

4. GRID METHOD

The central issue concerning fabric structures as they began to become more common was how to construct shape. There is a more general discussion of this matter in the author's book cited earlier in these Notes. The discussion here will be limited to the "grid method"¹.

In the grid method you begin with a cable net that is in equilibrium in the horizontal plane. Equilibrium in the vertical direction then can be used to obtain the z coordinates of the nodes. (The shape.) Solving for the shape then requires the solution of linear equations.

When there are no edge cables it is a trivial matter to set up a grid that is in equilibrium in the horizontal plane. Edge cables were an annoyance which required the designer go off and solve a separate cable problem to get the forces and locations in the edge cable segments. In this system edge cables are handled automatically.

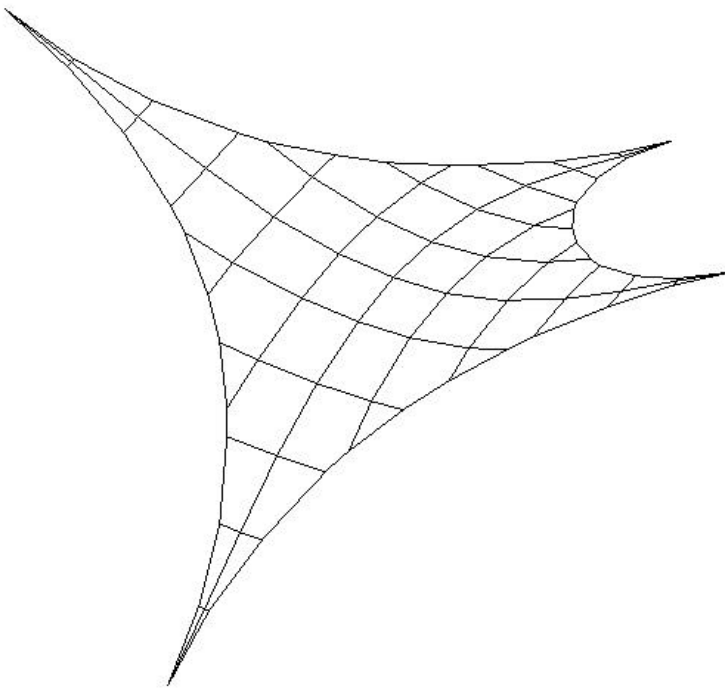


Figure 4.1 Hyperbolic Paraboloid

¹ "Stress Analysis of Prestressed Suspended Roofs" by A. Siev and J. Eidelman, *Proc. ASCE*, 90, ST4, pp. 103-121, August 1964.

In the system described here, a batch file, HOR.BAT is used to control the flow of a series of programs:

Hor.bat

```
horst fort.50 fort.60 fort.70
tr3dom fort.70 fort.60 fort.600
tr3do1 fort.600 fort.60 fort.601
tr3do2 fort.601 fort.60 fort.602
refl fort.602 fort.60
ali fort.602 fort.10 pat.dat
```

The program HORST.FOR generates a node map in the following manner:

- The input
6 12. 12 24.

85	86	87	88	89	90	91
78	79	80	81	82	83	84
71	72	73	74	75	76	77
64	65	66	67	68	69	70
57	58	59	60	61	62	63
50	51	52	53	54	55	56
43	44	45	46	47	48	49
36	37	38	39	40	41	42
29	30	31	32	33	34	35
22	23	24	25	26	27	28
15	16	17	18	19	20	21
8	9	10	11	12	13	14
1	2	3	4	5	6	7

Figure 4.2
A Rectangular Grid

generates the grid shown in Figure 4.2

This grid has 6 spaces at 2' in the x direction and 12 spaces at 2' in the y direction. Using symmetry it is only necessary to consider 1/2 on the paraboloid of Figure 4.1.

- The next step is the introduction of the edge cables. This is done using the input

49	85	3.5
1	49	3.5
0	0	0.

Moving around the grid in a clockwise manner this input defines two edge cables :

One cable with a +end of 49, a -end of 85, and an offset of 3.5'

One cable with a +end of 1, a -end of 49, and an offset of 3.5'

The row of zeroes indicates the end of the list

This input produces the node map of Figure 4.3.

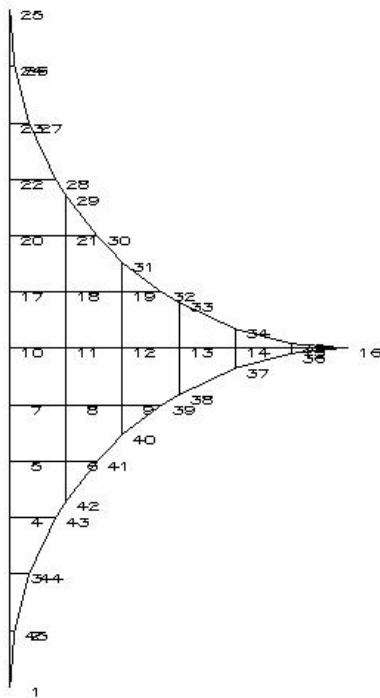


Figure 4.3
1/2 of a Hyperbolic Paraboloid

At this point the forces in the edge cables are approximate since they were computed as if the cables were circular arcs.

4.1 SMOOTHING

The actual cable forces in the horizontal plane will now be computed using a nonlinear analysis program. This step will be called "smoothing"² here and a version of it will be used again in the material that follows.

At this point the problem is to get the actual edge cable forces. When done by hand this simply requires the solution of cable forces using equilibrium as is done in elementary statics courses. That step is done here using "smoothing" as follows. You would like to keep the given forces in the cables that are not edge cables and let the edge cable forces be determined by equilibrium. That is done here by assigning fictitious moduli of zero to the interior cables and then letting the structure move slightly to find an equilibrium position. Since the cables all have a geometric stiffness since the cable forces are not zero, the structure does not become singular during this step.

This step is performed by TR3DOM.FOR. It requires the description of the fixed (support) points of the structure using the following list of node numbers and elevations

```
25 .0001
16 10.
1 .0001
0 0.
```

In this list the small numbers are used to represent the elevation zero and the last set of zeroes are used to terminate the list. The computer also asks about y-axis symmetry. The user responds with 2 in this case which does have y-axis symmetry. If not, 0 is the proper response.

5. SHAPE FINDING

In the next step the computer uses the program TR3DO1.FOR to determine shape. (Find the elevations of the nodes) After this has been done the structure has the appearance of 1/2 of Figure 4.1.

6. GEODESIC SHAPE

At this point the forces in the interior cables are no longer the initial forces that were specified in the design due to the fact that the cables are no longer horizontal. In the hope of trying to use the material of the structure in some optimal fashion, a shape with uniform interior cable forces is sometimes desired. (This may of course not be possible!) For this structure that shape is obtained by *smoothing* again, this time using the program TR3DO2.FOR which produces the result shown in Figure 4.4. It can be seen in this figure that the nodes have shifted to achieve this result.

² "Initial Equilibrium Solution Method for Cable Reinforced Membranes, Part 1" by R. B. Haber and J. F. Abel, *Computer Methods in Applied Mechanics and Engineering*, 30, pp. 263-284, 1982.

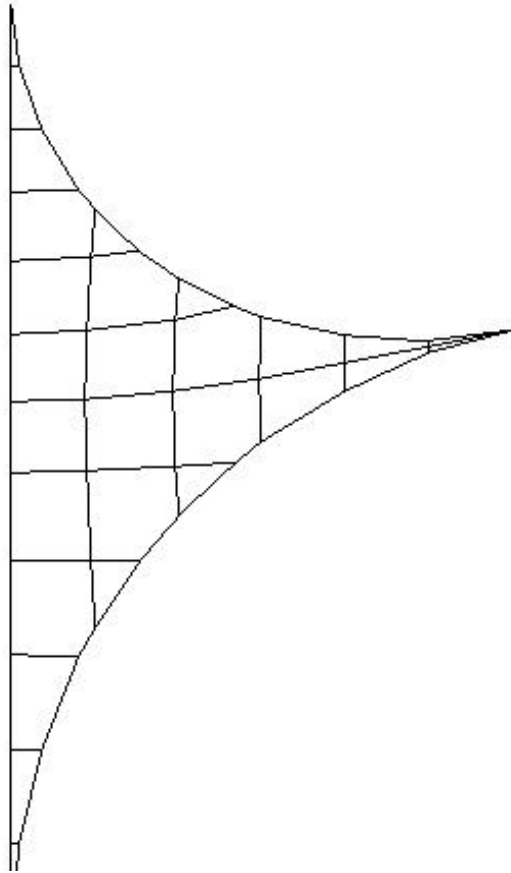


Figure 4.4
Geodesic Shape

The routine REFL.FOR is then used to generate the full structure of Figure 4.1 from the model of 1/2 of it. See again Figure 4.1.

The patterning program ALL.FOR is discussed in the next section.

Program Horst.for

```

C   horst.for
C   HORST BERGER'S LAYOUT PROGRAM
C   INCLUDE 'FGRAPH.FI'
C   DIMENSION R(900),NP(500),MI(500),for(500),MARK(500)
1  ,NNEW(500),ITYPE(500),NTYPE(500),NNP(500),NMI(500)
2  ,RNEW(900),FORN(300)
C   PI=3.14159
C   SMALL=.01
C
C   SET UP GRID
C   write(*,*)'THIS IS A CABLENET LAYOUT PROGRAM','/',
c   'DO YOU WISH TO RUN THE DEMO? (yes=1,no=0)'
C   read(*,*)nyes

```

```

        if(nyes.eq.1) go to 853
        write(*,*) 'ENTER GRID PARAMETERS: NX,DISTX,NY,DISTY'
        READ(*,*)NX,DISTX,NY,DISTY
        GO TO 854
853 read (50,150)NX,DISTX,NY,DISTY
854 CONTINUE
150 FORMAT(2(I5,F10.2))
    DX=DISTX/FLOAT(NX)
    DY=DISTY/FLOAT(NY)
    NN=(NY+1)*(NX+1)
    NN3=3*NN
    YCOORD=-DY
    DO 300 I=1,NN
300 MARK(I)=0
    DO 1 I=1,NN3
    1 R(I)=0.
    NX1=NX+1
    NY1=NY+1
    JT=0
    NB=0
    DO 2 J=1,NY1
    YCOORD=YCOORD+DY
    XCOORD=-DX
    DO 2 I=1,NX1
    JT=JT+1
    XCOORD=XCOORD+DX
    R(3*JT-2)=XCOORD
    R(3*JT-1)=YCOORD
    IF(I.EQ.NX1) GO TO 3
    NB=NB+1
    NP(NB)=JT+1
    MI(NB)=JT
    ITYPE(NB)=1
    FOR(NB)=1.
    3 IF(J.EQ.NY1) GO TO 2
    NB=NB+1
    NP(NB)=JT+NX1
    MI(NB)=JT
    ITYPE(NB)=2
    FOR(NB)=1.
    2 CONTINUE
    write(60,156)(I,R(3*I-2),R(3*I-1),R(3*I)
    1 ,I=1,NN)
156 FORMAT(I5,3f10.3)
c    CALL SPLOT(NP,MI,NN,NB,R,for,0,MARK)
    CALL SPLOT(NP,MI,NN,NB,R,for,2,MARK)
C
C    EDGE CABLES
    write(*,*) 'DO YOU WISH TO RUN THE DEMO? (yes=1,no=0)'
    read(*,*)nyes
    if(nyes.eq.1) go to 1999
    write(*,*) 'ENTER EDGE CABLE PARAMETERS: IP,IM,SAG'
2999 READ(*,*)IP,IM,SAG
    GO TO 1854
1999 READ(50,200)IP,IM,SAG
1854 CONTINUE
    write(*,200)IP,IM,SAG

