NASA TECHNICAL MEMORANDUM

NASA TM X-73679

NASA TM X-73679

```
(NASA-TM-X-73679)  FORTRAN PROGRAM FOR                     N77-27069
CALCULATING LEADING AND TRAILING-EDGE
GEOMETRY OF TURBOMACHINE BLADES (NASA)   22 p
HC A02/MF A01                          CSCL 01A            Unclas
                                                   G3/02   36702
```

FORTRAN PROGRAM FOR CALCULATING LEADING- AND TRAILING-EDGE GEOMETRY OF TURBOMACHINE BLADES

by Lawrence F. Schumann
Lewis Research Center and
U.S. Army Air Mobility R&D Laboratory
Cleveland, Ohio 44135
June 1977

1. Report No.
NASA TM X-73679

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
FORTRAN PROGRAM FOR CALCULATING LEADING- AND TRAILING-EDGE GEOMETRY OF TURBOMACHINE BLADES

5. Report Date

6. Performing Organization Code

7. Author(s)
Lawrence F. Schumann

8. Performing Organization Report No.
E-9211

9. Performing Organization Name and Address
NASA Lewis Research Center and
U.S. Army Air Mobility R&D Laboratory
Cleveland, Ohio 44135

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A FORTRAN IV program has been written which calculates leading- and trailing-edge circle radii, tangency angles on the leading- and trailing-edge circles, and stagger angles of turbomachinery blade sections using only spline points defining the blade surfaces. The program also shifts the origin of the blade coordinates to the leading edge of the blade. Required input includes (m, θ) coordinates of a sufficient number of spline points to adequately define the two surfaces of the blade. Other required input are the radii from the axis of rotation of the leading- and trailing-edges. The output from this program may be used directly as the geometrical input for a NASA developed program for calculating transonic velocities on a blade-to-blade stream surface of a turbomachine (TSONIC). The program may be used for axial, radial, and mixed flow turbomachine blades.

17. Key Words (Suggested by Author(s))
Turbomachine blades
Blade coordinates

18. Distribution Statement
Unclassified - unlimited
STAR Category 02

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages

22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

FORTRAN PROGRAM FOR CALCULATING LEADING- AND TRAILING-EDGE GEOMETRY OF TURBOMACHINE BLADES

by Lawrence F. Schumann

Lewis Research Center and
U.S. Army Air Mobility R&D Laboratory

SUMMARY

A FORTRAN IV program has been written which calculates leading- and trailing- edge circle radii, tangency angles on the leading- and trailing- edge circles, and stagger angle of turbomachinery blade sections using only spline points defining the blade surfaces. The program also shifts the origin of the blade coordinates to the leading edge of the blade. Required input includes (m, θ) coordinates of a sufficient number of spline points to adequately define the two surfaces of the blade. Other required input are the radii from the axis of rotation of the leading- and trailing- edges. The output from this program may be used directly as the geometrical input for a NASA developed program for calculating transonic velocities on a blade-to-blade stream surface of a turbomachine (TSONIC). The program may be used for axial, radial, and mixed flow turbomachine blades.

INTRODUCTION

In recent years, several computer programs have been written which calculate the velocities on a blade-to-blade surface of a turbomachine (ref. 1, 2, 3). These programs require an accurate geometrical description of the blade since spline fit curves (ref. 4) are used to define the blade surfaces. The inputs required include blade chord, blade stagger, leading- and trailing-edge circle radii, angles of tangency on the leading- and trailing-edge circles, and intermediate spline points. These inputs are usually obtained from measurements made on a physical layout of the blade or from a meridional flow analysis program such as MERIDL (ref. 5, 6). In either case, it is difficult to specify these inputs with the degree of accuracy necessary to insure a smooth blade shape without using a costly and time consuming trial-and-error procedure with the computer. Small errors in any of the input can result in a spline fit through the given input spline points in which the first and second derivatives of the curve do not vary smoothly. Since the computation of surface velocities is highly dependent on the blade surface curvatures which in turn are determined from the values of $d\theta/dm$ and $d^2\theta/dm^2$ obtained from the spline fit curve, it is important that the geometry input be specified accurately. If the first and second derivatives are not smooth unrealistic peaks and valleys in the surface velocity distribution will occur.

This report presents a Fortran IV computer program which analytically calculates the leading- and trailing-edge circle radii, tangency angles, and stagger angle of a turbomachine blade section using an iterative procedure. The required input includes enough blade surface (m, θ) coordinates to adequately define the general shape of the blade surfaces and the radius from the axis of rotation of the leading- and trailing-edges. This input is much easier to specify than the input for TSONIC. It will also yield smoother blade shapes. The output from this program may be used directly as the geometrical input for TSONIC (ref. 1) and TANDEM (ref. 3). A complete description of the input required and output obtained are given. Also given are two example cases and the program listing.

SYMBOLS

c	meridional chord length, meters, see fig. 1(b)
m	meridional distance, meters, see fig. 1
r_L	radius of leading edge circle, meters, see fig. 1(b)
r_T	radius of trailing edge circle, meters, see fig. 1(b)
R	radius from axis of rotation to the blade surface, meters, see fig. 1
β	angle between the blade surface and the meridional direction, degrees, see fig. 1(b)
δ	angular distance between the origin of the input blade coordinates and the center of the leading edge circle, radians, see fig. 1(b)
θ	angular coordinate about the axis of rotation, radians, see fig. 1
σ	blade stagger angle from the center of the leading edge circle to the center of the trailing edge circle, radians, see fig. 1(b)

Subscripts:

L	leading edge
LT	leading edge tangency point
T	trailing edge
TT	trailing edge tangency point
1	surface 1
2	surface 2

DESCRIPTION OF INPUT AND OUTPUT

The program computes leading- and trailing- edge circle radii, tangency angles, blade stagger and blade chord from input blade surface coordinates. The blade surface coordinates are defined by meridional distance, m, and the angle about the axis of rotation, θ. Also required as input are the radii from the axis of rotation of the leading- and trailing-edges. The input blade geometry is shown in figure 1(a). Since spline fit curves are used to define the two blade surfaces, the input spline points should not reflect the leading- and trailing- edge circle curvatures but should define a smooth blade with the surfaces extrapolated to the leading and trailing edges as shown in figure 1(a). This is the type of blade definition obtained from a known mean camber line and tangential thickness distribution.

The general procedure for the solution of the problem is described in the appendix. An iterative procedure is used which fits a circle in the region bounded by the two blade surfaces and the leading or trailing edge (fig. 1). The requirement that the circle be tangent to all three boundaries leads to a unique solution. Once the radius of this circle is determined, the remaining geometry is easily calculated. The variables calculated by the program are shown in fignure 1(b).

