2up

**NASA TECHNICAL MEMORANDUM**

**NASA TM X- 62,323**

NASA TM X- 62,323

# BOUNDARY CONDITION PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY


Richard T. Medan and K. Susan Ray

Ames Research Center
Moffett Field, Calif. 94035
and
Computer Sciences Corporation
Mt. View, Calif. 94043





November 1973



(NASA-TM-X-62323) BOUNDARY CONDITION PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY (NASA) 67 p HC $5.50 CSCL 20D

N74-16699

Unclas
G3/01 29339

TABLE OF CONTENTS

BOUNDARY CONDITION PROGRAM FOR

AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan
Ames Research Center
and
K. Susan Ray
Computer Sciences Corporation

## ABSTRACT

This document is a description of and users manual for a USA FORTRAN IV computer program which determines boundary conditions for a thin wing lifting surface program. This program, the geometry program, and several other programs are used together in the analysis of lifting, thin wings in steady, subsonic flow according to a kernel function lifting surface theory. The program calculates specific types of boundary conditions completely automatically such as those necessary to determine pitch and roll damping derivatives. The program also accepts descriptions of the camber or downwash and twist in the form of tables and/or coefficients of equations. The program performs interpolations so that tables and/or coefficients can apply at stations selected by the user and not at stations dictated by the control point locations. The program uses information stored on a geometry file and, optionally, on an influence matrix file. The boundary conditions that it calculates are stored on a boundary condition file. An equation solving program will read the influence matrix and boundary condition files and determine the coefficients in the expansion for the lifting pressure distribution.

# BOUNDARY CONDITION PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan
Ames Research Center
and
K. Susan Ray
Computer Sciences Corporation

## 1 INTRODUCTION

This document is a description of and users manual for a USA FORTRAN IV computer program which determines boundary conditions for a thin wing lifting surface program. This program, the geometry program, and several other programs are used together in the analysis of lifting, thin wings in steady, subsonic flow according to a kernel function lifting surface theory. The program calculates specific types of boundary conditions completely automatically such as those necessary to determine pitch and roll damping derivatives. The program also accepts descriptions of the camber or downwash and twist in the form of tables and/or coefficients of equations. The program performs interpolations so that tables and/or coefficients can apply at stations selected by the user and not at stations dictated by the control point locations. The program uses information stored on a geometry file and, optionally, on an influence matrix file. The boundary conditions that it calculates are stored on a boundary condition file. An equation solving program will read the influence matrix and boundary condition files and determine the coefficients in the expansion for the lifting pressure distribution.

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

R. T. Medan
Mail Stop 247-1
Ames Research Center
Moffett Field, Ca., 94035

# 2 USER'S INSTRUCTIONS

## 2.1 INITIAL SETUP FOR AMES' TSS SYSTEM

For either batch or conversational processing the following TSS commands must be given. These commands are required once and only once for each user ID. The first three commands create the identification number file named IDFILE. This file contains four zeroes in binary form.

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

## 2.2 CONVERSATIONAL USE ON AMES' TSS SYSTEM

All integer data should be entered in a 16I5 format, all floating point data in 8F10.0 format, and all logical data in 10L1 format. See section 8 for an example of a terminal session.

USER: After logging on enter the following:

```
AMES USYSLIB
JOBLIBS SYSULIB
JBLB MEDAN
```

It isn't necessary to issue DDEFs for anything except the input data since the program issues them using the subroutines AIMFIL, GEMFIL, and BCFIL.

USER: CALL BC$

PROG: ENTER BATCH

USER: Enter carriage return for conversational mode.

PROG: ENTER INPUT DEVICE NUMBER

USER: Hit carriage return for terminal input. Otherwise input starting with (BCS) will be from a dataset on disk as referenced by the input device number. This number must be a positive integer from 1 through 99 excluding 4, 5, 6, 7, 8, 9, and 11. This dataset must be named in a DDEF statement or else AMES' TSS will expect terminal input. If this method of input is used, then the program will not give conversational prompts for the data starting with (BCS).

To terminate execution, enter a negative number.

PROG: ENTER ODISK

USER: For terminal output enter carriage return. For output to a disk file enter a positive non-zero number less than 10. For TSS the output will be found on the file named OUTPUT.BC.NX where X is the numerical value of ODISK. The program issues its own DDEF commands for the output file so no control cards are needed. This output is written on logical unit 4.

PROG: ENTER ID1,ID2

USER: Enter identification numbers.

ID1--identification number of the geometry file. If a negative number is entered, then the most recent geometry file will be used. Enter zero to terminate execution.

ID2--identification number of AIM file. Enter a nonzero number only if the control points have been changed by the influence matrix program. If a negative number is entered, then the most recent AIM file will be used. Enter zero if no AIM file has been created or if the control points have not been changed by the influence matrix program.

PROG: ENTER (BCS)/(BCAS)/PRINTA,PRINTB,NSTORE

USER: Enter the array (BCS), carriage return.
Enter the array (BCAS), carriage return.
Enter PRINTA, PRINTB, NSTORE.

Each element of (BCS) and (BCAS) corresponds to a different downwash condition. A true value means that the corresponding boundary condition will be calculated and, if NSTORE is .FALSE., then written on a file for use in the equation solving program. Both arrays are read and acted upon if the wing is symmetric or unsymmetric. In the latter case the boundary condition will be calculated for the entire wing. See section 3 for a description of the types of cases referenced by (BCS) and (BCAS) and for auxilliary input required for each element of (BCS) and (BCAS).

PRINTA is a logical variable which should be set to .TRUE. if all the downwash modes which are calculated should also be printed (regardless of the value of PRINTB). PRINTB is a logical variable which should be set to .TRUE. if the 5th-10th symmetric and the 3rd-10th unsymmetric modes which are calculated should also be printed. NSTORE is a logical variable which should be set to .TRUE. if the boundary conditions should be calculated but not stored.

PROG: ENTER LL (if there is a flap or flaps)

USER: Enter LL.

LL is the number of times that subroutine FLPDWN will intergrate to find the flap downwash mode or modes. Successive integrations will use more points than previous ones up to a maximum of JJMAX, the maximum number of integration stations available from the geometry file. Only the results for the largest number of integration points will be retained and written on the boundary condition file.

PROG: The program will ask for auxilliary information (if any) required for the various cases referred to by (BCS) and (BCAS). See section 3 for the auxilliary information required. After this is done, the program will loop back to the point in the main program at which the input device number is requested. The user can run another case at this point or else enter a negative number to terminate execution.

## 2.3 AMES' TSS BATCH JOBS

The batch mode operates the same as the conversational mode with the following exceptions:
(1) A "T" should be put in column 1 of the first card.
(2) The input device number must not be entered since all input in this case is assumed to originate on unit 5.
(3) ODISK must not be entered since the program assumes all output should go onto unit 6.
(4) To terminate execution in the batch mode, enter zero for ID1 (i.e. a blank card at the end of the input deck of the final case).

## 2.4 CONVERSION TO OTHER COMPUTERS

Remove all calls to GEMFIL, BCFIL, AIMFIl, OBEY, and CVRT and use appropriate control cards. These subroutines issue DDEFs and RELEASE commands making control cards unnecessary on TSS. Only the main program needs to be changed. These, hopefully, are the only changes that need to be made since considerable effort was made to program in standard FORTRAN.

# 3 DOWNWASH DISTRIBUTIONS

In the following explanations of the downwash modes, XSI and ETA are chordwise and streamwise coordinates, respectively, normalized by the effective semi-span (the semispan after yawing). Any input required is prompted for in the conversational mode, if input is from the terminal.

## 3.1 BCS(1) uniform mode

ALFA(XSI,ETA)=1.

INPUT REQUIRED: NONE

## 3.2 BCS(2) pitching mode

ALFA(XSI,ETA)=(XSI/BRAT-XSICM)/CBARRR where XSICM=center of mass or other reference position. XSICM is to be given in the coordinate system fixed to the wing while XSI is in the wind-centered coordinate system This mode corresponds to a non-dimensional pitch rate of 1. The solution for this mode gives the wing contribution to the Q stability derivatives (in the wind centered coordinate system). BRAT is the ratio of effective to actual semispan (ref. 1).

INPUT REQUIRED: XSICM

## 3.3 BCS(3) linear, symmetric twist mode

ALFA(XSI,ETA)=ABS(ETA)/BRAT

INPUT REQUIRED: NONE

## 3.4 BCS(4) parabolic, symmetric twist mode

ALFA(XSI,ETA)=(ETA/BRAT)**2

INPUT REQUIRED: NONE

## 3.5 BCS(5) residual flap downwash mode.

If there are 2 flaps, then this mode is for the symmetric deflection of both. Refer to NASA TN D-7251 and to subroutine FLPDWN for further documentation. The flap downwash modes are the only ones affected by the Mach number. The value of BCS(5) and BCAS(3) will be ignored if there is no flap data available from the geometry file.

3.6 BCS(6)-BCS(10)

These are modes for which tables and/or certain types of coefficients will be read by the program. The modes are defined by streamwise distributions (coefficients or tables) at spanwise stations selected by the user. These modes are assumed symmetric only if the wing is symmetric. See subroutine BOUND for further documentation.

INPUT REQUIRED: SEE SECTION 4

3.7 BCAS(1) anti-symmetric uniform mode (zero at center).

INPUT REQUIRED: NONE

3.8 BCAS(2) rolling mode (right tip down)

ALFA(XSI,ETA)=ETA/BRAT
This mode corresponds to a non-dimensional roll rate of 1.0. The solution for this mode gives the wing contribution to the quasi-static, P stability derivatives (in the wind centered coordinate system).

INPUT REQUIRED: NONE

3.9 BCAS(3) anti-symmetric residual flap mode.

If the wing is unsymmetric, there is no need to invoke this mode.

3.10 BCAS(4)-BCAS(10)

For a symmetric planform these modes are the anti-symmetric counterpart of those referred to by BCS(6)-BCS(10). For an unsymmetric wing the treatment of these modes is the same as those corresponding to BCS(6)-BCS(10).

INPUT REQUIRED: SEE SECTION 4

4 INPUT FOR SUBROUTINE SLOPES

All integer data should be entered in a 16I5 format and all floating point data in 8F10.0 format.

4.1 The following input items are requested each time SUBROUTINE SLOPES is called, i.e. for each true value of BCS(6)-BCS(10) and BCAS(4)-BCAS(10). See section 8 for examples on the use of subroutine SLOPES.

PROG: ENTER NSPSEC,ITYPES

USER: Enter NSPSEC, ITYPES.

NSPSEC is the number of spanwise sections at which twist and/or camber data will be supplied by the user.

ITYPES is the type of spanwise interpolation.

0 implies straight line interpolation.
≠0 implies that CODIM2 will be used(ref. 1).

PROG: ENTER XK (only if ITYPES≠0).

USER: Enter XK.

This is an interpolation control constant used for the spanwise interpolation when ITYPES≠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.

4.2 The data in this section will be requested a total of NSPSEC times in the order given (i.e. ETA, TWIST; ITYPEC; data required for specific value of ITYPEC; ETA, TWIST; etc. repeated a total of NSPSEC times).

PROG: ENTER ETA, TWIST

USER: Enter ETA, TWIST.

ETA is the spanwise station at which the twist and camber data apply. It may exceed 1 in value in order to control the CODIM2 interpolation near the ends if CODIM2 is used.

TWIST is the angle of twist in radians. Positive twist will tend to increase the angle of attack of the wing.

PROG: ENTER ITYPEC

USER: Enter ITYPEC.

