```
      PROGRAM HEXB
C
C     Find the body axes and coordinates of the vertices
C     of the truncated pyramid hexagon B as the result of
C     the geodesic triangulation of an icosahedron. The
C     process described in chapter 6.1 is implemented here.
C     This progam finds the positions of the 50 hexagon B
C     in the calorimeter. Similar programs were written for
C     the other modules. The output of this routine and of the
C     others like it are the data files pent.dat, hea.dat,
C     heb.dat, hec.dat, hed.dat, hdh.dat and vet.dat which
C     contain the necessary information for producing the rotation
C     matrices: CALL GSROTM(...), and the translation vectors for
C     each of the 240 modules in Geant: CALL GSPOS(...).
C
C        Author: KETEVI ADIKLE ASSAMAGAN
C        Date: February 1993
C
C     Parts of fuller.f, written Prof. Dinko Pocanic, were
C     incorporated in this routine.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*6 name(2)
      INTEGER*2 type(2),count
      REAL*8 n(3)
      COMMON/points/xb(3),yb(3),zb(3)
      COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
      COMMON/triangle/na,n0,nthet,count
      COMMON/xaxis/theta1,phi1
      COMMON/yaxis/theta2,phi2
      REAL*8 AZ(6), CAZ(6),SAZ(6),NX(3)
     1,NY(3),NZ
      DATA count/0/
      DATA name/'HEXB1','HEXB2'/
      DATA theta1,phi1,theta2,phi2,theta3,phi3/6*0.0/
C
C     Assign the three digits nx, ny and nz to the geodesic vertices
C     following the conventions described in chapter 6.
C
      DATA n/3*0.0/
      DATA NX/3.0,2.0,4.0/
      DATA NY/0.0,0.0,0.0/
      DATA AZ /0.,36.,108.,180.,252.,324./
C
      OPEN(UNIT=20,NAME='hexb.dat',CARRIAGECONTROL='LIST',
     1    TYPE='UNKNOWN')
      OPEN(UNIT=21,NAME='heb.dat',CARRIAGECONTROL='LIST',
     1    TYPE='UNKNOWN')
      DO i=1,2
```

```
          type(i) = 0                                                    50
        END DO
C
C       Find the coordinates of the vertices in the local reference
C       frame according to equations 6.1, 6.2 and 6.3. Then perform
C       the two rotations phiz and thetay to get the coordinates in
C       master reference frame. Call other routine to find the
C       the body axes. Adjust the equations appropriately for each
C       class II triangle and repeat this process for all the 60
C       class II triangles in the calorimeter. Write the information
C       to files and manually remove some modules to make room for     60
C       beam entry and readout of inner detectors.
C
        AY = 63.4349488
        namin=1
        namax=6
        nthmin=1
        nthmax=2
        ntrmax=5
        DO NA = 1,6
              CAZ(NA) = COSD(AZ(NA))                                   70
              SAZ(NA) = SIND(AZ(NA))
        END DO

        S72 = SIND(72.)
        C72 = COSD(72.)
        C36 = COSD(36.)

        DO NA = NAMIN,NAMAX

          CAY = COSD(AY)                                               80
          SAY = SIND(AY)

          IF(NA.EQ.1)CAY = 1.
          IF(NA.EQ.1)SAY = 0.


          DO NTHET = NTHMIN, NTHMAX

            DO NC = 1,NTRMAX
              ANGLE = (NC-1)*72.                                       90
              n0=nc
              DO K = 1,3
                NZ = 10. - NX(K) - NY(K)
                X1 = NX(K)*S72
                Y1 = NY(K) + NX(K)*C72
                Z1 = 5. + NZ/(2.*C36)

                R1 = SQRT( X1**2 + Y1**2 )
```

```
            IF (r1.eq.0.0) THEN
               sphi1 = 0.0                                              100
            ELSE
               SPHI1 = X1/R1
            END IF

            PHI1 = ASIND(SPHI1)
            PHI_1 = PHI1 + ANGLE
            SPHI1 = SIND( PHI_1 )
            CPHI1 = COSD( PHI_1 )

            DEN1 = SQRT( X1**2 + Y1**2 + (Z1*C36)**2 )                  110
            STHETA1 = R1/DEN1
            CTHETA1 = Z1*C36/DEN1
            THETA_1 = ACOSD ( CTHETA1 )

            XX1 =STHETA1*CPHI1
            YY1 =STHETA1*SPHI1
            ZZ1 =CTHETA1

            IF (NTHET.EQ.2) THEN
              XX1 = -XX1                                                120
              YY1 = -YY1
              ZZ1 = -ZZ1
            END IF

            TX1 = XX1*CAY*CAZ(NA) - YY1*SAZ(NA) + ZZ1*CAZ(NA)*SAY
            TY1 = XX1*SAZ(NA)*CAY + YY1*CAZ(NA) + ZZ1*SAZ(NA)*SAY
            TZ1 = -XX1*SAY             +          ZZ1*CAY
*
            ARG = ABS(TY1) / SQRT (TX1**2 + TY1**2)
            PHI_1 = ASIND ( ARG )                                       130
            IF (TX1.LT.0.0.AND.TY1.GT.0.0) PHI_1 = 180.-PHI_1
            IF (TX1.LE.0.0.AND.TY1.LE.0.0) PHI_1 = 180.+PHI_1
            IF (TX1.GT.0.0.AND.TY1.LT.0.0) PHI_1 = 360.-PHI_1
            ARG = TZ1 / SQRT ( TX1**2 + TY1**2 + TZ1**2 )
            THETA_1 = ACOSD ( ARG )
*
            xb(k) = tx1
            yb(k) = ty1
            zb(k) = tz1
            END DO                                                      140
            CALL planehexb1
            CALL planehexb2
          END DO
        END DO
      END DO
      CLOSE (20)
      CLOSE (21)
```

```
      END
C
      SUBROUTINE  planehexb1                                          150
C
C      The calorimeter crystals are defined in Geant as
C      2 trapezoids connected together. This routine finds
C      the body axes of the first trapezoid of the truncated
C      pyramid hexB in the master refence frame where the
C      the modules are positioned and the tracking is performed.
