

7. CABLE NETS WITH ELEMENTS OF CONSTANT LENGTH

There are occasions when it is of interest to build cable nets or related structures with elements of constant length. This section describes a method of doing so. In this case we start with a shell structure from chapter 6 which has a circular base. See Figure 7.1.

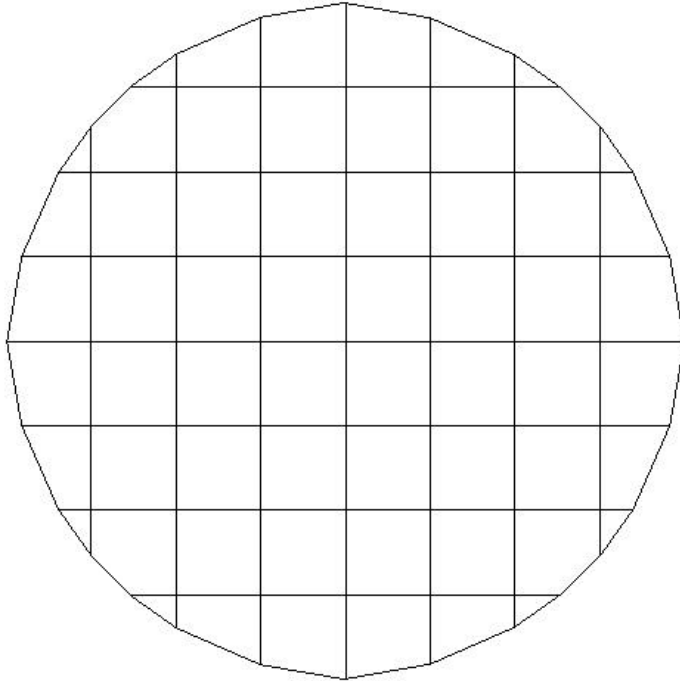


Figure 7.1
Dome with a Circular Base

The problem is to go from this structure in which the elements do not have constant length to one in which they do. (Only the "interior" bars will have constant length.) That distortion is performed by the program `tr3do2.for`. It is executed by the command

```
tr3do2 fort.8 fort.60
```

Fort.8 contains the input for the starting structure which is generated following the discussion of Chapter 6; Fort.60 is a scratch file. The output structure is shown in Figures 7.2 - 7.3.