ITYPEC denotes the type of chordwise slope definition that the user will use.

4.2.1 ITYPEC=0

In this case there is no camber, just twist.

INPUT REQUIRED: None.

4.2.2 ITYPEC=1

In this case a set of polynomials in CHI defines the slope distribution, which is the derivative of z/chord with respect to CHI. CHI is the local chordwise variable such that $0 \leq CHI \leq 1$.

INPUT REQUIRED: NPOLS/CHIMAX,SCALE,CHIREF, (NPOLS times)/POLYNOMIAL COEFFICIENTS, (NPOLS times).

NPOLS is the number of polynomials. CHIMAX denotes the upper limit of the polynomial. The default value is 1. SCALE is a scale factor for the polynomial coefficients. The default value is 1. CHIREF is the origin of the polynomial. The degree of each polynomial is limited to 8 or less. This data is prompted for and entered in the following manner:

PROG: ENTER NPOLS

USER: Enter NPOLS.

PROG: ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS

USER: Enter CHIMAX, SCALE, CHIREF, carriage return. Enter POLYNOMIAL COEFFICIENTS.

CHIMAX, SCALE, CHIREF, and the polynomial coefficients will be requested a total of NPOLS times.

4.2.3 ITYPEC=2

In this case a set of polynomials in CHI defines the camber distribution (z/chord as a function of CHI). This set of polynomials will be differentiated to determine the slope, so the leading coefficient in each polynomial is actually irrelevant.

INPUT REQUIRED: NPOLS/CHIMAX,SCALE,CHIREF, (NPOLS times)/POLYNOMIAL COEFFICIENTS, (NPOLS times).

NPOLS is the number of polynomials. CHIMAX denotes the upper limit of the polynomial. The default value is 1. SCALE is a scale factor for the polynomial coefficients. The default value is 1. CHIREF is the origin of the polynomial. The degree of each polynomial is limited to 8 or less. This data is prompted for and entered in the following manner:

PROG: ENTER NPOLS

USER: Enter NPOLS.

PROG: ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS

USER: Enter CHIMAX, SCALE, CHIREF, carriage return. Enter POLYNOMIAL COEFFICIENTS.

CHIMAX, SCALE, CHIREF, and the polynomial coefficients will be requested a total of NPOLS times.

4.2.4 ITYPEC=3

In this case a table of values defines the slope distribution, the derivative of z/chord with respect to CHI. Subroutine CODIM2 will be used for the interpolation.

INPUT REQUIRED: SCALE, XKC/ table of (CHI, ALFA) pairs.

SCALE is a scale factor for the table entries. The default value is 1. XKC is the end point interpolation control for CODIM2. If XKC=0., then straight lines will be used in the end intervals. If XKC=1., then full parabolic interpolation will be used in the end intervals. A value in between will give a curve in between. CHI is the local chordwise variable such that $0 \leq CHI \leq 1$ (values outside of this range may, however, be entered in the table to control the interpolation near the end points). ALFA is the derivative of z/chord with respect to CHI. A maximum of 40 table entries are allowed not including $CHI \geq 99.0$, which is the value used to mark the end of the table. This data is prompted for and entered in the following manner:

PROG: ENTER TABLE SCALE FACTOR,A1(OR XKC), AN(OR EPS),AND THE TABLE,(CHI=99.STOPS)

USER: Enter SCALE, XKC, carriage return.
Enter the (CHI, ALFA) table.

CHI and ALFA are pairs in the table. Each pair goes on a separate line or card. $CHI \geq 99.0$ marks the end of the table.

4.2.5 ITYPEC=4

In this case a table of values defines the camber distribution. CODIM2 will be used to determine the camber in the vicinity of the control points. Then numerical differentiation will be used to determine the slopes.

INPUT REQUIRED: SCALE, XKC, EPS/ table of (CHI, z/chord) values.

SCALE is a scale factor for the table entries. The default value is 1. XKC is the end point interpolation control for CODIM2. If XKC=0., then straight lines will be used in the end intervals. If XKC=1., then full parabolic interpolation will be used in the end intervals. A value in between will give a curve in between. EPS is used for numerical differentiation of the camber distribution determined by a table and CODIM2 (controlled deviation interpolation method). The default value is .005. CHI is the local chordwise variable such that $0 \leq CHI \leq 1$ (values outside of this range may, however, be entered in the table to control the interpolation near the end points). A maximum of 40 table entries are allowed not including $CHI \geq 99.0$, which is the value used to mark the end of the table. This data is prompted for and entered in the following manner:

PROG: ENTER TABLE SCALE FACTOR,A1(OR XKC), AN(OR EPS), AND THE TABLE,(CHI=99.STOPS)

USER: Enter SCALE, XKC, EPS, carriage return.
Enter the (CHI, z/c) table.

CHI and z/c are pairs in the table. Each pair goes on a separate line or card. $CHI \geq 99.0$ marks the end of the table.

4.2.6 ITYPEC=5

In this case a table of values defines the slope distribution. Cubic spline fits will be used to determine values at the chordwise control stations. Two types of spline fits are used. In the first (subroutine SPLIN1) the end point derivatives are not used. In the other (SPLIN2) the end point derivatives are used. The latter program generally gives a better fit.

INPUT REQUIRED: SCALE, A1, A2/table of (CHI, ALFA) values.

SCALE is a scale factor for the table entries. The default value is 1. A1 and AN are the derivatives of the slope at the end points of the table. A1 and AN should be put in if possible. If both values are actually zero, then input one as 1.E-30. If either is given, then both must be given. If both A1 and A2 are zero then SPLIN1 will be used. Otherwise SPLIN2 will be used. CHI is the local chordwise variable such that 0≤CHI≤1 (values outside of this range may, however, be entered in the table to control the interpolation near the end points). ALFA is the derivative of z/chord with respect to CHI. A maximum of 40 table entries are allowed not including CHI≥99.0, which is the value used to mark the end of the table. This data is prompted for and entered in the following manner:

PROG: ENTER TABLE SCALE FACTOR,A1(OR XKC), AN(OR EPS), AND THE TABLE(CHI=99.STOPS)

USER: Enter SCALE, A1, AN, carriage return.
Enter table of (CHI, ALFA) values.

CHI and ALFA are pairs in the table. Each pair goes on a separate line or card. CHI≥99.0 marks the end of the table.

4.2.7 ITYPEC=6

In this case a table of values defines the camber distribution. A cubic spline fit will be determined for the table and then differentiated to obtain the slopes. Two types of spline fits are used. In the first (subroutine SPLIN1) the end point derivatives are not used. In the other (SPLIN2) the end point derivatives are used. The latter program generally gives a better fit.

INPUT REQUIRED: SCALE, A1, A2/table of (CHI, z/chord) values.

SCALE is a scale factor for the table entries. The default value is 1. A1 and AN are the slopes at the end points of the table. A1 and

AN should be put in if possible. If both values are actually zero, then input one as 1.E-30. If either is given, then both must be given. If both A1 and A2 are zero then SPLIN1 will be used. Otherwise SPLIN2 will be used. CHI is the local chordwise variable such that $0 \leq CHI \leq 1$ (values outside of this range may, however, be entered in the table to control the interpolation near the end points). A maximum of 40 table entries are allowed not including $CHI \geq 99.0$, which is the value used to mark the end of the table. This data is prompted for and entered in the following manner:

PROG: ENTER TABLE SCALE FACTOR,A1(OR XKC),
AN(OR EPS), AND THE TABLE,(CHI=99.STOPS)

USER: Enter SCALE, A1, AN, carriage return.
Enter table of (CHI, ALFA) values.

CHI and ALFA are pairs in the table. Each pair goes on a separate line or card. $CHI \geq 99.0$ marks the end of the table.

4.2.8 ITYPEC>6 or ITYPEC<0

This causes SUBROUTINE USLOPE to be called, which is a program to be supplied by the user in the situation that he wants to define the chordwise slope distribution at some particular spanwise station by a method unavailable in subroutine SLOPES. See section 5.4 for an explanation of the arguments to subroutine USLOPE.

# 5 PROGRAM DESCRIPTIONS

## 5.1 MAIN PROGRAM

This program determines boundary conditions for a thin wing lifting surface program. It uses information stored on the geometry file and, optionally, on the influence matrix file. The boundary conditions that it calculates are stored on a boundary condition file. An equation solving program will read the influence matrix and boundary condition files and determine the coefficients in the expansion for delta-c.p..

This program reads from the geometry file (unit 7), the influence matrix file (unit 11), and the identification number file (unit 20). It writes on the identification number file and the boundary condition file (unit 8). See section 2 and 3 for conversational and batch use.

Arrays ETA, STHETA, TANLEL, TANLER, TANTEL, TANTER, CORDIP, XSILIP, CHIFPI, C2IP should be dimensioned at least as large as JJMAX. Arrays NINDEX and ETACP should be dimensioned at least as large as MM. ALFA should be dimensioned as large as the number of control points being used. CHICP should be dimensioned as large as PP.

## 5.2 SUBROUTINE BOUND

This subroutine is called by the main program to determine downwash cases.

This subroutine writes on the boundary condition file (unit 8). See section 2 for conversational and batch use.

INPUT VARIABLES:

| | |
|---|---|
| PP | Number of CHICP values (i.e. the number of chordwise control points) |
| (CHICP) | An array containing chordwise stations at which boundary conditions will be determined. |
| MMP | Number of spanwise control points on wing for symmetric boundary conditions. |

| | |
|---|---|
| MMPA | Number of spanwise control points on wing for antisymmetric boundary conditions. |
| (ETACP) | Spanwise stations at which boundary conditions will be determined. |
| CBARBR | Ratio of chordwise reference length to effective semispan. |
| BRAT | The ratio of effective to actual semispan. |
| (BCS) | An array of downwash condition indicators. See section 3. |
| (BCAS) | An array of downwash condition indicators. See section 3. |
| FLAP | Logical variable, whose value is to be .TRUE. if there is a control surface (the program currently does not support wings with control surfaces, however). |
| PRINTA | Logical variable whose value is .TRUE. if all the downwash modes which are calculated should be printed (regardless of the value of PRINTB). |
| PRINTB | Logical variable whose value is .TRUE. if the 5th-10th symmetric and the 3rd-10th unsymmetric modes which are calculated should be printed. |
| NSTORE | Logical variable whose value is .TRUE. if the boundary conditions should be calculated but not stored. |
| SYMM | Logical variable whose value is .TRUE. if the planform is symmetric. |
| CONV | Logical variable whose value is .TRUE. in conversational use. |
| CONV2 | Logical variable whose value is .TRUE. for conversational use when input is from the terminal. CONV2 controls conversational prompting. |
| UCI | Unit number for conversational input. |
| UCO | Unit number for conversational prompting. |
| U5 | Unit number for input. |
| U6 | Unit number for output. |
| U8 | Unit number for boundary condition file. |
| JRATIO | (JJMAX+1)/(MM+1), where MM = the number of spanwise control points on the entire wing. |
| LL | Number of times that subroutine FLPDWN should intergrate to find the flap downwash mode or modes. Subroutine FLPDWN has not been modified yet to be compatible with the boundary condition program, however. |

OTHER ARGUMENTS:

| | |
|---|---|
| (ALFA) | Storage space for downwash modes. |
| (ALFACS) | Storage space for flap downwash mode. |

## 5.3 SUBROUTINE SLOPES

This subroutine is called by subroutine BOUND to compute user supplied modes.

See sections 2 and 4 for conversational and batch use.

