595         CONTINUE
 5100         CONTINUE
 5110         CONTINUE
C-----
C--  590 END OF JF LOOP
C--  595 END OF IPSYM LOOP
C-- 5100 END OF IFE LOOP
C-- 5110 END OF NIDO LOOP
C-----
C----------------------------------
  603         CONTINUE
  650 CONTINUE
C-----
C-- 600 END OF KSSC LOOP
C-- 650 END OF KSS LOOP
C-----
      IF (ISC.EQ.0) GO TO 1000
      CALL SCATR (X,XCP,EPHI,N,NCP,NPCH,XFP,EPHIF,NFP)
      DO 652 I = 1,NPCH
  652 BTN(I) = BTN(I) + EPHI(I)*PIT4
      DO 653 I = 1,NFP
  653 PFP(I) = PFP(I) + EPHIF(I)*PIT4
 1000 IF (IDD.EQ.1) GO TO 656
      DO 655 L = 1,N
  655 A(L,L) = A(L,L) + PIT4
  656 CONTINUE
C     WRITE (6,9000)
C 9000 FORMAT (/2X,'A')
C     CALL COEMR (A,NPCH,N)
C             WRITE (6,9001)
C 9001 FORMAT (/2X,'B' )
C     CALL COEMR (B,NPCH,N)
C             WRITE (6,9002)
C 9002 FORMAT (/2X,'C')
C     CALL COEMR (C,NFP,N)
C             WRITE (6,9003)
C 9003 FORMAT (/2X,'D')
C     CALL COEMR (D,NFP,N)
C             WRITE (6,9004)
C 9004 FORMAT (/2X,'BTN')
C     CALL COEMR (BTN,NPCH)
C             WRITE (6,9005)
C 9005 FORMAT (/2X,'PFP')
C     CALL COEMR (PFP,NFP)
              RETURN
      END

cc udah betul
C------------------------
C=1=================================================================
      SUBROUTINE PEQ(SHAPQ,SHAPQ1,SHAPQ2,SHAPT,SHAPT1,SHAPT2,WFPQQ,
     1              WFPQT,NGPQ)
C=================================================================
C-----
C-- TO COMPUTE THE SHAPE AND WEIGHTING FUNCTION FOR P.EQ.Q
```

```
C-----
      DIMENSION SHAPQ(8,3,8,100),SHAPQ1(8,3,8,100),SHAPQ2(8,3,8,100),
     1          SHAPT(8,2,8,100),SHAPT1(8,2,8,100),SHAPT2(8,2,8,100),
     2          WFPQQ(8,3,100),WFPQT(8,2,100)
      COMMON/CGAUS/Y(12,12),W(12,12),NGTHET,NGRHO
      COMMON/NGTR/NGRH(8,2,2),NGTH(8,2,2)/SEV/KSS
      PI = 3.141593
      PID2 = PI/2.
      PID4 = PI/4.
      PID6 = PI/6.
      DT = PI/4.
      DT1 = ATAN(2.)
      DT2 = 2.*ATAN(.5)
      DT3 = DT1
      BT1 = - 3.*PI/2.
      BT2 = PI/2.
      ROOT3=SQRT(3.)
      NLN=8
      GO TO 2
    1 NLN=6
    2 CONTINUE
                DO 430 JA = 1,8
                IF (NLN.EQ.6.AND.(JA.EQ.2.OR.JA.EQ.7)) GO TO 430
                RO1 = (23 - NLN)/8
                RO2 = (3 + JA)/4
                KQT = INT(RO1)
                KLN = INT(RO2)
C-----
C-- NGRHO   NUMBER OF EVALUATIONS INI RHO DIRECTION
C-- NGTHET  NUMBER OF EVALUATIONS INI THETA DIRECTION
C-- NGPQ    TOTAL NUMBER OF EVALUATIONS INI POLAR COORDINATE
C--         FOR P.EQ.Q
C-----
                NGRHO = NGRH(KSS,KQT,KLN)
                NGTHET = NGTH(KSS,KQT,KLN)
                NGPQ = NGRHO*NGTHET
                LMAX=2
                IF (NLN.EQ.8.AND.JA.GT.4) LMAX = 3
C-----
C-- INCREMENT SUBDIVISION NUMBER
C-----
                DO 420 L = 1,LMAX
                IPQ=0
C-----
C-- EVALUATE INITIAL VALUE AND INCREMENT OF THETA
C-- BTHETA INITIAL VALUE
C-- DTHETA INCREMENT OF THETA
C-----
                IF (NLN.EQ.8) GO TO 310
                IF (JA.GT.4) GO TO 305
                IF (L.GT.1) GO TO 304
                BTHETA = 0.
                DTHETA = PID6
                GO TO 330
  304           BTHETA = PID6
                DTHETA = PID6
                GO TO 330
  305           IF (L.GT.1) GO TO 306
                BTHETA = 0.
                DTHETA = PID2
                GO TO 330
  306           BTHETA = PID2
                DTHETA = PID2
                GO TO 330
  310           CONTINUE
                IF (JA.LE.4) DTHETA = DT
                IF (JA.LE.4) GO TO 320
                IF (L.EQ.1) DTHETA = DT1
                IF (L.EQ.2) DTHETA = DT2
                IF (L.EQ.3) DTHETA = DT3
  320           CONTINUE
```

```
            BTHETA = BT1 + BT2*JA
            IF (L.GT.1) BTHETA = BTHETA + PID4
            IF (JA.LE.4) GO TO 330
            BTHETA = (JA-4)*PID2
            IF (L.GT.1) BTHETA = BTHETA + DT1
            IF (L.GT.2) BTHETA = BTHETA + DT2
 330        CONTINUE
C-----
C-- INCREMENT THETA NUMBER
C-----
            DO 410 IG2 = 1,NGTHET
            THETA = BTHETA + (1.+Y(NGTHET,IG2))*DTHETA/2.
C-----
C-- EVALUATE INITIAL VALUE AND INCREMENT OF RHO
C-----
            SINT = SIN(THETA)
            COST = COS(THETA)
            IF (NLN.EQ.8) GO TO 335
            IF (JA.GT.4) GO TO 333
            DRHO = 1./COS(THETA-PID6)
            GO TO 340
 333        IF (L.EQ.1) DRHO = 1./(ROOT3*COST + SINT)
            IF (L.EQ.2) DRHO = 1./(SINT - ROOT3*COST)
            GO TO 340
 335        CONTINUE
            IF (JA.GT.4) GO TO 337
            IF (L.EQ.1) DRHO = 2./COS(THETA-BTHETA)
            IF (L.EQ.2) DRHO = 2./COS(BTHETA+PID4-THETA)
            GO TO 340
 337        IF (L.EQ.1) DRHO = 1./COS(THETA-(JA-4)*PID2)
            IF (L.EQ.2) DRHO = 2./COS(THETA-(JA-3)*PID2)
            IF (L.EQ.3) DRHO = 1./COS((JA-2)*PID2-THETA)
 340        CONTINUE
C-----
C-- INCREMENT RHO NUMBER
C-----
            DO 400 IG1 = 1,NGRHO
            RHO = (1.+Y(NGRHO,IG1))*DRHO/2.
            IPQ=IPQ + 1
            IF (NLN.EQ.6) WFPQT(JA,L,IPQ)=W(NGRHO,IG1)*W(NGTHET,IG2)*DRHO*
     1                                          DTHETA*RHO
            IF (NLN.EQ.8) WFPQQ(JA,L,IPQ)=W(NGRHO,IG1)*W(NGTHET,IG2)*DRHO*
     1                                          DTHETA*RHO
C-----
C-- EVALUATE XI COORDINATES
C-----
            IF (NLN.EQ.8) GO TO 343
            Z1 = RHO * SINT
            Z2 = 1.-.5 * RHO * (SINT + ROOT3 * COST)
            Z22 = Z2 - .5
            GO TO (1341,1343,1343,1344,1345,1346,1343,1348),JA
 1341       XI3 = Z1
            XI1 = Z2
            XI2 = 1. - XI1 - XI3
            GO TO 344
 1343       XI1 = Z1
            XI2 = Z2
            XI3 = 1. - XI1 - XI2
            GO TO 344
 1344       XI2 = Z1
            XI3 = Z2
            XI1 = 1. - XI2 - XI3
            GO TO 344
 1345       XI2 = Z1
            XI3 = Z22
            XI1 = 1. - XI2 - XI3
            GO TO 344
 1346       XI3 = Z1
            XI1 = Z22
            XI2 = 1. - XI1 - XI3
            GO TO 344
```

```
 1348       XI1 = Z1
            XI2 = Z22
            XI3 = 1. - XI1 - XI2
            GO TO 344
  343       CONTINUE
            XIP1=RHO*COST
            XIP2=RHO*SINT
            IF (JA.EQ.1.OR.JA.EQ.4.OR.JA.EQ.5) XI1 = XIP1 + 1.
            IF (JA.EQ.6.OR.JA.EQ.8) XI1 = XIP1
            IF (JA.EQ.2.OR.JA.EQ.3.OR.JA.EQ.7) XI1 = XIP1 - 1.
            IF (JA.EQ.1.OR.JA.EQ.2.OR.JA.EQ.6) XI2 = XIP2 + 1.
            IF (JA.EQ.5.OR.JA.EQ.7) XI2 = XIP2
            IF (JA.EQ.3.OR.JA.EQ.4.OR.JA.EQ.8) XI2 = XIP2 - 1.
  344       CONTINUE
C-----
C-- EVALUATE SHAPE FUNCTIONS
C-----
            Y1 = XI1
            Y2 = XI2
            IF (NLN.EQ.6) GO TO 345
            SHAPQ(1,L,JA,IPQ) = .25*(Y1+1)*(Y2+1)*(Y1+Y2-1)
            SHAPQ(2,L,JA,IPQ) = .25*(Y1-1)*(Y2+1)*(Y1-Y2+1)
            SHAPQ(3,L,JA,IPQ) = .25*(-Y1+1)*(Y2-1)*(Y1+Y2+1)
            SHAPQ(4,L,JA,IPQ) = .25*(Y1+1)*(Y2-1)*(-Y1+Y2+1)
            SHAPQ(5,L,JA,IPQ) = .5*(Y1+1)*(1-Y2*Y2)
            SHAPQ(6,L,JA,IPQ) = .5*(Y2+1)*(1-Y1*Y1)
            SHAPQ(7,L,JA,IPQ) = .5*(Y1-1)*(-1+Y2*Y2)
            SHAPQ(8,L,JA,IPQ) = .5*(-Y2+1)*(1-Y1*Y1)
            SHAPQ1(1,L,JA,IPQ) = .25*(Y2+1)*(2*Y1+Y2)
            SHAPQ1(2,L,JA,IPQ) = .25*(Y2+1)*(2*Y1-Y2)
            SHAPQ1(3,L,JA,IPQ) = .25*(Y2-1)*(-2*Y1-Y2)
            SHAPQ1(4,L,JA,IPQ) = .25*(Y2-1)*(-2*Y1+Y2)
            SHAPQ1(5,L,JA,IPQ) = .5*(1-Y2*Y2)
            SHAPQ1(6,L,JA,IPQ) = -Y1*(Y2+1)
            SHAPQ1(7,L,JA,IPQ) = .5*(-1+Y2*Y2)
            SHAPQ1(8,L,JA,IPQ) = -Y1*(-Y2+1)
            SHAPQ2(1,L,JA,IPQ) = .25*(Y1+1)*(Y1+2*Y2)
            SHAPQ2(2,L,JA,IPQ) = .25*(Y1-1)*(Y1-2*Y2)
            SHAPQ2(3,L,JA,IPQ) = .25*(-Y1+1)*(Y1+2*Y2)
            SHAPQ2(4,L,JA,IPQ) = .25*(Y1+1)*(-Y1+2*Y2)
            SHAPQ2(5,L,JA,IPQ) = -Y2*(Y1+1)
            SHAPQ2(6,L,JA,IPQ) = .5*(1-Y1*Y1)
            SHAPQ2(7,L,JA,IPQ) = Y2*(Y1-1)
            SHAPQ2(8,L,JA,IPQ) = -.5*(1-Y1*Y1)
            GO TO 346
  345       CONTINUE
            Y3 = XI3
            SHAPT(1,L,JA,IPQ) = Y1*(2*Y1-1)
            SHAPT(2,L,JA,IPQ) = 0.0
            SHAPT(3,L,JA,IPQ) = Y2*(2*Y2-1)
            SHAPT(4,L,JA,IPQ) = Y3*(2*Y3-1)
            SHAPT(5,L,JA,IPQ) = 4*Y1*Y3
            SHAPT(6,L,JA,IPQ) = 4*Y1*Y2
            SHAPT(7,L,JA,IPQ) = 0.0
            SHAPT(8,L,JA,IPQ) = 4*Y2*Y3
            SHAPT1(1,L,JA,IPQ) = 4*Y1-1
            SHAPT1(2,L,JA,IPQ) = 0.0
            SHAPT1(3,L,JA,IPQ) = 0.
            SHAPT1(4,L,JA,IPQ) = 1-4*Y3
            SHAPT1(5,L,JA,IPQ) = 4*(1-2*Y1-Y2)
            SHAPT1(6,L,JA,IPQ) = 4*Y2
            SHAPT1(7,L,JA,IPQ) = 0.0
            SHAPT1(8,L,JA,IPQ) = -4*Y2
            SHAPT2(1,L,JA,IPQ) = 0.
            SHAPT2(2,L,JA,IPQ) = 0.0
            SHAPT2(3,L,JA,IPQ) = 4*Y2-1
            SHAPT2(4,L,JA,IPQ) = 1-4*Y3
            SHAPT2(5,L,JA,IPQ) = -4*Y1
            SHAPT2(6,L,JA,IPQ) = 4*Y1
            SHAPT2(7,L,JA,IPQ) = 0.0
            SHAPT2(8,L,JA,IPQ) = 4*(1-Y1-2*Y2)
```

```
  346       CONTINUE
  400       CONTINUE
  410       CONTINUE
  420       CONTINUE
  430       CONTINUE
C-----
C-- 400 END OF IG1 LOOP
C-- 410 END OF IG2 LOOP
C-- 420 END OF L LOOP
C-- 430 END 0F JA LOOP
C-----
      IF (NLN.EQ.8) GO TO 1
CINI PROGRAM TMBAHAN DARWIN
C-------------------------------------------------------------------------------
C             DO 1071 L = 1,2
C               DO 1071 I = 1,8
C                         WRITE(6,1078)
C 1078                    FORMAT(/1X,15(1H=),'NILAI VARIABEL SHAPT(1-8,L,I,K)',15(1H=))
C                         WRITE(6,1077)
C 1077                    FORMAT(1X,'L',2X,'I',2X,'K',8X,'1',8X,'2',8X,'3',8X,'4',8X,'5',
C     1                     8X,'6',8X,'7',8X,'8')
C                          DO 1071 K = 1,16
C                          WRITE(6,1072) L,I,K,(SHAPT(JA,L,I,K),JA=1,8)
C 1072 FORMAT (1X,I1,1X,I2,1X,I3,1X,8(1X,F8.5))
C 1071 CONTINUE
C
C             DO 1073 L = 1,2
C               DO 1073 I = 1,8
C                         WRITE(6,1079)
C 1079                    FORMAT(/1X,15(1H=),'NILAI VARIABEL SHAPT1(1-8,L,I,K)',15(1H=))
C                         WRITE(6,1077)
C                         DO 1073 K = 1,16
C                          WRITE(6,1074) L,I,K,(SHAPT1(JA,L,I,K),JA=1,8)
C 1074 FORMAT (1X,I1,1X,I2,1X,I3,1X,8(1X,F8.5))
C 1073 CONTINUE
C
C             DO 1075 L = 1,2
C               DO 1075 I = 1,8
C                         WRITE(6,1081)
C 1081                    FORMAT(/1X,15(1H=),'NILAI VARIABEL SHAPT2(1-8,L,I,K)',15(1H=))
C                         WRITE(6,1077)
C                         DO 1075 K = 1,16
C                          WRITE(6,1076) L,I,K,(SHAPT2(JA,L,I,K),JA=1,8)
C 1076 FORMAT (1X,I1,1X,I2,1X,I3,1X,8(1X,F8.5))
C 1075 CONTINUE
C-------------------------------------------------------------------------------
      RETURN
      END


