```
      program gap
*
*     This progam creates a gap between all the modules in
*     calorimeter and find the coordinates of the modular
*     vertices as a result. The gap is needed for surface
*     treatments and the wrapping of the modules. A similar
*     program was written by Rick Marshall for a gap of 200
*     microns in MathCad. At the time when there was still
*     a debate about the actual size of the gap, this fortran
*     code was needed with the possibility of varying the gap
*     size. The general method consists of finding the solutions
*     to the equations describing the intersection of three planes
*     taking into account the gap and requiring that the z-components
*     of the resulting vertices be same, i.e., 26 cm (front)
*     and 48 cm (back). The output of this routine is shown in
*     chapter 5, tables 5.3 - 5.11 and has been submitted for the
*     fabrication of the modules. Only the solution for the pentagon
*     is given here. The other routines are similar.
*
*           Author: KETEVI ADIKLE ASSAMAGAN
*           Date: October 23 1992.
*
      implicit real*8 (a-h,o-z)
      real*8 gin(3),gout(3),xx(3),xy(3),xz(3),
     1       yx(3),yy(3),yz(3),zx(3),zy(3),
     1       zz(3),nt(3),ntp(3),mt(6,3),mtp(6,3),n(3),
     1       n0(3),nm(6,3),inv(3,3),m(6,3)
      common/matrices/xx(3),xy(3),xz(3),yx(3),
     1              yy(3),yz(3),zx(3),zy(3),
     1              zz(3),theta1,theta2,theta3
      common/pmatrix/nt(3),ntp(3),mt(6,3),mtp(6,3),theta,phi
      common/fixed/gin(3),gout(3),nunit
      common/inverse/inv(3,3),nm(6,3),n0(3),l,
     1               nvert,m(6,3)
      common/body/xg,yg,n(3)
*
      call define_constants
      call penta
      call hexa
      call hexb
      call hexc
      call hexd
      call hexd1
      call hexd2
      call veto1
      call veto2
      close (nunit)
      end
      subroutine define_constants
```

```
      implicit real*8 (a-h,o-z)                                          50
      real*8 gin(3),gout(3)
      common/fixed/gin(3),gout(3),nunit
      write(6,*) 'Supply clearance in cm: '
      read(5,*) sigma
      do i=1,3
         if (i.le.2) then
              gin(i)=sigma
              gout(i)=sigma
         else
              gin(i)=26.0                                                60
              gout(i)=48.0
         end if
      end do
      nunit=20
      open(nunit,file='gap.dat',status='unknown')
      write(nunit,50)
50    format(/,30x,'****************VARIABLE GAP DATA*************',/)
      return
      end
      subroutine penta                                                   70
      implicit real*8 (a-h,o-z)
      real*8 gin(3),gout(3),xx(3),xy(3),xz(3),
     1       yx(3),yy(3),yz(3),zx(3),zy(3),xout(3),
     1       zz(3),n(3),
     1       n0(3),nm(6,3),inv(3,3),m(6,3),xin(3)
      common/matrices/xx(3),xy(3),xz(3),yx(3),
     1                yy(3),yz(3),zx(3),zy(3),
     1                zz(3),theta1,theta2,theta3
      common/fixed/gin(3),gout(3),nunit
      common/inverse/inv(3,3),nm(6,3),n0(3),l,                           80
     1                nvert,m(6,3)
      common/body/xg,yg,n(3)
      parameter (pi=3.1415927)
      write(nunit,60)
*
      do j=1,6
         do i=1,3
            m(j,i)=0.0
         end do
      end do                                                             90
      do j=0,2
         if (j.eq.0) then
            xg=0.0
            yg=0.0
         else if (j.eq.1) then
            xg=0.0
            yg=1.0
         else
```

```
            xg=1.0
            yg=0.0                                                    100
