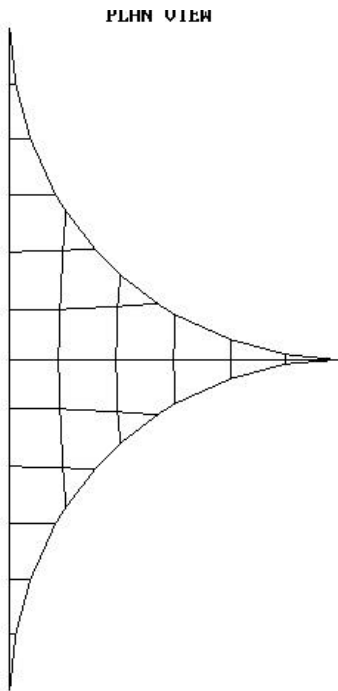


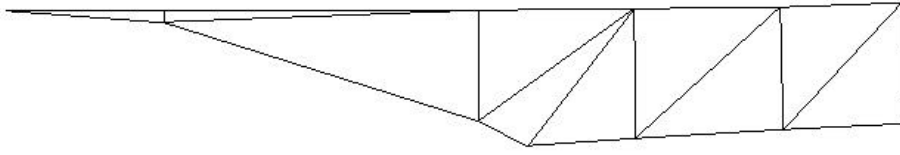
5. PATTERNING

This chapter continues Chapter 4. It discusses the program ALI.FOR listed in the batch file of that chapter which "patterns" the structure, that is it produces output describing how the curved surface of the roof is to be cut from flat rolls of material.

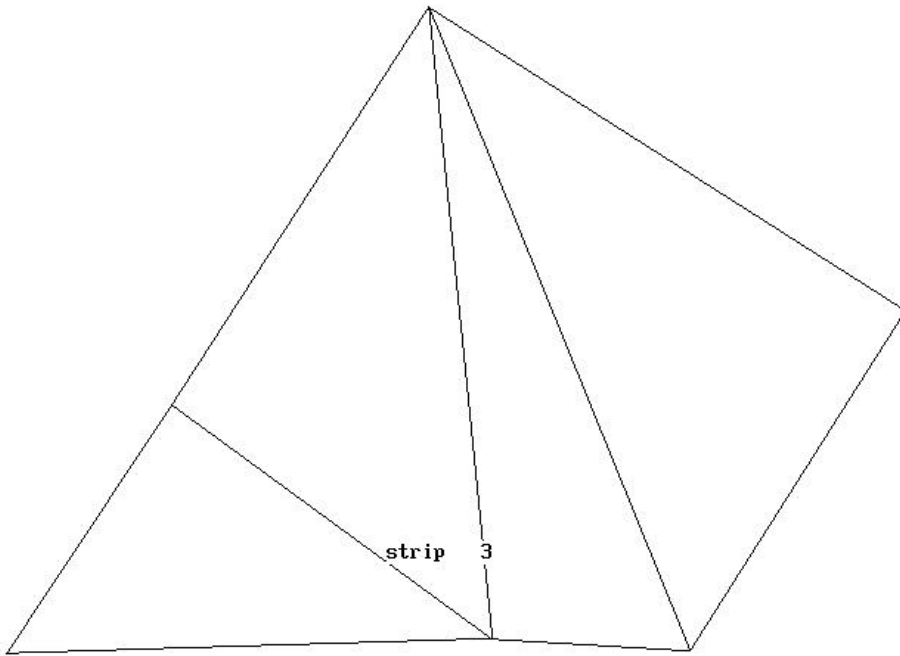
In the case of the work described here, patterning is done directly from the existing nodes. Strips are defined in terms of the existing nodes by listing a sequence of nodes. This curved strip is then projected onto a flat surface for cutting from rolls of material.

The graphical output from the program is shown in Figure 5.1. (Node numbers are omitted.) In this case only 1/4 of the structure is patterned due to symmetry. The program first shows a plan of the structure; then individual strips are indicated. Note that these strips are not drawn to scale.