C=2=====================================================================
      SUBROUTINE SCATR (X,XCP,EPHI,N,NCP,NPCH,XFP,EPHIF,NFP)
C=======================================================================
C-----
C-- THIS SUBROUTINE IS USED TO SELECT THE INCOMING WAVE EPHI IN THE
C-- SCATTERING PROBLEMS
C-----
      DIMENSION X(3,N),XCP(3,NCP),EPHI(NPCH),XFP(3,NFP),EPHIF(NFP)
      COMPLEX EPHI,EPHIF,IK
      COMMON/OMEGA/WN
      COMMON/SCAT/ISC,AF,BE,GA
      IK = CMPLX(0.0,WN)
      DO 100 I = 1,N
      IF (ISC.EQ.2) GO TO 50
      DIR = AF*X(1,I)+BE*X(2,I)+GA*X(3,I)
      EPHI(I) = CEXP(-IK*DIR)/(IK*415.0)
      GO TO 100
   50 DIR = SQRT( (X(1,I)-AF)**2 + (X(2,I)-BE)**2 + (X(3,I)-GA)**2 )
      EPHI(I) = CEXP(-IK*DIR)/(DIR*IK*415.0)
  100 CONTINUE
C--
```

```
      DO 200 I = 1,NCP
      J = N + I
      IF (ISC.EQ.2) GO TO 150
      DIR = AF*XCP(1,I)+BE*XCP(2,I)+GA*XCP(3,I)
      EPHI(J) = CEXP(-IK*DIR)/(IK*415.0)
      GO TO 200
  150 DIR = SQRT ((XCP(1,I)-AF)**2+(XCP(2,I)-BE)**2+(XCP(3,I)-GA)**2)
      EPHI(J) = CEXP(-IK*DIR)/(DIR*IK*415.0)
  200 CONTINUE
C--
      DO 300 I = 1,NFP
      IF (ISC.EQ.2) GO TO 250
      DIRF = AF*XFP(1,I)+BE*XFP(2,I)+GA*XFP(3,I)
      EPHIF(I) = CEXP(-IK*DIRF)/(IK*415.0)
      GO TO 300
  250 DIRF = SQRT( (XFP(1,I)-AF)**2+(XFP(2,I)-BE)**2+(XFP(3,I)-GA)**2)
      EPHIF(I) = CEXP(-IK*DIRF)/(DIRF*IK*415.0)
  300 CONTINUE
      RETURN
      END
C=2=================================================================================
      SUBROUTINE SOLVE (A,B,BTN,C,D,PFP,NFP,NPCH,N,NFPT2,NPCT2,NT2,
     1            PHI,DPHI,AIB,AP)
C=================================================================================
C-----
C-- TO FIND BOUNDARY CONDITION
C-----
      DIMENSION A(NPCH,N),B(NPCH,N),C(NFP,N),D(NFP,N),BTN(NPCH),
     1         PFP(NFP),PHI(N),DPHI(N),
     2         G(400,300),AIB(NT2,NT2),AP(NT2,NPCT2),YD(400),
     3         XD(300),V(300,300),SV(300,10),S2(300),S(300)
      COMPLEX A,B,BTN,C,D,PFP,PHI,DPHI,VEL,PRES,Z
      COMMON/OMEGA/WN
C-----
C-- USING EQUATION  [A] {PHI} = {BTN} + [B] {DPHI/DN}
C--            OR  {PHI} = [A+] {BTN} + [A+] [B] {DPHI/DN}
C-- AND   EQUATION  [C] {PHI} + [D] {DPHI/DN} = {PFP}
C--            OR  ([C][A+][B]+[D]) {DPHI/DN} = {PFP} - [C] [A+] {BTN}
C-----
C-- SVD ON [A]
C-----
      WRITE (6,5)
    5 FORMAT(/2X,'SVD ON [A]',/2X)
      IAP = 1
C             CALL SVD (AD,BD,CD,DD,DRM,G,NR2,NC2,NR,NC,NRM,NCM,IAP,
C     1             YD,XD,V,SV,S2,S)
      CALL SVD (A,PHI,BTN,B,AP,G,NPCT2,NT2,NPCH,N,NT2,NPCT2,IAP,
     1         YD,XD,V,SV,S2,S)
C-----
C-- INPUT : A,BTN,B,NPCT2,NT2,NPCH,N,NT2,NPCT2,IAP
C-- OUTPUT: PHI,AP,G,YD,XD,V,SV,S2,S
C-----
C-- NOW {PHI} = [AI] {BTN}
C-- [A+] = GENERAL INVERS OF [A] IN REAL
C-- FIND [AIB] = [A+] [B], [B] IN REAL
C-----
      DO 10 I = 1,NT2
      DO 10 K1 = 1,N
      K2 = N + K1
      AIB(I,K1) = 0.0
      AIB(I,K2) = 0.0
      DO 20 J1 = 1,NPCH
      J2 = NPCH + J1
      AIB(I,K1) = AIB(I,K1) + AP(I,J1) * REAL(B(J1,K1))
     1          + AP(I,J2) * AIMAG(B(J1,K1))
      AIB(I,K2) = AIB(I,K2) - AP(I,J1) * AIMAG(B(J1,K1))
     1          + AP(I,J2) * REAL(B(J1,K1))
   20 CONTINUE
   10 CONTINUE
C-----
C-- FIND {PFP} = {PFP} - [C] {PHI}
```

```
C-- FIND [D] = [D] + [C] [AIB], [AIB] IN REAL
C-----
      SG = -1.0
      CALL MTM (C,PHI,PFP,NFP,1,N,SG)
C-----
C-- SVD ON [D]
C-----
      WRITE (6,30)
   30 FORMAT(/2X,'SVD ON [G]',/2X)
      IAP = 0
C       CALL SVD (AD,BD,CD,DD,DRM,G,NR2,NC2,NR,NC,NRM,NCM,IAP,
C     1               YD,XD,V,SV,S2,S)
              CALL SVD (D,DPHI,PFP,C,AIB,G,NFPT2,NT2,NFP,N,NT2,NT2,IAP,
     1         YD,XD,V,SV,S2,S)
C-----
C-- FIND {PHI} = [A+] {BTN} + [AIB] {DPHI}
C-- INITIAL VALUE OF {PHI) = [A+] {BTN}
C-----
      DO 50 I1 = 1,N
      I2 = N + I1
      DRE = 0.0
      DIM = 0.0
      DO 40 J1 = 1,N
      J2 = N + J1
      DRE = DRE + AIB(I1,J1) * REAL(DPHI(J1))
     1           + AIB(I1,J2) * AIMAG(DPHI(J1))
      DIM = DIM + AIB(I2,J1) * REAL(DPHI(J1))
     1           + AIB(I2,J2) * AIMAG(DPHI(J1))
   40 CONTINUE
      PHI(I1) = PHI(I1) + CMPLX(DRE,DIM)
   50 CONTINUE
C--
      WRITE (6,317)
  317 FORMAT (//2X,10(1H-),'BOUNDARY CONDITION',10(1H-),/2X)
C--
      WRITE (6,318)
  318 FORMAT (2X,'NODE',5X,'VELOCITY POTENTIAL',14X,'PRESSURE',13X,
     1       'VELOCITY',16X,'IMPEDANCE')
      DO 320 I = 1,N
      PRES = CMPLX(0.0,WN) * 415.0 * PHI(I)
      VEL = -1.0 * DPHI(I)
      Z = PRES / VEL
      WRITE (6,319) I,PHI(I),PRES,VEL,Z
  319 FORMAT (3X,I3,2X,2(F12.6,1X,F12.6,2X),F8.5,1X,F8.5,2X,
     1       G14.8,1X,G14.8)
  320 CONTINUE
              RETURN
      END


C=======================================================================
      SUBROUTINE SVD (AD,BD,CD,DD,DRM,G,NR2,NC2,NR,NC,NRM,NCM,IAP,
     1              YD,XD,V,SV,S2,S)
C=======================================================================
C-----
C-- TO SOLVE [AD] {BD} = {CD}
C-----
      DIMENSION AD(NR,NC),BD(NC),CD(NR),DD(NR,NC),DRM(NRM,NCM),
     1G(NR2,NC2),YD(NR2),XD(NC2),V(NC2,NC2),SV(NC2,NC2),S2(NC2),
     2S(NC2)
      COMPLEX AD,BD,CD,DD
C-----
C-- CHANGE TO REAL
C-----
              IAO=0
      DO 5 I1 = 1,NR
      I2 = NR + I1
      YD(I1) = REAL(CD(I1))
      YD(I2) = AIMAG(CD(I1))
      DO 5 J1 = 1,NC
      J2 = NC + J1
```

```
      G(I1,J1) = REAL(AD(I1,J1))
      G(I1,J2) = -AIMAG(AD(I1,J1))
      G(I2,J1) = AIMAG(AD(I1,J1))
      G(I2,J2) = REAL(AD(I1,J1))
    5 CONTINUE
      IF (IAP.EQ.1) GO TO 25
C-----
C-- NOW [G] IS [AD] IN REAL
C-- FIND [G] = [G] + [DD] {DRM}
C-----
      DO 10 I1 =1,NR
      I2 = NR + I1
      DO 10 K = 1,NC2
      DO 20 J1 = 1,NC
      J2 = NC + J1
      G(I1,K) = G(I1,K) + REAL(DD(I1,J1))*DRM(J1,K)
     1           - AIMAG(DD(I1,J1))*DRM(J2,K)
      G(I2,K) = G(I2,K) + AIMAG(DD(I1,J1))*DRM(J1,K)
     1          + REAL(DD(I1,J1))*DRM(J2,K)
   20 CONTINUE
   10 CONTINUE
C-----
C-- NOW WE HAVE [G] [XD] = [YD]
C-- INPUT : G,YD,NR2,NC2
C-- OUTPUT: XD,V,S2,S,NRD
C-----

   25 CALL SVDSOL (G,XD,YD,V,S2,S,NR2,NC2,NRD)
      IF (NRD.EQ.0) GO TO 40
C-----
C-- PRINT SINGULAR VECTORS OF [G]
C-----
c      WRITE (6,30)
c    30 FORMAT (/2X,' SINGULAR VECTORS ')
c      DO 33 I = 1,NC2
c      DO 33 J = 1,NRD
c      K = NC2 - NRD + J
c      SV(I,J) = V(I,K)
c   33 CONTINUE
c      CALL COEMR (SV,NC2,NRD)
c      IF (IAP.EQ.1) GO TO 39
c      WRITE (6,35)
   35 FORMAT (/2X,'[A+] [B] IN REAL')
      CALL COEMR (DRM,NRM,NCM)
c       WRITE (6,37)
c   37 FORMAT (/2X,'USE EXEQ.F TO FIND EXTRA EQUATIONS',/2X)
c   39 STOP
C-----
C-- CONTINUE
C-----
   40 DO 45 I1 = 1,NC
      I2 = NC + I1
      BD(I1) = CMPLX(XD(I1),XD(I2))
c               WRITE (6,44)
c               FORMAT (/2X,'[A+] [B] IN REAL')
c     CALL COEMR (DRM,NRM,NCM)
c perintah cetak A
C    44 FORMAT (/2X,'[A]  IN REAL')
C      CALL COEMR (AD,NR,NC)
c   44         FORMAT (/2X,'[C]  IN REAL')
c      CALL COEMR (CD,NR,NR)
c===============
   45 CONTINUE
C               IAP=1
C       call coem (bd,nc,1)
      IF (IAP.NE.1) GO TO 60
C-----
C-- FIND GENERAL INVERS OF [G] = [DRM]
C-----
C       call coemr (g,nr2,nc2)
C       call coemr (v,nc2,nc2)
```

```
C      call coemr (s2,nc2,1)
       DO 50 I = 1,NC2
       DO 52 J = 1,NR2
       DRM(I,J) = 0.0
       DO 54 K = 1,NC2
       DRM(I,J) = DRM(I,J) + V(I,K) * G(J,K) / S2(K)
  54 CONTINUE
  52 CONTINUE
  50 CONTINUE
  60 RETURN
       END
C=======================================================================
       SUBROUTINE SVDSOL (US,XD,YD,VD,S2D,SD,NROW,NCOL,NRD)
C=======================================================================
       DIMENSION US(NROW,NCOL),XD(NCOL),YD(NROW),VD(NCOL,NCOL),SD(NCOL),
      1         S2D(NCOL)
       INTEGER ECR,SLIMIT
       COMMON/MAT/EPS,SVDQ,MSVD
       COMMON/IDEN/IDD
C-----
C-- FIND [US], [V], AND {S2}
C-----
C        call coemr(us,nrow,ncol)
C        call coemr(yd,nrow,1)
       SLIMIT = INT(NCOL/4)
       IF(SLIMIT.LT.6) SLIMIT = 6
       NSWEEP = 0
       E2 = 10.0 * NROW * EPS * EPS
       TOL = EPS * 0.1
       ECR = NCOL
C-----
C-- SET [V] As [I]
C-----
       DO 20 I = 1,NCOL
       DO 20 J = 1,NCOL
       VD(I,J) = 0.0
       VD(I,I) = 1.0
  20 CONTINUE
C-----
C-- START CALCULATE SVD
C-----
  25 NROT = ECR * (ECR-1)/2
       NSWEEP = NSWEEP + 1
       DO 100 J = 1,ECR-1
       DO 100 K = J+1,ECR
       P = 0.0
       Q = 0.0
       R = 0.0
       DO 30 I = 1,NROW
       X0 = US(I,J)
       Y0 = US(I,K)
       P = P + X0*Y0
       Q = Q + X0*X0
       R = R + Y0*Y0
  30 CONTINUE
       S2D(J) = Q
       S2D(K) = R
       IF (Q.LT.R) GO TO 70
       IF (Q.LE.E2*S2D(1).OR.ABS(P).LE.TOL*Q) GO TO 50
       P = P/Q
       R = 1 - R/Q
       VT = SQRT(4*P*P+R*R)
       CO = SQRT(0.5*(1+R/VT))
       SO = P/(VT*CO)
       DO 40 I = 1,NROW
       D1 = US(I,J)
       US(I,J) = CO*D1 + SO*US(I,K)
       US(I,K) = -SO*D1 + CO*US(I,K)
  40 CONTINUE
       DO 45 I = 1,NCOL
       D1 = VD(I,J)
```

```
      VD(I,J) = CO*D1 + SO*VD(I,K)
      VD(I,K) = -SO*D1 + CO*VD(I,K)
   45 CONTINUE
      GO TO 60
   50 NROT = NROT - 1
   60 GO TO 90
   70 P = P/R
      Q = Q/R - 1
      VT = SQRT(4*P*P+Q*Q)
      SO = SQRT(0.5*(1-Q/VT))
      IF(P.LT.0) SO = -SO
      CO = P/(VT*SO)
      DO 81 I = 1,NROW
      D1 = US(I,J)
      US(I,J) = CO*D1 + SO*US(I,K)
      US(I,K) = -SO*D1 + CO*US(I,K)
   81 CONTINUE
      DO 85 I = 1,NCOL
      D1 = VD(I,J)
      VD(I,J) = CO*D1 + SO*VD(I,K)
      VD(I,K) = -SO*D1 + CO*VD(I,K)
   85 CONTINUE
   90 CONTINUE
  100 CONTINUE
      WRITE(6,105) NSWEEP,NROT
  105 FORMAT (2X,'END OF SWEEP #',I2,' NO OF ROTATION',
     1        ' PERFORMED : ',I4)
  110 IF(ECR.LT.3.OR.S2D(ECR).GT.(S2D(1)*TOL+TOL*TOL)) GO TO 120
      ECR = ECR - 1
      GO TO 110
  120 CONTINUE
      IF(NROT.NE.0.AND.NSWEEP.LE.SLIMIT) GO TO 25
  130 CONTINUE
      IF (NSWEEP.GT.SLIMIT) WRITE (6,140)
  140 FORMAT (/2X,'SWEEP LIMIT EXCEEDED',/2X)
C-----
C-- END OF SVD CALCULATION
C-----
      WRITE (6,143) EPS
  143 FORMAT (/2X,'TOLERANCE : ',F12.10)
      IF (MSVD-1) 180,160,170
  160 WRITE (6,161)
  161 FORMAT (/2X,'[U] [S]')
      CALL COEMR (US,NROW,NCOL)
      GO TO 180
  170 WRITE (6,171)
  171 FORMAT (/2X,'[V]')
      CALL COEMR (VD,NCOL,NCOL)
  180 WRITE (6,181)
  181 FORMAT(/2X,'SINGULAR VALUE')
      DO 185 I = 1,NCOL
  185 SD(I) = SQRT(S2D(I))
      CALL COEMR (SD,NCOL,1)
      COND = SD(1)/SD(NCOL)
      WRITE (6,187) COND
  187 FORMAT(/2X,'CONDITION NUMBER : ',F14.6)
C-----
c-- IF INTERIOR CASE, PERFORMED EFI METHOD
C-----
C     IF (IDD.EQ.1) CALL EFI (US,S2D,NROW,NCOL)
C-----
C-- FIND RANK DEFICIENT
C-- TOLERANCE : SMIN * 15 > SMAX
C-----
C     call coemr(v,ncol,ncol)
C     call coemr(s2d,ncol,1)
C     call coemr(us,nrow,ncol)
C     TOLQ = SVDQ * SVDQ
C     Q = S2D(1)/TOLQ
C     NRD = 0
C     DO 190 I = 1,NCOL
```