The arrays A, B, C, L, G, E, and H should be dimensioned as large as the number of table entries. Arrays CHI and ALF should be dimensioned as large as the number of table entries plus 1. The dimension of the arrays TEST and CHIDUM should be as large as PP and 2*PP, respectively. The dimension of ALFA3 must be the maximum of 2*PP and NSMAX. NSMAX is the maximum number of spanwise stations at which data should be given. NSMAX is a dimension for ETA, ALFA2, and ALFA3. NPPMAX is the first dimension for ALFA2 and the maximum allowable value for PP.

INPUT VARIABLES:

| | |
|---|---|
| U5 | Unit number for input. |
| U6 | Unit number for output. |
| UCO | Unit number for conversational prompting. |
| CONV2 | Logical variable whose value is .TRUE. for conversational use and when input is from the terminal. |
| (ETACP) | Spanwise stations at which boundary conditions will be determined. |
| (CHICP) | Chordwise stations at which boundary conditions will be determined. |
| MMU | Number of ETACP values. |
| PP | Number of CHICP values. |

OUTPUT VARIABLES:

| | |
|---|---|
| (ALFA) | Downwash boundary condition at the control points defined by ETACP and CHICP. |

## 5.4 SUBROUTINE USLOPE

This subroutine is one which is to be supplied by the user in the case that he wants to define the chordwise slope distribution at some particular spanwise station by a method unavailable in subroutine SLOPES. This subroutine will be called by subroutine SLOPES whenever ITYPEC<0 or ITYPEC>6. The subroutine should return the actual slope values not including the twist. Subroutine SLOPES will apply the minus sign (because the induced downwash must equal minus the slope) and then add in the twist.

INPUT VARIABLES:

| | |
|---|---|
| U5 | Unit number for input. |
| U6 | Unit number for output. |
| UCO | Unit number for conversational prompting. |
| CONV2 | Logical variable whose value is .TRUE. for conversational use and when input is from the terminal. CONV2 controls conversational prompting. |
| N | The number of the spanwise section, $1 \le N \le NSPSEC$. |
| ETA | The spanwise location of the station at which USLOPE is to calculate the chordwise slope distribution. |
| PP | Number of (CHICP) values. |
| (CHICP) | Chordwise stations at which boundary conditions will be determined. |
| ITYPEC | In Subroutine USLOPE, this variable will be <0 or >6 upon entry and can be used as a parameter if desired in subroutine USLOPE. |

OUTPUT VARIABLE:

| | |
|---|---|
| (ALFA3) | The value of the slope at the PP stations defined by (CHICP). |

## 6 INPUT FILES

The following disk files are read by the program. The AMES' TSS version of the program issues its own DDEF commands for the files, so none need be given. For other systems appropriate control cards will have to be supplied for units 8, 9, 11, and 12.

### 6.1 GEOMETRY FILE

This file is a variable record length file and is read from unit 7.

The first record contains identification and title information including number of control points and integration points.

The next record contains the chordwise control points, the array of indices from which the spanwise control points are derived, the tangents of the wing edge sweep angles at the integration stations, etc. For a complete description of this file see ref. 2.

On the AMES' TSS system this file has the name GEOM.XI where I is the numerical value of ID1.

### 6.2 Aerodynamic Influence File (AIM file)

This file is a variable record length file and is read from unit 11. A detailed description is given in ref. 3.

The first record contains identification and title information plus information about the size of the matrix and location of spanwise and chordwise control points.

The second and subsequent records contain the influence matrix itself. This file is generated by the influence matrix program.

On the AMES' TSS system this file has the name AIM.XI.XJ where I is the numerical value of ID1 and J is the numerical value of ID2.

### 6.3 Identification File

This file is read from unit 9 and rewritten on unit 9 and contains identification numbers in binary form. The third number on this file (ID3) is incremented by 1 and then the file is rewritten using the incremented value of ID3. ID3 is the identification number for the boundary condition file.

On the AMES' TSS system this file has the name IDFILE.

## 7 Boundary Condition File (BC file)

This file is a variable record length file written on unit 8. A detailed description of this file is given in ref. 2.

The first record contains identification and title information plus information identifying the type and number of symmetric and antisymmetric cases.

The next NSYM records are right-hand sides (i.e. the (BCS) downwash modes calculated by the program) for symmetric cases. NSYM equals the number of symmetric cases. The next NASYM records are the right-hand sides (i.e. the (BCAS) downwash modes calculated by the program) for antisymmetric cases. NASYM equals the number of antisymmetric cases. In the case of an unsymmetric wing there will be NSYM + NASYM right-hand sides.

On the AMES' TSS system this file has the name BC.XI.XK where I is the numerical value of ID1 and K is the numerical value of ID3, which is determined from IDFILE at the time the program is run and is found in the program output.

## 8 SAMPLE TERMINAL SESSION

Given below is a sample terminal session illustrating the conversational use of the boundary condition program on the Ames' 360/67 TSS computer system. This session might also be useful for the batch user to study. The program used a previously created geometry file to obtain the information needed to calculate the boundary conditions and determined boundary conditions for 4 symmetric cases and 2 antisymmetric cases.

The symmetric cases consisted of (1) uniform downwash, (2) pitching about the origin, (3) and (4) user-supplied modes. The data supplied for each user-supplied mode was identical except that linear spanwise interpolation was requested in the first user-supplied case, while CODIM2 spanwise interpolation was requested in the second. The data in each of the two user-supplied cases consisted of twist and camber distributions at 4 spanwise stations: eta = 0.0, 0.2, 0.6, and 1.0. The twist in degrees was determined by the following equation:

twist = -10*eta

Values derived from the above equation had to be converted to radians.

The camber distribution at eta = 0.0 was that of an NACA, 5-digit, 230 mean line, whose equation is
z/c = 2.6595*(.114714984*CHI-.6075*CHI**2+CHI**3) for 0≤CHI≤.15
z/c = 2.6595*(1.-.00830377*CHI) for .15≤CHI≤1.
This section was defined to the program in terms of these polynomials.

The camber distribution at the remaining spanwise stations were all parabolic arcs given by the equation

z/c = .1*CHI*(1.-CHI)

This camber distribution was defined to the program in different forms at each of of the remaining 3 stations. At eta = 0.2 the polynomial representation was used. At eta = 0.6 and eta = 1.0 a table of 11 values derived from the above equation was used to define the section. At eta = 0.6 CODIM2 interpolation and differentiation was used while at eta = 1.0 a spline fit to the curve (SUBROUTINE

SPLIN1) was used and differentiated.

The antisymmetric cases consisted of the rolling mode and a single user-supplied mode. The user-supplied mode was simply a case of twist only with linear spanwise interpolation between the two stations eta = 0. and eta = 1. The twist in radians was simply equal to eta so that this mode was identical to the rolling mode with a minus sign.

The actual terminal session is reproduced below with additional comments added in parenthesis. The output from this session was directed to a disk file named OUTPUT.BC.N1. The DDEF (control card) was created automatically by the program and the file name was computed using the value of ODISK. The contents of this file were printed and are given in appendix 1.

```
LOGON userid,password,terminal id
AMES USYSLIB
JOBLIBS SYSULIB
JBLB MEDAN
CALL BC$
(The boundary condition program is now in control.)

ENTER BATCH
F
ENTER INPUT DEVICE NUMBER
0
ENTER ODISK
    1
OUTPUT (is) ON FILE ...OUTPUT.BC.N1...
ENTER ID1,ID2
    4
(Now the program opens the geometry file whose name
is GEOM.X4 and reads this file.)
(If an appropriate input device number were entered
and an appropriate DDEF command given, then the
remaining prompts would have been suppressed and the
program would read data from some previously created
data file.)
ENTER (BCS)/(BCAS)/PRINTA,PRINTB,NSTORE
TTFFFTTFFF
FTFTFFFFFF
TTF
(Now the program prints some heading and other basic
information and then enters subroutine BOUND.)
```

```
ENTER C.M. POS./BREF
0.
(Now BOUND calls subroutine SLOPES.)
ENTER NSPECS,ITYPES
    4    0
ENTER ETA, TWIST
0.         0.
ENTER ITYPEC
    2
ENTER NPOLS
    2
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0.15        2.6595      0.
0.          .114714984 -.6075     1.
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0           2.6595
0.          -.00830377
(The above data constitute the equations for the NACA
230 mean  line.  Now the second spanwise station will
be considered.)
ENTER ETA, TWIST
0.2          -.0349066
ENTER ITYPEC
    2
ENTER NPOLS
    1
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0          .1
0.         1.0        -1.0
ENTER ETA, TWIST
0.6          -.1047198
ENTER ITYPEC
    4
ENTER TABLE SCALE FACTOR, A1(OR XKC),
AN(OR EPS), AND THE TABLE. (CHI=99.STOPS)
0.         1.0
0.         0.
0.1         .009
0.2         .016
0.3         .021
0.4         .024
0.5         .025
0.6         .024
0.7         .021
```

```
0.8         .016
0.9         .009
1.         0.
99.
(The above is a table of the parabolic arc.)
ENTER ETA, TWIST
1.         -.174533
ENTER ITYPEC
    6
ENTER TABLE SCALE FACTOR, A1(OR XKC),
AN(OR EPS), AND THE TABLE. (CHI=99.STOPS)
0.         0.
0.1         .009
0.2         .016
0.3         .021
0.4         .024
0.5         .025
0.6         .024
0.7         .021
0.8         .016
0.9         .009
1.         0.
99.
(Now control leaves SLOPE returning to BOUND and then
calls SLOPE again for the second user-supplied mode.)
ENTER NSPSECS,ITYPES
    4    1
ENTER XK
1.
ENTER ETA, TWIST
0.         0.
ENTER ITYPEC
    2
ENTER NPOLS
    2
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0.15        2.6595     0.
0.         .114714984 -.6075     1.
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0          2.6595
0.         -.00830377
ENTER ETA, TWIST
0.2        -.0349066
ENTER ITYPEC
    2
```

```
ENTER NPOLS
    1
ENTER CHIMAX,SCALE,CHIREF/
POLYNOMIAL COEFFICIENTS
0          .1
0.         1.0        -1.0
ENTER ETA, TWIST
0.6         -.1047198
ENTER ITYPEC
    4
ENTER TABLE SCALE FACTOR, A1(OR XKC),
AN(OR EPS), AND THE TABLE. (CHI=99.STOPS)
0.         1.0
0.         0.
0.1         .009
0.2         .016
0.3         .021
0.4         .024
0.5         .025
0.6         .024
0.7         .021
0.8         .016
0.9         .009
1.         0.
99.
ENTER ETA, TWIST
1.          -.174533
ENTER ITYPEC
    6
ENTER TABLE SCALE FACTOR, A1(OR XKC),
AN(OR EPS), AND THE TABLE. (CHI=99.STOPS)
0.         0.
0.1         .009
0.2         .016
0.3         .021
0.4         .024
0.5         .025
0.6         .024
0.7         .021
0.8         .016
0.9         .009
1.         0.
99.
(Now control leaves SLOPE returning to BOUND and then
calls SLOPE again for the user-supplied antisymmetric
mode.)
ENTER NSPSECS,ITYPES
```

```
    2
ENTER ETA, TWIST
0.        0.
ENTER ITYPEC
    0
ENTER ETA, TWIST
1.        1.
ENTER ITYPEC
    0
(Now control returns to BOUND and then to the main
program.)
ENTER INPUT DEVICE NUMBER
-1
TERMINATED: STOP 777
(The operating system is now in control.)
PRINT OUTPUT.BC.N2,PRTSP=EDIT,STATION=RMT05
 PRINT BSN=????,    ??? LINES
LOGOFF
```