```

```

200 FORMAT(2I5,F10.2)
   IF(IP.EQ.0)GO TO 4
   XIP=R(3*IP-2)
   YIP=R(3*IP-1)
   XIM=R(3*IM-2)
   YIM=R(3*IM-1)
   DELTAX=XIP-XIM
   DELTAY=YIP-YIM
   ANGL=ATAN2(DELTAY,DELTAX)
   CHORD=SQRT(DELTAX**2+DELTAY**2)
   THETA=ATAN2(SAG,CHORD/2)*2.
   RAD=SAG/(1.-COS(THETA))
   X0=CHORD*.5*COS(ANGL)-(RAD-SAG)*SIN(ANGL)+XIM
   Y0=CHORD*.5*SIN(ANGL)+(RAD-SAG)*COS(ANGL)+YIM
   write(*,*)ANGL,THETA,RAD,X0,Y0
   PX=1.
   PY=1.
   CABLE=RAD/DX
   CALL AMARK(X0,Y0,RAD,MARK,R,NN,SMALL)
   UP=SIGN(1.,SIN(ANGL-THETA))
   IUP=UP
   OVER=SIGN(1.,COS(ANGL-THETA))
   IOVER=OVER
   write(*,*)up,iup,over,iover
   IPP=IM
   BOT=YIM
   TOP=BOT+UP*DY
   BACK=XIM
   SIDE=XIM+OVER*DX
   ISIDE=0
   BOXD=SQRT(DX**2+DY**2)
   write(*,*)TOP,BOT,SIDE,BACK
999  IF(IPP.EQ.IM) GO TO 399
      IF(SQRT((R(3*NN-2)-XIP)**2+(R(3*NN-1)-YIP)**2).LE.
1    BOXD) GO TO 6
C
C      TOP
399  WRITE(*,*) 'TOP'
      IF(ISIDE.EQ.1) GO TO 9
      CALL ROOTS(RAD,X0,Y0,TOP,R1,R2,KS)
      IF(KS.NE.0) GO TO 9
      IF(R1.GT.AMAX1(SIDE,BACK).OR.R1.LT.AMIN1(SIDE,BACK))
1    GO TO 5
8    ISIDE=2
      NN=NN+1
      R(3*NN-2)=R1
      R(3*NN-1)=TOP
      MARK(NN)=0
77   NB=NB+1
      ITYPE(NB)=3
      FOR(NB)=CABLE
      MI(NB)=NN-1
      IF(IPP.EQ.IM)MI(NB)=IM
      IPP=IPP+(NX+1)*IUP
      NP(NB)=NN
      NB=NB+1
      ITYPE(NB)=1

```

```

FOR(NB)=1.
NP(NB)=NN
MI(NB)=IPP+IOVER
IF(MARK(IPP+IOVER).NE.0) MI(NB)=IPP
TOP=TOP+UP*DY
BOT=BOT+UP*DY
GO TO 999
5 IF(R2.GT.AMAX1(SIDE,BACK).OR.R2.LT.AMIN1(SIDE,BACK))
1 GO TO 9
R1=R2
GO TO 8
C
C BOTTOM
9 IF(ISIDE.EQ.2)GO TO 91
IF(IPP.EQ.IM) GO TO 91
WRITE(*,*) 'BOTTOM'
CALL ROOTS(RAD,X0,Y0,BOT,R1,R2,KS)
IF(KS.NE.0) GO TO 91
IF(R1.GT.AMAX1(SIDE,BACK).OR.R1.LT.AMIN1(SIDE,BACK))
1 GO TO 51
81 ISIDE=1
NN=NN+1
R(3*NN-2)=R1
R(3*NN-1)=BOT
MARK(NN)=0
177 NB=NB+1
ITYPE(NB)=3
FOR(NB)=CABLE
MI(NB)=NN-1
IF(IPP.EQ.IM) MI(NB)=IM
IPP=IPP-(NX+1)*IUP
NP(NB)=NN
NB=NB+1
ITYPE(NB)=1
FOR(NB)=1.
NP(NB)=NN
MI(NB)=IPP+IOVER+IUP*(NX+1)
IF(MARK(IPP+IOVER+IUP*(NX+1)).NE.0)MI(NB)=IPP+IUP*(NX+1)
TOP=TOP-DY*UP
BOT=BOT-DY*UP
GO TO 999
51 IF(R2.GT.AMAX1(SIDE,BACK).OR.R2.LT.AMIN1(SIDE,BACK))
1 GO TO 91
R1=R2
GO TO 81
C
C SIDE
91 IF(ISIDE.EQ.3) GO TO 727
CALL ROOTS(RAD,Y0,X0,SIDE,R1,R2,KS)
WRITE(*,*) 'SIDE'
IF(KS.NE.0) GO TO 727
IF(R1.GT.AMAX1(TOP,BOT).OR.R1.LT.AMIN1(TOP,BOT))
1 GO TO 52
82 ISIDE=4
NN=NN+1
R(3*NN-2)=SIDE
R(3*NN-1)=R1

```

```

MARK(NN)=0
277 NB=NB+1
    ITYPE(NB)=3
    FOR(NB)=CABLE
    MI(NB)=NN-1
    IF(IPP.EQ.IM) MI(NB)=IM
    IPP=IPP+IOVER
    NP(NB)=NN
    NB=NB+1
    ITYPE(NB)=2
    FOR(NB)=1.
    NP(NB)=NN
    MI(NB)=IPP
    IF(MARK(IPP).NE.0)MI(NB)=IPP+UP*(NX+1)
    SIDE=SIDE+DX*OVER
    BACK=BACK+DX*OVER
    GO TO 999
52  IF(R2.GT.AMAX1(TOP,BOT).OR.R2.LT.AMIN1(TOP,BOT))
    1 GO TO 727
    R1=R2
    GO TO 82
C
C    BACK
727 IF(ISIDE.EQ.4) GO TO 72
    IF(IPP.EQ.IM) GO TO 72
    CALL ROOTS(RAD,Y0,X0,BACK,R1,R2,KS)
    WRITE(*,*) 'BACK'
    IF(KS.NE.0) GO TO 72
    IF(R1.GT.AMAX1(TOP,BOT).OR.R1.LT.AMIN1(TOP,BOT))
    1 GO TO 527
827 NN=NN+1
    R(3*NN-2)=BACK
    R(3*NN-1)=R1
    MARK(NN)=0
    ISIDE=3
    NB=NB+1
    ITYPE(NB)=3
    FOR(NB)=CABLE
    MI(NB)=NN-1
    IF(IPP.EQ.IM) MI(NB)=IM
    IPP=IPP-IOVER
    NP(NB)=NN
    NB=NB+1
    ITYPE(NB)=2
    FOR(NB)=1.
    NP(NB)=NN
    MI(NB)=IPP+IOVER
    IF(MARK(IPP+IOVER).NE.0) MI(NB)=IPP+IOVER+IUP*(NX+1)
    SIDE=SIDE-DX*OVER
    BACK=BACK-DX*OVER
    GO TO 999
527 IF(R2.GT.AMAX1(TOP,BOT).OR.R2.LT.AMIN1(TOP,BOT))
    1 GO TO 72
    R1=R2
    GO TO 827
72  write(*,*) 'NO WAY OUT'
    GO TO 999

```

```

6 CONTINUE
  NB=NB+1
  ITYPE(NB)=3
  FOR(NB)=CABLE
  NP(NB)=IP
  MI(NB)=NN
  IF(NYES.EQ.1)GO TO 1999
  GO TO 2999
4 CONTINUE
c CALL SPLOT(NP,MI,NN,NB,R,for,0,MARK)
  CALL SPLOT(NP,MI,NN,NB,R,for,2,MARK)
  WRITE(60,*)NB,NN
C
C RENUMBER
  NNNEW=0
  DO 1200 I=1,NN
  IF(MARK(I).NE.0) GO TO 1200
  NNNEW=NNNEW+1
  NNEW(I)=NNNEW
  RNEW(3*NNNEW-2)=R(3*I-2)
  RNEW(3*NNNEW-1)=R(3*I-1)
  RNEW(3*NNNEW)=R(3*I)
1200 CONTINUE
202 FORMAT(I5,3E20.8)
  NNB=0
  DO 201 I=1,NB
  IF(MARK(NP(I)).NE.0.OR.MARK(MI(I)).NE.0) GO TO 201
  NNB=NNB+1
  NNP(NNB)=NNEW(NP(I))
  NMI(NNB)=NNEW(MI(I))
  NTYPE(NNB)=ITYPE(I)
  FORN(NNB)=FOR(I)
201 CONTINUE
  DO 205 I=1,NNNEW
205 MARK(I)=0
203 FORMAT(4I5)
  WRITE(70,203) NNB,NNNEW
  WRITE(70,202)(I,RNEW(3*I-2),RNEW(3*I-1),RNEW(3*I),I=1,NNNEW)
  WRITE(70,204)(I,NNP(I),NMI(I),NTYPE(I),FORN(I),I=1,NNB)
  CALL SPLOT(NNP,NMI,NNNEW,NNB,RNEW,for,0,MARK)
  CALL SPLOT(NNP,NMI,NNNEW,NNB,RNEW,for,2,MARK)
204 FORMAT(4I5,E20.8)
  STOP
  END
C
  SUBROUTINE AMARK(X0,Y0,RAD,MARK,R,NN,SMALL)
  DIMENSION R(1),MARK(1)
  DO 1 I=1,NN
  X=R(3*I-2)-X0
  Y=R(3*I-1)-Y0
  IF(X**2+Y**2.GE.(RAD-2.*SMALL)**2) GO TO 1
  MARK(I)=1
1 CONTINUE
  RETURN
  END
C
  SUBROUTINE ROOTS(RAD,X0,Y0,Y,R1,R2,KS)

```

```

KS=0
DISC=(RAD**2-(Y-Y0)**2)
IF(DISC.LE.0.) GO TO 1
DISC=SQRT(DISC)
R1=X0+DISC
R2=X0-DISC
write(*,*)R1,R2
RETURN
1 KS=1
RETURN
END

C
SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite,MARK)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1), MI(1), X(1), Y(1),for(1),MARK(1)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
character*5 text
character*8 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH='\F32\lib\modern.fon'
LIST="t'modern' '// 'h8w8b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
AMAXX=639-20
AMAYY=479-20

c find extent of picture window
XMIN=X(1)
XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
DO 2 I=1,NN
XI=X(I)
YI=Y(I)
IF(XMIN.GT.XI) XMIN=XI
IF(XMAX.LT.XI) XMAX=XI
IF(YMIN.GT.YI) YMIN=YI
2 IF(YMAX.LT.YI) YMAX=YI

c scale to center of window
SCALE = AMAX1((XMAX-XMIN)/AMAXX,(YMAX-YMIN)/AMAYY)
XSHIFT = (XMAX+XMIN)/2.0 - 639/2*SCALE
YSHIFT = (YMAX+YMIN)/2.0 - 479/2*SCALE

c move and draw for each line
DO 3 I=1,NB
K=NP(I)
M=MI(I)
XK=(X(K)-XSHIFT)/SCALE
YK=(Y(K)-YSHIFT)/SCALE
XM=(X(M)-XSHIFT)/SCALE
YM=(Y(M)-YSHIFT)/SCALE

c invert picture
YK = 479-YK
YM = 479-YM
LX=((XK+XM)/2)
LY=((YK+YM)/2)

