NASA TECHNICAL MEMORANDUM

NASA TM X- 62,309

NASA TM X- 62,309

(NASA-TM-X-62309) GEOMETRY PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY (NASA) 108 p HC $7.50 CSCL 01A

N74-11810

Unclas

G3/01 23014

GEOMETRY PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan

Ames Research Center
Moffett Field, Calif. 94035

September 1973

TABLE OF CONTENTS

GEOMETRY PROGRAM FOR AERODYNAMIC

LIFTING SURFACE THEORY

Richard T. Medan

Ames Research Center

ABSTRACT

This document is a description of and user's manual for a USA FORTRAN IV computer program that provides the geometry and boundary conditions appropriate for an analysis of a lifting, thin wing with control surfaces in linearized, subsonic, steady flow using kernel function method lifting surface theories. The data which is generated by the program is stored on disk files or tapes for later use by programs which calculate an influence matrix, plot the wing planform, and evaluate the loads on the wing. In addition to processing data for subsequent use in a lifting surface analysis, the program is useful for computing area and mean geometric chords of the wing and control surfaces. Any planform shapes, including asymmetrical ones, ones with mixed straight and curved edges, and ones with control surfaces may be handled. The program is able to compute a control surface downwash mode and residual mode and a control surface lifting pressure mode. This in turn allows conventional lifting surface methods to effectively handle partial span, trailing edge control surfaces.

GEOMETRY PROGRAM FOR AERODYNAMIC

LIFTING SURFACE THEORY

Richard T. Medan
Ames Research Center
September 1973

1 INTRODUCTION

This document is a description of and user's manual for a USA FORTRAN IV computer program that provides the geometry and boundary conditions appropriate for an analysis of a lifting, thin wing with control surfaces in linearized, subsonic, steady flow using kernel function method lifting surface theories. The data which is generated by the program is stored on disk files or tapes for later use by programs which calculate an influence matrix, plot the wing planform, and evaluate the loads on the wing. In addition to processing data for subsequent use in a lifting surface analysis, the program is useful for computing area and mean geometric chords of the wing and control surfaces. Any planform shapes, including asymmetrical ones, ones with mixed straight and curved edges, and ones with control surfaces may be handled. The program is able to compute a control surface downwash mode and residual mode and a control surface lifting pressure mode. This in turn allows conventional lifting surface methods to effectively handle partial span, trailing edge control surfaces (ref. 1).

Questions concerning either this document or the computer program or the associated computer programs should be directed to

R. T. Medan
Mail Stop 221-2
Ames Research Center
Moffett Field,
Ca. 94035

2 NOTATION

2.1 ARRAY NOTATION

This notation will be explained by example.

(ETA)	means ETA is an array
(ETA(JJMAX))	refers to the elements of (ETA) from 1 through JJMAX
(ALFAB(PP,NF))	refers to the first PP rows and NF columns of the two dimensional array (ALFAB)
ALFAB(3,NF)	refers to the element of (ALFAB) in row 3 and column NF

3 INPUT AND OUTPUT DESCRIPTION

The function of the program is to determine and store on data files certain geometry information and boundary conditions for a wing planform given a very basic description of the planform. The specific information to be determined and stored on the data files, which are used by other computer programs, is discussed in sections 3.2 and 3.3 following a description of the input in section 3.1. The coordinate system is defined in fig. 1.

3.1 INPUT DATA

The input data is described and defined in the following paragraphs in the order in which it is read by the program. The input FORMAT is 8F10.0 for floating point numbers, 16I5 for integer numbers, and 10L1 for logical variables unless otherwise indicated. Each item in the following list represents one or more cards and, depending on the data of previous items, may or may not be read by the program. All coordinates and areas can be given in any consistent units.

3.1.1 ITEM 1

(TITLE)
This array is used for titling information. The information may consist of up to 80 characters (somewhat less if the integer word length of the computer does not divide evenly into 80). If the planform plotting program is to be used, the title should be centered in the first 42 columns.

3.1.2 ITEM 2

ID
This is an identification number which can either be given by the user or assigned by the program. If the input value is zero, the program will read an identification number file, increment the first value read by one, rewrite the identification number file using the incremented value, and use the incremented value as the identification number. In order to do this a file must already exist with four integer identification numbers on it in unformatted (binary) form. The unit number of this file is determined by the value of the integer W9 in the program (see the DATA statements).

CWTYPE (cols. 6-10)
This is an integer denoting the type of chordwise control point distribution for applying the boundary conditions.

<0 indicates that the program should accept a control point distribution given by the user. See section 3.1.21.
=0 indicates that the program should put control points first on the trailing edge, then on the leading edge, then equally-distributed by the cosine rule on the interior (fig. 2).
>0 indicates that the program should compute and store control points using a modified cosine rule (fig. 3).

SWTYPE (cols. 11-15)
This is an integer denoting the type of spanwise control point distribution.

=0 indicates that the program should compute and store a uniform cosine distribution (fig.4).
≠0 indicates that the program should accept and store a user-defined distribution (see 3.1.22).

IPINFO (cols. 16-20)
A non-zero value causes the integration stations and information at these stations to be printed (see test cases).

CPRESS (cols. 21-25)
A non-zero value causes the flap lifting pressure mode (provided other input indicates that a flap exists (see 3.1.3)) to be computed at NC (see 3.1.23) chordwise stations for each spanwise control station.

UNSYM (cols. 26-30)
This is an integer. It should be zero if and only if the leading and trailing edges are symmetric with respect to the center line.

TRNFRM (cols. 31-35)
This is an integer indicating the type of transformation to be performed upon the wing edge definitions input by the user. The transformation will only be done if TRNANG (see 3.1.6) is not zero. See Chapter 5 for considerations relevant to the use of the transformation capability.

<0 indicates a cranking transformation (fig. 5)
=0 indicates a shearing transformation (fig. 5)
>0 indicates a rotating transformation (fig. 5)

OUT (cols. 36-40)
This is an integer indicating the output level for SUBROUTINE INTGRT, the program which computes wing and flap areas and mean geometric chords.

- -1 gives nothing at all
- 0 gives warning messages
- 1 gives the above + answers
- 2 gives the above + the given and effective locations of planform kinks
- 3 gives the above + input function values and integration stations

NO7 (cols. 41-45)
A non-zero value suppresses writing the geometry file (section 3.2). A zero value causes it to be written on unit W7 (see the DATA statements).

NO8 (cols. 46-50)
A non-zero value suppresses writing the boundary condition file (section 3.2). A zero value causes it to be written on unit W8 (see the DATA statements).

3.1.3 ITEM 3

(BCS(10))
This is a logical array specifying which symmetric boundary conditions will be computed and/or read and stored on unit W8 (see 3.3). A full explanation of this item and also of the following item is not included because the boundary condition program which was developed after this one performs this function and is more convenient to use than the geometry program. See the boundary condition program listing for an explanation of the significance of each item of (BCS) and (BCAS). BCS(1)=.TRUE. (i.e. a T in column 1) refers to the uniform downwash case, causing this case to be computed and stored (NO8=0). BCS(2)=.TRUE. refers to the pitching mode from which quasi-steady pitching derivatives may be obtained. BCS(5)=.TRUE. refers to the flap downwash mode and tells the program to expect flap data.

3.1.4 ITEM 4

(BCAS(10))
This is a logical array specifying which anti-symmetric boundary conditions will be computed and/or read and stored on unit W8 (see 3.3). BCAS(2)=.TRUE. refers to the rolling mode from which quasi-steady rolling derivatives may be obtained.

3.1.5 ITEM 5

MACH
This is the Mach number for the flap pressure mode calculation. It is also the default Mach number for the influence matrix program and will be used if no other Mach number is specified at the time of running the influence matrix program.

3.1.6 ITEM 6 (FORMAT=2I5,3F10.0)

NL
This is the number of pairs of points on the entire wing for definition of the leading edge. The maximum value allowed is 100.

LETYPE (cols. 6-10)
This variable controls the type of interpolation to be used between the data points for the leading edge.

=0 Linear interpolation between points.
≠0 A controlled deviation interpolation will be used between points (ref. 2). This method can handle edges with mixed straight and curved sections. To input a section which is a straight line, give 3 points lying on the line.

LESCAL (cols. 11-20)
This is a scale factor (floating point) for the XL's. The default value is 1.0.

XCLE (cols. 21-30)
This is an interpolation control constant used when LETYPE≠0. A value of 0. will give linear interpolation in the end intervals. A value of 1.0 will give parabolic interpolation in the end intervals. A value in between will give a curve in between. The end intervals are the spaces closest to the wing tips.

TRNANG (cols. 31-40)
This is the cranking, shearing, or rotation angle to be imposed on the planform data. The angle should be in degrees. Input zero if no transformation is desired.

3.1.7 ITEM 7

(YL(NMAX))
This array is to contain the spanwise coordinates of the leading edge definition points. Only the points for Y≥0 need be given when the leading edge is symmetric with respect to Y. The values must be in descending order and YL(1) must be the spanwise coordinate of the right tip. If UNSYM=0, then NMAX=(NL+1)/2. If UNSYM≠0, then NMAX=NL.

3.1.8 ITEM 8

(XL(NMAX))
This array is to contain the streamwise coordinates of the leading edge definition points. These values will be multiplied by the scale factor LESCAL immediately after being read in. For the definition of NMAX see ITEM 7.

3.1.9 ITEM 9--read only if LETYPE≠0

NDL
The number of Y stations of ITEM 8 on the entire wing where the leading edge is kinked. The program needs to know where the kinks are in order to integrate for the wing area and mean geometric chord more accurately.

3.1.10 ITEM 10--read only if LETYPE≠0 and NDL≠0.

(INDDL(NMAX))
This is to be an integer array giving the Y stations where there is a kink. A value of 2 would, for example, mean that the second number of ITEM 7 would be the location of a kink. If the wing has a symmetric effective leading edge, only the indices corresponding to Y≥0 are required. If UNSYM≠0, then NMAX=NDL. If UNSYM=0, then NMAX=(NDL+1)/2.

3.1.11 ITEM 11 (FORMAT=2I5,2F10.0)

NT
This is the number of pairs of points on the entire wing for definition of the trailing edge. The maximum value allowed is 100.

TETYPE (cols. 6-10)
This integer variable controls the type of interpolation to be used between the data points for the trailing edge. It corresponds to LETYPE of ITEM 6.

TESCAL (cols. 11-20)
This is a scale factor for the XT's. The default value is LESCAL, not 1.0.

XCTE (cols. 21-30)
This is the interpolation control constant for the trailing edge when TETYPE≠0. It corresponds to XCLE of ITEM 6.

3.1.12 ITEM 12

(YT(NMAX))
This is the trailing edge counterpart of ITEM 7. (YT) must be in descending order. YL(1) defines the wing tip, so YT(1) could be greater or less than YL(1). NMAX=NT or NMAX=(NT+1)/2 according to the value of UNSYM.

3.1.13 ITEM 13

(XT(NMAX))
This is the trailing edge counterpart of ITEM 8. The values are scaled by TESCAL just after being read in.

3.1.14 ITEM 14--read only if TETYPE≠0

NDT
This is the trailing edge counterpart of ITEM 9.

3.1.15 ITEM 15

(INDDT(NMAX))
This is the trailing edge counterpart of ITEM 10. NMAX=NDT or NMAX=(NDT+1)/2 according to the value of UNSYM.

3.1.16 ITEM 16

AWING
This is the wing reference area and is optional. If not given, the program will compute the actual area.

CBARW (cols. 11-20)
This is the wing reference chord and is optional. If not given, the program will compute a reference chord. The value computed is the integral of c**2*dY over the span divided by the reference area.

AFLAP (cols. 21-30)
This is the flap reference area. It also will be computed if not given and if there is a flap (see 3.1.3).

CBARF (cols. 31-40)
This is the flap reference chord. If not given, and if there is a flap, then it will be computed as the integral of cf**2*dY over the flap span divided by the flap reference area.

3.1.17 ITEM 17--read only if there is a flap.

YF1
This is the spanwise position of the left flap edge.

YF2 (cols. 11-20)
The spanwise position of the right flap edge. YF2>YF1.

3.1.18 ITEM 18--read only if there is a flap.

CHI1
The chordwise position of either the left flap corner or the hingeline at Y=0 relative to the local chord and nondimensionalized by the local chord. 0<CHI1<1. If the wing is unsymmetric after being transformed, then CHI1 applies at the left flap corner. Also, if YF1≠-YF2 (indicating that there are two flaps), then CHI1 applies at the left flap corner. Otherwise there is only a single flap and CHI1 applies at the centerline, while CHI2 (below) applies at both corners. In the latter instance, the hingeline may be kinked at the centerline.

CHI2 (cols. 11-20)
The chordwise position of the right flap corner.

3.1.19 ITEM 19

JJMAX
This is the maximum number of spanwise integration stations. This number will be the upper limit for most of the integrations involved with the current wing. It will be used to determine the wing and flap areas and chords. It will be used for the maximum number of spanwise integration stations in the influence matrix program. It will be used to integrate for the pitching moment in the forces program. And it will be used elsewhere. The value should be chosen as large as possible. If it is input as zero, then the program will determine the largest possible value not exceeding JJMXMX (see DATA statements) such that the following relation will be satisfied: JJMAX+1=(NN+1)*2**K, where K is

some integer. NN is given below. This relationship between NN and JJMAX occurs because the control points must be a subset of the integration points.

3.1.20 ITEM 20

PP
This is the desired number of chordwise control points. The chordwise control points determined by the geometry program serve only as a default set. They may be changed at will in the influence matrix program. The maximum value for PP is 15.

NF (cols. 6-10)
This is the desired number of spanwise control points. The spanwise control points determined by the geometry program serve only as a default set. They may be changed in the influence matrix program, but not completely arbitrarily as in the case of the chordwise control points. In the influence matrix program (and in this program) the spanwise control points must be a subset of the spanwise integration points. The maximum value for NF is 47.

NN (cols. 11-15)
This is a reference number for computing the spanwise control points. The default value is NF. See 3.1.2 for further explanation.

LL (cols. 16-20) This is the number of times that the spanwise integrations will be done for the flap pressure mode. The default value is 4. **NOTE: currently the flap pressure mode calculation program is under revision and this capability is not available. Also it is more convenient to use the boundary condition program for this purpose***.

3.1.21 ITEM 21--read only if CWTYPE<0.

(CHICP(PP))
Values of the chordwise control point locations referenced to the local leading edge and nondimensionalized by the local chord. $0 \leq CHICP(P) \leq 1$. Values must be in ascending order.

3.1.22 ITEM 22--read only if SWTYPE≠0.

(NINDEX(NMAX))
This array is to contain integers for computing the spanwise control point positions by ETA(N)=COS(NINDEX(N)*PI/(NN+1)) where PI=3.14... If SWTYPE=0, then the program will compute (NINDEX) as 1,2,3,4,.... NMAX=NF for an unsymmetric wing and NMAX=(NF+1)/2 for a symmetric wing

3.1.23 ITEM 23--read only if there is a flap.

NC The number of chordwise points at which the flap net pressures will be computed. The spanwise stations that the program will use are the spanwise control stations. The default value for NC is 49. The maximum allowable value is 50.

3.1.24 ITEM 24--read only if there are some downwash distributions to be read from cards. This item is read for N = 1, NMAX. If $PP \leq 8$, then NMAX cards are read. If $9 \leq PP \leq 16$, then 2*NMAX cards are read. NMAX = (NF+1)/2 for symmetric cases on a symmetric wing. NMAX = NF/2 for antisymmetric cases on a symmetric wing. NMAX = NF for an unsymmetric wing. Input from right tip towards left tip.

(ALFAB(P,N),P=1,PP)
Downwash values to be stored on the boundary condition file. Input from leading edge towards trialing edge. The user will have to examine the program in order to use this capability of supplying downwash modes. The boundary condition program performs this function much better anyway.

3.1.25 ITEM 25

<u>DELTAO</u>
The chordwise integration accuracy parameter. This card is read by SUBROUTINE FLPDWN and may be repeated in order to assess the convergence. Use the smallest value last. A value of zero causes a return to the main program. DELTAO=the square of the aspect ratio is the default value.

3.2 GEOMETRY FILE

The following data is written on the file referenced by the integer W7 in the program. The data is written in binary form and comprises 2 records if there is no flap and 3 records if there is a flap. The writing of the data on the file can be suppressed by inputting NO7 as non-zero (see 3.1.2). All lengths written on this file have been nondimensionalized (usually by the effective semispan, B2).

3.2.1 FIRST LOGICAL RECORD

ID
Identification number. See 3.1.2.

PP
The number of chordwise control points.

NP
The number of spanwise control points. In subsequent programs this number is given the name MM.

CWTYPE
Integer denoting the type of chordwise control point distribution. See 3.1.2.

SWTYPE
Integer denoting the type of spanwise control point distribution. See 3.1.2.

UNSYM
Integer which, when ≠0, indicates an unsymmetric wing.

NDL
Number of discontinuities (kinks) on the leading edge. If there are none, then the number will be 1 and a fake kink will be put in (YDL). NDL=NL if LETYPE=0.

NDT
Number of kinks on the trailing edge. This number has a minimum value of 1. NDT=NT if TETYPE=0.

NN
Reference number for spanwise control points. See 3.1.20. In subsequent programs this value is referred to by the name MREF.

JJMAX
The maximum number of integration stations.

NFLAPS
The number of flaps.

(TITLE(26))
Alphanumeric titling information. See 3.1.1

NTITL
The number of words in the title. NTITL=80/(integer word length of the computer being used).

3.2.2 SECOND LOGICAL RECORD

(CHICP(PP))
Chordwise control points.

(NINDEX(NF))
Integer array from which the spanwise control points may be derived using (ETA). ETACP(I)=ETA(NINDEX(I)*(JJMAX+1)/(NN+1)). See below for (ETA).

(TANLEL(JJMAX))
Tangents of the sweep angles of the leading edge just to the left of the integration points. This array is denoted by LAMLEL in the program, but LAMLEL is a misnomer. The next 3 arrays have also been renamed.

(TANLER(JJMAX))
Tangents of the sweep angles of the leading edge just to the right of the integration points.

(TANTEL(JJMAX))
Tangents of the sweep angles of the trailing edge just to the left of the integration points.

(TANTER(JJMAX))

Tangents of the sweep angles of the trailing edge just to the right of the integration points.

(ETA(JJMAX))
This is a double precision array giving the spanwise locations of the integration points. The values are nondimensionalized by the effective semispan and are given by the following equation: ETA(J)=COS(J*PI/(JJMAX+1)).

(STHETA(JJMAX))
This is a double precision array related to the (ETA) array. STHETA(J)=SQRT(1.-ETA(J)**2).

(XSILIF(JJMAX))
This array gives the streamwise coordinates of the leading edge at the integration stations. Values are nondimensionalized by the effective semispan.

(CORDIF(JJMAX))
This array contains the chord lengths at the integration stations. Values are nondimensionalized by the effective semispan.

BRAT
Lateral reference length/effective semispan.

CBARBR
Reference chord/effective semispan.

AR
Aspect ratio.

TR
Wing taper ratio.

MACH
Default Mach number (floating point).

(YDL(NDT))
Spanwise locations of kinks in the leading edge nondimensionalized by the effective semispan. If no kinks actually exist, then there will be 1 number here anyway and its value is 1.E30. When LETYPE=0, this array is the same as (YL) except for being nondimensional (See 3.1.7).

(YDT(NDT))

Trailing edge counterpart of the above.

