

```

c-----Usable columns for a Fortran 77 program (1-72)-----
c
c Declare variables
c
c-----
c
  implicit real*8 (a,d,l,p,r,t,v,x,y,z)
  implicit integer(b,e,i,j,k,n,m)
  real*8 NVar(1:70000,1:3)
  real*8 NVar3D(1:150000,1:5)
  real*8 NVar3D2(1:150000,1:4)
  integer EN(1:70000,1:5)
  integer EN3D(1:150000,1:9)
  integer EN2(1:150000,1:9)
c
c-----
c
c Define primary dimension variables
c
c-----
c
c Diameter in inches -> converted to mm later
c
  Dia=1.5
c
c Rad2 must be < Dia/2
c
  Rad2=(Dia/2)-(0.15*Dia)
c
c Length of dimension A
c
  LA=(2.0/3.0)*Dia
c
c Length of section LC
c
  LC=(1.0/3.0)*Dia
c
c-----
c
c Define secondary dimension variables
c
c-----
c
c Radius
c
  Rad=Dia/2
c
c Set the thickness
c
  Thickness=Rad-Rad2
c
c Set the Length
c
  Length=(22.0/3.0)*Dia
c
c Positive angle of impact in degrees -> converted to rad later
c
  t2p=0.0
c
c Initial distance from the nose to the target in inches (converted to
mm later)

```

```

c
c     Zsep=0.1
c
c-----
c
c Define mesh variables
c
c-----
c
c Mesh type: [1] one materials, [2] 2 material
c
c     MeshType=2
c
c Rad3 must be < Rad2
c
c     Rad3=0.75*Rad2
c
c Rad4 must be < Rad3
c
c     Rad4=0.4*Rad3
c
c Number of layers in the curved end section
c
c     NLEcur=10
c
c Number of radial elements between Rad and Rad2
c
c     NLEshe=3
c
c Number of radial elements between Rad2 and Rad3
c
c     NLEcer=2
c
c Number of radial elements between Rad3 and Rad4 (including the trans.
c layer)
c
c     NLElay2=3
c
c Number of layers in the middle
c
c     NLEmid=60
c
c-----
c
c ATTENTION:      After this point, none of the variables are meant to be
c                  changed without rewriting code
c
c-----
c
c-----
c
c Write format statements
c
c-----
c
101  format (I8)
102  format (5(I8))
103  format (I8,I5,3(E20.12),I5)
104  format (I8,I5,8(I8))
105  format (A17)
106  format (A27)

```

```

107  format (A7,I8,A4,I8,A21)
108  format (3 (E20.12,1X),E20.12)
109  format (7 (I8,2X),I8)
110  format (A10,I2,A4,F10.5,A3)
111  format (E20.12)
c
c-----
c
c Open the Penetrator info file
c
c-----
c
c      open(9,file='Penetrator Info.txt')
c      write(9,*)' '
c
c-----
c
c Convert units
c
c-----
c
c      Dia=Dia*25.4
c
c      Rad2=Rad2*25.4
c
c      LA=LA*25.4
c
c      LC=LC*25.4
c
c      Rad=Rad*25.4
c
c      Thickness=Thickness*25.4
c
c      Length=Length*25.4
c
c      Zsep=Zsep*25.4
c
c Convert angle of incidence to radians
c
c      pi=3.14159265359
c
c      t2p=t2p*(pi/180)
c
c      Rad3=Rad3*25.4
c
c      Rad4=Rad4*25.4
c
c-----
c
c Rename variables
c
c-----
c
c      rTrans=Rad3
c
c      rIS=Rad4
c
c      Ltip=LA
c
c-----
c

```

```

c Define other constants
c
c-----
c
c Number of elements in the theta (divisible by 9!!!)
c
    NLEthe=36
c
c All BC's are initially zero
c
    BC=0
c
c Total number of layers in the penetrator
c
    NLE=NLEcur+NLEcer+NLEshe+NLEmid
c
c-----
c
c Set material constants and dependent variables
c
c-----
c
    if (MeshType.eq.1) then
        write(9,*)'1 material penetrator'
        write(9,*)' '
        Mat2=2
        Mat3=2
    else if (MeshType.eq.2) then
        write(9,*)'2 material penetrator'
        write(9,*)' '
        Mat2=2
        Mat3=3
    else if (MeshType.eq.3) then
        write(9,*)'Hollow penetrator'
        write(9,*)' '
        Mat2=2
        Mat3=3
    else
        write(*,*)' '
        write(*,*)'MeshType is preset'
        write(*,*)' '
        write(*,*)'1 material penetrator'
        write(9,*)'1 material penetrator'
        write(9,*)' '
        Mat2=2
        Mat3=2
    end if
c
c-----
c
c Check input for valid values
c
c-----
c
    if (Rad2.gt.Rad) then
        write(*,*)' '
        write(*,*)'Thicknesss exceeds max, use (Rad-rTrans)*0.5'
        Thickness=(Rad-rTrans)*0.5
    end if
c
c-----