```

```

IF(MARK(K).NE.0.OR.MARK(M).NE.0) GO TO 3
CALL MOVETO ( XK, YK, XY)
DUMMY = LINETO ( XM, YM)
IF(NB.GT.0) GO TO 3
if(iwrite.ne.2) go to 998
call moveto(lx,ly,xy)
write(text, '(i3)') i
call outgtext (text)
998 if(iwrite.eq.0.or.iwrite.eq.2) go to 3
call moveto(lx,ly,xy)
write(text1,'(f7.0)') for(i)
call outgtext (text1)
3 CONTINUE
if(iwrite.ne.2) go to 996
do 997 i=1,nn
lx=(x(i)-xshift)/scale
yk=(y(i)-yshift)/scale
ly=(479-yk)
call moveto(lx,ly,xy)
write(text, '(i3)') i
call outgtext (text)
997 continue
996 continue
RETURN
END

SUBROUTINE SPLOT ( NP,NM,NN,NB,R,for,iwrite,MARK)
INCLUDE 'FGRAPH.FD'
c iwrite = 0 no text
c 1 writes member forces
c 2 writes node map
DIMENSION NP(1),NM(1),RXY(1000),ROT(3,3),for(1)
DIMENSION ANGL(3),NT(3),A(3,3),R1(3,3,3)
INTEGER*2 DUMMY
DIMENSION R(1),X(900),Y(900),RZ(1000),MARK(1)
WRITE(*,1)
1 FORMAT(' YOU ARE ABOUT TO ENTER A GRAPHICS '
1 'DISPLAY MODE'/' THE KEYBOARD COMMANDS ARE'//
1 ' +1...POSITIVE ROTATION ABOUT X AXIS'//
1 ' -1...NEGATIVE ROTATION ABOUT X AXIS'//
1 ' +2...POSITIVE ROTATION ABOUT Y AXIS'//
1 ' -2...NEGATIVE ROTATION ABOUT Y AXIS'//
1 ' +3...POSITIVE ROTATION ABOUT Z AXIS'//
1 ' -3...NEGATIVE ROTATION ABOUT Z AXIS'//
1 ' 0...EXIT')
c delay for reading
READ(*,*)
DO 616 I=1,3
DO 617 J=1,3
DO 617 K=1,3
617 R1(I,J,K)=0.
616 R1(I,I,I)=1.
THX=0.
THY=00.
THZ=00.
c rotate using 10 deg increments
DTH=10.

```

```

70  PI=3.14159
    DO 604 I=1,3
      DO 603 J=1,3
603  ROT(J,I)=0.
604  ROT(I,I)=1.
      ANGL(1)=THX
      ANGL(2)=THY
      ANGL(3)=THZ
      NT(1)=1
      NT(2)=2
      NT(3)=3
      I=0
302  I=I+1
      IF(ANGL(I))606,605,606
606  L=NT(I)
      GO TO 612
618  DO 607 J=1,3
      DO 607 JA=1,3
      A(J,JA)=0.
      DO 607 JB=1,3
607  A(J,JA)=A(J,JA)+R1(L,J,JB)*ROT(JB,JA)
      DO 608 K=1,3
      DO 608 J=1,3
608  ROT(K,J)=A(K,J)
605  IF(I-3) 302,303,303
303  DO 805 I=1,NN
      RZ(I)=0.
      DO 806 K=1,3
806  RZ(I)=RZ(I)+ROT(3,K)*R(3*I-3+K)
      DO 805 J=1,2
      RXY(2*I-2+J)=0.
      DO 805 K=1,3
805  RXY(2*I-2+J)=RXY(2*I-2+J)+ROT(J,K)*R(3*I-3+K)
      GO TO 59
612  ANG=ANGL(I)*PI/180.
      IF(L-2)613,614,615
613  R1(1,2,2)=COS(ANG)
      R1(1,2,3)=SIN(ANG)
      R1(1,3,3)=R1(1,2,2)
      R1(1,3,2)=-R1(1,2,3)
      GO TO 618
614  R1(2,1,1)=COS(ANG)
      R1(2,1,3)=-SIN(ANG)
      R1(2,3,1)=-R1(2,1,3)
      R1(2,3,3)=R1(2,1,1)
      GO TO 618
615  R1(3,1,1)=COS(ANG)
      R1(3,1,2)=SIN(ANG)
      R1(3,2,1)=-R1(3,1,2)
      R1(3,2,2)=R1(3,1,1)
      GO TO 618
59  DO 24 I=1,NN
      X(I)=RXY(2*I-1)
24  Y(I)=RXY(2*I)
      CALL PLOT(NB,NN,X,Y,NP,NM,for,iwrite,MARK)
      READ(*,*) IVAL
      IF(IVAL.EQ.+1) GO TO 2000

```

```

        IF(IVAL.EQ.-1) GO TO 3000
        IF(IVAL.EQ. 2) GO TO 4000
        IF(IVAL.EQ.-2) GO TO 5000
        IF(IVAL.EQ. 3) GO TO 6000
        IF(IVAL.EQ.-3) GO TO 7000
        IF(IVAL.EQ. 0) GO TO 8000
2000 THX=THX+DTH
      GO TO 70
3000 THX=THX-DTH
      GO TO 70
4000 THY=THY+DTH
      GO TO 70
5000 THY=THY-DTH
      GO TO 70
6000 THZ=THZ+DTH
      GO TO 70
7000 THZ=THZ-DTH
      GO TO 70
8000 CALL UNREGISTERFONTS()
      DUMMY = SETVIDEOMODE( $DEFAULTMODE )
      RETURN
      END

```

Program tr3dom.for

```

c      TR3DOM.FOR
      INCLUDE 'FGRAPH.FI'
c      modified for shape finding
C      COMPUTES IN-PLANE FORCES
C      SPACE TRUSS
C      NONLINEAR VERSION
      DOUBLE PRECISION r(200),P(200),C(150,150),UVEC(3),C1,
1 F1,F2,FAC,C2,D1
      DIMENSION NP(200),NM(200),S(200),PSAVE(200),FSAVE(200)
1 ,ncons(200),r1(200),fi(200),ITYPE(100),nfix(10),ht(10)
1 ,YOUNG(3)
      common c
      MAXC=150

C
C      INITIALIZE PARAMETERS/ARRAYS
C
C      TYPE 3 MEMBER IS CABLE
C      PROGRAM ONLY DOES CASE NSTEP=1
      E = 30.0D6
      YOUNG(1)=0.
      YOUNG(2)=0.
      YOUNG(3)=E
      FAC=1.
      READ(70,2) NB,NN,NS,NSTEP,NIT
      nit=6
      IF(NSTEP.EQ.0) NSTEP=1
      IF(NIT.EQ.0) NIT=1
2 FORMAT (7I5)
      WRITE(60,1)NB,NN,NS,NSTEP,NIT
      READ(70,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K-1)

```

```

1 ,NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
  READ(70,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
  WRITE(60,160)(L,NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
  DO 3013 I=1,NB
3013 FI(I)=FSAVE(I)
151 FORMAT (5X,3I5,E20.8)
160 FORMAT (4I5,E20.8)
1 FORMAT(I5,' NO. MEMBERS'/I5,' NO. NODES'/I5,' NO. SUPPORTS'/
1I5,' NO. LOAD STEPS'/I5,' NO. ITERATIONS'//)
  NNS=NN-NS
  N=3*NNS
  WRITE(60,157)(K,R(3*K-2),R(3*K-1),r(3*k),
1 PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
157 FORMAT (1H1,25X,11HCOORDINATES,40X,5HLOADS//
114X,1HX,19X,1HY,19X,1HZ,18X,2HPX,18X,2HPY,18X,2HPZ//
1 (I4,3D17.8,3D17.8))
156 FORMAT (5X,3E20.8,3i2,3f3.0)
C
C   DEFINE FIXED NODES (CONSTRAINTS)
  NF=0
7778 WRITE(*,*)'FIXED NODES..NODE NUMBER,ELEV'
  READ(*,*) NOD,ELEV
  IF(NOD.EQ.0) GO TO 7777
  NF=NF+1
  NFIX(NF)=NOD
  HT(NF)=ELEV
  GO TO 7778
7777 WRITE(*,*)'Y-AXIS SYMMETRY?'
C   BRANCH ON ZERO FOR NO SYMMETRY
  READ(*,*) ISYM
  NSTEP=0
5 NSTEP=NSTEP+1
  IF(NSTEP.EQ.4)STOP
  do 341 i=1,n
341 NCONS(I)=0
  IF(NF.EQ.0) GO TO 5777
  DO 5778 I=1,NF
  NOD=NFIX(I)
  NCONS(3*NOD)=1
  NCONS(3*NOD-2)=1
  NCONS(3*NOD-1)=1
  R(3*NOD)=HT(I)
C   FIX NODEE OUT OF PLANE
5778 IF(NSTEP.EQ.1) R(3*NOD)=0.
  IF(NSTEP.EQ.2) CALL PRESTR(FSAVE,NP,NM,NB,R,E,ITYPE)
5777 IF(ISYM.EQ.0) GO TO 7775
  DO 7774 I=1,NN
  IF(R(3*I-2).NE.0.)GO TO 7774
  NCONS(3*I-2)=1
7774 CONTINUE
  IF(NSTEP.NE.1)GO TO 7775
C   ON Y-AXIS
  DO 7773 I=1,NB
  IF(R(3*NP(I)-2).NE.0..OR.R(3*NM(I)-2).NE.0.)GO TO 7773
  FSAVE(I)=.5*FSAVE(I)
7773 CONTINUE
7775 CONTINUE

```

```

        WRITE(60,159)
159  FORMAT (1H1,3X,6HMEMBER,5X,5H+  END,5X,5H-  END,16X,4HAREA,
        1 11X,9HPRESTRESS//)
        do 7769 i=1,n
7769  r1(i)=r(i)
        call splot(np,nm,nn,nb,r1,fi,0)
        GO TO (7703,7704,7705),NSTEP
7704  DO 7749 I=1,NN
        NCONS(3*I-2)=1
7749  NCONS(3*I-1)=1
        do 7725 i=1,nb
7725  FI(I)=FSAVE(I)
        GO TO 7706
7703  DO 7707 I=1,NN
7707  NCONS(3*I)=1
        GO TO 7706
7705  DO 7713 I=1,NB
7713  IF(ITYPE(I).NE.3)FSAVE(I)=1.
7706  CONTINUE
C
C      START LOADSTEPS AND ITERATIONS
        DO 997 LDSTP=1,NSTEP
        STEP=FLOAT(LDSTP+1)/FLOAT(NSTEP)
        IF(LDSTP.EQ.NSTEP) STEP=1.
        DO 997 ITER=1,NIT
        WRITE(60,897) ITER,LDSTP
897  FORMAT(///' ****ITERATION NUMBER',I4/
        1      '          LOAD STEP          ',I4)
C
C      SET UP SYSTEM MATRIX
C
        DO 904 I=1,N
        P(I)=PSAVE(I)*STEP
        DO 904 J=1,N
904  C(I,J)=0.
        DO 999 L=1,NB
        write(*,*) iter,l
        K = 3*NP(L)
        M = 3*NM(L)
        CALL UNITV(K,M,C1,UVEC,R)
        IF(K.GT.N) GO TO 888
        P(K-2)=P(K-2)-FSAVE(L)*UVEC(1)
        P(K-1)=P(K-1)-FSAVE(L)*UVEC(2)
        P(K )=P(K )-FSAVE(L)*UVEC(3)
888  IF(M.GT.N) GO TO 887
        P(M-2)=P(M-2)+FSAVE(L)*UVEC(1)
        P(M-1)=P(M-1)+FSAVE(L)*UVEC(2)
        P(M )=P(M )+FSAVE(L)*UVEC(3)
887  continue
        E1=YOUNG(ITYPE(L))
        S(L)=1.
        call sert(K,M,UVEC,MAXC,N,E1,S(1),C1,FSAVE(1),itype(1))
999  CONTINUE
C
C      ERROR AT START OF ITERATION
        C1=0.
        DO 500 I=1,N