3.2.3 THIRD LOGICAL RECORD (NFLAPS≠0)

MACH
Mach number again.

LAMDAC
Sweep angle of the hingeline in radians.

CBFCBW
Ratio of the reference chord of the flap to the reference chord of the wing.

AFAW
Ratio of the reference area of the flap to the reference area of the wing.

ETA1
Spanwise location of left flap edge.

ETA2
Spanwise location of right flap edge.

(XLF(2))
The coordinates of the leading edge at the flap side edges. If there are two flaps, these values apply for the flap on the right side of the wing. The values have been nondimensionalized by the effective semispan. The first value is for the leftmost side edge.

(CYF(2))
Chord lengths at the flap side edges.

XSI1
Streamwise coordinate of the left corner of the flap.

XSI2
Streamwise coordinate of the right corner of the flap.

(CHIFPI(JJMAX))
This is an array giving the location of the flap hingeline at the integration stations. The values are the usual CHI values such that

0<CHI<1.

(C2IP(JJMAX))
Values of the function given by eq. A4e of ref. 1 at the integration stations.

3.3 BOUNDARY CONDITION FILE

The following data is written on the file referenced by the integer W8 in the program. The data is written in binary form. The writing of the data on the file can be suppressed by inputting NO8 as non-zero (see 3.1.2). The boundary condition file gives the sets of right hand sides for the equation solving program.

3.3.1 FIRST RECORD

ID
The identification number.

ID3
ID3=0 for this program. For the boundary condition program this number is assigned.

(TITLE(26))
Alphanumeric titling information.

UNSYM

NSYM The number of symmetric cases. NSYM= the number of .TRUE. values of (BCS) (See 3.1.3).

NASYM
The number of antisymmetric cases. NASYM= the number of .TRUE. values of (BCAS) (See 3.1.4).

(BCS)

(BCAS)

PP

CWTYPE

NF
(NF is called MM in other programs.)

MMP
For a symmetric wing MMP=(NF+1)/2.

MMPA

For a symmetric wing MMPA=NF/2. For an unsymmetric wing, MMP = MMPA = NF.

<u>NF</u>

<u>SWTYPE</u>

<u>(CHICE(PP))</u>

<u>(NINDEX(MMP))</u>

<u>(ETACP(MMP))</u>
The control point locations given in 3.1.22. Note that this array is single precision and it is not necessary in view of having also stored (NINDEX(MMP)), but it was merely found convenient to do it this way. Note also that if the wing is symmetric, then only the control points actually used for the symmetric cases are written.

4 METHOD OF SOLUTION

The areas and mean geometric chords were computed first by making the substitution $y=(b/2)\cos\theta$ and using SUBROUTINE INTGRT. This subroutine is self-documented. The approach outlined in SUBROUTINE INTGRT was highly desirable because the integration locations for this subroutine coincide with the integration stations which must be computed and stored anyway. SUBROUTINE INTGRT is called twice for each integration to be performed. The first call uses approximately half of the available integration stations and the second call uses all of the integration stations. The two values are compared to give an indication of the error involved.

A method of weighted quadratics is used for the interpolation when the edges are curved. The interpolation routine, SUBROUTINE CODIM, was taken from ref. 2.

The method of obtaining the flap downwash mode is documented in ref. 1 and in SUBROUTINE FLPDWN.

5 USE OF TRANSFORMATION CAPABILITY

The capability to yaw, shear, or rotate the wing applies only to the wing edges as defined by the arrays (YL), (XL), (YDL), (YT), (XT), and (YDT). If LETYPE=0, then (YDL) is identical to (YL). Otherwise (YDL) is a subset of (YL) derived from (INDDL) (see 3.1.10). (YDT) is similarly related to (YT). After (XL) and (XT) are multiplied by the scale factors LESCAL and TESCAL the transformation is performed. This is all that the transformation capability does. The transformation is accomplished upon reaching statement 1190 of the program and just prior to reading ITEM 16 of the input. All program actions and input beyond this refer to the transformed wing. In particular, if the program computes the wing and/or flap reference chord, the computed value will be that of the transformed wing. This will generally be different from the untransformed wing if the wing has been transformed by yawing.

When TRNFRM<0 (i.e., a cranking transformation), the leading and trailing edge definitions must include Y=0.0 and Y=0.0 becomes a kink which must be named in (INDDL) and/or (INDDT) if either or both is read.

When the wing is to be transformed by rotating, the user must define the leading edge as the portion of the periphery which <u>becomes</u> the leading edge, not the portion of the periphery which is the leading edge prior to rotation. The same is true of the trailing edge. For example, suppose the wing were a square defined by the coordinate pairs (0,1), (0,-1), (2,-1), and (2,1). Ordinarily the leading edge would be thought of as the straight line segment joining (0,1) and (0,-1) and the trailing edge as the segment joining (2,1) and (2,-1). However, if the user were to want to yaw such a wing (TRNANG>0), he would have to tell the program that the wing is unsymmetric (UNSYM≠0) and that the leading edge is comprised of the two line segments joining (2,1) with (0,1) and (0,1) with (0,-1). Similarly, he would have to tell the program the the trailing edge is comprised of the two line segments joining (0,-1) with (2,-1) and (2,-1) with (2,1).

The transformation capability is useful for inputting simple, swept, tapered wings when the leading edge sweep angle is known. For this situation make TRNFRM<0 (see 3.1.2); NL=3 and TRNANG = leading edge sweep angle (3.1.6); YL = semispan (3.1.7); card 8 = blank (3.1.8); NT=3 (3.1.11); YT(1) = semispan (3.1.12); and XT(1) = tip chord and XT(2) = root chord (3.1.13).

The transformation capability is also useful for delta wings. For this situation make TRNFRM<0; NL=3 and TRNANG = leading edge sweep angle; YL(1)= semispan; card 8 = blank; NT=1; card 10 = blank; and XT(1) = root chord.

6 SAMPLE CASES

Some sample cases are given in appendix A. Note that blank fields are equivalent to zero for numerical data and to .FALSE. for logical data. Fig. 6 and fig. 7 are plots of the two sample cases. These plots were made with a planform plotting program which uses the geometry file created by the geometry program.

7 CONTROL CARDS

Job control cards needed to handle the files are given below.

7.1 Ames' IBM TSS system

For the first time that the program is to be used the following TSS commands must be given. These need to be given once and only once for each user ID.

```
SHARE MEDAN,FSARTM,INIDFILE
CDS MEDAN,IDFILE
DELETE MEDAN
SHARE MEDAN,FSARTM,ISPROG.V1
```

After the above commands have been issued by the user, the program is invoked by the following.

```
JBLB MEDAN
CALL GEOM$
(input data)
```

For the Ames' TSS computer system no control cards are needed for the geometry and boundary condition files. The DDEF cards are automatically issued by the GENFIL and BCFIL subroutines for the geometry file and the boundary condition file, respectively. The DDEF card for the identification number file is issued directly from the main program using the Ames' library programs CVRT and OBEY. The latter is the one which actually issues the control card to the operating system. The DSNAMES (data file names) assigned to the geometry file and boundary condition file would be GEOM.X23 and BC.X023.X000 if ID1 were equal to 23. The DSNAME of the identification number file is always IDFILE.

Because of this automatic file defining feature, the following messages may be received in the output from the program. These messages are normal and do not indicate any errors.

```
CANCELLED: DDNAME FT07F001 UNKNOWN
CANCELLED: DDNAME FT08F001 UNKNOWN
```

```
CANCELLED: DDNAME FT09F001 UNKNOWN
```

The program reads from unit 5 and writes on unit 6, so DDEF commands such as the following may be required if the program is run conversationally.

```
DDEF FT05F001,,input
DDEF FT06F001,,output
```

8 REFERENCES

1---Medan, Richard T.: Steady, Subsonic, Lifting Surface Theory for Wings with Swept, Partial Span, Trailing Edge Control Surfaces. NASA TN D-7251, 1973.

2---Tulinius,J.; Clever, W.; Niemann, A.; Dunn, K.; and Gaither, B.: Theoretical Prediction of Airplane Stability Derivatives at Subcritical Speeds. Report no. NA-72-803, North American Rockwell Corp., 1972.

APPENDIX A

FIRST SAMPLE CASE

```
                                   column
              1         2         3         4         5         6         7         8
ITEM 12345678901234567890123456789012345678901234567890123456789012345678901234567890

   1              SAMPLE WING  1
   2     0    1    0    1    0    0    0    2    0    0
   3 FFFFFFFFFF
   4 FFFFFFFFFF
   5 0.0
   6     7    1          0.0
   7 1.        0.7       0.5       0.0
   8 0.5       0.5       0.5       0.0
   9     2
  10     3
  11     5    1
  12 1.0       0.7       0.0
  13 1.5       1.5       2.0
  14     0
  16 0.        0.        0.
  19    71
  20     5   17
```

SECOND SAMPLE CASE

```
                                        column
              1         2         3         4         5         6         7         8
ITEM 12345678901234567890123456789012345678901234567890123456789012345678901234567890

   1               SAMPLE WING  2
   2          1                   1    1    2
   3 FFFFFFFFFF
   4 FFFFFFFFFF
   5 0.0
   6     9    1          .5        20.0
   7 1.        1.        1.        0.7       0.5       0.0       -0.5      -0.7
   7 -1.0
   8 1.5       1.0       0.5       0.5       0.5       0.0       0.5       0.5
   8 0.5
   9     3
  10     3    5    7
  11     7    1
  12 1.0       0.7       0.0       -0.7      -1.0      -1.0      -1.0
  13 1.5       1.5       2.0       1.5       1.5       1.0       0.5
  14     1
  15     5
  16
  19 10000
  20     5   17
```

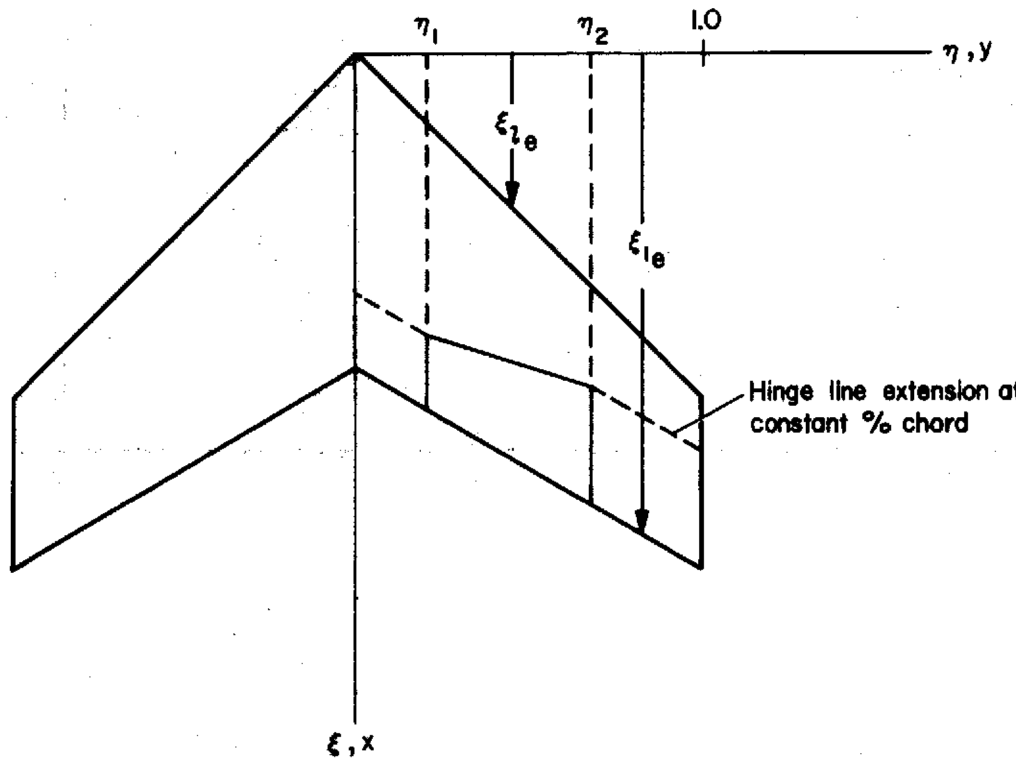


Figure 1.- Definition of coordinate system.

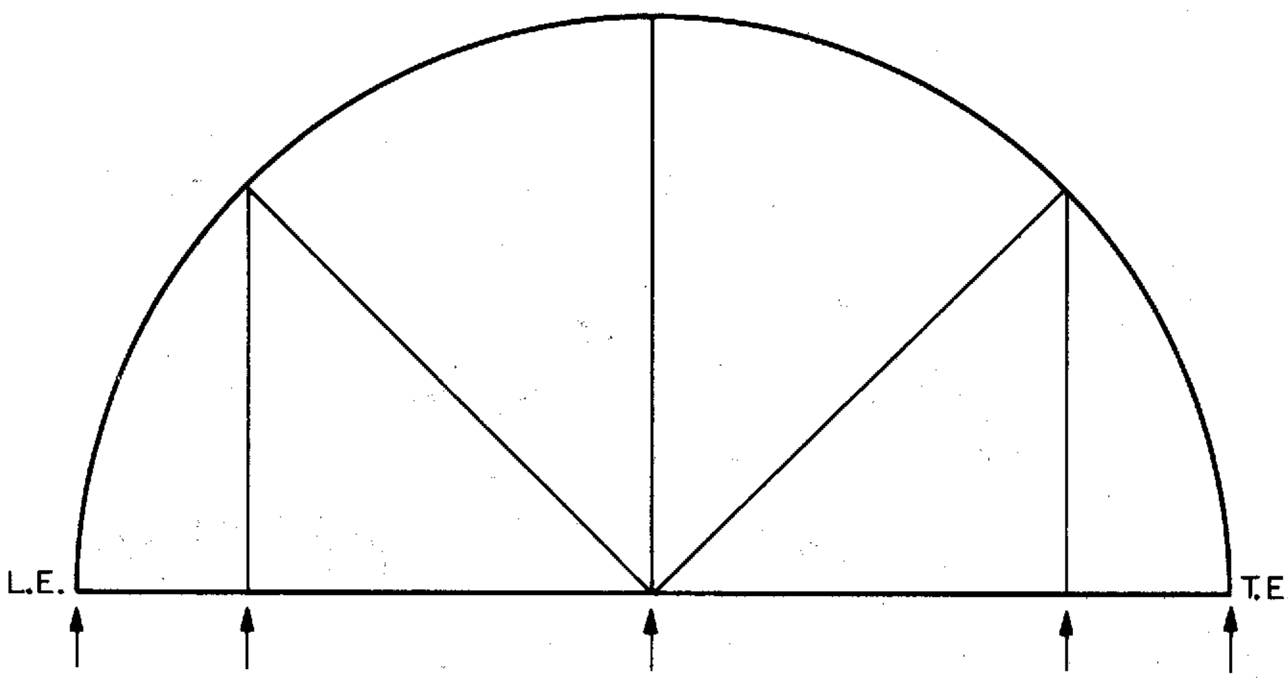


Figure 2.- The Wagner (CWTYPE = 0) chordwise control point distribution for PP = 5.

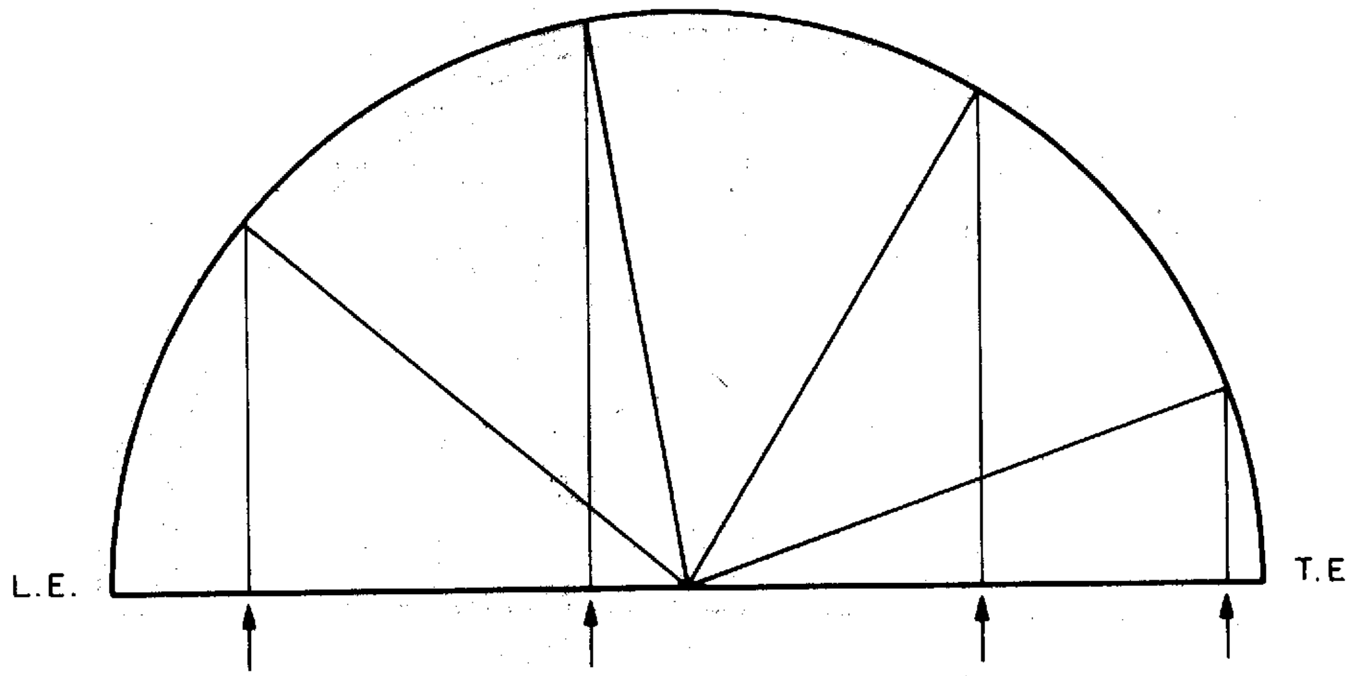


Figure 3.- The MULTHOPP (CWTYPE > 0) chordwise control point distribution for PP = 4.