INPUT

The input data sheet along with the values used in an example case are shown in figure 2. NP1 and NP2 are integer variables and must be right adjusted on 5-column format fields. The remaining input are real variables and are specified in 10-column fields. Either English or SI units may be used. The input variables are as follows:

NP1, NP2	Number of spline points on surfaces 1 and 2 respectively (maximum 50).
M1, M2	Arrays of m-coordinates of spline points on the two blade surfaces, meters, see fig. 1(a). The origin of the m coordinates may be anywhere but must be the same for both surfaces.
THETA1, THETA2	Arrays of θ-coordinates of spline points corresponding to M1 and M2, radians, see fig. 1(a). The origin of the θ-coordinates may be anywhere but must be the same for both surfaces.
ML, MT	Meridional distance, m, of the leading and trailing edges respectively, meters, see fig. 1(a). The origin for ML and MT must be the same as for M1 and M2.

RL, RT	Radius of blade section from the axis of rotation at the leading and trailing edges corresponding to ML and MT, meters.

Surface 1 must always be the upper surface, i.e., the surface with the larger numerical values of θ as shown in figure 1(a). If this is not the case, an error message will be printed and no further computation will be done. Also, the leading- and trailing edge thicknesses must not equal zero for this technique.

It is advisable, if possible, to specify spline points beyond the leading and trailing edges of the blade as shown in figure 1(a) since this reduces the effect of the end conditions used in the splint subroutine (i.e. that the second derivative at the end points is one-half of that at the adjacent point). However, this is not an absolute requirement since the splint subroutine has the capability of linear extrapolation and therefore the specified spline points do not have to extend up to or beyond the leading- and trailing-edges.

OUTPUT

Two examples of the output obtained from the program are shown in figure 3. The output is divided into three points. The first part is a listing of the input data. The second part is a listing of the variables computed by the program. They are:

RLE	r_L, meters, see figure 1(b)
BETAL1	β_{LT1}, degrees, see figure 1(b)
BETAL2	β_{LT2}, degrees, see figure 1(b)
RTE	r_T, meters, see figure 1(b)
BETAT1	β_{TT1}, degrees, see figure 1(b)
BETAT2	β_{TT2}, degrees, see figure 1(b)
CHORD	c, meters, see figure 1(b)
STGR	σ, radians, see figure 1(b)
DELTA	δ, radians, see figure 1(b)

The third part of the output is a list of the input blade coordinates with the origin shifted to the leading edge of the blade as shown in figure 1(b). Also shown are the calculated tangency points. The values of the first and second derivatives ($d\theta/dm$, $d^2\theta/dm^2$) obtained from the spline fit are also listed. A check on the smoothness of the blade can be made by examining the values of the second derivatives. They should vary smoothly from inlet to exit and have no wild oscillations. Any input coordinates whose m value is less than m_{LT} or greater than m_{TT} are not listed since these values should not be used as input to TSONIC. The values of the tangency points also should not be used as input coordinates to TSONIC (except as dummy variables) since these values will be calculated by TSONIC. The remaining coordinates may be used directly as input to TSONIC with no further modification.

APPLICATION OF THE PROGRAM

The primary purpose of this program is to easily obtain the input for TSONIC or TANDEM when only the blade surface coordinates are known or can easily be obtained (e.g. from a known mean camber line and tangential thickness distribution - fig. 4(a)). This is the type of output obtained from a meridional flow analysis program such as MERIDL. The current version of MERIDL (ref. 5 and 6) employs the technique presented in this report for the calculation of TSONIC input data. This program can also be used in conjunction with a physical layout since it is often difficult to measure angles and leading and trailing edge radii with any degree of accuracy while it is a relatively easy task to obtain 5 or 6 spline points on the blade surfaces which will yield a smooth blade shape (fig. 4(b)). It is also easy to obtain splitter blade geometry using this program by simply changing the value of m_L and R_L as is shown in example 2 (figs. 3(b) and 4(c)).