```

```

        if(ncons(i).ne.0)go to 500
        C1=C1+P(I)**2
500 continue
        C1=DSQRT(C1)
        WRITE(60,501) C1
501 FORMAT(//' ERROR = ',D20.8//)
        do 300 i=1,n
        if( ncons(i).eq.0)go to 300
        do 301 j=1,n
        c(i,j)=0.
301 c(j,i)=0.
        p(i)=0.
        c(i,i)=1.
300 continue
C
C      SOLVE FOR DISPLACEMENTS
C
c      write(*,9876)((i,j,c(i,j),i=1,n),j=1,n)
9876 format(3(2i5,e20.8))
927 M=N-1
        DO 91 I=1,M
        if(c(i,i).eq.0.) write(*,*) I
        L=I+1
        DO 91 J=L,N
        IF (C(J,I)) 93,91,93
93 DO 92 K=L,N
92 C(J,K)=C(J,K)-C(I,K)*C(J,I)/C(I,I)
        P(J)=P(J)-P(I) *C(J,I)/C(I,I)
91 CONTINUE
        P (N)=P(N)/C(N,N)
        IF(C(N,N).LE.0.) WRITE(60,298) N
298 FORMAT(///'***NEG TERM ON THE DIAGONAL AT ROW',I5///)
        DO 94 I=1,M
        K=N-I
        L=K+1
        DO 95 J=L,N
95 P(K)=P(K)-P (J)*C(K,J)
        IF(C(K,K).LE.0.) WRITE(6,298) K
94 P (K)=P(K)/C(K,K)
        WRITE(60,161)(I,P(3*I-2),P(3*I-1),P(3*I),I=1,NNS)
161 FORMAT (1H1,13HDISPLACEMENTS/20X,1HX,19X,1HY,19X,1HZ//
1 (I10,3D20.8))
        WRITE(60,162)
162 FORMAT (1H1,3X,6HMEMBER,9X,2HDL,17X,5HFORCE,14X,6HSTRESS,
1 7X,'UPDATED MEMBER FORCES'//)
C
C      COMPUTE MEMBER FORCES AND DISPLACEMENTS
C
c      do 5678 i=1,n
c 5678 p(i)=p(i)/2.
        DO 998 I=1,NB
        K = 3*NP(I)
        M = 3*NM(I)
        CALL UNITV(K,M,C1,UVEC,R)
        K1=K
        D1=0.
        FAC=1.

```

```

DO 297 J=1,2
IF(K1.GT.N) GO TO 996
D1=D1+FAC*(P(K1-2)*UVEC(1)+P(K1-1)*UVEC(2)+P(K1)*UVEC(3))
996 FAC=-1.
K1=M
297 CONTINUE
s(i)=1.
e1=YOUNG(ITYPE(I))
F1=D1*E1*S(I)/C1
F2=F1/S(I)
1000 FORMAT (I10,4D20.8)
UVEC(1)=R(K-2)-R(M-2)
UVEC(2)=R(K-1)-R(M-1)
UVEC(3)=R(K )-R(M )
IF(K.GT.N) GO TO 666
UVEC(1)=UVEC(1)+P(K-2)
UVEC(2)=UVEC(2)+P(K-1)
UVEC(3)=UVEC(3)+P(K )
666 IF(M.GT.N) GO TO 665
UVEC(1)=UVEC(1)-P(M-2)
UVEC(2)=UVEC(2)-P(M-1)
UVEC(3)=UVEC(3)-P(M )
665 C2=DSQRT(UVEC(1)**2+UVEC(2)**2+UVEC(3)**2)
C2=C2-C1
Fsave(i)=Fsave(i)+C2*S(I)*E1/C1
WRITE(60,1000) I,D1,F1,F2,fsave(i)
998 CONTINUE
1998 continue
C
C UNDATE COORDINATES
DO 444 I=1,N
R(I)=R(I)+P(I)
444 r1(i)=r(i)
C call splot(np,nm,nn,nb,r1,fi,0)
997 CONTINUE
call splot(np,nm,nn,nb,r1,fi,0)
WRITE(600,2)NB,NN,NS,NSTEP,NIT,nf,ism
WRITE(600,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K
1 -1),NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
WRITE(600,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
write(600,654)(nfix(i),ht(i),i=1,nf)
654 format(i5,f10.3)
IF(NSTEP.EQ.1)STOP
GO TO 5
299 STOP
END
C
SUBROUTINE UNITV(K,M,C1,UVEC,R)
DOUBLE PRECISION R(1),C1,UVEC(3)
C1=0.
DO 1 I=1,3
UVEC(I)=R(K+I-3)-R(M+I-3)
1 C1=C1+UVEC(I)**2
C1=DSQRT(C1)
DO 2 I=1,3
2 UVEC(I)=UVEC(I)/C1
RETURN

```

END

C

```
SUBROUTINE PRESTR(FSAVE,NP,NM,NB,R,E,ITYPE)
DIMENSION NP(1),NM(1),FSAVE(1),ITYPE(1)
DOUBLE PRECISION R(1)
DO 1 I=1,NB
IP=3*NP(I)
IM=3*NM(I)
IF(R(IP).EQ.0..AND.R(IM).EQ.0.) GO TO 1
E1=E
IF(ITYPE(I).NE.3) E1=0.
DX=R(IP-2)-R(IM-2)
DY=R(IP-1)-R(IM-1)
DZ=R(IP)-R(IM)
DS=SQRT(DX**2+DY**2)
AL=SQRT(DS**2+DZ**2)
FSAVE(I)=FSAVE(I)+(AL-DS)*E1/DS
1 CONTINUE
RETURN
END
```

C

```
SUBROUTINE sert(K,M,UVEC,MAXC,N,E,S,C1,FSAVE,itype)
DOUBLE PRECISION C(150,150),UVEC(3),C1
common c
K1=K
DO 1 I=1,2
IF(K1.GT.N) GO TO 1
M1=K
DO 2 J=1,2
IF(M1.GT.N) GO TO 2
FAC=1.
IF(I.NE.J) FAC=-1.
DO 3 L=1,3
I1=K1-3+L
DO 3 L1=1,3
J1=M1-3+L1
C(I1,J1)=C(I1,J1)+UVEC(L)*UVEC(L1)*(S*E-FSAVE)*FAC/C1
3 IF(L.EQ.L1) C(I1,J1)=C(I1,J1)+FAC*FSAVE/C1
2 M1=M
1 K1=M
RETURN
END
```

C

```
SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1), MI(1), X(1), Y(1),for(1)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
character*6 text
character*10 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH='\newfor\lib\modern.fon'
LIST="t'modern' '// 'h8w8b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
```

```

    AMAXX=639-20
    AMAYY=479-20
c   find extent of picture window
    XMIN=X(1)
    XMAX=X(1)
    YMIN=Y(1)
    YMAX=Y(1)
    DO 2 I=1,NN
    XI=X(I)
    YI=Y(I)
    IF(XMIN.GT.XI) XMIN=XI
    IF(XMAX.LT.XI) XMAX=XI
    IF(YMIN.GT.YI) YMIN=YI
    2 IF(YMAX.LT.YI) YMAX=YI
c   scale to center of window
    SCALE = AMAX1((XMAX-XMIN)/AMAXX,(YMAX-YMIN)/AMAYY)
    XSHIFT = (XMAX+XMIN)/2.0 - 639/2*SCALE
    YSHIFT = (YMAX+YMIN)/2.0 - 479/2*SCALE
c   move and draw for each line
    DO 3 I=1,NB
    K=NP(I)
    M=MI(I)
    XK=(X(K)-XSHIFT)/SCALE
    YK=(Y(K)-YSHIFT)/SCALE
    XM=(X(M)-XSHIFT)/SCALE
    YM=(Y(M)-YSHIFT)/SCALE
c   invert picture
    YK = 479-YK
    YM = 479-YM
    LX=((XK+XM)/2)
    LY=((YK+YM)/2)
    CALL MOVETO ( XK, YK, XY)
    DUMMY = LINETO ( XM, YM)
    if(iwrite.ne.2) go to 998
    call moveto(lx,ly,xy)
    write(text, '(i3)') i
    call outgtext (text)
998 if(iwrite.eq.0.or.iwrite.eq.2) go to 3
    call moveto(lx,ly,xy)
    write(text1,'(f7.0)') for(i)
    call outgtext (text1)
    3 CONTINUE
    if(iwrite.ne.2) go to 996
    do 997 i=1,nn
    lx=(x(i)-xshift)/scale
    yk=(y(i)-yshift)/scale
    ly=(479-yk)
    call moveto(lx,ly,xy)
    write(text, '(i3)') i
    call outgtext (text)
997 continue
996 continue
    RETURN
    END

    SUBROUTINE SPLOT ( NP,NM,NN,NB,R,for,iwrite)
    INCLUDE 'FGRAPH.FD'

```

```

c      iwrite = 0    no text
c              1    writes member forces
c              2    writes node map
      DIMENSION NP(1),NM(1),RXY(1000),ROT(3,3),for(1)
      DIMENSION ANGL(3),NT(3),A(3,3),R1(3,3,3)
      INTEGER*2 DUMMY
      DIMENSION R(1),X(200),Y(200),RZ(1000)
      WRITE(*,1)
1  FORMAT(' YOU ARE ABOUT TO ENTER A GRAPHICS '
1  ' DISPLAY MODE'/' THE KEYBOARD COMMANDS ARE'//
1  '      +1...POSITIVE ROTATION ABOUT X AXIS'/
1  '      -1...NEGATIVE ROTATION ABOUT X AXIS'/
1  '      +2...POSITIVE ROTATION ABOUT Y AXIS'/
1  '      -2...NEGATIVE ROTATION ABOUT Y AXIS'/
1  '      +3...POSITIVE ROTATION ABOUT Z AXIS'/
1  '      -3...NEGATIVE ROTATION ABOUT Z AXIS'/
1  '      0...EXIT')
c      delay for reading
      READ(*,*)
      DO 616 I=1,3
      DO 617 J=1,3
      DO 617 K=1,3
617  R1(I,J,K)=0.
616  R1(I,I,I)=1.
      THX=0.
      THY=00.
      THZ=00.
c      rotate using 10 deg increments
      DTH=10.
70  PI=3.14159
      DO 604 I=1,3
      DO 603 J=1,3
603  ROT(J,I)=0.
604  ROT(I,I)=1.
      ANGL(1)=THX
      ANGL(2)=THY
      ANGL(3)=THZ
      NT(1)=1
      NT(2)=2
      NT(3)=3
      I=0
302  I=I+1
      IF(ANGL(I))606,605,606
606  L=NT(I)
      GO TO 612
618  DO 607 J=1,3
      DO 607 JA=1,3
      A(J,JA)=0.
      DO 607 JB=1,3
607  A(J,JA)=A(J,JA)+R1(L,J,JB)*ROT(JB,JA)
      DO 608 K=1,3
      DO 608 J=1,3
608  ROT(K,J)=A(K,J)
605  IF(I-3) 302,303,303
303  DO 805 I=1,NN
      RZ(I)=0.
      DO 806 K=1,3