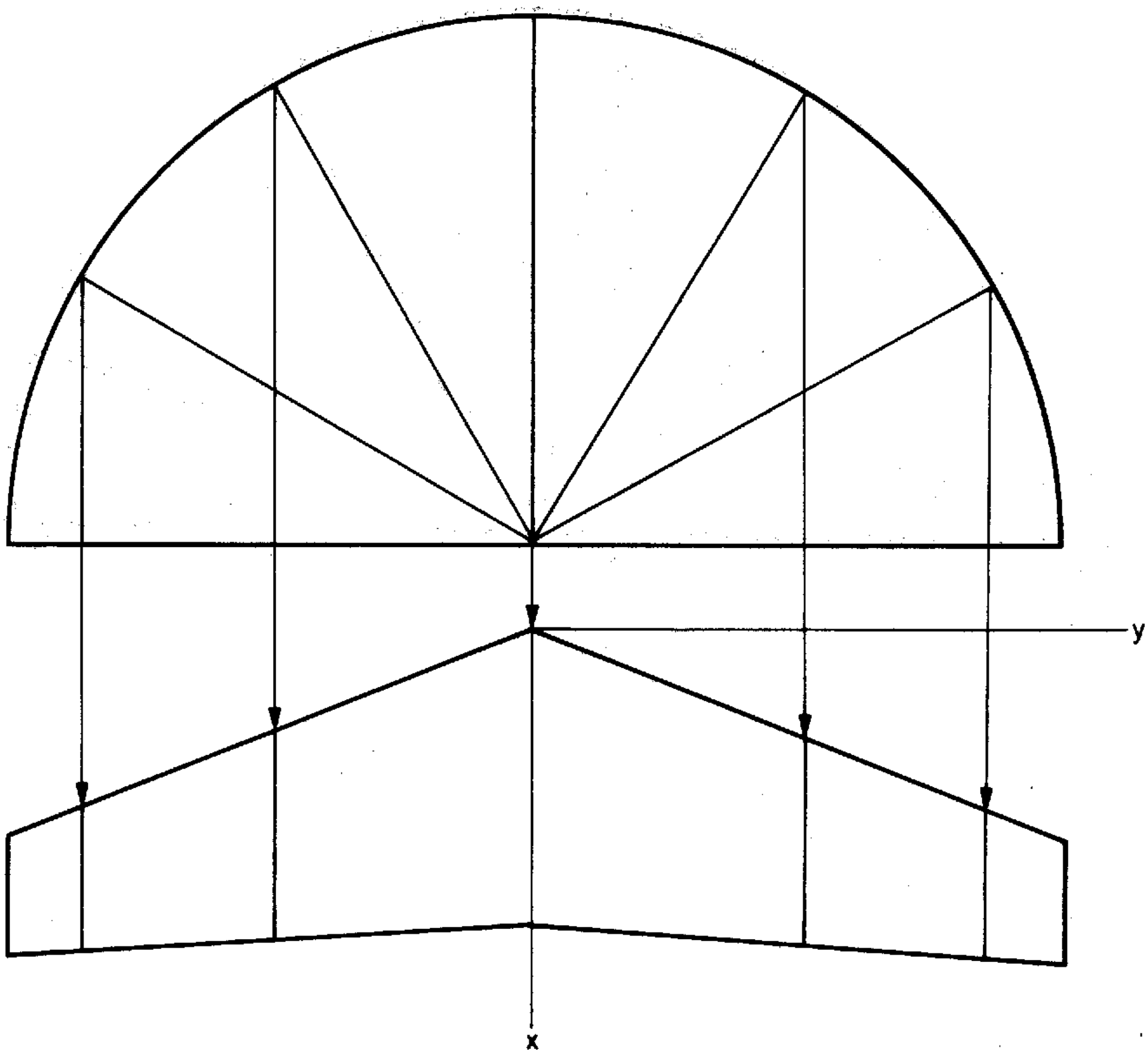


Figure 4.- The spanwise distribution of control points for SWTYPE = 0 and NF = 6.

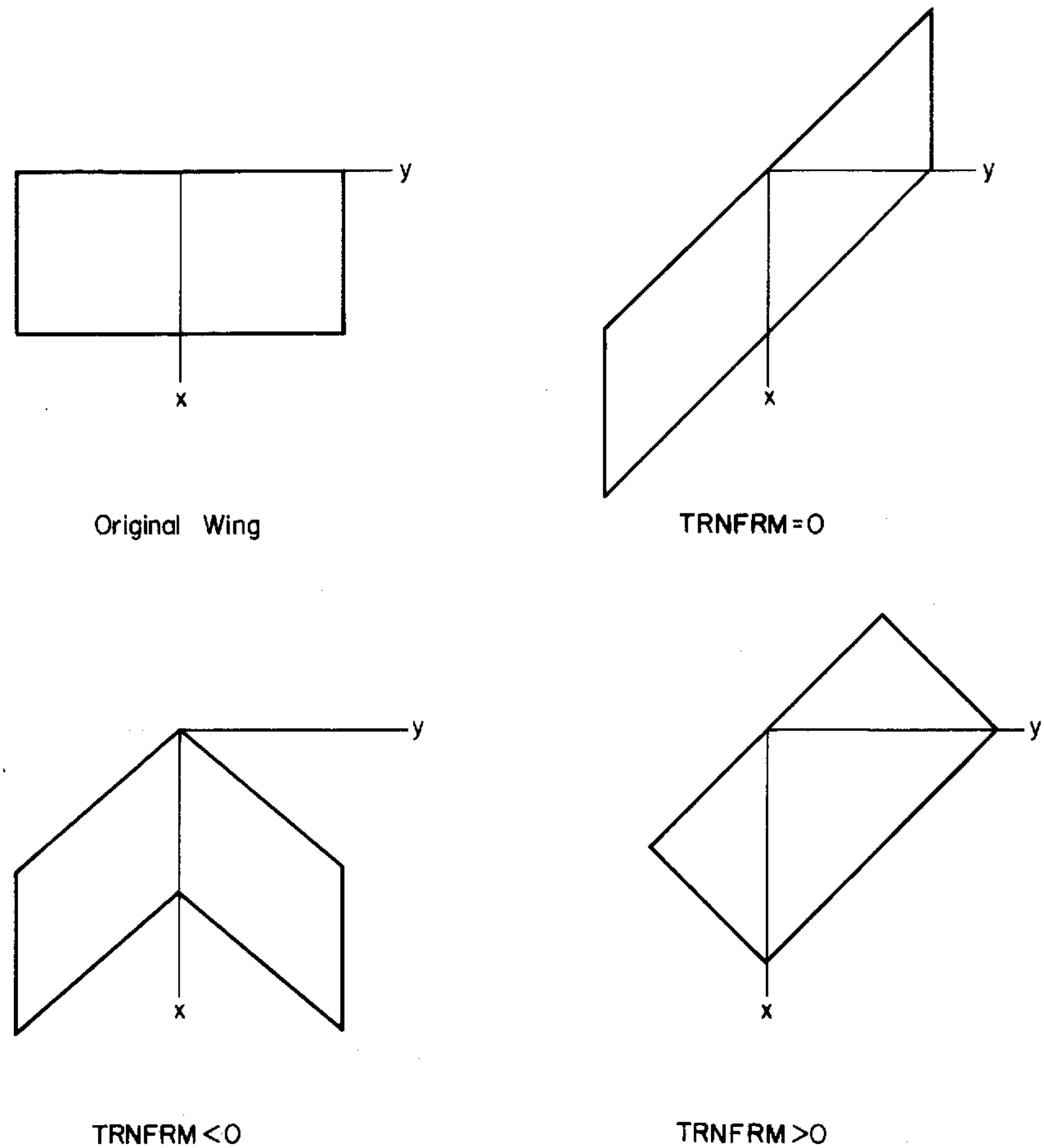


Figure 5.- Illustration of the transformation capability for TRNANG = 45°.

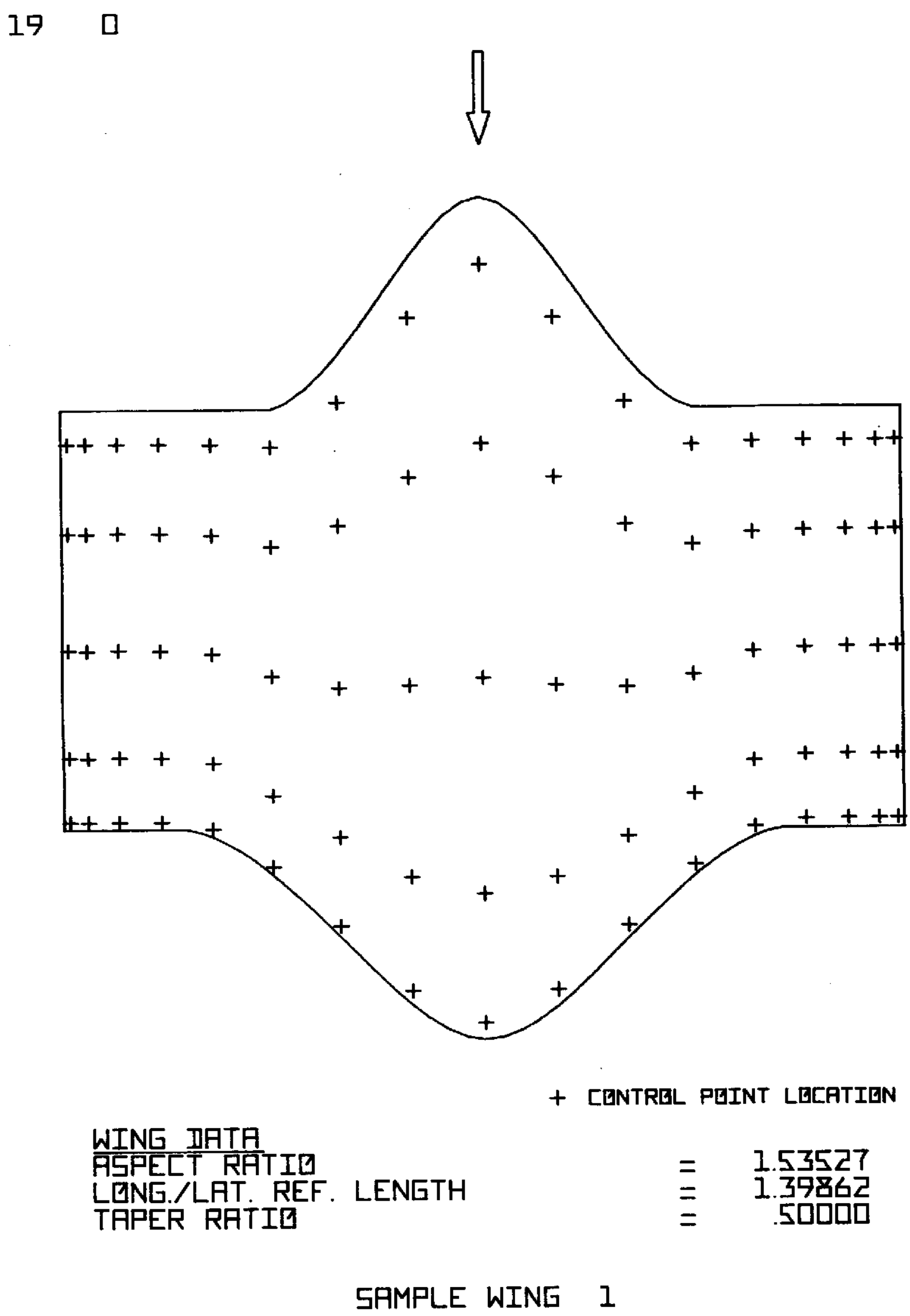
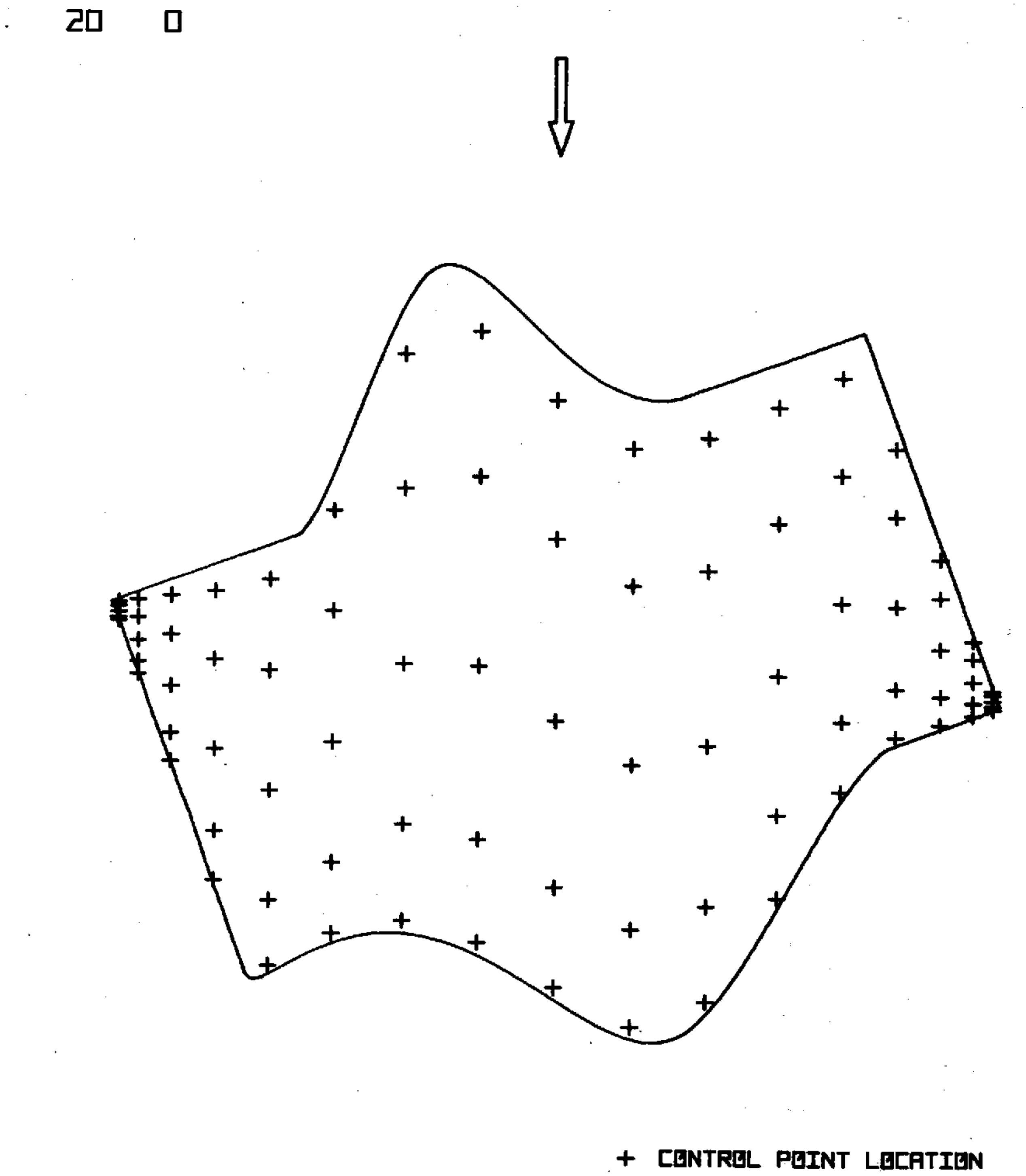


Figure 6.- The planform and control points of the first sample case.



WING DATA

ASPECT RATIO	=	1.53047
LONG./LAT. REF. LENGTH	=	1.38601
TAPER RATIO	=	.00000

SAMPLE WING 2

Figure 7.- The planform and control points of the second sample case.

FLAPPED WING GEOMETRY PROGRAM
==============================

TITLE OF CONFIGURATION = SAMPLE WING 1

ASSIGNED IDENTIFICATION NUMBER = 19

```
CWTYPE =   1
SWTYPE =   0
IPINFO =   1
CPRESS =   0
UNSYM  =   0
TRNFRM =   0
OUT    =   2
NO7    =   0
NO8    =   0
```

```
 INTEGRATIONS FOR THE WING AREA

VALUE OF THE INTEGRAL IS      2.6042147           THE NUMBER OF INTEGRATION POINTS USED =    35

NONE OF THE X-VALUES WERE CONSIDERED DISCONTINUITIES
THE X VECTOR IS
          0.500000            -0.500000

VALUE OF THE INTEGRAL IS      2.6053963           THE NUMBER OF INTEGRATION POINTS USED =    71


 INTEGRATIONS FOR THE WING CBAR

VALUE OF THE INTEGRAL IS      3.6425514           THE NUMBER OF INTEGRATION POINTS USED =    35

NONE OF THE X-VALUES WERE CONSIDERED DISCONTINUITIES
THE X VECTOR IS
          0.500000            -0.500000

VALUE OF THE INTEGRAL IS      3.6439657           THE NUMBER OF INTEGRATION POINTS USED =    71
```

GEOMETRICAL DATA FOR THE FLAPPED WING

WING DATA

WING AREA	=	2.60540	(0.00118)
BREF, LAT. REF. LENGTH	=	1.00000	
B2, EFFECTIVE SEMI-SPAN	=	1.00000	
BREF/B2	=	1.00000	
CBAR, LONG. REF. LENGTH	=	1.39862	(0.00054)
ROOT CHORD	=	2.00000	
ASPECT RATIO	=	1.53527	(0.00070)
TAPER RATIO	=	0.50000	
BREF/CBAR	=	0.71499	(0.00028)
CBAR/ROOT CHORD	=	0.69931	(0.00027)
MACH NUMBER	=	0.00000	

THE NUMBERS IN PARENTHESES ARE ERROR ESTIMATES
(SEE PROGRAM DOCUMENTATION)

LEADING EDGE CO-ORDINATES (SCALE FACTOR = 1.000000)

YL	XL
1.00000	0.50000
0.70000	0.50000
0.50000	0.50000
-0.00000	0.00000
-0.50000	0.50000
-0.70000	0.50000
-1.00000	0.50000

TRAILING EDGE CO-ORDINATES (SCALE FACTOR = 1.000000)

YT	XT
1.00000	1.50000
0.70000	1.50000
0.00000	2.00000
-0.70000	1.50000
-1.00000	1.50000

THE WING HAS KINKS IN THE LEADING EDGE AT THE FOLLOWING ETA LOCATIONS

```
 0.50000
-0.50000
```

GEOMETRICAL DATA AT THE INTEGRATION STATIONS

ETA	X(LE)/B2	C/B2	TANLEL	TANLER	TANTEL	TANTER
0.99905	0.50000	1.00000	0.00000	-0.00006	0.00000	-0.00100
0.99619	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.99144	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.98481	0.50000	1.00000	-0.00001	-0.00000	0.00000	-0.00000
0.97630	0.50000	1.00000	0.00000	0.00001	0.00000	-0.00000
0.96593	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.95372	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.93969	0.50000	1.00000	-0.00000	-0.00000	0.00000	-0.00000
0.92388	0.50000	1.00000	0.00000	0.00000	0.00000	-0.00000
0.90631	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.88701	0.50000	1.00000	0.00000	-0.00000	0.00000	-0.00000
0.86603	0.50000	1.00000	-0.00000	-0.00000	-0.00004	-0.00000
0.84339	0.50000	1.00000	0.00000	0.00000	0.00000	0.00004
0.81915	0.50000	1.00000	0.00000	-0.00000	0.00003	-0.00000
0.79335	0.50000	1.00000	-0.00000	-0.00000	0.00000	-0.00003
0.76604	0.50000	1.00000	0.00000	0.00000	0.00000	-0.00000
0.73728	0.50000	1.00000	0.00000	-0.00000	-0.00003	-0.00000
0.70711	0.50000	1.00000	0.00000	-0.00000	-0.20180	0.00003
0.67559	0.50000	1.00636	-0.00000	-0.00000	-0.36513	0.20180
0.64279	0.50000	1.01834	0.00000	0.00000	-0.47922	0.36513
0.60876	0.50000	1.03464	-0.00000	-0.00000	-0.58765	0.47922
0.57358	0.50000	1.05532	-0.00002	0.00000	-0.68845	0.58765
0.53730	0.50000	1.08029	0.00002	0.00002	-0.77967	0.68845
0.50000	0.50000	1.10937	0.42880	-0.00002	-0.85912	0.77967
0.46175	0.48360	1.15864	0.69587	-0.42880	-0.92452	0.85912
0.42262	0.45637	1.22204	0.93152	-0.69587	-0.97328	0.92452
0.38268	0.41917	1.29811	1.12920	-0.93152	-1.00278	0.97328
0.34202	0.37325	1.38480	1.28162	-1.12920	-1.01004	1.00278
0.30071	0.32030	1.47948	1.38072	-1.28162	-0.99195	1.01004
0.25882	0.26247	1.57886	1.41752	-1.38072	-0.94514	0.99195
0.21644	0.20239	1.67899	1.38202	-1.41752	-0.86601	0.94514
0.17365	0.14326	1.77519	1.26304	-1.38202	-0.75061	0.86601
0.13053	0.08879	1.86202	1.04805	-1.26304	-0.59474	0.75061
0.08716	0.04334	1.93326	0.72297	-1.04805	-0.39392	0.59474
0.04362	0.01186	1.98189	0.27190	-0.72297	-0.14332	0.39392
-0.00000	0.00000	2.00000	-0.27190	-0.27190	0.14332	0.14332