```

```

c
c Create the outer layer of nodes
c
c-----
c
c Create the theta increments
c
      tmax=pi
      tmin=0.0
      delt=(tmax-tmin)/(NLEthe)
c
c Create r increments before the thickness
c
      delr1=Thickness/(NLEshe)
c
c Create r increments after the thickness
c
      delr2=((Rad-Thickness)-rTrans)/(NLEcer)
c
c Set the node counter to zero
c
      n=0
c
c Start at rmax and tmin and go to the thickness
c
      r=Rad
      do i=1,NLEshe+NLEcer+1
        t=tmin
        do j=1,NLEthe+1
          x=r*COS(t)
          y=r*SIN(t)
          n=n+1
          NVar(n,1)=x
          NVar(n,2)=y
          NVar(n,3)=r
          t=t+delt
        end do
        if (i.le.NLEshe) then
          delr=delr1
        else if (i.eq.NLEshe+1) then
          temp=Rad-NVar(n,3)
          delr=delr2
        else
          delr=delr2
        end if
        r=r-delr
      end do
c
      write(9,*)'The thickness is',temp,' mm'
      temp=temp/25.4
      write(9,*)'           ',temp,' in'
      write(9,*)' '
c
c ns = the # of the last node in the outer section
c
      ns=n
c-----
c
c Create the outer area of elements
c
c-----

```

```

c
c Start at element zero
c
c     e=0
c
c Material type is 2
c
c     do i=1,NLEshe+NLEcer
c       do j=1,NLEthe
c         n1=j+((NLEthe+1)*i)
c         n4=j+((NLEthe+1)*(i-1))
c         n2=n1+1
c         n3=n4+1
c         e=e+1
c         EN(e,1)=n1
c         EN(e,2)=n2
c         EN(e,3)=n3
c         EN(e,4)=n4
c         if (i.le.NLEshe) then
c           EN(e,5)=Mat2
c         else
c           EN(e,5)=Mat3
c         end if
c       end do
c     end do
c
c -----
c
c Create the increments for the transition and next section of nodes
c
c -----
c
c Redefine the theta increments
c
c     delt=delt*3.0
c
c Redefine the number of theta layers
c
c     NLEthe=NLEthe/3
c
c Create r increments before the thickness
c
c     rmax=rTrans
c     rmin=rIS
c     delr=(rmax-rmin)/NLElay2
c
c -----
c
c Create a middle row of transition nodes
c
c -----
c
c     r1=rmax-delr*0.4
c     r2=rmax-delr
c
c     do i=1,NLEthe
c       tmid=(tmin+(delt*0.5))+delt*(i-1)
c       x=r1*COS(tmid-(delt*0.15))
c       y=r1*SIN(tmid-(delt*0.15))
c       n=n+1
c       NVar(n,1)=x

```

```

        NVar(n,2)=y
        NVar(n,3)=r1
        x=r1*COS(tmids+(delt*0.15))
        y=r1*SIN(tmids+(delt*0.15))
        n=n+1
        NVar(n,1)=x
        NVar(n,2)=y
        NVar(n,3)=r1
    end do
c
c nm = the # of the last node in the middle row of transition nodes
c
    nm=n
c
c-----
c
c Create an outer row of transition nodes
c
c-----
c
    x=r2
    y=0.0
    n=n+1
    NVar(n,1)=x
    NVar(n,2)=y
    NVar(n,3)=r2
    do i=1,NLEthe
        tmids=(tmin+(delt*0.5))+delt*(i-1)
        x=r2*COS(tmids+(delt*0.5))
        y=r2*SIN(tmids+(delt*0.5))
        n=n+1
        NVar(n,1)=x
        NVar(n,2)=y
        NVar(n,3)=r2
    end do
c
c n1 = the # of the last node the outer row of transition nodes
c
    n1=n
c
c-----
c
c Create the transition elements which are now in the mat3 region
c
c-----
c
    n1=ns-(NLEthe*3+1)+1
    n2=n1+1
    n3=n2+1
    n4=n3+1
    n5=ns+1
    n6=n5+1
    n7=nm+1
    n8=n7+1
c
    e=e+1
    EN(e,1)=n1
    EN(e,2)=n7
    EN(e,3)=n5
    EN(e,4)=n2
    EN(e,5)=Mat3