```

```

806 RZ(I)=RZ(I)+ROT(3,K)*R(3*I-3+K)
DO 805 J=1,2
RXY(2*I-2+J)=0.
DO 805 K=1,3
805 RXY(2*I-2+J)=RXY(2*I-2+J)+ROT(J,K)*R(3*I-3+K)
GO TO 59
612 ANG=ANGL(I)*PI/180.
IF(L-2)613,614,615
613 R1(1,2,2)=COS(ANG)
R1(1,2,3)=SIN(ANG)
R1(1,3,3)=R1(1,2,2)
R1(1,3,2)=-R1(1,2,3)
GO TO 618
614 R1(2,1,1)=COS(ANG)
R1(2,1,3)=-SIN(ANG)
R1(2,3,1)=-R1(2,1,3)
R1(2,3,3)=R1(2,1,1)
GO TO 618
615 R1(3,1,1)=COS(ANG)
R1(3,1,2)=SIN(ANG)
R1(3,2,1)=-R1(3,1,2)
R1(3,2,2)=R1(3,1,1)
GO TO 618
59 DO 24 I=1,NN
X(I)=RXY(2*I-1)
24 Y(I)=RXY(2*I)
CALL PLOT(NB,NN,X,Y,NP,NM,for,iwrite)
READ(*,*) IVAL
IF(IVAL.EQ.+1) GO TO 2000
IF(IVAL.EQ.-1) GO TO 3000
IF(IVAL.EQ. 2) GO TO 4000
IF(IVAL.EQ.-2) GO TO 5000
IF(IVAL.EQ. 3) GO TO 6000
IF(IVAL.EQ.-3) GO TO 7000
IF(IVAL.EQ. 0) GO TO 8000
2000 THX=THX+DTH
GO TO 70
3000 THX=THX-DTH
GO TO 70
4000 THY=THY+DTH
GO TO 70
5000 THY=THY-DTH
GO TO 70
6000 THZ=THZ+DTH
GO TO 70
7000 THZ=THZ-DTH
GO TO 70
8000 CALL UNREGISTERFONTS( )
DUMMY = SETVIDEOMODE( $DEFAULTMODE )
RETURN
END

```

Program tr3d01.for

INCLUDE 'FGRAPH.FI'

```

c   modified for shape finding
C   SPACE TRUSS
C   NONLINEAR VERSION
    DOUBLE PRECISION r(200),P(200),C(150,150),UVEC(3),C1,
1   F1,F2,FAC,C2,D1
    DIMENSION NP(200),NM(200),S(200),PSAVE(200),FSAVE(100)
1   ,ncons(200),r1(200),fi(200),ITYPE(100),nfix(10),ht(10)
1   ,YOUNG(3)
    common c
    MAXC=150

C
C   INITIALIZE PARAMETERS/ARRAYS
C
    E = 0.
    YOUNG(1)=E/1000.
    YOUNG(2)=YOUNG(1)
    YOUNG(3)=E
    READ(70,2) NB,NN,NS,NSTEP,NIT,nf, isym
    nit=10
    IF(NSTEP.EQ.0) NSTEP=1
    IF(NIT.EQ.0) NIT=1
2   FORMAT (7I5)
    WRITE(60,1)NB,NN,NS,NSTEP,NIT,nf, isym
    READ(70,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K-1)
1   ,NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
    READ(70,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
    WRITE(60,160)(L,NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
    read(70,654)(nfix(i),ht(i),i=1,nf)
    write(70,654)(nfix(i),ht(i),i=1,nf)
654 format(i5,f10.3)
    DO 3013 I=1,NB
3013 FI(I)=FSAVE(I)
151  FORMAT (5X,3I5,E20.8)
160  FORMAT (4I5,E20.8)
1   FORMAT(I5,' NO. MEMBERS'/I5,' NO. NODES'/I5,' NO. SUPPORTS'/
1I5,' NO. LOAD STEPS'/I5,' NO. ITERATIONS'//)
    NNS=NN-NS
    N=3*NNS
    WRITE(60,157)(K,R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K
1   -1),NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
157  FORMAT (1H1,25X,11HCOORDINATES,40X,5HLOADS//
114X,1HX,19X,1HY,19X,1HZ,18X,2HPX,18X,2HPY,18X,2HPZ//
1   (I4,3D17.8,3I2,3D17.8))
156  FORMAT (5X,3E20.8,3i2,3f3.0)
    DO 341 I=1,N
341  NCONS(I)=0
    DO 7774 I=1,NN
    IF(R(3*I-2).NE.0.) GO TO 7774
    NCONS(3*I-2)=1
7774 CONTINUE
7777 CONTINUE
    IF(NF.EQ.0) GO TO 7773
    DO 5778 I=1,NF
    NOD=NFIX(I)
    NCONS(3*NOD)=1
    NCONS(3*NOD-2)=1
    NCONS(3*NOD-1)=1

```

```

      R(3*NOD)=HT(I)
5778 CONTINUE
      CALL PRESTR(FSAVE,NP,NM,NB,R,YOUNG,ITYPE)
7773 CONTINUE
7775 CONTINUE
      WRITE(60,159)
159  FORMAT (1H1,3X,6HMEMBER,5X,5H+ END,5X,5H- END,16X,4HAREA,
1 11X,9HPRESTRESS//)
      do 7769 i=1,n
7769 r1(i)=r(i)
      call splot(np,nm,nn,nb,r1,fi,0)
7704 DO 7749 I=1,NN
      NCONS(3*I-2)=1
7749 NCONS(3*I-1)=1
C
C      START LOADSTEPS AND ITERATIONS
DO 997 LDSTP=1,NSTEP
STEP=FLOAT(LDSTP+1)/FLOAT(NSTEP)
IF(LDSTP.EQ.NSTEP) STEP=1.
DO 997 ITER=1,NIT
WRITE(60,897) ITER,LDSTP
897 FORMAT(///' ****ITERATION NUMBER',I4/
1      '      LOAD STEP      ',I4)
C
C      SET UP SYSTEM MATRIX
C
DO 904 I=1,N
P(I)=PSAVE(I)*STEP
DO 904 J=1,N
904 C(I,J)=0.
DO 999 L=1,NB
s(l)=1.
write(*,*) iter,l
K = 3*NP(L)
M = 3*NM(L)
CALL UNITV(K,M,C1,UVEC,R)
IF(K.GT.N) GO TO 888
P(K-2)=P(K-2)-FSAVE(L)*UVEC(1)
P(K-1)=P(K-1)-FSAVE(L)*UVEC(2)
P(K )=P(K )-FSAVE(L)*UVEC(3)
888 IF(M.GT.N) GO TO 887
P(M-2)=P(M-2)+FSAVE(L)*UVEC(1)
P(M-1)=P(M-1)+FSAVE(L)*UVEC(2)
P(M )=P(M )+FSAVE(L)*UVEC(3)
887 continue
E1=YOUNG(ITYPE(L))
call sert(K,M,UVEC,MAXC,N,E1,S(1),C1,FSAVE(1))
999 CONTINUE
C
C      ERROR AT START OF ITERATION
C1=0.
DO 500 I=1,N
if(ncons(i).ne.0)go to 500
C1=C1+P(I)**2
500 continue
C1=DSQRT(C1)
WRITE(60,501) C1

```

```

501 FORMAT(///' ERROR = ',D20.8//)
do 300 i=1,n
  if( ncons(i).eq.0)go to 300
  do 301 j=1,n
    c(i,j)=0.
301 c(j,i)=0.
  p(i)=0.
  c(i,i)=1.
300 continue
C
C   SOLVE FOR DISPLACEMENTS
C
c   write(*,9876)((i,j,c(i,j),i=1,n),j=1,n)
9876 format(3(2i5,e20.8))
927 M=N-1
DO 91 I=1,M
  if(c(i,i).eq.0.) write(*,*) I
  L=I+1
  DO 91 J=L,N
    IF (C(J,I)) 93,91,93
93 DO 92 K=L,N
92 C(J,K)=C(J,K)-C(I,K)*C(J,I)/C(I,I)
  P(J)=P(J)-P(I) *C(J,I)/C(I,I)
91 CONTINUE
  P (N)=P(N)/C(N,N)
  IF(C(N,N).LE.0.) WRITE(60,298) N
298 FORMAT(///'***NEG TERM ON THE DIAGONAL AT ROW',I5//)
DO 94 I=1,M
  K=N-I
  L=K+1
  DO 95 J=L,N
95 P(K)=P(K)-P (J)*C(K,J)
  IF(C(K,K).LE.0.) WRITE(6,298) K
94 P (K)=P(K)/C(K,K)
  WRITE(60,161)(I,P(3*I-2),P(3*I-1),P(3*I),I=1,NNS)
161 FORMAT (1H1,13HDISPLACEMENTS/20X,1HX,19X,1HY,19X,1HZ//
1 (I10,3D20.8))
  WRITE(60,162)
162 FORMAT (1H1,3X,6HMEMBER,9X,2HDL,17X,5HFORCE,14X,6HSTRESS,
1 7X,'UPDATED MEMBER FORCES'//)
C
C   COMPUTE MEMBER FORCES AND DISPLACEMENTS
C
DO 8888 I=1,N
8888 P(I)=P(I)/2.
DO 998 I=1,NB
  K = 3*NP(I)
  M = 3*NM(I)
  CALL UNITV(K,M,C1,UVEC,R)
  K1=K
  D1=0.
  FAC=1.
  DO 297 J=1,2
  IF(K1.GT.N) GO TO 996
  D1=D1+FAC*(P(K1-2)*UVEC(1)+P(K1-1)*UVEC(2)+P(K1)*UVEC(3))
996 FAC=-1.
  K1=M

```

```

297 CONTINUE
   e1=YOUNG(ITYPE(I))
   F1=D1*E1*S(I)/C1
   F2=F1/S(I)
1000 FORMAT (I10,4D20.8)
   UVEC(1)=R(K-2)-R(M-2)
   UVEC(2)=R(K-1)-R(M-1)
   UVEC(3)=R(K )-R(M )
   IF(K.GT.N) GO TO 666
   UVEC(1)=UVEC(1)+P(K-2)
   UVEC(2)=UVEC(2)+P(K-1)
   UVEC(3)=UVEC(3)+P(K )
666 IF(M.GT.N) GO TO 665
   UVEC(1)=UVEC(1)-P(M-2)
   UVEC(2)=UVEC(2)-P(M-1)
   UVEC(3)=UVEC(3)-P(M )
665 C2=DSQRT(UVEC(1)**2+UVEC(2)**2+UVEC(3)**2)
   C2=C2-C1
   Fsave(i)=Fsave(i)+C2*S(I)*E1/C1
   WRITE(60,1000) I,D1,F1,F2,fsave(i)
998 CONTINUE
1998 continue
C
C   UNDATE COORDINATES
   DO 444 I=1,N
   R(I)=R(I)+P(I)
444 r1(i)=r(i)
C   call splot(np,nm,nn,nb,r1,fi,0)
997 CONTINUE
   call splot(np,nm,nn,nb,r1,fi,0)
   WRITE(601,2)NB,NN,NS,NSTEP,NIT,nf,isymp
   WRITE(601,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K
1 -1),NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
   WRITE(601,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
   write(601,654)(nfix(i),ht(i),i=1,nf)
299 STOP
   END
C
   SUBROUTINE UNITV(K,M,C1,UVEC,R)
   DOUBLE PRECISION R(1),C1,UVEC(3)
   C1=0.
   DO 1 I=1,3
   UVEC(I)=R(K+I-3)-R(M+I-3)
1 C1=C1+UVEC(I)**2
   C1=DSQRT(C1)
   DO 2 I=1,3
2 UVEC(I)=UVEC(I)/C1
   RETURN
   END
C
   SUBROUTINE PRESTR(FSAVE,NP,NM,NB,R,YOUNG,ITYPE)
   DIMENSION NP(1),NM(1),FSAVE(1),ITYPE(1),YOUNG(1)
   DOUBLE PRECISION R(1)
   DO 1 I=1,NB
   E=YOUNG(ITYPE(I))
   IP=3*NP(I)
   IM=3*NM(I)