# 9 REFERENCES

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

2. Medan, R. T.: Geometry Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,309.

3. Medan, R. T. and Ray, K. S.: Aerodynamic Influence Matrix Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,324.

APPENDIX-I

OUTPUT FROM SAMPLE TERMINAL SESSION

DETERMINATION OF THIN LIFTING SURFACE BOUNDARY CONDITIONS
=========================================================

```
TITLE      = RECTANGULAR WING   AR = 2   11-13-73
ID1        =    4
ID2        =    0
ID3        =    7
BRAT       =    1.00000
CBARBR     =    1.00000

CHORDWISE CONTROL POINT LOCATIONS
                0.000000
                0.146447
                0.500000
                0.853553
                1.000000

 SPANWISE CONTROL POINT LOCATIONS
                0.965926
                0.866025
                0.707107
                0.500000
                0.258819
               -0.000000

BCS
T T F F F T T F F F

BCAS
F T F T F F F F F F

PRINTA, PRINTB, NSTORE
T T F
```

UNIFORM DOWNWASH

| | | | | |
|---|---|---|---|---|
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |



PITCHING MODE

C.M. POS./BREF = 0.000000

| | | | | |
|---|---|---|---|---|
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |
| 0.0000000 | 0.1464466 | 0.5000000 | 0.8535533 | 1.0000000 |

USER-SUPPLIED SYMMETRIC MODE

| | | | | |
|---|---|---|---|---|
| -0.2329614 | -0.2400104 | -0.1685513 | -0.0980664 | -0.0743155 |
| -0.2238179 | -0.2223796 | -0.1511248 | -0.0805781 | -0.0567508 |
| -0.2092727 | -0.1943331 | -0.1234033 | -0.0527584 | -0.0288094 |
| -0.1833289 | -0.1579772 | -0.0872664 | -0.0165554 | 0.0087901 |
| -0.1444004 | -0.1158831 | -0.0451724 | 0.0255383 | 0.0540544 |
| -0.3050843 | -0.0029844 | 0.0220839 | 0.0220839 | 0.0220839 |

USER-SUPPLIED SYMMETRIC MODE

| | | | | |
|---|---|---|---|---|
| -0.2340505 | -0.2399799 | -0.1685526 | -0.0980581 | -0.0745003 |
| -0.2269311 | -0.2222926 | -0.1511289 | -0.0805547 | -0.0572790 |
| -0.2120129 | -0.1942565 | -0.1234068 | -0.0527377 | -0.0292743 |
| -0.1850461 | -0.1579284 | -0.0872711 | -0.0165292 | 0.0084959 |
| -0.1379284 | -0.1161122 | -0.0451886 | 0.0256270 | 0.0553199 |
| -0.3050843 | -0.0029844 | 0.0220839 | 0.0220839 | 0.0220839 |

ROLLING MODE

| | | | | |
|---|---|---|---|---|
| 0.9659258 | 0.9659258 | 0.9659258 | 0.9659258 | 0.9659258 |
| 0.8660254 | 0.8660254 | 0.8660254 | 0.8660254 | 0.8660254 |
| 0.7071068 | 0.7071068 | 0.7071068 | 0.7071068 | 0.7071068 |
| 0.5000000 | 0.5000000 | 0.5000000 | 0.5000000 | 0.5000000 |
| 0.2588190 | 0.2588190 | 0.2588190 | 0.2588190 | 0.2588190 |

USER-SUPPLIED ANTI-SYMMETRIC MODE

| | | | | |
|---|---|---|---|---|
| -0.9659258 | -0.9659258 | -0.9659258 | -0.9659258 | -0.9659258 |
| -0.8660254 | -0.8660254 | -0.8660254 | -0.8660254 | -0.8660254 |
| -0.7071068 | -0.7071068 | -0.7071068 | -0.7071068 | -0.7071068 |
| -0.5000000 | -0.5000000 | -0.5000000 | -0.5000000 | -0.5000000 |
| -0.2588190 | -0.2588190 | -0.2588190 | -0.2588190 | -0.2588190 |