PROGRAM LISTING

```
      DIMENSION M1(50),M2(50),THETA1(50),THETA2(50),DTDM1(50),DTDM2(50),-
     1D2TDM1(50),D2TDM2(50)
      REAL M1,M2,ML,MT,MTL1,MTL2,MTT1,MTT2
C
C   READ AND WRITE INPUT DATA
C
   10 READ(5,1000,END=999)
      WRITE(6,1001)
      WRITE(6,1000)
      WRITE(6,1100)
      READ(5,1010)NP1,NP2
      WRITE(6,1010)NP1,NP2
      WRITE(6,1110)
      READ(5,1020) (M1(I),I=1,NP1)
      WRITE(6,1030) (M1(I),I=1,NP1)
      WRITE(6,1120)
      READ(5,1020) (THETA1(I),I=1,NP1)
      WRITE(6,1030) (THETA1(I),I=1,NP1)
      WRITE(6,1130)
      READ(5,1020) (M2(I),I=1,NP2)
      WRITE(6,1030) (M2(I),I=1,NP2)
      WRITE(6,1140)
      READ(5,1020) (THETA2(I),I=1,NP2)
      WRITE(6,1030) (THETA2(I),I=1,NP2)
      WRITE(6,1150)
      READ(5,1020)ML,MT,RL,RT
      WRITE(6,1030)ML,MT,RL,RT
C
C   COMPUTE LEADING EDGE RADIUS AND TANGENCY ANGLES
C
      CALL SPLINT(M1,THETA1,NP1,ML,1,THL1,DL1,D2L1)
      CALL SPLENT(M1,NP1,THETA1,DTDM1,D2TDM1)
      CALL SPLINT(M2,THETA2,NP2,ML,1,THL2,DL2,D2L2)
      CALL SPLENT(M2,NP2,THETA2,DTDM2,D2TDM2)
      DAMP=1.
      ITER=0
   20 BETAL1=ATAN(RL*DL1)
      BETAL2=ATAN(RL*DL2)
      RLE1=(THL1-THL2)*RL*COS((BETAL1+BETAL2)/2.)/2.
      IF(RLE1.LE.0.)GO TO 120
   30 MTL1=ML+RLE1*(1.-SIN(BETAL1))
      MTL2=ML+RLE1*(1.+SIN(BETAL2))
      CALL SPLINT(M1,THETA1,NP1,MTL1,1,TTL1,DTL1,D2TL1)
      CALL SPLINT(M2,THETA2,NP2,MTL2,1,TTL2,DTL2,D2TL2)
      BETAL1=ATAN(RL*DTL1)
      BETAL2=ATAN(RL*DTL2)
      RLE2=RL*(TTL1-TTL2)/(COS(BETAL1)+COS(BETAL2))
      IF(ABS((RLE2-RLE1)/RLE1).LT..0001)GO TO 50
      ITER=ITER+1
      IF(ITER.LE.100)GO TO 40
      WRITE(6,1200)
      GO TO 50
   40 RLE1=(DAMP*RLE1+RLE2)/(DAMP+1.)
      IF(RLE1.GT.0.)GO TO 30
```

```
      DAMP=DAMP+20.
      GO TO 20
C
C   COMPUTE TRAILING EDGE RADIUS AND TANGENCY ANGLES
C
   50 CALL SPLINT(M1,THETA1,NP1,MT,1,THT1,DT1,D2T1)
      CALL SPLINT(M2,THETA2,NP2,MT,1,THT2,DT2,D2T2)
      DAMP=1.
      ITER=0
   60 BETAT1=ATAN(RT*DT1)
      BETAT2=ATAN(RT*DT2)
      RTE1=(THT1-THT2)*RT*COS((BETAT1+BETAT2)/2.)/2.
      IF(RTE1.LE.0.)GO TO 120
   70 MTT1=MT-RTE1*(1.+SIN(BETAT1))
      MTT2=MT-RTE1*(1.-SIN(BETAT2))
      CALL SPLINT(M1,THETA1,NP1,MTT1,1,TTT1,DTT1,D2TT1)
      CALL SPLINT(M2,THETA2,NP2,MTT2,1,TTT2,DTT2,D2TT2)
      BETAT1=ATAN(RT*DTT1)
      BETAT2=ATAN(RT*DTT2)
      RTE2=RT*(TTT1-TTT2)/(COS(BETAT1)+COS(BETAT2))
      IF(ABS((RTE2-RTE1)/RTE1).LT..0001)GO TO 90
      ITER=ITER+1
      IF(ITER.LE.100)GO TO 80
      WRITE(6,1210)
      GO TO 90
   80 RTE1=(DAMP*RTE1+RTE2)/(DAMP+1.)
      IF(RTE1.GT.0.)GO TO 70
      DAMP=DAMP+20.
      GO TO 60
C
C   CALCULATE AND PRINT OUTPUT DATA
C
   90 DELTA=(TTL1*COS(BETAL2)+TTL2*COS(BETAL1))/(COS(BETAL1)+COS(BETAL2)-
     1)
      STGR=(TTT1*COS(BETAT2)+TTT2*COS(BETAT1))/(COS(BETAT1)+COS(BETAT2))-
     1-DELTA
      CHORD=MT-ML
      BETAL1=BETAL1*57.29578
      BETAL2=BETAL2*57.29578
      BETAT1=BETAT1*57.29578
      BETAT2=BETAT2*57.29578
      MTL1=MTL1-ML
      MTL2=MTL2-ML
      MTT1=MTT1-ML
      MTT2=MTT2-ML
      TTL1=TTL1-DELTA
      TTL2=TTL2-DELTA
      TTT1=TTT1-DELTA
      TTT2=TTT2-DELTA
      WRITE(6,1160)
      WRITE(6,1030)RLE1,BETAL1,BETAL2,RTE1,BETAT1,BETAT2
      WRITE(6,1170)
      WRITE(6,1030)CHORD,STGR,DELTA
      WRITE(6,1180)
```

```
      J=1
      WRITE(6,1040)J,MTL1,TTL1,DTL1,D2TL1
      DO 100 I=1,NP1
      M1(I)=M1(I)-ML
      THETA1(I)=THETA1(I)-DELTA
      IF(M1(I).LE.MTL1.OR.M1(I).GE.MTT1)GO TO 100
      J=J+1
      WRITE(6,1050)J,M1(I),THETA1(I),DTDM1(I),D2TDM1(I)
  100 CONTINUE
      J=J+1
      WRITE(6,1040)J,MTT1,TTT1,DTT1,D2TT1
      WRITE(6,1190)
      J=1
      WRITE(6,1040)J,MTL2,TTL2,DTL2,D2TL2
      DO 110 I=1,NP2
      M2(I)=M2(I)-ML
      THETA2(I)=THETA2(I)-DELTA
      IF(M2(I).LE.MTL2.OR.M2(I).GE.MTT2)GO TO 110
      J=J+1
      WRITE(6,1050)J,M2(I),THETA2(I),DTDM2(I),D2TDM2(I)
  110 CONTINUE
      J=J+1
      WRITE(6,1040)J,MTT2,TTT2,DTT2,D2TT2
      GO TO 10
  120 WRITE(6,1220)
      GO TO 10
C
C    FORMAT STATEMENTS
C
 1000 FORMAT(80H                                                              -
     1                    )
 1001 FORMAT(1H1)
 1010 FORMAT(16I5)
 1020 FORMAT(8F10.6)
 1030 FORMAT(8G15.6)
 1040 FORMAT(6X,14HTANGENCY POINT,3X,I2,4G15.6)
 1050 FORMAT(23X,I2,4G15.6)
 1100 FORMAT(/10H  NP1  NP2)
 1110 FORMAT(8X,8HM1 ARRAY)
 1120 FORMAT(8X,12HTHETA1 ARRAY)
 1130 FORMAT(8X,8HM2 ARRAY)
 1140 FORMAT(8X,12HTHETA2 ARRAY)
 1150 FORMAT(6X,2HML,13X,2HMT,13X,2HRL,13X,2HRT)
 1160 FORMAT(///32X,25H** COMPUTED BLADE DATA **///6X,3HRLE,12X,           -
     16HBETAL1,9X,6HBETAL2,9X,3HRTE,12X,6HBETAT1,9X,6HBETAT2)
 1170 FORMAT(6X,5HCHORD,10X,4HSTGR,11X,5HDELTA)
 1180 FORMAT(///34X,21H* BLADE COORDINATES */30X,30H(ORIGIN AT BLADE LEA-
     1DING EDGE)//40X,9HSURFACE 1//24X,1HI,6X,1HM,14X,5HTHETA,10X,4HDTDM-
     1,11X,6HD2TDM2)
 1190 FORMAT(//40X,9HSURFACE 2//24X,1HI,6X,1HM,14X,5HTHETA,10X,4HDTDM,  -
     111X,6HD2TDM2)
 1200 FORMAT(//5X,61HSOLUTION FOR LEADING EDGE HAS NOT CONVERGED IN 100 -
     1ITERATIONS)
 1210 FORMAT(//5X,62HSOLUTION FOR TRAILING EDGE HAS NOT CONVERGED IN 100-
```

```
    1 ITERATICNS)
1220 FORMAT(///10X,43HINPUT ERROR -- ONE OF THREE POSSIBLE CAUSES/      -
    117X,32H1. SURFACES 1 AND 2 ARE INVERTED/                           -
    217X,54H2. BLADE THICKNESS IS ZERO AT LEADING OR TRAILING EDGE/     -
    317X,68H3. SURFACES 1 AND 2 CROSS OVER AT SOME POINT DUE TO A BAD S-
    4PLINE FIT)
 999 STOP
     END
```

```
      SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT,DYDX,D2YDX2)
C
C--SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
C--FOR A SPLINE CURVE
C--END CONDITION - SECOND DERIVATIVES AT END POINTS ARE
C--SDR1 AND SDRN TIMES SECOND DERIVATIVES AT ADJACENT POINTS
C
      DIMENSION X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX),D2YDX2(MAX)
      DIMENSION G(101),SB(101),EM(101)
      IERR = 0
      SDR1 = .5
      SDRN = .5
      TOLER= ABS(X(N)-X(1))/FLOAT(N)*1.E-5
      C = X(2)-X(1)
      IF (C.EQ.0.) GO TO 130
      SB(1) = -SDR1
      G(1) = 0.
      NO = N-1
      IF (NO.LE.0) GO TO 140
      IF (NO.EQ.1) GO TO 20
      DO 10 I=2,NO
      A = C
      C = X(I+1)-X(I)
      IF (A*C.EQ.0.) GO TO 130
      IF (A*C.LT.0.) IERR = 1
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/C-(Y(I)-Y(I-1))/A
   10 G(I) = (6.*F-A*G(I-1))/W
   20 EM(N) = SDRN*G(N-1)/(1.+SDRN*SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      IF (MAX.LE.0) RETURN
C
      ENTRY SPLENT (Z,MAX,YINT,DYDX,D2YDX2)
      DO 120 I=1,MAX
      K=2
      IF (ABS(Z(I)-X(1)).LT.TOLER) GO TO 40
      IF (Z(I).GT.2.0*X(1)-X(2)) GO TO 50
      GO TO 80
   40 YINT(I) = Y(1)
      SK = X(K)-X(K-1)
      GO TO 110
   50 IF (ABS(Z(I)-X(K)).LT.TOLER) GO TO 60
      IF (Z(I).GT.X(K)) GO TO 70
      GO TO 100
   60 YINT(I) = Y(K)
      SK = X(K)-X(K-1)
      GO TO 110
   70 IF (K.GE.N) GO TO 90
      K = K+1
      GO TO 50
   80 S2 = X(2)-X(1)
```

```
      Y0 = EM(1)*S2**2+2.*Y(1)-Y(2)
      DYDX(I) = (Y(2)-Y(1))/S2-7.*EM(1)/6.*S2
      YINT(I) = Y0+DYDX(I)*(Z(I)-X(1)+S2)
      D2YDX2(I) = 0.
      GO TO 120
   90 IF (Z(I).LT.2.*X(N)-X(N-1)) GO TO 100
      SN = X(N)-X(N-1)
      YNP1 = EM(N)*SN**2+2.*Y(N)-Y(N-1)
      DYDX(I) = (Y(N)-Y(N-1))/SN+7.*EM(N)/6.*SN
      YINT(I) = YNP1+DYDX(I)*(Z(I)-X(N)-SN)
      D2YDX2(I) = 0.
      GO TO 120
  100 SK = X(K)-X(K-1)
      YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6./SK   +EM(K)*(Z(I)-X(K-1))**3/6.-
     1  /SK+(Y(K)/SK   -EM(K)*SK   /6.)*(Z(I)-X(K-1))+(Y(K-1)/SK   -EM(K-1)-
     2  *SK/6.)*(X(K)-Z(I))
  110 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/SK   +EM(K)*(X(K-1)-Z(I))**2/2.-
     1   /SK+(Y(K)-Y(K-1))/SK   -(EM(K)-EM(K-1))*SK/6.
      D2YDX2(I) = EM(K)-(X(K)-Z(I))/SK*(EM(K)-EM(K-1))
  120 CONTINUE
      IF (IERR.EQ.0) RETURN
  130 WRITE(6,1000)
      WRITE(6,1020) N,(X(I),Y(I),I=1,N)
      IF (IERR.EQ.0) STOP
      WRITE(6,1030)
      RETURN
  140 WRITE(6,1010)
      WRITE(6,1020) N,(X(I),Y(I),I=1,N)
      STOP
 1000 FORMAT (1H1,10X,44HSPLINT ERROR -- ONE OF THREE POSSIBLE CAUSES/  -
     117X,51H1.  ADJACENT X POINTS ARE DUPLICATES OF EACH OTHER./       -
     217X,38H2.  SOME X POINTS ARE OUT OF SEQUENCE./                    -
     317X,32H3.  SOME X POINTS ARE UNDEFINED.)
 1010 FORMAT (1H1,10X,62HSPLINT ERROR -- NUMBER OF SPLINE POINTS GIVEN I-
     1S LESS THAN TWO)
 1020 FORMAT (//17X,18HNUMBER OF POINTS =,I4//17X,8HX  ARRAY,6X,8HY  ARR-
     1AY/(17X,2G13.5))
 1030 FORMAT (1H1)
      END
```