```

```

IF(R(IP).EQ.0..AND.R(IM).EQ.0.) GO TO 1
E1=E
IF(ITYPE(I).NE.3) E1=0.
DX=R(IP-2)-R(IM-2)
DY=R(IP-1)-R(IM-1)
DZ=R(IP)-R(IM)
DS=SQRT(DX**2+DY**2)
AL=SQRT(DS**2+DZ**2)
FSAVE(I)=FSAVE(I)+(AL-DS)*E/DS
1 CONTINUE
RETURN
END

```

C

```

SUBROUTINE sert(K,M,UVEC,MAXC,N,E,S,C1,FSAVE)
DOUBLE PRECISION C(150,150),UVEC(3),C1
COMMON C
K1=K
DO 1 I=1,2
IF(K1.GT.N) GO TO 1
M1=K
DO 2 J=1,2
IF(M1.GT.N) GO TO 2
FAC=1.
IF(I.NE.J) FAC=-1.
DO 3 L=1,3
I1=K1-3+L
DO 3 L1=1,3
J1=M1-3+L1
C(I1,J1)=C(I1,J1)+UVEC(L)*UVEC(L1)*(S*E-FSAVE)*FAC/C1
3 IF(L.EQ.L1) C(I1,J1)=C(I1,J1)+FAC*FSAVE/C1
2 M1=M
1 K1=M
RETURN
END

```

C

```

SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1), MI(1), X(1), Y(1),for(1)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
character*6 text
character*10 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH='\newfor\lib\modern.fon'
LIST="t'modern' '// 'h8w8b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
AMAXX=639-20
AMAYY=479-20

```

c

```

find extent of picture window
XMIN=X(1)
XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
DO 2 I=1,NN

```

```

        XI=X(I)
        YI=Y(I)
        IF(XMIN.GT.XI) XMIN=XI
        IF(XMAX.LT.XI) XMAX=XI
        IF(YMIN.GT.YI) YMIN=YI
2      IF(YMAX.LT.YI) YMAX=YI
c      scale to center of window
        SCALE = AMAX1((XMAX-XMIN)/AMAXX,(YMAX-YMIN)/AMAYY)
        XSHIFT = (XMAX+XMIN)/2.0 - 639/2*SCALE
        YSHIFT = (YMAX+YMIN)/2.0 - 479/2*SCALE
c      move and draw for each line
        DO 3 I=1,NB
        K=NP(I)
        M=MI(I)
        XK=(X(K)-XSHIFT)/SCALE
        YK=(Y(K)-YSHIFT)/SCALE
        XM=(X(M)-XSHIFT)/SCALE
        YM=(Y(M)-YSHIFT)/SCALE
c      invert picture
        YK = 479-YK
        YM = 479-YM
        LX=((XK+XM)/2)
        LY=((YK+YM)/2)
        CALL MOVETO ( XK, YK, XY)
        DUMMY = LINETO ( XM, YM)
        if(iwrite.ne.2) go to 998
        call moveto(lx,ly,xy)
        write(text, '(i3)') i
        call outgtext (text)
998 if(iwrite.eq.0.or.iwrite.eq.2) go to 3
        call moveto(lx,ly,xy)
        write(text1,'(f7.0)') for(i)
        call outgtext (text1)
3      CONTINUE
        if(iwrite.ne.2) go to 996
        do 997 i=1,nb
        lx=(x(i)-xshift)/scale
        yk=(y(i)-yshift)/scale
        ly=(479-yk)
        call moveto(lx,ly,xy)
        write(text, '(i3)') i
        call outgtext (text)
997 continue
996 continue
        RETURN
        END

SUBROUTINE SPLOT ( NP,NM,NN,NB,R,for,iwrite)
INCLUDE 'FGRAPH.FD'
c      iwrite = 0    no text
c              1    writes member forces
c              2    writes node map
DIMENSION NP(1),NM(1),RXY(1000),ROT(3,3),for(1)
DIMENSION ANGL(3),NT(3),A(3,3),R1(3,3,3)
INTEGER*2 DUMMY
DIMENSION R(1),X(200),Y(200),RZ(1000)
WRITE(*,1)

```

```

1 FORMAT(' YOU ARE ABOUT TO ENTER A GRAPHICS '
1 'DISPLAY MODE'/' THE KEYBOARD COMMANDS ARE'//
1 '      +1...POSITIVE ROTATION ABOUT X AXIS'/
1 '      -1...NEGATIVE ROTATION ABOUT X AXIS'/
1 '      +2...POSITIVE ROTATION ABOUT Y AXIS'/
1 '      -2...NEGATIVE ROTATION ABOUT Y AXIS'/
1 '      +3...POSITIVE ROTATION ABOUT Z AXIS'/
1 '      -3...NEGATIVE ROTATION ABOUT Z AXIS'/
1 '      0...EXIT')
c   delay for reading
    READ(*,*)
    DO 616 I=1,3
    DO 617 J=1,3
    DO 617 K=1,3
617 R1(I,J,K)=0.
616 R1(I,I,I)=1.
    THX=0.
    THY=00.
    THZ=00.
c   rotate using 10 deg increments
    DTH=10.
70  PI=3.14159
    DO 604 I=1,3
    DO 603 J=1,3
603 ROT(J,I)=0.
604 ROT(I,I)=1.
    ANGL(1)=THX
    ANGL(2)=THY
    ANGL(3)=THZ
    NT(1)=1
    NT(2)=2
    NT(3)=3
    I=0
302 I=I+1
    IF(ANGL(I))606,605,606
606 L=NT(I)
    GO TO 612
618 DO 607 J=1,3
    DO 607 JA=1,3
    A(J,JA)=0.
    DO 607 JB=1,3
607 A(J,JA)=A(J,JA)+R1(L,J,JB)*ROT(JB,JA)
    DO 608 K=1,3
    DO 608 J=1,3
608 ROT(K,J)=A(K,J)
605 IF(I-3) 302,303,303
303 DO 805 I=1,NN
    RZ(I)=0.
    DO 806 K=1,3
806 RZ(I)=RZ(I)+ROT(3,K)*R(3*I-3+K)
    DO 805 J=1,2
    RXY(2*I-2+J)=0.
    DO 805 K=1,3
805 RXY(2*I-2+J)=RXY(2*I-2+J)+ROT(J,K)*R(3*I-3+K)
    GO TO 59
612 ANG=ANGL(I)*PI/180.
    IF(L-2)613,614,615

```

```

613 R1(1,2,2)=COS(ANG)
    R1(1,2,3)=SIN(ANG)
    R1(1,3,3)=R1(1,2,2)
    R1(1,3,2)=-R1(1,2,3)
    GO TO 618
614 R1(2,1,1)=COS(ANG)
    R1(2,1,3)=-SIN(ANG)
    R1(2,3,1)=-R1(2,1,3)
    R1(2,3,3)=R1(2,1,1)
    GO TO 618
615 R1(3,1,1)=COS(ANG)
    R1(3,1,2)=SIN(ANG)
    R1(3,2,1)=-R1(3,1,2)
    R1(3,2,2)=R1(3,1,1)
    GO TO 618
59 DO 24 I=1,NN
    X(I)=RXY(2*I-1)
24 Y(I)=RXY(2*I)
    CALL PLOT(NB,NN,X,Y,NP,NM,for,iwrite)
    READ(*,*) IVAL
    IF(IVAL.EQ.+1) GO TO 2000
    IF(IVAL.EQ.-1) GO TO 3000
    IF(IVAL.EQ. 2) GO TO 4000
    IF(IVAL.EQ.-2) GO TO 5000
    IF(IVAL.EQ. 3) GO TO 6000
    IF(IVAL.EQ.-3) GO TO 7000
    IF(IVAL.EQ. 0) GO TO 8000
2000 THX=THX+DTH
    GO TO 70
3000 THX=THX-DTH
    GO TO 70
4000 THY=THY+DTH
    GO TO 70
5000 THY=THY-DTH
    GO TO 70
6000 THZ=THZ+DTH
    GO TO 70
7000 THZ=THZ-DTH
    GO TO 70
8000 CALL UNREGISTERFONTS()
    DUMMY = SETVIDEOMODE( $DEFAULTMODE )
    RETURN
    END

```

Program tr3d02.for

```

    INCLUDE 'FGRAPH.FI'
c   modified for shape finding
C   SPACE TRUSS
C   NONLINEAR VERSION
    DOUBLE PRECISION r(200),P(200),C(150,150),UVEC(3),C1,
1   F1,F2,FAC,C2,D1
    DIMENSION NP(200),NM(200),S(200),PSAVE(200),FSAVE(200)
1   ,ncons(200),r1(200),fi(200),ITYPE(100),nfix(10),ht(10)
1   ,YOUNG(3)

```

```

common c
MAXC=150
C
C   INITIALIZE PARAMETERS/ARRAYS
C
E = 30.0D6
YOUNG(1)=0.
YOUNG(2)=0.
YOUNG(3)=E
FAC=1.
READ(70,2) NB,NN,NS,NSTEP,NIT,nf, isym
nit=6
IF(NSTEP.EQ.0) NSTEP=1
IF(NIT.EQ.0) NIT=1
2 FORMAT (7I5)
WRITE(60,1)NB,NN,NS,NSTEP,NIT,nf, isym
READ(70,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K-1)
1 ,NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
READ(70,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
WRITE(60,160)(L,NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
DO 3013 I=1,NB
3013 FI(I)=FSAVE(I)
151 FORMAT (5X,3I5,E20.8)
160 FORMAT (4I5,E20.8)
1 FORMAT(I5,' NO. MEMBERS'/I5,' NO. NODES'/I5,' NO. SUPPORTS'/
1I5,' NO. LOAD STEPS'/I5,' NO. ITERATIONS'//)
NNS=NN-NS
N=3*NNS
WRITE(60,157)(K,R(3*K-2),R(3*K-1),r(3*k),
1 PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
read(70,654)(nfix(i),ht(i),i=1,nf)
write(60,654)(nfix(i),ht(i),i=1,nf)
654 format(i5,f10.3)
157 FORMAT (1H1,25X,11HCOORDINATES,40X,5HLOADS//
114X,1HX,19X,1HY,19X,1HZ,18X,2HPX,18X,2HPY,18X,2HPZ//
1 (I4,3D17.8,3D17.8))
156 FORMAT (5X,3E20.8,3i2,3f3.0)
do 341 i=1,n
341 NCONS(I)=0
IF(NF.EQ.0) GO TO 5777
DO 5778 I=1,NF
NOD=NFIX(I)
NCONS(3*NOD)=1
NCONS(3*NOD-2)=1
NCONS(3*NOD-1)=1
5778 CONTINUE
5777 IF(ISYM.EQ.0) GO TO 7775
DO 7774 I=1,NN
IF(R(3*I-2).NE.0.)GO TO 7774
NCONS(3*I-2)=1
7774 CONTINUE
7775 CONTINUE
WRITE(60,159)
159 FORMAT (1H1,3X,6HMEMBER,5X,5H+ END,5X,5H- END,16X,4HAREA,
1 11X,9HPRESTRESS//)
do 7769 i=1,n
7769 r1(i)=r(i)