MODES STORED AND FILE CLOSED

APPENDIX-II

COMPUTER PROGRAM LISTING

```
C       .       .       .       .       .       .       .
C
C.....THIS PROGRAM DETERMINES BOUNDARY CONDITIONS FOR A THIN WING LIFTING
C.....SURFACE PROGRAM.  IT USES INFORMATION STORED ON THE GEOMETRY FILE
C.....AND, OPTIONALLY, ON THE INFLUENCE MATRIX FILE.  THE BOUNDARY
C.....CONDITIONS THAT IT CALCULATES ARE STORED ON A BOUNDARY CONDITION FILE.
C.....AN EQUATION SOLVING PROGRAM WILL READ THE INFLUENCE MATRIX AND
C.....BOUNDARY CONDITION FILES AND DETERMINE THE COEFFICIENTS IN
C.....THE EXPANSION FOR DELTA-C.P..
      DOUBLE PRECISION ETA, STHETA
      INTEGER UCI,UCO,U5,U5R,U6,U8,U11,U20,ODISK, U7
      INTEGER TITLE,P,PP,UNSYM
      LOGICAL CONV,BATCH,LOG1,CONV2,BCS,BCAS,PRINTA,PRINTB
      LOGICAL STORE, NSTORE, FLAP, SYMM
      REAL MACH,LAMDAC
      DIMENSION TITLE(26), CHICP(20), NINDEX(47),ETACP(47)
      DIMENSION ETA(383),STHETA(383),TANLEL(383),TANLER(383)
      DIMENSION TANTEL(383),TANTER(383),CORDIP(383),XSILIP(383)
      DIMENSION CHIFPI(383),C2IP(383),BCS(10),BCAS(10)
      DIMENSION XDUM2(2),XDUM6(6),NDUM9(9),NDUM2(2)
      DIMENSION ALFA(200), ALFACS(001)
      EQUIVALENCE (XDUM2,XDUM6,NDUM9,NDUM2)
      DATA UCI,UCO,U5,U5R,U6,U8,U11,U20/5,6, 5, 5, 6, 8, 11,9/,U7/7/
      DATA CONV2/.FALSE./
C
C.....UCI   =  UNIT NUMBER FOR CONVERSATIONAL INPUT.
C.....UCO   =  UNIT NUMBER FOR CONVERSATIONAL OUTPUT.
C.....U5    =  UNIT NUMBER FOR INPUT OF THE FIRST CARD AND
C.....         THE UNIT NUMBER FOR INPUT UNDER CONVERSATIONAL USE.
C.....U5R   =  UNIT NUMBER FOR INPUT (EXCEPT FOR THE FIRST CARD)
C.....         UNDER BATCH CONTROL.
C.....U6    =  THE UNIT NUMBER FOR REGULAR OUTPUT UNDER EITHER
C.....         BATCH OR CONVERSATIONAL CONTROL.
C.....U8    =  UNIT NUMBER FOR BOUNDARY CONDITION FILE.
C.....U7    =  UNIT NUMBER FOR READING THE GEOMETRY FILE
C.....U11   =  UNIT NUMBER FOR READING THE INFLUENCE MATRIX FILE
C.....U20   =  UNIT NUMBER FOR IDENTIFICATION NUMBER FILE.
C
C.....THE FOLLOWING IS NECESSARY FOR TSS
```

```
C
      U5=5
C
      WRITE(UCO,6004)
C.....OR BATCH USE.
C
C.....THE FOLLOWING ALLOWS THE USER TO CHOOSE CONVERSATIONAL
      READ(U5,5000) BATCH
      CONV=.NOT.BATCH
C
20    CONTINUE
      IF(BATCH) GO TO 40
      WRITE(UCO,6001)
      READ(UCI,5001) U5
      IF (U5 .EQ. 0) U5 = UCI
      CONV2=U5.EQ.UCI
C
C.....U5.LT.0 IS FOR TERMINATING EXECUTION CONVERSATIONALLY.
C
      IF(U5.LT.0) STOP 777
C.....IF (CONV2), THEN ALL THE INPUT WILL BE FROM THE TERMINAL.
30    CONTINUE
      WRITE(UCO,6005)
C
C.....ODISK IS FOR THE AMES' TSS VERSION OF THE PROGRAM.
C.....ODISK MUST BE AN INTEGER FROM 0 TO 9.
C.....IF ODISK = 0, OUPUT WILL BE ON UNIT 6, IF ODISK IS BETWEEN 0 AND 10
C.....THEN OUPUT WILL BE ON UNIT 4 AND ON A FILE NAMED  OUTPUT.BC.NX
C.....WHERE X = VALUE OF ODISK.      THE AMES' PROGRAMS OBEY AND CVRT ARE
C.....USED TO GENERATE AND GIVE THE NEEDED DDEF AND RELEASE COMMANDS TO
C.....TSS OPERATING SYSTEM.
C
      READ (UCI,5001) ODISK
      U6 = 6
      IF (ODISK .EQ. 0) GO TO 50
      U6 = 4
      ODISK = MOD(MAX0(1,ODISK),10)
      WRITE (UCO,6006) ODISK
      CALL OBEY(16,16HRELEASE FT04F001 )
```

```
      CALL CVRT(ODISK,1,
     1 44H(!DDEF FT04F001,,OUTPUT,BC,N!,I1,6X)
     2ALFA, 8, 8H(8A4)                )
      CALL OBEY(32,ALFA)
      REWIND U6
      GO TO 50
40    CONTINUE
      U5=U5R
50    CONTINUE
      IF (CONV) WRITE(UCO,6000)
      IF(CONV)READ(UCI,5001) IDD1,IDD2
      IF(BATCH)READ(U5,5001) IDD1,IDD2
      IF(IDD1.LT.0)STOP
C
C.....IN THE PDP-10 VERSION IDD1 AND IDD2 ARE USED TO DETERMINE THE
C.....GEOMETRY AND INFLUENCE MATRIX FILE NAMES.   SAME FOR AMES! TSS VER.
C.....IN ANY CASE, THE VALUES READ IN WILL, IF GREATER THAN ZERO,
C.....BE CHECKED AGAINST THE VALUES ON THE GEOMETRY AND INFLUENCE
C.....MATRIX FILES AND A STOP OR PAUSE WILL BE EXECUTED IF THERE
C.....IS NO AGREEMENT.
C.....IF IDD2 .LE. 0, THE INFLUENCE MATRIX FILE WILL NOT BE USED.
C
C
C.....FOR AMES! TSS VERSION, ISSUES DDEF COMMANDS TO OPERATING SYSTEM.
C
      CALL GEMFIL(IDD1)
      READ(U7 )ID1,PP,MM,NDUM2,UNSYM,NDUM2,MREF,JJMAX,NFLAPS,
     1TITLE,NTITL
C
C.....PP     = NUMBER OF CHORDWISE CONTROL POINTS
C.....MM     = NUMBER OF SPANWISE CONTROL POINTS.
C.....UNSYM  = NON-ZERO FOR UNSYMMETRICAL WING.
C.....MREF   = REFERENCE INTEGER FOR DETERMINING THE SPANWISE
C.....         CONTROL POINTS FROM (NINDEX) AND (ETA).  THESE
C.....         CONTROL POINTS WILL BE STORED IN (ETACP).
C.....JJMAX  = THE NUMBER OF SPANWISE INTEGRATION POINTS.  THE
C.....         SPANWISE CONTROL POINTS ARE A SUBSET OF THE
C.....         SPANWISE INTEGRATION POINTS.
C.....NFLAPS = NUMBER OF CONTROL SURFACES (0,1, OR 2).
```

```
C.....(TITLE)= ARRAY OF ALPHANUMERIC TITLING INFORMATION.
C.....NTITLE = THE NUMBER OF WORDS IN THE TITLE.
C
      LOG1=ID1.NE.IDD1 .AND. IDD1 .GT. 0
      IF(CONV.AND.LOG1)PAUSE ' ID NUMBER CONFLICT FOR GEOM FILE '
      IF (BATCH.AND. LOG1) CALL STOP2(U6,' ID NUMBER CONFLICT FOR
     1GEOM FILE   ',FLOAT(ID1))
      READ(U7 ) (CHICP(I),I=1,PP),(NINDEX(I),I=1,MM),(TANLEL(I),I=1,
     1JJMAX),(TANLER(I),I=1,JJMAX),(TANTEL(I),I=1,JJMAX),
     2(TANTER(I),I=1,JJMAX),(ETA(I),I=1,JJMAX),
     3(STHETA(I),I=1,JJMAX),(XSILIP(I),I=1,JJMAX),
     4(CORDIP(I),I=1,JJMAX),BRAT,CBARBR
C
C.....(CHICP)  = ARRAY OF CHORDWISE CONTROL POINTS SUCH THAT
C.....           0.0 .LE. CHICP(I) .LE. 1.000.
C.....(NINDEX) = INTEGER ARRAY FROM WHICH THE SPANWISE CONTROL
C.....           POINTS ARE DERIVED FROM THE SPANWISE
C.....           INTEGRATION POINTS.
C.....(TANLEL), (TANLER), (TANTEL), AND (TANTER) ARE THE
C.....           TANGENTS OF THE WING EDGE SWEEP ANGLES AT THE
C.....           INTEGRATION STATIONS.
C.....(ETA)    = THE SPANWISE INTEGRATION STATIONS NON-DIMENSIONAL-
C.....           IZED BY THE EFFECTIVE SEMI-SPAN, B2 ((ETA) IS
C.....           DOUBLE PRECISION).
C.....(STHETA) = (SQRT(1.-ETA**2))  (DOUBLE PRECISION).
C.....(XSILIP) = LEADING EDGE LOCATIONS AT THE INTEGRATION STATIONS
C.....           NORMALIZED BY B2.
C.....(CORDIP) = STREAMWISE CHORD LENGTHS AT THE INTEGRATION STA
C.....           TIONS NORMALIZED BY B2.
C.....BRAT     = LATERAL REFERENCE LENGTH (USUALLY THE TRUE SEMI-
C.....           SPAN)/B2.  USUALLY BRAT WOULD BE 1.0 IF THE WING HAS
C.....           NOT BEEN YAWED.
C.....CBARBR   = LONGITUDINAL REFERENCE LENGTH (USUALLY THE MEAN
C.....           GEOMETRIC CHORD)/LATERAL REFERENCE LENGTH.
C
      IF(NFLAPS.NE.0) READ (U7 ) XDUM,LAMDAC,XDUM2,ETA1,ETA2,
     1XDUM6, (CHIFFI(I),I=1,JJMAX), (C2IP(I),I=1,JJMAX)
C
C.....THE INFLUENCE MATRIX FILE WILL NOT BE READ IF IDD2.LT.0.
```

```
C
      IF(IDD2.LE.0) GO TO 60
C
C
C
C.....FOR AMES' TSS VERSION.  ISSUES DDEF COMMANDS.
C
      CALL AIMFIL(IDD1,IDD2)
      READ (U11) ID1,ID2,NDUM,TITLE,NTITL,PP,NDUM,MM,MREF,
     1NDUM9,MACH,EPS,XDUM,(CHICP(I),I=1,PP),(NINDEX(I),I=1,MM)
C.....INFORMATION FROM THE INFLUENCE MATRIX FILE OVERLAYS INFOR-
C.....MATION FROM THE GEOMETRY FILE.
      LOG1 = IDD1.NE.0 .AND. IDD1.NE.ID1
      LOG1=LOG1 .OR. (IDD2.NE.0 .AND. IDD2.NE.ID2)
      IF (CONV.AND.LOG1) PAUSE ' ID NUMBER CONFLICT FOR INFLUENCE
     1 MATRIX FILE  '
      IF(BATCH.AND.LOG1) CALL STOP2(U6,' ID NUMBER CONFLICT FOR
     1INFLUENCE MATRIX FILE   ',FLOAT(ID2))
60    CONTINUE
C
      IF(CONV2) WRITE(UCO,6007)
      READ(U5,5000) BCS
      READ(U5,5000) BCAS
      READ(U5,5000) PRINTA,PRINTB,NSTORE
      PRINTB=PRINTB.OR.PRINTA
C
C.....EACH ELEMENT OF BCS AND BCAS CORRESPONDS TO A DIFFERENT DOWNWASH
C.....CONDITION.  A TRUE VALUE MEANS THAT THE CORRESPONDING BOUNDARY
C.....CONDITION WILL BE CALCULATED OR READ FROM CARDS AND, IF
C.....NSTORE IS .FALSE., THEN WRITTEN ON A FILE FOR USE IN THE
C.....EQUATION SOLVING PROGRAM.  BOTH ARRAYS ARE READ AND ACTED UPON IF
C.....THE WING IS SYMMETRIC OR UNSYMMETRIC.  IN THE LATTER CASE THE
C.....BOUNDARY CONDITION WILL BE CALCULATED FOR THE ENTIRE WING.
C.....THE CASES CORRESPONDING TO EACH OF THE ELEMENTS ARE AS FOLLOWS--
C
C BCS(1)    UNIFORM MODE
C BCS(2)    PITCHING MODE
C           ALFA(XSI,ETA)=(XSI/BRAT-XSICM)/CHARBR
C           WHERE XSICM=CENTER OF MASS OR OTHER REFERENCE POSITION.
```

```
C              XSICM IS TO BE GIVEN IN THE COORDINATE SYSTEM FIXED
C              TO THE WING WHILE XSI IS IN THE WIND-CENTERED
C              COORDINATE SYSTEM.
C              THIS MODE CORRESPONDS TO A NON-DIMENSIONAL PITCH RATE OF 1.
C              THE SOLUTION FOR THIS MODE GIVES THE WING CONTRIBUTION TO THE
C              Q STABILITY DERIVATIVES (IN THE WIND CENTERED COORDINATE
C              SYSTEM).
C BCS(3)       LINEAR, SYMMETRIC TWIST MODE
C              ALFA(XSI,ETA)=ABS(ETA)/BRAT
C BCS(4)       PARABOLIC, SYMMETRIC TWIST MODE
C              ALFA(XSI,ETA)=(ETA/BRAT)**2
C BCS(5)       RESIDUAL FLAP DOWNWASH MODE.  IF THERE ARE 2 FLAPS, THEN
C              THIS MODE IS FOR THE SYMMETRIC DEFLECTION OF BOTH.
C              REFER TO NASA TN D-7251 AND TO SUBROUTINE FLPDWN FOR
C              FURTHER DOCUMENTATION.  THE FLAP DOWNWASH MODES ARE THE ONLY
C              ONLY ONES AFFECTED BY THE MACH NUMBER.
C              THE VALUE OF BCS(5) AND BCAS(3) WILL BE IGNORED IF THERE
C              IS NO FLAP DATA AVAILABLE FROM THE GEOMETRY FILE.
C BCS(6)-      THESE ARE MODES FOR WHICH TABLES AND/OR CERTAIN TYPES
C BCS(10)      OF COEFFICIENTS WILL BE READ BY THE PROGRAM.
C              THE MODES ARE DEFINED BY STREAMWISE DISTRIBUTIONS
C              (COEFFICIENTS OR TABLES) AT SPANWISE STATIONS SELECTED
C              BY THE USER.  THESE MODES ARE ASSUMED SYMMETRIC ONLY
C              IF THE WING IS SYMMETRIC.  SEE SUBROUTINE BOUND FOR
C              FURTHER DOCUMENTATION.
C
C
C BCAS(1)      ANTI-SYMMETRIC UNIFORM MODE (ZERO AT CENTER).
C BCAS(2)      ROLLING MODE (RIGHT TIP DOWN)
C              ALFA(XSI,ETA)=ETA/BRAT
C              THIS MODE CORRESPONDS TO A NON-DIMENSIONAL ROLL RATE
C              OF 1.0.  THE SOLUTION FOR THIS MODE GIVES THE WING CON-
C              TRIBUTION TO THE QUASI-STATIC, P STABILITY DERIV-
C              ATIVES (IN THE WIND CENTERED COORDINATE SYSTEM).
C BCAS(3)      ANTI-SYMMETRIC RESIDUAL FLAP MODE.  IF THE WING IS
C              UNSYMMETRIC, THERE IS NO NEED TO INVOKE THIS MODE.
C BCAS(4)-     FOR A SYMMETRIC PLANFORM THESE MODES ARE THE ANTI-SYMMETRIC
C BCAS(10)     COUNTERPART OF THOSE REFERRED TO BY BCS(6)-BCS(10).
C              FOR AN UNSYMMETRIC WING THE TREATMENT OF THESE MODES IS
```

```