C      the output of this routine is sent to heb.dat and used
C      as an input for the simulation codes (pibeta.f, ...etc).
C      The information needed is the polar angles of the body axes
C      of all the trapezoids which define the 50 hexb in the       160
C      calorimeter.
C
      IMPLICIT REAL*8 (a-h,o-z)
      REAL*8 n(3)
      CHARACTER*6 name(2)
      INTEGER*2 type(2),count
      COMMON/points/xb(3),yb(3),zb(3)
      COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
      COMMON/triangle/na,n0,nthet,count
      COMMON/xaxis/theta1,phi1                                        170
      COMMON/yaxis/theta2,phi2
      count=1
      n(1)=xb(1)
      n(2)=yb(1)
      n(3)=zb(1)
      CALL intersect
      CALL x_axis
      CALL y_axis
      type(1)=type(1)+1
      WRITE (20,200) name(1),type(1),n(1),n(2),n(3),                  180
     1               na,n0,nthet
      WRITE (21,201) theta1,phi1,theta2,phi2,
     1               theta3,phi3
201   FORMAT (2x,6F15.6)
200   FORMAT (5x,A9,2x,I3,2x,3F15.4,3I5)
      RETURN
      END
C
      SUBROUTINE  planehexb2
C                                                                     190
C      Find the body axes of the second trapezoid of
C      the hexagon B.
C
      IMPLICIT REAL*8 (a-h,o-z)
      REAL*8 n(3)
      CHARACTER*6 name(2)
```

```
      INTEGER*2 type(2),count
      COMMON/points/xb(3),yb(3),zb(3)
      COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
      COMMON/triangle/na,n0,nthet,count                                  200
      COMMON/xaxis/theta1,phi1
      COMMON/yaxis/theta2,phi2
      count=2
      n(1)=xb(1)
      n(2)=yb(1)
      n(3)=zb(1)
      CALL intersect
      CALL x_axis
      CALL y_axis
      type(2)=type(2)+1                                                  210
      WRITE (20,300) name(2),type(2),n(1),n(2),n(3),
     1               na,n0,nthet
      WRITE (21,301) theta1,phi1,theta2,phi2,
     1               theta3,phi3
301   FORMAT (2x,6F15.6)
300   FORMAT (5x,A9,2x,I3,2x,3F15.4,3I5)
      RETURN
      END
C
      SUBROUTINE findnormal                                              220
      IMPLICT REAL*8 (a-h,o-z)
      REAL*8 n(3),norm1,norm2
      COMMON/normdata/n(3),x1,x2,x3,y1,y2,y3,
     1                z1,z2,z3
      norm1=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)+
     1          (z2-z1)*(z2-z1))
      norm2=sqrt((x3-x1)*(x3-x1)+(y3-y1)*(y3-y1)+
     1          (z3-z1)*(z3-z1))
      temp = (x2-x1)*(x3-x1)+(y2-y1)*(y3-y1)+(z2-z1)*(z3-z1)
      ang = acos(temp/(norm1*norm2))                                     230
      n(1)= ((y3-y1)*(z2-z1)-(y2-y1)*(z3-z1))/(norm1*norm2*abs(sin(ang)))
      n(2)= ((x2-x1)*(z3-z1)-(z2-z1)*(x3-x1))/(norm1*norm2*abs(sin(ang)))
      n(3)= ((x3-x1)*(y2-y1)-(x2-x1)*(y3-y1))/(norm1*norm2*abs(sin(ang)))
      temp=sqrt(n(1)*n(1)+n(2)*n(2)+n(3)*n(3))
      DO i=1,3
        n(i)=n(i)/temp
      END DO
      RETURN
      END
C                                                                        240
      SUBROUTINE intersect
C
C     The Z axis points toward a geodesic vertex defined
C     the vector n(3). Find the coordinates of the Z axis
C     in the master reference frame.