```

```

        call splot(np,nm,nn,nb,r1,fi,0)
7706 CONTINUE
C
C      START LOADSTEPS AND ITERATIONS
      DO 997 LDSTP=1,NSTEP
      STEP=FLOAT(LDSTP+1)/FLOAT(NSTEP)
      IF(LDSTP.EQ.NSTEP) STEP=1.
      DO 997 ITER=1,NIT
      WRITE(60,897) ITER,LDSTP
897  FORMAT(///' ****ITERATION NUMBER',I4/
1      '          LOAD STEP          ',I4)
C
C      SET UP SYSTEM MATRIX
C
      DO 904 I=1,N
      P(I)=PSAVE(I)*STEP
      DO 904 J=1,N
904  C(I,J)=0.
      DO 999 L=1,NB
      write(*,*) iter,l
      K = 3*NP(L)
      M = 3*NM(L)
      CALL UNITV(K,M,C1,UVEC,R)
      IF(K.GT.N) GO TO 888
      P(K-2)=P(K-2)-FSAVE(L)*UVEC(1)
      P(K-1)=P(K-1)-FSAVE(L)*UVEC(2)
      P(K )=P(K )-FSAVE(L)*UVEC(3)
888  IF(M.GT.N) GO TO 887
      P(M-2)=P(M-2)+FSAVE(L)*UVEC(1)
      P(M-1)=P(M-1)+FSAVE(L)*UVEC(2)
      P(M )=P(M )+FSAVE(L)*UVEC(3)
887  continue
      E1=YOUNG(ITYPE(L))
      S(L)=1.
      call sert(K,M,UVEC,MAXC,N,E1,S(1),C1,FSAVE(1),itype(1))
999  CONTINUE
C
C      ERROR AT START OF ITERATION
      C1=0.
      DO 500 I=1,N
      if(ncons(i).ne.0)go to 500
      C1=C1+P(I)**2
500  continue
      C1=DSQRT(C1)
      WRITE(60,501) C1
501  FORMAT(///' ERROR = ',D20.8//)
      do 300 i=1,n
      if( ncons(i).eq.0)go to 300
      do 301 j=1,n
      c(i,j)=0.
301  c(j,i)=0.
      p(i)=0.
      c(i,i)=1.
300  continue
C
C      SOLVE FOR DISPLACEMENTS
C

```

```

c      write(*,9876)((i,j,c(i,j),i=1,n),j=1,n)
9876 format(3(2i5,e20.8))
927 M=N-1
      DO 91 I=1,M
        if(c(i,i).eq.0.) write(*,*) I
        L=I+1
        DO 91 J=L,N
          IF (C(J,I)) 93,91,93
93      DO 92 K=L,N
92      C(J,K)=C(J,K)-C(I,K)*C(J,I)/C(I,I)
          P(J)=P(J)-P(I)          *C(J,I)/C(I,I)
91      CONTINUE
          P(N)=P(N)/C(N,N)
          IF(C(N,N).LE.0.) WRITE(60,298) N
1298 FORMAT(///'***NEG TERM ON THE DIAGONAL AT ROW',I5///)
          DO 94 I=1,M
            K=N-I
            L=K+1
            DO 95 J=L,N
95      P(K)=P(K)-P(J)*C(K,J)
            IF(C(K,K).LE.0.) WRITE(6,298) K
94      P(K)=P(K)/C(K,K)
            WRITE(60,161)(I,P(3*I-2),P(3*I-1),P(3*I),I=1,NNS)
161     FORMAT (1H1,13HDISPLACEMENTS/20X,1HX,19X,1HY,19X,1HZ//
1       1 (I10,3D20.8))
            WRITE(60,162)
162     FORMAT (1H1,3X,6HMEMBER,9X,2HDL,17X,5HFORCE,14X,6HSTRESS,
1       1 7X,'UPDATED MEMBER FORCES'//)
C
C      COMPUTE MEMBER FORCES AND DISPLACEMENTS
C
      DO 998 I=1,NB
        K = 3*NP(I)
        M = 3*NM(I)
        CALL UNITV(K,M,C1,UVEC,R)
        K1=K
        D1=0.
        FAC=1.
        DO 297 J=1,2
          IF(K1.GT.N) GO TO 996
          D1=D1+FAC*(P(K1-2)*UVEC(1)+P(K1-1)*UVEC(2)+P(K1)*UVEC(3))
996     FAC=-1.
          K1=M
297     CONTINUE
          s(i)=1.
          e1=YOUNG(ITYPE(I))
          F1=D1*E1*S(I)/C1
          F2=F1/S(I)
1000    FORMAT (I10,4D20.8)
          UVEC(1)=R(K-2)-R(M-2)
          UVEC(2)=R(K-1)-R(M-1)
          UVEC(3)=R(K )-R(M )
          IF(K.GT.N) GO TO 666
          UVEC(1)=UVEC(1)+P(K-2)
          UVEC(2)=UVEC(2)+P(K-1)
          UVEC(3)=UVEC(3)+P(K )
666     IF(M.GT.N) GO TO 665

```

```

        UVEC(1)=UVEC(1)-P(M-2)
        UVEC(2)=UVEC(2)-P(M-1)
        UVEC(3)=UVEC(3)-P(M )
665 C2=DSQRT(UVEC(1)**2+UVEC(2)**2+UVEC(3)**2)
        C2=C2-C1
        Fsave(i)=Fsave(i)+C2*S(I)*E1/C1
        WRITE(60,1000) I,D1,F1,F2,fsave(i)
998 CONTINUE
1998 continue
C
C      UNDATE COORDINATES
        DO 444 I=1,N
        R(I)=R(I)+P(I)
444 r1(i)=r(i)
C      call splot(np,nm,nn,nb,r1,fi,0)
997 CONTINUE
        call splot(np,nm,nn,nb,r1,fi,0)
        WRITE(600,2)NB,NN,NS,NSTEP,NIT
        WRITE(600,156)(R(3*K-2),R(3*K-1),r(3*k),NCONS(3*K-2),NCONS(3*K
1 -1),NCONS(3*K),PSAVE(3*K-2),PSAVE(3*K-1),PSAVE(3*K),K=1,NN)
        WRITE(600,151)(NP(L),NM(L),ITYPE(L),FSAVE(L),L=1,NB)
        write(600,654)(nfix(i),ht(i),i=1,nf)
299 STOP
        END
C
        SUBROUTINE UNITV(K,M,C1,UVEC,R)
        DOUBLE PRECISION R(1),C1,UVEC(3)
        C1=0.
        DO 1 I=1,3
        UVEC(I)=R(K+I-3)-R(M+I-3)
1 C1=C1+UVEC(I)**2
        C1=DSQRT(C1)
        DO 2 I=1,3
2 UVEC(I)=UVEC(I)/C1
        RETURN
        END
C
C
        SUBROUTINE sert(K,M,UVEC,MAXC,N,E,S,C1,FSAVE,itype)
        DOUBLE PRECISION C(150,150),UVEC(3),C1
        common c
        K1=K
        DO 1 I=1,2
        IF(K1.GT.N) GO TO 1
        M1=K
        DO 2 J=1,2
        IF(M1.GT.N) GO TO 2
        FAC=1.
        IF(I.NE.J) FAC=-1.
        DO 3 L=1,3
        I1=K1-3+L
        DO 3 L1=1,3
        J1=M1-3+L1
        C(I1,J1)=C(I1,J1)+UVEC(L)*UVEC(L1)*(S*E-FSAVE)*FAC/C1
3 IF(L.EQ.L1) C(I1,J1)=C(I1,J1)+FAC*FSAVE/C1
2 M1=M
1 K1=M

```

```

RETURN
END
c
SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1), MI(1), X(1), Y(1),for(1)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
character*6 text
character*10 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH='\newfor\lib\modern.fon'
LIST="t'modern'"/'/h8w8b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
AMAXX=639-20
AMAYY=479-20
c find extent of picture window
XMIN=X(1)
XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
DO 2 I=1,NN
XI=X(I)
YI=Y(I)
IF(XMIN.GT.XI) XMIN=XI
IF(XMAX.LT.XI) XMAX=XI
IF(YMIN.GT.YI) YMIN=YI
2 IF(YMAX.LT.YI) YMAX=YI
c scale to center of window
SCALE = AMAX1((XMAX-XMIN)/AMAXX,(YMAX-YMIN)/AMAYY)
XSHIFT = (XMAX+XMIN)/2.0 - 639/2*SCALE
YSHIFT = (YMAX+YMIN)/2.0 - 479/2*SCALE
c move and draw for each line
DO 3 I=1,NB
K=NP(I)
M=MI(I)
XK=(X(K)-XSHIFT)/SCALE
YK=(Y(K)-YSHIFT)/SCALE
XM=(X(M)-XSHIFT)/SCALE
YM=(Y(M)-YSHIFT)/SCALE
c invert picture
YK = 479-YK
YM = 479-YM
LX=((XK+XM)/2)
LY=((YK+YM)/2)
CALL MOVETO ( XK, YK, XY)
DUMMY = LINETO ( XM, YM)
if(iwrite.ne.2) go to 998
call moveto(lx,ly,xy)
write(text, '(i3)') i
call outgtext (text)
998 if(iwrite.eq.0.or.iwrite.eq.2) go to 3
call moveto(lx,ly,xy)
write(text1,'(f7.0)') for(i)

```

```

    call outgtext (text1)
3  CONTINUE
    if(iwrite.ne.2) go to 996
    do 997 i=1,nn
    lx=(x(i)-xshift)/scale
    yk=(y(i)-yshift)/scale
    ly=(479-yk)
    call moveto(lx,ly,xy)
    write(text, '(i3)') i
    call outgtext (text)
997 continue
996 continue
    RETURN
    END

SUBROUTINE SPLOT ( NP,NM,NN,NB,R,for,iwrite)
INCLUDE 'FGRAPH.FD'
c   iwrite = 0   no text
c           1   writes member forces
c           2   writes node map
DIMENSION NP(1),NM(1),RXY(1000),ROT(3,3),for(1)
DIMENSION ANGL(3),NT(3),A(3,3),R1(3,3,3)
INTEGER*2 DUMMY
DIMENSION R(1),X(200),Y(200),RZ(1000)
WRITE(*,1)
1 FORMAT(' YOU ARE ABOUT TO ENTER A GRAPHICS '
1 'DISPLAY MODE'/' THE KEYBOARD COMMANDS ARE'//
1 '      +1...POSITIVE ROTATION ABOUT X AXIS'/
1 '      -1...NEGATIVE ROTATION ABOUT X AXIS'/
1 '      +2...POSITIVE ROTATION ABOUT Y AXIS'/
1 '      -2...NEGATIVE ROTATION ABOUT Y AXIS'/
1 '      +3...POSITIVE ROTATION ABOUT Z AXIS'/
1 '      -3...NEGATIVE ROTATION ABOUT Z AXIS'/
1 '      0...EXIT')
c   delay for reading
    READ(*,*)
    DO 616 I=1,3
    DO 617 J=1,3
    DO 617 K=1,3
617 R1(I,J,K)=0.
616 R1(I,I,I)=1.
    THX=0.
    THY=00.
    THZ=00.
c   rotate using 10 deg increments
    DTH=10.
70  PI=3.14159
    DO 604 I=1,3
    DO 603 J=1,3
603 ROT(J,I)=0.
604 ROT(I,I)=1.
    ANGL(1)=THX
    ANGL(2)=THY
    ANGL(3)=THZ
    NT(1)=1
    NT(2)=2
    NT(3)=3

