NASA TECHNICAL MEMORANDUM

NASA TM X- 62,325

NASA TM X- 62,325

EQUATION SOLVING PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan and Opal J. Lemmer

Ames Research Center
Moffett Field, California 94035

and

Computer Sciences Corporation
Mountain View, California 94043

January 1974

TABLE OF CONTENTS

EQUATION SOLVING PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan
Ames Research Center
and
Opal J. Lemmer
Computer Sciences Corporation

ABSTRACT

This document is a description of and user's manual for one of a group of FORTRAN programs which, together, can be used for the analysis and design of wings in steady, subsonic flow according to a kernel function method lifting surface theory. This particular program is the one which solves the sets of simultaneous, linear, algebraic equations arising from the thin wing analysis. This program has the capability of striking out rows and columns of the aerodynamic influence matrix and rows of the associated boundary condition vectors (right hand sides). This capability significantly enhances the effectiveness of the kernel function method of lifting surface theory because studies of the convergence of solutions with the number of control points can be done with the calculation of only a single influence matrix.

EQUATION SOLVING PROGRAM FOR AERODYNAMIC LIFTING SURFACE THEORY

Richard T. Medan
Ames Research Center
and
Opal J. Lemmer
Computer Sciences Corporation

1 INTRODUCTION

This document is a description of and user's manual for one of a group of FORTRAN programs which, together, can be used for the analysis and design of wings in steady, subsonic flow according to a kernel function method lifting surface theory. This particular program is the one which solves the sets of simultaneous, linear, algebraic equations arising from the thin wing analysis. This program has the capability of striking out rows and columns of the aerodynamic influence matrix and rows of the associated boundary condition vectors (right hand sides). This capability significantly enhances the effectiveness of the kernel function method of lifting surface theory because studies of the convergence of solutions with the number of control points can be done with the calculation of only a single influence matrix.

The theory behind the program is described in ref. 1 while the other, associated computer programs are described in refs. 2-6.

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 GENERAL DESCRIPTION

This computer program solves sets of simultaneous, linear equations derived from an aerodynamic influence matrix which is stored on disk along with the boundary condition matrix. After the solution is obtained it is stored on a disk file for later use by programs which evaluate loads, pressure distributions, etc.

The program can be operated either in a batch or conversational mode. If done conversationally the user is prompted for input. By entering various commands the user can have the solutions stored or printed, the boundary condition matrix printed, spanwise and chordwise control points printed, etc. Operating instructions are given in section 4 while a complete description of the commands is included in sections 5 and 6.

The program has the capability of deleting rows and columns by the use of the DSM, DCM, DSP, DCP, and DPP commands. Deleting rows deletes control points while deleting columns deletes pressure modes. After a case is completed the user has the option of: (1) going through the same file again deleting more rows and/or columns, (2) starting a new wing, or (3) halting the program.

If the wing is symmetric, then the program will employ symmetry to reduce the amount of computation. In the case of wing symmetry the boundary conditions stored on the boundary condition file can be symmetric, antisymmetric, or both and the aerodynamic influence matrix can contain symmetric, antisymmetric or both types of spanwise pressure modes. If the boundary condition file contains both types of boundary conditions and the influence matrix file both types of spanwise modes, then the user can obtain solutions for symmetric cases only, antisymmetric cases only, or both.

3 METHOD OF SOLUTION

Householder's method for solving simultaneous linear equations is utilized. This method was chosen because it is applicable to both square and rectangular matrices and in the case of rectangular matrices it is not necessary to least square the matrices first (ref. 7).

There is a storage advantage in that the whole matrix need not be in core at one time. The program reads in only as many rows as there are chordwise control points. The method was taken from ref. 7 and modified to achieve the capability of striking out rows and columns.

This subroutine triangularizes the influence matrix by means of orthogonal transformation matrices which preserve the conditioning of the matrix. This, along with the elimination of the need for a least squares routine greatly improves the numerical accuracy and stability of the solution over that of the Gaussian reduction method.

4 USERS INSTRUCTIONS

4.1 INITIAL SETUP--AMES' 360/67 TSS

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
```

4.2 CONVERSATIONAL MODE--AMES' 360/67 TSS

See section 4.1 for the initial setup. All integer data should be entered in a 16I5 format, all alphanumeric data left justified beginning in column 1.

After logging on proceed as follows:

User: ames usyslib
joblibs sysulib
jblb medan

User: call solvr$

Prog: ENTER BATCH

User: Carriage return

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 the AMES' TSS system the output will be found on the file named OUTPUT.SOLVR.NX where X is the numerical value of ODISK. The program issues its own DDEF commands so no control cards are

needed. The program uses logical unit 4 for this output.

Prog: ENTER ID1, ID2, ID3

User: Enter identification numbers

ID1 Identification number of the geometry file from which the AIM and BC files have been derived.

ID2 Identification number of AIM file.

ID3 Identification number of the boundary condition file.

These identification numbers are used to create DDEF's which are issued by the subroutines AIMFIL, BCFIL and BNKFIL. This makes it unnecessary for the user to issue DDEF commands on the TSS system. These numbers are also checked against the identification numbers on the files. This is unnecessary in the TSS version, but would be helpful as an aid in insuring that the correct files are being used if the program were to be converted to another computer.

Prog: ENTER NTYPE

(NTYPE will not be requested for an unsymmetric wing)

User: Enter:

-1 to obtain solutions to the antisymmetric cases only,

0 to obtain solutions to the symmetric cases only, or

+1 to obtain solutions for both types of cases.

Prog: +

User: At this point the user must begin entering commands. After each command he will be prompted for supplemental input or with another + sign indicating that he should enter another command. The full list of commands is given in section 6 while a basic list for the beginner is included in section 5. After the NEW or KNEW command the next input will be ODISK followed by ID1, ID2, ID3, and NTYPE. Use the HALT command to terminate execution.

4.3 BATCH MODE--AMES' 360/67 TSS

See section 4.1 for the initial setup. All integer data should be entered in a 16I5 format, all alphanumeric data left justified beginning in column 1.

CARD 1....AMES USYSLIB

CARD 2....JOBLIBS SYSULIB

CARD 3....JBLB MEDAN

CARD 4....CALL SOLVR$

CARD 5....T

CARD 6....File identification numbers--ID1, ID2, ID3

ID1 Identification number of the geometry file from which the AIM and BC files have been derived.

ID2 Identification number for AIM file.

ID3 Identification number of BC file.

These identification numbers are used to create DDEF's which are issued by the subroutines AIMFIL, BCFIL and BNKFIL. This makes it unnecessary to include control cards on the TSS system. These numbers are also checked against the identification numbers on the files. This is unnecessary in the TSS version,

but would be helpful as an aid in insuring that the correct files are being used if the program were to be converted to another computer.

Card 7....NTYPE (omit for an unsymmetric wing).

Enter:

-1 to obtain solutions to the antisymmetric cases only.

0 to obtain solutions to the symmetric cases only.

+1 to obtain solutions for both types of cases.

NEXT.....Follow by cards containing commands (described in sections 5 and 6) in columns 1-3 each followed by a card containing command arguments (if the command calls for arguments) as specified in section 6. Following a NEW or KNEW command input resumes with card 6. The HALT command must be the last card of the input deck.

4.4 OTHER COMPUTERS

Remove all calls to BCFIL, AIMFIL, BNKFIL, OBEY, and CVRT in the main program and use appropriate tape or disk control cards in their place. These, hopefully, are the only changes that need to be made since considerable effort was made to code the program in standard FORTRAN. Then follow the instructions, where appropriate, in section 4.2 and 4.3.

5 BASIC COMMANDS

These commands will enable a new user to run the program without deleting rows or columns or having any options. The solutions will be stored. There is no input associated with these commands.

HALT.....Terminates program. This command is needed and is the last item of input.

NEW.....Begin a new wing.

START.....Begin processing.

6 FULL DESCRIPTION OF COMMANDS

In this section the commands which control the flow of the program are described. In all cases the first three letters of a command are sufficient input. All numeric fields are to be entered in a 1615 format. All alphanumeric data is to be left justified and begin in column 1. The input associated with some of the commands is prompted for in conversational processing and in batch processing is to be on cards immediately following the command card. All commands except HALT may be issued more than once. This is especially useful in the conversational mode for correcting data entered in error. Many commands have opposite commands (i.e. STORE and NSTORE); the one given most recently is the one that governs. The commands are given below in alphabetical order:

CONTINUE

EFFECT....Causes processing to continue in the batch mode when an erroneous command is encountered. Otherwise execution will terminate. In the conversational mode an invalid command message will be issued and processing will continue.

DCM--Delete Chordwise Modes

Input.....NMAX2
Effect....Will limit the number of chordwise modes to NMAX2. NMAX2 must be $\leq$ any previously entered NMAX2 and must be $\leq$ the original number of chordwise modes, NMAX.

DCP--Delete Chordwise control Points

Input.....(PDEL)--an integer array of not more than 17 points to be deleted. The array must be in ascending order. The indices referred to are always the original indices (i.e. if there were 5 chordwise control points and the user deleted the 2nd and 4th, then the indices of the remaining points would be 1, 3, 5 and not 1, 2, 3).
Effect....The chordwise control points specified are deleted.

DPP--Delete a Particular Point

Input.....P,M (P and M are integers)--These are always the original indices. P denotes the chordwise control point number (P = 1 denotes the point closest to the leading edge). M denotes the spanwise control point number (M = 1 denotes the spanwise control point closest to the right side edge).
Effect....The control point with the indices P,M will be deleted.

DSM--Delete Spanwise Modes

Input.....KK2; The new maximum number of spanwise modes. The number includes all modes, not just the ones being used. For example if KK2 were 11, then this would include 6 symmetric and 5 antisymmetric modes. KK2 must be $\leq$ to any previously entered KK2 and/or $\leq$ KK.
Effect....Limits the new maximum number of spanwise modes to KK2.

DSP--Delete Spanwise Control Points

Input.....(MDEL) --an integer array of not more than 17 points to be deleted. These always refer to the original control points and must be in ascending order.

Effect....The spanwise control points with indices in array (MDEL) will be deleted.

HALT

Effect....Terminates execution.

KNEW

Effect....Begins a new wing leaving the following commands in effect: STORE or NSTORE, SUM or NSUM, PCP or NPCP, RHS or NRHS, CONTINUE, PIND or NPIND.

NEW

Effect....Begins a new wing resetting all commands and variables to their default settings.

NPC

Effect....Reverses effect of PCP command; the control points will not be printed. (Sets logical variable POINTS to .FALSE.). This command need not be given unless the PCP command was previously given since this command reflects the default status.

NPIND

Effect....Reverses effect of PIND command; indices of rows and columns retained will not be printed. Sets logical variable PIND to .TRUE. This command need not be given unless the PIND command was previously given since this command reflects the default status.

NPS

Effect....Reverses the effect of PSOL command; the solutions will not be printed. This command need not be given unless the PSOL command was previously given since this command reflects the default status.

NRHS

Effect....Reverses the effect of RHS command; the right hand sides will not be printed. This command need not be given unless the RHS command was previously given since this command reflects the default status.

NSTORE

Effect....Reverses effect of STORE command--solutions will not be stored on solution file. Since STORE is a default this command must be issued to prevent solutions being stored.

NSUM

Effect....Reverses effect of SUMMARY command--a summary of commands will not be printed. This command need not be given unless the SUM command was previously given since this command reflects the default status.

PCP--Print Control Points

Effect....Control points will be printed.

PIND--Print INDices

Effect....The indices of rows and columns retained will be printed.

PSOL--Print SOLutions

Effect....Solutions will be printed.

RCP--Restore Control Points

Effect....Restores the control points to that set originally on the influence matrix and boundary condition files. This command can be used if the conversational user accidently specifies the wrong control points with the DCP or DSP commands.

RHS--Right Hand Sides

Effect....Right hand sides will be printed.

RPM--Restore Pressure Modes

Effect....Restores the pressure modes. This command can be used if the conversational user accidently gives a smaller value for NMAX2 or KK2 (DCM and DSM commands) than intended.

START

Effect....Causes processing to begin. This is the last command that can be given for a particular solution. After the solution has been obtained and (if NSTORE has not been given) stored, the user may delete more rows and/or columns and obtain more solutions.

STORE

Effect....Solutions will be stored on solution file. This command need not be given unless the NSTORE command was previously given since this command reflects the default status.

SUMMARY

Effect....A summary of commands will be printed.

TSS

Input.....A TSS command of 80 characters or less.
Effect....The command is passed to the AMES' TSS operating system. After the system processes the command, control returns to the program. This command is a special one for the AMES' TSS version of the program.

7 SAMPLE CASE

A sample conversational terminal session on the Ames' 360/67 TSS computer system is reproduced in this section with additional comments added in parenthesis. During this session 5 sets of solutions were determined. Each set consisted of 2 symmetric cases and 1 antisymmetric case. Each of these solution sets was derived from the same influence matrix to illustrate how one can study the convergence of the kernel function method in a very efficient manner (i.e. influence matrices which are submatrices of larger influence matrices are not calculated).

For the first solution set the entire influence matrix was used (i.e., no control points or pressure modes were deleted). The chordwise control points used and the numbering of these control points are shown in fig. 1 while the spanwise control points and numbering are shown in fig. 2. For this case the total number of control points equalled the total number of pressure modes. (30 for the symmetric cases and 25 for the antisymmetric case)

For the second solution set the number of spanwise pressure modes was decreased from 11 to 5 (3 symmetric and 2 antisymmetric modes) while no control points were deleted. The solution thus obtained was a solution in the least squares sense (30 or 25 equations in either 15 or 10 unknowns).

For the third solution set the odd numbered spanwise control points were deleted leaving the spanwise control points shown in fig. 3. Since the effect of row and column deletion are accumulative, this left as many control points as unknowns (15 for the symmetric cases and 10 for the antisymmetric cases).

For the fourth solution set the second and fourth chordwise control points were deleted leaving chordwise control points on the leading edge, mid-chord, and trailing edge. Also the number of chordwise pressure modes was reduced from 5 to 3 leaving as many equations as unknowns (9 for the symmetric case and 6 for the unsymmetric case).

For the fifth case it was desired to consider 3

chordwise control points and chordwise pressure modes and 11 spanwise control points and spanwise pressure modes. For this it was necessary to restore all the control points and pressure modes using the RCP and RPM commands. Then the second and fourth chordwise control points were deleted and the number of chordwise pressure modes was reduced to 3, yielding the desired matrix.

The output from the following session is given in Appendix II:

```
LOGON userid,password,terminal id
AMES USYSLIB
JOBLIBS SYSULIB
JBLB MEDAN
CALL SOLVR$
ENTER BATCH
(carriage return)
ENTER ODISK
    1
 OUTPUT IS ON ...OUTPUT.SOLVR.N1...
 CANCELLED: DDNAME FT04F001 UNKNOWN
(Messages such as the above occur because of the
automatic file defining feature and do not indicate any
error.)
ENTER IDD1,IDD2,IDD3
    4    2    9
 CANCELLED: DDNAME FT11F001 UNKNOWN
 CANCELLED: DDNAME FT08F001 UNKNOWN
ENTER VALUE FOR NTYPE, 0 FOR SYMMETRIC, -1 FOR
ANTI-SYMMETRIC, 1 FOR BOTH
    1
+
PCP
+
PSOL
+
RHS
+
SUMMARY
+
TSS
CPUTIME?
  4.350 SECONDS
+
START
```

```
ID4=   15
 CANCELLED: DDNAME FT12F001 UNKNOWN
+
TSS
CPUTIME?
  9.208 SECONDS
+
DSM
ENTER NEW MAXIMUM NUMBER OF SPANWISE MODES  OLD MAXIMUM
IS 11
ORIGINAL VALUE WAS 11
    5
+
PIND
+
START
ID4=   16
+
TSS
CPUTIME?
  11.996 SECONDS
+
DSP
ENTER SPANWISE CONTROL POINTS TO BE DELETED, 17 MAXIMUM
ASCENDING ORDER
    1    3    5
+
START
ID4=   17
+
TSS
CPUTIME?
  14.072 SECONDS
+
DCP
ENTER CHORDWISE  POINTS TO  BE DELETED  17  MAXIMUM  IN
ASCENDING ORDER
    2    4
+
DCM
ENTER  NEW NUMBER  OF CHORDWISE  MODES OLD  VALUE IS  5
ORIGINAL VALUE WAS 5
    3
+
START
ID4=   18
+
```

```
TSS
CPUTIME?
  15.956 SECONDS
+
RCP
+
RPM
+
DCP
ENTER CHORDWISE  POINTS  TO  BE DELETED  17 MAXIMUM  IN
ASCENDING ORDER
    2    4
+
DCM
ENTER NEW NUMBER OF  CHORDWISE  MODES  OLD VALUE  IS  5
ORIGINAL VALUE WAS 5
    3
+
START
ID4=   19
+
TSS
CPUTIME?
  18.529 SECONDS
+
HALT
PROGRAM TERMINATED BY HALT COMMAND
 TERMINATED: STOP
PRINT OUTPUT.SOLVR.N1,PRTSP=EDIT,STATION=RMT05
 PRINT BSN=2232,   300 LINES
LOGOFF
```

8 SUBROUTINES

8.1 STOP2 This subroutine prints an error message and terminates execution. It assumes an 80 character message with words of 4 characters each. If the message from the calling program is not this long, then some garbage will appear at the end of the message. This program will require revision in the FORMAT statement if the program is used on a system with word length differing from 4 characters.

8.2 SOLVIT Equation solving subroutine which uses Householder's method described in section 3. Also see the program listing for more documentation.

8.3 KRUNCH This subroutine is used in determining the arrays (INDCOL) and (INDROW) which are the indices of rows and columns to be retained. See the program listing for further explanation.

8.4 AIMFIL This subroutine is used in the AMES' TSS version only. Its purpose is to issue a RELEASE command and DDEF command for the influence matrix file.

8.5 BCFIL This subroutine is used in the AMES' TSS version only. Its purpose is to issue a RELEASE command and a DDEF command for the boundary condition file.

8.6 BNKFIL This subroutine is used in the AMES' TSS version only. Its purpose is to issue a RELEASE command and a DDEF command for the solution file.

8.7 OBEY — This is an Ames' library subroutine used for passing commands from the program to the TSS operating system.

8.8 CVRT — This is an Ames' library program used for preparing RELEASE and DDEF commands for use in OBEY.

9 INPUT FILES

The following disk files are read by the program. For a complete description of the information on the files see refs. 2-3. 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.

9.1 Aerodynamic Influence Matrix File (AIM file)

This file is a variable record length file and is read from unit 11.

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 (ref. 2).

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.

9.2 Boundary Condition File (BC file)

This file is a variable record length file read from unit 8.

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 for

symmetric cases where NSYM equals the number of symmetric cases. The next NASYM records are the right-hand sides for antisymmetric cases where NASYM equals the number of antisymmetric cases. In the case of an unsymmetric wing there will be NSYM + NASYM right-hand sides.

This file is generated either by the geometry program (ref. 2) or the boundary condition program (ref. 4).

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.

9.3 Identification Number File

This file is read from unit 9 and rewritten on unit 9.

This file contains identification numbers in binary form. The fourth number of the file is the one used to identify the solution. Whenever solutions are to be stored the file is read and updated. This new identification number is then written on the solution file.

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

10 SOLUTION FILE

This is a binary file with variable record lengths and is written on unit 12. The AMES' TSS version issues its own DDEF commands (control cards) for the file so none need be given by the user. On the Ames' TSS system this file has the name BNK.XI.XJ.XK.XL where I is the numerical value of ID1, J is the numerical value of ID2, K is the numerical value of ID3, and L is the numerical value of ID4. On another system appropriate control cards will have to be supplied for unit 12.

FIRST RECORD

ID1, ID2, ID3, ID4, TITLE, SYM, NSYM, (BCS(I),I=1,10), (BCAS(I),I=1,10), NMAX2, KK2, JJ, PPNEW, MMNEW, NROWSA, DELTA0, EPS, MACH

ID1	Identification number of the geometry file from which AIM file and BC files were derived.
ID2	Identification number from AIM file.
ID3	Identification number from BC file.
ID4	Identification number assigned to this solution.
(TITLE(26))	Title information.
SYM	Logical variable whose value is .TRUE. for a symmetric wing and .FALSE. for an unsymmetric wing.
NSYM	Number of symmetric cases.
NASYM	Number of antisymmetric cases.
(BCS)	Ten element logical array identifying the types of the symmetric cases.
(BCAS)	Ten element logical array identifying the types of the antisymmetric cases.

NMAX2	New number of chordwise modes.
KK2	New number of spanwise modes.
JJ	The number of spanwise integration points used by the influence matrix program. This number comes from the AIM file.
PPNEW	New number of chordwise points.
MMNEW	New number of spanwise points.
NROWSA	The number of rows after deletion. This number is equal to the number of control points actually used. If both symmetric and antisymmetric solutions have been obtained, then NROWSA is equal to the number of control points used for the symmetric cases.
DELTA0	Parameter in the calculation of the influence matrix.
EPS	Parameter in the calculation of the influence matrix.
MACH	Mach number (floating point).

NEXT NSYM RECORDS

(X(J),J=1,NROWSA)	The nonzero coefficients, Bnk, of the expansion for the pressure for the symmetric cases. These are stored so as to be compatible with the following READ statement: READ(12)((BNK(N,K),N=1,NMAX2),K=1, KK2,KJUMP) where KJUMP=2 if the wing is symmetric or KJUMP=1 if the wing is unsymmetric.

NEXT NASYM RECORDS

(X(J),J=1,NROWSA)	The nonzero coefficients, Bnk, of the expansion for the pressure for the antisymmetric cases. These are stored so as to be compatible with the following READ statement:

```
READ(12)((BNK(N,K),N=1,NMAX2),K=
KJUMP,KK2,KJUMP).
```

11 REFERENCES

1. Medan, R. T.: Improvements to the Kernel Function Method of Steady, Subsonic Lifting Surface Theory. NASA Rept. No. TMX-62,327, Mar. 1974.

2. Medan, R. T.: Geometry Program for Aerodynamic Lifting Surface Theory. NASA Rept. No. TMX-62,309, Sept. 1973.

3. Medan, R. T., and Ray, K. S.: Influence Matrix Program for Aerodynmaic Lifting Surface Theory. NASA Rept. No. TMX 62,324, Dec. 1973.

4. Medan, R. T., and Ray, K. S.: Boundary Condition Program for Aerodynamic Lifting Surface Theory, NASA Rept. No. TMX-62,323, Dec. 1973.

5. Medan, R. T., and Ray, K. S.: Normal Loads Program for Aerodynamic Lifting Surface Theory, NASA Rept. No. TMX-62,326, Feb. 1974.

6. Medan, R. T., and Ray, K. S.: Plotting Program for Aerodynamic Lifting Surface Theory, NASA Rept. No. TMX-62,321, Nov. 1973.

7. 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.

FIGURES

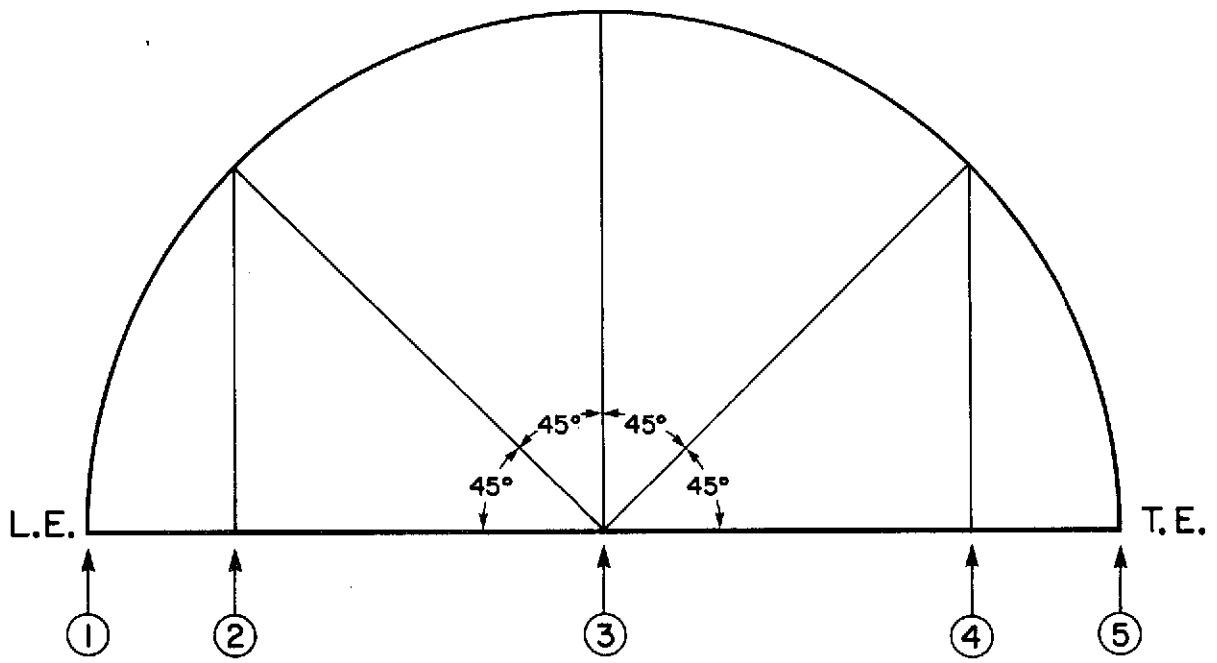

Figure 1 - The Chordwise Control Point Distribution and Numbering Scheme for the Sample Case.

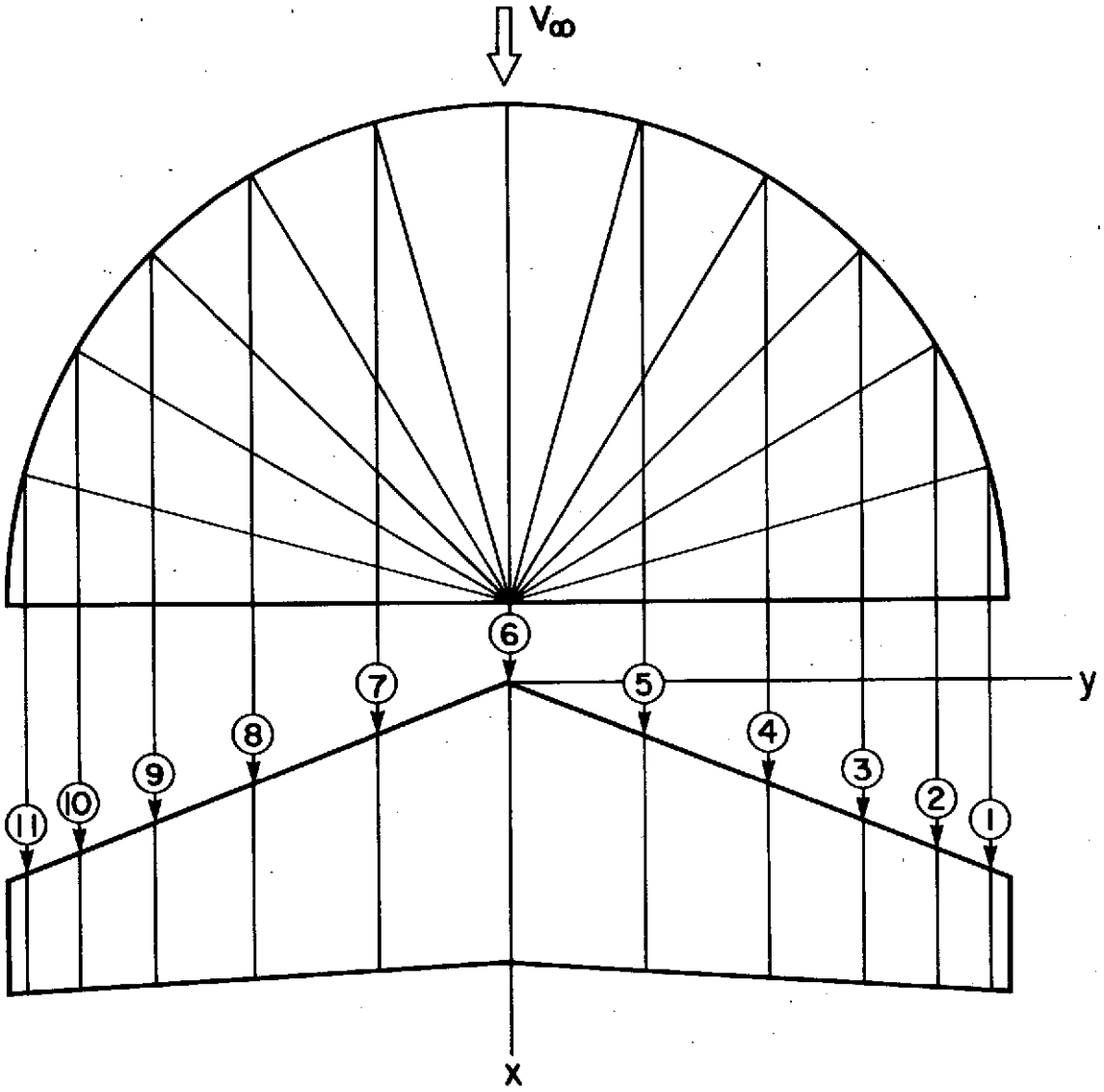

Figure 2 - The Spanwise Control Point Distribution and Numbering Scheme for the Sample Case. (Note: The wing in the figure is not the wing used in the sample case.)

APPENDIX I

OUTPUT FROM SAMPLE CASE