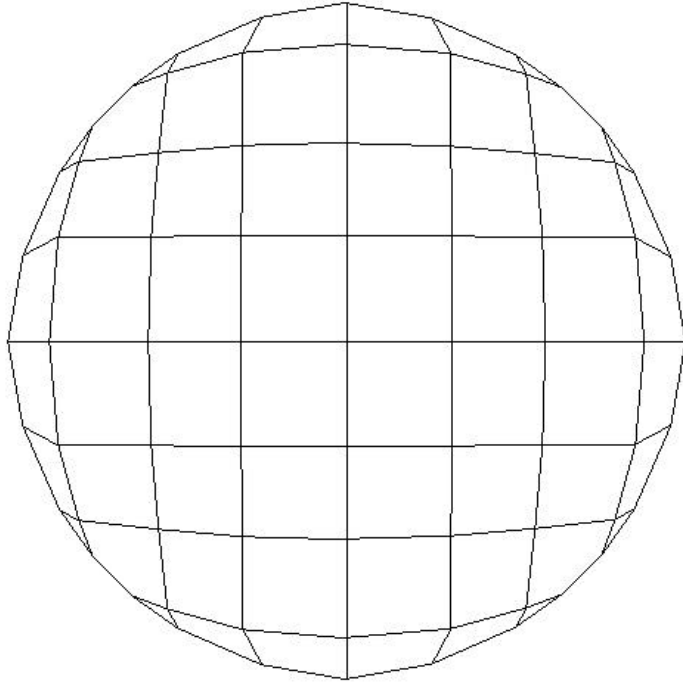


Figure 7.2
Structure with Elements of Equal Length

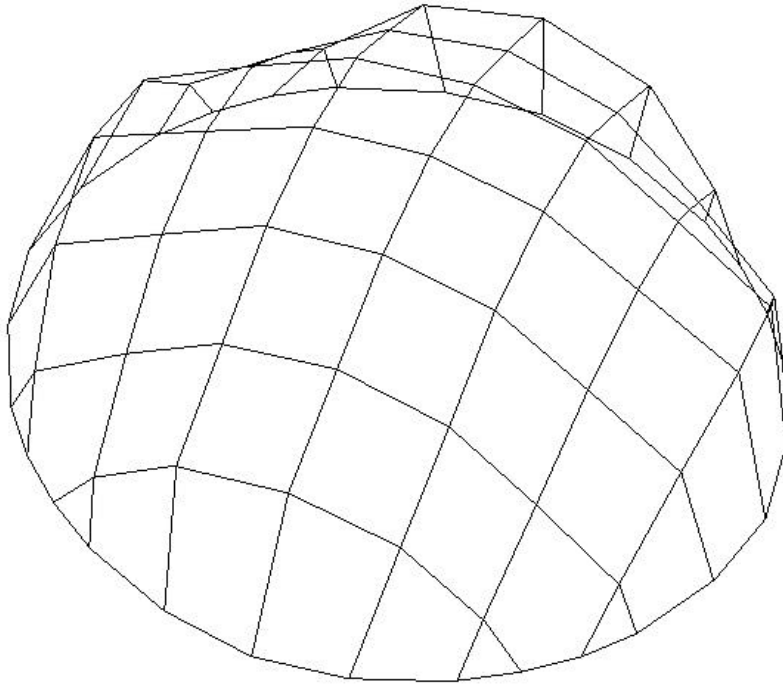


Figure 7.3
Figure 7.2 Rotated

Fort.8

132	73	
4	1	.19847810E+01
3	2	.12608830E+01
8	2	.16068590E+01
4	3	.10338670E+01
9	3	.15988660E+01
5	4	.10338670E+01
10	4	.15924820E+01
6	5	.12608830E+01
11	5	.15988660E+01
12	6	.16068590E+01
8	7	.16068590E+01
14	7	.12608830E+01
9	8	.12546800E+01
15	8	.12546800E+01
10	9	.10318170E+01
16	9	.12506390E+01
11	10	.10318170E+01
17	10	.12487230E+01

12	11	.12546800E+01
18	11	.12506390E+01
13	12	.16068590E+01
19	12	.12546800E+01
20	13	.12608830E+01
15	14	.15988660E+01
22	14	.10338670E+01
16	15	.12506390E+01
23	15	.10318170E+01
17	16	.10310350E+01
24	16	.10310350E+01
18	17	.10310350E+01
25	17	.10307760E+01
19	18	.12506390E+01
26	18	.10310350E+01
20	19	.15988660E+01
27	19	.10318170E+01
28	20	.10338670E+01
22	21	.19847810E+01
23	22	.15924820E+01
30	22	.10338670E+01
24	23	.12487230E+01
31	23	.10318170E+01
25	24	.10307760E+01
32	24	.10310350E+01
26	25	.10307760E+01
33	25	.10307760E+01
27	26	.12487230E+01
34	26	.10310350E+01
28	27	.15924820E+01
35	27	.10318170E+01
29	28	.19847810E+01
36	28	.10338670E+01
31	30	.15988660E+01
37	30	.12608830E+01
32	31	.12506390E+01
38	31	.12546800E+01
33	32	.10310350E+01
39	32	.12506380E+01
34	33	.10310350E+01
40	33	.12487230E+01
35	34	.12506390E+01
41	34	.12506380E+01
36	35	.15988660E+01
42	35	.12546800E+01
43	36	.12608830E+01
38	37	.16068590E+01
39	38	.12546800E+01
44	38	.16068590E+01
40	39	.10318170E+01
45	39	.15988660E+01
41	40	.10318170E+01
46	40	.15924820E+01
42	41	.12546800E+01
47	41	.15988670E+01
43	42	.16068590E+01
48	42	.16068590E+01