```

```

      I=0
302 I=I+1
      IF(ANGL(I)) 606,605,606
606 L=NT(I)
      GO TO 612
618 DO 607 J=1,3
      DO 607 JA=1,3
      A(J,JA)=0.
      DO 607 JB=1,3
607 A(J,JA)=A(J,JA)+R1(L,J,JB)*ROT(JB,JA)
      DO 608 K=1,3
      DO 608 J=1,3
608 ROT(K,J)=A(K,J)
605 IF(I-3) 302,303,303
303 DO 805 I=1,NN
      RZ(I)=0.
      DO 806 K=1,3
806 RZ(I)=RZ(I)+ROT(3,K)*R(3*I-3+K)
      DO 805 J=1,2
      RXY(2*I-2+J)=0.
      DO 805 K=1,3
805 RXY(2*I-2+J)=RXY(2*I-2+J)+ROT(J,K)*R(3*I-3+K)
      GO TO 59
612 ANG=ANGL(I)*PI/180.
      IF(L-2)613,614,615
613 R1(1,2,2)=COS(ANG)
      R1(1,2,3)=SIN(ANG)
      R1(1,3,3)=R1(1,2,2)
      R1(1,3,2)=-R1(1,2,3)
      GO TO 618
614 R1(2,1,1)=COS(ANG)
      R1(2,1,3)=-SIN(ANG)
      R1(2,3,1)=-R1(2,1,3)
      R1(2,3,3)=R1(2,1,1)
      GO TO 618
615 R1(3,1,1)=COS(ANG)
      R1(3,1,2)=SIN(ANG)
      R1(3,2,1)=-R1(3,1,2)
      R1(3,2,2)=R1(3,1,1)
      GO TO 618
59 DO 24 I=1,NN
      X(I)=RXY(2*I-1)
24 Y(I)=RXY(2*I)
      CALL PLOT(NB,NN,X,Y,NP,NM,for,iwrite)
      READ(*,*) IVAL
      IF(IVAL.EQ.+1) GO TO 2000
      IF(IVAL.EQ.-1) GO TO 3000
      IF(IVAL.EQ. 2) GO TO 4000
      IF(IVAL.EQ.-2) GO TO 5000
      IF(IVAL.EQ. 3) GO TO 6000
      IF(IVAL.EQ.-3) GO TO 7000
      IF(IVAL.EQ. 0) GO TO 8000
2000 THX=THX+DTH
      GO TO 70
3000 THX=THX-DTH
      GO TO 70
4000 THY=THY+DTH

```

```

GO TO 70
5000 THY=THY-DTH
GO TO 70
6000 THZ=THZ+DTH
GO TO 70
7000 THZ=THZ-DTH
GO TO 70
8000 CALL UNREGISTERFONTS()
DUMMY = SETVIDEOMODE( $DEFAULTMODE )
RETURN
END

```

Program refl.for

```

include 'fgraph.fi'
C GENERAL MEMBRANE LAYOUT PLOT PROGRAM
c For 1/4 to full layout
DIMENSION FORH(600),X(400),Y(400),Z(400),JT(400,4),IFIX(400)
1 ,NP(600),MI(600),ISYM(400),R(1200),NSTART(1200)
READ(50,1)NB,NN
WRITE(60,1)NB,NN
1 FORMAT(2I5)
READ(50,3)(X(I),Y(I),Z(I),I=1,NN)
WRITE(60,3)(X(I),Y(I),Z(I),I=1,NN)
3 FORMAT(5x,3e20.8)
READ(50,222)(NP(I),MI(I),forh(i),I=1,NB)
222 format(5x,2i5,5x,e20.8)
WRITE(60,2)(FORH(I),NP(I),MI(I),I=1,NB)
2 FORMAT(F10.2,2I5)
c Reflect node point about coord axes
c isym=1 points on the x axis
c isym=2 points on the y axis
c isym=3 center of symmetry

do 777 i=1,nn
isym(i)=0
777 if(x(i).eq.0.) isym(i)=2
NB0=NB
NN0=NN
DO 31 I=1,NN0
NSTART(I)=NN+1
IF(ISYM(I).EQ.2) GO TO 31
22 NN=NN+1
ifix(nn)=ifix(i)
X(NN)=-X(I)
Y(NN)=Y(I)
Z(NN)=Z(I)
31 CONTINUE
DO 61 I=1,NB0
IP=NP(I)
IM=MI(I)
IPS=NSTART(IP)-1
IMS=NSTART(IM)-1
ISUM=ISYM(IP)+ISYM(IM)
if(isum.ge.4)go to 61
62 NB=NB+1

```

```

IPS=IPS+1
IMS=IMS+1
NP(NB)=IPS
MI(NB)=IMS
FORH(NB)=FORH(I)
IF(isym(ip).eq.2) np(nb)=ip
IF(isym(im).eq.2) mi(nb)=im
61 CONTINUE
DO 71 I=1,NN
R(3*I-2)=X(I)
R(3*I-1)=Y(I)
R(3*I) =Z(I)
71 WRITE(6,72)I, R(3*I-2),R(3*I-1),R(3*I)
C   WRITE(150,1)NB,NN
C   WRITE(150,222)(NP(I),MI(I),forh(i),I=1,NB)
C   WRITE(150,3)(X(I),Y(I),Z(I),IFIX(I),ISYM(I),(JT(I,J),J=1,4),I=1,
C   1 NN)
72 FORMAT(I5,3E20.8)
CALL SPLOT(NP,MI,NN,NB,R,forh,0)
STOP
END

C
SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1), MI(1), X(1), Y(1),for(1)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
C record /rccoord/ curpos
character*6 text
character*15 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH=' \NEWFOR\lib\modern.fon'
LIST="t'modern' //" 'h6w6b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
AMAXX=639-20
AMAYY=479-20
XMIN=X(1)
XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
DO 2 I=1,NN
XI=X(I)
YI=Y(I)
IF(XMIN.GT.XI) XMIN=XI
IF(XMAX.LT.XI) XMAX=XI
IF(YMIN.GT.YI) YMIN=YI
2 IF(YMAX.LT.YI) YMAX=YI
SCALE = AMAX1((XMAX-XMIN)/AMAXX,(YMAX-YMIN)/AMAYY)
XSHIFT = (XMAX+XMIN)/2.0 - 639/2*SCALE
YSHIFT = (YMAX+YMIN)/2.0 - 479/2*SCALE
DO 3 I=1,NB
K=NP(I)
M=MI(I)
XK=(X(K)-XSHIFT)/SCALE

```

```

YK=(Y(K)-YSHIFT)/SCALE
XM=(X(M)-XSHIFT)/SCALE
YM=(Y(M)-YSHIFT)/SCALE
YK = 479-YK
YM = 479-YM
LX=((XK+XM)/2)
LY=((YK+YM)/2)
CALL MOVETO ( XK, YK, XY)
DUMMY = LINETO ( XM, YM)
if(iwrite.ne.2) go to 998
call moveto(lx,ly,xy)
write(text, '(i3)') i
call outgtext (text)
998 if(iwrite.eq.0.or.iwrite.eq.2) go to 3
call moveto(lx,ly,xy)
write(text1,'(f7.0)') for(i)
call outgtext (text1)
3 CONTINUE
if(iwrite.ne.2) go to 996
do 997 i=1,nn
lx=(x(i)-xshift)/scale
yk=(y(i)-yshift)/scale
ly=(479-yk)
call moveto(lx,ly,xy)
write(text, '(i3)') i
call outgtext (text)
997 continue
996 continue
RETURN
END

SUBROUTINE SPLOT ( NP,NM,NN,NB,R,for,iwrite)
INCLUDE 'FGRAPH.FD'
DIMENSION NP(1),NM(1),RXY(1000),ROT(3,3),for(1)
DIMENSION ANGL(3),NT(3),A(3,3),R1(3,3,3)
INTEGER*2 DUMMY
DIMENSION R(1),X(500),Y(500),RZ(1000)
WRITE(*,1)
1 FORMAT(' YOU ARE ABOUT TO ENTER A GRAPHICS '
1 'DISPLAY MODE'/' THE KEYBOARD COMMANDS ARE'//
1 ' +1...POSITIVE ROTATION ABOUT X AXIS'/'
1 ' -1...NEGATIVE ROTATION ABOUT X AXIS'/'
1 ' +2...POSITIVE ROTATION ABOUT Y AXIS'/'
1 ' -2...NEGATIVE ROTATION ABOUT Y AXIS'/'
1 ' +3...POSITIVE ROTATION ABOUT Z AXIS'/'
1 ' -3...NEGATIVE ROTATION ABOUT Z AXIS'/'
1 ' 0...EXIT')
READ(*,*)
DO 616 I=1,3
DO 617 J=1,3
DO 617 K=1,3
617 R1(I,J,K)=0.
616 R1(I,I,I)=1.
THX=0.
THY=00.
THZ=00.
DTH=10.

```

```

70  PI=3.14159
    DO 604 I=1,3
    DO 603 J=1,3
603  ROT(J,I)=0.
604  ROT(I,I)=1.
    ANGL(1)=THX
    ANGL(2)=THY
    ANGL(3)=THZ
    NT(1)=1
    NT(2)=2
    NT(3)=3
    I=0
302  I=I+1
    IF(ANGL(I))606,605,606
606  L=NT(I)
    GO TO 612
618  DO 607 J=1,3
    DO 607 JA=1,3
    A(J,JA)=0.
    DO 607 JB=1,3
607  A(J,JA)=A(J,JA)+R1(L,J,JB)*ROT(JB,JA)
    DO 608 K=1,3
    DO 608 J=1,3
608  ROT(K,J)=A(K,J)
605  IF(I-3) 302,303,303
303  DO 805 I=1,NN
    RZ(I)=0.
    DO 806 K=1,3
806  RZ(I)=RZ(I)+ROT(3,K)*R(3*I-3+K)
    DO 805 J=1,2
    RXY(2*I-2+J)=0.
    DO 805 K=1,3
805  RXY(2*I-2+J)=RXY(2*I-2+J)+ROT(J,K)*R(3*I-3+K)
    GO TO 59
612  ANG=ANGL(I)*PI/180.
    IF(L-2)613,614,615
613  R1(1,2,2)=COS(ANG)
    R1(1,2,3)=SIN(ANG)
    R1(1,3,3)=R1(1,2,2)
    R1(1,3,2)=-R1(1,2,3)
    GO TO 618
614  R1(2,1,1)=COS(ANG)
    R1(2,1,3)=-SIN(ANG)
    R1(2,3,1)=-R1(2,1,3)
    R1(2,3,3)=R1(2,1,1)
    GO TO 618
615  R1(3,1,1)=COS(ANG)
    R1(3,1,2)=SIN(ANG)
    R1(3,2,1)=-R1(3,1,2)
    R1(3,2,2)=R1(3,1,1)
    GO TO 618
59  DO 24 I=1,NN
    X(I)=RXY(2*I-1)
24  Y(I)=RXY(2*I)
    CALL PLOT(NB,NN,X,Y,NP,NM,for,iwrite)
    READ(*,*) IVAL
    IF(IVAL.EQ.+1) GO TO 2000

```

```
IF(IVAL.EQ.-1) GO TO 3000
IF(IVAL.EQ. 2) GO TO 4000
IF(IVAL.EQ.-2) GO TO 5000
IF(IVAL.EQ. 3) GO TO 6000
IF(IVAL.EQ.-3) GO TO 7000
IF(IVAL.EQ. 0) GO TO 8000
2000 THX=THX+DTH
GO TO 70
3000 THX=THX-DTH
GO TO 70
4000 THY=THY+DTH
GO TO 70
5000 THY=THY-DTH
GO TO 70
6000 THZ=THZ+DTH
GO TO 70
7000 THZ=THZ-DTH
GO TO 70
8000 CALL UNREGISTERFONTS( )
DUMMY = SETVIDEOMODE( $DEFAULTMODE )
RETURN
END
```