```
DETERMINATION OF BNK COEFFICIENTS
=================================
RECTANGULAR WING    AR = 2    11-13-73

ID1      =    4
ID2      =    2
ID3      =    9
ID4      =   15
JJ       =  191
UNSYM    =    0
MODES    =    1
NTYPE    =    1
PPNEW    =    5
MMNEW    =   11
NMAX2    =    5
KK2      =   11
DELTAO   =    4.0000
MACH     =    0.0000
RCS      = TTFFFFFFFF
BCAS     = FTFFFFFFFF


SUMMARY OF COMMANDS

PCP
PSOL
RHS
SUMMARY
TSS
START

CHORDWISE CONTROL POINTS USED

   0.00000
   0.14645
   0.50000
   0.85355
   1.00000
```

SPANWISE CONTROL POINTS USED

0.96593
0.86603
0.70711
0.50000
0.25882
-0.00000

BOUNDARY CONDITIONS AT ORIGINAL CONTROL POINTS
(L.E. TO T.E.. RIGHT TIP TO LEFT TIP)

CASE 1

1.00000	1.00000	1.00000	1.00000	1.00000
1.00000	1.00000	1.00000	1.00000	1.00000
1.00000	1.00000	1.00000	1.00000	1.00000
1.00000	1.00000	1.00000	1.00000	1.00000
1.00000	1.00000	1.00000	1.00000	1.00000
1.00000	1.00000	1.00000	1.00000	1.00000

CASE 2

0.00000	0.14645	0.50000	0.85355	1.00000
0.00000	0.14645	0.50000	0.85355	1.00000
0.00000	0.14645	0.50000	0.85355	1.00000
0.00000	0.14645	0.50000	0.85355	1.00000
0.00000	0.14645	0.50000	0.85355	1.00000
0.00000	0.14645	0.50000	0.85355	1.00000

NUMBER OF CONTROL POINTS = 30
NUMBER OF PRESSURE MODES = 30

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.11773389E 22

SOLUTION FOR SYMMETRIC CASES

	CASE 1	CASE 2
1	0.9592875	0.2937329
2	-0.3434652	0.6579308
3	-0.0878328	-0.0227847
4	-0.0263244	0.0056143
5	-0.0096903	-0.0019472

6	0.1084463	0.0032115
7	-0.1937814	0.0297757
8	-0.1148094	-0.0201028
9	-0.0630316	0.0021874
10	-0.0266566	-0.0050513
11	0.0439750	0.0042856
12	-0.0883476	-0.0059570
13	-0.0770952	-0.0120187
14	-0.0612980	-0.0035019
15	-0.0312412	-0.0055371
16	0.0233391	0.0031349
17	-0.0468022	-0.0057873
18	-0.0452612	-0.0070768
19	-0.0428134	-0.0045091
20	-0.0246244	-0.0042022
21	0.0119725	0.0017669
22	-0.0239923	-0.0034023
23	-0.0237498	-0.0037836
24	-0.0236830	-0.0030724
25	-0.0144601	-0.0024254
26	0.0036820	0.0005621
27	-0.0073745	-0.0010958
28	-0.0073520	-0.0011825
29	-0.0074018	-0.0010369
30	-0.0046311	-0.0007719

BOUNDARY CONDITIONS AT ORIGINAL CONTROL POINTS
(L.E. TO T.E.. RIGHT TIP TO LEFT TIP)

CASE 3

0.96593	0.96593	0.96593	0.96593	0.96593
0.86603	0.86603	0.86603	0.86603	0.86603
0.70711	0.70711	0.70711	0.70711	0.70711
0.50000	0.50000	0.50000	0.50000	0.50000
0.25882	0.25882	0.25882	0.25882	0.25882

NUMBER OF CONTROL POINTS = 25
NUMBER OF PRESSURE MODES = 25

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.38441723E 18

SOLUTION FOR ANTI-SYMMETRIC CASES

CASE 1

1 0.3958588
2 -0.3087006
3 -0.1107676
4 -0.0433692
5 -0.0158296

6 0.0606708
7 -0.1189108
8 -0.0854544
9 -0.0566794
10 -0.0255904

```
11  0.0270721
12 -0.0544521
13 -0.0503660
14 -0.0443864
15 -0.0238526

16  0.0141472
17 -0.0283824
18 -0.0277684
19 -0.0272557
20 -0.0161263

21  0.0062318
22 -0.0124904
23 -0.0124014
24 -0.0124801
25 -0.0077080

SOLUTION FOUND AND STORED


FINISHED
```

```
DETERMINATION OF BNK COEFFICIENTS
=================================
RECTANGULAR WING    AR = 2    11-13-73

ID1      =    4
ID2      =    2
ID3      =    9
ID4      =   16
JJ       =  191
UNSYM    =    0
MODES    =    1
NTYPE    =    1
PPNEW    =    5
MMNEW    =   11
NMAX2    =    5
KK2      =    5
DELTAO   =    4.0000
MACH     =    0.0000
BCS      = TTFFFFFFFF
BCAS     = FTFFFFFFFF


SUMMARY OF COMMANDS

TSS
DSM
PIND
START

CHORDWISE CONTROL POINTS USED

    0.00000
    0.14645
    0.50000
    0.85355
    1.00000
```

SPANWISE CONTROL POINTS USED

0.96593
0.86603
0.70711
0.50000
0.25882
-0.00000

INDICES OF COLUMNS TO BE RETAINED

1	2	3	4	5	11	12
13	14	15	21	22	23	24
25						

INDICES OF ROWS TO BE RETAINED

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

NUMBER OF CONTROL POINTS = 30
NUMBER OF PRESSURE MODES = 15

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.92918963E 09

SOLUTION FOR SYMMETRIC CASES

	CASE 1	CASE 2
1	0.9563015	0.2933132
2	-0.3391642	0.6583294
3	-0.0858390	-0.0223944
4	-0.0247415	0.0057314
5	-0.0088764	-0.0018154
6	0.1117279	0.0036379
7	-0.2003296	0.0290937
8	-0.1217275	-0.0213837
9	-0.0688910	0.0018656
10	-0.0288136	-0.0055897
11	0.0466561	0.0046587
12	-0.0936345	-0.0063517
13	-0.0796093	-0.0128674
14	-0.0597491	-0.0029139
15	-0.0285094	-0.0053198

INDICES OF COLUMNS TO BE RETAINED

6	7	8	9	10	16	17
18	19	20				

INDICES OF ROWS TO BE RETAINED

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25			

NUMBER OF CONTROL POINTS = 25
NUMBER OF PRESSURE MODES = 10

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.74841275E 06

SOLUTION FOR ANTI-SYMMETRIC CASES

CASE 1

1	0.3965770
2	-0.3107769
3	-0.1147077
4	-0.0454141
5	-0.0156609
6	0.0668647
7	-0.1312739
8	-0.0916794
9	-0.0548593
10	-0.0220116

SOLUTION FOUND AND STORED

FINISHED

```
DETERMINATION OF BNK COEFFICIENTS
=================================
RECTANGULAR WING    AR = 2    11-13-73

ID1      =    4
ID2      =    2
ID3      =    9
ID4      =   17
JJ       =  191
UNSYM    =    0
MODES    =    1
NTYPE    =    1
PPNEW    =    5
MMNEW    =    5
NMAX2    =    5
KK2      =    5
DELTAO   =    4.0000
MACH     =    0.0000
BCS      =  TTFFFFFFFF
BCAS     =  FTFFFFFFFF


SUMMARY OF COMMANDS

TSS
DSP
START

CHORDWISE CONTROL POINTS USED

    0.00000
    0.14645
    0.50000
    0.85355
    1.00000
```

SPANWISE CONTROL POINTS USED

0.86603
0.50000
-0.00000

INDICES OF COLUMNS TO BE RETAINED

1	2	3	4	5	11	12
13	14	15	21	22	23	24
25						

INDICES OF ROWS TO BE RETAINED

6	7	8	9	10	16	17
18	19	20	26	27	28	29
30						

NUMBER OF CONTROL POINTS = 15
NUMBER OF PRESSURE MODES = 15

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.17823940E 07

SOLUTION FOR SYMMETRIC CASES

	CASE 1	CASE 2
1	0.9548624	0.2930498
2	-0.3345991	0.6591860
3	-0.0790099	-0.0212807
4	-0.0175583	0.0067213
5	-0.0047418	-0.0010586

```
  6   0.0948424     0.0012067
  7  -0.1665186     0.0335271
  8  -0.0880090    -0.0156435
  9  -0.0367432     0.0053368
 10  -0.0117176    -0.0023978

 11   0.0195498     0.0010045
 12  -0.0393625     0.0000173
 13  -0.0300078    -0.0045080
 14  -0.0172923     0.0009768
 15  -0.0065783    -0.0012288

INDICES OF COLUMNS TO BE RETAINED

  6      7      8      9     10     16     17
 18     19     20

INDICES OF ROWS TO BE RETAINED

  6      7      8      9     10     16     17
 18     19     20
NUMBER OF CONTROL POINTS =   10
NUMBER OF PRESSURE MODES =   10


THE DETERMINANT OF(A-TRANSPOSE)*A =  0.86715039E 04


SOLUTION FOR ANTI-SYMMETRIC CASES


      CASE  1

  1   0.3885300
  2  -0.2940048
  3  -0.0962846
  4  -0.0290404
  5  -0.0077700
```

```
 6  0.0452002
 7 -0.0878599
 8 -0.0553636
 9 -0.0276744
10 -0.0093136

SOLUTION FOUND AND STORED


FINISHED
```

```
DETERMINATION OF BNK COEFFICIENTS
=================================
RECTANGULAR WING   AR = 2   11-13-73

ID1      =   4
ID2      =   2
ID3      =   9
ID4      =  18
JJ       = 191
UNSYM    =   0
MODES    =   1
NTYPE    =   1
PPNEW    =   3
MMNEW    =   5
NMAX2    =   3
KK2      =   5
DELTAO   =   4.0000
MACH     =   0.0000
BCS      = TTFFFFFFFF
BCAS     = FTFFFFFFFF


SUMMARY OF COMMANDS

TSS
DCP
DCM
START

CHORDWISE CONTROL POINTS USED

   0.00000
   0.50000
   1.00000
```

SPANWISE CONTROL POINTS USED

0.86603
0.50000
-0.00000

INDICES OF COLUMNS TO BE RETAINED

1 2 3 11 12 13 21
22 23

INDICES OF ROWS TO BE RETAINED

6 8 10 16 18 20 26
28 30

NUMBER OF CONTROL POINTS = 9
NUMBER OF PRESSURE MODES = 9

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.10721305E 05

SOLUTION FOR SYMMETRIC CASES

	CASE 1	CASE 2
1	0.9595866	0.2921880
2	-0.3473339	0.6642578
3	-0.0799000	-0.0210888
4	0.1041752	0.0008584
5	-0.1885846	0.0355538
6	-0.0907550	-0.0151825
7	0.0233645	0.0011664
8	-0.0477085	-0.0001797
9	-0.0314681	-0.0043278