CHORDWISE CONTROL POINT LOCATIONS REFERENCED TO THE LOCAL CHORD

0.07937
0.29229
0.57116
0.82743
0.97975

SPANWISE CONTROL POINT LOCATIONS REFERENCED TO B2

0.98481
0.93969
0.86603
0.76604
0.64279
0.50000
0.34202
0.17365
-0.00000
-0.17365
-0.34202
-0.50000
-0.64279
-0.76604
-0.86603
-0.93969
-0.98481

JJMAX = 71

THE GEOMETRY INFORMATION HAS BEEN STORED ON UNIT W7

NO BOUNDARY CONDITIONS WERE STORED ON UNIT W8

FLAPPED WING GEOMETRY PROGRAM

```
============================

TITLE OF CONFIGURATION =                            SAMPLE WING  2

ASSIGNED IDENTIFICATION NUMBER =   21


CWTYPE =    1
SWTYPE =    0
IPINFO =    0
CPRESS =    0
UNSYM  =    1
TRNFRM =    1
OUT    =    2
NO7    =    0
NO8    =    0
```

INTEGRATIONS FOR THE WING AREA

THE UNCORRECTED VALUE OF THE INTEGRAL = 2.1186180
THE CORRECTED VALUE OF THE INTEGRAL = 2.1185694
THE NUMBER OF INTEGRATION POINTS WAS = 143

ORIGINAL DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

REDUCED DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

THE UNCORRECTED VALUE OF THE INTEGRAL = 2.1185522
THE CORRECTED VALUE OF THE INTEGRAL = 2.1185484
THE NUMBER OF INTEGRATION POINTS WAS = 287

ORIGINAL DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

REDUCED DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

INTEGRATIONS FOR THE WING CBAR

THE UNCORRECTED VALUE OF THE INTEGRAL = 2.6438274
THE CORRECTED VALUE OF THE INTEGRAL = 2.6436501
THE NUMBER OF INTEGRATION POINTS WAS = 143

ORIGINAL DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

REDUCED DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

THE UNCORRECTED VALUE OF THE INTEGRAL = 2.6436939
THE CORRECTED VALUE OF THE INTEGRAL = 2.6436787
THE NUMBER OF INTEGRATION POINTS WAS = 287

ORIGINAL DISCONTINUITIES

0.692069	0.269052	-0.576983	-0.692069

REDUCED DISCONTINUITIES

0.692069 0.269052 -0.576983 -0.692069

GEOMETRICAL DATA FOR THE FLAPPED WING

WING DATA

WING AREA	=	2.61357	(0.00003)
BREF, LAT. REF. LENGTH	=	1.00000	
B2, EFFECTIVE SEMI-SPAN	=	1.11070	
BREF/B2	=	0.90033	
CBAR, LONG. REF. LENGTH	=	1.38601	(0.00001)
ROOT CHORD	=	1.64005	
ASPECT RATIO	=	1.53047	(0.00002)
TAPER RATIO	=	0.00000	
BREF/CBAR	=	0.72149	(0.00001)
CBAR/ROOT CHORD	=	0.84510	(0.00001)
MACH NUMBER	=	0.00000	

THE NUMBERS IN PARENTHESES ARE ERROR ESTIMATES
(SEE PROGRAM DOCUMENTATION)

LEADING EDGE CO-ORDINATES (SCALE FACTOR = 1.000000)

YL	XL
1.45272	1.06752
1.28171	0.59767
1.11070	0.12783
0.82879	0.23043
0.64086	0.29884
0.00000	0.00000
-0.29884	0.64086
-0.48677	0.70926
-0.76868	0.81187

TRAILING EDGE CO-ORDINATES (SCALE FACTOR = 1.000000)

YT	XT
1.45272	1.06752
1.17081	1.17012
0.68404	1.87938
-0.14475	1.64895
-0.42666	1.75156
-0.59767	1.28171
-0.76868	0.81187

```
THE WING HAS KINKS IN THE  LEADING EDGE AT THE FOLLOWING ETA LOCATIONS

        1.11070
        0.64086
       -0.29884

THE WING HAS KINKS IN THE TRAILING EDGE AT THE FOLLOWING ETA LOCATIONS

       -0.42666

THE ABOVE VALUES WERE DETERMINED BY APPLYING
A  20.000 DEGREE ROTATION TRANSFORMATION TO THE INPUT
```

CHORDWISE CONTROL POINT LOCATIONS REFERENCED TO THE LOCAL CHORD

```
0.07937
0.29229
0.57116
0.82743
0.97975
```

SPANWISE CONTROL POINT LOCATIONS REFERENCED TO B2

```
 0.98481
 0.93969
 0.86603
 0.76604
 0.64279
 0.50000
 0.34202
 0.17365
-0.00000
-0.17365
-0.34202
-0.50000
-0.64279
-0.76604
-0.86603
-0.93969
-0.98481
```