45	44	.12608830E+01
46	45	.10338670E+01
47	46	.10338670E+01
49	46	.19847810E+01
48	47	.12608830E+01
50	1	.41863640E+01
50	3	.20086190E+01
51	50	.41863640E+01
51	2	.20184450E+01
52	51	.41863640E+01
52	2	.16183780E+01
53	52	.41863640E+01
53	7	.16183780E+01
54	53	.41863640E+01
54	7	.20184450E+01
55	54	.41863640E+01
55	14	.20086190E+01
21	55	.41863640E+01
56	21	.41863640E+01
56	30	.20086190E+01
57	56	.41863640E+01
57	37	.20184450E+01
58	57	.41863640E+01
58	37	.16183780E+01
59	58	.41863640E+01
59	44	.16183780E+01
60	59	.41863640E+01
60	44	.20184460E+01
61	60	.41863640E+01
61	45	.20086190E+01
49	61	.41863640E+01
62	49	.41863640E+01
62	47	.20086190E+01
63	62	.41863640E+01
63	48	.20184450E+01
64	63	.41863640E+01
64	48	.16183780E+01
65	64	.41863640E+01
65	43	.16183780E+01
66	65	.41863640E+01
66	43	.20184450E+01
67	66	.41863640E+01
67	36	.20086190E+01
29	67	.41863640E+01
68	29	.41863640E+01
68	20	.20086190E+01
69	68	.41863640E+01
69	13	.20184450E+01
70	69	.41863640E+01
70	13	.16183780E+01
71	70	.41863640E+01
71	6	.16183780E+01
72	71	.41863640E+01
72	6	.20184450E+01
73	72	.41863640E+01
73	5	.20086190E+01
1	73	.41863640E+01

0	40.000	.000	.001	1	0	4	50	73	0	0	0	0
0	20.000	10.000	6.841	0	0	3	8	51	52	0	0	0
0	30.000	10.000	14.521	0	0	2	4	9	50	0	0	0
0	40.000	10.000	17.146	0	0	1	3	5	10	0	0	0
0	50.000	10.000	14.521	0	0	4	6	11	73	0	0	0
0	60.000	10.000	6.841	0	0	5	12	71	72	0	0	0
0	10.000	20.000	6.841	0	0	8	14	53	54	0	0	0
0	20.000	20.000	19.419	0	0	2	7	9	15	0	0	0
0	30.000	20.000	26.997	0	0	3	8	10	16	0	0	0
0	40.000	20.000	29.539	0	0	4	9	11	17	0	0	0
0	50.000	20.000	26.997	0	0	5	10	12	18	0	0	0
0	60.000	20.000	19.419	0	0	6	11	13	19	0	0	0
0	70.000	20.000	6.841	0	0	12	20	69	70	0	0	0
0	10.000	30.000	14.521	0	0	7	15	22	55	0	0	0
0	20.000	30.000	26.997	0	0	8	14	16	23	0	0	0
0	30.000	30.000	34.507	0	0	9	15	17	24	0	0	0
0	40.000	30.000	37.018	0	0	10	16	18	25	0	0	0
0	50.000	30.000	34.507	0	0	11	17	19	26	0	0	0
0	60.000	30.000	26.997	0	0	12	18	20	27	0	0	0
0	70.000	30.000	14.521	0	0	13	19	28	68	0	0	0
0	.000	40.000	.001	1	0	22	55	56	0	0	0	0
0	10.000	40.000	17.146	0	0	14	21	23	30	0	0	0
0	20.000	40.000	29.539	0	0	15	22	24	31	0	0	0
0	30.000	40.000	37.018	0	0	16	23	25	32	0	0	0
0	40.000	40.000	39.518	0	0	17	24	26	33	0	0	0
0	50.000	40.000	37.018	0	0	18	25	27	34	0	0	0
0	60.000	40.000	29.539	0	0	19	26	28	35	0	0	0
0	70.000	40.000	17.146	0	0	20	27	29	36	0	0	0