```
INDICES OF COLUMNS TO BE RETAINED

  6       7       8      16      17      18

INDICES OF ROWS TO BE RETAINED

  6       8      10      16      18      20
NUMBER OF CONTROL POINTS =     6
NUMBER OF PRESSURE MODES =     6


THE DETERMINANT OF(A-TRANSPOSE)*A =  0.41941284E 03


SOLUTION FOR ANTI-SYMMETRIC CASES


       CASE  1

  1  0.3961701
  2 -0.3133379
  3 -0.0981125

  4  0.0517030
  5 -0.1026420
  6 -0.0576963

SOLUTION FOUND AND STORED


FINISHED
```

DETERMINATION OF BNK COEFFICIENTS
==========
RECTANGULAR WING AR = 2 11-13-73

```
ID1      =   4
ID2      =   2
ID3      =   9
ID4      =  19
JJ       = 191
UNSYM    =   0
MODES    =   1
NTYPE    =   1
PPNEW    =   3
MMNEW    =  11
NMAX2    =   3
KK2      =  11
DELTAO   =   4.0000
MACH     =   0.0000
RCS      = TTFFFFFFFF
BCAS     = FTFFFFFFFF
```

SUMMARY OF COMMANDS

```
TSS
RCP
RPM
DCP
DCM
START
```

CHORDWISE CONTROL POINTS USED

```
   0.00000
   0.50000
   1.00000
```

SPANWISE CONTROL POINTS USED

```
  0.96593
  0.86603
  0.70711
  0.50000
  0.25882
 -0.00000
```

INDICES OF COLUMNS TO BE RETAINED

```
 1     2     3    11    12    13    21
22    23    31    32    33    41    42
43    51    52    53
```

INDICES OF ROWS TO BE RETAINED

```
 1     3     5     6     8    10    11
13    15    16    18    20    21    23
25    26    28    30
```

NUMBER OF CONTROL POINTS = 18
NUMBER OF PRESSURE MODES = 18

THE DETERMINANT OF(A-TRANSPOSE)*A = 0.62640890E 14

SOLUTION FOR SYMMETRIC CASES

	CASE 1	CASE 2
1	0.9661451	0.2932169
2	-0.3604457	0.6623058
3	-0.0887532	-0.0225372

```
 4  0.1228009     0.0036634
 5 -0.2260956     0.0301290
 6 -0.1176664    -0.0194774

 7  0.0547528     0.0054766
 8 -0.1107294    -0.0082956
 9 -0.0782859    -0.0114408

10  0.0287693     0.0039528
11 -0.0575409    -0.0074988
12 -0.0435586    -0.0065170

13  0.0140763     0.0021257
14 -0.0280615    -0.0041540
15 -0.0213801    -0.0033279

16  0.0041469     0.0006469
17 -0.0082601    -0.0012729
18 -0.0063017    -0.0010010

INDICES OF COLUMNS TO BE RETAINED

 6      7      8     16     17     18     26
27     28     36     37     38     46     47
48

INDICES OF ROWS TO BE RETAINED

 1      3      5      6      8     10     11
13     15     16     18     20     21     23
25
NUMBER OF CONTROL POINTS =   15
NUMBER OF PRESSURE MODES =   15


THE DETERMINANT OF(A-TRANSPOSE)*A =  0.43119084E 12
```

SOLUTION FOR ANTI-SYMMETRIC CASES

CASE 1

1 0.4065139
2 -0.3341637
3 -0.1126727

4 0.0720457
5 -0.1435348
6 -0.0877573

7 0.0336897
8 -0.0677919
9 -0.0501169

10 0.0170100
11 -0.0339469
12 -0.0258488

13 0.0071492
14 -0.0142447
15 -0.0108606

SOLUTION FOUND AND STORED

FINISHED

APPENDIX II

COMPUTER PROGRAM LISTING

```
      INTEGER UCI,U6,UCO,U8,U20,U11,U12,UNSYM,PP,P,CWTYPE,SWTYPE,
     1PDEL,HOLD,PPNEW,ODISK
      LOGICAL BCS,BCAS,INVERS,READA,BATCH,STORE,RHS,COMSUM,
     1POINTS,CONTIN,CONV,PSOL,PIND,SKIP,CHECK,OPEN6,SYM,ASYM,
     2BCS2,BCAS2
      REAL MACH
      REAL HALS,NEWS,KNES,NSTS,NSUS,NPCS,NRHS,NPSS,NPIS
C
      DIMENSION PDEL(17),MDEL(17),TITLE(26),INDROW(100),
     1INDCOL(100),INDDEL(100),CARDIM(4,100),ETACP(47),
     2NINDEX(47),CHICP(20),ALFA(80,10),BCS(10),BCAS(10),
     3XETACP(47),A(1000),AR(120),
     4IL(120),ATA(80,90),XCHICP(20),RHSRES(20),X(80,10),
     5AINV(1,100),LIST3(2,25),LIST1(20),LIST2(47),BCS2(10),BCAS2(10)
C
C.....TSSCOM IS ONLY USED TO STORE AMES TSS COMMANDS.
C
      DIMENSION TSSCOM(20)
C
      EQUIVALENCE (AINV(1,1),XETACP(1)),(MDEL(1),PDEL(1)),
     1(XCHICP(1),A(1))
C
C.....NX SHOULD BE THE FIRST DIMENSION OF (X) AND SHOULD BE
C.....AS LARGE AS THE NUMBER OF MODES TOTAL.  THE SECOND DIM-
C.....ENSION OF (X) SHOULD BE AS LARGE AS THE NUMBER OF CASES
C.....(BOTH SYMMETRIC AND ANTISYMMETRIC).
C.....NB SHOULD BE THE FIRST DIMENSION OF (ALFA) AND SHOULD BE
C.....AS LARGE AS THE NUMBER OF CONTROL POINTS BEFORE ANY
C.....ARE DELETED.  THE SECOND DIMENSION OF (ALFA) SHOULD BE AS
C.....LARGE AS THE NUMBER OF CASES BEING TREATED.
      DATA UCI/5/,UCO/6/,U8/8/,U20/09/,U11/11/,U12/12/,BLANK /1H /,
     1CASE/4HCASE/,IFMT/0/,NX/80/,NB/80/,NAINV/1/,
     2INVERS/.FALSE./,READA/.TRUE./,NCOMK/100/
      DATA HALS / 3HHAL  /
      DATA NEWS / 3HNEW  /
      DATA KNES / 3HKNE  /
      DATA DCMS / 3HDCM  /
      DATA DSMS / 3HDSM  /
      DATA DCPS / 3HDCP  /
```

```
      DATA DSP$ / 3HDSP  /
      DATA DPP$ / 3HDPP  /
      DATA CON$ / 3HCON  /
      DATA STA$ / 3HSTA  /
      DATA NST$ / 3HNST  /
      DATA STO$ / 3HSTO  /
      DATA SUM$ / 3HSUM  /
      DATA NSU$ / 3HNSU  /
      DATA PCP$ / 3HPCP  /
      DATA NPC$ / 3HNPC  /
      DATA RHS$ / 3HRHS  /
      DATA NRH$ / 3HNRH  /
      DATA PSO$ / 3HPSO  /
      DATA NPS$ / 3HNPS  /
      DATA PIN$ / 3HPIN  /
      DATA NPI$ / 3HNPI  /
      DATA SET$ / 3HSET  /
      DATA TSS$ / 3HTSS  /
      DATA RPM$ / 3HRPM  /
      DATA RCP$ / 3HRCP  /
C
      OPEN6 = .FALSE.
      COM = BLANK
      WRITE(UCO,801)
C
C     ESTABLISH WHETHER THIS IS A BATCH OR CONVERSATIONAL JOB
      READ(UCI,799) BATCH
      CONV = .NOT.BATCH
C
   17 CONTINUE
      IND = 0
C     IND = 1 WHEN BOTH SYMMETRIC AND ANTISYMMETRIC SOLUTIONS
C     ARE TO BE OBTAINED
C         = 0 WHEN ONLY ONE TYPE OF SOLUTION  IS TO BE
C           OBTAINED.
      LIST1(1) = 0
      LIST2(1) = 0
      ASYM = .FALSE.
      IF(OPEN6) ENDFILE 06
```

```
      IF(OPEN6) CALL OBEY(16,16HRELEASE FT04F001 )
      U6 = UCO
      IF(CONV) WRITE(UCO,815)
      READ(UCI,810)ODISK
      IF(ODISK.NE.0) U6 = 4
      IF(U6.NE.UCO) OPEN6=.TRUE.
      IF(.NOT.OPEN6) GO TO 18
C
C.....FOR AMES TSS SYSTEM ONLY.  OBEY SENDS COMMANDS TO TSS.
C.....CVRT IS A LIBRARY PROGRAM FOR IN-CORE READ,REWIND,WRITE.
      ODISK = MOD(MAX0(1,ODISK),10)
      WRITE( UCO,931) ODISK
      CALL OBEY(16,16HRELEASE FT04F001    )
      CALL CVRT(ODISK,1,
     1 44H(!DDEF FT04F001,,OUTPUT,SOLVR,N!,I1,6X)          ,
     2X,8,8H(8A4)       )
      CALL OBEY(32,X)
      REWIND 4
   18 CONTINUE
C
C     ENTER ID NUMBERS
      IF(CONV) WRITE(UCO,820)
      READ(UCI,810) IDD1,IDD2,IDD3
      IF(IDD1.NE.0) CHECK = .TRUE.
C
C     CHECK IF FALSE SUPPRESSES CHECKING OF AIM AND
C     BC FILE IDENTIFICATION NUMBERS
C
      SKIP = .FALSE.
C     SKIP WHEN .TRUE. SUPPRESSES READING OF BOUNDARY CONDITION
C     FILE--IT IS SET TO TRUE WHEN A CASE IS COMPLETED AND
C     MORE SOLUTIONS ARE TO BE OBTAINED BY DELETING ROWS AND
C     COLUMNS
C
C     OPEN AND READ AIM FILE
C     ON STANDARD COMPUTER REMOVE FOLLOWING CALL AND OPEN
C     STATEMENTS AND ATTACH STATEMENT NUMBER 20 TO READ
C     STATEMENT
C
```

```
C
C.....FOR AMES' TSS VERSION ONLY.   AIMFIL ISSUES DDEF TO OPEN
C.....AIM FILE.
C
      CALL AIMFIL(IDD1,IDD2)
      READ(U11) ID1,ID2,NSEQ,TITLE,NTITL,PP,CWTYPE,MM,MREF,
     1SWTYPE,NMAX,KK,MODES,UNSYM,LMIN,LMAX,JJMAX,JJ,MACH,EPS,
     2DELTA0,(CHICP(I),I=1,PP),(NINDEX(I),I=1,MM),
     3(ETACP(M),M=1,MM)
C
C     PP      = NUMBER OF CHORDWISE CONTROL POINTS
C     MM      = NUMBER OF SPANWISE CONTROL POINTS ON ENTIRE WING
C     MODES   = TYPES OF MODES AVAILABLE
C                        0 = SYMMETRIC ONLY
C              LESS THAN  0 = ANTI-SYMMETRIC ONLY
C              MORE THAN  0 = ALL MODES AVAILABLE
C     UNSYM   = SYMMETRY INDICATOR
C                     NOT 0 = UNSYMMETRIC WING
C                        0 = SYMMETRIC WING
C     CHICP   = LOCATIONS OF CHORDWISE CONTROL POINTS
C     ETACP   = LOCATION OF SPANWISE CONTROL POINTS
C
C     ASYM    = .TRUE. FOR UNSYMMETRIC WING
C             = .FALSE. FOR SYMMETRIC WING
C     SYM     = .TRUE. FOR SYMMETRIC WING
C             = .FALSE. FOR UNSYMMETRIC WING
C
      IF(UNSYM.NE.0) ASYM = .TRUE.
      SYM = .NOT.ASYM
C     UNLESS CHECK = .FALSE., CHECK TO BE SURE ID ON FILES
C     MATCH THOSE READ IN
C
C.....THIS CHECKING IS NOT NECESSARY FOR THE AMES' VERSION, BUT IS
C.....DONE ANYWAY.
      IF(.NOT.CHECK) GO TO 43
      IF(ID1.EQ.IDD1) GO TO 35
      IF (BATCH) CALL STOP2(UCO,'   IDD1 DOES NOT MATCH AIM FILE',
     1FLOAT(ID1))
      PAUSE ' IDD1 DOES NOT MATCH AIM FILE VALUE'
```