APPENDIX

CALCULATION PROCEDURE OF THE PROGRAM

The program computes the leading- and trailing-edge circle radii and tangency angles by an iterative procedure. Figure 1 shows the basic blade geometry. The input coordinates are given as (m,θ) coordinates where m is the length in the meridional direction and θ is the angle in radians about the axis of rotation. The origin for both the m and θ coordinates may be anywhere since it will be shifted by the program to the leading edge of the blade.

From figure 1(b) (with the assumption that $r_L << R_L$), the following equations can be written for the leading edge circle:

$$m_{LT1} = m_L + r_L\,(1 - \sin\beta_{LT1}) \tag{1}$$

$$m_{LT2} = m_L + r_L\,(1 + \sin\beta_{LT2}) \tag{2}$$

$$\theta_{LT1} = \delta + \frac{r_L \cos\beta_{LT1}}{R_L} \tag{3}$$

$$\theta_{LT2} = \delta - \frac{r_L \cos\beta_{LT2}}{R_L} \tag{4}$$

Equations (3) and (4) can be solved for r_L which gives

$$r_L = \frac{R_L\,(\theta_{LT1} - \theta_{LT2})}{\cos\beta_{LT1} + \cos\beta_{LT2}} \tag{5}$$

The tangency angles are determined from the equations

$$\tan\beta_{LT1} = R_L \left(\frac{d\theta}{dm}\right)_{LT1} \tag{6}$$

$$\tan\beta_{LT2} = R_L \left(\frac{d\theta}{dm}\right)_{LT2} \tag{7}$$

The angle θ and $d\theta/dm$ for a given value of m are obtained from a spline fit curve (subroutine SPLINT, ref. 4) on the input surface coordinates. Equations (1), (2), (5), (6) and (7) along with the spline fit curves on the surfaces are sufficient to solve the problem. Similar equations can be written for the trailing edge.

The program procedure is as follows:

(1) A first estimate of the leading edge radius is obtained from the equation

$$r_L \approx R_2 \, (\theta_{L1} - \theta_{L2}) \cos\left(\frac{\beta_{L1} + \beta_{L2}}{2}\right) \tag{8}$$

where θ_L and β_L are values obtained from a spline fit curve at the leading edge (see fig. 1(a)).

(2) Values of m_{LT1} and m_{LT2} are obtained from equation (1) and (2) using the estimated value of r_L, the leading edge values of β_L, and the input value of m_L.

