

6. SHELLS

It should be clear that a procedure to design cable nets could be used to design membrane shells. (To some extent a membrane shell is simply an inverted cable net.) That will be described in this section using the batch file hor2.bat:

Hor2.bat

```
horst hor2.dat fort.60 fort.70  
SHELL FORT.70 FORT.60 FORT.150 FORT.8
```

In this simple example shape is determined for the structure shown in Figure 6.1.

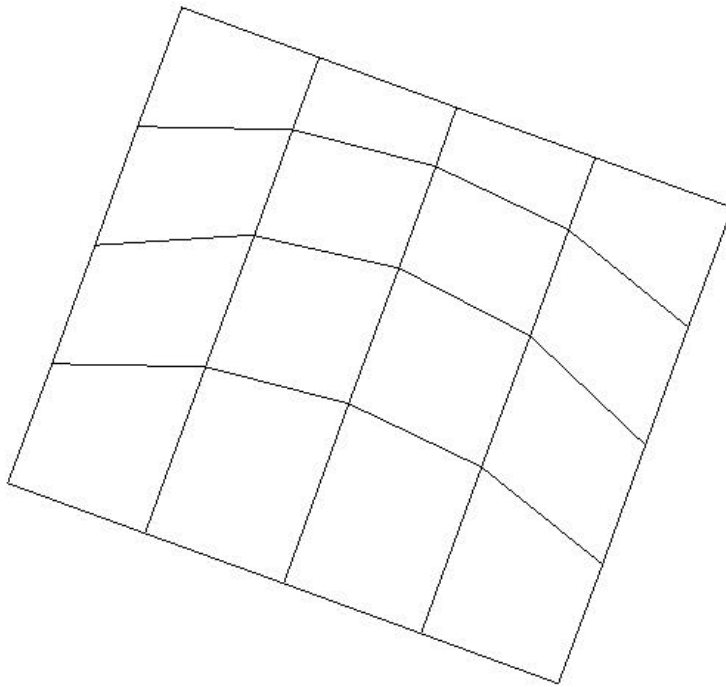


Figure 6.1 A Membrane Shell

This routine first uses the program HORST.FOR to create a node map in this case using the data

```
4 40. 4 40.
```

The program SHELL.FOR is then use to compute shape. This requires modifying the grid method to introduce node loads. (Without node loads the shell of Figure 6.1 would of coures turn out to be flat.) At some point the user is asked to describe the fixed points on the boundary of the shell. That is done by typing the following list

```

1 .001
2 .001
3 .001
4 .001
5 .001
6 .001
10 .001
11 .001
15 .001
16 .001
20 .001
21 .001
22 .001
23 .001
24 .001
25 .001
0 0.

```

The user then types

```

0 (for symmetry)
1. (for node loads)

```

The shell of Figure 6.1 is then produced.

Shell.for

```

include 'fgraph.fi'

C GENERAL MEMBRANE LAYOUT PROGRAM

C Grid Method

DIMENSION FORH(999),X(500),Y(500),Z(500),JT(500,8),IFIX(500)

1 ,NP(999),MI(999),ISYM(500),FOR(999),R(999)

READ(50,1)NB,NN
WRITE(60,1)NB,NN
1 FORMAT(2I5)
READ(50,42)(X(I),Y(I),Z(I),IFIX(I),ISYM(I),(JT(I,J),J=1,8),I=1,NN)
42 FORMAT(5X,3E20.8,2I2,8I5)
READ(50,222)(NP(I),MI(I),FORH(I),I=1,NB)
222 FORMAT(5X,2I5,5X,E20.8)
WRITE(60,2)(I,NP(I),MI(I),FORH(I),I=1,NB)
2 FORMAT(3I5,F10.3)
7778 WRITE(*,*) 'FIXED POINTS...NODE,ELEV'
READ(*,*) NOD,ELEV
IF(NOD.EQ.0) GO TO 7777
Z(NOD)=ELEV
GO TO 7778