```
   35 IF(ID2.EQ.IDD2) GO TO 43
      IF (BATCH)CALL
     1STOP2(UCO,'  IDD2 READ IN DOES NOT MATCH AIM VALUE ',
     2FLOAT(ID2))
      PAUSE ' IDD2 DOES NOT MATCH AIM FILE VALUE '
C
C     OPEN BOUNDARY CONDITION FILE
43    CONTINUE
C
C.....BCFIL ISSUES DDEF CONTROL CARD FOR AMES' TSS VERSION.
C
      CALL BCFIL(ID1,IDD3)
      READ(UB) ID1,ID3,TITLE,UNSYM,NSYM,NASYM,(BCS(I),
     1I=1,10),(BCAS(I),I=1,10),PP,CNTYPE,MM,MMP,MMPA,MREF,
     2SWTYPE,(XCHICP(I),I=1,PP),(NINDEX(I),I=1,MMP),
     3(XETACP(I),I=1,MMP)
C
C     NSYM   = NUMBER OF SYMMETRIC CASES
C     NASYM  = NUMBER OF ANTI-SYMMETRIC CASES
C     MMP    = NUMBER OF SPANWSIE CONTROL POINTS USED FOR
C              THE SYMMETRIC CASES
C     MMPA   = NUMBER OF SPANWISE CONTROL POINTS USED FOR
C              THE ANTI-SYMMETRIC CASES
C     XCHICP = LOCATIONS OF CHORDWISE CONTROL POINTS
C     XETACP = LOCATIONS OF SPANWISE CONTOROL POINTS
C
      IF(.NOT.CHECK) GO TO 60
      IF(ID1.EQ.IDD1) GO TO 48
      IF(BATCH) CALL STOP2(UCO,'  BC FILE AND AIM FILE DO NOT
     1HAVE MATCHING ID  ',FLOAT(ID1))
      PAUSE ' BC FILE AND AIM FILE DO NOT HAVE MATCHING ID '
   48 IF(IDD3.EQ.ID3) GO TO 60
      IF(BATCH)CALL STOP2(UCO,
     1'  IDD3 DOES NOT MATCH BC FILE VALUE ',FLOAT(ID3))
      PAUSE 'IDD3 DOES NOT MATCH BC FILE VALUE'
C
C     CHECK THAT CHORDWISE AND SPANWISE LOCATIONS MATCH ON
C     TWO FILES
   60 DO 61 I =1,PP
```

```
      IF(CHICP(I).EQ.XCHICP(I)) GO TO 61
      IF(BATCH) CALL STOP2
     1(U6,'  CHORDWISE CONTROL POINTS ON FILES DO NOT MATCH  ',
     2FLOAT(I))
      PAUSE 'CHORDWISE CONTROL POINTS DO NOT MATCH'
   61 CONTINUE
      DO 62 I = 1,MMP
      IF(ETACP(I).EQ.XETACP(I)) GO TO 62
      IF(BATCH) CALL STOP2(U6, 'SPANWISE CONTROL POINTS DO NOT
     1 MATCH ON 2 FILES AT INDEX  ',FLOAT(I))
      PAUSE 'SPANWISE CONTROL POINTS DO NOT MATCH'
   62 CONTINUE
C
C     INITIALIZE VARIABLES
C     KJUMP = 2 FOR SYMMETRIC WING
C           = 1 FOR UNSYMMETRIC WING
      IF(SYM) KJUMP = 2
      IF(ASYM) KJUMP = 1
      KK2 = KK
      NMAX2 = NMAX
      NROWS = PP
      L1 = 0
      L2 = 0
      L3 = 0
      PPNEW = PP
      NTYPE = 0
      MMNEW = MM
      MIDDLE = MOD(MM,2)*MMP
      IF(SYM) GO TO 65
      IF(MODES.GT.0) GO TO 70
      IF(BATCH) CALL STOP2(U6,
     1    '  UNSYM WING BUT NOT ALL MODES ARE AVAILABLE  ',
     2FLOAT(MODES))
      PAUSE 'UNSYM WING BUT NOT ALL MODES ARE AVAILABLE  '
   65 IF(CONV) WRITE(UCO,840)
C
C     READ IN VALUE FOR NTYPE
C         0 = SYMMETRIC CASES
C        -1 = ANTISYMMETRIC CASES
```

```
C          1 = BOTH SYMMETRIC AND ANTISYMMETRIC CASES
C      FOR AN UNSYMMETRIC WING NTYPE = 0
       READ(UCI,810) NTYPE
       IF(MODES.GT.0) GO TO 85
       IF(MODES.EQ.0.AND.NTYPE.EQ.0) GO TO 75
       IF(MODES.LT.0.AND.NTYPE.LT.0) GO TO 80
       IF(MODES.EQ.0.AND.NTYPE.GT.0) GO TO 68
       IF(MODES.LT.0.AND.NTYPE.GT.0) GO TO 69
       IF(BATCH.AND.NTYPE.LT.0) CALL STOP2(U6,
      1' ANTI-SYMMETRIC MODES ARE NOT AVAILABLE ',
      2FLOAT(MODES))
       IF(BATCH.AND.NTYPE.GE.0) CALL
      1STOP2(U6,'  SYMMETRIC MODES ARE NOT AVAILABLE ',
      2FLOAT(MODES))
       IF(NTYPE.LT.0)
      1PAUSE 'ANTI-SYMMETRIC MODES ARE NOT AVAILABLE'
       IF(NTYPE.GE.0) PAUSE
      1'SYMMETRIC MODES ARE NOT AVAILABLE'
       GO TO 65
   68  NTYPE = 0
       WRITE(UCO,881)
       GO TO 75
   69  NTYPE = -1
       WRITE(UCO,882)
       GO TO 80
C
C      INDDEL IS AN ARRAY CONTAINING INDICES OF COLUMNS OR
C      ROWS TO BE DELETED
C      INDROW IS AN ARRAY INITIALLY FILLED WITH INDICES OF ALL
C      ROWS FOR THE PARTICULAR TYPE OF WING.
C      INDCOL IS AN ARRAY INITIALLY FILLED WITH INDICES
C      OF ALL COLUMNS FOR THE PARTICULAR CASE.
C      AFTER EACH DELETION A CALL IS MADE TO SUBROUTINE KRUNCH
C      AFTER WHICH INDROW CONTAINS INDICES OF ROWS TO BE
C      RETAINED OR INDCOL CONTAINS INDICES OF COLUMNS TO
C      BE RETAINED.
C      NRECD  = NUMBER OF LOGICAL RECORDS
C      NCOLSB = NUMBER OF COLUMNS BEFORE DELETION OF
C               UNWANTED COLUMNS
```

```
C      NCOLSA = NUMBER OF COLUMNS AFTER DELETION  OF
C               UNWANTED COLUMNS.
C      NROWSB = NUMBER OF ROWS BEFORE DELETION OF UNWANTED
C               ROWS
C      NROWSA = NUMBER OF ROWSA AFTER DELETION OF UNWANTED
C               ROWS.
C
C      UNSYMMETRIC WING
   70  NXR = PP * MM
       NCOLSB=NMAX * KK
       NRECDS = MM
       GO TO 120
C
C      SYMMETRIC BOUNDARY CONDITIONS ONLY
   75  NXR = PP * MMP
       NCOLSB = NMAX *((KK+1)/2 )
       NRECDS = MMP
       NXC = NCOLSB
       GO TO 121
C
C
C      SYMMETRIC WING WITH ANTI-SYMMETRIC B, C, ONLY
   80  NXR = PP * MMPA
       NRECDS = MMPA
       NXC = NMAX * (KK/2)
       NCOLSB = NXC
       GO TO 121
C
C      SYMMETRIC CASES FOR SYMMETRIC WING WITH BOTH
C      SYMMETRIC AND ANTI-SYMMETRIC BOUNDARY CONDITIONS
   85  IF(NTYPE.LT.0) GO TO 95
       NXC = NMAX *((KK+1)/2)
       NCOLSB = NMAX * KK
       NXR = PP * MMP
       NRECDS = MMP
       IX = 0
       DO 90 I = 1,KK,2
       L = (I-1) * NMAX + 1
       K = L + NMAX - 1
```

```
      DO 90 J = L,K
      IX = IX + 1
   90 INDCOL(IX) = J
      GO TO 123
C
C     ANTI-SYMMETRIC CASES FOR A SYMMETRIC WING WITH BOTH
C     SYMMETRIC AND ANTI-SYMMETRIC BOUNDARY CONDITIONS
   95 NXC = NMAX *( KK /2)
      NROWSB = PP * MMP
C
C     SPACE PAST THE SYMMETRIC BOUNDARY CONDITIONS AS THEY
C     ARE NOT NEEDED
C
      DO 96 I = 1,NSYM
   96 READ(U8)
      NCOLSB = NMAX * KK
      IX = 0
      DO 97 I = 2,KK,2
      L = (I-1) * NMAX + 1
      K = L + NMAX - 1
      DO 97 J = L,K
      IX = IX + 1
   97 INDCOL(IX) = J
      NXR = MMPA * PP
      NRECDS = MMPA
      GO TO 123
C
  120 NXC = NMAX * KK
  121 DO 122 I = 1,NXC
  122 INDCOL(I) = I
  123 DO 124 I = 1,NXR
  124 INDROW(I) = I
      NROWSB = NXR
      N2 = NXC
      N3 = NXR
      NCOLSA = NXC
      NROWSA = NXR
C
C.....NSYM2 AND NASYM2 ARE THE NUMBERS OF SYMMETRIC AND ANTI-
```

```
C.....SYMMETRIC SOLUTIONS TO BE OBTAINED AND BCS2
C.....AND BCAS2 DENOTED THE TYPES OF CASES.
C
      NSYM2=NSYM
      NASYM2=NASYM
      DO 130 I=1,10
      BCS2(I)=BCS(I)
130   BCAS2(I)=BCAS(I)
      IF (UNSYM.NE.0 .OR. NTYPE.GT.0) GO TO 150
      IF(NTYPE.LT.0) GO TO 140
      NASYM2=0
      DO 132 I=1,10
      BCAS2(I)=.FALSE.
132   CONTINUE
      GO TO 150
140   NSYM2=0
      DO 142 I=1,10
      BCS2(I)=.FALSE.
142   CONTINUE
150   CONTINUE
C
      NCOM = 0
      IF(COM.EQ. KNES) GO TO 200
C
C     INITIALIZE COMMANDS
153   STORE = .TRUE.
      COMSUM = .FALSE.
      POINTS = .FALSE.
      RHS = .FALSE.
      PSOL = .FALSE.
      CONTIN = CONV
      PIND = .FALSE.
200   CONTINUE
      NCOM = NCOM + 1
      IF(NCOM.GT.NCOMX) CALL STOP2(U6,
     1'  TOO MANY COMMANDS  ',FLOAT(NCOM))
C
      IF (CONV) WRITE(UCO,845)
      READ(UCI,805) (CARDIM(I,NCOM),I=1,4)
```

```
      COM = CARDIM(1,NCOM)
  210 CONTINUE
      IF (COM .EQ. HAL$) GO TO  225
      IF (COM .EQ. NEW$) GO TO   17
      IF (COM .EQ. KNE$) GO TO   17
      IF (COM .EQ. DCM$) GO TO  340
      IF (COM .EQ. DSM$) GO TO  360
      IF (COM .EQ. DCP$) GO TO  370
      IF (COM .EQ. DSP$) GO TO  390
      IF (COM .EQ. DPP$) GO TO  398
      IF (COM .EQ. CON$) GO TO  235
      IF (COM .EQ. STA$) GO TO  400
      IF (COM .EQ. NST$) GO TO  211
      IF (COM .EQ. STO$) GO TO  212
      IF (COM .EQ. SUM$) GO TO  213
      IF (COM .EQ. NSU$) GO TO  214
      IF (COM .EQ. PCP$) GO TO  215
      IF (COM .EQ. NPC$) GO TO  216
      IF (COM .EQ. RHS$) GO TO  217
      IF (COM .EQ. NRH$) GO TO  218
      IF (COM .EQ. PSO$) GO TO  219
      IF (COM .EQ. NPS$) GO TO  220
      IF (COM .EQ. PIN$) GO TO  221
      IF (COM .EQ. NPI$) GO TO  222
      IF (COM .EQ. SET$) GO TO  153
      IF (COM .EQ. TSS$) GO TO  223
      IF (COM .EQ. RPM$) GO TO  226
      IF (COM .EQ. RCP$) GO TO  232
      WRITE(UCO,850)(CARDIM(I,NCOM),I=1,4)
      IF(.NOT.CONTIN) STOP
      NCOM = NCOM - 1
      GO TO 200
C
  211 STORE = .FALSE.
      ID4=0
      GO TO 200
  212 STORE = .TRUE.
      GO TO 200
  213 COMSUM = .TRUE.
```

```
      GO TO 200
  214 COMSUM = .FALSE.
      GO TO 200
  215 POINTS = .TRUE.
      GO TO 200
  216 POINTS = .FALSE.
      GO TO 200
  217 RHS = .TRUE.
      GO TO 200
  218 RHS = .FALSE.
      GO TO 200
  219 PSOL = .TRUE.
      GO TO 200
  220 PSOL = .FALSE.
      GO TO 200
  221 PIND = .TRUE.
      GO TO 200
  222 PIND = .FALSE.
      GO TO 200
99
C
  223 CONTINUE
C
C     TSS COMMAND IS FOR THE AMES VERSION ONLY.  IT ALLOWS COMMANDS
C     TO BE GIVEN TO THE OPERATING SYSTEM FROM THE PROGRAM.
C
      READ(UCI,806) TSSCOM
      CALL OBEY(80,TSSCOM)
      GO TO 200
C
C
C     HALT
  225 IF(OPEN6) ENDFILE U6
      IF(OPEN6) CALL OBEY(16,16HRELEASE FT04F001 )
      WRITE(UCO,855)
      GO TO 999
C
  226 CONTINUE
C
C     RPM---RESTORE PRESSURE MODES
```