```

```
C
      IMPLICIT REAL*8 (a-h,o-z)
      REAL*8 n(3)
      CHARACTER*6 name(2)
      INTEGER*2 type(2)                                                 250
      COMMON/points/xb(3),yb(3),zb(3)
      COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
      x=n(1)
      y=n(2)
      z=n(3)
      phi3=asind(abs(y)/sqrt(x*x+y*y))
      IF (x.LT.0.0.AND.y.GT.0.0) phi3 = 180.-phi3
      IF (x.LE.0.0.AND.y.LE.0.0) phi3 = 180.+phi3
      IF (x.GT.0.0.AND.y.LT.0.0) phi3 = 360.-phi3
      theta3=acosd(z/sqrt(x*x+y*y+z*z))                                 260
      RETURN
      END
C
      SUBROUTINE x_axis
      IMPLICIT REAL*8 (a-h,o-z)
      REAL*8 n(3)
      CHARACTER*6 name(2)
      INTEGER*2 type(2),count
      COMMON/xaxis/theta1,phi1
      COMMON/triangle/na,n0,nthet,count                                 270
      COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
      COMMON/points/xb(3),yb(3),zb(3)
C
C     Take an educated guess at the X axis
C
      IF (count.eq.2) THEN
         x=xb(3)-xb(2)
         y=yb(3)-yb(2)
         z=zb(3)-zb(2)
      ELSE                                                              280
         x=xb(2)-xb(3)
         y=yb(2)-yb(3)
         z=zb(2)-zb(3)
      END IF
      phi0=asind(abs(y)/sqrt(x*x+y*y))
      theta0=acosd(z/sqrt(x*x+y*y+z*z))
      IF (x.LT.0.0.AND.y.GT.0.0) phi0 = 180.-phi0
      IF (x.LE.0.0.AND.y.LE.0.0) phi0 = 180.+phi0
      IF (x.GT.0.0.AND.y.LT.0.0) phi0 = 360.-phi0
C       x0=-cosd(theta3)                                                290
C       y0=sind(theta3)*cosd(-phi1+phi3)
C       theta1=atan2d(x0,y0)
C
C     The previous guess at the X axis is very close to being
```

```
C       perpendicular to the Z axis. Now find the correct
C       X axis with a simple perturbative approach: find the
C       angular corrections sta-thet0 and tb to the polar angles
C       of the X axis such that the latter becomes perpendicular to Z.
C
        cg=cosd(theta3)*cosd(theta0)+sind(theta3)*                          300
       1      sind(theta0)*cosd(phi0-phi3)
        ga=acosd(cg)
        dph=(phi0-phi3)
        al=cosd(theta0)*sind(theta3)*sind(dph)
        de=cosd(theta0)*sind(theta3)*cosd(dph)-
       1   cosd(theta3)*sind(theta0)
        be=sind(ga)*cosd(dph)*sind(theta3)
        dl=sind(theta3)*sind(dph)*sind(ga)*cosd(theta3)
       1       *sind(theta0)
        ep=sind(ga)*sind(dph)*sind(theta3)                                  310
        g0=sind(ga)*cosd(theta3)
        a1=g0*g0-al*al+ep*ep
        b1=al*de-be*ep
        c1=be*be+g0*g0-de*de
        discr=b1*b1-a1*c1
        tb=atan2d(-b1,a1)
        phi1=phi0+tb
        sta=((al*be-dl)*sind(tb)-be*de*cosd(tb))/
       1        (de*(al*sind(tb)-de*cosd(tb)))
        theta1=acosd(sta)                                                   320
        RETURN
        END
C
        SUBROUTINE y_axis
C
C       find the Y axis as the cross product Z x X.
C
        IMPLICIT REAL*8 (a-h,o-z)
        REAL*8 norm1,norm2,n(3)
        CHARACTER*6 name(2)                                                 330
        INTEGER*2 type(2)
        COMMON/xaxis/theta1,phi1
        COMMON/yaxis/theta2,phi2
        COMMON/stuff/n(3),theta3,phi3,name(2),type(2)
        xx=sind(theta1)*cosd(phi1)
        xy=sind(theta1)*sind(phi1)
        xz=cosd(theta1)
        zx=sind(theta3)*cosd(phi3)
        zy=sind(theta3)*sind(phi3)
        zz=cosd(theta3)                                                     340
        norm1=sqrt(xx*xx+xy*xy+xz*xz)
        norm2=sqrt(zx*zx+zy*zy+zz*zz)
        temp=xx*zx+xy*zy+xz*zz
```

```
      ang=acos(temp/(norm1*norm2))
      x=(zy*xz-xy*zz)/(norm1*norm2*abs(sin(ang)))
      y=(xx*zz-zx*xz)/(norm1*norm2*abs(sin(ang)))
      z=(zx*xy-xx*zy)/(norm1*norm2*abs(sin(ang)))
      theta2=acosd(z)
      phi2=asind(abs(y)/sqrt(x*x+y*y))
      IF (x.LT.0.0.AND.y.GT.0.0) phi2 = 180.-phi2                    350
      IF (x.LE.0.0.AND.y.LE.0.0) phi2 = 180.+phi2
      IF (x.GT.0.0.AND.y.LT.0.0) phi2 = 360.-phi2
      RETURN
      END
```