(3) Splint calls are done on both surfaces using the computed values m_{LT1} and m_{LT2} which will yield values of θ_{LT} and β_{LT}.

(4) A new value of leading edge radius is obtained from equation (5).

(5) An average value of r_L is used to start the next iteration. The average value will be weighted toward the previous value if r_L becomes negative. The values of β_{LT} from the previous iteration are used.

(6) When the values of r_L for two successive iterations are within 0.01% of each other, the solution is converged.

(7) The solution for the trailing edge is obtained.

(8) Values of δ, stagger (σ), and chord (c) are obtained from the equations

$$\delta = \frac{\theta_{LT1} \cos \beta_{LT2} + \theta_{LT2} \cos \beta_{LT1}}{\cos \beta_{LT1} + \cos \beta_{LT2}} \tag{9}$$

$$\sigma = \frac{\theta_{TT1} \cos \beta_{TT2} + \theta_{TT2} \cos \beta_{TT1}}{\cos \beta_{TT1} + \cos \beta_{TT2}} - \delta \tag{10}$$

$$c = m_T - m_L \tag{11}$$

(9) The input blade coordinates are shifted so that the origin is at the blade leading edge by the following transformation

$$m = m - m_L \tag{12}$$

$$\theta = \theta - \delta \tag{13}$$

REFERENCES

1. Katsanis, Theodore: FORTRAN Program for Calculating Transonic Velocities on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-5427, 1969.

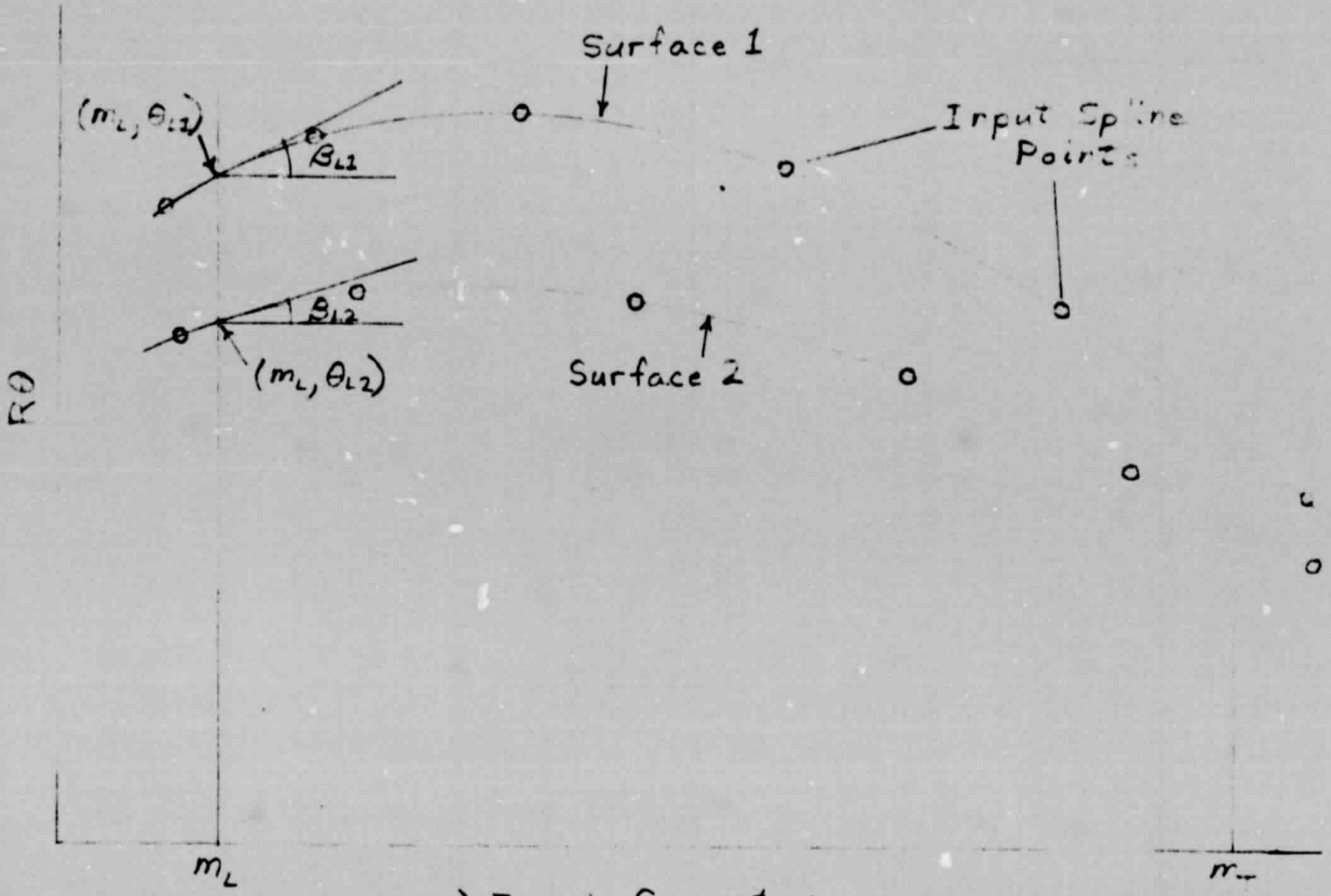
2. Katsanis, Theodore; and McNally, William D.: Revised FORTRAN Program for Calculating Velocities on a Blade to Blade Stream Surface of a Turbomachine. NASA TM X-1764, 1969.

3. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Tandem Blade Turbomachine. NASA TN D-5044, 1969.