```
C      IF (S2D(I).LE.Q) NRD = NRD + 1
C  190 CONTINUE
C      IF (NRD.EQ.0) GO TO 200
C      WRITE (6,195) NRD
C  195 FORMAT (/2X,'RANK DEFICIENT: ',I3,' NO UNIQUE SOLUTION EXIST')
C      GO TO 300
C-----
C-- START LEAST SQUARES SOLUTION
C-----
                ALP=0.9
  200 DO 220 I = 1,NCOL
      XS = 0.0
      DO 210 J = 1,NCOL
      DO 210 K = 1,NROW
      XS = XS + VD(I,J)*US(K,J)*YD(K)/S2D(J)
C             XS=XS+(S2D(J)*VD(I,J)*US(K,J)*YD(K))/(S2D(J)*(S2D(J)+ALP))
  210 CONTINUE
      XD(I) = XS
  220 CONTINUE
  300 RETURN
      END
C=1=================================================================
      SUBROUTINE EFI (US,S2D,NROW,NCOL)
C=====================================================================
C-----
C-- TO CALCULATE THE MAGNITUDE OF ROW VECTORS [U]
C-----
      DIMENSION US(NROW,NCOL),S2D(NCOL)
      WRITE (6,5)
    5 FORMAT (/2X,'EFFECTIVE INDEPENDENCE METHOD WAS PERFORMED',
     1       /2X,'ROW',6X,'EFI')
      DO 10 I = 1,NROW
      SUM = 0.0
      DO 20 J = 1,NCOL
      SUM = SUM + US(I,J)*US(I,J)/S2D(J)
   20 CONTINUE
      WRITE (6,30) I,SUM
   30 FORMAT (2X,I3,2X,F10.6)
   10 CONTINUE
      RETURN
      END
C=====================================================================
      SUBROUTINE MTM (AD,BD,CD,NROW,NCOL,ND,SG)
C=====================================================================
C-----
C-- TO CALCULATE [CD] = [CD] + [AD] [BD]
C-- THEY CAN BE A VECTOR
C-----
      DIMENSION AD(NROW,ND),BD(ND,NCOL),CD(NROW,NCOL)
      COMPLEX AD,BD,CD
C--
      DO 10 I = 1,NROW
      DO 10 J = 1,NCOL
      DO 10 K = 1,ND
      CD(I,J) = CD(I,J) + SG * AD(I,K) * BD(K,J)
   10 CONTINUE
      RETURN
      END
C=1=================================================================
      SUBROUTINE COEM (DM,NROW,NCOL)
C=====================================================================
C-----
C-- TO PRINT MATRIX COEFICIENT
C-----
      DIMENSION DM(NROW,NCOL)
      COMPLEX DM
      JB = 1
      JE = 5
   10 IF (JE.GT.NCOL) JE = NCOL
      WRITE (6,20) (J,J = JB,JE)
   20 FORMAT(/5X,5(13X,I3,11X),/2X)
```

```
      DO 30 I = 1,NROW
    30 WRITE(6,40) I,(DM(I,L),L = JB,JE)
    40 FORMAT(2X,I3,5(2X,G12.6,1X,G12.6))
      IF (JE.EQ.NCOL) GO TO 50
      JB = JE + 1
      JE = JB + 4
      GO TO 10
    50 RETURN
      END
C=1====================================================================
      SUBROUTINE COEMR (DM,NROW,NCOL)
C======================================================================
C-----
C-- TO PRINT MATRIX COEFICIENT
C-----
      DIMENSION DM(NROW,NCOL)
      IF (NCOL.EQ.1) GO TO 50
      JB = 1
      JE = 10
    10 IF (JE.GT.NCOL) JE = NCOL
      WRITE (6,20) (J,J = JB,JE)
    20 FORMAT(/3X,10(6X,I3,4X))
      DO 30 I = 1,NROW
    30 WRITE(6,40) I,(DM(I,L),L = JB,JE)
    40 FORMAT(I3,10(2X,F11.6))
      IF (JE.EQ.NCOL) GO TO 80
      JB = JE + 1
      JE = JB + 9
      GO TO 10
    50 JB = 1
      JE = 7
    60 IF (JE.GT.NROW) JE = NROW
      WRITE (6,70) (I,DM(I,1),I = JB,JE)
    70 FORMAT (8(2X,I3,1X,G12.6,'|'))
      IF (JE.EQ.NROW) GO TO 80
      JB = JE + 1
      JE = JB + 6
      GO TO 60
    80 RETURN
      END
C-------------------------------------------------------------------------------------
C-TAMBAHAN DARWIN---------------------------------------------------------------------
C-------------------------------------------------------------------------------------
C======================================================================
      SUBROUTINE SVDT (AD,BD,CD,DD,DRM,G,NR2,NC2,NR,NC,NRM,NCM,IAP,
     1              YD,XD,V,SV,S2,S)
C======================================================================
C-----
C-- TO SOLVE [AD] {BD} = {CD}
C-----
      DIMENSION AD(NR,NC),BD(NC),CD(NR),DD(NR,NC),DRM(NRM,NCM),
     1G(NR2,NC2),YD(NR2),XD(NC2),V(NC2,NC2),SV(NC2,10),S2(NC2),
     2S(NC2)
      COMPLEX AD,BD,CD,DD
C-----
C-- CHANGE TO REAL
C-----
      DO 5 I1 = 1,NR
      I2 = NR + I1
      YD(I1) = REAL(CD(I1))
      YD(I2) = AIMAG(CD(I1))
      DO 5 J1 = 1,NC
      J2 = NC + J1
      G(I1,J1) = REAL(AD(I1,J1))
      G(I1,J2) = -AIMAG(AD(I1,J1))
      G(I2,J1) = AIMAG(AD(I1,J1))
      G(I2,J2) = REAL(AD(I1,J1))
    5 CONTINUE
      IF (IAP.EQ.1) GO TO 25
C-----
C-- NOW [G] IS [AD] IN REAL
```

```
C-- FIND [G] = [G] + [DD] {DRM}
C-----
      DO 10 I1 =1,NR
      I2 = NR + I1
      DO 10 K = 1,NC2
      DO 20 J1 = 1,NC
      J2 = NC + J1
      G(I1,K) = G(I1,K) + REAL(DD(I1,J1))*DRM(J1,K)
     1           - AIMAG(DD(I1,J1))*DRM(J2,K)
      G(I2,K) = G(I2,K) + AIMAG(DD(I1,J1))*DRM(J1,K)
     1          + REAL(DD(I1,J1))*DRM(J2,K)
   20 CONTINUE
   10 CONTINUE
C-----
C-- NOW WE HAVE [G] [XD] = [YD]
C-- INPUT : G,YD,NR2,NC2
C-- OUTPUT: XD,V,S2,S,NRD
C-----
   25 CALL SVDSOL (G,XD,YD,V,S2,S,NR2,NC2,NRD)
C-----
C-- SEKARANG SUDAH DIDAPAT G,XD,YD, CARI ALPHA!
C-- INPUT : G,XD,YD,NCOL,NROW
C-- OUTPUT: ALP
C-----
c              CALL ALPHA (G,XD,YD,ALP,NR2,NC2)
c              IF (NRD.EQ.0) GO TO 40
C-----
C-- PRINT SINGULAR VECTORS OF [G]
C-----
      WRITE (6,30)
   30 FORMAT (/2X,' SINGULAR VECTORS ')
      DO 33 I = 1,NC2
      DO 33 J = 1,NRD
      K = NC2 - NRD + J
      SV(I,J) = V(I,K)
   33 CONTINUE
      CALL COEMR (SV,NC2,NRD)
      IF (IAP.EQ.1) GO TO 39
      WRITE (6,35)
   35 FORMAT (/2X,'[A+] [B] IN REAL')
      CALL COEMR (DRM,NRM,NCM)
      WRITE (6,37)
   37 FORMAT (/2X,'USE EXEQ.F TO FIND EXTRA EQUATIONS',/2X)
   39 STOP
C-----
C-- CONTINUE
C-----
   40 DO 45 I1 = 1,NC
      I2 = NC + I1
      BD(I1) = CMPLX(XD(I1),XD(I2))
   45 CONTINUE
C       call coem (bd,nc,1)
      IF (IAP.NE.1) GO TO 60
C-----
C-- FIND GENERAL INVERS OF [G] = [DRM]
C-----
C       call coemr (g,nr2,nc2)
C       call coemr (v,nc2,nc2)
C       call coemr (s2,nc2,1)
      DO 50 I = 1,NC2
      DO 52 J = 1,NR2
      DRM(I,J) = 0.0
      DO 54 K = 1,NC2
      DRM(I,J) = DRM(I,J) + V(I,K) * G(J,K) / S2(K)
   54 CONTINUE
   52 CONTINUE
   50 CONTINUE
   60 RETURN
      END
```

```
C*************************************************************************************************
C**              INVERSE ACOUSTIC THREE DIMENSION AXYSIMETRIC                                   **]
C**             MAXIMUM NUMBER OF NODE (NMAX) = 150                                             **]
C**             MAXIMUM NUMBER OF ELEMEN (MMAX) = 69                                            **]
C*************************************************************************************************
      COMPLEX*16 A,BTN,B,E,D,PFP(146),FPP(146)
      COMPLEX*16 EPHI(146),EPHIF(146)
      DIMENSION A(160,146),B(160,146),E(160,146),D(160,146),BTN(146)
      DIMENSION NODE(3,146)
      REAL*8 SHAP(3),SHAP1(3),SI
      REAL*8 X(2,146),XCP(2,10),XFP(2,160)
      REAL*8 Y(32),W(32),SHAPF(3,32),SHAPF1(3,32)
      COMMON/ANGLE/NGG/SUBDV/NS
      COMMON/DOMA/IHF
C=======================================================================
C     MAIN PROGRAM
C=======================================================================
      OPEN (5,FILE='INPUT.TXT',STATUS='OLD')
      OPEN (6,FILE='ELEMEN.TXT',STATUS='NEW')
      NMAX = 146
      MMAX = NMAX/2
C--
      CALL INPDAT (NMAX,N,MMAX,M,X,NODE,XCP,XFP,NCP,NFP,FPP,
     1           NPCH,ISEVCP,ISEVFP,NTL,MTL,NDEL)
C--
C--
      NG = 2
      NGG = 2
      NS = 32
      NGCH = 2
      IF(NG.LT.2.OR.NG.GT.10)  NG = 2
      IF ( (NGCH .LT. 2) .OR. (NGCH .GT. 10) ) NGCH = 2
      CALL GAUSS (NG,Y,W)
C--
C--

      IF (IHF.EQ.0) THEN
      CALL COEFC2(N,NPCH,M,A,B,NG,BTN,E,D,PFP,X,XCP,XFP,NCP,
     1           NFP,NODE,Y,W,SHAPF,SHAPF1,EPHI,EPHIF,FPP,NTL,MTL)
      ENDIF

      IF (IHF.EQ.1 .OR. IHF.EQ.2) THEN
      CALL COEFC2(N,NPCH,M,A,B,NG,BTN,E,D,PFP,X,XCP,XFP,NCP,
     1           NFP,NODE,Y,W,SHAPF,SHAPF1,EPHI,EPHIF,FPP,NTL,MTL)
      ENDIF
      IF (IHF.EQ.3) THEN
      CALL COEFC2(N,NPCH,M,A,B,NG,BTN,E,D,PFP,X,XCP,XFP,NCP,
     1           NFP,NODE,Y,W,SHAPF,SHAPF1,EPHI,EPHIF,FPP,NTL,MTL)
      ENDIF
      CALL SHAPE (SI,SHAP,SHAP1,NGCH,Y,W)
C--
      NPCT2 = NPCH*2
      NFPT2 = NFP*2
      NT2   = N * 2
C--
      CALL SOLVE (A,B,BTN,E,D,PFP,NFP,NPCH,N,NPCT2,NFPT2,NT2)
C--
      END
C=======================================================================
      SUBROUTINE INPDAT (NMAX,N,MMAX,M,X,NODE,XCP,XFP,NCP,NFP,FPP,
     1                NPCH,ISEVCP,ISEVFP,NTL,MTL,NDEL)
C=======================================================================
C-----
C-- TO READ AND OUTPUT THE DATA INPUT FILE
C-----
C- ONE DIMENSION
      DIMENSION XDC(2),XDF(2),FPP(146)
C- TWO DIMENSION
      COMMON/DOMA/IHF
      COMMON/HALF/RH
      COMMON/IMAGE/RB
```

```
      COMMON/OMEGA/WN
      COMMON/IMP/ZO
      COMMON/REFP/PO
      COMMON/FREQ/KFREQ
      COMMON/IDEN/IDD
      COMMON/SCAT/ISC,DIRR,DIRZ,AMP
      DIMENSION NODE(3,MMAX)
      REAL*8 RB,ZO,C,RHO
      REAL*8 DIRR,DIRZ,AMP
      REAL*8 X(2,NMAX),XCP(2,10),XFP(2,160),R,RMAX,RRMAX
      REAL*8 XDC,XDF,PRE,PIM
 1030 FORMAT ( 5X, I3, 5X, 2F15.5 )
 1035 FORMAT ( /1X, 'GLOBAL NODE NUMBERS CORRESPONDING TO INDICATED',
     *            ' LOCAL NODE NUMBERS' / )
 1037 FORMAT ( 5X, 'SEGMENT', 4X, 'LOCAL  NODE # : ', 12X, 3(I1,4X) /)
 1060 FORMAT ( 6X, I3, 30X, 3( 2X,I3 ) )
 1070 FORMAT ( 1H0, 'NODE(', I1, ',', I3, ') IMPROPERLY ',
     *                          'SPECIFIED : ', I5 )
1116 FORMAT (3X,'DIRECTION COSINES OF THE INCIDENT WAVE :',2(F8.4,2X))
 1117 FORMAT (3X,'AMPLITUDO OF THE INCIDENT WAVE : ',1X,G12.6)
 1123 FORMAT (3X,'REFERENCE PRESSURE :',1X,F9.5)
 1121 FORMAT (3X,'POSITION OF THE POINT SOURCE :',2(F8.4,2X))
 1122 FORMAT (3X,'POWER THE POINT SOURCE :',1X,G12.6)
 1090 FORMAT ( / )
 1921 FORMAT (5X,I3,4X,2F25.15,5X,2F25.15)
 2030 FORMAT('0 * * * * *   E R R O R   * * * * *   N IMPROPERLY ',
     1       'SPECIFIED:  ',I3)
 2035 FORMAT('0 * * * * *   E R R O R   * * * * *   M IMPROPERLY ',
     1      'SPECIFIED:  ',I3)
      CHARACTER*4 DATMOD
      CHARACTER*40 TITLE
      COMPLEX*16 FPP,WN,RH
C--
      WRITE (6,53)
   53 FORMAT (/72(1H=)/12X,'I N P U T   D A T A   F I L E',/
     1       72(1H=))
  100 READ (5,110) DATMOD
  110 FORMAT (A4,15X)
      IF (DATMOD.EQ.'TITL') GO TO 1000
      IF (DATMOD.EQ.'HALF'.OR.DATMOD.EQ.'FULL') GO TO 900
      IF (DATMOD.EQ.'EXTE'.OR.DATMOD.EQ.'INTE') GO TO 1100
      IF (DATMOD.EQ.'PLAN'.OR.DATMOD.EQ.'POIN') GO TO 1200
      IF (DATMOD.EQ.'PROP') GO TO 1400
      IF (DATMOD.EQ.'NODE') GO TO 1500
C     IF (DATMOD.EQ.'CHIE') GO TO 1800
      IF (DATMOD.EQ.'FIEL') GO TO 1900
      IF (DATMOD.EQ.'ENDE') GO TO 2000
C-----
C- READ TITLE
C-----
 1000 READ (5,1010) TITLE
 1010 FORMAT (A40)
      WRITE (6,1020) TITLE
 1020 FORMAT (//2X,10(1H-),' TITLE ',10(1H-)/2X,A40)
      GO TO 100
C-----
C- IDENTIFY SPACE
C-----
  900 IHF = 0
      READ (5,*) RB,RRH,GIRH,REALWN,WNIMAG,DFREQ,NFREQ
      RH = CMPLX(RRH,GIRH)
      IF (DATMOD.EQ.'FULL') GO TO 930
      IF (RB.EQ.0.0) GO TO 950
      IF (DATMOD.EQ.'HALF') GO TO 940
  930 WRITE (6,935)
  935 FORMAT (2X,'***** FULL SPACE PROBLEM *****')
      GO TO 946
  950 IHF = 2
      WRITE (6,955)
  955 FORMAT (2X,'*****  CONTACT  PROBLEM  *****')
      GO TO 946
```

