A COMPUTER PROGRAM TO PLOT AN ISOMETRIC PROJECTION OF A SOLUTION SPACE SURFACE

Charles R. Brauer

Utah University

Prepared for:

Advanced Research Projects Agency

August 1968

Technical Report 4-9

Charles R. Brauer

A COMPUTER PROGRAM TO PLOT AN ISOMETRIC PROJECTION OF A SOLUTION SPACE SURFACE

August 1968

COMPUTER SCIENCE
Information Processing Systems
University of Utah
Salt Lake City, Utah

Advanced Research Projects Agency Department of Defense , ARPA order 829
Program code number 6D30

Subroutine ISOPLT was developed for the computer produced display of a solution space surface. The solution space, for the sample plots given in Appendix B, is that of various flow functions of a time dependent, viscous, incompressible fuild flow for various boundary configurations. Also shown in Appendix B is the surface plot of

$$\text{SIN}\ (8*(X-1)/X_L+1/4(Y-1))+1.0 \qquad (1)$$

Appendix A gives a sample of the calling sequence for equation (1) in the MAIN program, and also a listing of subrouting ISOPLT.

The algorithm used in ISOPLT can be stated as:

1. Draw the right most line first, moving from right to left.
2. Lift the pen (blank the display vector) when a line segment drops below (vertical reference) any previously drawn line segment, when the line segment again moves above the vertical reference line.)

PROGRAM VARIABLES

TA(I,J) - functional value at the point I,J

M - number of columns in TA

N - number of rows in TA

XP - size of plot (in inches) in horizontal direction

YP - size of plot in vertical direction

BND(I,J) - mode point type

BND(I,J) = 0 interior point

= 1 boundary point

= 2 exterior point

A(I,J) - spacing between nodes at (I.J)

A(I+1,J) = distance from A(I,J) to A(I+1,J)

A(I,J+1) = distance from A(I,J) to A(I,J+1)

.

.

etc.

In the program for which ISOPLT was developed, both A(I,J) and BND(I,J) were calculated from the intersection of a physical boundary description and a mesh grid. In general, however, the arrays A(I,J) and BND(I,J) may be initialized as shown in the sample MAIN program in Appendix A.