```
C
      KK2 = KK
      NMAX2 = NMAX
      IF(MODES.GT.0 .AND. SYM) GO TO 228
      DO 227 I=1,NCOLSB
227   INDCOL(I)=I
      NCOLSA=NCOLSB
      N2=NCOLSB
      GO TO 200
228   CONTINUE
C
C.....MORE MODES ARE AVAILABLE THAN REQUIRED, I.E., SYMMETRIC WING WITH BOTH
C.....SYMMETRIC AND ANTISYMMETRIC MODES AVAILABLE.
C
      KL=1
      IF (NTYPE.LT.0) KL=2
      NCOLSA = 0
      DO 229 K=KL,KK,2
      I=PP*(K-1)
      DO 229 P=1,PP
      NCOLSA=NCOLSA+1
      INDCOL(NCOLSA) = P+I
229   CONTINUE
      N2=NCOLSA
      GO TO 200
C
232   CONTINUE
C
C     RCP---RESTORE ALL CONTROL POINTS
C
      L1 =0
      L2=0
      L3=0
      LIST1(1) = 0
      LIST2(1) = 0
      PPNEW=PP
      MMNEW=MM
      NROWSA=NROWSB
C
```

```
C.....FOR ANTISYMMETRIC CASES ONLY THE CONTROL POINTS AT THE CENTERLINE
C.....HAVE TO BE DELETED.  CONTROL POINTS WILL BE AT THE CENTERLINE
C.....IF MM IS ODD  (MOD(MM,2)=1).
C
      IF(MODES.LT.0) NROWSA=NROWSA-MOD(MM,2)*PP
      N3 = NROWSA
      DO 233 I=1,NROWSA
233   INDROW(I)=I
      GO TO 200
C
C     CONTINUE
235   CONTIN = .TRUE.
      GO TO 200
C
C     DCM-----DELETE CHORDWISE MODES
C     NMAX2 IS THE NEW NUMBER OF CHORDWISE MODES
340   IF(CONV) WRITE(UCO,865) NMAX2,NMAX
      READ(UCI,810) NMAX2
      NMAX2 = MIN0(NMAX,NMAX2)
      IF(NMAX2.EQ.NMAX) GO TO 200
      IF(NMAX2.GT.0) GO TO 344
      IF(BATCH) CALL STOP2(U6,'  INCORRECT VALUE FOR NMAX2  ',
     1FLOAT(NMAX2))
      WRITE(UCO,867)
      NCOM = NCOM - 1
      GO TO 200
344   J = 0
      IADD = 0
      IMAX = NMAX2 + 1
346   DO 348 I = IMAX,NMAX
      J = J+ 1
      L = I + IADD
      INDDEL(J) = L
      IF(INDDEL(J).EQ.NCOLSH) GO TO 350
348   CONTINUE
      IADD = IADD + NMAX
      GO TO 346
350   CALL KRUNCH(J,INDDEL,N2,INDCOL,NCOLSA)
      N2 = NCOLSA
```

```
      GO TO 200
C
C     DSM---DELETE SPANWISE MODES
C     KK2 IS THE NEW NUMBER OF SPANWISE MODES,  ALL
C     COLUMNS TO THE RIGHT OF KK2 * NMAX WILL BE DELETED.
  360 IF(CONV) WRITE(UCO,870) KK2,KK
      READ(UCI,810) KK2
      KK2 = MINO(KK,KK2)
      IF(KK2.EQ.KK) GO TO 200
      IF(KK2.GT.0) GO TO 364
      IF(CONV) GO TO 362
      CALL STOP2(U6,'  INCORRECT VALUE FOR KK2    ',FLOAT(KK2))
  362 WRITE(UCO,874)
      NCOM = NCOM + 1
      GO TO 200
  364 IF(MODES.GT.0) LX = KK2 * NMAX + 1
      IF(MODES.EQ.0) LX = ((KK2+1)/2)*NMAX+1
      IF(MODES.LT.0) LX = (KK2/2)*NMAX+1
      J = 0
      DO 367 I = LX,NCOLSB
      J = J + 1
      INDDEL(J) = I
  367 CONTINUE
      GO TO 350
C
C     DCP---DELETE CHORDWISE POINTS
C     PDEL IS AN INTEGER ARRAY OF NOT MORE THAN 17
C     CHORDWISE POINTS TO BE DELETED.  MUST BE
C     IN ASCENDING ORDER.
  370 IF(CONV) WRITE(UCO,875)
      READ(UCI,876) (PDEL(I),I=1,17)
      IF(PDEL(1) .EQ.0) GO TO 200
      DO 373 I = 1,17
      IF(PDEL(I).EQ.0) GO TO 372
      L1 = L1 + 1
      LIST1(L1) = PDEL(I)
C     LIST1 IS AN ARRAY CONTAINING THE INDICES OF DELETED
C     CHORDWISE POINTS.
      GO TO 373
```

```
  372 LAST = I - 1
      GO TO 374
  373 CONTINUE
      LAST=17
  374 PPNEW = PPNEW - LAST
      N1 = 0
      DO 376 I = 1,NRECDS
      DO 376 K = 1, LAST
      N1 = N1 + 1
      IPOINT = PP * (I-1) + PDEL(K)
  376 INDDEL(N1)= IPOINT
  384 CALL KRUNCH(N1,INDDEL,N3,INDROW,NROWSA)
      N3 = NROWSA
      GO TO 200
C
C     DSP---DELETE SPANWISE POINTS
C     MDEL IS AN INTEGER ARRAY OF NOT MORE THAN 17
C     SPANWISE POINTS TO BE DELETED IN ASCENDING ORDER.
  390 IF(CONV) WRITE(UCO,877)
C     TO DELETE THESE POINTS INCREMENTS OF PP ARE
C     ADDED TO REACH POINT (NRECD-1) TIMES AND THE
C     RESULTS STORED IN THE INDDEL ARRAY.  SUBROUTINE
C     KRUNCH IS THEN CALLED TO DELETE THESE POINTS.
      READ(UC1,876) (MDEL(I),I=1,17)
      IF(MDEL(1) .EQ.0) GO TO 200
      N1 = 0
      DO 394 I = 1,17
      IX = MDEL(I)
      IF(IX.EQ.0) GO TO 384
      IF(SYM.AND.IX.GT.MIDDLE)GO TO 394
      MMNEW = MMNEW - 1
      IF(IX.NE.MIDDLE.AND.SYM) MMNEW = MMNEW - 1
      L2 = L2 + 1
      LIST2(L2) = IX
      LX = (IX-1) * PP + 1
      JX = IX * PP
      DO 392 J = LX,JX
      N1 = N1 + 1
  392 INDDEL(N1) = J
```

```
  394 CONTINUE
      GO TO 384
C
C     DPP---DELETE PARTICULAR POINT
C     READ INDICES OF POINTS TO BE DELETED
  398 IF(CONV) WRITE(UCO,878)
      READ(UCI,810) P,M
      L3 = L3 + 1
      LIST3(1,L3) = P
      LIST3(2,L3) = M
      INDDEL(1) = (M-1) * PP + P
      N1 = 1
      GO TO 384
C
C     START
  400 IF(NROWSA.GE.NCOLSA) GO TO 420
      IF(BATCH) CALL STOP2(U6,'  MORE UNKNOWNS THAN EQUATIONS',
     1FLOAT(NROWSA))
      PAUSE 'MORE UNKNOWNS THAN EQUATIONS'
      GO TO 200
C
  420 IF(.NOT.STORE) GO TO 430
C
      CALL OBEY (22,24HDDEF FT09F001,,IDFILE     )
C
      REWIND U20
      READ(U20) IDA,IDB,IDC,ID4
      ID4 = ID4 + 1
      REWIND U20
      WRITE(U20)IDA,IDB,IDC,ID4
      ENDFILE U20
      CALL OBEY( 16,16HRELEASE FT09F001)
      IF (ODISK.NE.0) WRITE(UCO,932) ID4
C
C     WRITE HEADING INFORMATION
  430 WRITE(U6,886) (TITLE(I),I=1,NTITL),ID1,ID2,ID3,ID4,
     1JJ,UNSYM,MODES,NTYPE,PPNEW,MMNEW,NMAX2,KK2,
     2DELTA0,MACH,(BCS(I),I=1,10),(BCAS(I),I=1,10)
      WRITE(U6,812)
```

```
C
C       WRITE SUMMARY OF COMMANDS
        IF(.NOT.COMSUM) GO TO 440
        WRITE(U6,887)
        DO 435 J = 1,NCOM
        WRITE(U6,807) (CARDIM(I,J),I=1,4)
  435 CONTINUE
C
C       SORT DELETED INDICES
  440 IF(.NOT.POINTS) GO TO 477
        IF(L1.LE.1) GO TO 444
        IX = L1
        K = L1 - 1
        DO 442 J = 1,K
        IX = IX - 1
        DO 442 L = 1,IX
        I = L + 1
        IF(LIST1(L).LE.LIST1(I)) GO TO 442
        HOLD = LIST1(L)
        LIST1(L) = LIST1(I)
        LIST1(I) = HOLD
  442 CONTINUE
  444 IF (L2.LE.1) GO TO 470
        IX = L2
        K = L2 - 1
        DO 446 J = 1,K
        IX = IX - 1
        DO 446 L = 1,IX
        I = L + 1
        IF(LIST2(L).LE.LIST2(I)) GO TO 446
        HOLD = LIST2(L)
        LIST2(L) = LIST2(I)
        LIST2(I) = HOLD
  446 CONTINUE
C
C       WRITE OUT CHORDWISE CONTROL POINTS
  470 WRITE(U6,910)
        IX = 1
        DO 473 I = 1,PP
```

```
      IF(LIST1(IX),NE,I) GO TO 472
      IX = IX + 1
      GO TO 473
  472 WRITE(U6,906) CHICP(I)
  473 CONTINUE
C     WRITE OUT SPANWISE CONTROL POINTS
      WRITE(U6,912)
      IX = 1
      DO 475 I = 1,NRECDS
      IF(LIST2(IX),NE,I) GO TO 474
      IX = IX + 1
      GO TO 475
  474 WRITE(U6,906) ETACP(I)
  475 CONTINUE
      IF(L3,EQ,0) GO TO 477
C
C     WRITE OUT DELETED CONTROL POINTS
      WRITE(U6,914)
      DO 476 I = 1,L3
      J = LIST3(1,I)
      K = LIST3(2,I)
  476 WRITE(U6,915) CHICP(J),ETACP(K)
C
C     READ BOUNDARY CONDITION FILE
  477 IZERO = 0
      IF(SKIP) GO TO 486
C
C     IEND = NUMBER OF RIGHT HAND SIDES
C
      IF(NTYPE,GE,0) IEND = NSYM
      IF(NTYPE,LT,0) IEND = NASYM
      IF(ASYM) IEND = NSYM + NASYM
      ISTOP = IEND
      IF(NTYPE,GT,0) ISTOP = NSYM + NASYM
C
C     BOTH SETS OF BOUNDARY CONDITIONS ARE READ IN AT THE
C     SAME TIME IF BOTH TYPES OF SOLUTIONS ARE REQUIRED.
C     THIS ELIMINATES THE NEED TO REWIND THE BOUNDARY CONDITION
C     FILE FOR SUBSEQUENT CASES OF THE SAME CONFIGURATION.
```

```
C
      IX = NROWSB
      DO 479 I = 1,ISTOP
      IF(NTYPE.NE.1) GO TO 479
      IF(I.GT.NSYM) IX = MMPA + PP
  479 READ(U8)(ALFA(K,I),K=1,IX)
  480 IF(.NOT.RHS) GO TO 486
C
C     WRITE BOUNDARY CONDITONS
      WRITE(U6,898)
      DO 484 I1 = 1,IEND
      I = IZERO + I1
      WRITE(U6,897) I
      J = 1
      K = PP
      DO 483 M = 1,NRECDS
      WRITE(U6,899) (ALFA(L,I),L=J,K)
      J = J + PP
  483 K= K + PP
      WRITE(U6,812)
  484 CONTINUE
  486 IF(.NOT.PIND) GO TO 491
C
C     WRITE OUT INDCOL, INDROW
      WRITE(U6,890)
      WRITE(U6,892) (INDCOL(J),J=1,NCOLSA)
      WRITE(U6,894)
      WRITE(U6,892)(INDROW(J),J=1,NROWSA)
C
C     CALL EQUATION SOLVING SUBROUTINE
  491 NKTP = NCOLSA + IEND
      WRITE (U6,933) NROWSA,NCOLSA
      CALL SOLVIT(NCOLSB,NCOLSA,NROWSB,NROWSA,IEND,READA,
     1IFMT,U11,INDCOL,INDROW,NRECDS,INVERS,A,ALFA,IZERO,NB,NX,
     2NAINV,ATA,AR,IL,X,AINV,RMSRES,DET,IERR,NKTP,NROWS)
      IF(IERR.NE.0) CALL STOP2(U6,
     1' ERROR IN EQUATION SOLVER, IERR IS ',
     2FLOAT(IERR))
C
```