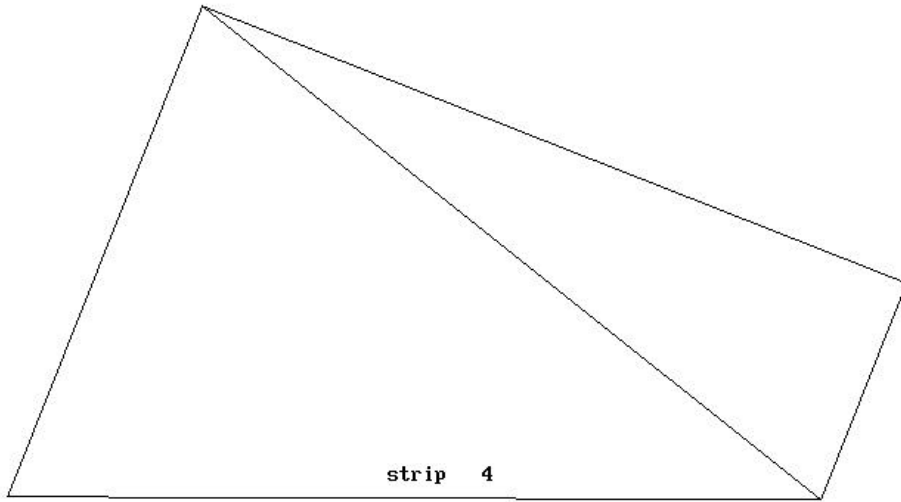


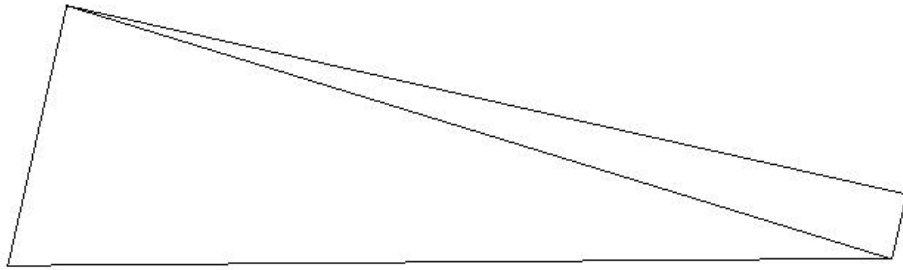


strip 1

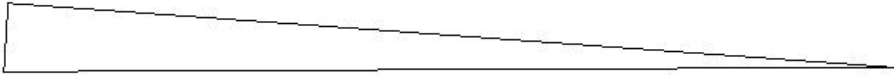


strip 3





strip 5



strip 6

These figures have been produced from the following input.

Pat.dat

```
6
12
1 2 45 4 43
5 42 6 -7 8
10 11
8
42 6 41 8 40
9 -11 12
6
40 9 39 12 38
13
4
38 13 37 14
4
37 14 36 15
3
36 15 16
```

The first line indicated that there are going to be 6 strips. Following that there are six data sets, each begins with the number of points in the strip followed by the points themselves.

Strip 1 then is to contain 12 points. The 12 points are then listed. Strip 2 is to contain 8 points, etc.