          end if
          call body_axis
          if (j.eq.0) then
              do i=1,3
                n0(i)=n(i)
              end do
          else
              do i=1,3
                m(j,i)=n(i)
              end do                                                  110
          end if
        end do
        theta1=0.0
        theta2=0.0
        theta3=-72.0*(pi/180.0)
        call matrix
        nvert=5
        do j=3,nvert
          do i=1,3
           m(j,1)=m(j,1)+zx(i)*m(j-1,i)                             120
           m(j,2)=m(j,2)+zy(i)*m(j-1,i)
           m(j,3)=m(j,3)+zz(i)*m(j-1,i)
          end do
        end do
        do j=1,nvert
          call findnormal(j)
        end do
        do j=1,nvert
          l=j
          call invert_matrix                                          130
          do i=1,2
            xin(i)=0.0
            xout(i)=0.0
            do k=1,3
              xin(i)=xin(i)+inv(i,k)*gin(k)
              xout(i)=xout(i)+inv(i,k)*gout(k)
            end do
           end do
           xin(3) = gin(3)
           xout(3) = gout(3)                                          140
           write(nunit,100) xin(1),xin(2),xin(3),
     1                      xout(1),xout(2),xout(3)
100        format(5x,2f15.7,f15.7,2f15.7,f15.7)
        end do
60      format(/,35x,'*****************PENTA***************',/)
        return
        end
```

```
      subroutine body_axis
      implicit real*8 (a-h,o-z)
      real*8 n(3)                                                    150
      common/body/xg,yg,n(3)
      f=10.0
      s_72=sind(72.0)
      c_72=cosd(72.0)
      c_36=cosd(36.0)
      if ((xg.eq.0.0).and.(yg.eq.0.0)) then
         n(1)=0.0
         n(2)=0.0
         n(3)=1.0
      else                                                           160
         u=xg*c_72+yg
         t=xg*s_72
         v=(f-xg-yg+f*c_36)/2.0
         theta=atan2(sqrt(u*u+t*t),v)
         phi=atan2(t,u)
         n(1)=sin(theta)*cos(phi)
         n(2)=sin(theta)*sin(phi)
         n(3)=cos(theta)
      end if
      return                                                         170
      end
      subroutine dmatrix(nvert)
      implicit real*8 (a-h,o-z)
      real*8 nt(3),ntp(3),mt(6,3),mtp(6,3),nt0(3),mt0(6,3)
      common/pmatrix/nt(3),ntp(3),mt(6,3),mtp(6,3),theta,phi
      common/matrices/xx(3),xy(3),xz(3),yx(3),
     1                yy(3),yz(3),zx(3),zy(3),
     1                zz(3),theta1,theta2,theta3
      theta1=0.0
      theta2=theta                                                   180
      theta3=phi
      call matrix
      do j=0,nvert
        if (j.eq.0) then
          do i=1,3
            nt0(i)=0.0
          end do
        else
          do i=1,3
            mt0(j,i)=0.0                                             190
          end do
        end if
      end do
      do j=0,nvert
        if (j.eq.0) then
          do i=1,3
```

```
          nt0(1)=nt0(1)+zx(i)*nt(i)
          nt0(2)=nt0(2)+zy(i)*nt(i)
          nt0(3)=nt0(3)+zz(i)*nt(i)
         end do                                                    200
        else
         do i=1,3
          mt0(j,1)=mt0(j,1)+zx(i)*mt(j,i)
          mt0(j,2)=mt0(j,2)+zy(i)*mt(j,i)
          mt0(j,3)=mt0(j,3)+zz(i)*mt(j,i)
         end do
        end if
       end do
       do j=0,nvert
        if (j.eq.0) then                                           210
         do i=1,3
          ntp(1)=ntp(1)+yx(i)*nt0(i)
          ntp(2)=ntp(2)+yy(i)*nt0(i)
          ntp(3)=ntp(3)+yz(i)*nt0(i)
         end do
        else
         do i=1,3
          mtp(j,1)=mtp(j,1)+yx(i)*mt0(j,i)