```

```

e=e+1
EN(e,1)=n2
EN(e,2)=n5
EN(e,3)=n6
EN(e,4)=n3
EN(e,5)=Mat3
e=e+1
EN(e,1)=n3
EN(e,2)=n6
EN(e,3)=n8
EN(e,4)=n4
EN(e,5)=Mat3
e=e+1
EN(e,1)=n7
EN(e,2)=n8
EN(e,3)=n6
EN(e,4)=n5
EN(e,5)=Mat3
c
do i=1,NLEthe-1
  n1=n4
  n2=n1+1
  n3=n2+1
  n4=n3+1
  n5=n6+1
  n6=n5+1
  n7=n8
  n8=n7+1
  e=e+1
  EN(e,1)=n1
  EN(e,2)=n7
  EN(e,3)=n5
  EN(e,4)=n2
  EN(e,5)=Mat3
  e=e+1
  EN(e,1)=n2
  EN(e,2)=n5
  EN(e,3)=n6
  EN(e,4)=n3
  EN(e,5)=Mat3
  e=e+1
  EN(e,1)=n3
  EN(e,2)=n6
  EN(e,3)=n8
  EN(e,4)=n4
  EN(e,5)=Mat3
  e=e+1
  EN(e,1)=n7
  EN(e,2)=n8
  EN(e,3)=n6
  EN(e,4)=n5
  EN(e,5)=Mat3
end do
c
c-----
c
c Create the nodes for the second section
c
c-----
c
c 3 layers are already in place so we begin a step ahead of r2

```

```

c
  r=r2-delt
c
  do i=1,NLElay2-1
c
    do i=1,2
      t=tmin
        do j=1,NLEthe+1
          x=r*COS(t)
          y=r*SIN(t)
          n=n+1
          NVar(n,1)=x
          NVar(n,2)=y
          NVar(n,3)=r
          t=t+delt
        end do
      r=r-delt
    end do
c
c-----
c
c Create the elements for the second section
c-----
c
c Material type is 3 (ceramic)
c
c ns= the # of the first node in the this section
c
  ns=n1-NLEthe
c
  do i=1,NLElay2-1
c
    do i=1,2
      do j=1,NLEthe
        n4=ns+(j-1)+(NLEthe+1)*(i-1)
        n1=ns+(j-1)+(NLEthe+1)*(i)
        n2=n1+1
        n3=n4+1
        e=e+1
        EN(e,1)=n1
        EN(e,2)=n2
        EN(e,3)=n3
        EN(e,4)=n4
        EN(e,5)=Mat3
      end do
    end do
c
c ns = the # of the last node in this second section
c
  ns=n
c-----
c
c Create the increments for the 2nd transition
c-----
c
c Redefine the theta increments
c
  tmax=pi
  tmin=0.0

```

```

        delt=delt*3.0
c
c Redefine the number of theta layers
c
        NLEthe=NLEthe/3
c
c Create r increments before the thickness
c
        r1=rIS*0.7
        r2=rIS*0.5
c
c-----
c
c Create a middle row of transition nodes
c
c-----
c
        do i=1,NLEthe
            tmid=(tmin+(delt*0.5))+delt*(i-1)
            x=r1*COS(tmid-(delt*0.15))
            y=r1*SIN(tmid-(delt*0.15))
            n=n+1
            NVar(n,1)=x
            NVar(n,2)=y
            NVar(n,3)=r1
            x=r1*COS(tmid+(delt*0.15))
            y=r1*SIN(tmid+(delt*0.15))
            n=n+1
            NVar(n,1)=x
            NVar(n,2)=y
            NVar(n,3)=r1
        end do
c
c nm = the # of the last node in the middle row of transition nodes
c
        nm=n
c
c-----
c
c Create an outer row of transition nodes
c
c-----
c
        x=r2
        y=0.0
        n=n+1
        NVar(n,1)=x
        NVar(n,2)=y
        NVar(n,3)=r2
        do i=1,NLEthe
            tmid=(tmin+(delt*0.5))+delt*(i-1)
            x=r2*COS(tmid+(delt*0.5))
            y=r2*SIN(tmid+(delt*0.5))
            n=n+1
            NVar(n,1)=x
            NVar(n,2)=y
            NVar(n,3)=r2
        end do
c
c nl = the # of the last node the outer row of transition nodes
c