Ali.for

```

include 'fgraph.fi'
C   GENERAL MEMBRANE LAYOUT PLOT PROGRAM
C   PATTERNING PROGRAM
COMMON xmax,xmin,ymax,ymin,i,list
DIMENSION FORH(600),X(400),Y(400),Z(400),JT(400,8),IFIX(400)
1  ,NP(600),MI(600),ISYM(600),R(1000)
2  ,LIST(200),XNEW(200),YNEW(200),XB(200),YB(200),OFFSET(300,2)
READ(50,1)NB,NN
WRITE(*,1)NB,NN
1  FORMAT(2I5,F10.0)
READ(50,3)(X(I),Y(I),Z(I),I=1,
1  NN)
C   WRITE(60,3)(X(I),Y(I),Z(I),I=1,
C   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(*,2)(FORH(I),NP(I),MI(I),I=1,NB)
2  FORMAT(F10.2,2I5)
c   Plot bar lengths
DO 10 I=1,NB
J=NP(I)
K=MI(I)
10 FORH(I)=SQRT((X(J)-X(K))**2+(Y(J)-Y(K))**2+(Z(J)-Z(K))**2)
CALL CONVRT(X,Y,z,R,NN)
CALL SPLOT(NP,MI,NN,NB,R,FORH,4)
write(10,75)(i,x(i),y(i),z(i),i=1,nn)
75 FORMAT('1 STARTING JOINT COORDINATES'///5X,'JOINT',19X,'X',
1 19X,'Y',19X,'Z'/(I10,3E20.8))
WRITE(10,76)(I,NP(I),MI(I),FORH(I),I=1,NB)
76 FORMAT('1 STARTING MEMBER LENGTHS'///15X,'+ END',5X,'- END',
1 14X,'LENGTH'/(3I10,E20.8))
WRITE(*,*)'Do you wish to compensate for initial strain?'
1 (yes=1,no=2)'
ex=0.
ey=0.
read(*,*)noyes
if(noyes.eq.2) go to 771
write(*,*) 'Enter strain in the strip direction'
read(*,*) ex
write(*,*) 'Enter strain in the transverse direction'
read(*,*) ey
write(*,*) ex,ey
771 READ(150,1)NSTR
YORIG=29.
STR=NSTR
WIDS=2.
THETA=0.
c   Loop over strips

```

```

DO 11 I=1,NSTR
ANGLE=0.
FAC=1.
YORIG=YORIG-WIDS
READ(150,1)NODES
C write(60,1)nodes
READ(150,20)(LIST(J),J=1,NODES)
C WRITE(60,20)(LIST(J),J=1,NODES)
20 FORMAT(5I5)
IBAR=NODES-1
N1=NODES-1
J=LIST(1)
XNEW(1)=0.
YNEW(1)=0.
DO 12 J=1,N1
c This is the typical case
IF(LIST(J+1).GT.0) GO TO 1200
IP=LIST(J)
IM=LIST(J-2)
DX=X(IP)-X(IM)
DY=Y(IP)-Y(IM)
DZ=Z(IP)-Z(IM)
ALEN=SQRT(DX*DX+DY*DY+DZ*DZ)
UVX=DX/ALEN
UVY=DY/ALEN
UVZ=DZ/ALEN
FAC=1.
THETA=ACOS(UVX*OLDX+UVY*OLDY+UVZ*OLDZ)
IF(OLDX*UVY-OLDY*UVX.LT.0.) FAC=-1.
ANGLE=ANGLE+FAC*THETA
OLDX=UVX
OLDY=UVY
OLDZ=UVZ
c Unit vector for new line
1200 IP=IABS(LIST(J+1))
IM=LIST(J)
DX=X(IP)-X(IM)
DY=Y(IP)-Y(IM)
DZ=Z(IP)-Z(IM)
ALEN=SQRT(DX*DX+DY*DY+DZ*DZ)
FORH(J)=ALEN
UVX=DX/ALEN
UVY=DY/ALEN
UVZ=DZ/ALEN
IF(J.EQ.1)ANGLE=3.14159/2.-ACOS(UVY)
IF(J.EQ.1) GO TO 13
FAC=1.
THETA=ACOS(UVX*OLDX+UVY*OLDY+UVZ*OLDZ)
IF(OLDX*UVY-OLDY*UVX.LT.0.) FAC=-1.
ANGLE=ANGLE+FAC*THETA
cCoordinates of next point
13 XNEW(J+1)=XNEW(J)+ALEN*COS(ANGLE)
YNEW(J+1)=YNEW(J)+ALEN*SIN(ANGLE)
NP(J)=J+1
MI(J)=J
IF(J.EQ.1) GO TO 85
IBAR=IBAR+1