APPENDIX A

```
@I FOR MAIN
      DIMENSION TA(61,26), BND(61,26), A(61,26)
      INTEGER BND
      COMMON /D1/ CALCMP, TYPE
      DATA CALCMP/'CALCMP'/
      DATA TYPE/'CALCMP'/

      M = 61
      N = 26
      CALL IDPLOT
      CALL PLOT(3.0,6.0,-3)

      DO 1 I=1,M
      DO 1 J=1,N
      X = 8.0*FLOAT(I-1)/FLOAT(M-1)
      TA(I,J) = SIN(X + 0.25*FLOAT(J-1)) + 1.
      A(I,J) = 1.0
    1 BND(I,J) = 0

      DO 2 I=1,M
      BND(I,1) = 1
    2 BND(I,N) = 1

      DO 3 J=1,N
      BND(1,J) = 1
    3 BND(M,J) = 1

      XP = 5.0
      YP = 2.5

      CALL ISOPLT(M,N,XP,YP,TA,BND,A)
      CALL PLOT(9.0,-3.0,-3)
      CALL FINI
      CALL EXIT
      END
```

```
@1 FO.  SOPLT
      SUBROUTINE ISOPLT(M,N,XP,YP,TA,BND,A)
      DIMENSION BND (61,26), A(61,26)
      DIMENSION TA(61,26), U(75), V(75), T(75), S(75), TEM(75)
      DIMENSION UA(75), VA(75)
      REAL M1, M2
      LOGICAL DOWN, FIRST
      DIMENSION FL(3000)
      COMMON/PLI/ SCALE, FL
      COMMON /D1/ CALCMP, TYPE
      INTEGER  BND, FL, TYPE, CALCMP

      M1 = N + 1
      N2 = N + 2
      DY =  80*YP/FLOAT(N-1)
      DX = DY
      X = XP + DX
      ANG = 0 86602540
      ER = 0 2*ANG*DX
      ZSCALE = 0 0
      DO 1 J=1,N
      DO 1 I=1,M
      IF(BND(I,J).EQ 2) GO TO 1
      ZSCALE = AMAX1(ZSCALE, ABS(TA(I,J)))
    1 CONTINUE

      DO 10 II=1,M
      I = M + 1 - II

      X = X - DX
      Y = -DY
      U(1) = ANG*X
      V(1) = - 5*X
      UA(1) = 0.0
      VA(1) = 0.0
      JJ =1

      NN = N - 1
      DO 2 J=1,NN
      JJ = JJ + 1
      UA(JJ) = 0.0
      VA(JJ) = 0 0
      Y = Y + DY
      U(JJ) = ANG*(X + Y)
      IF(BND(I,J).EQ.0) GO TO 31
      IF(BND(I,J).EQ.1) GO TO 30
      V(JJ) = .5*(Y-X)
      IF(J EQ.N) GO TO 2
      IF(BND(I,J+1).EQ.2) GO TO 2
      UA(JJ) = ANG*DY*(1.-A(I,J+1))
      VA(JJ) = .5*DY*(1.-A(I,J+1))
      JJ = JJ + 1
      U(JJ) = U(JJ-1)
      V(JJ) = .5*(Y-X) + TA(I,J+1)/ZSCALE
      UA(JJ) = UA(JJ-1)
      VA(JJ) = VA(JJ-1)
      GO TO 2
   31 V(JJ) =  5*(Y-X) + TA(I,J+1)/ZSCALE
```

```
      UA(JJ) = ANG*DY*(1.-A(I,J+1))
      VA(JJ) =  5*DY*(1 -A(I,J+1))
      IF(J EQ.NN) GO TO 22
      IF(BND(I,J+1) EQ 0) GO TO 2
      IF(J+2 GT.N) GO TO 2
      IF(BND(I,J+2) LT.2) GO TO 2
   22 JJ = JJ + 1
      U(JJ) = U(JJ-1)
      V(JJ) =  5*(Y-X)
      UA(JJ) = UA(JJ-1)
      VA(JJ) = VA(JJ-1)
      GO TO 2
   30 IF(BND(I,J+1) EQ 2) IA(I,J+1) = 0.0
      V(JJ) =  5*(Y-X) + IA(I,J+1)/ZSCALE
      UA(JJ) = ANG*DY*(1.-A(I,J+1))
      VA(JJ) =  5*DY*(1.-A(I,J+1))
      IF(J.LT.NN) GO TO 2
      JJ = JJ + 1
      U(JJ) = U(JJ-1)
      V(JJ) =  5*(Y-X)
    2 CONTINUE
      DOWN =  TRUE

      IF(II GT.1) GO TO 19
C      PLOT FIRST LINE WITHOUT HIDDEN LINE ALGORITHM...
      CALL PLOT1(U(1), V(1), 3)
      DO 21 J=2,JJ
   21 CALL PLOT1(U(J)-UA(J), V(J)-VA(J), 2)
      DO 23 K=2,JJ
   23 T(K) = -10.0
      GO TO 33

   19 CALL PLOT1(U(1), V(1), 3)
      DO 20 K=2,3
   20 CALL PLOT1(U(K)-UA(K), V(K)-VA(K), 2)
      S(1) = U(1)
      T(1) = V(1)
      FIRST =  FALSE.
      IF(V(4).LT.T(3)) FIRST = .TRUE.

      DO 8 K=4,JJ
      DO 43 KK=1,N1
      IF(ABS(S(KK)-U(K)) GT.ER) GO TO 43
      IF(V(K)-T(KK)) 50, 12, 12
   50 IF(DOWN) GO TO 14
      CALL PLOT1(U(K), V(K), 3)
      DOWN =  FALSE
      GO TO 8
   43 CONTINUE
      PRINT 44, K
   44 FORMAT(1HO, 30X, 8HHELP... , 2HL=I3)
      RETURN

   12 FIRST =  FALSE.
   13 IF(.NOT  DOWN) GO TO 14
      CALL PLOT1(U(K)-UA(K), V(K)-VA(K), 2)
      GO TO 8

   14 M1 = (T(KK) - T(KK-1))/(S(KK) - S(KK-1))
```

```
      IF(ABS(U(K)-U(K-1)).GT.ER) GO TO 4
      SS = U(K)
      GO TO 5
    4 M2 = (V(K) - V(K-1))/(U(K) - U(K-1))
      SS = (M2*U(K-1) - M1*S(KK-1 + T(KK-1) - V(K-1))/(M2 - M1)
    5 TT = M1*(SS - S(KK-1)) + T(KK-1)
      IF(S(KK-1)-ER.GT.SS .OR. SS.GT.S(KK)+ER) GO TO 16
      IF(DOWN) GO TO 27
      CALL PLOT1(SS, TT, 3)
      CALL PLOT1(U(K)-UA(K), V(K)-VA(K), 2)
      DOWN = TRUE.
      GO TO 8
   27 CALL PLOT1(SS-UA(K), TT, 2)
   16 CALL PLOT1(U(K)-UA(K), V(K)-VA(K), 3)
      DOWN = FALSE.

    8 CONTINUE

   33 S(2) = U(1)
      DO 40 K=2,N
      DO 45 J=K,JJ
      IF(U(J) GT.S(K)+ER) GO TO 40
   45 CONTINUE
   40 S(K+1) = U(J)
C        STORE THE MAXIMUM VALUE OF THE V(K) LINE IN ARRAY T(K)
      T(1) = V(1)
      DO 41 K=2,N1
      TEM(K) = -10 0
      DO 42 KK=K,JJ
      IF(U(KK)-S(K)) 42, 51, 42
   51 TEM(K) = AMAX1(V(KK), T(K-1), TEM(K))
   42 CONTINUE
   41 CONTINUE
      DO 11 K=2,N1
   11 T(K) = TEM(K)
    6 CONTINUE
   10 CONTINUE

      IF(TYPE.NE.CALCMP) CALL SNDFLE(FL)

      RETURN

      SUBROUTINE PLOT1(A,B,KK)
      DIMENSION FL(3000)
      COMMON /D1/ CALCMP, TYPE
      COMMON/PLT/ SCALE, FL
      COMMON/PLT2/ ISW, IX1, IY1
      INTEGER TYPE, CALCMP, FL

      IF(TYPE NE.CALCMP) GO TO 1

      CALL PLOT(A,B,KK)
      GO TO 6

    1 IF(KK.LT 0) GO TO 6
      IX = 70.0*A + 300.0
      IY = 70 0*B = 600.0
      GO TO (2,2,3), KK
    2 IF(KK.EQ ISW) GO TO 7
```

```
      CALL INITYPE(0)
      CALL LINE(FL,IX1,IY1,IA)
      CALL INITYPE(3)
      ISW = 2
      CALL LINE(FL,IX,IY,IA)
      GO TO 6
    5 ISW = 2
      IX1 = IX
      IY1 = IY

    6 RETURN
      END
```

APPENDIX B