JJMAX = 287

THE GEOMETRY INFORMATION HAS BEEN STORED ON UNIT W7

NO BOUNDARY CONDITIONS WERE STORED ON UNIT W8

```
C
C     LIFTING SURFACE GEOMETRY PROGRAM
C
C     THE WING OUTLINE CAN BE
C     DEFINED BY UP TO 100 BREAKPOINTS ON THE LEADING EDGE AND UP TO
C     100 BREAKPOINTS ON THE TRAILING EDGE
C
C     SUBROUTINES USED
C
C     Q         A FUNCTION SUBPROGRAM USED TO CALCULATE A QUANTITY ASSO-
C               CIATED WITH THE FLAP PRESSURE MODE.
C     C2I       A FUNCTION SUBPROGRAM USED TO CALCULATE THE FUNCTION
C               GIVEN BY EQ. A4E OF NASA TN D-7251 (1973).
C     OBEY      AN AMES' LIBRARY PROGRAM TO PASS COMMANDS TO THE TSS
C               OPERATING SYSTEM.  OBEY IS USED TO RELEASE AND DDEF
C               UNIT 9, WHICH IS THE IDENTIFICATION NUMBER FILE.
C     CODIM     A SUBROUTINE SUBPROGRAM USED FOR INTERPOLATION.  SEE
C               PROGRAM DOCUMENTATION.
C     FLPDWN    A SUBROUTINE SUBPROGRAM USED TO DETERMINE THE CONTROL
C               SURFACE DOWNWASH MODE.  SEE PROGRAM DOCUMENTATION.
C     STRAIT    A SUBROUTINE SUBPROGRAM USED FOR STRAIGHT LINE INTERPOLA-
C               TION.
C     INTGRT    A SUBROUTINE SUBPROGRAM USED FOR NUMERICALLY INTE-
C               GRATING TO FIND THE WING AND CONTROL SURFACE AREAS
C               AND MEAN GEOMETRIC CHORDS.
C     GEMFIL    A SUBROUTINE SUBPROGRAM WHICH ISSUES A RELEASE AND DDEF
C               COMMAND TO THE AMES' TSS OPERATING SYSTEM FOR THE
C               GEOMETRY FILE.  GEMFIL WOULD NOT BE NEEDED ON ANOTHER
C               COMPUTER SYSTEM.
C     BCFIL     A SUBROUTINE SUBPROGRAM WHICH ISSUES A RELEASE AND DDEF
C               COMMAND FOR THE BOUNDARY CONDITION FILE.  BCFIL WOULD
C               NOT BE NEED ON ANOTHER SYSTEM.
C
C
C
C     SPECIFICATION STATEMENTS
C
      COMMON FIRST,X,CHI,SQX,DCP
      COMMON ETALI
```

```
      COMMON ETATI
      COMMON XDIS
      COMMON XL
      COMMON XLE
      COMMON XLF
      COMMON XT
      COMMON XTE
      COMMON XTF
      COMMON YDIS
      COMMON YL
      COMMON YLF
      COMMON YT
      COMMON ZDIS
      COMMON INDDL
      COMMON INDDT
C
C
C
      DOUBLE PRECISION D1
      DOUBLE PRECISION ETA
      DOUBLE PRECISION ETAIF
      DOUBLE PRECISION ETAZ
      DOUBLE PRECISION DTHETA
      DOUBLE PRECISION PID
      DOUBLE PRECISION STHETA
      DOUBLE PRECISION THETAD
      INTEGER CPRESS
      INTEGER CWTYPE
      INTEGER OUT, OUT2
      INTEGER P
      INTEGER TRNFRM
      INTEGER PP
      INTEGER PP1
      INTEGER R5
      INTEGER W9
      INTEGER SWTYPE
      INTEGER TETYPE
      INTEGER TITLE
      INTEGER UNSYM
```

```
      INTEGER W6
      INTEGER W7
      INTEGER W8
C
      LOGICAL BCAS
      LOGICAL BCS
      LOGICAL LOG1
      LOGICAL PAIR
      LOGICAL TDUM
C
      REAL LAMDAC
      REAL LAMDCD
      REAL LAMLEL
      REAL LAMLER
      REAL LAMTEL
      REAL LAMTER
      REAL LESCAL
      REAL LINIT
      REAL MACH
C
      DIMENSION ALFAB (15,47)  ,ALPHA(4,15,47),BCAS(10)
      DIMENSION BCS(10)        ,CFSQ( 383)     ,CHI(50)
      DIMENSION CHICP(15)      ,CHIFPI( 383)
      DIMENSION CORDF( 383)    ,CORDIP( 383)   ,CORDSQ( 383)
      DIMENSION CYF(2)         ,C2IP( 383)     ,DCP(50)
      DIMENSION ETA( 383)      ,ETAF(2,6)      ,ETAIF(2)
      DIMENSION ETALI(100)     ,ETATI(100)     ,LAMLEL( 383)
      DIMENSION LAMLER( 383)   ,LAMTEL( 383)   ,LAMTER( 383)
      DIMENSION NINDEX(47)     ,SQX(50)        ,STHETA( 383)
      DIMENSION TITLE(26)      ,X(50)          ,XDIS(200)
      DIMENSION XL(100)        ,XLE( 383)
      DIMENSION XLF(2)         ,XSILIP( 383)   ,XT(100)
      DIMENSION XTE( 383)      ,XTF(2)         ,YDIS(200)
      DIMENSION YDL(104)       ,YDT(104)       ,YL(100)
      DIMENSION YLF(2)         ,YT(100)        ,ZDIS(200)
      DIMENSION XDL(104)       ,XDT(104)       ,INDDL(104)
      DIMENSION INDDT(104)     ,ETACP(47)
C
C
```

```
C
      EQUIVALENCE (LAMLER,CORDSQ)
      EQUIVALENCE (LAMTEL,CORDF)
      EQUIVALENCE (LAMTER,CFSQ)
C
C
C.....LINEAR INTERPOLATION FUNCTION FOR X=F(Y)
C
      LINIT(Y,X1,X2,Y1,Y2)=X1+(Y-Y1)/(Y2-Y1)*(X2-X1)
C
C
C
      D1=1.D0
      ETAZ=0.D0
      JJMXMX= 383
      PI=3.141593
      PID=3.14159265358979 3   D0
      R2D=57.29578
      BETAP=0.
      BETA=0.
      TANSHR =0.
      R5=5
      W9=9
      W6=6
      W7=7
      W8=8
      YAVG=0.
      YDL(1)=1.E30
      YDT(1)=1.E30
C.....IWORDL SHOULD BE SET TO THE INTEGER WORD LENGTH (IN CHARACTERS)
C.....OF THE PARTICULAR COMPUTER BEING USED IF THE PLOT PROGRAMS ARE TO BE
C.....USED.
      IWORDL=4
C
C
C
C     TITLE     =  HEADING FOR PRINTED OUTPUT
C     ID        = IDENTIFICATION NUMBER
C     CWTYPE    = TYPE OF CHORDWISE CONTROL POINTS
```

```
C       SWTYPE    = TYPE OF SPANWISE CONTROL POINTS
C       IPINFO    = PRINT INFORMATION AT INTEGRATION STATIONS
C       CPRESS    = COMPUTE DELTA CP(FLAP) AT NC CHORDWISE CONTROL STATIONS
C       UNSYM     = UNSYMETRICAL WING PLANFORM
C       TRNFRM    = INTEGER FLAG FOR YAWING, CRANKING, OR SHEARING THE WING.
C       OUT       = OUTPUT LEVEL FROM SUBROUTINE INTGRT
C       NO7       = FLAG TO PREVENT WRITING ON UNIT W7
C       NO8       = FLAG TO PREVENT WRITING ON UNIT W8
C
        NTITL = 80/IWORDL
        READ (R5,5001)(TITLE(N), N = 1, NTITL)
        READ (R5, 5003)  ID, CWTYPE, SWTYPE, IPINFO, CPRESS, UNSYM,TRNFRM,-
       1 OUT, NO7, NO8
        OUT2 = MINO(OUT,2)
        IF(ID.NE.0) GO TO 25
C
C.....FILE DEFINITION FOR AMES' TSS SYSTEM.
        CALL OBEY(16,16HRELEASE FT09F001   )
        CALL OBEY(22,24HDDEF FT09F001,,IDFILE       )
        REWIND 9
        READ (W9) ID, ID1, ID2, ID3
        ID=ID+1
        REWIND 9
        WRITE (W9) ID, ID1, ID2, ID3
        END FILE W9
   25   CONTINUE
C
C
        WRITE(W6,6019)
        WRITE(W6,6006) (TITLE(N),N=1,NTITL)
        WRITE(W6,6007) ID
        WRITE(W6,6021)  CWTYPE, SWTYPE, IPINFO, CPRESS, UNSYM,TRNFRM, OUT,-
       1 NO7, NO8
C       BCS       = LOGICAL ARRAY SPECIFING THE SYMETRIC BOUNDARY CONDITIONS
C
        READ (R5,5004) (BCS(I),I=1,10)
C
C       BCAS      = LOGICAL ARRAY SPECIFIING THE ANTI-SYMMETRIC BOUNDARY CONDITION
C
```

```
      READ (R5,5004) (BCAS(I),I=1,10)
      NFLAPS    =  0
      IF (BCS(4).OR.BCAS(3))  NFLAPS =-1
C
C     MACH      = MACH NUMBER
C
      READ (R5,5002) MACH
  109  CONTINUE
C
C     NL        = NUMBER OF CO-ORDINATE PAIRS FOR LEADING EDGE DEFINITION
C     LETYPE    = TYPE OF INTERPOLATION USED ON LEADING EDGE
C     LESCAL    = SCALE FACTOR       XL(I)=XL(I)*LESCAL
C     XCLE      = INTERPOLATION CONTROL CONSTANT FOR END INTERVALS
C                 AND LETYPE .NE. 0
C     TRNANG    = SIDESLIP, CRANK,  OR SHEAR ANGLE IN DEGREES
C
      READ (R5, 5005) NL, LETYPE, LESCAL, XCLE, TRNANG
      IF(NL.LT.2) NL=2
      IF(LESCAL.EQ.0.0) LESCAL=1.0
      IF(UNSYM.EQ.0) NLP= (NL+1)/2
      IF(UNSYM.NE.0) NLP=NL
C
C     YL(I)     = SPANWISE CO-ORDINATES OF LEADING EDGE
C     XL(I)     = CHORDWISE CO-ORDINATES OF LEADING EDGE
C
      READ (R5,5002) (YL(I),I=1,NLP)
      READ (R5,5002) (XL(I),I=1,NLP)
      IF (UNSYM.NE.0)  GO TO 1102
C     IF (UNSYM.EQ.0)  CONTINUE
      DO 1101 I = 1, NLP
      NLL =  NL + 1 - I
      YL(NLL)  =  -YL(I)
      XL(NLL)  =  XL(I)
 1101 CONTINUE
 1102 CONTINUE
      BREFD = (YL(1)-YL(NL))/2.
      DO 110 I = 1, NL
      XL(I)=XL(I)*LESCAL
  110   CONTINUE
```

```
C
C      NDL         =  NUMBER OF DISCONTINUITIES ON LEADING EDGE
C      YDL(I)      =  Y LOCATION OF DISCONTINUITIES
C      INDDL       = INTEGERS FOR DETERMINING YDL WHEN LETYPE .NE. 0
C
       IF(LETYPE.NE.0) GO TO 1104
       NDL         =  NL
       DO 1103 I = 1, NDL
       YDL(I)      =  YL(I)
       XDL(I) = XL(I)
 1103  CONTINUE
       GO TO 1110
 1104  CONTINUE
       READ (R5,5003) NDL
       IF (NDL .EQ. 0) GO TO 1110
       IF (UNSYM.NE.0) GO TO 1107
C      IF (UNSYM.EQ.0)  CONTINUE
       NDLP        =  (NDL+1)/2
       NDL1 = NDL + 1
       READ (R5, 5003) (INDDL(I), I=1, NDLP)
       DO 1106 I = 1, NDLP
       NDLE = NDL1 - I
       INDDL(NDLE) = NL + 1 - INDDL(I)
 1106  CONTINUE
       GO TO 1108
 1107  CONTINUE
       NDLP        =  NDL
       READ (R5, 5003) (INDDL(I), I=1, NDLP)
 1108  CONTINUE
       DO 1109 I = 1, NDL
       IND = INDDL(I)
       XDL(I) = XL(IND)
       YDL(I) = YL(IND)
 1109   CONTINUE
 1110   CONTINUE
C
C      NT          = NUMBER OF CO-ORDINATE PAIRS FOR TRAILING EDGE DEFINITIO
C      TETYPE      = TYPE OF INTERPOLATION USED ON TRAILING EDGE
C      TESCAL      = SCALE FACTOR       XT(I)=XT(I)*TESCAL
```

```
C       XCTE        = INTERPOLATION CONTROL CONSTANT FOR END INTERVALS
C                       AND TETYPE .NE. 0.
C
        READ (R5,5005) NT,TETYPE,TESCAL,XCTE
        IF(NT.LT.2) NT=2
        IF(UNSYM.EQ.0) NTP= (NT+1)/2
        IF(UNSYM.NE.0) NTP=NT
        IF(TESCAL.EQ.0.0) TESCAL=LESCAL
C
C       YT(I)       = SPANWISE CO-ORDINATES OF TRAILING EDGE
C       XT(I)       = CHORDWISE CO-ORDINATES OF TRAILING EDGE
C
        READ (R5,5002) (YT(I),I=1,NTP)
        READ (R5,5002) (XT(I),I=1,NTP)
        IF (UNSYM.NE.0)  GO TO 1112
C       IF (UNSYM.EQ.0)  CONTINUE
        NTA         =  NTP - 1
        NTB         =  NT  + 1
        DO 1111 I = 1, NTA
        NTC         =  NTB - I
        YT(NTC)     =  -YT(I)
        XT(NTC)     =  XT(I)
 1111   CONTINUE
 1112   CONTINUE
        DO 111 I=1,NL
        XT(I)=XT(I)*TESCAL
  111    CONTINUE
C
C       NDT         =  NUMBER OF DISCONTINUITIES ON TRAILING EDGE
C       YDT(I)      =  Y LOCATION OF TRAILING EDGE DISCONTINUITIES
C
        IF (TETYPE.NE.0) GO TO 1114
        NDT         =  NT
        DO 1113 I = 1, NDT
        YDT(I)      =  YT(I)
        XDT(I) = XT(I)
 1113   CONTINUE
        GO TO 1120
 1114   CONTINUE
```

```
      READ (R5,5003) NDT
      IF(NDT .EQ. 0) GO TO 1120
      IF (UNSYM.NE.0) GO TO 1117
C     IF (UNSYM.EQ.0)  CONTINUE
      NDTP      =  (NDT+1)/2
      NDT1 = NDT + 1
      READ (R5, 5003) (INDDT(I), I=1, NDTP)
      DO 1116 I = 1, NDTP
      NDTE = NDT1 - I
      INDDT(NDTE) =  NT + 1 - INDDT(I)
 1116 CONTINUE
      GO TO 1118
 1117 CONTINUE
      NDTP      =  NDT
      READ (R5, 5003) (INDDT(I), I=1, NDTP)
 1118 CONTINUE
      DO 1119  I = 1, NDT
      IND = INDDT(I)
      XDT(I) = XT(IND)
      YDT(I) = YT(IND)
 1119  CONTINUE
 1120  CONTINUE
C
      IF (TRNANG .EQ. 0.) GO TO  1190
C
C.....THE WING WILL BE TRANSFORMED BY CRANKING, SHEARING, OR ROTATING.
C
      IF (TRNFRM) 1121,1124,1124
 1121  CONTINUE
C.....THE CRANKING TRANSFORMATION WILL BE APPLIED TO THE X-VALUES.
      TANSHR = TAN(TRNANG/R2D)
      DO 1122 I = 1, NL
      XL(I) = XL(I) + ABS(YL(I))*TANSHR
      XDL(I) = XDL(I)+ABS(YDL(I))*TANSHR
 1122  CONTINUE
      DO 1123 I = 1, NT
      XT(I) = XT(I) + ABS(YT(I))*TANSHR
      XDT(I)=XDT(I)+ABS(YDT(I))*TANSHR
 1123  CONTINUE
```

```
      GO TO 1190
 1124  CONTINUE
      IF (UNSYM .NE. 0) GO TO 1125
C
C.....IT IS ASSUMED THAT THE WING WILL BECOME UNSYMMETRIC IF
C.....IT IS SHEARED OR ROTATED.
C
      UNSYM = 888
      NLP = NL
      NTP = NT
      NDLP = NDL
      NDTP = NDT
 1125  CONTINUE
      IF (TRNFRM) 1130, 1130, 1160
 1130  CONTINUE
C
C.....THE SHEARING TRANSFORMATION IS APPLIED TO THE VARIOUS X-VALUES.
C
      TANSHR = TAN(TRNANG/R2D)
      DO 1135 I = 1, NL
      XL(I) = XL(I) - YL(I)*TANSHR
      XDL(I)=XDL(I)-YDL(I)*TANSHR
 1135  CONTINUE
      DO 1140 I = 1, NT
      XT(I) = XT(I) - YT(I)*TANSHR
      XDT(I)=XDT(I)-YDT(I)*TANSHR
 1140  CONTINUE
      GO TO 1190
 1160  CONTINUE
C
C.....THE ROTATION BY ANGLE TRNANG IS APPLIED TO (X,Y) PAIRS.
      BETA = TRNANG
      BETAR = BETA/R2D
      CB = COS(BETAR)
      SB = SIN(BETAR)
      DO 1165 I = 1, NL
      XTEMP = XL(I)
      XL(I) = CB*XTEMP - SB*YL(I)
      YL(I) = SB*XTEMP + CB*YL(I)
```

```
 1165  CONTINUE
       DO 1170 I = 1, NT
       XTEMP = XT(I)
       XT(I) = CB*XTEMP - SB*YT(I)
       YT(I) = SB*XTEMP + CB*YT(I)
 1170  CONTINUE
       IF (NDL) 1177, 1177, 1174
 1174  DO 1175 I = 1, NDL
       XTEMP = XDL(I)
       XDL(I) = CB*XTEMP - SB*YDL(I)
       YDL(I) = SB*XTEMP + CB*YDL(I)
 1175  CONTINUE
 1177  IF (NDT) 1190, 1190, 1179
 1179  DO 1180 I = 1, NDT
       XTEMP = XDT(I)
       XDT(I) = CB*XTEMP - SB*YDT(I)
       YDT(I) = XTEMP*SB + CB*YDT(I)
 1180  CONTINUE
 1190  CONTINUE
       YAVG     =  (YL(1) + YL(NL ))/2.0
       IF(UNSYM.EQ.0) GO TO 113
       B=YL(1)-YL(NL )
       B2=B/2.0
       GO TO 114
  113  CONTINUE
       B2=YL(1)
       B=2.0*B2
  114  CONTINUE
       DO 115 I=1,NL
       ETALI(I) =  (YL(I) - YAVG)/B2
  115  CONTINUE
       DO 116 I=1,NT
       ETATI(I) =  (YT(I) - YAVG)/B2
  116  CONTINUE
C
C
C      AREAW     = AREA OF THE WING
C      CBARW     = MEAN AERODYNAMIC CHORD (REFERENCE CHORD)
C      BREF      = STREAMWISE WING REFERENCE LENGTH
```

```
C       AREAF     = AREA OF THE FLAP
C       CBARF     = CHORD REFERENCE LENGTH OF THE FLAP
C
        READ (R5,5002) AREAW, CBARW, BREF, AREAF, CBARF
        IF (BREF .EQ. 0.) BREF = BREFD
        IF(NFLAPS.EQ.0) GO TO 112
C
C       YF1,YF2  = SPANWISE POSITIONS OF THE FLAP EDGES (YF1.LT.YF2)
C       CHI1,CHI2= CHORDWISE POSITIONS OF THE FLAP CORNERS, PERCENT OF CHORD
C
        READ (R5,5002) YF1,YF2
C
C.....YF1 AND YF2 ARE DATA AFTER THE WING HAS BEEN YAWED.    THE USER
C.....MUST DETERMINE THEM FOR EACH SIDESLIP ANGLE.
C
        READ (R5,5002) CHI1,CHI2
        IF (YF1.GE.YF2)  WRITE (W6,6201) YF1, YF2
        IF (YF1.GE.YF2)  STOP
        ETA1R    =  (YF1 - YAVG)/B2
        ETA1 = ETA1R
        IF (UNSYM .EQ. 0) ETA1 = AMAX1(ETA1R, 0.0)
        ETA2      =  (YF2 - YAVG)/B2
        ETAIF(1)=ETA1
        ETAIF(2)=ETA2
C
C.....A SINGLE FLAP ON A SYMMETRICAL WING IS ALLOWED TO HAVE A
C.....KINKED HINGE LINE.  THIS IS THE REASON FOR HAVING BOTH ETA1 AND
C.....ETA1R.  IF THERE IS A SINGLE FLAP ON A SYMMETRICAL WING, CHI1
C.....APPLIES AT THE CENTERLINE, NOT AT ETA1R AS IN THE OTHER CASES.
C.....THE PROGRAM ONLY WORKS WITH THE PART OF THE FLAP ON ETA .GT. 0.
        NFLAPS = 1
        IF (UNSYM .EQ. 0 .AND. ETA1R .GE. 0.0) NFLAPS=2
  112   CONTINUE
C
C       JJMAX     = MAXIMUM NUMBER OF INTEGRATION STATIONS
C       PP        = NUMBER OF CHORDWISE CONTROL POINTS
C       NF        = NUMBER OF SPANWISE CONTROL POINTS ON ENTIRE SPAN
C       NN        = A REFERENCE NUMBER FOR COMPUTING SPANWISE POSITIONS
C       LL        = NUMBER OF TIMES THE SPANWISE INTEGRATION WILL BE DONE
```

```
C
      READ (R5,5003) JJMAX
      READ (R5,5003) PP,NF,NN,LL
      IF (PP.LE.0.OR.NF.LE.0)  WRITE (W6,6202)  PP, NF
      IF (PP.LE.0.OR.NF.LE.0)  STOP
      IF(SWTYPE .EQ.0) NN = NF
      IF (NN .LE. 0) WRITE(W6,6206)
      IF (NN.LE.0)STOP
C
C.....JJMAX MUST BE AN ODD NUMBER
C
      IF  (MOD(JJMAX,2) .EQ. 0) JJMAX = 2*JJMAX + 1
      JJMAX = MIN0(JJMAX,JJMXMX)