```

```

NP( IBAR )=J+1
MI( IBAR )=J-1
IP=LIST(J+1)
IM=LIST(J-1)
IF(IP.GT.0) GO TO 1201
MI( IBAR )=J-2
IP=-IP
LIST(J+1)=-LIST(J+1)
MI(J-1)=J-2
MI( IBAR-1)=J-1
1201 DX=X(IP)-X(IM)
DY=Y(IP)-Y(IM)
DZ=Z(IP)-Z(IM)
ALEN=SQRT(DX*DX+DY*DY+DZ*DZ)
FORH( IBAR )=ALEN
c Save computed unit vector
85 OLDX=UVX
OLDY=UVY
OLDZ=UVZ
95 FORMAT(6F10.3)
12 CONTINUE
MARK=0
DTH=2.*3.14159/360.
TH=0.
CALL ROTATE(XNEW,YNEW,XB,YB,TH,DIFF,NODES)
SAVE=DIFF
TH=TH+DTH
51 CALL ROTATE(XNEW,YNEW,XB,YB,TH,DIFF,NODES)
IF(DIFF.GE.SAVE) GO TO 14
SAVE=DIFF
TH=TH+DTH
MARK=1
GO TO 51
14 IF(MARK.EQ.1) GO TO 15
TH=TH-2*DTH
DTH=-DTH
16 CALL ROTATE(XNEW,YNEW,XB,YB,TH,DIFF,NODES)
IF(DIFF.GE.SAVE) GO TO 15
SAVE=DIFF
TH=TH+DTH
GO TO 16
99 CONTINUE
15 CONTINUE
TH=TH-DTH
CALL ROTATE(XNEW,YNEW,XB,YB,TH,DIFF,NODES)
WRITE(10,77)I,DIFF,EX,EY,(L,XB(L),YB(L),LIST(L),L=1,NODES)
77 FORMAT('1 STRIP ',I5/' STRIP WIDTH ',E20.8/
1 ' COMPENSATION'/' LONGITUDINAL',E20.8/
1 ' TRANSVERSE',E20.8//
1 ' STRIP COORDINATES'//23X,'X',14X,'Y',6X,'NODES'/
1 (I5,2E20.8,I5))
WRITE(10,78)(L,LIST(NP(L)),LIST(MI(L)),FORH(L),L=1,IBAR)
78 FORMAT('1 STRIP LENGTHS'//15X,'+ END',5X,'- END',14X,'LENGTH'/
1 (3I10,E20.8))
CALL CONVRT(XB,YB,z,R,NODES)
CALL SPLOT(NP,MI,NODES,2*NODES-3,R,FORH,5)
C WRITE(60,19)(K,XB(K),YB(K),K=1,NODES)

```

```

WRITE(10,200)
200 FORMAT(////'1          2 FT OFFSETS'///' DIST',7X,'TOP',4X,'BOTTOM'
1 )
DO 765 j=1,NODES
XB(j)=XB(j)*(1.-EX)
765 YB(j)=YB(j)*(1.-EY)
DO 100 J=1,300
DO 100 K=1,2
100 OFFSET(J,K)=0.
SPG=2.
NBS=2*NODES-3
DO 101 J=1,NBS
IF(J.GT.1.AND.J.LT.NODES-1) GO TO 101
IP=NP(J)
IM=MI(J)
IF(XB(IP).GT.XB(IM)) GO TO 102
IM=IP
IP=MI(J)
102 START=XB(IM)-XMIN
DX=XB(IP)-XB(IM)
DY=YB(IP)-YB(IM)
NEXT=(START+SPG)/SPG
104 XNEXT=NEXT*SPG
IF(XNEXT.GT.XB(IP)-XMIN) GO TO 101
YNEXT=DY*((XNEXT-XB(IM)+XMIN)/DX)+YB(IM)-YMIN
DO 103 K=1,2
IF(OFFSET(NEXT,K).NE.0.) GO TO 103
OFFSET(NEXT,K)=YNEXT
NEXT=NEXT+1
GO TO 104
103 CONTINUE
101 CONTINUE
DO 105 J=1,300
IF(OFFSET(J,1).EQ.0.) GO TO 106
J1=SPG*J
I1=1
I2=2
IF(ABS(OFFSET(J,1)).GT.ABS(OFFSET(J,2))) GO TO 107
I1=2
I2=1
107 WRITE(10,108) J1,OFFSET(J,I1),OFFSET(J,I2)
108 FORMAT(I5,2F10.3)
105 CONTINUE
106 CONTINUE
19 FORMAT(I5,2E20.8)
X1=XB(1)-XMIN
Y1=YB(1)-YMIN
X2=XB(2)-XMIN
Y2=YB(2)-YMIN
X3=XB(NODES-1)-XMIN
Y3=YB(NODES-1)-YMIN
X4=XB(NODES)-XMIN
Y4=YB(NODES)-YMIN
WRITE(10,201) X1,Y1,LIST(1),X2,Y2,LIST(2),X3,Y3,LIST(NODES-1),
1 X4,Y4,LIST(NODES)
201 FORMAT(////'          END POINT OFFSETS'//6X,'X',10X,'Y',5X,'NODE'//
1 (2F10.3,I5))