C            THE SAME AS THOSE CORRESPONDING TO BCS(6)-BCS(10).
C
C
C
C
C.....PRINTA  .TRUE. MEANS THAT ALL THE DOWNWASH MODES WHICH
C.....         ARE CALCULATED WILL BE PRINTED (REGARDLESS OF THE VALUE OF
C.....         PRINTB).
C.....PRINTB  .TRUE. MEANS THAT THE 5TH-10TH SYMMETRIC AND THE
C.....         3RD-10TH UNSYMMETRIC MODES WHICH ARE CALCULATED WILL
C.....         BE PRINTED.
C.....NSTORE  .TRUE. MEANS THAT THE BOUNDARY CONDITIONS WILL BE
C.....         CALCULATED BUT NOT STORED.
C
      BCS(5)=BCS(5) .AND. NFLAPS.GT.0
      BCAS(3)=BCAS(3).AND.NFLAPS.GT.0
      FLAP=BCS(5).OR.BCAS(3)
      IF(CONV2.AND.FLAP) WRITE(UCO,6021)
      IF(FLAP)READ(U5,5001) LL
C
      STORE=.NOT.NSTORE
      ID3=0
      SYMM=UNSYM.EQ.0
      MMP=MM
      MMPA=MM
      IF(SYMM)MMP=(MM+1)/2
      IF(SYMM)MMPA=MM/2
C
C.....(ETACP)=LOCATIONS OF SPANWISE CONTROL STATIONS.
      JRATIO=(JJMAX+1)/(MREF+1)
      DO 85 M=1,MMP
      INDEX = NINDEX(M)*JRATIO
      ETACP(M)=ETA(INDEX)
 85   CONTINUE
      IF(NSTORE) GO TO 90
C
C.....DETERMINING IDENTIFICATION NUMBER OF THIS RUN
      CALL OBEY(22,24HDDEF FT09F001,,IDFILE      )
      REWIND U20
      READ(U20) IDA,IDB,ID3,IDD
```

```
      ID3=ID3+1
      REWIND U20
      WRITE(U20) IDA,IDB,ID3,IDD
      END FILE U20
C.....FOR AMES' TSS VERSION ONLY
      CALL OBEY(16,16HRELEASE FT09F001 )
C
C.....NSYM  = NUMBER OF      SYMMETRICAL CASES.
C.....NASYM = NUMBER OF ANTI-SYMMETRICAL CASES.
      NSYM=0
      NASYM=0
      DO 87 I=1,10
      IF(BCS(I))NSYM=NSYM+1
      IF(BCAS(I))NASYM=NASYM+1
87    CONTINUE
C
C.....WRITING THE INTRODUCTORY RECORD OF THE BOUNDARY CONDITION FILE.
C
C
C.....FOR AMES' TSS VERSION ONLY.  BCFIL ISSUES DDEF COMMANDS TO THE
C.....OPERATING SYSTEM.
C
      CALL BCFIL(ID1,ID3)
      WRITE(U8) ID1,ID3,TITLE,UNSYM,NSYM,NASYM,BCS,BCAS,PP,CWTYPE,
     1MM,MMP,MMPA,MREF,SWTYPE, (CHICP(P),P=1,PP),(NINDEX(I),I=1,MMP),
     2(ETACP(M),M=1,MMP)
90    CONTINUE
C
C
      WRITE(U6,6003)
      WRITE(U6,6008) (TITLE(N),N=1,NTITL)
      IF(ID3.EQ.0) WRITE(U6,6009) ID1,ID2
      IF(ID3.NE.0) WRITE(U6,6020) ID1,ID2,ID3
      WRITE(U6,6010) BRAT,CBARBR
      IF(FLAP) WRITE(U6,6022) LL
      WRITE(U6,6011)
      WRITE(U6,6012) (CHICP(P),P=1,PP)
      WRITE(U6,6013)
      WRITE(U6,6012) (ETACP(I),I=1,MMP)
```

```
      WRITE(U6,6014)
      WRITE(U6,6015) BCS
      WRITE(U6,6016)
      WRITE(U6,6015) BCAS
      WRITE(U6,6017)
      WRITE(U6,6015) PRINTA,PRINTB,NSTORE
      WRITE(U6,6023)
      CALL       BOUND (PP,CHICP,MMP,MMPA,ETACP,CBARBR,BRAT,
     1BCS,BCAS,FLAP, PRINTA, PRINTB, STORE, SYMM, CONV, CONV2,
     2UCI, UCO, U5, U6, U8,
     3XSILIP,CORDIP, NINDEX, TANLEL, TANLER, TANTEL, TANTER,
     4ETA, STHETA, LAMDAC, ETA1, ETA2, CHIFPI, C2IP,
     5JJMAX, JRATIO, ALFA, LL, ALFACS)
      IF(U6.EQ.UCO) GO TO 120
      END FILE U6
120   CONTINUE
      GO TO 20
C
C.....INPUT FORMATS
C
5000  FORMAT(80L1)
5001  FORMAT(16I5 )
5002  FORMAT(8F10.0)
C
C.....OUTPUT FORMATS
C
6000  FORMAT(' ENTER ID1,ID2' )
6001  FORMAT(/////' ENTER INPUT DEVICE NUMBER' )
6003  FORMAT(1H1/
     1'0DETERMINATION OF THIN LIFTING SURFACE BOUNDARY CONDITIONS'/
     2' ============================================================'///)
6004  FORMAT(' ENTER BATCH' )
6005  FORMAT(' ENTER ODISK')
6006  FORMAT(' OUTPUT ON FILE ...OUTPUT.BC.N',I1,3H...)
6007  FORMAT(  ' ENTER (BCS)/(BCAS)/PRINTA,PRINTB,NSTORE' )
C
C.....THE FOLLOWING FORMAT DEPENDS ON THE INTEGER WORD LENGTH IN
C.....CHARACTERS.  CHANGE AS REQUIRED FOR THE COMPUTER BEING USED.
6008  FORMAT(//12H TITLE    = , 20A4)
```

```
6009  FORMAT(4H ID1, 6X, 1H= , I5/
     1        4H ID2, 6X, 1H= , I5)
6010  FORMAT(5H BHAT, 5X, 1H=,F11.5/
     17H CBARBP, 3X, 1H=, F11.5)
6011  FORMAT(/35H CHORDWISE CONTROL POINT LOCATIONS   )
6012  FORMAT(F23.6)
6013  FORMAT(/35H  SPANWISE CONTROL POINT LOCATIONS   )
6014  FORMAT(/4H BCS)
6015  FORMAT(1X,10(L1,1X))
6016  FORMAT(/5H BCAS)
6017  FORMAT(/24H PRINTA, PRINTB, NSTORE   )
6020  FORMAT(4H ID1, 6X, 1H=, I5/4H ID2, 6X, 1H=, I5/
     14H ID3, 6X, 1H=, I5)
6021  FORMAT(' ENTER LL'/)
6022  FORMAT(3H LL, 7X, 1H=, I5)
6023  FORMAT(/////)
      END
```

```
      SUBROUTINE CODIM2(YI,XI,NI,T,ANS,NA,XK)
C
C****             A CONTROLLED DEVIATION ITERPOLATION METHOD
C
```



```
      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
```

```
      SUBROUTINE BOUND (PP,CHICP,MMP,MMPA,ETACP,CBARBR,BRAT,
     1BCS,BCAS,FLAP, PRINTA, PRINTB, STORE, SYMM, CONV, CONV2,
     2UCI, UCO, U5, U6, U8,
     3XSILIP,CORDIP, NINDEX, TANLEL, TANLER, TANTEL, TANTER,
     4ETA, STHETA, LAMDAC, ETA1, ETA2, CHIFPI, C2IP,
     5JJMAX, JRATIO, ALFA, LL, ALFACS)
      DOUBLE PRECISION ETA, STHETA
      INTEGER P, PP, UCI, UCO, U5, U6, U8
      LOGICAL BCS, BCAS, PRINTA, PRINTB, STORE, CONV, CONV2
      LOGICAL FLAP, SYMM, FLPCAL
      DIMENSION CHICP(PP), ETACP(MMP), NINDEX(MMP), BCS(10), BCAS(10)
      DIMENSION ALFA(PP, MMP), ALFACS(LL, PP, MMP)
      DIMENSION XSILIP(JJMAX), CORDIP(JJMAX), TANLEL(JJMAX),
     1TANLER(JJMAX), TANTEL(JJMAX), TANTER(JJMAX), ETA(JJMAX),
     2STHETA(JJMAX), CHIFPI(JJMAX), C2IP(JJMAX)
      FLPCAL=.TRUE.
 10   CONTINUE
      MMU=MMP
      IF ( BCS( 1)) GO TO  100
 22   IF ( BCS( 2)) GO TO  200
 23   IF ( BCS( 3)) GO TO  300
 24   IF ( BCS( 4)) GO TO  400
```

```
25      IF ( BCS( 5)) GO TO  500
        NC=6
26      IF ( BCS( 6)) GO TO  600
        NC=7
27      IF ( BCS( 7)) GO TO  600
        NC=8
28      IF ( BCS( 8)) GO TO  600
        NC=9
29      IF ( BCS( 9)) GO TO  600
        NC=10
30      IF ( BCS(10)) GO TO  600
        MMU=MMPA
        IF (BCAS( 1)) GO TO  700
32      IF (BCAS( 2)) GO TO  800
33      IF (BCAS( 3)) GO TO  900
        NC=4
34      IF (BCAS( 4)) GO TO 1000
        NC=5
35      IF (BCAS( 5)) GO TO 1000
        NC=6
        IF (BCAS( 6)) GO TO 1000
        NC=7
        IF (BCAS( 7)) GO TO 1000
        NC=8
        IF (BCAS( 8)) GO TO 1000
        NC=9
        IF (BCAS( 9)) GO TO 1000
        NC=10
        IF (BCAS(10)) GO TO 1000
        IF(.NOT.STORE) RETURN
        END FILE U8
        WRITE(U6,6013)
        RETURN
C
C
C.....START DETERMINING DOWNWASH CASES.
C
C
100     CONTINUE
```

```
C
C.....UNIFORM DOWNWASH
      DO 110 M=1,MMP
      DO 110 P=1,PP
      ALFA(P,M)=1.
110   CONTINUE
      IF(PRINTA) WRITE(U6,6000)
      ASSIGN 22 TO NSTAT
      GO TO 2000
200   CONTINUE
C
C.....PITCHING MODE
      IF (CONV2) WRITE(UC0,6001)
      READ(U5,5000) XSICM
      DO 210 M=1,MMP
      J=JRATIO*NINDEX(M)
      DO 210 P=1,PP
      XSI=XSILIP(J) + CHICP(P)*CORDIP(J)
      ALFA(P,M)=(XSI/BRAT-XSICM)/CBARBR
210   CONTINUE
      IF(PRINTA) WRITE(U6,6002) XSICM
      ASSIGN 23 TO NSTAT
      GO TO 2000
300   CONTINUE
C
C.....LINEAR, SYMMETRIC TWIST MODE
      DO 310 P =1,PP
      DO 310 M=1,MMP
      ALFA(P,M)=ABS(ETACP(M))/BRAT
310   CONTINUE
      IF(PRINTA) WRITE (U6,6003)
      ASSIGN 24 TO NSTAT
      GO TO 2000
400   CONTINUE
C
C.....PARABOLIC, SYMMETRIC TWIST MODE
      DO 410 M=1,MMP
      ALF = (ETACP(M)/BRAT)**2
      DO 410 P=1,PP
```

```
      ALFA(P,M)=ALF
 410  CONTINUE
      IF(PRINTA) WRITE(U6,6004)
      ASSIGN 25 TO NSTAT
      GO TO 2000
 500  CONTINUE
C
C.....CONTROL SURFACE DOWNWASH MODE
      FLPCAL=.FALSE.
      IF(PRINTB.AND.SYMM)WRITE(U6,6005)
      IF(PRINTB.AND. .NOT.SYMM)WRITE(U6,6006)
      ASSIGN 26 TO NSTAT
      GO TO 2010
 600  CONTINUE
C
C.....USER-SUPPLIED MODES
      BCS(NC)=.FALSE.
      CALL SLOPES(U5,U6,UCO,CONV2,ETACP,CHICP,MMP,
     1PP,ALFA)
      IF(PRINTB.AND.SYMM) WRITE(U6,6007)
      IF(PRINTB.AND. .NOT.SYMM) WRITE(U6,6008)
      ASSIGN 26 TO NSTAT
      GO TO 2010
 700  CONTINUE
C
C.....UNIFORM ANTI-SYMMETRIC MODE
      DO 710 M=1,MMPA
         ALF=0.
      IF(ETACP(M).GT.1.E-5) ALF=1.
      IF(ETACP(M).LT.-1.E-5) ALF=-1.
      DO 710 P=1,PP
      ALFA(P,M)=ALF
 710  CONTINUE
      IF(PRINTA) WRITE(U6,6009)
      ASSIGN 32 TO NSTAT
      GO TO 2000
 800  CONTINUE
C
C.....ROLLING MODE
```

```
      DO 810 M=1,MMPA
      DO 810 P=1,PP
      ALFA(P,M)=ETACP(M)/BRAT
 810  CONTINUE
      IF(PRINTA) WRITE(U6,6010)
      ASSIGN 33 TO NSTAT
      GO TO 2000
 900  CONTINUE
C
C.....ANTISYMMETRIC CONTROL SURFACE DOWNWASH MODE.
C     IF(FLPCAL) CALL FLPDWN(....)
      IF(PRINTB) WRITE(U6,6011)
      ASSIGN 34 TO NSTAT
      GO TO 2010
1000  CONTINUE
C
C.....USER SUPPLIED MODES
      RCAS(NC)=.FALSE.
      CALL SLOPES(U5,U6,UCO,CONV2,ETACP,CHICP,MMPA,
     1PP,ALFA)
      IF(PRINTB.AND.SYMM) WRITE(U6,6012)
      IF(PRINTB.AND. .NOT.SYMM) WRITE(U6,6008)
      ASSIGN 34 TO NSTAT
      GO TO 2010
C
C.....PRINTING AND STORING MODES
C
2000  IF(.NOT.PRINTA) GO TO 2040
      GO TO 2020
2010  IF(.NOT.PRINTB) GO TO 2040
2020  DO 2030 M=1,MMU
2030  WRITE(U6,5001) (ALFA(P,M),P=1,PP)
2040  IF(STORE) WRITE(U8) ((ALFA(P,M),P=1,PP),M=1,MMU)
      GO TO NSTAT, (22,23,24,25,26,32,33,34)
C
C.....CONTROL SHOULD NEVER GET TO HERE
C
C.....INPUT FORMATS
5000  FORMAT(8F10.0  )
```

```
5001  FORMAT(/(1X,8F15.7))
C
C.....OUTPUT FORMATS
6000  FORMAT(////18H UNIFORM DOWNWASH /)
6001  FORMAT(22H ENTER C.M. POS./BREF  /)
6002  FORMAT(////14H PITCHING MODE  //
     1          15H C.M. POS./BREF,5X,1H=,F12.6)
6003  FORMAT(////25H LINEAR, SYMMETRIC TWIST  /)
6004  FORMAT(////28H PARABOLIC, SYMMETRIC TWIST  /)
6005  FORMAT(////21H SYMMETRIC FLAP MODE  /)
6006  FORMAT(////23H UNSYMMETRIC FLAP MODE  /)
6007  FORMAT(////30H USER-SUPPLIED SYMMETRIC MODE  /)
6008  FORMAT(////32H USER-SUPPLIED UNSYMMETRIC MODE  /)
6009  FORMAT(////29H ANTI-SYMMETRIC UNIFORM MODE  /)
6010  FORMAT(////14H ROLLING MODE  /)
6011  FORMAT(////26H ANTI-SYMMETRIC FLAP MODE  /)
6012  FORMAT(////35H USER-SUPPLIED ANTI-SYMMETRIC MODE  /)
6013  FORMAT(////30H MODES STORED AND FILE CLOSED  ////)
      END
      SUBROUTINE STRAT2(XIN,FIN,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
      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. STRAT2'/(1X,1PE20.5))
      STOP 13
C
  500 CONTINUE
      RETURN
  600 DO 610 NO=1,NOUT
  610 FOUT(NO)=FIN(1)
      RETURN
      END
      SUBROUTINE SLOPES(U5,U6,UCO,CONV2,ETACP,CHICP,MMU,
     1PP,ALFA)
      INTEGER U5,U6,UCO,PP,P
      LOGICAL CONV2,LOG1,TEST
      REAL L
      DIMENSION ETACP(MMU), CHICP(PP), ALFA(PP,MMU)
      DIMENSION ETA(20),ALFA2(15,20), ALFA3(30), CHI(41),
     1ALF(41), A(40), B(40), C(40), L(40), G(40),
     2E(40), H(40)
      DIMENSION TEST(15),CHIDUM(30)
C
C.....THE DIMENSION OF THE ABOVE TWO ARRAYS SHOULD BE AS LARGE
```

```
C.....AS PP AND 2*PP, RESPECTIVELEY
C