```
 940 IHF = 1
     WRITE (6,945)
 945 FORMAT (2X,'***** HALF SPACE PROBLEM *****')
 946 WNN = CMPLX(REALWN,WNIMAG)
     DFREQ=0.0
     NFREQ=0
     IF (DFREQ.LE.0.0) NFREQ=1
     IF (NFREQ.GT.10) NFREQ=10
     DO 111 KFREQ = 1,NFREQ
     WN = WNN + CMPLX(((KFREQ-1)*DFREQ),0.0)
     WRITE(*,*)
     WRITE(*,*) 'NOW EXECUTING WAVE NUMBER = ',WN
 111 CONTINUE
 990 GO TO 100
C-----
C- IDENTIFY THE PROBLEM
C-----
1100 ISC = 0
     PSYM = 0
     IF (DATMOD.EQ.'INTE') GO TO 1120
     WRITE (6,1110)
1110 FORMAT (2X,'***** EXTERIOR PROBLEM *****')
     IDD = 0
     GO TO 1140
1120 WRITE (6,1130)
1130 FORMAT (2X,'***** INTERIOR PROBLEM *****')
     IDD = 1
1140 IF (PSYM.EQ.0) GO TO 1180
     IF (PSYM.GT.0.AND.PSYM.LE.3) GO TO 1160
     WRITE (6,1150)
1150 FORMAT (//2X,'!!!!! ERROR --- PSYM > 3 !!!!!',/2X)
     STOP
1160 WRITE (6,1170) PSYM
1170 FORMAT (2X,'USING SIMETRY',/2X,
    1     'PLANE OF SIMETRY PERPENDICULAR TO X(',I1,') AXIS')
1180 GO TO 100
C-----
C- SPECIFY THE INCIDENT WAVE IN SCATTERING PROBLEM
C- ISC = 1 : PLANE WAVE; DIRR,DIRZ : THE DIRECTION COSINES
C- ISC = 2 : POINT SOURCE; DIRR,DIRZ : LOCATION OF THE POINT SOURCE
C-----
1200 IF (DATMOD.EQ.'PLAN') ISC = 1
     IHF = 2
     IF (DATMOD.EQ.'POIN') ISC = 2
     IF (DATMOD.EQ.'POIN'.AND.IHF.EQ.2) IHF = 2
     READ (5,*) DIRR,DIRZ,AMP
     IF (ISC.EQ.1) THEN
     WRITE(6,*) '  PLANE WAVE'
     WRITE(6,1116) DIRR,DIRZ
     WRITE(6,1117) AMP
     ENDIF
     IF (ISC.EQ.2) THEN
     WRITE(6,*)   '  POINT SOURCE'
     WRITE(6,1121) DIRR, DIRZ
     WRITE(6,1122) AMP
     ENDIF
     GO TO 100
C
C-----
C- PRESCRIBE   THE FREQUENCY RANGE
C-----
C-----
C- SPECIFY THE PROPERTY OF AIR
C-----
1400 READ (5,*) C, RHO, PO
     ZO = RHO*C
     IF (ISC.EQ.1.OR.ISC.EQ.2) GO TO 1320
     WRITE (6,1310)
1310 FORMAT (2X'***** RADIATION PROBLEM *****')
     GO TO 1360
1320 WRITE (6,1330)
```

```
 1330 FORMAT (2X'***** SCATTERING PROBLEM *****')
 1360 WRITE (6,1410) C
 1410 FORMAT (/2X,'SPEED OF SOUND    : ',F10.4)
      PI = 3.141592653589793D0
      F = C*WN/(2*PI)
 1300 IF (IDD.EQ.1) GO TO 1340
 1340 WRITE (6,1350) F
 1350 FORMAT (2X,'FREQUENCY        : ',F10.4)
      WRITE (6,1420) WN
 1420 FORMAT (2X,'WAVE NUMBER      : ',F10.5)
      WRITE (6,1430) RHO
 1430 FORMAT (2X,'DENSITY          : ',F10.4)
      WRITE (6,1440) PO
 1440 FORMAT (2X,'REFERENCE PRESSURE: ',F10.5)
  910 WRITE (6,920) IHF
  920 FORMAT (2X,'IHF              : ',I1)
C      WRITE (6,1114) RB
c      WRITE (6,1115) RH
 1450 CONTINUE
      GO TO 100
C-----
C- READ THE NODAL COORDINATES
C-----
 1500 WRITE (6,1510)
 1510 FORMAT (//2X,10(1H-),' NODAL COORDINATES ',10(1H-),
     1       //2X,'NODE',7X,'X(1)',9X,'X(2)')
      CALL COORD1( NMAX, N, X,0 )
      NTL=N


C
      IF(IHF.EQ.2 .OR. IHF.EQ.3) CALL COORD1(NMAX,NTL,X,N)
      IF(NTL.LE.NMAX.AND.NTL.GT.0)  GO TO 120
              WRITE (6,2030) NTL
      STOP
  120 CONTINUE
      RMAX = 0.D0
      DO 50 I=1,NTL
      DO 51 J=1,2
      R = DABS( X(J,I) )
      IF (R .GT. RMAX) RMAX = R
   51 CONTINUE
   50 CONTINUE
      RRMAX = 1.D0/RMAX
      DO 70 K=1,MMAX
      DO 71 J=1,3
      NODE(J,K) = 0
   71 CONTINUE
   70 CONTINUE

C
C  SPECIFY SEGMENT CONFIGURATION
C

              IF (IHF.EQ.3) GO TO 58
C
      M=N/2
      IF(IHF.EQ.0 .OR. IHF.EQ.1 ) CALL CONFIG(NODE,N,M,MMAX)
      MTL = M
      IF(IHF.EQ.2) CALL CONFIG(NODE,NTL,MTL,MMAX)
      GO TO 59
C
   58 M = N/2 - 1
      MTL = NTL/2 - 1
      CALL CONFIG(NODE,NTL,MTL,MMAX)
C
   59 IF ((MTL .LE. MMAX) .AND. (MTL .GT. 0)) GO TO 60
      WRITE (6,2035) MTL
      STOP
   60 CONTINUE
      DO 80 K=1,MTL
```

```
      DO 75 I=1,3
      IF ( (NODE(I,K) .GT. 0) .AND. (NODE(I,K) .LE. NTL) )
     *        GO TO 75
      WRITE (6,1070) I, K, NODE(I,K)
      STOP
   75 CONTINUE
   80 CONTINUE
C
      NDEL = 1
      IF (NDEL.EQ.0) GO TO 1590
C
C     WRITE (6,1000)
C     WRITE (6,1510)
      DO 10 J=1,NTL
      WRITE (6,1030) J,(X(I,J), I=1,2)
   10 CONTINUE
      WRITE (6,1090)
      WRITE (6,1035)
      WRITE (6,1037) ( J,J=1,3)
      DO 30 K=1,MTL
      WRITE (6,1060) K, ( NODE(J,K),J=1,3 )
   30 CONTINUE
      DO 35 K=M+1,MTL
      WRITE (6,1060) K, ( NODE(J,K),J=1,3 )
   35 CONTINUE
      WRITE (6,1090)
C=================================
 1590 CONTINUE
 1591 GO TO 100
C-----
C- READ THE CHIEF POINT COORDINATES
C-----
 1800 WRITE (6,1810)
 1810 FORMAT (//2X,10(1H-),' CHIEF POINT ',10(1H-),//2X,'POINT',
     1  6X,'X(1)',9X,'X(2)')
      NCMAX = 15
      I = 0
 1820 I = I + 1
      READ (5,*) IC,(XDC(J),J=1,2)
      IF (IC.EQ.0) GO TO 1860
      IF (IC.LE.NCMAX.AND.IC.EQ.I) GO TO 1840
      WRITE (6,1830) I,IC
 1830 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'CHIEF POINT',
     1  ' NUMBER (',I3,') IMPROPERLY SPECIFIED (',I3,')')
      STOP
 1840 DO 1850 J=1,2
 1850 XCP(J,IC) = XDC(J)
      GO TO 1820
 1860 NCP = I - 1
      ISEVCP = XDC(1)
      DO 1880 IC = 1,NCP
      WRITE (6,1870) IC,(XCP(J,IC),J=1,2)
 1870 FORMAT(3X,I3,1X,3(3X,F20.15))
 1880 CONTINUE
      IF (ISEVCP.EQ.0) GO TO 1894
      IF (ISEVCP.GT.0.AND.ISEVCP.LE.8) GO TO 1890
      WRITE (6,1882) ISEVCP
 1882 FORMAT(//12X,'!!!!! E R R O R !!!!!',/3X,'IMPROPER DEGREE',
     1       ' OF SEVERITY SPECIFIED FOR CHIEF POINT  : ',I2)
      STOP
 1890 WRITE (6,1892) ISEVCP
 1892 FORMAT(/2X,'USING NUMBER OF SEVERITY : ',I2,
     1       ' FOR ALL CHIEF POINT')
 1894 GO TO 100
C-----
C- READ THE FIELD POINT COORDINATES AND THE KNOWN PRESSURE
C-----
 1900 NPCH = N + NCP
      WRITE (6,1910)
 1910 FORMAT (//2X,10(1H-),' FIELD POINT ',10(1H-),//56X,'PRESSURE',
     1        /2X,'POINT',5X,'X(1)',7X,'X(2)',7X,'REAL',
```

```
     2       5X,'IMAGINER')
      NFMAX = 200
      I = 0
 1920 I = I + 1
      READ (5,1921) IF,(XDF(J),J=1,2),PRE,PIM
      IF (IF.EQ.0) GO TO 1960
      IF (IF.LE.NFMAX.AND.IF.EQ.I) GO TO 1940
      WRITE (6,1930) I,IF
 1930 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'FIELD POINT',
     1  ' NUMBER (',I3,') IMPROPERLY SPECIFIED (',I3,')')
      STOP
 1940 DO 1950 J=1,2
 1950 XFP(J,IF) = XDF(J)
      FPP(IF) = CMPLX(PRE,PIM)
      GO TO 1920
 1960 NFP = I - 1
      ISEVFP = XDF(1)
      DO 1980 IF = 1,NFP
      WRITE (6,1970) IF, (XFP(J,IF),J=1,2), FPP(IF)
 1970 FORMAT (3X,I3,1X,4(3X,F25.15))
 1980 CONTINUE
      IF (ISEVFP.EQ.0) GO TO 1994
      IF (ISEVFP.GT.0.AND.ISEVFP.LE.8) GO TO 1990
      WRITE (6,1982) ISEVFP
 1982 FORMAT (//12X,'!!!!! E R R O R !!!!!',/3X,'IMPROPER DEGREE',
     1       ' OF SEVERITY SPECIFIED FOR FIELD POINT : ',I2)
      STOP
 1990 WRITE (6,1992) ISEVFP
 1992 FORMAT(/2X,'USING DEGREE OF SEVERITY : ',I2,
     1       ' FOR ALL FIELD POINT')
 1994 IF (NFP.GE.N) GO TO 1998
      WRITE (6,1997)
 1997 FORMAT (//2X,'!!!!! ERROR --- ',
     1       ' TOO SEDIKIT FIELD POINT SPECIFIED !!!!!',/2X)
 1998 GO TO 100
C-----
C- BACK TO MAIN PROGRAM
C-----
 2000 WRITE (6,2010)
 2010 FORMAT (/72(1H=)/12X,'END OF INPUT DATA FILE',/
     1       72(1H=))
      RETURN
      END
C=====================================================================
      SUBROUTINE COORD1(NMAX,N,X,IS)
C=====================================================================
      REAL*8 X(2,NMAX),XR(2)
 1000 FORMAT (11X,I3,2(5X,F20.15))
 1010 FORMAT (1H0,'NODE NUMBER IMPROPERLY SPECIFIED:',I5/)
C
C  SPECIFY NODAL COORDINATES
C
      I = IS
   10 I = I + 1
      READ (5,*) II,(XR(J),J=1,2)
      IF (II.EQ.0) GO TO 40
      IF (II.LE.NMAX) GO TO 20
              WRITE (6,1010) II
      STOP
   20 CONTINUE
              X(1,II) = XR(1)
      X(2,II) = XR(2)
      GO TO 10
   40 N = I - 1
      RETURN
      END
C
C=====================================================================
      SUBROUTINE CONFIG(NODE,NTL,MTL,MMAX)
C=====================================================================
      DIMENSION NODE(3,MMAX)
```

```
      COMMON/DOMA/IHF
C
      IF (IHF.EQ.3) GO TO 38
C
      MTL = NTL/2
      DO 30 K = 1,MTL
      DO 31 J = 1,3
      NODE(J,K) = K*2 + J - 2
   31 CONTINUE
   30 CONTINUE
      IF (2*MTL.EQ.NTL) NODE(3,MTL)=1
      GO TO 47
C
   38 DO 40 K = 1,1
      DO 41 J = 1,3
      NODE(J,K) = K*2 + J - 2
   41 CONTINUE
   40 CONTINUE
      DO 44 K = 2,Mtl
      DO 45 J = 1,3
      NODE(J,K) = K*2 + J - 1
   45 CONTINUE
   44 CONTINUE
      IF (2*MTL.EQ.NTL) NODE(3,MTL)=1
C
   47 CONTINUE
      RETURN
      END
C=====================================================================
      SUBROUTINE GAUSS (NG,Y,W)
C=====================================================================
C
      REAL*8 Y(NG),W(NG)
C
 1000 FORMAT (' ****** ERROR *****  NO GAUSS DATA FOR NG = ', I2 /)
C
      IF (NG.GT.10) WRITE (6,1000) NG
      GO TO (1,2,3,4,5,6,7,8,9,10,32),NG
C
    1 WRITE (6,1000) NG
      STOP
C
    2 Y(1) = -0.5773503D0
      Y(2) = - Y(1)
      W(1) = 0.5D0
      W(2) = 0.5D0
      RETURN
C
    3 Y(1) = -0.7745967D0
      Y(2) = 0.0D0
      Y(3) = - Y(1)
      W(1) = 0.55555556D0 / 2.0D0
      W(2) = 0.88888889D0 / 2.0D0
      W(3) = W(1)
      RETURN
C
    4 Y(1) = -0.8611363D0
      Y(2) = -0.3399810D0
      Y(3) = - Y(2)
      Y(4) =- Y(1)
      W(1) = 0.1739274D0
      W(2) = 0.3260726D0
      W(3) =  W(2)
      W(4) = W(1)
      RETURN
C
    5 Y(1) = -0.9061798
      Y(2) = -0.5384693
      Y(3) = 0.0
      Y(4) = - Y(2)
      Y(5) = -Y(1)
```

```
      W(1) = 0.1184634
      W(2) = 0.2393143
      W(3) = 0.2844444
      W(4)=W(2)
      W(5) = W(1)
      RETURN
C
    6 Y(1) = -0.9324695
      Y(2) = -0.6612094
      Y(3) = -0.2386192
      Y(4) =-Y(3)
      Y(5) = -Y(2)
      Y(6) = -Y(1)
      W(1) = 0.1713245/2.
      W(2) = 0.3607616/2.
      W(3) = 0.4679139/2.
      W(4)=W(3)
      W(5) = W(2)
      W(6) = W(1)
      RETURN
C
    7 Y(1) = -.9491079
      Y(2) = -.7415312
      Y(3) = -.4058452
      Y(4) = 0
      Y(5) = - Y(3)
      Y(6) = - Y(2)
      Y(7) = - Y(1)
      W(1) = .12948497/2
      W(2) = .27970539/2
      W(3) = .38183005/2
      W(4) = .41795918/2
      W(5) = W(3)
      W(6) = W(2)
      W(7) = W(1)
      RETURN
C
    8 Y(1) = -0.9602899
      Y(2) = -0.7966665
      Y(3) = -0.5255324
      Y(4) = -0.1834346
      Y(5) = -Y(4)
      Y(6) = -Y(3)
      Y(7) = -Y(2)
      Y(8) = -Y(1)
      W(1) = 0.1012285/2.
      W(2) = 0.2223810/2.
      W(3) = 0.3137066/2.
      W(4) = 0.3626839/2.
      W(5) = W(4)
      W(6) = W(3)
      W(7) = W(2)
      W(8) = W(1)
      RETURN
C
    9 Y(1) = -0.9681602
      Y(2) = -0.8360311
      Y(3) = -0.6133714
      Y(4) = -0.3242534
      Y(5) = 0.0
      Y(6) = -Y(4)
      Y(7) = -Y(3)
      Y(8) = -Y(2)
      Y(9) = -Y(1)
      W(1) = 0.0812744/2.
      W(2) = 0.1806482/2.
      W(3) = 0.2606127/2.
      W(4) = 0.3123471/2.
      W(5) = 0.3302394/2.
      W(6) = W(4)
      W(7) = W(3)
```