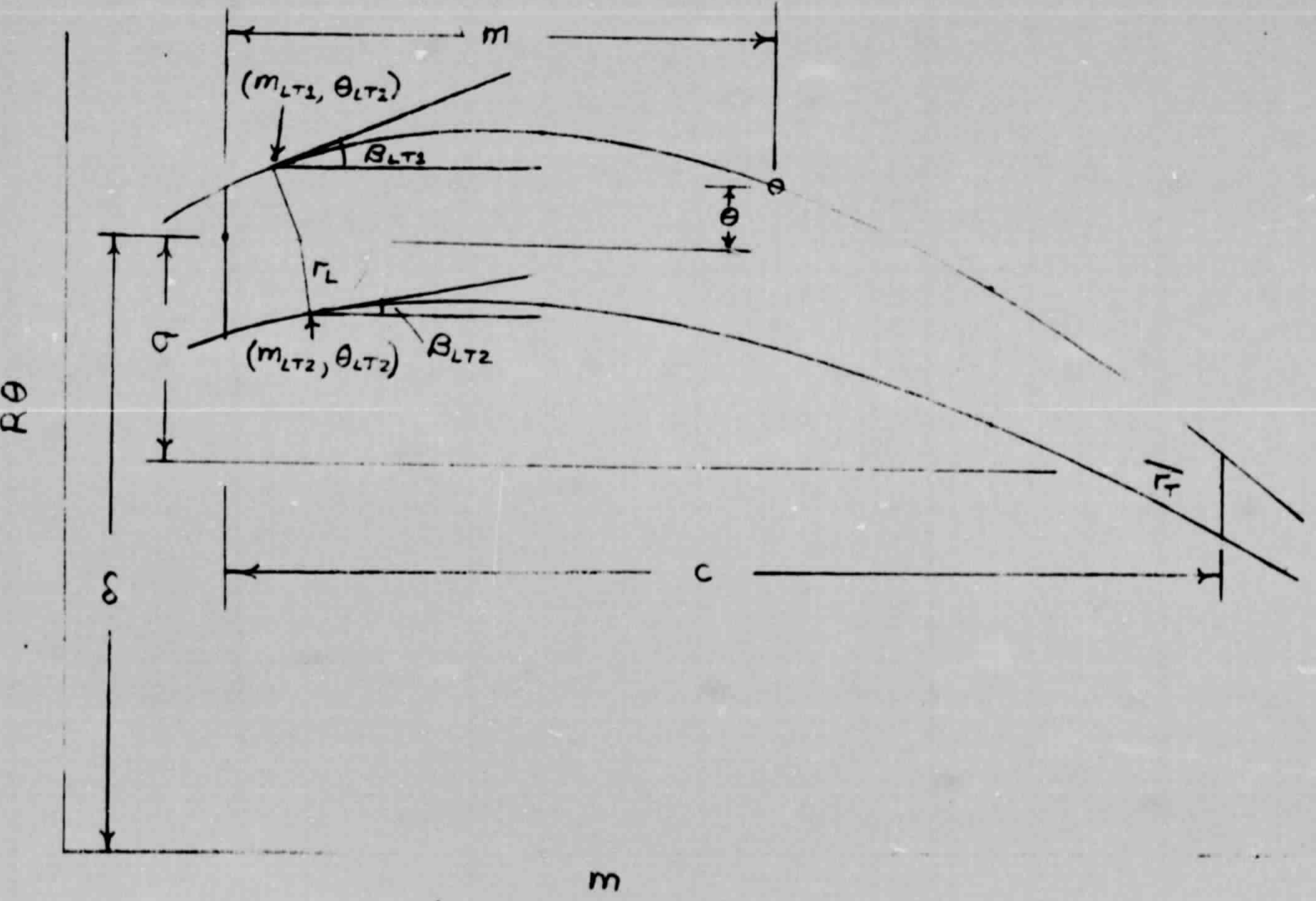
4. Katsanis, Theodore: FORTRAN Program for Spline Fit Curve. NASA TM X-1707, 1968.

5. Katsanis, Theodore; and McNally, William D.: Revised FORTRAN Program for Calculating Velocities and Streamlines on the Hub-Shroud Midchannel Stream Surface of an Axial-, Radial-, or Mixed-Flow Turbomachine or Annular Duct. I - User's Manual. NASA TN D-8430, 1977.

6. Katsanis, Theodore; and McNally, William D.: Revised FORTRAN Program for Calculating Velocities and Streamlines on the Hub-Shroud Midchannel Stream Surface of an Axial-, Radial-, or Mixed-Flow Turbomachine or Annular Duct. II - Programmer's Manual. NASA TN D-8431, 1977.

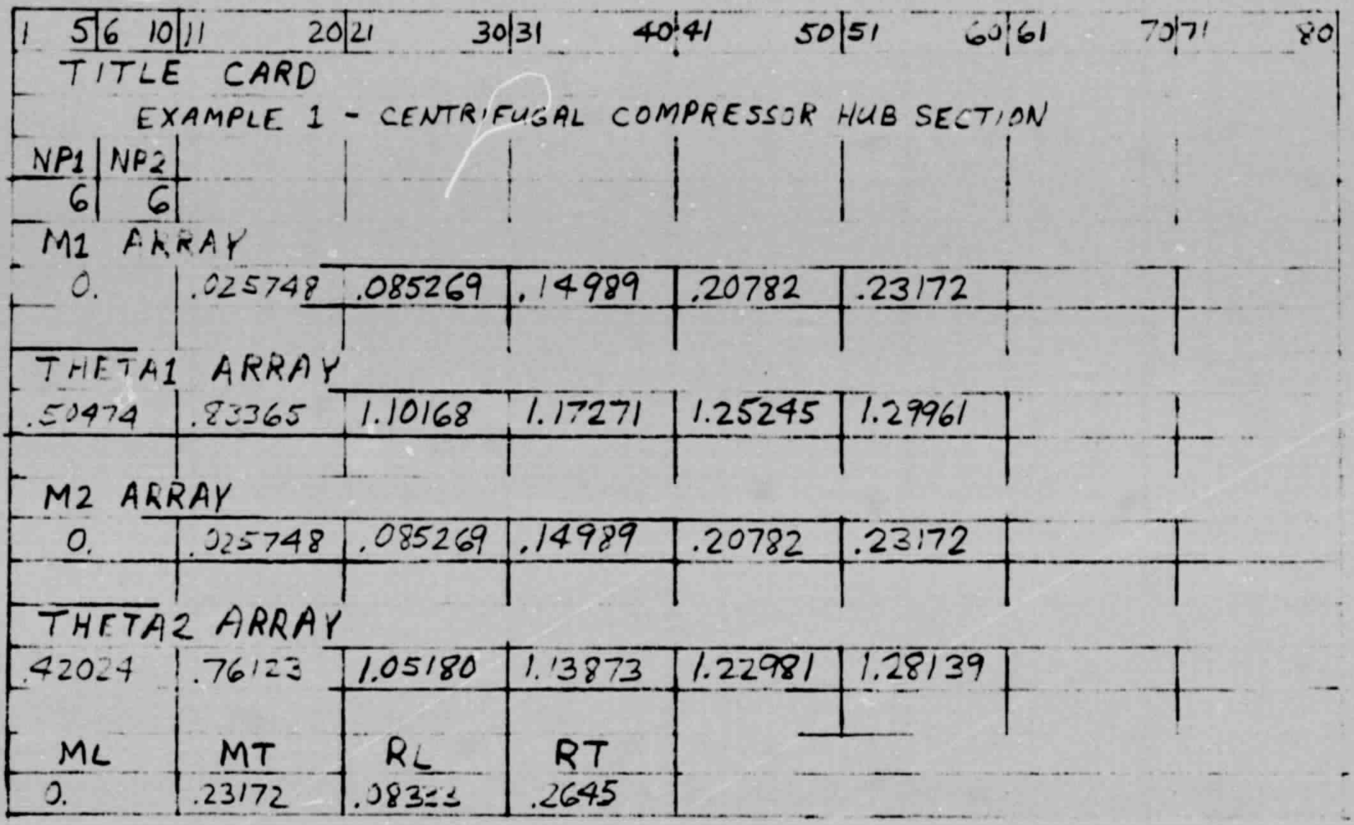


a) Input Geometry



b) Output Geometry

Figure 1 - Blade Geometry



1 5	6 10	11 20	21 30	31 40	41 50	51 60	61 70	71 80
TITLE CARD								
EXAMPLE 1 - CENTRIFUGAL COMPRESSOR HUB SECTION								
NP1	NP2							
6	6							
M1 ARRAY								
0.		.025748	.085269	.14989	.20782	.23172		
THETA1 ARRAY								
.50474		.83365	1.10168	1.17271	1.25245	1.29961		
M2 ARRAY								
0.		.025748	.085269	.14989	.20782	.23172		
THETA2 ARRAY								
.42024		.76123	1.05180	1.13873	1.22981	1.28139		
ML		MT	RL	RT				
0.		.23172	.08333	.2645				