0	80.000	40.000	.001	1	0	28	67	68	0	0	0	0
0	10.000	50.000	14.521	0	0	22	31	37	56	0	0	0
0	20.000	50.000	26.997	0	0	23	30	32	38	0	0	0
0	30.000	50.000	34.507	0	0	24	31	33	39	0	0	0
0	40.000	50.000	37.018	0	0	25	32	34	40	0	0	0
0	50.000	50.000	34.507	0	0	26	33	35	41	0	0	0
0	60.000	50.000	26.997	0	0	27	34	36	42	0	0	0
0	70.000	50.000	14.521	0	0	28	35	43	67	0	0	0
0	10.000	60.000	6.841	0	0	30	38	57	58	0	0	0
0	20.000	60.000	19.419	0	0	31	37	39	44	0	0	0
0	30.000	60.000	26.997	0	0	32	38	40	45	0	0	0
0	40.000	60.000	29.539	0	0	33	39	41	46	0	0	0
0	50.000	60.000	26.997	0	0	34	40	42	47	0	0	0
0	60.000	60.000	19.419	0	0	35	41	43	48	0	0	0
0	70.000	60.000	6.841	0	0	36	42	65	66	0	0	0
0	20.000	70.000	6.841	0	0	38	45	59	60	0	0	0
0	30.000	70.000	14.521	0	0	39	44	46	61	0	0	0
0	40.000	70.000	17.146	0	0	40	45	47	49	0	0	0
0	50.000	70.000	14.521	0	0	41	46	48	62	0	0	0
0	60.000	70.000	6.841	0	0	42	47	63	64	0	0	0
0	40.000	80.000	.001	1	0	46	61	62	0	0	0	0
0	30.000	1.665	.001	1	0	1	3	51	0	0	0	0
0	20.000	6.099	.001	1	0	50	2	52	0	0	0	0
0	14.625	10.000	.001	1	0	51	2	53	0	0	0	0
0	10.000	14.625	.001	1	0	52	7	54	0	0	0	0
0	6.099	20.000	.001	1	0	53	7	55	0	0	0	0
0	1.665	30.000	.001	1	0	54	14	21	0	0	0	0
0	1.665	50.000	.001	1	0	21	30	57	0	0	0	0

0	6.099	60.000	.001	1	0	56	37	58	0	0	0	0
0	10.000	65.375	.001	1	0	57	37	59	0	0	0	0
0	14.625	70.000	.001	1	0	58	44	60	0	0	0	0
0	20.000	73.901	.001	1	0	59	44	61	0	0	0	0
0	30.000	78.335	.001	1	0	60	45	49	0	0	0	0
0	50.000	78.335	.001	1	0	49	47	63	0	0	0	0
0	60.000	73.901	.001	1	0	62	48	64	0	0	0	0
0	65.375	70.000	.001	1	0	63	48	65	0	0	0	0
0	70.000	65.375	.001	1	0	64	43	66	0	0	0	0
0	73.901	60.000	.001	1	0	65	43	67	0	0	0	0
0	78.335	50.000	.001	1	0	66	36	29	0	0	0	0
0	78.335	30.000	.001	1	0	29	20	69	0	0	0	0
0	73.901	20.000	.001	1	0	68	13	70	0	0	0	0
0	70.000	14.625	.001	1	0	69	13	71	0	0	0	0
0	65.375	10.000	.001	1	0	70	6	72	0	0	0	0
0	60.000	6.099	.001	1	0	71	6	73	0	0	0	0
0	50.000	1.665	.001	1	0	72	5	1	0	0	0	0

tr3d02.for

```

INCLUDE 'FGRAPH.FI'
c   modified for shape finding
C   SPACE TRUSS
C   NONLINEAR VERSION
DOUBLE PRECISION r(220),P(220),C(220,220),UVEC(3),C1,
1  F1,F2,FAC,C2,D1
DIMENSION NP(220),NM(220),S(220),PSAVE(220),FSAVE(220)
1  ,ncons(220),r1(220),fi(220),ITYPE(220),nfix(220),ht(220)
1  ,YOUNG(3),d(220),al(220),mark(220)
common c
MAXC=220

C
C   INITIALIZE PARAMETERS/ARRAYS
C
E = 30.0D6
YOUNG(1)=0.
YOUNG(2)=0.