```
      WRITE(U6,895) DET
C
C     SOLVIT DOES NOT CALCULATE RESIDUALS (YET).
C     WRITE(UCO,903)
C     DO 495 I = 1,IEND
C     WRITE(UCO,930) I, RMSRES(I)
C 495 CONTINUE
C
      IF(.NOT.PSOL) GO TO 512
C
C     PRINT SOLUTIONS FIVE CASES AT A TIME
      IF(IND.NE.1) GO TO 503
      WRITE(U6,902)
      GO TO 504
  503 IF(ASYM)WRITE(U6,900)
      IF(SYM.AND.NTYPE.GE.0)WRITE(U6,904)
      IF(SYM.AND.NTYPE.LT.0)WRITE(U6,902)
C
  504 CONTINUE
      DO 511 JJJ= 1,IEND,5
      J4 = MIND(JJJ+4,IEND)
      WRITE(U6,901)(CASE,JK,JK=JJJ,J4)
      ICOUNT = 0
      DO 505 II = 1,NCOLSA
      IF(ICOUNT.EQ.0) WRITE(U6,812)
      ICOUNT = ICOUNT + 1
      IF(ICOUNT.EQ.NMAX2) ICOUNT = 0
  505 WRITE(U6,907)(II,(X(II,JK),JK=JJJ,J4))
  511 CONTINUE
  512 IF(.NOT.STORE) GO TO 518
C
C     STORE RESULTS
C     IF BOTH SYMMETRIC AND ANTIYSMMETRIC SOLUTIONS ARE TO
C     STORED BYPASS OPENING FILE AND WRITING TITLE INFORMATION.
      IF(IND.EQ.1) GO TO 513
C
C.....BNKFIL IS FOR AMES TSS VERSION.  BNKFIL ISSUES CONTROL CARDS TO THE TSS
C.....OPERATING SYSTEM TO DEFINE (DDEF) THE SOLUTION FILE.
C
```

```
      CALL BNKFIL(ID1,ID2,ID3,ID4)
      WRITE(U12)ID1,ID2,ID3,ID4,TITLE,SYM,NSYM2,NASYM2,
     1(BCS2(I),I=1,10),(BCAS2(I),I=1,10),NMAX2,KK2,JJ,
     2PPNEW,MMNEW,NROWSA,DELTA0,EPS,MACH
  513 DO 514 I = 1,IEND
  514 WRITE(U12)(X(J,I),J=1,NCOLSA)
  518 IF(NTYPE.LE.0) GO TO 550
C
C     CASE WHERE THE WING IS SYMMETRIC AND SOLUTIONS ARE
C     TO BE OBTAINED FOR BOTH SYMMETRIC AND ANTI-SYMMETRIC
C     BOUNDARY CONDITIONS
      IF(IND.EQ.1) GO TO 540
C
C     ADD NMAX TO EACH ELEMENT OF (INDCOL) TO OBTAIN
C     ANTISYMMETRIC MODES
      DO 522 I = 1,NCOLSA
  522 INDCOL(I) = INDCOL(I) + NMAX
C
      IEND = NASYM
      IND = 1
      IZERO = NSYM
      IF(MOD(KK2,2).NE.0) NCOLSA = NCOLSA-NMAX2
C     NCOLSB REMAINS THE SAME
      IF(MMP.EQ.MMPA) GO TO 526
C
C     DELETING CONTROL POINTS ON CENTERLINE.
C
C     A DIRECT COUNT OF POINTS ON THE CENTERLINE IS
C     NECESSARY SINCE THE DPPOINT COMMAND MAY HAVE BEEN
C     USED.  NUMCL = NUMBER CURRENTLY ON THE CENTERLINE.
      NUMCL = 0
      JMIN = NROWSB-PP+1
      IMIN = MAX0(NROWSA-PP+1,1)
      DO 525 J = JMIN,NROWSB
      DO 523 I = IMIN,NROWSA
      IF(J.EQ.INDROW(I)) GO TO 524
  523 CONTINUE
      GO TO 525
  524 NUMCL = NUMCL + 1
```

```
  525 IMIN = MIN0(IMIN+1,NROWSA)
      NROWSA = NROWSA - NUMCL
      NROWSB = NROWSB - PP
      NRECDS = NRECDS - 1
  526 CONTINUE
C
C     BACKSPACE THE INFLUENCE MATRIX FILE TO THE BEGINNING
C     OF THE INFLUENCE MATRIX.
      REWIND U11
      READ (U11)
C
C     IF SKIP = .TRUE. BYPASS READING OF BOUNDARY
C                 CONDITION FILE.
      IF(SKIP) GO TO 486
      GO TO 480
  540 CONTINUE
C
C     NOW THAT THE ANTISYMMETRIC SOLUTIONS HAVE BEEN
C     OBTAINED AND (IF REQUIRED) STORED, THE
C     AFFECTED VARIABLES WILL BE RESET TO VALUES APPROPRIATE
C     FOR FURTHER SOLUTIONS TO THE SYMMETRIC CASES
      IEND = NSYM
      IND = 0
      IF(MOD(KK2,2).NE.0) NCOLSA = NCOLSA + NMAX2
      DO 542 I = 1,NCOLSA
  542 INDCOL(I) = INDCOL(I) - NMAX
      IF(MMP.EQ.MMPA) GO TO 546
      NROWSA = NROWSA + NUMCL
      NROWSB = NROWSB + PP
      NRECDS = NRECDS + 1
  546 CONTINUE
  550 CONTINUE
C
C
      IF(STORE) ENDFILE U12
      IF(STORE) WRITE(U6,885)
C
C     AT THIS POINT THE SOLUTION(S) HAVE BEEN OBTAINED
C     AND(IF REQUIRED) STORED ON THE SOLUTION FILE.
```

```
C
C       IF MORE SOLUTIONS ARE TO BE OBTAINED WITH THE CURRENT
C       INFLUENCE MATRIX, U11 WILL HAVE TO BE REWOUND AND
C       SPACED TO THE BEGINNING OF THE INFLUENCE MATRIX.
C       THE PROGRAM ASSUMES THIS IS THE CASE UNLESS THE NEXT
C       COMMAND IS HALT, NEW, OR KNEW
        WRITE(U6,888)
        IF(CONV) WRITE(UCO,845)
        NCOM = 1
        READ(UCI,805)(CARDIM(I,NCOM),I=1,4)
        COM = CARDIM(1,NCOM)
        IF (COM .EQ. HALS) GO TO  225
        IF (COM .EQ. NEWS) GO TO   17
        IF (COM .EQ. KNES) GO TO   17
        SKIP = .TRUE.
        NCOM = 1
C
        REWIND U11
        READ (U11)
        WRITE(U6,812)
        GO TO 210
C
  799 FORMAT(L1)
  801 FORMAT(1H ,'ENTER BATCH',/)
  805 FORMAT(4A3)
  806 FORMAT(20A4)
  807 FORMAT(1H ,4A3)
  810 FORMAT(16I5)
  812 FORMAT(1H )
  815 FORMAT(1H ,'ENTER ODISK')
  820 FORMAT(1H ,'ENTER IDD1,IDD2,IDD3 ' )
  840 FORMAT(1H ,'ENTER VALUE FOR NTYPE, 0 FOR SYMMETRIC,',1X,
     1'=1 FOR ANTI-SYMMETRIC,',' 1 FOR BOTH' )
  845 FORMAT(1H ,'+' )
  850 FORMAT(1H ,'UNRECOGNIZED COMMAND  ',4A3 )
  855 FORMAT(1H ,'PROGRAM TERMINATED BY HALT COMMAND'/)
  865 FORMAT(1H ,'ENTER NEW NUMBER OF CHORDWISE MODES',
     11X,'OLD VALUE IS',I5,  ' ORIGINAL VALUE WAS',I2  )
  867 FORMAT(1H ,'INCORRECT VALUE FOR NMAX2,  AT + SIGN',
```

```
     11X,'REENTER COMMAND DCM  ' )
  870 FORMAT(1H ,'ENTER NEW MAXIMUM NUMBER OF SPANWISE MODES',
     11X,'OLD MAXIMUM IS  ',I2/' ORIGINAL VALUE WAS ',I2/)
  874 FORMAT(1H ,'INCORRECT VALUE FOR KK2,  AT + SIGN ',
     1'REENTER COMMAND DSM' )
  875 FORMAT(1H ,'ENTER CHORDWISE POINTS TO BE DELETED ',
     11X,'17 MAXIMUM IN ASCENDING ORDER' )
  876 FORMAT(1715)
  877 FORMAT(1H ,'ENTER SPANWISE CONTROL POINTS TO BE',
     11X,'DELETED, 17 MAXIMUM ASCENDING ORDER' )
  878 FORMAT(1H ,'ENTER INDICES OF THE PARTICULAR ',1X,
     1'POINTS TO BE DELETED--P,M' )
  881 FORMAT(//'0ANTISYMMETRIC MODES ARE NOT AVAILABLE.',
     12X,'PROCEEDING WITH JUST SYMMETRIC MODES.'//)
  882 FORMAT(//'0SYMMETRIC MODES ARE NOT AVAILABLE.',2X,
     1'PROCEEDING WITH JUST ANTISYMMETRIC MODES'//)
  885 FORMAT(1H0,'SOLUTION FOUND AND STORED',/)
  886 FORMAT(1H1/' DETERMINATION OF BNK COEFFICIENTS'/
     1' ***********************************'/
     11X,20A4//' ID1     = ',I3/' ID2     = ',I3/,
     1' ID3     = ',I3/' ID4     = ',I3/,' JJ      = ',I3/,
     2' UNSYM   = ',I3/,' MODES   = ',I3,/' NTYPE   = ',
     3I3/' PPNEW   = ',I3/' MMNEW   = ',I3/' NMAX2   = ',
     4I3/' KK2     = ',I3/' DELTA0  = ',F8.4/' MACH    = ',
     5F8.4/' BCS     = ',10L1/' BCAS    = ',10L1/)
  887 FORMAT(1H , 'SUMMARY OF COMMANDS',/)
  888 FORMAT(//' FINISHED'/)
  890 FORMAT(1H0,'INDICES OF COLUMNS TO BE RETAINED'/)
  892 FORMAT(7(1X,I3,3X))
  894 FORMAT(1H0,'INDICES OF ROWS TO BE RETAINED',/)
  895 FORMAT(/' THE DETERMINANT OF(A-TRANSPOSE)*A = ',
     1E15.8/)
  897 FORMAT(1H0,'CASE ',I2/)
  898 FORMAT(1H0,'BOUNDARY CONDITIONS AT ORIGINAL CONTROL POINTS',
     1/' (L.E. TO T.E.,  RIGHT TIP TO LEFT TIP)'/)
  899 FORMAT(1H0,8(F10.5,3X))
  900 FORMAT(1H0,'SOLUTION FOR UNSYMMETRIC CASES'/)
  901 FORMAT(1H0,2X,5(5X,A4,I3,1X))
  902 FORMAT(1H0,'SOLUTION FOR ANTI-SYMMETRIC CASES'/)
```

```
C 903 FORMAT(/' CASE',5X,'RMS RESIDUAL'/)
  904 FORMAT(1H0,'SOLUTION FOR SYMMETRIC CASES'/)
  906 FORMAT(1H ,F10.5)
  907 FORMAT(1H ,I3,1X,5(F10.7,3X))
  910 FORMAT(1H0,'CHORDWISE CONTROL POINTS USED'/)
  912 FORMAT(1H0,'SPANWISE CONTROL POINTS USED',/)
  914 FORMAT(1H0,'SPECIFIC POINTS DELETED'/'  CHICP',
     110X,'ETACP'/)
  915 FORMAT(1H ,F8.5,9X,F8.5)
  930 FORMAT(2X,I2,7X,F10.6)
931   FORMAT(' OUTPUT IS ON ...OUTPUT,SOLVR,N=',I1,3H...)
932   FORMAT(' IO4=',I5)
933   FORMAT(' NUMBER OF CONTROL POINTS =',I5/
     1       ' NUMBER OF PRESSURE MODES =',I5/)
  999 STOP
      END
      SUBROUTINE SOLVIT(NCOLSB,NCOLSA,NROWSB,NROWSA,NRHSS,READA,IFMT,
     1NUNIT,INDCOL,INDROW,NRECD,INVERS,A,B,IZERO,NB,NX,NAINV,
     2ATA,AR,IL,X,AINV,RMSRES,DET,IERR,NKTP,NROWS)
C
C
C  THIS SUBROUTINE SOLVES A SET OF SIMULTANEOUS EQUATIONS
C  (A)(X)=(B) AND/OR DETERMINES THE INVERSE MATRIX FOR A SQUARE OR
C  RECTANGULAR (A) MATRIX IN THE LEAST SQUARES SENSE USING
C  HOUSEHOLDER'S METHOD (UNITARY TRIANGULARIZATION OF A NON-
C  SYMMETRIC MATRIX; J. ASSOC. COMP. MACH. 5, 1958).
C  THE (A) MATRIX CAN EITHER BE INPUT THROUGH THE ARGUMENT LIST
C  OR READ FROM AN EXTERNAL DEVICE.  IN EITHER CASE THE PROGRAM HAS
C  THE CAPABILITY OF STRIKING OUT ROWS AND/OR COLUMNS BEFORE DETER-
C  MINING THE SOLUTION AND/OR THE INVERSE.
C
C  INPUT:
C  ------
C  NCOLSB = NUMBER OF COLS. BEFORE DELETION OF UNWANTED COLS.
C  NCOLSA = NUMBER OF COLS. AFTER  DELETION OF UNWANTED COLS.
C  NROWSB = NUMBER OF ROWS  BEFORE DELETION OF UNWANTED ROWS.
C  NROWSA = NUMBER OF ROWS  AFTER  DELETION OF UNWANTED ROWS.
C  NRHSS  = NUMBER OF RIGHT HAND SIDES.
C  READA  = LOGICAL VARIABLE WHOSE TRUTH CAUSES THE (A) MATRIX
```