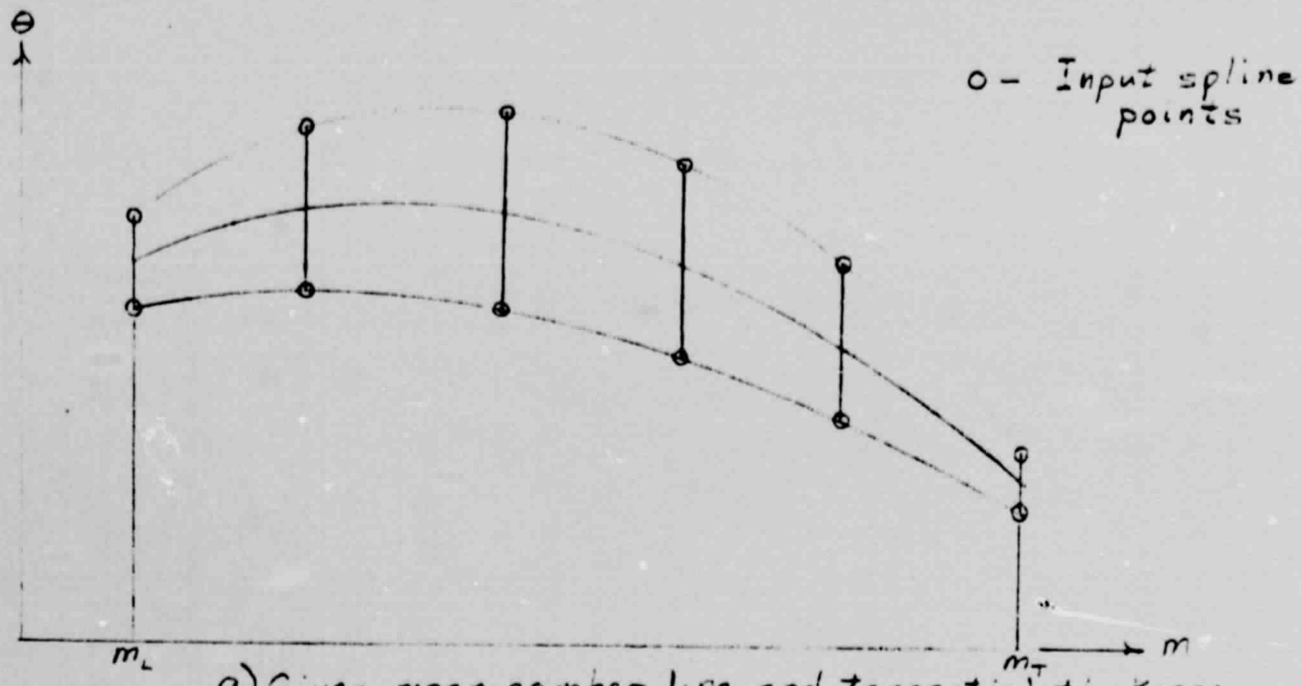
Figure 2 - Input Form

Figure 3(a) EXAMPLE OUTPUT

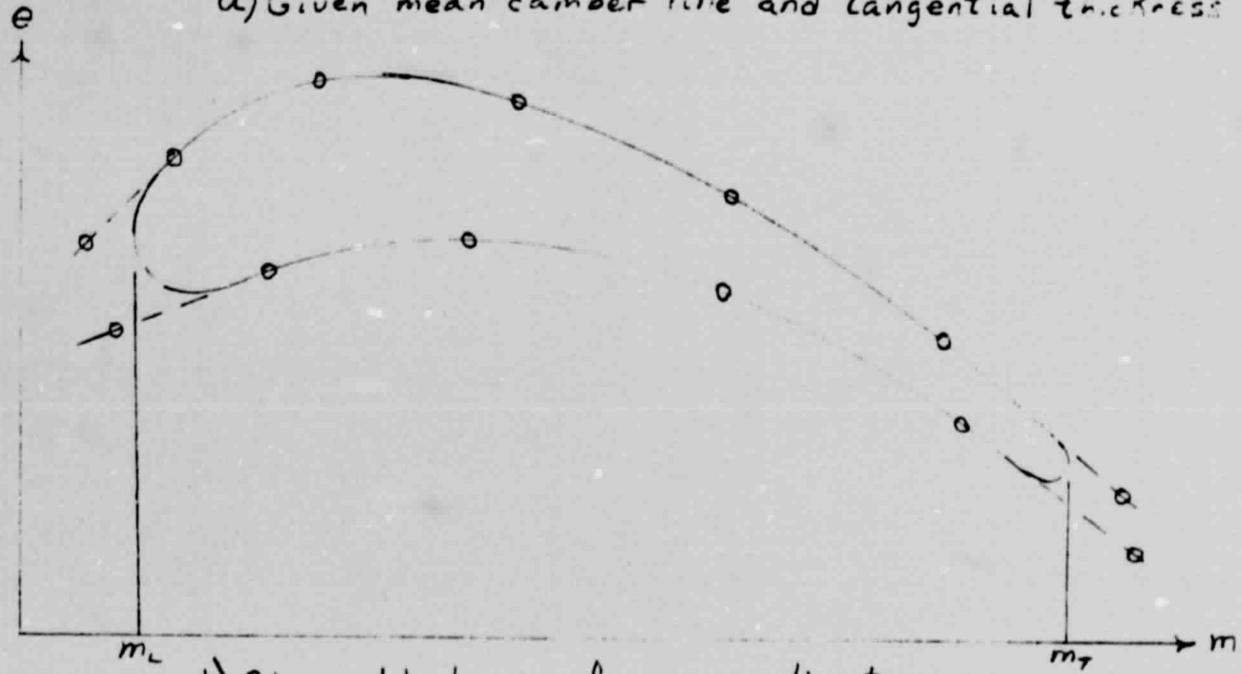
```
1
    EXAMPLE 1 - CENTRIFUGAL COMPRESSOR HUB SECTION

 NP1  NP2
  6    6
      M1 ARRAY
 0.000000        0.257480E-01   0.852690E-01   0.149890         0.207820          0.231720
      THETA1 ARRAY
 0.504740        0.833650         1.10168         1.17271         1.25245           1.29961
      M2 ARRAY
 0.000000        0.257480E-01   0.852690E-01   0.149890         0.207820          0.231720
      THETA2 ARRAY
 0.420240        0.761230         1.05180         1.13873         1.22981           1.28139
    ML                MT              RL              RT
 0.000000        0.231720       0.833300E-01   0.264500



                               ** COMPUTED BLADE DATA **


    RLE              BETAL1            BETAL2            RTE             BETAT1            BETAT2
 0.217386E-02     51.2904           51.2865        0.213013E-02     28.8679           31.1841
    CHORD            STGR              DELTA
 0.231720        0.790331       0.495592



                               * BLADE COORDINATES *
                              (ORIGIN AT BLADE LEADING EDGE)

                                     SURFACE 1

                       I       M                 THETA           DTDM            D2TDM2
    TANGENCY POINT     1   0.477540E-03    0.163135E-01     14.9739          -134.296
                       2   0.257480E-01    0.338058         9.94511          -263.702
                       3   0.852690E-01    0.606088         1.46705          -21.1741
                       4   0.149890        0.677117         1.04757            8.19146
                       5   0.207820        0.756858         1.79708           17.6848
    TANGENCY POINT     6   0.228561        0.797385         2.08430           10.0110


                                     SURFACE 2

                       I       M                 THETA           DTDM            D2TDM2
    TANGENCY POINT     1   0.387008E-02   -0.163153E-01     14.9718          -152.893
                       2   0.257480E-01    0.265638         10.3914          -265.830
                       3   0.852690E-01    0.556208         1.77377          -23.7373
                       4   0.149890        0.643137         1.25508            7.68391
                       5   0.207820        0.734218         1.98404           17.4830
    TANGENCY POINT     6   0.230693        0.783442         2.28825            9.11722
```