```

```

n1=n
c
c-----
c
c Create the transition elements
c-----
c
c
n1=ns-(NLEthe*3+1)+1
n2=n1+1
n3=n2+1
n4=n3+1
n5=ns+1
n6=n5+1
n7=nm+1
n8=n7+1
c
e=e+1
EN(e,1)=n1
EN(e,2)=n7
EN(e,3)=n5
EN(e,4)=n2
EN(e,5)=Mat3
e=e+1
EN(e,1)=n2
EN(e,2)=n5
EN(e,3)=n6
EN(e,4)=n3
EN(e,5)=Mat3
e=e+1
EN(e,1)=n3
EN(e,2)=n6
EN(e,3)=n8
EN(e,4)=n4
EN(e,5)=Mat3
e=e+1
EN(e,1)=n7
EN(e,2)=n8
EN(e,3)=n6
EN(e,4)=n5
EN(e,5)=Mat3
c
do i=1,NLEthe-1
n1=n4
n2=n1+1
n3=n2+1
n4=n3+1
n5=n6+1
n6=n5+1
n7=n8
n8=n7+1
e=e+1
EN(e,1)=n1
EN(e,2)=n7
EN(e,3)=n5
EN(e,4)=n2
EN(e,5)=Mat3
e=e+1
EN(e,1)=n2
EN(e,2)=n5
EN(e,3)=n6

```

```

        EN(e,4)=n3
        EN(e,5)=Mat3
        e=e+1
        EN(e,1)=n3
        EN(e,2)=n6
        EN(e,3)=n8
        EN(e,4)=n4
        EN(e,5)=Mat3
        e=e+1
        EN(e,1)=n7
        EN(e,2)=n8
        EN(e,3)=n6
        EN(e,4)=n5
        EN(e,5)=Mat3
    end do
c
c-----
c
c Create the last node and the last two elements
c
c-----
c
    x=0.0
    y=0.0
    n=n+1
    NVar(n,1)=x
    NVar(n,2)=y
    NVar(n,3)=sqrt((x)**2+(y)**2)
c
c n1 = The last node
c
    n1=n
c
    n1=n1
    n2=n1-3
    n3=n1-4
    n4=n1-5
    e=e+1
    EN(e,1)=n1
    EN(e,2)=n2
    EN(e,3)=n3
    EN(e,4)=n4
    EN(e,5)=Mat3
    n2=n1-1
    n3=n1-2
    n4=n1-3
    e=e+1
    EN(e,1)=n1
    EN(e,2)=n2
    EN(e,3)=n3
    EN(e,4)=n4
    EN(e,5)=Mat3
c
    write(9,*)'There are',n,' 2D nodes.'
    write(9,*)'There are',e,' 2D elements.'
c
    NumN2D=n
    NumE2D=e
c
c-----
c