```
      W(8) = W(2)
      W(9) = W(1)
      RETURN
C
   10 Y(1) = -.9739065
      Y(2) = -.8650634
      Y(3) = -.6794096
      Y(4) = -.4333954
      Y(5) = -.1488743
      Y(6) = - Y(5)
      Y(7) = - Y(4)
      Y(8) = - Y(3)
      Y(9) = - Y(2)
      Y(10) = - Y(1)
      W(1) = .06667134/2
      W(2) = .14945135/2
      W(3) = .21908636/2
      W(4) = .26926672/2
      W(5) = .29552422/2
      W(6) = W(5)
      W(7) = W(4)
      W(8) = W(3)
      W(9) = W(2)
      W(10) = W(1)
C
   32 Y(1)  = -0.99726386184948156354498112866504072713854D0
      Y(2)  = -0.98561151154526833540017504463090197863240
      Y(3)  = -0.96476225558750643077381192811827496038890
      Y(4)  = -0.93490607593773968917091913483540932552870
      Y(5)  = -0.89632115576605212396530724371921226847900D0
      Y(6)  = -0.84936761373256997013369300496774253895489D0
      Y(7)  = -0.79448379596794240696309729897042890209547 9D0
      Y(8)  = -0.73218211874028968038742666509126714663027 0D0
      Y(9)  = -0.66304426693021520097511516866323836897702 23D0
      Y(10) = -0.58771575724076232904074547640182685845094012D0
      Y(11) = -0.50689990893222939002374747437782123018028 3700D0
      Y(12) = -0.42135127613063534536411943617242647833587 72886D0
      Y(13) = -0.33186860228212764977991680573018799619577 513681D0
      Y(14) = -0.23928736225213707454460320916550152060885 5421960D0
      Y(15) = -0.14447196158279649348518637359881065220384 59913156D0
      Y(16) = -0.04830766568773831623481257044050216369084 7251730849D0
      Y(17) = -Y(16)
      Y(18) = -Y(15)
      Y(19) = -Y(14)
      Y(20) = -Y(13)
      Y(21) = -Y(12)
      Y(22) = -Y(11)
      Y(23) = -Y(10)
      Y(24) = -Y(9)
      Y(25) = -Y(8)
      Y(26) = -Y(7)
      Y(27) = -Y(6)
      Y(28) = -Y(5)
      Y(29) = -Y(4)
      Y(30) = -Y(3)
      Y(31) = -Y(2)
      Y(32) = -Y(1)
      W(1)  = 0.00701861000947009660040706373 89D0/2.D0
      W(2)  = 0.01627439473090567060517056220 6D0/2.D0
      W(3)  = 0.02539206530926205945575258978 9D0/2.D0
      W(4)  = 0.03427386291302143310268773225 2D0/2.D0
      W(5)  = 0.04283589802222668065687864660 6D0/2.D0
      W(6)  = 0.05099805926237617619616324468 95D0/2.D0
      W(7)  = 0.05868409347853554714528363730 017D0/2.D0
      W(8)  = 0.06582222277636184683765006370 6939D0/2.D0
      W(9)  = 0.07234579410884850622539935647 84878D0/2.D0
      W(10) = 0.07819389578707030647174091882 8306671D0/2.D0
      W(11) = 0.08331192422694675522219907460 434861154D0/2.D0
      W(12) = 0.08765209300440381114277146275 18022875484D0/2.D0
      W(13) = 0.09117387869576388471286857711 1637062544861D0/2.D0
      W(14)=0.09384439908080456563918023766811726003610008D0/2.D0
```

```
      W(15)=0.0956387200792748594190820022041311005948905081 6D0/2.D0
      W(16)=0.0965400885147278005667648300635757947368606312 3557D0/2.D0
      W(17) = W(16)
      W(18) = W(15)
      W(19) = W(14)
      W(20) = W(13)
      W(21) = W(12)
      W(22) = W(11)
      W(23) = W(10)
      W(24) = W(9)
      W(25) = W(8)
      W(26) = W(7)
      W(27) = W(6)
      W(28) = W(5)
      W(29) = W(4)
      W(30) = W(3)
      W(31) = W(2)
      W(32) = W(1)
      RETURN
      END
C=======================================================================
      SUBROUTINE SHAPE (SI,SHAP,SHAP1,NGCH,Y,W)
C=======================================================================
      REAL*8 SHAP(3),SHAP1(3),SI
      REAL*8 Y(NGCH),W(NGCH)
      REAL*8 CSHPF(3,32),CSHPF1(3,32)
C--
      CALL GAUSS(NGCH,Y,W)
      DO 10 IG=1,NGCH
      SI = Y(IG)
      CALL SHFUN( SI, SHAP, SHAP1)
      DO 20 J=1,3
      CSHPF(J,IG) = SHAP(J)
      CSHPF1(J,IG) = SHAP1(J)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END

C=======================================================================
      SUBROUTINE SHFUN(SI,SHAP,SHAP1)
C=======================================================================
      REAL*8 SHAP(3),SHAP1(3),SI
C
      SHAP(1) = .5D0*SI*(-1.D0+SI)
      SHAP(2) = 1.D0 - SI**2
      SHAP(3) = .5D0*SI*(1.D0+SI)
C
      SHAP1(1) = -.5D0 + SI
      SHAP1(2) = -2.D0*SI
      SHAP1(3) = .5D0 + SI
C
      RETURN
      END
C
C=======================================================================
      SUBROUTINE COEFC2(N,NPCH,M,A,B,NG,BTN,E,D,PFP,X,XCP,XFP,NCP,
     1          NFP,NODE,Y,W,SHAPF,SHAPF1,EPHI,EPHIF,FPP,NTL,MTL)
C=======================================================================

      COMPLEX*16 A,B,E,D,BTN,ARG,ARGT,U1,U2,UKERN,UKERN1,TSTAR,RH
      COMPLEX*16 WN,UK,EPHI(N),TSTAR1,JC,PFP(NFP),FPP(NFP),EPHIF(NFP)
      complex*16 UKZ,ARGZ,ARGTZ
      DIMENSION BTN(NPCH)
      DIMENSION NODE(3,MTL),A(NPCH,N),B(NPCH,N),E(NFP,N),D(NFP,N)
      REAL*8 SHAPF(3,NG),SHAPF1(3,NG),DRDN,AR,SI,XCP(2,NCP),XFP(2,NFP)
      REAL*8 X1,X2,PI,SC,WIG,YI,RB
      REAL*8 RS,RS2,Y(NG),W(NG),HUGE,YY(32),WW(32)
      REAL*8 EE,EK,MMDELE,MMDELK,RK,RK2,RR,RR2,DX(3),R,R2,RKP2
      REAL*8 RK2Z,DDX(3),RKP2Z,PRKR,PRRR,PEKRK,PEERK
      REAL*8 GNR,GNRA,GNZA,DIRR,DIRZ,AMP
```

```
      REAL*8 JAC,NSTAR(2),SHAP(3),SHAP1(3),XQ(2),X(2,NTL)
      REAL*8 R2Z,RR2Z,RRZ,RS2Z,RSZ,RKZ,PRKZ,PRRZ,GNZ
      REAL*8 G,GN,EEZ,EKZ,GZ,PRKRZ,PRKZZ,PRRRZ,PRRZZ,PEKRKZ,PEERKZ
      REAL*8 GNRZ,GNZZ,GNRAZ,GNZAZ,RZ,DRDNZ,RRMAX
C
C
      COMMON/H/HUGE/SCALE/RRMAX
      COMMON/B/BIG
      COMMON/OMEGA/WN
      COMMON/IDEN/IDD
      COMMON/SCAT/ISC,DIRR,DIRZ,AMP
      COMMON/ANGLE/NGG/SUBDV/NS
      COMMON / COSBLK / MM, ALPHA
      COMMON/HALF/RH
      COMMON/IMAGE/RB
      COMMON/DOMA/IHF
      IOPT = 1
C
C  PARAMETER DEFINITIONS FOR DCOS(MM*THETA+ALPHA) VARIATION
C
C     MM = 1
C     ALPHA = 0.0
C
C
C EVALUATE SHAPE FUNCTIONS AND DERIVATIVES FOR DISCRETEPOINTS IN
C*INTERIOR OF -1,+1 SEGMENT
C
      PI = 3.141592653589793D0
      PIT4 = PI * 4.D0
      PI2 = PI * 2.D0
      IOPT = 1
      IF (IHF.EQ.1 .OR. IHF.EQ.2) THEN
      MTL = NTL/2
      ENDIF
      IF (IHF.EQ.3) THEN
      MTL = NTL/2 - 1
      ENDIF
      DO 10 IG = 1,NG
      SI = Y(IG)
      CALL SHFUN(SI,SHAP,SHAP1)
      DO 10 J = 1,3
      SHAPF(J,IG) = SHAP(J)
   10 SHAPF1(J,IG) = SHAP1(J)
C
C INITIALIZE "BTN" VECTOR & "A" & "B" MATRIX
C
      DO 20 I = 1,NPCH
      BTN(I) = (0.D0,0.D0)
      DO 20 J = 1,NPCH
      A(I,J) = (0.D0,0.D0)
   20 B(I,J) = (0.D0,0.D0)
C
C INITIALIZE "E" & "D" MATRIX
C
      JC = (0.D0,1.D0)
      DO 25 I = 1,NFP
      PFP(I) = -PIT4*FPP(I)/(JC*WN*415.0)
      DO 25 J = 1,N
      E(I,J) = (0.D0,0.D0)
      D(I,J) = (0.D0,0.D0)
   25 CONTINUE
C
C INCREMENT SEGMENT NUMBER
C
      DO 500 K = 1,MTL
C
C INCREMENT INDICES IDENTIFYING POINTS FOR GAUSSIAN QUADRATURE
C
      DO 100 IG = 1,NG
      DO 29 I = 1,2
      J = 2
```

```
      IF (I.EQ.2) J = 1
      XQ(I) = 0.D0
      NSTAR(I) = 0.D0
      DO 30 JALPH = 1,3
      NOD = NODE(JALPH,K)
      XQ(I) = XQ(I) + SHAPF(JALPH,IG)*X(I,NOD)
      NSTAR(I) = NSTAR(I) + SHAPF1(JALPH,IG)*X(J,NOD)*(-1.D0)**J
   30 continue
   29 CONTINUE
      JAC = DSQRT(NSTAR(1)**2+NSTAR(2)**2)
C
C INCREMENT "P" NODE NUMBER
C
      DO 90 IR = 1,NTL
      IF(IR.GT.N) GO TO 90
      DO 40 JALPH = 1,3
      IF(NODE(JALPH,K).EQ.IR) GO TO 90
   40 CONTINUE
C
C P.NE.Q
C
C  MODIFIKASI UNTUK MASALAH HALF-SPACE : HEKSAPUTRA / 13388314
C
      DX(1) = XQ(1) - X(1,IR)
      DX(2) = XQ(2) - X(2,IR)
      R2 = DX(1)**2 + DX(2)**2
      R = DSQRT(R2)
C
C EVALUATE UKERN: U(LOGR)*JAC, TSTAR: T(DTHETA/DSI)
C
      DX(3)=XQ(1)+X(1,IR)
      RR2=DX(2)**2+DX(3)**2
      RR=DSQRT(RR2)
c
      X1 = X(1,IR)
      X2 = 2.D0*RB - X(2,IR)
      DDX(1)=DX(1)
      DDX(2)=XQ(2) - X2
      R2Z = DDX(1)**2 + DDX(2)**2
      RZ  = DSQRT(R2Z)
      DDX(3) = XQ(1)+X1
      RR2Z = DDX(2)**2 + DDX(3)**2
      RRZ = DSQRT(RR2Z)
c
      UKERN1 = (0.D0,0.D0)
      TSTAR1 = (0.D0,0.D0)
      PI = 3.141592653589793D0
C
C SC IS THE SCALE FACTOR OF THE INTEGRATION OVER THE ANGLE OF REVOLUTION
C TAKING INTO ACCOUNT THE SUBDIVISION OF THE ANGLE
C
      SC = PI/(2.D0*NS)
      CALL GAUSS(NGG,YY,WW)
      DO 515 IJ = 1,NS
      DO 51 IIG = 1,NGG
      SI = YY(IIG)
      WIG = WW(IIG)*2.D0
      AR = SC*SI+SC*(2.D0*IJ-1)
      RS2 = RR2-2.D0*XQ(1)*X(1,IR)*(1.D0+DCOS(AR))
      RS = DSQRT(RS2)
      ARG = -JC*WN*RS
c
      RS2Z = RR2Z-2.D0*XQ(1)*X1*(1.D0+DCOS(AR))
      RSZ  = DSQRT(RS2Z)
      ARGZ = -JC*WN*RSZ
c
C                          *****  DCOS THETA MODIFICATION *****
      U1 = ( CDEXP(ARG) - 1.D0) * SC/RS
     *    +RH*(CDEXP(ARGZ)-1.D0) * SC/RSZ
      UKERN1 = UKERN1 + 2.D0*WIG*U1
      ARGT = -JC*WN*RS + (-1.D0,0.D0)
```

```
      UK = CDEXP(ARG)
      DRDN = ((XQ(1)-X(1,IR)*DCOS(AR))/RS)*NSTAR(1)+((XQ(2)-X(2,IR))/RS)
     1      *NSTAR(2)
c
      ARGTZ = -JC*WN*RSZ + (-1.D0,0.D0)
      UKZ = CDEXP(ARGZ)
      DRDNZ = ((XQ(1)-X1*DCOS(AR))/RSZ)*NSTAR(1)+((XQ(2)-X2)
     1       /RSZ)*NSTAR(2)
c
C                          *****  DCOS THETA MODIFICATION *****
C
      U2 = ( ( ARGT*UK + 1.D0) / RS2 )*DRDN*SC
     1    +RH*((ARGTZ*UKZ + 1.D0)/RS2Z)*DRDNZ*SC

      IF (IIG.NE.1) GO TO 51
   51 TSTAR1 = TSTAR1 + 2*WIG*U2
  515 CONTINUE
      IF (K.NE.1)GO TO 555
 1000 FORMAT (2X,'ELEMENT #',3X,'NODE #',8X,'XQ(1)',12X,'XQ(2)',17X,
     *'UKERN1',29X,'TSTAR1/JAC'/)
 1001 FORMAT (4X,I2,8X,I2,7X,G15.5,2X,G15.5,2X,2G15.5,3X,2G15.5/)
  555 IF (X(1,IR).EQ.0.) GO TO 9
      RKP2=R2/(R2+4.D0*XQ(1)*X(1,IR))
      RK2=1.D0-RKP2
      RK=DSQRT(RK2)
c
      RKP2Z=R2Z/(R2Z+4.D0*XQ(1)*X1)
      RK2Z=1.D0-RKP2Z
      RKZ=DSQRT(RK2Z)
c
C
C EVALUATE KERNELS
C
   306 EE=MMDELE(RK2)
      EK=MMDELK(RK2)
      G=4.D0*EK/RR
      PRKR=(2.D0*X(1,IR)-RK2*DX(3))/(RK*RR2)
      PRKZ=-(RK*DX(2)/RR2)
      PRRR=DX(3)/RR
      PRRZ=DX(2)/RR
      PEKRK=EE/(RK*RKP2)-EK/RK
      PEERK = (EE - EK)/RK
      GNR=4.D0*(RR*PEKRK*PRKR-EK*PRRR)/RR2
      GNZ=4.D0*(RR*PEKRK*PRKZ-EK*PRRZ)/RR2
      GNRA = 4.D0*(RR*PEERK*PRKR + EE*PRRR)
      GNZA = 4.D0*(RR*PEERK*PRKZ + EE*PRRZ)
c
      EEZ=MMDELE(RK2Z)
      EKZ=MMDELK(RK2Z)
      GZ=4.D0*EKZ/RRZ
      PRKRZ=(2.D0*X1-RK2Z*DDX(3))/(RKZ*RR2Z)
      PRKZZ=-(RKZ*DDX(2)/RR2Z)
      PRRRZ=DDX(3)/RRZ
      PRRZZ=DDX(2)/RRZ
      PEKRKZ=EEZ/(RKZ*RKP2Z)-EKZ/RKZ
      PEERKZ=(EEZ - EKZ)/RKZ
      GNRZ=4.D0*(RRZ*PEKRKZ*PRKRZ-EKZ*PRRRZ)/RR2Z
      GNZZ=4.D0*(RRZ*PEKRKZ*PRKZZ-EKZ*PRRZZ)/RR2Z
      GNRAZ= 4.D0*(RRZ*PEERKZ*PRKRZ + EEZ*PRRRZ)
      GNZAZ= 4.D0*(RRZ*PEERKZ*PRKZZ + EEZ*PRRZZ)
c
      GO TO 21
    9 PI = 3.141592653589793D0
      PRRR = DX(3)/RR
      PRRZ = DX(2)/RR
      GN = PI/2.D0
      G = 4.D0*GN/RR
      GNZ = 4.D0*(-GN*PRRZ)/RR2
      GNR = 4.D0*(-GN*PRRR)/RR2
      EE = GN
      GNRA = 4.D0*EE*PRRR
```