      IF(JJMAX.LE. 0) JJMAX = JJMXMX
C
C.....AT THIS POINT JJMAX IS THE MAXIMUM NUMBER OF SPANWISE INTEGRATION
C.....TO BE ALLOWED
C.....JJMAX AND NN MUST BE RELATED BY
C.....    JJMAX+1 = (NN + 1)*2**K
C.....WHERE K = 0,1,2,...
C.....THE FOLLOWING CODE MAKES THE RELATIONSHIP ABOVE HOLD FOR THE
C.....LARGEST K SUCH THAT JJMAX IS NO LARGER THAN ITS CURRENT VALUE
C
      IDUM = 2*(NN+1)
      JJMAX1 = JJMAX + 1
      JJMAX = NN
      DO 350 J = 1, 20
      LLR= J
      IF (IDUM .GT. JJMAX1) GO TO 380
      JJMAX = IDUM - 1
      IDUM = IDUM*2
  350   CONTINUE
C.....CONTROL SHOULD NEVER GET TO HERE
      WRITE (W6,6205)
      STOP
  380   CONTINUE
      LOG1 = MOD(JJMAX,2) .EQ. 0
      IF (LOG1) WRITE (W6,6203 )     JJMAX, NN
      IF (LOG1) STOP
```

```
      IF(LL .LE. 0) LL = LLR
      LL        =  MINO(LL,4)
C
C.....COMPUTING XSI(LE), CHORDS, CHIFLAP, AND C2 AT A
C.....SET OF INTEGRATION POINTS DEFINED BY JJMAX.
C
      ETAFMX    =  AMAX1 (ABS(ETA1R), ABS(ETA2))
      JJMI      =  (JJMAX-1)/2
      JJM2=(JJMAX+1)/2
      JJMAX1    =  JJMAX - 1
      NRATIO=(JJMAX+1)/(NN+1)
C
      DTHETA=PID/DFLOAT(JJMAX+1)
      DO  900 J=1, JJM2
      J2=JJMAX-J+1
      THETAD = DTHETA*DFLOAT(J)
      ETA(J) = DCOS(THETAD)
      STHETA(J)=DSIN(THETAD)
      ETAJ=ETA(J)
      ETA(J2)=-ETA(J)
      STHETA(J2)=STHETA(J)
C
      IF (NFLAPS.NE.0) C2IP(J) =C2I(ETAJ,ETAFMX)
      IF (NFLAPS.NE.0) C2IP(J2)  =C2IP(J)
  900   CONTINUE
C
C
C     CHICP     = CHORDWISE CONTROL POINT POSITIONS (PP VALUES)
C
      IF(CWTYPE) 1040, 1035, 1045
C
 1035  CONTINUE
C
C.....COMPUTING THE CHICP'S ACCORDING TO A SPECIFIED RULE FOR DR. WAGNER
C.....N-INPUT PROGRAM
C
      PP1 = PP - 1
      IF (2 - PP1) 1036, 1036, 1038
 1036  DTHETA = PID/DFLOAT(PP1)
```

```
      DO 1037 P = 2, PP1
      XSI = -DCOS(DFLOAT(P - 1)* DTHETA)
 1037  CHICP(P) = (1. + XSI)/2.
 1038  CHICP(1) = 0.
      CHICP(PP) = 1.
      GO TO 1050
 1040  CONTINUE
C
C.....ARBITRARY CHORDWISE POSITIONS
C
      READ(R5,5002) (CHICP(P), P = 1, PP)
      GO TO 1050
 1045  CONTINUE
C
C.....MULTHOPP, HSU CHORDWISE POSITIONS
C
      IDUM = 2*PP + 1
      DTHETA = PID/DFLOAT(IDUM)
      DO 1047 I = 2, IDUM, 2
      P = I/2
      XSI = -DCOS(DTHETA*DFLOAT(I))
 1047  CHICP(P) = (1. + XSI)/2.
 1050  CONTINUE
C
C     NINDEX   = INDEX FOR SPANWISE CONTROL POINT POSITIONS
C
      IF(SWTYPE .NE. 0) GO TO 1070
C
 1060  CONTINUE
C
C.....THE MULTHOPP SPANWISE POSITIONS
C
      DO 1065 N = 1, NF
 1065  NINDEX(N) = N
      GO TO 1090
 1070  CONTINUE
C
C.....USER-SUPPLIED INDICES FOR SPANWISE POSITIONS
C
```

```
      IF (UNSYM.NE.0)  NFP = NF
      IF (UNSYM.EQ.0)  NFP = (NF+1)/2
      READ(R5,5003) (NINDEX(N), N = 1, NFP)
      IF (UNSYM.NE.0)  GO TO 1090
C     IF (UNSYM.EQ.0)  CONTINUE
      DO 1071 I = 1, NFP
      NIN      =  NF + 1 - I
      NINDEX(NIN)  =  NF + 1 - NINDEX(I)
 1071 CONTINUE
 1090  CONTINUE
C
C     CALCULATION OF LEADING EDGE POINTS AT  INTEGRATION STATIONS
C
      IF(LFTYPE.EQ.0) GO TO 117
      CALL CODIM(XL,ETALI,NL ,ETA,XLE,JJMAX,XCLE)
      CALL CODIM(XL,ETALI,NL ,ETAZ,XLZ,1,XCLF)
      GO TO 118
  117  CONTINUE
      CALL STRAIT(XL,ETALI,NL ,ETA,XLE,JJMAX,W6)
C
      CALL STRAIT(XL,ETALI,NL ,ETAZ,XLZ,1,W6)
  118  CONTINUE
C
C     CALCULATION OF TRAILING EDGE POINTS AT INTEGRATION STATIONS
C
      IF(TETYPE.EQ.0) GO TO 119
      CALL CODIM(XT,ETATI,NT ,ETA,XTE,JJMAX,XCTE)
      CALL CODIM (XT, ETATI, NT , ETAZ, XTZ, 1,XCTE)
      GO TO 120
  119  CONTINUE
      CALL STRAIT(XT,ETATI,NT ,ETA,XTE,JJMAX,W6)
C
      CALL STRAIT(XT,ETATI,NT ,ETAZ,XTZ,1,W6)
  120  CONTINUE
      DO 121 M=1,JJMAX
      XSILIP(M)=XLE(M)/B2
      CORDIP(M)=(XTE(M)-XLE(M))/B2
      CORDSQ(M)=CORDIP(M)**2
  121  CONTINUE
```

```
C
      AREAWP   =  AREAW
      CBARWP   =  CBARW
      IF (AREAW.NE.0.0.AND.CBARW.NE.0.0)  GO TO 123
C
C     DISCONTINUITIES FOR SUBROUTINE INTGRT
C
      ND       =  NDL + NDT
      DO 1211 I = 1, NDL
      XDIS(I) = (YDL(I)-YAVG)/B2
 1211 CONTINUE
      DO 1212 I = 1, NDT
      N        =  NDL + I
      XDIS(N) = (YDT(I)-YAVG)/B2
 1212 CONTINUE
      IF(AREAW.NE.0) GO TO 122
C
C     COMPUTE AREAW IF IT IS NOT SUPPLIED
C
      WRITE (W6,6001)
      WRITE (W6,6002)
      WRITE (W6,6162)
      CALL INTGRT (JJMI,JJMAX,JJMAX,ETA,STHETA,CORDIP,ND,XDIS,YDIS,W6,OU-
     1T2,NLEFT,VALUE)
      AREAWP=B2**2*VALUE
C
      CALL INTGRT(JJMAX, JJMAX, JJMAX, ETA, STHETA, CORDIP,  NLEFT,YDIS,-
     1 ZDIS, W6,OUT, I, VALUE )
      AREAW=VALUE*B2**2
  122  CONTINUE
      IF(CBARW.NE.0.0) GO TO 123
C
C     COMPUTE CBARW IF IT IS NOT SUPPLIED
C
      WRITE (W6,6002)
      WRITE (W6,6164)
      CALL INTGRT(JJMI,JJMAX,JJMAX,ETA,STHETA,CORDSQ,ND,XDIS,YDIS,W6,    -
     1OUT2,NLEFT,VALUE)
      CBARWP   =  B**3/(8.0*AREAW )*VALUE
```

```
C
      CALL INTGRT(JJMAX, JJMAX, JJMAX, ETA, STHETA, CORDSQ,  NLEFT,YDIS,-
     1 ZDIS, W6,OUT, I, VALUE )
      CBARW    =  B**3/(8.0*AREAW)*VALUE
C
  123   CONTINUE
      FBREF2=4.*BREF**2
      AR = FBREF2/AREAW
      ARP= FBREF2/ AREAWP
      DIFAW=ABS(AREAW-AREAWP)
      DIFAR=ABS(AR-ARP)
      DIFCB=ABS(CBARW-CBARWP)
      CBARBR=CBARW/BREF
      CORDTI =XT(1)-XL(1)
      CORDSS=XTZ-XLZ
C
      TR=CORDTI /CORDSS
      SSCBAR=BREF/CBARW
      DIFSS=ABS(SSCBAR-BREF/CBARWP)
      CBARCD=CBARW/CORDSS
      DIFGC=ABS(CBARCD-CBARWP/CORDSS)
      IF(NFLAPS.EQ.0) GO TO 135
C
C     COMPUTE FLAP INFORMATION
C
      IF(LETYPE.EQ.0) GO TO 124
      CALL CODIM(XL,ETALI, NL, ETAIF, XLF, 2,XCLE)
      GO TO 125
  124  CONTINUE
      CALL STRAIT(XL, ETALI, NL, ETAIF, XLF, 2,W6)
C
  125  CONTINUE
      IF(TETYPE.EQ.0) GO TO 126
      CALL CODIM(XT, ETATI, NT, ETAIF, XTF, 2,XCTE)
      GO TO 127
  126  CONTINUE
      CALL STRAIT(XT, ETATI, NT, ETAIF, XTF, 2,W6)
C
  127  CONTINUE
```

```
      CYF(1)=XTF(1)-XLF(1)
      CYF(2)=XTF(2)-XLF(2)
      XHL1=XLF(1)+CYF(1)*CHI1
      XHL2=XLF(2)+CYF(2)*CHI2
      XSI1      =  XHL1/B2
      XSI2      =  XHL2/B2
      DO 132 M=1,JJMAX
      IF(ETA(M)-ETAIF(2)-1.0E-5) 128,128,131
  128  CONTINUE
      IF(ETA(M)-ETAIF(1)+1.0E-5) 129,130,130
  129  CONTINUE
      CORDF(M)=0.0
      CFSQ(M)  =  0.0
      CHIFPI(M)=CHI1
      GO TO 132
  130  CONTINUE
      ETAM      =  ETA(M)
      XHL=LINIT(ETAM  ,XHL1,XHL2,ETA1, ETA2)
      CORDF(M)=XTE(M)-XHL
      CFSQ(M)=CORDF(M)**2
      CHIFPI(M)=(XHL-XLE(M))/CORDIP(M)/B2
C
C USE AVERAGE VALUES FOR FLAP CORNERS WITHIN AN EPSILON OF ETA(M) VALUE
C
      IF (DABS(ETA(M) - ETA1).LT.1.0E-5)  GO TO 1303
      IF (DABS(ETA(M) - ETA2).LT.1.0E-5)  GO TO 1303
      GO TO 132
 1303 CONTINUE
      CORDF(M)=CORDF(M)/2.0
      CFSQ(M)=CFSQ(M)/2.0
      GO TO 132
  131  CONTINUE
      CORDF(M)=0.0
      CFSQ(M)  =  0.0
      CHIFPI(M)=CHI2
  132  CONTINUE
      IF (UNSYM.NE.0)  GO TO 1325
C     IF (UNSYM.EQ.0)  CONTINUE
      DO 1324 I = 1, JJM2
```

```
      M          =  JJMAX + 1 - I
      CHIFPI(M)=  CHIFPI(I)
 1324 CONTINUE
 1325 CONTINUE
C
      IF (AREAF.NE.0.0.AND.CBARF.NE.0.0)  GO TO 134
C
C     DISCONTINUITIES FOR SUBROUTINE INTGRT
C
      DO 1321 I = 1, NDT
      XDIS(I) = (YDT(I)-YAVG)/B2
 1321 CONTINUE
      IF (NFLAPS.EQ.2)  GO TO 1322
      ND          =  NDT + 2
      XDIS(NDT+1)  =  ETA1
      XDIS(NDT+2)  =  ETA2
      GO TO 1323
 1322 CONTINUE
      ND          =  NDT + 4
      XDIS(NDT+1)  =  ETA1
      XDIS(NDT+2)  =  ETA2
      XDIS(NDT+3)  =  -ETA1
      XDIS(NDT+4)  =  -ETA2
 1323 CONTINUE
C
C   COMPUTE AREA OF THE FLAP IF NOT GIVEN
C     FOR A SYMMETRIC WING ONLY THE AREA ON ETA.GT.0 IS CONSIDERED,
C     SO A FACTOR OF 2 IS NEEDED IF THERE IS ONLY A SINGLE FLAP
C
      AREAFP   =  AREAF
      IF(AREAF.NE.0) GO TO 133
      WRITE (W6,6002)
      WRITE (W6,6166)
      CALL INTGRT(JJMI ,JJMAX,JJMAX,ETA,STHETA,CORDF,ND,XDIS,YDIS,W6,   -
     1 OUT2, NLEFT, VALUE)
      AREAFP   =  B2**2*VALUE
      IF (UNSYM.EQ.0.AND.NFLAPS.EQ.1)  AREAFP  =  2.0*AREAFP
C
      CALL INTGRT (JJMAX, JJMAX, JJMAX, ETA, STHETA, CORDF, NLEFT, YDIS,-
```

```
     1ZDIS, W6,OUT, IND, VALUE)
      AREAF=VALUE*B2**2
      IF (UNSYM.EQ.0.AND.NFLAPS.EQ.1)  AREAF    =  2.0*AREAF
C
  133 CONTINUE
      DIFAF     =  ABS(AREAF - AREAFP)
C
C   COMPUTE FLAP CBAR IF IT IS NOT SUPPLIED
C
      CBARFP    =  CBARF
      IF(CBARF.NE.0) GO TO 134
      WRITE (W6,6002)
      WRITE (W6,6168)
      CALL INTGRT(JJMI ,JJMAX,JJMAX,ETA,STHETA,CFSQ,ND,XDIS,YDIS,W6,    -
     1 OUT2, NLEFT, VALUE)
      CBARFP    =  B**3/(8.0*AREAF )*VALUE
      IF (UNSYM.EQ.0.AND.NFLAPS.EQ.1)  CBARFP   =  2.0*CBARFP
C
      CALL INTGRT (JJMAX, JJMAX, JJMAX, ETA, STHETA, CFSQ,  NLEFT, YDIS,-
     1ZDIS, W6,OUT, IND, VALUE)
      CBARF     =  B**3/(8.0*AREAF )*VALUE
      IF (UNSYM.EQ.0.AND.NFLAPS.EQ.1)  CBARF    =  2.0*CBARF
C
  134 CONTINUE
      DIFCBF    =  ABS(CBARF - CBARFP)
      AFAW=AREAF/AREAW
      CBFCBW=CBARF/CBARW
      DIFCBF    =  ABS(CBARF - CBARFP)
      DIFAF     =  ABS(AREAF - AREAFP)
      LAMDAC    =  ATAN2 ((XSI2 - XSI1), (ETA2 - ETA1))
      LAMDCD=LAMDAC*R2D
C
C     STORE FLAP CORNERS IN ETAF(2,6)
C
      ETAF(1,1)=ETA1R
      ETAF(2,1)=ETA2
      ETAF(1,2)=-ETA2
      ETAF(2,2)=-ETA1R
  135 CONTINUE
```

```
C
C     COMPUTE TANGENTS OF SWEEP ANGLES OF LEADING AND TRAILING EDGES
C
      DYLMAX    =  XSILIP(JJMAX) - XL(NL)/B2
      DXLMAX    =  ETA(JJMAX) + 1.0D0
      DYL1      =  XSILIP(1) - XL(1)/B2
      DXL1      =  1.0D0 - ETA(1)
      DYTMAX    = -XT(NT)/B2 + XSILIP(JJMAX) + CORDIP(JJMAX)
      DYT1      =  XSILIP(1) + CORDIP(1) - XT(1)/B2
      LAMLEL(JJMAX) = DYLMAX/DXLMAX
      LAMLER(1)     = DYL1/DXL1
      LAMTEL(JJMAX) = DYTMAX/DXLMAX
      LAMTER(1)     = DYT1/DXL1
C
      DO 136 M=1,JJMAX1
C
      DYLLL     =  XSILIP(M) - XSILIP(M+1)
      DXLLL     =  ETA(M) - ETA(M+1)
      DYLTL     =  XSILIP(M) + CORDIP(M) - XSILIP(M+1) - CORDIP(M+1)
      LAMLEL(M)=  DYLLL/DXLLL
      LAMTEL(M)=  DYLTL/DXLLL
      LAMLER(M+1)= -LAMLEL(M)
      LAMTER(M+1)= -LAMTEL(M)
C
  136  CONTINUE
C
C     NC         = NUMBER OF CHORDWISE POINTS AT WHICH PRESSURES WILL BE COMPUTED
C
      IF(CPRESS .EQ.0.OR.NFLAPS.EQ.0)  GO TO 1700
      READ (R5,5003) NC
      IF (NC.LT.1.OR.NC.GT.50) NC = 49
C
C.....COMPUTING THE PRESSURE COEFFICIENT FOR THE FLAP
C.....MODE OF PRESSURE AT THE SPANWISE CONTROL POINT LOCATIONS
C
      WRITE (W6,6001)
      WRITE (W6,6004)
      DTHETA = PI/FLOAT(NC+1)
      DO 1520 N = 1, NC
```

```
      THETA = FLOAT(N)*DTHETA
      X(N) = -COS(THETA)
      CHI(N) = (X(N)+1.)/2.
 1520 SQX(N) = SIN(THETA)
      WRITE (W6,6035) (CHI(N),N=1,NC)
      WRITE (W6,6036) (X(N),N=1,NC)
      PAIR = NFLAPS .GT. 1
      COSLC = COS(LAMDAC)
      CON1= SQRT(1.-(MACH*COSLC)**2)/COSLC
      CON2=-2./(PI*CON1)
      SIGN=1.
      WRITE (W6,6001)
      WRITE (W6,6004)
      WRITE (W6,6038)
 1550 CONTINUE
      DO 1600 J = NRATIO,JJMAX,  NRATIO
      ETAJ = ETA(J)
      WRITE (W6,6037) ETAJ
      DUMY = 2./CORDIP(J)*CON1
      Y1R = DUMY*(ETAJ - ETA1R)
      Y2R = DUMY*(ETAJ - ETA2)
      Y1L = -DUMY*(ETAJ+ETA1R)
      Y2L = -DUMY*(ETAJ+ETA2)
      XC = 2.*CHIFPI(J)- 1.
      SQRTXC = SQRT(1.-XC*XC)
      DUMY = CON2*C2IP(J)/SQRTXC
      DO 1560 N = 1, NC
      XNXC = X(N)- XC
      IF(ABS(XNXC).LT.1.E-7) XNXC=1.E-7
      T1R = Q(XNXC, -Y1R)
      T2R = Q(XNXC, -Y2R)
      T1L = Q(XNXC, -Y1L)
      T2L = Q(XNXC, -Y2L)
      DCP(N) = DUMY * SQX(N) * ALOG( T1R/T2R )
 1560 IF(PAIR) DCP(N) = DCP(N) + DUMY*SQX(N)*ALOG(T1L/T2L)*SIGN
      WRITE (W6,6041) (DCP(N),N=1,NC)
 1600 CONTINUE
      IF (.NOT.PAIR .OR. SIGN.LT.0.) GO TO 1700
      WRITE (W6,6001)
```

```
      WRITE (W6,6004)
      WRITE (W6,6039)
      SIGN = -1.
      GO TO 1550
 1700 CONTINUE
C
C     LINE PRINTER OUTPUT
C
      WRITE (W6,6001)
      WRITE (W6,6004)
      WRITE (W6,6005)
      WRITE (W6,6020)
      WRITE (W6,6009) AREAW, DIFAW
      WRITE(W6,6301) BREF
      WRITE (W6,6008) B2
      BRAT=BREF/B2
      WRITE(W6,6302)BRAT
      WRITE (W6,6011) CBARW, DIFCB
      WRITE (W6,6010) CORDSS
      WRITE (W6,6012) AR, DIFAR
      WRITE (W6,6013) TR
      WRITE (W6,6014) SSCBAR, DIFSS
      WRITE (W6,6015) CBARCD, DIFGC
      WRITE(W6,6151) MACH
      WRITE (W6,6002)
      WRITE (W6,6042)
      WRITE (W6,6001)
      WRITE (W6,6002)
      WRITE (W6,6016) LESCAL
      WRITE (W6,6017) (YL(I),XL(I),I=1,NL )
      WRITE (W6,6002)
      WRITE (W6,6018) TESCAL
      WRITE (W6,6017) (YT(I),XT(I),I=1,NT )
      IF (NDL.EQ.0)  GO TO 1801
      IF (LETYPE.EQ.0)  GO TO 1801
      WRITE (W6,6001)
      WRITE (W6,6002)
      WRITE (W6,6170)
      WRITE (W6,6002)
```

```
      WRITE (W6,6172) (YDL(I),I=1,NDL)
 1801 CONTINUE
      IF (NDT.EQ.0)  GO TO 1802
      IF (TETYPE.EQ.0)  GO TO 1802
      IF (LETYPE.EQ.0)WRITE (W6,6001)
      WRITE (W6,6002)
      WRITE (W6,6171)
      WRITE (W6,6002)
      WRITE (W6,6172) (YDT(I),I=1,NDT)
 1802 CONTINUE
      IF (TRNANG.EQ.0.) GO TO 1803
      WRITE (W6,6043)
      IF(TRNFRM.LT.0) WRITE(W6,6044) TRNANG
      IF(TRNFRM.EQ.0) WRITE(W6,6045) TRNANG
      IF(TRNFRM.GT.0) WRITE(W6,6046) TRNANG
 1803 CONTINUE
      IF (NFLAPS.EQ.0) GO TO 1249
C
C     PRINT THE FLAP DATA
C
      WRITE (W6,6001)
      WRITE (W6,6004)
      WRITE (W6,6112)
      WRITE (W6,6108) LAMDCD
      WRITE (W6,6134) AFAW, CBFCBW
      WRITE (W6,6135) AREAF, DIFAF, CBARF, DIFCBF
      IF(NFLAPS.EQ. 2) GO TO 1247
      IF (UNSYM.EQ.0)  GO TO 1246
C
C     UNSYMMETRICAL WING - SINGLE FLAP
C
      WRITE (W6,6161)
      WRITE (W6,6115) ETA1R, ETA2, CHI1, CHI2
      WRITE (W6,6117) CYF(1), CYF(2), XSI1, XSI2
      GO TO 1249
 1246 CONTINUE
C
C     SINGLE SYMETRICAL FLAP
C
```

```
      WRITE (W6,6132) ETA1R, ETA1, ETA2, CHI2, CHI1, CHI2
      WRITE (W6,6133) XSI2, XSI1, XSI2, CYF(2), CYF(1), CYF(2)
      GO TO 1249
 1247 CONTINUE
C
C     SYMMETRICAL WING - TWO FLAPS
C
      WRITE (W6,6129)
      WRITE (W6,6115) ETA1R, ETA2, CHI1, CHI2
      WRITE (W6,6117) CYF(1), CYF(2), XSI1, XSI2
      WRITE (W6, 6131)
      WRITE (W6,6115) ETAF(1,2), ETAF(2,2), CHI2, CHI1
      WRITE (W6,6117) CYF(2), CYF(1), XSI2, XSI1
 1249  CONTINUE
      IF(IPINFO .EQ. 0) GO TO 1255
      WRITE (W6,6001)
      WRITE (W6,6004)
      IF (UNSYM.EQ.0)  JJMP = JJM2
      IF (UNSYM.NE.0)  JJMP = JJMAX
      IF (NFLAPS.EQ.0)  GO TO 1251
      WRITE (W6,6107)
      DO 1250 J=1,JJMP
      ETAJ=ETA(J)
      WRITE (W6,6105) ETAJ, XSILIP(J), CORDIP(J), CHIFPI(J), C2IP(J),   -
     1     LAMLEL(J), LAMLER(J), LAMTEL(J), LAMTER(J)
 1250 CONTINUE
      GO TO 1255
 1251 CONTINUE
      WRITE (W6,6158)
      DO 1252 J = 1, JJMP
      ETAJ=ETA(J)
      WRITE (W6,6105) ETAJ, XSILIP(J), CORDIP(J),         LAMLEL(J), LAML-
     1ER(J), LAMTEL(J), LAMTER(J)
 1252 CONTINUE
 1255 CONTINUE
      WRITE (W6,6001)
      WRITE (W6,6002)
      WRITE (W6,6109)