```

```

YOUNG(3)=E
FAC=1.
READ(8,2) NB,NN
ns=0
write(60,*)nb,nn
READ(8,151)(NP(L),NM(L),FSAVE(L),L=1,NB)
write(60,151)(NP(L),NM(L),FSAVE(L),L=1,NB)
nit=6
IF(NSTEP.EQ.0) NSTEP=1
IF(NIT.EQ.0) NIT=1
2 FORMAT (7I5)
READ(8,156)(R(3*K-2),R(3*K-1),r(3*k),nfix(k),K=1,NN)
WRITE(60,156)(R(3*K-2),R(3*K-1),r(3*k),nfix(k),K=1,NN)
DO 3013 I=1,NB
s(i)=10.
mark(i)=0
if(nfix(np(i)).ne.1.and.nfix(nm(i)).ne.1) mark(i)=1
3013 FI(I)=FSAVE(I)
N=3*NN
do 3769 i=1,n
ncons(i)=0
3769 r1(i)=r(i)
call splot(np,nm,nn,nb,r1,fi,0)
do 2769 i=1,nn
psave(3*i-2)=0.
psave(3*i-1)=0.
psave(3*i)=1.
if (nfix(i).eq.0) go to 2768
ncons(3*i-2)=1
ncons(3*i-1)=1
ncons(3*i)=1
2768 continue
2769 continue
sum=0.
nct=0
do 2767 i=1,nb
K = 3*NP(i)
M = 3*NM(i)
CALL UNITV(K,M,C1,UVEC,R)
al(i)=c1
if(mark(i).eq.0) goto 2767
nct=nct+1
sum=sum+c1
2767 continue
avgl=sum/float(nct)
write(60,*)avgl,nct
do 2765 i=1,nb
d(i)=0.
if(mark(i).ne.0)d(i)=avgl-al(i)
write(60,*)'D',i,d(i)
2765 continue
151 FORMAT (2I5,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
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)
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 (3f10.2,i2)
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
nstep=1
DO 997 LDSTP=1,NSTEP
STEP=FLOAT(LDSTP+1)/FLOAT(NSTEP)
IF(LDSTP.EQ.NSTEP) STEP=1.
nit=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)
ak=s(l)*e/c1
IF(K.GT.N) GO TO 888
P(K-2)=P(K-2)-(fsave(l)-d(l)*ak)*UVEC(1)
P(K-1)=P(K-1)-(fsave(l)-d(l)*ak)*UVEC(2)
P(K )=P(K )-(fsave(l)-d(l)*ak)*UVEC(3)
888 IF(M.GT.N) GO TO 887
P(M-2)=P(M-2)+(fsave(l)-d(l)*ak)*UVEC(1)
P(M-1)=P(M-1)+(fsave(l)-d(l)*ak)*UVEC(2)
P(M )=P(M )+(fsave(l)-d(l)*ak)*UVEC(3)
887 continue
E1=YOUNG(ITYPE(L))
call sert(K,M,UVEC,MAXC,N,E,S(l),C1,FSAVE(l),itype(l))
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 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=fsave(i)+(D1-d(i))*E*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-d(i)
   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)
   do 3365 i=1,nb
   K = 3*NP(i)
   M = 3*NM(i)
   CALL UNITV(K,M,C1,UVEC,R)
   al(i)=c1
3365 write(60,*) 'length',i,c1,mark(i)
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(220,220),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='\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)
      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,nm
      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
```