```
      GNZA = 4.D0*EE*PRRZ
c
      PRRRZ = DDX(3)/RRZ
      PRRZZ = DDX(2)/RRZ
      GZ = 4.D0*GN/RRZ
      GNZZ = 4.D0*(-GN*PRRZZ)/RR2Z
      GNRZ = 4.D0*(-GN*PRRRZ)/RR2Z
      GNRAZ = 4.D0*EE*PRRRZ
      GNZAZ = 4.D0*EE*PRRZZ
c
   21 UKERN= (G + RH*GZ + UKERN1)*XQ(1)*JAC
      TTSTAR=GNR*XQ(1)*NSTAR(1)+GNZ*XQ(1)*NSTAR(2)
      TSTAR = (GNR*NSTAR(1) + GNZ*NSTAR(2) + TSTAR1
     1  +RH*(GNRZ*NSTAR(1) + GNZZ*NSTAR(2)) )*XQ(1)
C
C***** CONTACT PROBLEM *************
C
      if(x(2,ir).eq.rb) ttstar = (1+rh)*ttstar
      WIG = W(IG)*2.D0
      IF(K.GT.M) GO TO 81
C
C*************************************************
C
      DO 80 JA = 1,3
      SHALPH = SHAPF(JA,IG)
      IC = NODE(JA,K)
      A(IR,IC) = A(IR,IC) + WIG*SHALPH*TSTAR
      B(IR,IC) = B(IR,IC) + WIG*SHALPH*UKERN
   80 CONTINUE
C STORE CONTRIBUTION TO BTN VECTOR (INCLUDING CONTRIBUTION OF C)
C
   81 A(IR,IR) = A(IR,IR) - WIG*TTSTAR
C END OF IR LOOP
C
   90 CONTINUE
C
C END OF IG LOOP
C
  100 CONTINUE
C
C------
C P.EQ.Q
C------
C PLACE "P" AT EACH NODE OF THE SEGMENT
C
      IF(K.GE.M+1) GOTO 300
      DO 200 JA = 1,3
      IR = NODE(JA,K)
      DO 190 L = 1,2
      DO 180 IG = 1,NG
      IF (JA.EQ.1.AND.L.EQ.2.OR.JA.EQ.3.AND.L.EQ.1) GO TO 110
      IF (L.EQ.1) YI = -.5D0 + Y(IG)*.5D0
      IF (L.EQ.2) YI = .5D0 + Y(IG)*.5D0
      IF (JA.EQ.1) SI = YI**2 - 1.D0
      IF (JA.EQ.3) SI = 1.D0 - YI**2
      IF (L.EQ.1.AND.JA.EQ.2) SI = -YI**2
      IF (L.EQ.2.AND.JA.EQ.2) SI = YI**2
      GO TO 120
  110 SI = -.5D0 + Y(IG)*.5D0
      IF (L.EQ.2) SI = SI + 1.D0
  120 CALL SHFUN(SI,SHAP,SHAP1)
      DO 130 I = 1,2
      J = 2
      IF (I.EQ.2) J = 1
      XQ(I) = 0.D0
      NSTAR(I) = 0.D0
      DO 130 JALPH = 1,3
      NOD = NODE(JALPH,K)
      XQ(I) = XQ(I) + SHAP(JALPH)*X(I,NOD)
  130 NSTAR(I) = NSTAR(I) + SHAP1(JALPH)*X(J,NOD)*(-1.D0)**J
      JAC = DSQRT(NSTAR(1)**2+NSTAR(2)**2)
```

```
      DX(1) = XQ(1) - X(1,IR)
      DX(2) = XQ(2) - X(2,IR)

      R2 = DX(1)**2 + DX(2)**2
      R = DSQRT(R2)
      DX(3)=XQ(1)+X(1,IR)
      RR2=DX(2)**2+DX(3)**2
      RR=DSQRT(RR2)
c
      X1 =X(1,IR)
      X2 =2.D0*RB - X(2,IR)
      DDX(1) = DX(1)
      DDX(2) = XQ(2) - X2
      R2Z = DDX(1)**2 + DDX(2)**2
      RZ = DSQRT(R2Z)
      DDX(3) = XQ(1)+X1
      RR2Z = DDX(2)**2 + DDX(3)**2
      RRZ = DSQRT(RR2Z)
c
      UKERN1 = (0.D0,0.D0)
      TSTAR1 = (0.D0,0.D0)
      PI = 3.141592653589793D0
      SC = PI/(2.D0*NS)
      CALL GAUSS(NGG,YY,WW)
      DO 525 IJ = 1,NS
      DO 52 IIG = 1,NGG
      SI = YY(IIG)
      AR = SC*SI+SC*(2*IJ-1)
      WIG = WW(IIG)*2.D0
      RS2 = RR2-2.D0*XQ(1)*X(1,IR)*(1.D0+DCOS(AR))
      RS = DSQRT(RS2)
      ARG = -JC*WN*RS
c
      RS2Z = RR2Z-2.D0*XQ(1)*X1*(1.D0+DCOS(AR))
      RSZ = DSQRT(RS2Z)
      ARGZ = -JC*WN*RSZ
c
C                          *****  DCOS THETA MODIFICATION *****
      U1 = ( CDEXP(ARG) - 1.D0) * SC/RS
     1   +RH*( CDEXP(ARGZ) - 1.D0) * SC/RSZ
      UKERN1 = UKERN1 + 2*WIG*U1
      ARGT = -JC*WN*RS + (-1.D0,0.D0)
      UK = CDEXP(ARG)
      DRDN= ((XQ(1)-X(1,IR)*DCOS(AR))/RS)*NSTAR(1)+((XQ(2)-X(2,IR))/RS)*
     1      NSTAR(2)
c
      ARGTZ = -JC*WN*RSZ + (-1.D0,0.D0)
      UKZ = CDEXP(ARGZ)
      DRDNZ = ((XQ(1)-X1*DCOS(AR))/RSZ)*NSTAR(1)
     1     + ((XQ(2)-X2)/RSZ)*NSTAR(2)
c
C                          *****  DCOS THETA MODIFICATION *****
      U2 = ( ( ARGT*UK + 1.D0) / RS2 )*DRDN*SC
     1   +RH*(( ARGTZ*UKZ + 1.D0) / RS2Z )*DRDNZ*SC
      IF (IIG.NE.1) GO TO 52
   52 TSTAR1 = TSTAR1 + 2.D0*WIG*U2
  525 CONTINUE
      IF (K.NE.1) GO TO 556
  556 IF (X(1,IR).EQ.0.) GO TO 19
      RKP2=R2/(R2+4.D0*XQ(1)*X(1,IR))
      RK2=1.D0-RKP2
      RK=DSQRT(RK2)
c
      RKP2Z=R2Z/(R2Z+4.D0*XQ(1)*X1)
      RK2Z=1.D0-RKP2Z
      RKZ=DSQRT(RK2Z)
c
C
C EVALUATE KERNELS
C
  206 EE=MMDELE(RK2)
```

```
      EK=MMDELK(RK2)
      G=4.D0*EK/RR
      PRKR=(2.D0*X(1,IR)-RK2*DX(3))/(RK*RR2)
      PRKZ=-(RK*DX(2)/RR2)
      PRRR=DX(3)/RR
      PRRZ=DX(2)/RR
      PEKRK=EE/(RK*RKP2)-EK/RK
      PEERK = (EE - EK)/RK
      GNR=4.D0*(RR*PEKRK*PRKR-EK*PRRR)/RR2
      GNZ=4.D0*(RR*PEKRK*PRKZ-EK*PRRZ)/RR2
      GNRA = 4.D0*(RR*PEERK*PRKR + EE*PRRR)
      GNZA = 4.D0*(RR*PEERK*PRKZ + EE*PRRZ)
c
      EEZ=MMDELE(RK2Z)
      EKZ=MMDELK(RK2Z)
      GZ=4.D0*EKZ/RRZ
      PRKRZ=(2.D0*X1-RK2Z*DDX(3))/(RKZ*RR2Z)
      PRKZZ=-(RKZ*DDX(2)/RR2Z)
      PRRRZ=DDX(3)/RRZ
      PRRZZ=DDX(2)/RRZ
      PEKRKZ=EEZ/(RKZ*RKP2Z)-EKZ/RKZ
      PEERKZ = (EEZ - EKZ)/RKZ
      GNRZ=4.D0*(RRZ*PEKRKZ*PRKRZ-EKZ*PRRRZ)/RR2Z
      GNZZ=4.D0*(RRZ*PEKRKZ*PRKZZ-EKZ*PRRZZ)/RR2Z
      GNRAZ = 4.D0*(RRZ*PEERKZ*PRKRZ + EEZ*PRRRZ)
      GNZAZ = 4.D0*(RRZ*PEERKZ*PRKZZ + EEZ*PRRZZ)
c
      GO TO 31
   19 PI = 3.141592653589793D0
      PRRR = DX(3)/RR
      PRRZ = DX(2)/RR
      GN = PI/2.D0
      G = 4.D0*GN/RR
      GNZ = 4.D0*(-GN*PRRZ)/RR2
      GNR = 4.D0*(-GN*PRRR)/RR2
      EE = GN
      GNRA = 4.D0*EE*PRRR
      GNZA = 4.D0*EE*PRRZ
c
      PRRRZ = DDX(3)/RRZ
      PRRZZ = DDX(2)/RRZ
      GZ = 4.D0*GN/RRZ
      GNZZ = 4.D0*(-GN*PRRZZ)/RR2Z
      GNRZ = 4.D0*(-GN*PRRRZ)/RR2Z
      GNRAZ = 4.D0*EE*PRRRZ
      GNZAZ = 4.D0*EE*PRRZZ
c
   31 UKERN= (G + RH*GZ + UKERN1)*XQ(1)*JAC
      TTSTAR=GNR*XQ(1)*NSTAR(1)+GNZ*XQ(1)*NSTAR(2)
      TSTAR = (GNR*NSTAR(1) + GNZ*NSTAR(2) + TSTAR1
     1  +RH*(GNRZ*NSTAR(1) + GNZZ*NSTAR(2)))*XQ(1)
C
C***** CONTACT PROBLEM OLEH HEKSAPUTRA / 13388314
C
      if(x(2,ir).eq.rb) ttstar = (1+rh)*ttstar
C
C************************************************
C
      IF (L.EQ.1.AND.(JA.EQ.1.OR.JA.EQ.2)) GO TO 133
      IF (L.EQ.2.AND.(JA.EQ.2.OR.JA.EQ.3)) GO TO 135
      GO TO 137
  133 UKERN = -2.D0*YI*UKERN
      TSTAR = -2.D0*YI*TSTAR
      TTSTAR = -2.D0*YI*TTSTAR
      GO TO 137
  135 UKERN = 2.D0*YI*UKERN
      TSTAR = 2.D0*YI*TSTAR
      TTSTAR = 2.D0*YI*TTSTAR
  137 BT = 0.D0
      WIG = W(IG)
      DO 170 J = 1,3
```

```
      IC = NODE(J,K)
      SHALPH = SHAP(J)
      A(IR,IC) = A(IR,IC) + WIG*SHALPH*TSTAR
      B(IR,IC) = B(IR,IC) + WIG*SHALPH*UKERN
  170 CONTINUE

C STORE CONTRIBUTION TO BTN VECTOR (INCLUDING CONTRIBUTION OF C)
C
      A(IR,IR) = A(IR,IR) - WIG*(TTSTAR)
  180 CONTINUE
  190 CONTINUE
  200 CONTINUE
C
C*******************************************************************************************
C-----
C P CHIEF POINT
C-----
C
  300 if (NCP.eq.0) go to 333
      if(k.ge.m+1) go to 500
      DO 101 IG = 1,NG
      DO 291 I = 1,2
      JCH = 2
      IF (I.EQ.2) JCH = 1
      XQ(I) = 0.D0
      NSTAR(I) = 0.D0
      DO 301 JALPH = 1,3
      NOD = NODE(JALPH,K)
      XQ(I) = XQ(I) + SHAPF(JALPH,IG)*X(I,NOD)
  301 NSTAR(I) = NSTAR(I) + SHAPF1(JALPH,IG)*X(JCH,NOD)*(-1.D0)**JCH
  291 CONTINUE
      JAC = DSQRT(NSTAR(1)**2+NSTAR(2)**2)
C
C INCREMENT "P" NODE NUMBER
C
      DO 901 IR = 1,NCP
      IF(IR.GT.NCP) GO TO 901
C
C P.NE.Q
C
C  MODIFIKASI UNTUK MASALAH HALF-SPACE : HEKSAPUTRA / 13388314
C
      DX(1) = XQ(1) - XCP(1,IR)
      DX(2) = XQ(2) - XCP(2,IR)
      R2 = DX(1)**2 + DX(2)**2
      R = DSQRT(R2)
C
C EVALUATE UKERN: U(LOGR)*JAC, TSTAR: T(DTHETA/DSI)
C
      DX(3)=XQ(1)+XCP(1,IR)
      RR2=DX(2)**2+DX(3)**2
      RR=DSQRT(RR2)
c
      X1 = XCP(1,IR)
      X2 = 2.D0*RB - XCP(2,IR)
      DDX(1)=DX(1)
      DDX(2)=XQ(2) - X2
      R2Z = DDX(1)**2 + DDX(2)**2
      RZ  = DSQRT(R2Z)
      DDX(3) = XQ(1)+X1
      RR2Z = DDX(2)**2 + DDX(3)**2
      RRZ = DSQRT(RR2Z)
c
      UKERN1 = (0.D0,0.D0)
      TSTAR1 = (0.D0,0.D0)
      PI = 3.141592653589793D0
C
C SC IS THE SCALE FACTOR OF THE INTEGRATION OVER THE ANGLE OF REVOLUTION
C TAKING INTO ACCOUNT THE SUBDIVISION OF THE ANGLE
C
      SC = PI/(2.D0*NS)
```

```
      CALL GAUSS(NGG,YY,WW)
      DO 5151 IJ = 1,NS
      DO 511 IIG = 1,NGG
      SI = YY(IIG)
      WIG = WW(IIG)*2.D0
      AR = SC*SI+SC*(2.D0*IJ-1)
      RS2 = RR2-2.D0*XQ(1)*XCP(1,IR)*(1.D0+DCOS(AR))
      RS = DSQRT(RS2)
      ARG = -JC*WN*RS
c
      RS2Z = RR2Z-2.D0*XQ(1)*X1*(1.D0+DCOS(AR))
      RSZ  = DSQRT(RS2Z)
      ARGZ = -JC*WN*RSZ
C                          *****  DCOS THETA MODIFICATION *****
      U1 = ( CDEXP(ARG) - 1.D0) * SC/RS
     *    +RH*(CDEXP(ARGZ)-1.D0) * SC/RSZ
C
      UKERN1 = UKERN1 + 2.D0*WIG*U1
      ARGT = -JC*WN*RS + (-1.D0,0.D0)
      UK = CDEXP(ARG)
      DRDN = ((XQ(1)-XCP(1,IR)*DCOS(AR))/RS)*NSTAR(1)+
     1       ((XQ(2)-XCP(2,IR))/RS)*NSTAR(2)
c
      ARGTZ = -JC*WN*RSZ + (-1.D0,0.D0)
      UKZ = CDEXP(ARGZ)
      DRDNZ = ((XQ(1)-X1*DCOS(AR))/RSZ)*NSTAR(1)+((XQ(2)-X2)
     1       /RSZ)*NSTAR(2)
C                          *****  DCOS THETA MODIFICATION *****
      U2 = ( ( ARGT*UK + 1.D0) / RS2 )*DRDN*SC
     1    +RH*((ARGTZ*UKZ + 1.D0)/RS2Z)*DRDNZ*SC
      IF (IIG.NE.1) GO TO 511
  511 TSTAR1 = TSTAR1 + 2*WIG*U2
 5151 CONTINUE
      IF (K.NE.1)GO TO 5551
 1101 FORMAT (2X,'ELEMENT #',3X,'NODE #',8X,'XQ(1)',12X,'XQ(2)',17X,
     *'UKERN1',29X,'TSTAR1/JAC'/)
 1011 FORMAT (4X,I2,8X,I2,7X,G15.5,2X,G15.5,2X,2G15.5,3X,2G15.5/)
 5551 IF (XCP(1,IR).EQ.0.) GO TO 91
      RKP2=R2/(R2+4.D0*XQ(1)*XCP(1,IR))
      RK2=1.D0-RKP2
      RK=DSQRT(RK2)
c
      RKP2Z=R2Z/(R2Z+4.D0*XQ(1)*X1)
      RK2Z=1.D0-RKP2Z
      RKZ=DSQRT(RK2Z)
c
C
C EVALUATE KERNELS
C
 3061 EE=MMDELE(RK2)
      EK=MMDELK(RK2)
      G=4.D0*EK/RR
      PRKR=(2.D0*XCP(1,IR)-RK2*DX(3))/(RK*RR2)
      PRKZ=-(RK*DX(2)/RR2)
      PRRR=DX(3)/RR
      PRRZ=DX(2)/RR
      PEKRK=EE/(RK*RKP2)-EK/RK
      PEERK = (EE - EK)/RK
      GNR=4.D0*(RR*PEKRK*PRKR-EK*PRRR)/RR2
      GNZ=4.D0*(RR*PEKRK*PRKZ-EK*PRRZ)/RR2
      GNRA = 4.D0*(RR*PEERK*PRKR + EE*PRRR)
      GNZA = 4.D0*(RR*PEERK*PRKZ + EE*PRRZ)
c
      EEZ=MMDELE(RK2Z)
      EKZ=MMDELK(RK2Z)
      GZ=4.D0*EKZ/RRZ
      PRKRZ=(2.D0*X1-RK2Z*DDX(3))/(RKZ*RR2Z)
      PRKZZ=-(RKZ*DDX(2)/RR2Z)
      PRRRZ=DDX(3)/RRZ
      PRRZZ=DDX(2)/RRZ
      PEKRKZ=EEZ/(RKZ*RKP2Z)-EKZ/RKZ
```