```

```

7777 CONTINUE
      write(*,*) 'y axis symmetry?'
      read(*,*) ism
      if(ism.eq.0) go to 6667
      DO 7776 I=1,NB
      IF(X(NP(I)).NE.0..OR.X(MI(I)).NE.0.) GO TO 7776
      FORH(I)=FORH(I)/2.
7776 CONTINUE
6667 continue
      DO 7775 I=1,NN
      K=0
      DO 7775 J=1,NB
      IF(NP(J).NE.I) GO TO 7774
      K=K+1
      JT(I,K)=MI(J)
      GO TO 7775
7774 IF(MI(J).NE.I) GO TO 7775
      K=K+1
      JT(I,K)=NP(J)
7775 CONTINUE
      DO 44 I=1,NN
      R(3*I)=Z(I)
      44 IF(Z(I).NE.0.) IFIX(I)=1
      WRITE(60,333)(X(I),Y(I),Z(I),IFIX(I),ISYM(I),(JT(I,J),J=1,8)
      1,I=1,NN)
333 FORMAT(3F10.3,2I2,8I5)
      3 FORMAT(3F10.3,2I2)
678 FORMAT(5X,E20.8)
      write(*,*) 'ENTER NODE LOAD'
      READ(*,*) PLOAD
      NIT=200
      DO 6 ITER=1,NIT
      DO 5 I=1,NN
      IF(IFIX(I).NE.0) GO TO 5
      SUM=0.
      SUM1=0.
      DO 52 J=1,8
      IF(JT(I,J).EQ.0) GO TO 51
      K=JT(I,J)
      DX=ABS(X(I)-X(K))
      DY=ABS(Y(I)-Y(K))
      DL=SQRT(DX*DX+DY*DY)
      IF(DL.EQ.0.) GO TO 52
      IBAR=0
      DO 62 L=1,NB
      IF(NP(L).EQ.I.AND.MI(L).EQ.K) IBAR=L
      IF(NP(L).EQ.K.AND.MI(L).EQ.I) IBAR=L
      IF(IBAR.NE.0) GO TO 63
62 CONTINUE
      WRITE(6,64)
      STOP
63 CONTINUE
64 FORMAT('DATA ERROR STOP')
      SUM=SUM+FORH(IBAR)/DL
      SUM1=SUM1+FORH(IBAR)*Z(K)/DL
52 CONTINUE
51 Z(I)=(SUM1+pload)/SUM

```

```

5 WRITE(6,10)ITER,I,Z(I)
10 FORMAT(2I5,F10.2)
6 CONTINUE
WRITE(8,24)
24 FORMAT(///16X,'COORDINATES',24X,'FORCE BALLANCE'/
1 16X,'X',11X,'Y',11X,'Z',11X,'X',11X,'Y',11X,'Z')
c Check node equilibrium
c Set up coords for plot routine
DO 151 I=1,NN
R(3*I-2)=X(I)
R(3*I-1)=Y(I)
R(3*I)=Z(I)
SUMX=0.
SUMY=0.
SUMZ=0.
DO 152 J=1,8
IF(JT(I,J).EQ.0) GO TO 151
K=JT(I,J)
DX=X(K)-X(I)
DY=Y(K)-Y(I)
DL=SQRT(DX*DX+DY*DY)
DZ=Z(K)-Z(I)
DL1=SQRT(DL*DL+DZ*DZ)
IF(DL.LT..001.OR.DL1.LT..001) GO TO 152
IBAR=0
DO 162 L=1,NB
IF(NP(L).EQ.I.AND.MI(L).EQ.K) IBAR=L
IF(NP(L).EQ.K.AND.MI(L).EQ.I) IBAR=L
IF(IBAR.NE.0) GO TO 163
162 CONTINUE
WRITE(6,64)
STOP
163 CONTINUE
FORCE=FORH(IBAR)*DL1/DL
FOR(IBAR)=FORCE
SUMX=SUMX+FORCE*DX/DL1
SUMY=SUMY+FORCE*DY/DL1
SUMZ=SUMZ+FORCE*DZ/DL1
152 CONTINUE
151 WRITE(8,22)I,X(I),Y(I),Z(I),SUMX,SUMY,SUMZ
22 FORMAT(I5,6E12.5)
WRITE(150,1) NB,NN
WRITE(150,73)(NP(I),MI(I),FOR(I),I=1,NB)
73 FORMAT(2I5,E20.8)
WRITE(8,23)(I,NP(I),MI(I),FOR(I),I=1,NB)
WRITE(150,333)(X(I),Y(I),Z(I),IFIX(I),ISYM(I),(JT(I,J),J=1,8),I=1,
1 NN)
23 FORMAT(/////7X,'MEMBER OUTPUT'/5X,' +END -END',15X,'FORCE'/
1 (3I5,E20.8))
C call splot(np,mi,nn,nb,r,for,2)
C CALL SPLOT(NP,MI,NN,NB,R,FOR,1)
CALL SPLOT(NP,MI,NN,NB,R,FOR,0)
STOP
END
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'"//'h6w6b'
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

```