      DO 1260 P=1,PP
```

```
      WRITE (W6,6105) CHICP(P)
 1260 CONTINUE
      WRITE (W6,6004)
      WRITE (W6,6110)
      WRITE (W6,6002)
      DO 1300 N=1,NF
      INDEX=NINDEX(N)*NRATIO
      ETAN = ETA(INDEX)
      WRITE (W6,6105) ETAN
 1300 CONTINUE
      WRITE(W6,6305) JJMAX
C
C     UNIT 7 OUTPUT
C
      IF (NO7 .NE. 0) WRITE (W6,6303)
      IF (NO7 .NE. 0) GO TO 213
C
C.....GEMFIL IS A SUBROUTINE WHICH COMPUTES THE GEOMETRY FILE NAME AND
C.....ISSUES APPROPRIATE DDEF COMMANDS.  THIS SUBROUTINE IS TO BE USED ONLY
C.....ON THE AMES' TSS COMPUTER SYSTEM.
C
      CALL GEMFIL(ID)
      REWIND W7
      IF (NDL .LE. 0) GO TO 214
      DO 211 I = 1, NDL
      YDL(I)   =  YDL(I)/B2
  211  CONTINUE
  214   NDL = MAX0 (NDL, 1)
      IF (NDT .LE. 0) GO TO 215
      DO 212 I = 1, NDT
      YDT(I)   =  YDT(I)/B2
  212  CONTINUE
  215   NDT = MAX0(NDT,1)
      WRITE (W7) ID, PP, NF, CWTYPE, SWTYPE, UNSYM, NDL, NDT, NN, JJMAX,-
     1         NFLAPS, TITLE, NTITL
      C=CBARBR
      BR=BRAT
      K=NF
      P=PP
```

```
      N=NDL
      L=NDT
      J=JJMAX
      WRITE(W7)(CHICP(I),I=1,P),(NINDEX(I),I=1,K),(LAMLEL(I),I=1,J),(LAM-
     1LER(I),I=1,J),(LAMTEL(I),I=1,J),(LAMTER(I),I=1,J),                 -
     1(ETA(I),I=1,J),(STHETA(I),I=1,J),(XSILIP(I),I=1,J),(CORDI          -
     1P(I),I=1,J),BR,C,AR,TR,MACH,(YDL(I),I=1,N),(YDT(I),I=1,L)
      IF (NFLAPS.NE.0)WRITE(W7) MACH, LAMDAC, CBFCBW, AFAW, ETA1, ETA2, -
     1XLF, CYF,XSI1, XSI2,             (CHIFPI(I),I=1,JJMAX), (C2IP(I),I=1-
     1,JJMAX)
      ENDFILE W7
      WRITE(W6,6304)
  213 CONTINUE
C
C     DOUNWASH VALUES FOR UNIT 8
C     ALFAB    = ARRAY OF DOWNWASH VALUES
C
      WRITE (W6,6001)
      NSYM     =  0
      NASYM    =  0
      DO 179 I = 1, 10
      IF (BCS(I))  NSYM  =  NSYM + 1
      IF (BCAS(I))  NASYM  =  NASYM + 1
  179  CONTINUE
      IF (NSYM.EQ.0.AND.NASYM.EQ.0)  WRITE (W6,6174)
      IF (NSYM.EQ.0.AND.NASYM.EQ.0)  GO TO 198
C
      LOG1 = NO8 .EQ. 0
      ID3=0
      MMP=(NF+1)/2
      IF(UNSYM.NE.0)MMP=MM
      MMPA=NF/2
      IF(UNSYM.NE.0)MMPA=MM
      DO 180 M=1,MM
      INDEX=NINDEX(M)
      ETACP(M)=ETA(INDEX)
180   CONTINUE
      IF(.NOT.LOG1) GO TO 1809
C
```

```
C.....BCFIL IS A SUBROUTINE WHICH COMPUTES THE BOUNDARY CONDITION FILE NAME
C.....AND ISSUES APPROPRIATE DDEF COMMANDS.  THIS SUBROUTINE IS TO BE USED ONLY
C.....ON THE AMES' TSS COMPUTER SYSTEM.
C
      ID3=0
      CALL BCFIL(ID1,ID3)
      IF(LOG1) REWIND W8
      IF (LOG1)WRITE (W8) ID,ID3,TITLE, UNSYM, NSYM, NASYM, BCS, BCAS,
     1PP,CWTYPE,NF,MMP,MMPA,SWTYPE,(CHICP(P),P=1,PP),
     2(NINDEX(M),M=1,MMP),(ETACP(M),M=1,MMP)
C
1809  CONTINUE
      WRITE (W6,6002)
      IF (LOG1)WRITE (W6,6175)
      IF  (.NOT.LOG1)WRITE (W6,6300)
      WRITE (W6,6002)
      NMAX      = NF
      IF (UNSYM.EQ.0) NMAX = (NF+1)/2
      IF (.NOT.BCS(1)) GO TO 182
C
C     UNIFORM DOWNWASH MODE
C
      DO 181 N = 1, NMAX
      DO 181 P = 1, PP
      ALFAB(P,N)  = 1.0
  181  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6152)
      WRITE (W6,6104)
      ASSIGN 182 TO LOBO
      GO TO 301
  182  CONTINUE
      IF (.NOT.BCS(2)) GO TO 184
C
C     PITCHING MODE
C
      DO 183 N = 1, NMAX
      DO 183  P = 1, PP
```

```
      JRA       =  NRATIO*NINDEX(N)
      ALFAB(P, N) = (XSILIP(JRA) + CHICP(P)*CORDIP(JRA))*SSCBAR
  183  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6153)
      WRITE (W6,6104)
      ASSIGN 184 TO LOAD
      GO TO 301
  184  CONTINUE
      IF (.NOT.BCS(3)) GO TO 186
C
C     LINEAR SYMMETRIC TWIST
C
      DO 185 N = 1, NMAX
      DO 185 P = 1, PP
      JRA       =  NRATIO*NINDEX(N)
      ALFAB (P,N) =DABS(ETA(JRA))
  185  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6157)
      WRITE (W6,6104)
      ASSIGN 186 TO LOAD
      GO TO 301
  186  CONTINUE
      IF (.NOT.BCS(4)) GO TO 188
C
C     CALL TO FLPDOWN
C
      CALL        FLPDOWN(CWTYPE, LL, PP, NF, MACH, NFLAPS, NN, JJMAX,TITL-
     1E, LAMDAC, NINDEX, CHICP, LAMLEL, LAMLER, LAMTFL, LAMTER,ETAF,     -
     1R5,W6,ETA,                                                          -
     1 STHETA, XSILIP, CORDIP, CHIFPI, C2IP, ALPHA)
      IF (NFLAPS.EQ.1) GO TO 1872
      DO 1871 N = 1, NMAX
      NFN       =  NF + 1 - N
      DO 1871 P = 1, PP
      ALFAB(P,N)  = -(ALPHA(LL,P,N) + ALPHA(LL,P,NFN))
```

```
 1871 CONTINUE
      GO TO 187
 1872 CONTINUE
      IF (UNSYM.EQ.0) GO TO 1874
      DO 1873 N = 1, NMAX
      DO 1873 P = 1, PP
      ALFAB(P,N)  =-ALPHA(LL,P,N)
 1873 CONTINUE
      GO TO 187
 1874 CONTINUE
      DO 1875 N = 1, NMAX
      DO 1875 P = 1, PP
      ALFAB(P,N)  =-ALPHA(LL,P,N)
 1875 CONTINUE
  187  CONTINUE
      WRITE (W6,6104)
      WRITE (W6,6154)
      WRITE (W6,6104)
      ASSIGN 188 TO LOCO
      GO TO 201
  188  CONTINUE
C
C     ALFAB(PP,NMAX) READ FROM CARDS
C
      DO 189 I = 5, 10
      IF (.NOT.BCS(I)) GO TO 189
      READ (R5,5002)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6156)
      WRITE (W6,6104)
      DO 1891 N = 1, NMAX
      WRITE (W6,6105) (ALFAB(P,N),P=1,PP)
 1891 CONTINUE
  189  CONTINUE
  190  CONTINUE
      IF (UNSYM.EQ.0) NMAX = NF/2
      IF (.NOT.BCAS(1)) GO TO 192
C
```

```
C     UNIFORM ANTI-SYMMETRIC MODE
C
      DO 191 N = 1, NMAX
      DO 191 P = 1, PP
      JRA      =  NRATIO*NINDEX(N)
      ALFAB(P,N) =  DSIGN(1.D0,ETA(JRA))
      IF(DABS(ETA(JRA)).LT.1.D-5) ALFAB(P,N)=0.
  191  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6159)
      WRITE (W6,6104)
      ASSIGN 192 TO LOBO
      GO TO 301
  192  CONTINUE
      IF (.NOT.BCAS(2)) GO TO 194
C
C     LINEAR ANTI-SYMMETRIC TWIST
C
      DO 193 N = 1, NMAX
      DO 193 P = 1, PP
      JRA      =  NRATIO*NINDEX(N)
      ALFAB(P,N) = -ETA(JRA)
  193  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6160)
      WRITE (W6,6104)
      ASSIGN 194 TO LOBO
      GO TO 301
  194  CONTINUE
      IF (.NOT.BCAS(3)) GO TO 196
C
C     CALL TO FLPDWN
C
      IF (.NOT. BCS(4))CALL       FLPDWN(CWTYPE, LL, PP, NF, MACH, NFLAP-
     1S, NN, JJMAX,TITLE,  LAMDAC, NINDEX, CHICP,LAMLEL, LAMLER, LAMTEL,-
     1 LAMTER, ETAF,RS,W6,ETA, STHETA, XSILIP, CORDIP,                -
     1C2IP,ALPHA)
```

```
      DO 195 N = 1, NMAX
      NFN      =  NF + 1 - N
      DO 195 P = 1, PP
      ALFAB(P,N)  = -(ALPHA(LL,P,N) - ALPHA(LL,P,NFN))
  195  CONTINUE
      WRITE (W6,6104)
      WRITE (W6,6155)
      WRITE (W6,6104)
      ASSIGN 196 TO LOCO
      GO TO 201
  196  CONTINUE
C
C     ALFAB(PP,NMAX) READ FROM CARDS
C
      DO 197 I =4, 10
      IF (.NOT.BCAS(I)) GO TO 197
      READ (R5,5002)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      WRITE (W6,6104)
      WRITE (W6,6156)
      WRITE (W6,6104)
      DO 1971 N = 1, NMAX
      WRITE (W6,6105) (ALFAB(P,N),P=1,PP)
 1971 CONTINUE
  197  CONTINUE
C
  198  CONTINUE
      IF (LOG1) ENDFILE W8
      STOP
C
  201  CONTINUE
      DO 1400 P = 1, PP
      DO 1400 N = 1, NMAX
      I = NRATIO * NINDEX(N)
      TDUM = CHICP(P) .GT. CHIFPI(I)
      Z = ETA(I)
      IF(TDUM .AND. (ABS(Z-ETA1) .LE. 1.E-5 .OR.ABS(Z-ETA2) .LE. 1.E-5))-
     1 ALFAB(P,N)  =  ALFAB(P,N) + 0.5
      TDUM = TDUM .AND. SNGL(ETA(I)).LT. ETA2 - 1.E-5
```

```
      TDUM = TDUM .AND. SNGL(ETA(I)).GT. ETA1 + 1.E-5
      IF(TDUM) ALFAB(P,N)  =  ALFAB(P,N) + 1.0
 1400  CONTINUE
      IF (LOG1)WRITE (W8)  ((ALFAB(P,N),P=1,PP),N=1,NMAX)
      DO 1510 N = 1, NMAX
      WRITE (W6,6105) (ALFAB(P,N),P=1,PP)
 1510 CONTINUE
      GO TO LOCO, (188,196)
  301  CONTINUE
      DO 302  N = 1, NMAX
      WRITE (W6,6105) (ALFAB(P,N),P=1,PP)
  302   CONTINUE
      GO TO LOBO, (182, 184, 186, 190, 192, 194)
C
C      FORMAT STATEMENTS
C
C
C.....THE FOLLOWING FORMAT DEPENDS ON THE INTEGER WORD LENGTH
C.....OF THE COMPUTER BEING USED.
C.....CHANGE THE FOLLOWING FORMAT AS REQUIRED BY INTEGER WORD LENGTH
 5001 FORMAT(20A4)
 5002 FORMAT(8F10.0)
 5003 FORMAT(16I5)
 5004 FORMAT(10L1)
 5005  FORMAT(2I5, 3F10.0)
C
 6001 FORMAT(1H1)
 6002 FORMAT(1H0)
 6003 FORMAT(1H )
 6004 FORMAT(////)
 6005 FORMAT(1X,40HGEOMETRICAL DATA FOR THE FLAPPED WING        )
C.....CHANGE THE FOLLOWING FORMAT AS REQUIRED BY INTEGER WORD LENGTH
 6006 FORMAT(26H0TITLE OF CONFIGURATION =   , 7X, 20A4)
 6007 FORMAT(33H0ASSIGNED IDENTIFICATION NUMBER =  ,I5)
 6008 FORMAT(37H   B2, EFFECTIVE SEMI-SPAN                 ,1H=,F12.5)
 6009 FORMAT(1X,12H  WING AREA ,24X,1H= , F12.5, 3X,1H( , F12.5,1H) )
 6010 FORMAT(1X,13H  ROOT CHORD ,23X,1H=,F12.5)
 6011 FORMAT(37H   CBAR, LONG. REF. LENGTH                 ,1H=,F12.5,3X, 1-
     1H(,       F12.5,1H) )
```

```
6012 FORMAT(  16H0  ASPECT RATIO , 21X,1H= , F12.5, 3X,1H( ,F12.5, 1H))
6013 FORMAT(1X,14H  TAPER RATIO , 22X,1H= , F12.5)
6014 FORMAT(1X,33H  BREF/CBAR                        , 3X,1H= , F12.5,   -
    1     3X,1H( , F12.5,1H) )
6015 FORMAT(1X,34H  CBAR/ROOT CHORD                  , 2X,1H= , F12.5,   -
    1       3X,1H( , F12.5,1H) )
6016 FORMAT(1X,28H  LEADING EDGE CO-ORDINATES ,5X, 16H(SCALE FACTOR =  -
    1, F10.6, 1H)//,11X, 2HYL, 11X,2HXL)
6017 FORMAT(2F13.5)
6018 FORMAT (1X,28H  TRAILING EDGE CO-ORDINATES,5X, 16H(SCALE FACTOR = -
    1 , F10.6, 1H)//,11X, 2HYT, 11X,2HXT)
6019 FORMAT(1H0/31H0FLAPPED WING GEOMETRY PROGRAM   /31H ==============-
    1================  //)
6020 FORMAT(1H0/11H0WING DATA /11H ---------- /1H0)
6021 FORMAT(//10H0CWTYPE = ,I3/10H SWTYPE = ,I3/10H IPINFO = ,I3/10H CP-
    1RESS = ,I3/10H UNSYM  = ,I3/10H TRNFRM = ,I3/10H OUT    = ,I3/10H -
    1NO7    = ,I3/10H NO8    = ,I3)
C
6035 FORMAT (     /49H0CHORDWISE STATIONS FOR THE PRESSURE COEFFICIENT -
    1          ,31H(PERCENT LOCAL CHORD FROM L.E.),//,(2P10F12.5))
C
6036 FORMAT (     /49H0CHORDWISE STATIONS FOR THE PRESSURE COEFFICIENT -
    1,45H(FRACTION OF LOCAL SEMI-CHORD FROM MID-CHORD),/(10F12.5) )
C
6037 FORMAT (     26H0SPANWISE POSITION, ETA = , F8.5)
C
6038 FORMAT (1H0,43HPRESSURE COEFFICIENT FOR THE SYMMETRIC FLAP,       -
    1 14H PRESSURE MODE,///)
C
6039 FORMAT (1X,49HPRESSURE COEFFICIENT FOR THE ANTI-SYMMETRIC FLAP,   -
    1     14H PRESSURE MODE,///)
C
6041 FORMAT(10F12.5)
6042 FORMAT (1X,51HTHE NUMBERS IN PARENTHESES ARE ERROR ESTIMATES      -
    1   /30H (SEE PROGRAM DOCUMENTATION)        )
6043 FORMAT (50H0THE ABOVE VALUES WERE DETERMINED BY APPLYING      )
6044 FORMAT (1X,2HA , F7.3,49H DEGREE CRANKING TRANSFORMATION TO THE IN-
    1PUT      )
6045 FORMAT (1X,2HA , F7.3,49H DEGREE SHEARING TRANSFORMATION TO THE IN-
```

```
     1PUT      )
 6046 FORMAT (1X,2HA , F7.3,49H DEGREE ROTATION TRANSFORMATION TO THE IN-
     1PUT      )
C
 6104 FORMAT (////)
 6105 FORMAT (9F14.5)
 6107 FORMAT (1X,44HGEOMETRICAL DATA AT THE INTEGRATION STATIONS,        -
     1 //, 10X, 3HETA, 7X, 8HX(LE)/B2, 9X, 5HC/B2, 5X,         15H(X(LE)--
     1X(FL))/C, 5X, 2HC2,         10X, 6HTANLEL, 8X, 6HTANLER,8X,         -
     1 6HTANTEL, 8X, 6HTANTER)
 6108 FORMAT(24H0HINGELINE SWEEP      =  ,F12.5, 3X, 7HDEGREES  )
 6109 FORMAT (1X,51HCHORDWISE CONTROL POINT LOCATIONS REFERENCED TO THE,-
     1            12H LOCAL CHORD,///)
 6110 FORMAT(1X,50H SPANWISE CONTROL POINT LOCATIONS REFERENCED TO B2 )
 6112 FORMAT (10H FLAP DATA  /10H --------- /1H0)
 6115 FORMAT (1X,37HSPANWISE LOCATION OF FLAP EDGES/B2   ,/,2F12.5,/,40H-
     1 RELATIVE CHORDWISE LOCATIONS OF CORNERS,/2F12.5)
 6117 FORMAT (   32H CHORD LENGTHS AT FLAP EDGES ARE,/,2F12.5,/,         -
     1  25H XSI'S OF THE CORNERS ARE,/,2F12.5)
 6129 FORMAT (1X,//,20H FOR FLAP NUMBER ONE)
 6131 FORMAT (1X,//,20H FOR FLAP NUMBER TWO)
 6132 FORMAT (1X,//,46H DATA FOR THE SINGLE FLAP (L.H.S., CENTERLINE.,  -
     1          7H R.H.S),//,25H SPANWISE POSITIONS/B2   ,/,3F15.5,//,   -
     1        52H CHORDWISE LOCATIONS (PERCENT LOCAL CHORD FROM L.E.),   -
     1      /,2P3F15.5)
 6133 FORMAT (1X,//,37H CHORDWISE LOCATIONS (ACTUAL POS/B2)  /3F15.5,    -
     1      //,14H CHORD LENGTHS,/,3F15.5)
 6134 FORMAT (24H FLAP AREA/WING AREA  = , F12.5/,          24H FLAP CBA-
     1R/WING CBAR  = , F12.5)
 6135 FORMAT (1X, 23HFLAP AREA             = , F12.5, 3X, 1H(,F12.5,1H),/-
     1,        24H CBAR OF THE FLAP     = , F12.5, 3X, 1H(, F12.5, 1H))
 6151 FORMAT (2X,12H MACH NUMBER,23X,1H=,F12.5)
 6152 FORMAT (1X, 40HUNIFORM DOWNWASH MODE                      )
 6153 FORMAT (1X, 40HPITCHING MODE                              )
 6154 FORMAT (1X, 40HCOMPUTED BY FLPDWN (SYMETRIC MODE)         )
 6155 FORMAT (1X, 40HCOMPUTED BY FLPDWN (ANTI-SYMETRIC MODE)    )
6156  FORMAT (1X, 40HDISTRIBUTIONS READ FROM CARDS              )
 6157 FORMAT (1X, 40HLINEAR SYMMETRIC TWIST                     )
 6158 FORMAT (1X,44HGEOMETRICAL DATA AT THE INTEGRATION STATIONS,        -
```

```
     1 //, 10X, 3HETA, 7X, 8HX(LE)/B2, 9X, 4HC/B2,          9X, 6HTANLEL, -
     18X, 6HTANLER,          8X, 6HTANTEL, 8X, 6HTANTER)
 6159 FORMAT (1X,    40HUNIFORM ANTI-SYMMETRIC                       )
 6160 FORMAT (1X,    40HLINEAR ANTI-SYMMETRIC TWIST                  )
 6161 FORMAT (1X, //, 40H DATA FOR A SINGLE ASYMMETRICAL FLAP        )
 6162 FORMAT (1X,40H INTEGRATIONS FOR THE WING AREA             )
 6164 FORMAT (1X,40H INTEGRATIONS FOR THE WING CBAR             )
 6166 FORMAT (1X,40H INTEGRATIONS FOR THE FLAP AREA             )
 6168 FORMAT (1X,40H INTEGRATIONS FOR THE FLAP CBAR             )
 6170 FORMAT (1X,40HTHE WING HAS KINKS IN THE  LEADING EDGE  ,          4-
     10HAT THE FOLLOWING ETA LOCATIONS                     )
 6171 FORMAT (1X,40HTHE WING HAS KINKS IN THE TRAILING EDGE  ,          4-
     10HAT THE FOLLOWING ETA LOCATIONS                     )
 6172 FORMAT (F15.5)
 6174 FORMAT (       46H0NO BOUNDARY CONDITIONS WERE STORED ON UNIT W8  )
 6175 FORMAT (      42H0THE FOLLOWING SETS OF BOUNDARY CONDITIONS        -
     1 40H WERE STORED ON UNIT W8                          )
C
 6201 FORMAT (1X, 29HBAD FLAP GEOMETRY YF1.GE.YF2, 4X, 6HYF1 = , F15.5, -
     1          4X, 6HYF2 = , F15.5)
 6202 FORMAT (1X, 42HNUMBER OF CONTROL POINTS MUST BE POSITIVE,         4-
     1X, 6HPP =  , I5, 4X, 6HNF =  , I5)
 6203 FORMAT( 50H0   ERROR---JJMAX   MUST BE MADE AN ODD NUMBER         /-
     110H JJMAX =  , I6, 10X,    5HNN = , I6     )
 6205 FORMAT (1X, 8HSTOP 350)
 6206 FORMAT(1X,54HNN MUST BE ASSIGNED A VALUE WHEN SWTYPE IS NOT ZERO  -
     1  )
 6300  FORMAT(    72H0THE FOLLOWING SETS OF BOUNDARY CONDITIONS WERE COM-
     1PUTED BUT NOT STORED   )
 6301  FORMAT (37H   BREF, LAT. REF. LENGTH                    ,1H=, F12.5)
 6302  FORMAT(11H   BREF/B2    , 26X, 1H=, F12.5)
 6303  FORMAT(//47H0NO GEOMETRY INFORMATION WAS STORED ON UNIT W7  )
 6304 FORMAT(//54H0THE GEOMETRY INFORMATION HAS BEEN STORED ON UNIT W7  -
     1     )
6305  FORMAT(///1X,7HJJMAX = ,I4/////////////)
      END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
```