```
      PEERKZ=(EEZ - EKZ)/RKZ
      GNRZ=4.D0*(RRZ*PEKRKZ*PRKRZ-EKZ*PRRRZ)/RR2Z
      GNZZ=4.D0*(RRZ*PEKRKZ*PRKZZ-EKZ*PRRZZ)/RR2Z
      GNRAZ= 4.D0*(RRZ*PEERKZ*PRKRZ + EEZ*PRRRZ)
      GNZAZ= 4.D0*(RRZ*PEERKZ*PRKZZ + EEZ*PRRZZ)
c
      GO TO 211
   91 PI = 3.141592653589793D0
      PRRR = DX(3)/RR
      PRRZ = DX(2)/RR
      GN = PI/2.D0
      G = 4.D0*GN/RR
      GNZ = 4.D0*(-GN*PRRZ)/RR2
      GNR = 4.D0*(-GN*PRRR)/RR2
      EE = GN
      GNRA = 4.D0*EE*PRRR
      GNZA = 4.D0*EE*PRRZ
c
      PRRRZ = DDX(3)/RRZ
      PRRZZ = DDX(2)/RRZ
      GZ = 4.D0*GN/RRZ
      GNZZ = 4.D0*(-GN*PRRZZ)/RR2Z
      GNRZ = 4.D0*(-GN*PRRRZ)/RR2Z
      GNRAZ = 4.D0*EE*PRRRZ
      GNZAZ = 4.D0*EE*PRRZZ
c
   211 UKERN= (G + RH*GZ + UKERN1)*XQ(1)*JAC
      TTSTAR=GNR*XQ(1)*NSTAR(1)+GNZ*XQ(1)*NSTAR(2)
      TSTAR = (GNR*NSTAR(1) + GNZ*NSTAR(2) + TSTAR1
     1  +RH*(GNRZ*NSTAR(1) + GNZZ*NSTAR(2)) )*XQ(1)
C
C***** CONTACT PROBLEM *************
C
      if(x(2,ir).eq.rb) ttstar = (1+rh)*ttstar
      WIG = W(IG)*2.D0
C
C************************************************
C
      DO 801 JA = 1,3
      SHALPH = SHAPF(JA,IG)
      IC = NODE(JA,K)
      ICH = N + IR
      A(ICH,IC) = A(ICH,IC) + WIG*SHALPH*TSTAR
      B(ICH,IC) = B(ICH,IC) + WIG*SHALPH*UKERN
   801 CONTINUE
C STORE CONTRIBUTION TO BTN VECTOR (INCLUDING CONTRIBUTION OF C)
C
C END OF IR LOOP
C
   901 CONTINUE
C
C END OF IG LOOP
C
   101 CONTINUE
C
C-------
C P FIELD POINT
C MODIFIKASI OLEH DWI URIKA (10295077)
C-------
C
   333 if(k.ge.m+1) go to 500
      DO 102 IG = 1,NG
      DO 292 I = 1,2
      JF = 2
      IF (I.EQ.2) JF = 1
      XQ(I) = 0.D0
      NSTAR(I) = 0.D0
      DO 302 JALPH = 1,3
      NOD = NODE(JALPH,K)
      XQ(I) = XQ(I) + SHAPF(JALPH,IG)*X(I,NOD)
      NSTAR(I) = NSTAR(I) + SHAPF1(JALPH,IG)*X(JF,NOD)*(-1.D0)**JF
```

```
  302 CONTINUE
  292 CONTINUE
      JAC = DSQRT(NSTAR(1)**2+NSTAR(2)**2)
C
C INCREMENT "P" NODE NUMBER
C
      DO 902 IR = 1,NFP
C
C P.NE.Q
C
C  MODIFIKASI UNTUK MASALAH HALF-SPACE : HEKSAPUTRA / 13388314
C
      DX(1) = XQ(1) - XFP(1,IR)
      DX(2) = XQ(2) - XFP(2,IR)
      R2 = DX(1)**2 + DX(2)**2
      R = DSQRT(R2)
C
C EVALUATE UKERN: U(LOGR)*JAC, TSTAR: T(DTHETA/DSI)
C
      DX(3)=XQ(1)+XFP(1,IR)
      RR2=DX(2)**2+DX(3)**2
      RR=DSQRT(RR2)
c
      X1 = XFP(1,IR)
      X2 = 2.D0*RB - XFP(2,IR)
      DDX(1) = DX(1)
      DDX(2) = XQ(2) - X2
      R2Z = DDX(1)**2 + DDX(2)**2
      RZ  = DSQRT(R2Z)
      DDX(3) = XQ(1)+X1
      RR2Z = DDX(2)**2 + DDX(3)**2
      RRZ = DSQRT(RR2Z)
c
      UKERN1 = (0.D0,0.D0)
      TSTAR1 = (0.D0,0.D0)
      PI = 3.141592653589793D0
C
C SC IS THE SCALE FACTOR OF THE INTEGRATION OVER THE ANGLE OF REVOLUTION
C TAKING INTO ACCOUNT THE SUBDIVISION OF THE ANGLE
C
C=== NS = 32
C
      SC = PI/(2.D0*NS)
      CALL GAUSS(NGG,YY,WW)
      DO 5152 IJ = 1,NS
      DO 512 IIG = 1,NGG
      SI = YY(IIG)
      WIG = WW(IIG)*2.D0
      AR = SC*SI+SC*(2.D0*IJ-1)
      RS2 = RR2-2.D0*XQ(1)*XFP(1,IR)*(1.D0+DCOS(AR))
      RS = DSQRT(RS2)
      ARG = -JC*WN*RS
c
      RS2Z = RR2Z-2.D0*XQ(1)*X1*(1.D0+DCOS(AR))
      RSZ  = DSQRT(RS2Z)
      ARGZ = -JC*WN*RSZ
c
C                          *****  DCOS THETA MODIFICATION *****
C
      U1 = ( CDEXP(ARG)) * SC/RS
     *    +RH*(CDEXP(ARGZ)) * SC/RSZ
      UKERN1 = UKERN1 + 2.D0*WIG*U1
      ARGT = -JC*WN*RS + (-1.D0,0.D0)
      UK = CDEXP(ARG)
      DRDN = ((XQ(1)-XFP(1,IR)*DCOS(AR))/RS)*NSTAR(1)+
     1       ((XQ(2)-XFP(2,IR))/RS)*NSTAR(2)
c
      ARGTZ = -JC*WN*RSZ + (-1.D0,0.D0)
      UKZ = CDEXP(ARGZ)
      DRDNZ = ((XQ(1)-X1*DCOS(AR))/RSZ)*NSTAR(1)+((XQ(2)-X2)
     1       /RSZ)*NSTAR(2)
```

```
c
C                          *****  DCOS THETA MODIFICATION *****
C
      U2 = ( ( ARGT*UK ) / RS2 )*DRDN*SC
     1    +RH*((ARGTZ*UKZ )/RS2Z)*DRDNZ*SC
      TSTAR1 = TSTAR1 + 2*WIG*U2
  512 CONTINUE
 5152 CONTINUE
c
      G = 0.D0
      GNR = 0.D0
      GNZ = 0.D0
  212 UKERN= (UKERN1 + G)*XQ(1)*JAC
      TSTAR = (TSTAR1 + GNR*NSTAR(1) + GNZ*NSTAR(2)) *XQ(1)
C
C***** CONTACT PROBLEM *************
C
c     if(x(2,ir).eq.rb) ttstar = (1+rh)*ttstar
      WIG = W(IG)*2.D0
c     IF(K.GT.M) GO TO 902
C***********************************************
C
      DO 802 JA = 1,3
      SHALPH = SHAPF(JA,IG)
      IC = NODE(JA,K)
      E(IR,IC) = E(IR,IC) + WIG*SHALPH*TSTAR
      D(IR,IC) = D(IR,IC) - WIG*SHALPH*UKERN
  802 CONTINUE
C STORE CONTRIBUTION TO BTN VECTOR (INCLUDING CONTRIBUTION OF C)
C
C END OF IR LOOP
C
  902 CONTINUE
C
C END OF IG LOOP
C
  102 CONTINUE

C*****************************************************************************************
C
C END OF K LOOP
C
  500 CONTINUE
C
  501 IF (ISC.EQ.0) GO TO 656
      PI4 = 4.D0*3.141592653589793D0
      CALL SCATRX(X,XCP,XFP,EPHI,EPHIF,N,NCP,NFP,NPCH)
      DO 652 I = 1,NPCH
  652 BTN(I) = BTN(I) + EPHI(I)*PI4
      DO 653 J = 1,NFP
  653 PFP(J) = PFP(J) + EPHIF(J)*PI4
  656 IF (IDD.EQ.1) GO TO 551
      DO 655 L = 1,N
      A(L,L) = A(L,L) + PIT4
  655 CONTINUE
  551 CONTINUE
      RETURN
      END

C=======================================================================
      REAL*8 FUNCTION MMDELK(XK)
C=======================================================================
C
C INTEGRAL ELIPTIK OLEH HEKSAPUTRA / 13388314
C
      REAL*8 A(16),U(16)
      REAL*8 XPOS,XNEG
      REAL*8 PHI,F1,PHIK,XK
      DATA U/
     +0.9972638618494815635449811286650407271385 4D0,
     +0.9856115115452683354001750446309019786324D0,
```

```
     +0.96476225558750643077381192811827496038890D0,
     +0.93490607593773968917091913483540932552870D0,
     +0.89632115576605212396530724371921226847900D0,
     +0.84936761373256997013369300496774253895489D0,
     +0.79448379596794240696309729897042890209547 9D0,
     +0.73218211874028968038742666509126714663027 0D0,
     +0.66304426693021520097511516866323836897702 23D0,
     +0.58771575724076232904074547640182685845094012D0,
     +0.50689990893222939002374747437782123018028 3700D0,
     +0.42135127613063534536411943617242647833587 72886D0,
     +0.33186860228212764977991680573018799619577 513681D0,
     +0.23928736225213707454460320916550152060885 5421960D0,
     +0.14447196158279649348518637359881065220384 59913156D0,
     +0.04830766568773831623481257044050216369084 7251730849D0 /
      DATA A/
     +0.00701861000947009660040706373 89D0,
     +0.01627439473090567060517056220 6D0,
     +0.02539206530926205945575258978 9D0,
     +0.03427386291302143310268773225 2D0,
     +0.04283589802222668065687864660 6D0,
     +0.05099805926237617619616324468 95D0,
     +0.05868409347853554714528363730 017D0,
     +0.06582222277636184683765006370 6939D0,
     +0.07234579410884850622539935647 84878D0,
     +0.07819389578707030647174091882 8306671D0,
     +0.08331192422694675522219907460 434861154D0,
     +0.08765209300440381114277146275 18022875484D0,
     +0.09117387869576388471286857711163706254 4861D0,
     +0.09384439908080456563918023766811726003 610008D0,
     +0.09563872007927485941908200220413110059 489050816D0,
     +0.09654008851472780056676483006357579473 6860631235 57D0 /
C
      PHIK=0.D0
      B = 3.141592653589793D0/2.D0
      A0=0.D0
      DO 100 K=1,16
      XPOS=(B-A0)/2.D0*U(K)+(B+A0)/2.D0
      XNEG= -U(K)*(B-A0)/2.D0+(B+A0)/2.D0
      PHI=.5D0*(B-A0)*( F1(XPOS,XK)*A(K)+F1(XNEG,XK)*A(K) )
      PHIK= PHIK+PHI
  100 CONTINUE
      MMDELK=PHIK
      END

      REAL*8 FUNCTION F1(X,XK)
      REAL*8 X,XK
      F1=1.D0/(DSQRT(1.D0-XK*(DSIN(X))**2))
      END
C=====================================================================
      REAL*8 FUNCTION MMDELE(XK)
C=====================================================================
C
C INTEGRAL ELIPTIK OLEH HEKSAPUTRA / 13388314
C
      REAL*8 A(16),U(16)
      REAL*8 XPOS,XNEG
      REAL*8 PHI,F2,PHIK,XK
      DATA U/
     +0.99726386184948156354498112866504072713854D0,
     +0.98561151154526833540017504463090197863 24D0,
     +0.96476225558750643077381192811827496038 89D0,
     +0.93490607593773968917091913483540932552 87D0,
     +0.89632115576605212396530724371921226847 900D0,
     +0.84936761373256997013369300496774253895 489D0,
     +0.79448379596794240696309729897042890209 5479D0,
     +0.73218211874028968038742666509126714663 0270D0,
     +0.66304426693021520097511516866323836897 7022 3D0,
     +0.58771575724076232904074547640182685845 094012D0,
     +0.50689990893222939002374747437782123018 0283700D0,
     +0.42135127613063534536411943617242647833 58772886D0,
     +0.33186860228212764977991680573018799619 577513681D0,
```

```
     +0.23928736225213707454460320916550152060885542196OD0,
     +0.14447196158279649348518637359881065220384599131560D0,
     +0.04830766568773831623481257044050216369084725173084 9D0 /
      DATA A/
     +0.00701861000947009660040706373 89D0,
     +0.01627439473090567060517056220 6D0,
     +0.02539206530926205945575258978 9D0,
     +0.03427386291302143310268773225 2D0,
     +0.04283589802222668065687864660 6D0,
     +0.05099805926237617619616324468 95D0,
     +0.05868409347853554714528363730017D0,
     +0.06582222277636184683765006370 6939D0,
     +0.07234579410884850622539935647 84878D0,
     +0.07819389578707030647174091882 8306671D0,
     +0.08331192422694675522219907460 434861154D0,
     +0.08765209300440381114277146275 18022875484D0,
     +0.09117387869576388471286857711 1637062544861D0,
     +0.09384439908080456563918023766 81172600361000 8D0,
     +0.09563872007927485941908200220 413110059489050816D0,
     +0.09654008851472780056676483006 3575794736860631235 57D0 /
C
      PHIK=0.D0
       B = 3.141592653589793D0/2.D0
       A0=0.D0
      DO 100 K=1,16
      XPOS=(B-A0)/2.D0*U(K)+(B+A0)/2.D0
      XNEG= -U(K)*(B-A0)/2.D0+(B+A0)/2.D0
      PHI=.5D0*(B-A0)*( F2(XPOS,XK)*A(K)+F2(XNEG,XK)*A(K) )
      PHIK= PHIK+PHI
  100 CONTINUE
      MMDELE=PHIK
      END

C=======================================================================
      REAL*8 FUNCTION F2(X,XK)
C=======================================================================
      REAL*8 X,XK
      F2=DSQRT(1.D0-XK*(DSIN(X))**2)
      END


C=======================================================================
      SUBROUTINE SCATRX(X,XCP,XFP,EPHI,EPHIF,N,NCP,NFP,NPCH)
C=======================================================================
      REAL*8 X(2,N),XCP(2,NCP),XFP(2,NFP)
      REAL*8 DX(2),R,Z1,PI,PI2,DIRP,DIRM,X2,DIRR,DIRZ,AMP,RB,ZO
      REAL*8 RF,DIRPF,DIRMF
      COMPLEX*16 PHI,PHIF,PHIM,PHIMF,EPHI(NPCH),EPHIF(NFP),RH
      COMPLEX*16 WN,JC
      COMMON/SCAT/ISC,DIRR,DIRZ,AMP
      COMMON/OMEGA/WN
      COMMON/DOMA/IHF
      COMMON/IMAGE/RB
      COMMON/IMP/ZO
      COMMON/HALF/RH
C
C MODIFIKASI OLEH HEKSAPUTRA / 13388314
C
      PI = 3.14159265358979300
      PI2  = PI * 2
      JC = (0.0,1.0)
      IF (ISC.EQ.2) GO TO 500
      DO 100 I = 1,N
      DIRP = DIRR*X(1,I) + DIRZ*(X(2,I) - RB)
      PHI = -JC*WN*DIRP
      EPHI(I) = AMP*CDEXP(PHI)/(JC*WN*ZO)
  100 CONTINUE
      IF(IHF.EQ.0 .OR. DIRZ.EQ.0.) GO TO 1009
      DO 200 I = 1,N
      X2 = - X(2,I) + RB
      DIRM = DIRR*X(1,I) + DIRZ*X2
```