```

```

C      read(*,*)pdq1
11 CONTINUE
STOP
END

C
SUBROUTINE ROTATE(XNEW,YNEW,X,Y,TH,DIFF,NN)
COMMON XMAX,XMIN,YMAX,YMIN
DIMENSION XNEW(1),YNEW(1),X(1),Y(1)
DO 1 I=1,NN
X(I)=XNEW(I)*COS(TH)+YNEW(I)*SIN(TH)
1 Y(I)=-XNEW(I)*SIN(TH)+YNEW(I)*COS(TH)
XMIN=X(1)
XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
DO 2 I=1,NN
IF(X(I).LT.XMIN) XMIN=X(I)
IF(X(I).GT.XMAX) XMAX=X(I)
IF(Y(I).LT.YMIN) YMIN=Y(I)
IF(Y(I).GT.YMAX) YMAX=Y(I)
2 CONTINUE
DIFF=YMAX-YMIN
C WRITE(7,10)TH,DIFF
10 FORMAT(2E20.8)
RETURN
END

C
SUBROUTINE CONVRT(X,Y,z,R,NN)
DIMENSION X(1),Y(1), R(1),z(1)
DO 1 I=1,NN
R(3*I-2)=x(i)
R(3*I-1)=y(I)
1 R(3*I) =z(I)
RETURN
END

C
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
C 4 writes member forces and node numbers
c 5 writes pattern data
common axmax,axmin,aymax,aymin,nstr,list
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),list(200)
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
C
SUBROUTINE PLOT(NB, NN, X, Y, NP, MI,for,iwrite)
INCLUDE 'FGRAPH.FD'
common axmax,axmin,aymax,aymin,nstr,tisl
DIMENSION NP(1), MI(1), X(1), Y(1),for(1)
integer tisl(200)
INTEGER*2 DUMMY,xk,yk,xm,ym,lx,ly
RECORD /XYCOORD/ XY
record / rccoord / curpos
character*3 text
character*6 text1
CHARACTER*64 FONTPATH
CHARACTER*20 LIST
FONTPATH='\NEWFOR\lib\modern.fon'
LIST="t'modern' //" 'h7w7b'
DUMMY = SETVIDEOMODE( $VRES16COLOR)
DUMMY=REGISTERFONTS(FONTPATH)
DUMMY=SETFONT(LIST)
AMAXX=639-30
AMAYY=479-30
c find extent of picture window
XMIN=X(1)

```

```

XMAX=X(1)
YMIN=Y(1)
YMAX=Y(1)
if(iwrite.eq.4) call settextposition (1,35,xy)
if(iwrite.eq.4) call outtext('PLAN VIEW')
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.eq.4.or .iwrite.eq.5) go to 9
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
9 call moveto(lx,ly,xy)
write(text1,'(f4.1)') for(i)
call outgtext (text1)
3 CONTINUE
if(iwrite.eq.4.or .iwrite.eq.5) go to 99
if(iwrite.ne.2) go to 996
99 do 997 i=1,nn
lx=(x(i)-xshift)/scale
yk=(y(i)-yshift)/scale
ly=(479-yk)
call moveto(lx,ly,xy)
if(iwrite.ne.5) go to 10
write(text,'(i3)') tisl(i)
go to 11
10 write(text, '(i3)') i
11 call outgtext (text)
997 continue
996 continue
if(iwrite.ne.5) return
call settextposition(25,35,xy)

```

```
write(text,'(i3)')nstr
call outtext('strip '//text)
RETURN
END
```