          mtp(j,2)=mtp(j,2)+yy(i)*mt0(j,i)
          mtp(j,3)=mtp(j,3)+yz(i)*mt0(j,i)                         220
         end do
        end if
       end do
       return
       end
       subroutine matrix
       implicit real*8 (a-h,o-z)
       real*8 xx(3),xy(3),xz(3),yx(3),yy(3),yz(3),
      1       zx(3),zy(3),zz(3),theta1,theta2,theta3
       common/matrices/xx(3),xy(3),xz(3),yx(3),                    230
      1                yy(3),yz(3),zx(3),zy(3),
      1                zz(3),theta1,theta2,theta3
       xx(1)=1.0
       xx(2)=0.0
       xx(3)=0.0
       xy(1)=0.0
       xy(2)=cos(theta1)
       xy(3)=sin(theta1)
       xz(1)=0.0
       xz(2)=-sin(theta1)                                          240
       xz(3)=cos(theta1)
*
       yx(1)=cos(theta2)
       yx(2)=0.0
       yx(3)=-sin(theta2)
```

```
      yy(1)=0.0
      yy(2)=1.0
      yy(3)=0.0
      yz(1)=sin(theta2)
      yz(2)=0.0                                                    250
      yz(3)=cos(theta2)
*
      zx(1)=cos(theta3)
      zx(2)=sin(theta3)
      zx(3)=0.0
      zy(1)=-sin(theta3)
      zy(2)=cos(theta3)
      zy(3)=0.0
      zz(1)=0.0
      zz(2)=0.0                                                    260
      zz(3)=1.0
      return
      end
      subroutine findnormal(i)
      implicit real*8 (a-h,o-z)
      real*8 norm1,norm2,n0(3)
      real*8 inv(3,3),nm(6,3),m(6,3)
      common/inverse/inv(3,3),nm(6,3),n0(3),
     1               l,nvert,m(6,3)
*                                                                  270
      if (i.lt.nvert) then
         x1=m(i,1)
         y1=m(i,2)
         z1=m(i,3)
         x2=m(i+1,1)
         y2=m(i+1,2)
         z2=m(i+1,3)
      else
         x1=m(nvert,1)
         y1=m(nvert,2)                                             280
         z1=m(nvert,3)
         x2=m(1,1)
         y2=m(1,2)
         z2=m(1,3)
      end if
      norm1=sqrt(x1*x1+y1*y1+z1*z1)
      norm2=sqrt(x2*x2+y2*y2+z2*z2)
      temp=x1*x2+y1*y2+z1*z2
      ang=acos(temp/(norm1*norm2))
      nm(i,1)=(y1*z2-y2*z1)/(norm1*norm2*abs(sin(ang)))            290
      nm(i,2)=(z1*x2-x1*z2)/(norm1*norm2*abs(sin(ang)))
      nm(i,3)=(x1*y2-x2*y1)/(norm1*norm2*abs(sin(ang)))
      return
      end
```

```
      subroutine invert_matrix
      implicit real*8 (a-h,o-z)
      real*8 tr(3,3),n0(3),nm(6,3),mat(3,3),m(6,3),
     1       inv(3,3)
      common/inverse/inv(3,3),nm(6,3),n0(3),
     1               l,nvert,m(6,3)                                    300
      if (l.lt.nvert) then
         do i=1,3
            mat(i,1)=nm(l,i)
            mat(i,2)=nm(l+1,i)
            mat(i,3)=n0(i)
          end do
      else
         do i=1,3
            mat(i,1)=nm(nvert,i)
            mat(i,2)=nm(1,i)                                           310
            mat(i,3)=n0(i)
         end do
      end if
      do i=1,3
         do j=1,3
            tr(i,j)=mat(j,i)
         end do
      end do
      d1=tr(1,1)*(tr(2,2)*tr(3,3)-tr(2,3)*tr(3,2))
      d2=tr(1,2)*(tr(3,1)*tr(2,3)-tr(2,1)*tr(3,3))                     320
      d3=tr(1,3)*(tr(2,1)*tr(3,2)-tr(3,1)*tr(2,2))
      det=d1+d2+d3
      inv(1,1)=(tr(2,2)*tr(3,3)-tr(3,2)*tr(2,3))/det
      inv(1,2)=(tr(1,3)*tr(3,2)-tr(1,2)*tr(3,3))/det
      inv(1,3)=(tr(1,2)*tr(2,3)-tr(2,2)*tr(1,3))/det
      inv(2,1)=(tr(3,1)*tr(2,3)-tr(2,1)*tr(3,3))/det
      inv(2,2)=(tr(1,1)*tr(3,3)-tr(3,1)*tr(1,3))/det
      inv(2,3)=(tr(2,1)*tr(1,3)-tr(1,1)*tr(2,3))/det
      inv(3,1)=(tr(2,1)*tr(3,2)-tr(3,1)*tr(2,2))/det
      inv(3,2)=(tr(1,3)*tr(1,2)-tr(1,1)*tr(3,2))/det                   330
      inv(3,3)=(tr(1,1)*tr(2,2)-tr(2,1)*tr(1,2))/det
      return
      end
```