```
      PHIM = -JC*WN*DIRM
      EPHI(I) = EPHI(I) + RH*AMP*CDEXP(PHIM)/(JC*WN*ZO)
  200 CONTINUE
C
  500 DO 600 I=1,N
      DX(1)=X(1,I)-DIRR
      DX(2)=X(2,I)-DIRZ
      R=DSQRT(DX(1)**2+DX(2)**2)
      APT = DSQRT(ZO*AMP/PI2)
  600 EPHI(I) = APT*CDEXP(-JC*WN*R)/(WN*ZO*R)
      IF(IHF.EQ.0) GO TO 1009
      Z1=2.*RB-DIRZ
      DO 700 I=1,N
      DX(1)=X(1,I)-DIRR
      DX(2)=X(2,I)-Z1
      R=DSQRT(DX(1)**2+DX(2)**2)
      APT = DSQRT(ZO*AMP/PI2)
  700 EPHI(I) = EPHI(I) + RH*APT*CDEXP(-JC*WN*R)/(WN*ZO*R)
  800 CONTINUE
C
 1009 IF (NCP.EQ.0) GO TO 2009
      IF (ISC.EQ.2) GO TO 5001
      DO 1001 I = 1,NCP
      J = N + I
      DIRP = DIRR*XCP(1,I) + DIRZ*(XCP(2,I) - RB)
      PHI = -JC*WN*DIRP
      EPHI(J) = AMP*CDEXP(PHI)/(JC*WN*ZO)
 1001 CONTINUE
      IF(IHF.EQ.0 .OR. DIRZ.EQ.0.) GO TO 2009
      DO 2001 I = 1,NCP
      X2 = - XCP(2,I) + RB
      DIRM = DIRR*XCP(1,I) + DIRZ*X2
      PHIM = -JC*WN*DIRM
      EPHI(J) = EPHI(J) + RH*AMP*CDEXP(PHIM)/(JC*WN*ZO)
 2001 CONTINUE
C
 5001 DO 6001 I=1,NCP
      J = N + I
      DX(1)=XCP(1,I)-DIRR
      DX(2)=XCP(2,I)-DIRZ
      R=DSQRT(DX(1)**2+DX(2)**2)
      APT = DSQRT(ZO*AMP/PI2)
 6001 EPHI(J) = APT*CDEXP(-JC*WN*R)/(WN*ZO*R)
      IF(IHF.EQ.0) GO TO 2009
      Z1=2.*RB-DIRZ
      DO 7001 I=1,N
      DX(1)=XCP(1,I)-DIRR
      DX(2)=XCP(2,I)-Z1
      R=DSQRT(DX(1)**2+DX(2)**2)
      APT = DSQRT(ZO*AMP/PI2)
      EPHI(J) = EPHI(J) + RH*APT*CDEXP(-JC*WN*R)/(WN*ZO*R)
 7001 CONTINUE
C
 2009 IF (ISC.EQ.2) GO TO 5002
      DO 1002 I = 1,NFP
      DIRPF = DIRR*XFP(1,I) + DIRZ*(XFP(2,I) - RB)
      PHIF = -JC*WN*DIRPF
      EPHIF(I) = AMP*CDEXP(PHIF)/(JC*WN*ZO)
 1002 CONTINUE
      IF(IHF.EQ.0 .OR. DIRZ.EQ.0.) RETURN
      DO 2002 I = 1,NFP
      X2 = - XFP(2,I) + RB
      DIRMF = DIRR*XFP(1,I) + DIRZ*X2
      PHIMF = -JC*WN*DIRMF
      EPHIF(I) = EPHIF(I) + RH*AMP*CDEXP(PHIMF)/(JC*WN*ZO)
 2002 CONTINUE
      RETURN
C
 5002 DO 6002 I=1,NFP
      DX(1)=XFP(1,I)-DIRR
      DX(2)=XFP(2,I)-DIRZ
```

```
      RF=DSQRT(DX(1)**2+DX(2)**2)
      APTF = DSQRT(ZO*AMP/PI2)
 6002 EPHIF(I) = APTF*CDEXP(-JC*WN*RF)/(WN*ZO*RF)
      IF(IHF.EQ.0) RETURN
      Z1=2.*RB-DIRZ
      DO 7002 I=1,NFP
      DX(1)=XFP(1,I)-DIRR
      DX(2)=XFP(2,I)-Z1
      RF=DSQRT(DX(1)**2+DX(2)**2)
      APTF = DSQRT(ZO*AMP/PI2)
 7002 EPHIF(I) = EPHIF(I) + RH*APTF*CDEXP(-JC*WN*RF)/(WN*ZO*RF)
C
      WRITE(6,1201)
      DO 44 I = 1,N
      WRITE (6,1202) I,ZO,EPHI(I),EPHIF(I)
   44 CONTINUE
 1201 FORMAT (5X,'NODE',2X,'ZO',12X,'INCOMING WAVE EPHI',20X,'INCOMING
     1       WAVE EPHIF',/)
 1202 FORMAT (5X,I3,2X,F10.5,2G20.5,2G20.5)
C
      RETURN
      END
C

C=2====================================================================
      SUBROUTINE SOLVE (A,B,BTN,E,D,PFP,NFP,NPCH,N,NPCT2,NFPT2,NT2)
C=======================================================================
C-----
C-- TO FIND BOUNDARY CONDITION
C-----
C
      DIMENSION PFP(NFP),YD(400),BTN(NPCH)
      DIMENSION A(NPCH,N),B(NPCH,N),E(NFP,N),D(NFP,N),G(400,300)
              DIMENSION AR(npct2,nt2),BR(npct2,nt2),CR(nfpt2,nt2),DR(npct2,nt2)
              DIMENSION YD1(300,1),YD2(300,1),tr(npct2,nt2),AI(NPCT2,NT2)
              DIMENSION CAI(NPCT2,NT2),CAIB(NPCT2,NT2),GR(NPCT2,NT2)
              DIMENSION GI(NPCT2,NT2),FR(NPCT2,NT2),DPH(NPCT2,1),BD(NPCT2,1)
              DIMENSION SE(NPCT2,1),PH(NPCT2,1),SR(NPCT2,1),tek(npct2,1)
      COMPLEX*16 A,B,BTN,E,D,PFP
      INTEGER NPCH,NFP,N,NPCT2,NFPT2,NT2,C
      REAL*8 g,yd,AR,BR,CR,DR,YD1,YD2,AI,CAI,CAIB,GR,GI,FR,DPH,SE
              REAL*8 BD,SR,PH,tek
C-----
C-- USING EQUATION  [A] {PHI} = {BTN} + [B] {DPHI/DN}
C--             OR  {PHI} = [A+] {BTN} + [A+] [B] {DPHI/DN}
C-- AND   EQUATION  [C] {PHI} + [D] {DPHI/DN} = {PFP}
C--             OR  ([C][A+][B]+[D]) {DPHI/DN} = {PFP} - [C] [A+] {BTN}
C-----
C--
C-----
C-- SVD ON [A]     AD(NR,NC),CD(NR),G(NR2,NC2),YD(NR2)
C-----
      WRITE (6,5)
    5 FORMAT(/2X,'ELEMEN MATRIKS [A] DAN {BTN}',/2X)
      CALL MATRIX (A,BTN,AR,NPCH,N,YD1,NPCT2,NT2)


C
      WRITE (6,6)
    6 FORMAT(/2X,'ELEMEN MATRIKS [B] DAN {BTN}',/2X)
      CALL MATRIX (B,BTN,BR,NPCH,N,YD1,NPCT2,NT2)

C
      WRITE (6,7)
    7 FORMAT(/2X,'ELEMEN MATRIKS [C] DAN {PFP}',/2X)
      CALL MATRIX (E,PFP,CR,NFP,N,YD2,NFPT2,NT2)

C
      WRITE (6,8)
    8 FORMAT(/2X,'ELEMEN MATRIKS [D] DAN {PFP}',/2X)
      CALL MATRIX (D,PFP,DR,NFP,N,YD2,NFPT2,NT2)
```

```
C       SEKARANG DICARI NILAI MATRIX [G]=[C][A+][B]+[D]

        CALL INVERS (AR,AI,npct2,nt2)


        CALL MULT (CAI,CR,AI,NPCT2,NT2,NT2)
        CALL MULT (CAIB,CAI,BR,NPCT2,NT2,NT2)

        CALL COEMR (CAIB,NCPT2,NT2)

        CALL ADD (GR,CAIB,DR,NPCT2,NT2)
        CALL INVERS (GR,GI,npct2,nt2)



C       DICARI NILAI MATRIX F

        CALL MULT (SR,CAI,YD1,NPCT2,NT2,1)
        CALL SUBSTRACT (FR,YD2,SR,NPCT2,1)

c       call coemr(fr,npct2,1)

        CALL MULT (DPH,GI,FR,NPCT2,NT2,1)

        WRITE(6,8933)
 8933 FORMAT(/,5X,'MATRIKS DPHI/DN',/)

    DO 9815 C=1,NPCT2
        WRITE(6,91122) DPH(C,1)
91122 FORMAT(42(G15.7),';')
9815    CONTINUE


C       DIDAPAT NILAI DPHI/DN=DPH

        CALL MULT (BD,BR,DPH,NPCT2,NT2,1)
        CALL ADD (SE,YD1,BD,NPCT2,1)

C       NILAI [PH]=[A+][SE]

        CALL MULT (PH,AI,SE,NPCT2,NT2,1)

        WRITE(6,8923)
 8923 FORMAT(/,5X,'MATRIKS PHI',/)

        DO 9895 C=1,NPCT2
        WRITE(6,9122) PH(C,1)
 9122 FORMAT(42(G15.7),';')
 9895   CONTINUE

        do 452 c=1,npct2
        tek(c,1)=-10*415*ph(c,1)
  452   continue

        WRITE(6,89)
   89 FORMAT(/,5X,'MATRIKS P',/)

        DO 983 C=1,NPCT2
        WRITE(6,912) tek(C,1)
  912 FORMAT(42(G15.7),';')
  983   CONTINUE
C
    RETURN
    END
C
C=======================================================================
    SUBROUTINE MATRIX (AD,CD,G,NR,NC,YD,NR2,NC2)
C=======================================================================
C-----
```

```
      DIMENSION AD(NR,NC),CD(NR),G(NR2,NC2),YD(NR2,1)
      COMPLEX*16 AD,CD
      real*8 g,yd
      integer nr,nc
C-----
C-- CHANGE TO REAL
C-----
      DO 5 I1 = 1,NR
      I2 = NR + I1
      YD(I1,1) = REAL(CD(I1))
      YD(I2,1) = AIMAG(CD(I1))
      DO 5 J1 = 1,NC
      J2 = NC + J1
      G(I1,J1) = REAL(AD(I1,J1))
      G(I1,J2) = -AIMAG(AD(I1,J1))
      G(I2,J1) = AIMAG(AD(I1,J1))
      G(I2,J2) = REAL(AD(I1,J1))
    5 CONTINUE

            call coemr(g,nr2,nc2)

      WRITE (6,94)
   94 FORMAT(/,5X,'MATRIKS KOLOM',/)

            DO 95 C3=1,NR2
            WRITE(6,961) YD(C3,1)
   95       CONTINUE
  961 FORMAT((G15.7),';')

      RETURN
      END
C

C=======================================================================
C
C=======================================================================


C=======================================================================
      SUBROUTINE ADD (TR,M1,M2,I,J)
C=======================================================================
            DIMENSION M1(I,J),M2(I,J),TR(I,J)
            INTEGER I,J,C1,C2
            REAL*8 TR,M1,M2

            C1=0
            C2=0

            DO 450 C1=1,I
            DO 450 C2=1,J
            TR(C1,C2)=M1(C1,C2)+M2(C1,C2)
  450 CONTINUE

            WRITE(6,678)
  678       FORMAT(/,2X,'HASIL JUMLAH')

            RETURN
            END

C=======================================================================
      SUBROUTINE MULT (EQ,M1,M2,I,J,K)
C=======================================================================

            DIMENSION M1(I,J),M2(J,K),EQ(I,K)
            INTEGER I,J,C1,C2,C3
            REAL*8 EQ,M1,M2,ELM

      C1=0
            C2=0
```

```
            DO 560 C3=1,K
            DO 562 C1=1,I
            ELM=0
            DO 561 C2=1,J
            ELM=ELM+M1(C1,C2)*M2(C2,C3)
            EQ(C1,C3)=ELM

 561 CONTINUE
 562        CONTINUE
 560 CONTINUE

    WRITE(6,679)
 679        FORMAT(/,2X,'HASIL KALI')

            RETURN
            END

C=====================================================================
    SUBROUTINE SUBSTRACT (SB,M1,M2,I,J)
C=====================================================================
            DIMENSION M1(I,J),M2(I,J),SB(I,J)
            INTEGER I,J,C1,C2
            REAL*8 SB,M1,M2

            C1=0
            C2=0


            DO 451 C2=1,J
            DO 451 C1=1,I
            SB(C1,C2)=M1(C1,C2)-M2(C1,C2)
 451 CONTINUE

            RETURN
            END


C=====================================================================
    SUBROUTINE INVERS (R,F,npct2,nt2)
C=====================================================================
C           PROGRAM INVERS MATRIX
C           R = MATRIX YANG AKAN DIINVERS
C           F = MATRIX IDENTITAS, DIMENSI SAMA DENGAN A
C           B = BARIS, K = KOLOM

            DIMENSION R(npct2,nt2)
            DIMENSION F(npct2,nt2)
            REAL*8 R,F,XTZZ,XT

            DO 2123 I=1,npct2
            DO 2124 J=1,nt2
            F(I,J) = 0
2124        CONTINUE
2123        CONTINUE

C           BIKIN MATRIX IDENTITY N*N
            DO 9970 S=1,nt2
            F(S,S) = 1
9970        CONTINUE


            WRITE(6,7278)
7278        FORMAT(/,2X,'MATRIX F AWAL :')


C           PROSES MATRIX MENGGUNAKAN  O.B.E.
            DO 9950 M=1,npct2
            XTZZ= R(M,M)

C           ELEMEN N,N (DIAGONAL UTAMA) DIBUAT = 1
            DO 9949 Z=1,nt2
```

```
            R(M,Z) = R(M,Z)/XTZZ
            F(M,Z) = F(M,Z)/XTZZ
9949        CONTINUE

C           ELEMEN SELAIN N,N DIBUAT = 0
            DO 9948 D=1,npct2
            IF (D.EQ.M) GOTO 9948
            XT=R(D,M)
            DO 9947 L=1,nt2
            R(D,L) = R(D,L)-(XT*R(M,L))
            F(D,L) = F(D,L)-(XT*F(M,L))
9947        CONTINUE
9948        CONTINUE
9950  CONTINUE

            RETURN
            END

C=1====================================================================
      SUBROUTINE COEMR (DM,NROW1,NCOL1)
C=====================================================================
C-----
C-- TO PRINT MATRIX COEFICIENT
C-----
c       DIMENSION DM(144,144)
      DIMENSION DM(NROW1,NCOL1)
            real*8 dm
            integer nrow1,ncol1

      IF (NCOL1.EQ.1) GO TO 50
      JB = 1
      JE = 10
   10 IF (JE.GT.NCOL1) JE = NCOL1
      WRITE (6,20) (J,J = JB,JE)
   20 FORMAT(/3X,10(6X,I3,4X))
      DO 30 I = 1,NROW1
   30 WRITE(6,40) I,(DM(I,L),L = JB,JE)
   40 FORMAT(I3,10(2X,F11.6))
      IF (JE.EQ.NCOL1) GO TO 80
      JB = JE + 1
      JE = JB + 9
      GO TO 10
   50 JB = 1
      JE = 7
   60 IF (JE.GT.NROW1) JE = NROW1
      WRITE (6,70) (I,DM(I,1),I = JB,JE)
   70 FORMAT (8(2X,I3,1X,G12.6,'|'))
      IF (JE.EQ.NROW1) GO TO 80
      JB = JE + 1
      JE = JB + 6
      GO TO 60
   80 RETURN
      END
```