C.....THE DIMENSION OF ALFA3 MUST BE THE MAXIMUM OF 2*PP AND
C.....NSMAX.
C.....NSMAX IS THE MAXIMUM NUMBER OF SPANWISE STATIONS AT
C.....WHICH DATA SHOULD BE GIVEN.
C.....NSMAX IS A DIMENSION FOR ETA,ALFA2, AND ALFA3.
      DATA NSMAX /20/
C
C.....NTBMAX-1 IS THE MAXIMUM NUMBER OF CHORDWISE LOCATIONS
C.....AT WHICH THE CAMBER OR SLOPE DATA SHOULD BE GIVEN.
C.....NTBMAX IS THE DIMENSION OF CHI AND ALF.
C.....NTBMAX-1 IS THE DIMENSION OF A, B, C, L, G,
C.....E, AND H.  SINCE A IS ALSO USED FOR A CAMBER OR SLOPE
C.....POLYNOMIAL, THE DIMENSION OF A MUST BE AT LEAST 8.
      DATA NTBMAX /41/
C
C.....EPS IS USED FOR NUMERICAL DIFFERENTIATION
C.....OF THE CAMBER DISTRIBUTION DETERMINED BY A TABLE AND
C.....CODIM2 (CONTROLLED DEVIATION INTERPOLATION METHOD).
      DATA EPSDEF /5.E-3/
C
C.....NPPMAX IS THE FIRST DIMENSION OF ALFA2 AND THE MAXIMUM
C.....ALLOWABLE VALUE FOR PP.
      DATA NPPMAX /15/
C
C
      IF(PP.GT.NPPMAX) CALL STOP2(U6,' PP TOO LARGE IN SUB. SLOPES    ',
     1FLOAT(PP))
C
C
C
C
C
C.....NSPSEC IS THE NUMBER OF SPANWISE SECTIONS.
C.....ITYPES DENOTES THE TYPE OF SPANWISE INTERPOLATION.
C.....       0 IMPLIES STRAIGHT LINE INTERPOLATION.
C.....       NOT 0 IMPLIES THAT CODIM2 WILL BE USED.
      IF(CONV2) WRITE(UC0,6000)
```

```
      IF(NSPSEC.GT.NSMAX) CALL STOP2(U6,
     1'  TOO MANY SPANWISE SECTIONS IN SUR. SLOPES.     ',FLOAT(NSPSEC))
      READ(U5,5000) NSPSEC,ITYPES
      IF(ITYPES.EQ.0) GO TO 20
      IF(CONV2)WRITE(UCO,6001)
      READ(U5,5001) XK
20    CONTINUE
C
      DO 500 N=1,NSPSEC
      IF(CONV2)WRITE(UCO,6006)
      READ(U5,5001) ETA(N),TWIST
C
C.....ETA IS THE SPANWISE VARIABLE.  IT MAY EXCEED 1. IN VALUE
C.....IN ORDER TO CONTROL THE CODIM2 INTERPOLATION NEAR THE ENDS.
C.....TWIST IS THE ANGLE OF TWIST.
      IF(CONV2) WRITE(UCO,6002)
      READ(U5,5000) ITYPEC
C
C.....ITYPEC DENOTES THE TYPE OF CHORDWISE SLOPE DEFINITION
C.....THAT THE USER WANTS TO USE.
C.....  0  IMPLIES THAT THERE IS NO CAMBER, JUST TWIST.
C.....  1  IMPLIES THAT A SET OF POLYNOMIALS DEFINES THE SLOPE
C.....     DISTRIBUTION
C.....  2  IMPLIES THAT A SET OF POLYNOMIALS DEFINES THE
C.....     CAMBER DISTRIBUTION.
C.....  3  IMPLIES THAT A TABLE OF VALUES DEFINES THE SLOPE
C.....     DISTRIBUTION.  SUBROUTINE CODIM2 WILL BE USED FOR THE
C.....     INTERPOLATION.
C.....  4  IMPLIES THAT A TABLE OF VALUES DEFINES THE
C.....     CAMBER DISTRIBUTION.  CODIM2 WILL BE USED TO
C.....     DETERMINE THE CAMBER IN THE VICINITY OF THE CONTROL
C.....     POINTS.  THEN NUMERICAL DIFFERENTIATION WILL BE USED
C.....     TO DETERMINE THE SLOPES.
C.....  5  IMPLIES THAT A TABLE OF VALUES DEFINES THE SLOPE
C.....     DISTRIBUTION.  A CUBIC SPLINE FIT WILL BE USED TO
C.....     DETERMINE VALUES AT THE CHORDWISE CONTROL STATIONS.
C.....  6  IMPLIES THAT A TABLE OF VALUES DEFINES THE CAMBER
C.....     DISTRIBUTION.  A CUBIC SPLINE WILL BE DETERMINED FOR
C.....     THE TABLE AND THEN DIFFERENTIATED TO OBTAIN THE
```

```
C.....      SLOPES.
C
        IF(ITYPEC.EQ.0) GO TO 100
        IF(ITYPEC.LT.0) GO TO  450
        IF(ITYPEC.LT.3) GO TO 200
        IF(ITYPEC.LT.7) GO TO 300
        GO TO 450
100     CONTINUE
        DO 110 P=1,PP
110     ALFA2(P,N)=TWIST
        GO TO 500
200     CONTINUE
C
C
C
C.....EVALUATING CHORDWISE SLOPES FROM POLYNOMIALS.
C
        IF(CONV2) WRITE(UCO,6003)
        READ(U5,5000) NPOLS
        CHIMIN=0.
        DO 210 P=1,PP
210     TEST(P)=.TRUE.
        DO 280 NP=1,NPOLS
        IF(CONV2) WRITE(UCO,6004)
C
C.....CHIMAX DENOTES THE UPPER LIMIT OF THE POLYNOMIAL.  THE
C.....            DEFAULT VALUE IS 1.
C.....SCALE  IS A SCALE FACTOR FOR THE POLYNOMIAL
C.....            COEFFICIENTS.  DEFAULT VALUE IS 1.
C.....CHIREF IS THE ORIGIN OF THE POLYNOMIAL.
        READ(U5,5001) CHIMAX,SCALE,CHIREF
        IF(CHIMAX.EQ.0.) CHIMAX=1.
        READ(U5,5001) (A(I),I=1,8)
C
C.....THE NUMBER OF COEFFICIENTS IS LIMITED TO 8.
C.....THE FOLLOWING DETERMINES THE ACTUAL NUMBER.
        NTERMS=8
        DO 215 I=1,8
        IF(A(NTERMS).NE.0.) GO TO 220
```

```
      NTERMS=NTERMS-1
215   CONTINUE
      CALL STOP2(U6,' A CAMBER OR SLOPE POLYNOMIAL IN SUB. SLOPES
     1 HAS ALL 0 COEFFICIENT   ETA=  ',ETA(N))
220   IF(SCALE.EQ.0.)GO TO 226
      DO 225 I=1,NTERMS
225   A(I)=A(I)*SCALE
226   CONTINUE
C
C.....THE FOLLOWING EVALUATES THE SLOPE FOR ALL CONTROL POINTS
C.....WHICH LIE WITHIN THE RANGE OF THE POLYNOMIAL.
      DO 275 P=1,PP
      IF(CHICP(P).LT.CHIMIN) GO TO 275
      IF(CHICP(P).GT.CHIMAX) GO TO 275
      IF(ITYPEC.EQ.2) GO TO 240
      ALFA2(P,N)=A(1) - TWIST
      IF (NTERMS.EQ.1) GO TO 270
      X=CHICP(P)-CHIREF
      DO 230 I=2,NTERMS
      ALFA2(P,N)=ALFA2(P,N) + X*A(I)
230   X=X*(CHICP(P)-CHIREF)
      GO TO 270
240   ALFA2(P,N)=A(2) - TWIST
      IF (NTERMS .LT.3) GO TO 270
      X=CHICP(P)-CHIREF
      DO 245 I=3,NTERMS
      ALFA2(P,N)=ALFA2(P,N) + FLOAT(I-1)*X*A(I)
      X=X*(CHICP(P)-CHIREF)
245   CONTINUE
270   TEST(P)=.FALSE.
275   CONTINUE
      CHIMIN=CHIMAX
280   CONTINUE
      LOG1=.FALSE.
      DO 285 P=1,PP
285   LOG1=LOG1.OR.TEST(P)
      IF(LOG1) CALL STOP2(U6,
     1'  A CHORDWISE POLY. DID NOT COVER ENTIRE CHORD.  ETA=  ',
     2ETA(N))
```

```
      GO TO 500
C
C
C.....EVALUATING THE CHORDWISE SLOPES FROM TABLES.
300   CONTINUE
      IF(CONV2) WRITE(UCO,6005)
      READ(U5,5001) SCALE,A1,AN
      XKC=A1
C
C.....SCALE IS A SCALE FACTOR FOR THE TABLE ENTRIES.
C.....A1 AND AN ARE THE SLOPES AT THE END POINTS OF THE TABLE.
C.....A1 AND AN SHOULD BE PUT IN IF POSSIBLE, ESPECIALLY IF
C.....THE TABLE IS ONE GIVING THE CAMBER.  IF BOTH VALUES ARE
C.....ACTUALLY ZERO, THEN INPUT ONE AS 1.E-30.  IF EITHER IS
C.....GIVEN, THEN BOTH MUST BE GIVEN.
C.....XKC IS THE END POINT INTERPOLATION CONTROL FOR CODIM2.
C.....IF XKC=0., THEN STRAIGHT LINES WILL BE USED IN THE END
C.....INTERVALS.  IF XKC=1., THEN FULL PARABOLIC
C.....INTERPOLATION WILL BE USED IN THE END INTERVALS.  A
C.....VALUE IN BETWEEN WILL GIVE A CURVE IN BETWEEN.
      IF(SCALE.EQ.0.) SCALE=1.
      NTAB=0
310   CONTINUE
      NTAB=NTAB+1
      READ(U5,5001) CHI(NTAB),ALF(NTAB)
      IF(NTAB.GT.NTBMAX) CALL STOP2(U6,
     1' TOO MANY TABLE ENTRIES IN SUB. SLOPES.  ETA=  ',ETA(N))
      IF(CHI(NTAB).LT.98.) GO TO 310
      NTAB=NTAB-1
      IF(ITYPEC.EQ.3) GO TO 315
      IF(ITYPEC.EQ.4) GO TO 350
      GO TO 400
315   CONTINUE
C
C.....THE TABLE IS ONE OF SLOPES AND CODIM2 WILL BE USED.
      CALL CODIM2(ALF,CHI,NTAB,CHICP,ALFA3,PP,XKC)
      DO 320 P=1,PP
      ALFA2(P,N)=ALFA3(P)-TWIST
320   CONTINUE
```

```
      GO TO 500
350   CONTINUE
C
C.....THE TABLE IS ONE OF CAMBER AND CODIM2 AND NUMERICAL DIF-
C.....FERENTIATION WILL BE USED.
      J=-1
C
C     EPSDEF IS THE DEFAULT VALUE FOR EPS (SEE DATA STATEMENTS)
C
      EPS = EPSDEF
      IF(AN.NE.0.)EPS=AN
      DO 355 P=1,PP
      J=J+2
      CHIDUM(J)=CHICP(P)-EPS
      CHIDUM(J+1)=CHICP(P)+EPS
355   CONTINUE
      CALL CODIM2(ALF,CHI,NTAB,CHIDUM,ALFA3,2*PP,XKC)
      J=-1
      DO 360 P=1,PP
      J=J+2
360   ALFA2(P,N)=(ALFA3(J+1)-ALFA3(J))/(2.*EPS) - TWIST
      GO TO 500
400   CONTINUE
C
C.....A CUBIC SPLINE FIT WILL BE USED TO DETERMINE THE CAMBER
C.....OR SLOPE DISTRIBUTION.   SPLIN1 IS CALLED
C.....WHEN THE END POINT DERIVATIVES ARE NOT GIVEN.  IT SETS
C.....THE SECOND DERIVATIVE TO ZERO AT THE END POINTS.  IF THE
C.....END POINT DERIVATIVES ARE KNOWN TO A HIGH DEGREE OF
C.....ACCURACY (.1 PERCENT?) THEN SPLIN2 SHOULD BE USED AS IT
C.....WORKS MUCH BETTER.
      IF(A1.EQ.0. .AND. AN.EQ.0.)
     1CALL SPLIN1(NTAB,CHI,ALF,A,B,C,L,G,E,H)
      IF(A1.NE.0.  .OR. AN.NE.0.)
     1CALL SPLIN2(NTAB,CHI,ALF,A,B,C,L,G,E,H,A1,AN)
C
C.....IN THE ABOVE A,B,C ARE THE LOCAL SPLINE COEFFICIENTS
C.....AND L,G,E, AND H ARE WORKING SPACE.
      DO 430 P=1,PP
```

```
      DO 410 J=1,NTAB
      N1=J-1
      DX=CHICP(P)-CHI(J)
      IF(DX.LT.0.) GO TO 415
  410 CONTINUE
      N1=NTAB
  415 N1=MAX0(1,N1)
      DX=CHICP(P)-CHI(N1)
      IF(ITYPEC.EQ.6) GO TO 420
      ALFA2(P,N)=ALF(N1)+DX*(A(N1)+DX*(B(N1)+DX*C(N1)))-TWIST
      GO TO 430
  420 ALFA2(P,N)=A(N1)+2.*B(N1)*DX+3.*C(N1)*DX**2 - TWIST
C
C.....IN EVALUATING THE ABOVE SPLINE CURVES IT WAS ASSUMED THAT
C.....NO CONTROL POINTS LIED OUTSIDE OF THE TABLE.
  430 CONTINUE
      GO TO 500
  450 CONTINUE
C
C.....EVALUATING THE SLOPES USING A USER-SUPPLIED
C.....SUBROUTINE.
C
      CALL USLOPE(U5,U6,UCO,CONV2,N,ETA(N),PP,CHICP,
     1ITYPEC,ALFA3)
      DO 460 P=1,PP
      ALFA2(P,N)=ALFA3(P)+TWIST
  460 CONTINUE
  500 CONTINUE
C
C
C
C
C
C
C.....AT THIS POINT THE DETERMINATION OF THE SLOPE DISTRIBUTION
C.....AT THE INTERSECTIONS OF THE CHORDWISE CONTROL LINES AND
C.....THE GIVEN ETA STATIONS HAS BEEN ACCOMPLISHED.  NOW IT
C.....IS NECESSARY TO INTERPOLATE TO OBTAIN THE SLOPES
C.....AT THE SPANWISE CONTROL STATIONS GIVEN BY ETACP.
```

```
C
      DO 1000 P=1,PP
      DO 600 N=1,NSPSEC
C
C.....THE MINUS SIGN BELOW OCCURS BECAUSE THE INDUCED DOWNWASH
C.....EQUALS MINUS THE GIVEN SLOPE.
      ALFA3(N)=-ALFA2(P,N)
600   CONTINUE
      IF(ITYPES.EQ.0) CALL STRAT2(ETA,ALFA3,NSPSEC,ETACP,
     1ALF,MMU,UCO)
      IF(ITYPES.NE.0) CALL CODIM2(ALFA3,ETA,NSPSEC,ETACP,
     1ALF,MMU,XK)
      DO 620 M=1,MMU
      ALFA(P,M)=ALF(M)
620   CONTINUE
1000  CONTINUE
      RETURN
C
C
C.....FORMATS
C
5000  FORMAT(16I5  )
5001  FORMAT(8F10.0 )
6000  FORMAT(' ENTER NSPSECS,ITYPES'/)
6001  FORMAT(' ENTER XK'/)
6002  FORMAT(' ENTER ITYPEC'/)
6003  FORMAT(' ENTER NPOLS'/)
6004  FORMAT(' ENTER CHIMAX,SCALE,CHIREF/'/' POLYNOMIAL COEFFICIENTS'/)
6005  FORMAT(' ENTER TABLE SCALE FACTOR, A1(OR XKC),'/
     1' AN(OR EPS), AND THE TABLE. (CHI=99.STOPS)'/)
6006  FORMAT(' ENTER ETA, TWIST'/)
      END
      SUBROUTINE SPLIN1(N,X,F,A,B,C,L,G,E,H)
      REAL L
      DIMENSION X(N)
      DIMENSION F(N)
      DIMENSION A(N)
      DIMENSION B(N)
      DIMENSION C(N)
```

```
      DIMENSION L(N)
      DIMENSION G(N)
      DIMENSION E(N)
      DIMENSION H(N)
      M=N-1
      C(N)=0.
      B(1)=0.
      B(N)=0.
      DO 10 I=1,M
      L(I)=X(I+1)-X(I)
      G(I)=(F(I+1)-F(I))/L(I)
   10 CONTINUE
      E(1)=0.
      DO 20 I=2,M
      J=I-1
      D=2.*(L(J)+L(I))-L(J)*E(J)
      E(I)=L(I)/D
      H(I)=(3.*(G(I)-G(J))-L(J)*H(J))/D
   20 CONTINUE
      E(M)=0.
      I=M
      DO 30 J=2,M
      B(I)=H(I)-E(I)*B(I+1)
      I=I-1
   30 CONTINUE
      DO 40 I=1,M
      A(I)=G(I)-L(I)*(2.*B(I)+B(I+1))/3.
      C(I)=(B(I+1)-B(I))/(3.*L(I))
   40 CONTINUE
      A(N)=A(M)+L(M)*B(M)
      RETURN
      END
```