```
C              TO BE READ FROM AN EXTERNAL DEVICE.
C   IFMT    = FORMAT INDICATOR (READA=.TRUE.).
C              0  MEANS UNFORMATTED
C              1  MEANS FORMAT (8F10.0)
C              2  MEANS FORMAT (5F15.0)
C              3  MEANS FREE FORMAT (8F) (IF SUPPORTED BY COMPILER).
C              ANYTHING ELSE CAUSES A RETURN WITH IERR=1.
C   NUNIT   = FORTRAN UNIT NUMBER FROM WHICH TO READ (A) (READA=
C              .TRUE.).
C  (INDCOL)= INDICES OF COLUMNS TO BE RETAINED.
C  (INDROW)= INDICES OF ROWS    TO BE RETAINED.
C   NROWS   = THE NUMBER OF ROWS STORED IN EACH LOGICAL RECORD
C              OR THE NUMBER OF ROWS YOU WISH TO PROCESS AT ONE TIME
C              (READA=.TRUE.) OR THE FIRST DIMENSION OF (A) IN THE
C              PROGRAM WHICH DETERMINED (A) (READA=.FALSE.).
C   NKTP    = THE NUMBER OF COLUMNS OF ATA.  SHOULD BE SET IN CALLING
C              PROGRAM TO NCOLSA + NRHSS FOR INVERS=.FALSE. OR TO TWO
C              TIMES NCOLSA + NRHSS FOR INVERS=.TRUE.
C  (B)      = THE SET OF RIGHT HAND SIDES BEFORE STRIKING OUT UN-
C              WANTED ROWS.  (B) WILL NOT BE DESTROYED.
C   NB      = THE FIRST DIMENSION OF (B) IN THE PROGRAM WHICH DETER-
C              MINED (B).
C   NX      = THE FIRST DIMENSION OF (X) IN THE PROGRAM IN WHICH
C              (X) WILL BE USED.
C   NAINV   = THE FIRST DIMENSION OF (AINV) IN THE PROGRAM IN WHICH
C              (AINV) WILL BE USED.
C   NRECD   = NUMBER OF LOGICAL RECORDS OR NUMBER OF TIMES YOU WISH TO GO
C              THROUGH READ LOOP.  NRECD TIMES NROWS MUST BE EQUAL TO
C              NROWSB PLUS 0 TO ROWS-1
C   IZERO   = COLUMN OF B ARRAY WHERE RIGHT HAND SIDES FOR
C              THIS CASE BEGIN
C
C   OUTPUT:
C   -------
C  (X)      = THE SOLUTION.  IT HAS NCOLSA ROWS AND NRHSS COLUMNS.
C  (AINV)   = THE LEAST SQUARE INVERS OF (A) AFTER THE REQUESTED
C              ROWS AND COLUMNS ARE DELETED.  (AINV) HAS NCOLSA ROWS
C              AND NROWSA COLUMNS.
C  (RMSRES)= THE SQUARE ROOTS OF THE AVERAGE SQUARED ERRORS FOR EACH
```

```
C             SET OF RIGHT HAND SIDES.
C  DET     = THE DETERMINANT OF THE LEAST SQUARE MATRIX.
C  IERR    = ERROR CODE.
C             0  MEANS NO ERROR.
C             1  MEANS READA=.TRUE. AND IFMT WAS NOT 0,1,2, OR 3.
C             2  MEANS THERE WERE FEWER EQUATIONS THAN UNKNOWNS
C                (NROWSA.LT.NCOLSA).
C             3  MEANS NB LESS THAN NROWSB
C
C  OTHER ARGUMENTS:
C  ----------------
C (ATA)     = STORAGE SPACE FOR PERFORMING CALCULATIONS.  THE NUMBER
C             OF ROWS IS NCOLSA AND THE NUMBER OF COLUMNS
C             IS NCOLSA+NRHSS (INVERS=.FALSE.) OR NCOLSA+
C             NRHSS+NROWSA (INVERS=.TRUE.).
C (AR)      = STORAGE SPACE.  THE NUMBER OF ELEMENTS USED = THE
C             NUMBER OF COLUMNS OF (ATA).
C (IL)      = STORAGE SPACE.  THE NUMBER OF ELEMENTS USED = NCOLSA.
C
C  OTHER NOTES:
C  ------------
C  1.  THERE IS A STORAGE ADVANTAGE IN THIS PROGRAM SINCE
C      ONLY 1 ROW OF (A) NEEDS TO BE IN CORE AR ONE TIME.
C  2.  (INDCOL) WILL BE COMPUTED AUTOMATICALLY IF NCOLSA=NCOLSB.
C  3.  (INDROW) WILL BE COMPUTED AUTOMATICALLY IF NROWSA=NROWSB.
C  4.  NB SHOULD BE .GE. NROWSB.
C  5.  IF READA=.FALSE., THEN NROWS SHOULD BE .GE. NROWSB.
C
C
C
C
      LOGICAL READA,INVERS
      DIMENSION A(NROWS,NCOLSB),B(NB,NRHSS),ATA(NCOLSA,NKTP),
     1AR(NKTP),IL(NCOLSA),INDCOL(NCOLSA),INDROW(NROWSA),X(NX,NRHSS),
     2AINV(NAINV,NROWSA),RMSRES(NRHSS)
C
  810 FORMAT(8F10.0)
  820 FORMAT(5F15.0)
  830 FORMAT(8F10.0)
```

```
C
C       ERROR CHECKS
        IF(READA)GO TO 1
        NRECD = 1
        GO TO 2
    1   IF(IFMT.LE.3.AND.IFMT.GE.0) GO TO 2
        IERR = 1
        RETURN
    2   IF(NB.GE.NROWSB) GO TO 3
        IERR = 3
        RETURN
    3   IF(NROWSA.GE.NCOLSA) GO TO 5
        IERR = 2
        RETURN
    5   IF(NCOLSA.NE.NCOLSB) GO TO 7
        DO 6 I = 1,NCOLSA
    6   INDCOL(I) = I
    7   IF(NROWSA.NE.NROWSB) GO TO 9
        DO 8 I = 1,NROWSA
    8   INDROW(I) = I
C
C       INITIALIZATION
C       CLEAR ATA ARRAY
    9   DO 10 J = 1,NCOLSA
        DO 10 K = 1,NKTP
   10   ATA(J,K) = 0.0
        IRSTOP = NROWS
        IXX = IZERO + 1
        JXX = IZERO + NRHSS
        IRBEGN = 1
        IPTROW = 1
        IX = NCOLSA + NRHSS + 1
        INV = IX
        IFMTCK = IFMT + 1
C
C       READ IN ONE LOGICAL RECORD OR AS MANY ROWS AS YOU WISH TO
C       OPERATE ON AT A TIME
        DO 200 K = 1,NRECD
        IF(.NOT.READA) GO TO 40
```

```
      ICHECK = NROWSB-(K-1)*NROWS
      IEND = MIN0(NROWS,ICHECK)
      GO TO (19,20,21,22),IFMTCK
C     CONTROL SHOULD NEVER GET TO HERE
      STOP 13
   19 READ(NUNIT)((A(L,N),N=1,NCOLSB),L=1,IEND)
      GO TO 26
   20 READ(NUNIT,810)((A(L,N),N=1,NC0LSB),L=1,IEND)
      GO TO 26
   21 READ(NUNIT,820)((A(L,N),N=1,NCOLSB),L=1,IEND)
      GO TO 26
   22 READ(NUNIT,830)((A(L,N),N=1,NC0LSB),L=1,IEND)
C     CHECK TO BE SURE WE WANT TO RETAIN AT LEAST ONE ROW
   26 DO 30 J = IRBEGN,IRSTOP
      IF(INDROW(IPTROW).EQ.J) GO TO 40
   30 CONTINUE
      GO TO 190
C
C     BEGIN MOVING ELEMENTS TO AR ARRAY CHECKING THAT THIS A ROW AND
C     COLUMN WE WISH TO RETAIN.
   40 IROW = 1
      DO 170 J = IRBEGN,IRSTOP
      IF(IPTROW.GT.NROWSA) GO TO 170
      IF(INDROW(IPTROW).NE.J) GO TO 170
      M=1
      DO 50 L = 1,NCOLSB
      IF(M.GT.NCOLSA) GO TO 50
      IF(INDCOL(M).NE.L) GO TO 50
      AR(M) = A(IROW,L)
      M = M+ 1
   50 CONTINUE
      I = INDROW(IPTROW)
      DO 60 L = IXX,JXX
      AR(M) = B(I,L)
   60 M = M+1
      IPTROW = IPTROW + 1
      IF(.NOT.INVERS) GO TO 75
      DO 70 L = IX,NKTP
   70 AR(L) = 0.0
```

```
      AR(INV) = 1.0
      INV = INV + 1
C
   75 DO 90 I = 1,NCOLSA
      R = SQRT(ATA(I,I)**2 + AR(I)**2)
      IF(R.EQ.0.0) GO TO 90
      C = ATA(I,I)/R
      S = AR(I) / R
      DO 80 L = 1,NKTP
      T2 = C * ATA(I,L) + S * AR(L)
      AR(L) = -S * ATA(I,L) + C * AR(L)
   80 ATA(I,L) = T2
   90 CONTINUE
  170 IROW = IROW + 1
      IF(READA) GO TO 190
      GO TO 205
  190 IRBEGN = IRSTOP + 1
      IRSTOP = IRSTOP + IEND
  200 CONTINUE
C
  205 II = 1
      DO 220 I = 1,NCOLSA
      IF(ATA(I,I).LE.0.0000001) GO TO 210
      IL(I) = II
      II = II + 1
      GO TO 220
  210 IL(I) = 0
  220 CONTINUE
      DO 365 J = 1,NRHSS
      NKTJ = NCOLSA + J
      DO 300 I = 1,NCOLSA
  300 AR(I) = 0.0
      II = NCOLSA
      DO 340 I = 1,NCOLSA
      IF(IL(II).LE.0) GO TO 340
      JI = IL(II)
      IF(II-NCOLSA) 310,330,350
  310 IK = II + 1
      DO 320 K = IK,NCOLSA
```

```
  320 AR(II) = AR(II) - ATA(JI,K) * AR(K)
  330 AR(II) = (AR(II) + ATA(JI,NKTJ))/ATA(JI,II)
  340 II = II - 1
  350 CONTINUE
      DO 360 I = 1,NCOLSA
      X(I,J) = AR(I)
  360 CONTINUE
  365 CONTINUE
      DET = 1.
      DO 370 I = 1,NCOLSA
  370 DET = DET * ATA(I,I)
      IF(.NOT.INVERS)GO TO 400
C
C     FIND INVERSE
      DO 390 J = 1,NCOLSA
      NKTJ = NCOLSA + NRHSS + J
      DO 373 I = 1,NCOLSA
  373 AR(I) = 0.0
      II = NCOLSA
      DO 380 I = 1,NCOLSA
      IF(IL(II).LE.0.0) GO TO 380
      JI = IL(II)
      IF(II-NCOLSA)375,379,385
  375 IK = II + 1
      DO 377 K = IK,NCOLSA
  377 AR(II) = AR(II) - ATA(JI,K) * AR(K)
  379 AR(II) =(AR(II) + ATA(JI,NKTJ))/ATA(JI,II)
  380 II = II - 1
  385 CONTINUE
      DO 388 I = 1,NCOLSA
  388 AINV(I,J) = AR(I)
  390 CONTINUE
  400 IERR = 0
      RETURN
      END
      SUBROUTINE KRUNCH(N1,INDDEL,N2,INDEX,NU)
C.....THIS SUBROUTINE COMPUTES THE LOGICAL UNION OF THE SET
C.....OF INTEGERS IN (INDEX) AND THE SET OF INTEGERS NOT
C.....IN (INDDEL).
```

```
C.....THE INTEGERS IN (INDDEL) AND (INDEX) ARE ASSUMED TO BE
C.....ARRANGED IN ASCENDING ORDER.  N1 AND N2 ARE THE
C.....NUMBER OF ELEMENTS IN (INDDEL) AND (INDEX), RESPECTIVELY.
C.....ON RETURN, THE UNION IS STORED IN (INDEX) AND THERE ARE
C.....NU ELEMENTS.
C
C.....EXAMPLE
C.....IF (INDDEL)  = 2,4,6,8,9,10,15
C.....AND (INDEX)  = 1,2,3,4,9,10,11,12 ,
C.....THEN ON RETURN (INDEX)=1,3,11,12 AND NU=4.
      DIMENSION INDDEL(N1)
      DIMENSION INDEX(N2)
      N2I=1
      N2M1=N2-1
      NU=N2
      DO 20 I=1,N1
      DO 10 J=N2I,N2
      K=J
      IF(INDEX(J).GT.INDDEL(I)) GO TO 15
      IF(INDEX(J).LT.INDDEL(I)) GO TO 10
      NU=NU-1
      IF(J.GE.N2)RETURN
      DO  8 L=J,N2M1
      INDEX(L)=INDEX(L+1)
    8 CONTINUE
      GO TO 15
   10 CONTINUE
   15 N2I=K
      IF(N2I.GT.N2)RETURN
   20 CONTINUE
      RETURN
      END
      SUBROUTINE STOP2(N,MESAGE,VAL)
      DIMENSION MESAGE(20)
      WRITE(N,1) MESAGE,VAL
      STOP 13
    1 FORMAT(///' ***** ',20A4,' *****'/' VAL=',1PE15.7)
      END
```