Figure 3(b). - EXAMPLE OUTPUT

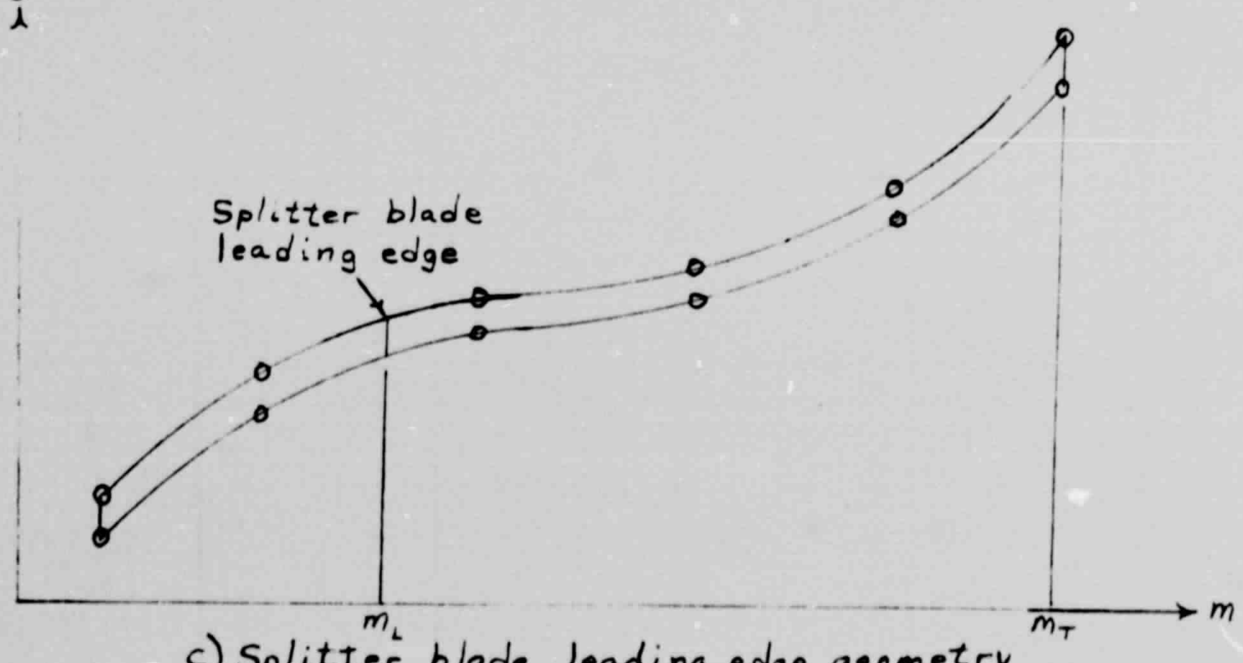
```
1
    EXAMPLE 2 - CENTRIFUGAL COMPRESSOR HUB SECTION - SPLITTER BLADE

 NP1  NP2
   6    6
        M1 ARRAY
  0.000000       0.257480E-01   0.852690E-01   0.149890        0.207820        0.231720
        THETA1 ARRAY
  0.504740       0.833650        1.10168        1.17271         1.25245         1.29961
        M2 ARRAY
  0.000000       0.257480E-01   0.852690E-01   0.149890        0.207820        0.231720
        THETA2 ARRAY
  0.420240       0.761230        1.05180        1.13873         1.22981         1.28139
     ML             MT             RL             RT
  0.517320E-01   0.231720       0.102740       0.264500


                              ** COMPUTED BLADE DATA **


     RLE            BETAL1         BETAL2         RTE             BETAT1          BFTAT2
  0.284874E-02    23.3568        23.5919       0.213013E-02    28.8679         31.1841
     CHORD          STGR           DELTA
  0.179988       0.288974       0.996949


                                * BLADE COORDINATES *
                             (ORIGIN AT BLADE LEADING EDGE)

                                      SURFACE 1

                        I       M             THETA          DTDM          D2TDM2
     TANGENCY POINT     1   0.171933E-02   0.254581E-01    4.20327       -150.820
                        2   0.335370E-01   0.104730        1.46705       -21.1741
                        3   0.981579E-01   0.175760        1.04757        8.19146
                        4   0.156088       0.255501        1.79708        17.6848
     TANGENCY POINT     5   0.176829       0.296028        2.08430        10.0110

                                      SURFACE 2

                        I       M             THETA          DTDM          D2TDM2
     TANGENCY POINT     1   0.398887E-02  -0.254126E-01    4.25075       -143.920
                        2   0.335370E-01   0.548503E-01    1.77377       -23.7373
                        3   0.981579E-01   0.141780        1.25508        7.68391
                        4   0.156088       0.232860        1.98404        17.4830
     TANGENCY POINT     5   0.178961       0.282084        2.28825        9.11722
```



a) Given mean camber line and tangential thickness



b) Given blade surface coordinates



c) Splitter blade leading edge geometry

Figure 4 - Various Applications of the Program