---

```
      SUBROUTINE SPLIN2(N,X,F,A,B,C,L,G,E,H,A1,AN)
      REAL L
      DIMENSION X(N)
      DIMENSION F(N)
      DIMENSION A(N)
      DIMENSION B(N)
      DIMENSION C(N)
```

```
      DIMENSION L(N)
      DIMENSION G(N)
      DIMENSION E(N)
      DIMENSION H(N)
      M=N-1
      DO 10 I=1,M
      L(I)=X(I+1)-X(I)
      G(I)=(F(I+1)-F(I))/L(I)
   10 CONTINUE
      D=1.5*L(1)+2.*L(2)
      E(2)=L(2)/D
      H(2)=(3.*G(2) - 4.5*G(1) + 1.5*A1)/D
      M=M-1
      DO 20 I=3,M
      J=I-1
      D=2.*(L(J)+L(I))-L(J)*E(J)
      E(I)=L(I)/D
      H(I)=(3.*(G(I)-G(J))-L(J)*H(J))/D
   20 CONTINUE
      D=2.*L(M) + 1.5*L(M+1)-L(M)*E(M)
      B(M+1)=(4.5*G(M+1)-3.*G(M)-1.5*AN-L(M)*H(M))/D
      I=M
      DO 30 J=2,M
      B(I)=H(I)-E(I)*B(I+1)
      I=I-1
   30 CONTINUE
      B(1)=3.*(G(1)-A1)/(2.*L(1)) - .5*B(2)
      M=N-1
      B(N)=3.*(AN-G(M))/(2.*L(M))- .5*B(M)
      DO 40 I=1,M
      A(I)=G(I)-L(I)*(2.*B(I)+B(I+1))/3.
      C(I)=(B(I+1) - B(I))/(3.*L(I))
   40 CONTINUE
      A(N)=AN
      C(N)=0.
      RETURN
      END
```

---

```
      SUBROUTINE STOP2(N,MESAGE,VAL)
      DIMENSION MESAGE(20)
```

```
      WRITE(N,1) MESAGE,VAL
      STOP 13
    1 FORMAT(///' ***** ',20A4,' *****'/' VAL=',1PE15.7)
      END
```