```

```
c Create the z increments for extruding the nodes
```

```
c
```

```
c-----
```

```
c
```

```
z1=Length  
z2=z1-Thickness  
z3=z1-(Rad-rTrans)  
z4=LC  
z5=0.0
```

```
c
```

```
delz12=(z1-z2)/NLEshe  
delz23=(z2-z3)/NLEcer  
delz34=(z3-z4)/NLEmid  
delz45=(z4-z5)/NLEcur
```

```
c
```

```
c-----
```

```
c
```

```
c Extrude the first section of nodes
```

```
c
```

```
c-----
```

```
c
```

```
c Reset the node counter
```

```
c
```

```
n=0
```

```
c
```

```
zmax=z1  
zmin=z2
```

```
c
```

```
c Start at zmax
```

```
c
```

```
z=zmax
```

```
c
```

```
do i=1,NLEshe+1  
  do j=1,NumN2D  
    n=n+1  
    NVar3D(n,1)=NVar(j,1)  
    NVar3D(n,2)=NVar(j,2)  
    NVar3D(n,3)=z  
  end do  
  z=z-delz12  
end do
```

```
c
```

```
c-----
```

```
c
```

```
c Extrude the second section of nodes
```

```
c
```

```
c-----
```

```
c
```

```
zmax=z2  
zmin=z3
```

```
c
```

```
c Start at zmax minus one step
```

```
c
```

```
z=zmax-delz23
```

```
c
```

```
do i=1,NLEcer  
  do j=1,NumN2D  
    n=n+1  
    NVar3D(n,1)=NVar(j,1)  
    NVar3D(n,2)=NVar(j,2)  
    NVar3D(n,3)=z
```



```

c Now add the curvature to last end section
c
c-----
c
c      n=0
c
c      NLEtemp=NLEshe+NLEcer+NLEmid+1
c
c      do i=1,NLEtemp
c        do j=1,NumN2D
c          n=n+1
c        end do
c      end do
c
c      do i=1,NLEcur
c        zstep=(NLEcur+1-(i+1))*1.0
c        zfactor=1-zstep/(NLEcur)
c        do j=1,NumN2D
c          n=n+1
c          rmult=COS((NVar(j,3)*pi)/Dia)
c          zextra=Ltip*rmult*zfactor
c          z=NVar3D(n,3)
c          znew=z-zextra
c          NVar3D(n,3)=znew
c        end do
c      end do
c
c      temp=(NVar3D(1,3)-NVar3D(NumN,3))
c      write(9,*)' '
c      write(9,*)'With curvature, the length is',temp,' mm'
c      temp=temp/25.4
c      write(9,*)'                               ',temp,' in'
c
c-----
c
c Extrude the elements
c
c-----
c
c      e=0
c
c Elements in the end cap
c
c      do i=1,NLE
c        do j=1,NumE2D
c          e=e+1
c          EN3D(e,1)=EN(j,1)+NumN2D*(i-1)
c          EN3D(e,2)=EN(j,2)+NumN2D*(i-1)
c          EN3D(e,3)=EN(j,3)+NumN2D*(i-1)
c          EN3D(e,4)=EN(j,4)+NumN2D*(i-1)
c          EN3D(e,5)=EN3D(e,1)+NumN2D
c          EN3D(e,6)=EN3D(e,2)+NumN2D
c          EN3D(e,7)=EN3D(e,3)+NumN2D
c          EN3D(e,8)=EN3D(e,4)+NumN2D
c          if (i.le.NLEshe) then
c            EN3D(e,9)=Mat2
c          else if (i.le.(NLE-NLEcur)) then
c            EN3D(e,9)=EN(j,5)
c          else
c            EN3D(e,9)=Mat2
c          end if
c        end do
c      end do

```

```

        end do
    end do
c
    NumE=e
c
    write(9,*)' '
    write(9,*)'There are',NumN,' 3D nodes.'
    write(9,*)'There are',NumE,' 3D elements.'
c
    write(*,*)' '
    write(*,*)"Penetrator Info.txt" has been written.'
c
c-----
c
c Move the tip of the penetrator to the origin
c
c-----
c
    do i=1,NumN
        NVar3D(i,3)=NVar3D(i,3)+Ltip
    end do
c
c-----
c
c Move the penetrator completely in the positive quadrant
c
c-----
c
    do i=1,NumN
        NVar3D(i,1)=NVar3D(i,1)+Dia
        NVar3D(i,3)=NVar3D(i,3)+Dia
    end do
c
c-----
c
c Transform to the angle of incidence
c
c-----
c
    write(9,*)' '
    write(9,*)'Angle of incidence is',t2p,' radians'
    temp=t2p*(180/pi)
    write(9,*)'Angle of incidence is',temp,' degrees'
c
    do i=1,NumN
        xo=NVar3D(i,1)
        zo=NVar3D(i,3)
        r=xo
        x1n=r*COS(-t2p)
        z1n=r*SIN(-t2p)
        r=zo
        x2n=r*SIN(t2p)
        z2n=r*COS(t2p)
        xn=x1n+x2n
        zn=z1n+z2n
        NVar3D(i,1)=xn
        NVar3D(i,3)=zn
    end do
c
c Find the unit vector for the angle of incidence (toward the target)
c