END END END END END END END END END END END END END END END END

```
      SUBROUTINE CODIM (YI,XI,NI,T,ANS,NA,XK)
C
C****             A CONTROLLED DEVIATION ITERPOLATION METHOD
C
      DOUBLE PRECISION T
      DIMENSION  XI(NI), YI(NI), T(NA), ANS(NA)
C
C
      N=NI
      SIGN     =  1.0
      IF (XI(NI).LT.XI(1))  SIGN  =  -1.0
      DO 910 IE=1,NA
      X=T(IE)
  100 IF(N-2)110,120,200
  110 Y = YI(N)
      GO TO 900
  120 Y = (YI(2)-YI(1))/(XI(2)-XI(1))* (X-XI(1)) +YI(1)
      GO TO 900
  200 J = 1
 210  IF (SIGN*(XI(J) - X)) 230,220,250
  220 Y =YI(J)
      GO TO 900
  230 J = J+1
      IF(J-N)210,210,250
  250 IF(J-2)120,155,260
  155 J = 3
      JJ = 1
      GO TO 285
  260 IF(J-N)280,265,270
  265 J = N-1
      JJ = 2
      GO TO 285
  270 Y= (YI(N)-YI(N-1))/(XI(N)-XI(N-1))* (X-XI(N-1))+YI(N-1)
```

```
      GO TO 900
  280 JJ = 3
  285 IF(N-3)290,290,295
  290 J = 3
  295 K = J-1
      M = K-1
      L = J+1
      A1 = X-XI(M)
      A2 = X-XI(K)
      A3 = X-XI(J)
      AL = (X-XI(K))/(XI(J)-XI(K))
      S = AL*YI(J)+(1.0-AL)*YI(K)
      C1= A3*A2/((XI(M)-XI(K))*(XI(M)-XI(J)))
      C2= A1*A3/((XI(K)-XI(M))*(XI(K)-XI(J)))
      C3= A2*A1/((XI(J)-XI(M))*(XI(J)-XI(K)))
      P1 = C1*YI(M)+C2*YI(K)+C3*YI(J)
      IF(N-3)305,305,310
  305 P2 = P1
      GO TO 315
  310 A4 = X-XI(L)
      C4= A4*A3/((XI(K)-XI(J))*(XI(K)-XI(L)))
      C5= A2*A4/((XI(J)-XI(K))*(XI(J)-XI(L)))
      C6= A3*A2/((XI(L)-XI(K))*(XI(L)-XI(J)))
      P2 = C4*YI(K)+C5*YI(J)+C6*YI(L)
  315 GO TO (320,330,350),JJ
  320 P2 = P1
      AL = (X-XI(1))/(XI(2)-XI(1))
      S =  AL*YI(2)+ (1.0-AL)*YI(1)
      P1=  S + XK*(P2-S)
      GO TO 350
  330 P1 = P2
      AL = (X-XI(N-1))/(XI(N)-XI(N-1))
      S  = AL*YI(N) +(1.0-AL)*YI(N-1)
      P2 = S+ XK*(P1-S)
  350 E1 =  ABS(P1-S)
      E2 =  ABS(P2-S)
      IF(E1+E2)400,400,410
  400 Y = S
      GO TO 900
```

```
  410 BT =  (E1*AL)/(E1*AL+(1.0-AL)*E2)
      Y  = BT*P2+(1.0-BT)*P1
  900 ANS(IE)=Y
  910 CONTINUE
      RETURN
      END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END



      SUBROUTINE STRAIT(FIN,XIN,NIN,XOUT,FOUT,NOUT,NRITE)
C
C THIS IS A SUBROUTINE TO DETERMINE THE FUNCTION FOUT AT THE
C XOUT LOCATIONS USING LINEAR INTERPOLATION FROM THE
C (XIN,FIN) TABLE.
C
C XIN MUST EITHER BE IN ASCENDING OR DESCENDING ORDER.
C XOUT MAY BE IN ANY NUMERICAL ORDER.
C NIN MAY BE 1, IN WHICH CASE THE VALUE FIN(1) IS ASSIGNED TO (FOUT).
C
C*****  THIS VERSION IS PARTIALLY DOUBLE PRECISION  *****
C
      DOUBLE PRECISION XOUT,X
      REAL LINIT
      DIMENSION XIN(NIN),FIN(NIN),XOUT(NOUT),FOUT(NOUT)
      LINIT(X,X1,X2,F1,F2)=F1+(F2-F1)/(X2-X1)*(X-X1)
C
C
C
      IF(NIN.EQ.1) GO TO 600
      SIGN=1.
      IF(XIN(1).GT.XIN(2))SIGN=-1.
      NUPR=NIN-1
      DO 500 NO=1,NOUT
      IF(SIGN*XOUT(NO).GT.SIGN*XIN(2)) GO TO 100
```

```
      FOUT(NO) = LINIT(XOUT(NO),XIN(1),XIN(2),FIN(1),FIN(2))
      GO TO 500
100   IF(SIGN*XOUT(NO).LT.SIGN*XIN(NUPR)) GO TO 200
      FOUT(NO) = LINIT(XOUT(NO),XIN(NUPR),XIN(NIN),FIN(NUPR),FIN(NIN))
      GO TO 500
200   CONTINUE
      DO 300 NI=3,NUPR
      IF(SIGN*XOUT(NO).GE.SIGN*XIN(NI)) GO TO 300
      FOUT(NO) = LINIT(XOUT(NO),XIN(NI-1),XIN(NI),
     1FIN(NI-1),FIN(NI))
      GO TO 500
300   CONTINUE
C
      WRITE(NRITE,1) XIN
1     FORMAT(1H1,   'TABLE NOT IN ASCENDING OR DESCENDING ORDER IN'
     1,' SUB. STRAIT'/(1X,1PE20.5))
      STOP 13
C
500   CONTINUE
      RETURN
600   DO 610 NO=1,NOUT
610   FOUT(NO)=FIN(1)
      RETURN
      END
END   END   END   END   END   END   END   END   END   END   END   END   END   END   END   END
END   END   END   END   END   END   END   END   END   END   END   END   END   END   END   END
END   END   END   END   END   END   END   END   END   END   END   END   END   END   END   END
END   END   END   END   END   END   END   END   END   END   END   END   END   END   END   END


      SUBROUTINE INTGRT (JJ, JJR, JJMAX, ETA, STHETA, F, ND, X, Y,W,
     1OUTR, NLEFT, VALU)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C  PROGRAM TO INTEGRATE F(X)*SIN(X) FROM 0 TO PI BY A TRAPEZOIDAL RULE        C
C  MODIFIED TO ALLOW FOR SPECIFIED DISCONTINUITIES.                            C
C   THE DISCONTINUITIES MAY BE DISCONTINUITIES IN SLOPE OR VALUE.              C
```

```
C    IF ANY DISCONTINUITY IS A DISCONTINUITY IN VALUE AND THE LOCATION         C
C    OF THE DISCONIUITY COINCIDES WITH AN INTEGRATION POINT, THE PROGRAM       C
C    ASSUMES THAT THE CORRESPONDING ELEMENT OF F IS FILLED WITH THE LIMIT      C
C    AVERAGE VALUE.                                                            C
C                                                                              C
C   THE PROGRAM IS WRITTEN TO BE MACHINE INDEPENDENT                           C
C                                                                              C
C                                                                              C
C      JJ         = NUMBER OF STATIONS TO USE IN THE INTEGRATION               C
C      JJR        = NUMBER OF VALUES OF F                                      C
C      JJMAX      = NUMBER OF VALUES OF ETA AND STHETA                         C
C      ETA        = COS(THETA) AT JJMAX STATIONS                               C
C      STHETA     = SIN(THETA) AT JJMAX STATIONS                               C
C      F          = THE FUNCTION   VALUES AT JJR STATIONS                      C
C      ND         = THE NUMBER OF DISCONTINUITIES                              C
C      X          = A VECTOR CONTAINING THE DISCONTINUITIES                    C
C      Y          = THE X VECTOR AFTER REORDERING FROM HIGH TO LOW AND         C
C                   ELIMINATING VALUES OUT OF RANGE, DUPLICATED VALUES,        C
C                   AND VALUES WITHIN EPS OF ANY INTEGRATION STATION.          C
C      OUTR       = AN INTEGER GOVERNING THE OUTPUT OF THIS PROGRAM.           C
C      NLEFT      = THE NUMBER OF Y-VALUES.                                    C
C      VALU       = THE OUTPUT VALUE OF THE INTEGRAL                           C
C                                                                              C
C                                                                              C
C                                                                              C
C    ALL THE INPUT VARIABLES ARE RETURNED                                      C
C                                                                              C
C    THE X-VALUES ARE COMPLETELY UNRESTRICTED, BUT THE CORRECTIONS WILL        C
C    NOT NECESSARILY BE DONE CORRECTLY IN ALL CASES.                           C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C.....SPECIFICATION STATEMENTS
      DOUBLE PRECISION ETA, SUM, PI, TD , STHETA
      INTEGER OUTR, OUT, W
      LOGICAL CONT, CONT1
      DIMENSION ETA(JJMAX), STHETA(JJMAX), F(JJR), X(ND), Y(ND)
      DIMENSION FT(4), FUN(4)
      EQUIVALENCE (YTEMP,YDUM)
```

```
C
      DATA PI /3.141592653589793      D0/
      DATA EPS /1.E-5/
C
C.....JJ, JJR, AND JJMAX MUST LINE UP PROPERLY.  IF THEY DO NOT THE
C.....PROGRAM WILL WRITE A MESSAGE TO THAT EFFECT AND HALT EXECUTION.
C
      IF(MOD(JJMAX+1, JJR+1) .NE. 0) GO TO 910
      IF(MOD(JJR  +1, JJ +1) .NE. 0) GO TO 910
      JJ1=JJ+1
      JR1 =(JJR+1)/JJ1
      JR2 =(JJMAX+1)/JJ1
      DELTA = PI/DFLOAT(JJ1)
      SUM=0
      OUT= OUTR
C
C.....THE 50 LOOP DOES THE UNCORRECTED INTEGRATION
C
      DO 50 J=1,JJ
      J1=J*JR1
      J2=J*JR2
      SUM=SUM+STHETA(J2)*F(J1)
50    CONTINUE
      VALU  = DELTA*SUM
C
C.....IF THERE DISCONTINUITIES, TRANSFER TO 200.
C
      IF(ND.GT.0) GO TO 200
60    NLEFT=0
      IF(OUT.LT.1) RETURN
      WRITE (W,1)VALU , JJ
      IF(OUT .LT. 2) RETURN
      IF(ND .EQ. 0) GO TO 70
      WRITE(W,2)
      WRITE(W,3) X
70    IF(OUT.LT.3) RETURN
      WRITE(W,4)
      DO 75 J=1,JJ
      J1=J*JR1
```

```
      J2=J+JR2
75    WRITE (W,5) ETA(J2), F(J1)
      RETURN
C-----------------------------------------------------------------------
200   NLEFT = ND
C
C.....PRIOR TO CORRECTING FOR DISCONTINUITIES IN SLOPE AND VALUE THE
C.....EXTRANEOUS POINTS ARE ELIMINATED FROM CONSIDERATION.  A TRANSFER
C.....BACK TO 60 WILL OCCUR IF THERE ARE NO REAL DISCONTINUITIES.
C
C
C.....ARRANGING X IN Y IN DESCENDING ORDER
C
      DO 210 N=1,ND
210   Y(N)= X(N)
      DO 220 N=1,ND
      DO 220 J=N,ND
      IF(Y(N).GE.Y(J))GO TO 220
      YTEMP=Y(J)
      Y(J)=Y(N)
      Y(N)=YTEMP
220   CONTINUE
      IF(NLEFT.EQ.1)GO TO 300
      LMIN=1
      LMAX=ND-1
C
C.....REPEATED VALUES ARE ELIMINATED NEXT
C
230   DO 240 L=LMIN,LMAX
      IF(ABS(Y(L)-Y(L+1)).LT.EPS)GO TO 250
240   CONTINUE
      GO TO 260
250   CALL CRUNCH (Y,L,LMAX)
      IF(L.GT.LMAX) GO TO 260
      LMIN=L
      GO TO 230
260   NLEFT = LMAX+1
300   CONTINUE
C
```

```
C.....TOO SMALL Y-VALUES ARE ELIMINATED NEXT
C
      YDUM= EPS-1.
      DO 310 L=1,NLEFT
      IF( Y(L).LT.YDUM) GO TO 320
310   CONTINUE
      GO TO 330
320   NLEFT=L-1
      IF(NLEFT.EQ.0) GO TO 60
330   IF(Y(NLEFT).GT.-YDUM) GO TO 60
400   IF(NLEFT. EQ. 1) GO TO 500
      LMAX=NLEFT-1
      YDUM=-YDUM
C
C.....TOO LARGE Y-VALUES ARE ELIMINATED NEXT
C
410   IF(Y(1).LE.YDUM) GO TO 440
      CALL CRUNCH(Y,1,LMAX)
      GO TO 410
440   NLEFT = LMAX+1
500   LMIN=1
C
C.....Y-VALUES NEARLY COINCIDENT WITH INTEGRATION STATIONS ARE ELIMINATED NEXT
C
      DO 530 J=JR2,JJMAX,JR2
      ES =  ETA(J)
      DO 510 L=LMIN, NLEFT
      IF(ABS(Y(L)-ES).LT. EPS) GO TO 520
510   CONTINUE
      GO TO 530
520   CALL CRUNCH (Y,L, NLEFT)
      IF(NLEFT.EQ.0)GO TO 60
      IF(L.GT.NLEFT)GO TO 540
      LMIN = L
530   CONTINUE
C
C.....AT THIS POINT ALL THE EXTRANEOUS DISCONTINUITIES HAVE BEEN
C.....ELIMINATED AND THERE ARE STILL SOME REMAINING.  THE REMAINING DIS-
C.....CONTINUITIES ARE STORED IN (Y).
```

```
C
540   KOUNT = 0
      CONT = .FALSE.
      VALU1=VALU
      IF(JJ.LT. 4) GO TO  800
C
C.....AT THIS POINT THE PROGRAM WILL MAKE THE NECESSARY CORRECTIONS.
C.....IT WILL NOT BE ABLE TO DO THIS RIGHT UNDER CERTAIN CONDITIONS.
C.....IF THERE ARE DISCONTINUITIES IN THE FIRST OR LAST TWO INTERVALS,
C.....OR MORE THAN ONE DISCONTINUITY IN AN INTERVAL, OR DISCONTINUITIES
C.....IN ADJACENT INTERVALS, THE CORRECTIONS WILL NOT BE DONE RIGHT.
C
      ET(1)=ETA(JR2)
      ET(2)=ETA(2*JR2)
      ET(3)=ETA(3*JR2)
      CORR=0
      CONT1=.FALSE.
      JMAX=JJ-3
      LMIN=1
      DO 700 J=1,JMAX
      INDEX=(J+3)*JR2
      ET(4)=ETA(INDEX)
      DO 600 L= LMIN,NLEFT
      IF(Y(L) .GT.ET(2)) GO TO 600
      IF(Y(L) .GT.ET(3)) GO TO 610
      CONT1=.FALSE.
      GO TO 690
600   CONTINUE
      GO TO 790
610   INDEX=0
      JP3=J+3
      DO 620 K=J,JP3
      K1=K*JR1
      K2=K*JR2
      INDEX=INDEX+1
620   FUN(INDEX)=F(K1)*STHETA(K2)
      TD = DARCOS(DBLE(Y(L)))
      D2= TD/DELTA-DFLOAT(J+1)
      D3 = 1. - D2
```

```
      CORR = CORR + D2**2*(FUN(2)-FUN(1)) + D3**2*(FUN(3)-FUN(4))       -
     1 + (2.*D2 - 1.)*(FUN(2)-FUN(3))
      CONT = CONT.OR.CONT1
      CONT1= .TRUE.
      KOUNT=KOUNT+1
      IF(KOUNT.EQ.NLEFT)GO TO 790
690   CONTINUE
      ET(1)=ET(2)
      ET(2)=ET(3)
      ET(3)=ET(4)
      LMIN=L
700   CONTINUE
790   VALU=VALU +CORR*DELTA/2.
C-----------------------------------------------------------------------------
C
C.....THIS IS THE END OF THE COMPUTATION.  THE REMAINDER OF THE PROGRAM
C.....IS OUTPUT.
C
800   IF (OUT .LT. 0) RETURN
      IF(KOUNT.NE.NLEFT)GO TO 810
      IF(CONT) GO TO 810
      IF(JJ.LT.4) GO TO 810
      IF (OUT.GT.0) GO TO 820
      RETURN
810   OUT =MAX0(OUT,2)
820   WRITE(W,6) VALU1,VALU,JJ
      IF (OUT.LT.2) RETURN
      IF (KOUNT.EQ.NLEFT) GO TO 830
      KOUNT = NLEFT - KOUNT
      WRITE(W,7) KOUNT
830   IF (.NOT.CONT) GO TO 840
      WRITE(W,8)
840   IF(JJ.GT.3)GO TO 850
      WRITE(W,9)
850   WRITE(W,10)
      WRITE(W,3) X
      WRITE(W,11)
      WRITE(W,3) (Y(N),N=1,NLEFT)
      GO TO  70
```

```
910    WRITE (W, 12) JJ, JJR, JJMAX
       STOP
1      FORMAT(25H0VALUE OF THE INTEGRAL IS  ,F15.7 , 10X,                -
      140H THE NUMBER OF INTEGRATION POINTS USED =  ,I5)
2      FORMAT(54H0NONE OF THE X-VALUES WERE CONSIDERED DISCONTINUITIES  /-
      117H THE X VECTOR IS )
3      FORMAT (  6F20.6 )
4      FORMAT (10H0       ETA   , 10X, 1HF)
5      FORMAT ( F12.6, F13.6)
6      FORMAT (                                                          -
      141H0THE UNCORRECTED VALUE OF THE INTEGRAL =    ,F15.7/            -
      241H THE   CORRECTED VALUE OF THE INTEGRAL =    ,F15.7/            -
      341H THE NUMBER OF INTEGRATION POINTS WAS  =    ,I7   )
7      FORMAT(25H0---WARNING---THERE WERE      , I5,                      -
      152H DISCONTINUITIES WHICH COULD NOT BE CORRECTED FOR.            )
8      FORMAT ( 80H0---WARNING---THERE WERE DISCONTINUITIES IN CONTIGUOUS-
      1 INTEGRATION INTERVALS         )
9      FORMAT (100H0---WARNING---NO DISCONTINUITIES ARE CORRECTED FOR WHE-
      IN THERE ARE LESS THAN 4 INTEGRATION POINTS           )
10     FORMAT (26H0ORIGINAL DISCONTINUITIES    )
11     FORMAT (25H0REDUCED DISCONTINUITIES    )
12     FORMAT(53H0JJ, JJR, AND JJMAX ARE NOT PROPERLY RELATED.          /-
      19H0    JJ =   , I7/                                               -
      19H    JJR =   , I7/                                               -
      19H  JJMAX =   , I7/                                               -
      134H0EXECUTION TERMINATED IN INTGRT                  )
       END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END


..RJSTART RMT05
       SUBROUTINE CRUNCH (Y, L, LMAX)
       DIMENSION Y(1)
       DO 10 K = L, LMAX
```

```
10    Y(K) = Y(K+1)
      LMAX = LMAX - 1
      RETURN
      END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END


      FUNCTION C2I(ETA,ETA2)
C.....THE FIRST 3 DERIVATIVES W.R.T. ETA VANISH AT ETA2.
      AE2=ABS(ETA2)
      AE=ABS(ETA)
      C2I=1.
      IF(AE.LE.AE2)RETURN
      Z=(AE-AE2)/(1.-AE2)
      Z=Z*Z
      C2I=SQRT(1.-Z)*(1.+Z/2.)
      RETURN
      END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END


      FUNCTION Q(X,Y)
C
C.....THIS FUNCTION IS FOR COMPUTING ACCURATELY THE VALUE OF THE EXPRES-
C.....SION  SQRT(X**2+Y**2) + Y  EVEN WHEN Y .LT. 0. AND ABS(X/Y) IS
C.....MUCH SMALLER THAN 1.
C
      SIGN = +1.
      GO TO 5
```

```
      ENTRY QMINUS(X,Y)
C
C.....ENTRY HERE COMPUTES SQRT(Y**2 - X**2) + Y FOR THE SAME CONDITIONS.
C
      SIGN = -1.
5     IF (Y.GE.0.)GO TO 10
      IF(ABS(X/Y).GT. .15) GO TO 10
      R= (X/Y)**2*SIGN
C
C.....TAYLOR'S EXPANSION OF -Y*(SQRT(1+R**2)-1).
C
      Q= -Y*R/2.*(1. - R/4.*(1. - R/2.*(1. - .625*R)))
      RETURN
10    Q = SQRT( Y*Y + SIGN*X*X) + Y
      RETURN
      END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END
END  END  END  END  END  END  END  END  END  END  END  END  END  END  END  END

*888888888888888888888888888888888888888888888888888888888888888888888888888
*****                                                                   *****
***** DELETE CALLS TO CVRT,OBEY,BCFIL, AND GEMFIL FOR OTHER THAN        *****
***** AMES' VERSION OF THE PROGRAM                                      *****
*****                                                                   *****
*****************************************************************************
*****************************************************************************
```