```

```

dx=-SIN(t2p)
dy=0.0
dz=-COS(t2p)
c
write(9,*)' '
write(9,*)'The unit vector (d) for the vector of incidence is:'
write(9,*)' '
write(9,*)'      dx =',dx
write(9,*)'      dy =',dy
write(9,*)'      dz =',dz
c
c-----
c
c Move the tip of the penetrator back to the origin
c
c-----
c
do i=1,NumN
  NVar3D(i,1)=NVar3D(i,1)-NVar3D(NumN,1)
  NVar3D(i,3)=NVar3D(i,3)-NVar3D(NumN,3)
end do
c
c-----
c
c Move the tip of the penetrator back along the vector of incidence
c
c-----
c
do i=1,NumN
  NVar3D(i,1)=NVar3D(i,1)+Zsep*abs(dx)
  NVar3D(i,2)=NVar3D(i,2)+Zsep*abs(dy)
  NVar3D(i,3)=NVar3D(i,3)+Zsep*abs(dz)
end do
c
c-----
c
c Add element types to the proper nodes
c
c-----
c
do i=1,NumE
  MatE=EN3D(i,9)
  do j=1,8
    n=EN3D(i,j)
    vMatN=NVar3D(n,4)
    if (vMatN.eq.0.0) then
      NVar3D(n,4)=MatE*1.0
    else if (MatE.lt.vMatN) then
      NVar3D(n,4)=MatE*1.0
    end if
  end do
end do
c
c-----
c
c Write the dyna3d slave surface cards
c
c-----
c
open(1,file='D P SS Cards.txt')
c

```

```

n=0
c
do i=(NumE-NumE2D),NumE
  n=n+1
  n1=EN3D(i,5)
  n2=EN3D(i,6)
  n3=EN3D(i,7)
  n4=EN3D(i,8)
  write(1,102)n,n1,n2,n3,n4
end do
c
close(1)
c
write(*,*)' '
write(*,*)"D P SS Cards.txt" has been written.'
c
c-----
c
c Create the 2-D ANSTS test file
c
c-----
c
open(1,file='ANSYS Contact Surface Plot.txt')
write(1,*)'/FILNAM, Contact Surface'
write(1,*)'/prep7'
write(1,*)'/TITLE, Contact Surface'
write(1,*)'ANTYPE,static'
write(1,*)'!*'
write(1,*)'et,1,plane42'
write(1,*)'!*'
do i=(NumN-NumN2D),NumN
  write(1,*)'n,',i,',',NVar3D(i,1),',',NVar3D(i,2)
end do
write(1,*)'!*'
do i=(NumE-NumE2D),NumE
  write(1,*)'Mat,',EN3D(i,9)
  n1=EN3D(i,5)
  n2=EN3D(i,6)
  n3=EN3D(i,7)
  n4=EN3D(i,8)
  write(1,*)'e,',n1,',',n2,',',n3,',',n4
end do
write(1,*)'epplot'
write(1,*)'/number,1'
write(1,*)'/pnum,mat,1'
write(1,*)'/replot'
close(1)
c
write(*,*)' '
write(*,*)"ANSYS Contact Surface Plot.txt" has been written.'
c
c-----
c
c Write the dyna3d node cards
c
c-----
c
open(1,file='D P N Cards.txt')
c
c Write the 10 info lines
c

```

```

write(1,101)NumN2D
write(1,101)NumE2D
write(1,101)NumN
write(1,101)NumE
write(1,111)dx
write(1,111)dy
write(1,111)dz
write(1,*)'line 8'
write(1,*)'line 9'
write(1,*)'line 10'
c
  ymin=0.0
c
c Write each card while checking and instituting the proper BC
c
c BCt=2 - no y translation (symmetry)
c BCr=0 - no rotational BC
c
c Set a tolerance for locations
c
  Tol=0.00001
c
  BCr=0
c
  do i=1,NumN
    x=NVar3D(i,1)
    y=NVar3D(i,2)
    z=NVar3D(i,3)
    if (y.lt.(ymin+Tol).and.y.gt.(ymin-Tol)) then
      BCt=2
    else
      BCt=0
    end if
    write(1,103)i,BCt,x,y,z,BCr
  end do
c
  close(1)
c
  write(*,*)' '
  write(*,*)'"D P N Cards.txt" has been written.'
c
c-----
c
c Write the dyna3d element cards
c
c-----
c
  open(1,file='D P E Cards.txt')
c
  do i=1,NumE
    n1=EN3D(i,1)
    n2=EN3D(i,2)
    n3=EN3D(i,3)
    n4=EN3D(i,4)
    n5=EN3D(i,5)
    n6=EN3D(i,6)
    n7=EN3D(i,7)
    n8=EN3D(i,8)
    Mat=EN3D(i,9)
    write(1,104)i,Mat,n1,n2,n3,n4,n5,n6,n7,n8
  end do

```

```

c
    close(1)
c
    write(*,*)' '
    write(*,*)'"D P E Cards.txt" has been written.'

```