
Henry E. Herbert
John E. Lamar

April 1982

NASA Technical Memorandum 83304
PRODUCTION VERSION OF THE
EXTENDED NASA-LANGLEY VORTEX LATTICE
FORTRAN COMPUTER PROGRAM
VOL. II SOURCE CODE

Henry E. Herbert, Computer Sciences Corporation
and

John E. Lamar, NASA - Langley Research Center

Hampton, Va. 23666
CONTENTS

1. ABSTRACT / SUMMARY ........................................... 4
2. INTRODUCTION .................................................. 4
3. PROGRAM STRUCTURE .......................................... 6

APPENDIX A - LRCGOSF ROUTINES USED BY VLM .......... 15
APPENDIX B - LISTING OF THE VLM SOURCE CODE ... 16
REFERENCES ....................................................... 147
FIGURE .......................................................... 148
1 ABSTRACT / SUMMARY

This document presents the source code for the latest production version, MARK IV, of the NASA - Langley Vortex Lattice Computer Program. All viable subcritical aerodynamic features of previous versions have been retained. This version extends the previously documented program capabilities to four planforms, 400 panels, and enables the user to obtain vortex-flow aerodynamics on cambered planforms, flow field properties off the configuration in attached flow, and planform longitudinal load distributions.

2 INTRODUCTION

The NASA - Langley Vortex Lattice FORTRAN Program (VLM) is designed to estimate the subsonic aerodynamic characteristics of up to four complex planforms. The concepts embodied in this program are mostly detailed in references 1,2 and 3; this document is intended to serve as an update to these references for users and computer specialists who have an interest in implementing this program on their computers. Basically, the VLM Program is a segmented program designed to run on the Control Data Corporation (CDC) computers with the NOS operating system. This program requires a run-time field length of approximately 130K (octal) words of memory, and uses the Langley Research Center
Graphics Output System in FORTRAN (LRCGOSF), along with numerous CDC routines that provide random access of mass storage files. The User's Guide associated with this program is listed as reference 4.

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.
3 PROGRAM STRUCTURE

3.1 OVERVIEW

The VLM program consists of one FORTRAN program, 39 FORTRAN subroutines, seven labeled common blocks and nine files. The overall structure of the code can be represented as a tree diagram with seven major branches, as follows:

<table>
<thead>
<tr>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AERODYNAMIC</td>
<td>AERODYNAMIC</td>
<td>NEAR-FIELD SOLUTION</td>
<td>SIDE-EDGE SUCTION</td>
<td>VORTEX FLOW</td>
</tr>
<tr>
<td>EQUATIONS</td>
<td>ANALYSIS</td>
<td></td>
<td>SUCTION ANALYSIS</td>
<td></td>
</tr>
<tr>
<td>MATRIX SOLVER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEOMETRIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUTATIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|                |                |                     |                    |              |
|                |                |                     |                    |              |
| (7)             |                |                     |                    |              |
|                |                |                     |                    | LONGITUINAL LOAD ANALYSIS |

PROGRAM WINGAL
The main program, WINGAL, forms the "root node" to the tree. Each of the seven major branches, or nodes, has a specific function in the overall computational process, and some of these branches consist of more than one routine (this will be explained in the next section). The point of this tree diagram is that it constitutes the way the program is loaded into memory by the operating system, and hence, the way it executes for any given input set. The actual file of loader directives file based on this structure will be discussed in Section 4.

3.2 FUNCTION BY NODE

The root node consists of the main program (WINGAL) and four subroutines. The main program is used to declare all files and common blocks, as well as direct the overall processing done by this code. The four subroutines contained in this node are LOADING, FTLUP, INFSUB and READIN. LOADING is not called by any routine in VLM; its sole purpose is to force the loading of the CDC mass storage routines at this level so they are accessible to all higher level nodes that require them. FTLUP and INFSUB, however, are used by several of the higher nodes, and hence, must be in memory at all times. FTLUP is a linear interpolation routine that uses the CDC Fortran routine LOCF to determine the absolute location (address) of a variable. READIN is used to read in and print out the input data, with line numbers, for the user's reference. The common blocks declared in WINGAL serve as
the principal method of information transfer between the higher nodes; they are considered "global" and must also remain in memory at all times. The higher nodes in the tree are moved in and out of central memory by the loader when they are called by the main program, and this overlapping results in considerable reduction of central memory requirements. The function and associated routines of the higher nodes are as follows:

3.2.1 GEOMETRIC COMPUTATIONS

This node consists of two subroutines, GEOMETRY and PLNPLT. Subroutine GEOMETRY is called by WINGAL to determine, from the input data, the geometry of the configuration. GEOMETRY then makes a call to PLNPLT, which, in turn, produces the "printer plot" of the configuration.

3.2.2 AERODYNAMIC EQUATIONS MATRIX SOLVER

This node consists of six routines and performs the Given's Method of Matrix Solution to determine the circulation terms of the horseshoe vortices; i.e., these routines solve the matrix of the basic linear aerodynamic equations. The routines in this node are:

1. MATXSOL- Called by WINGAL, MATXSOL is the main routine in this node and generates the elements in the aerodynamic influence
coefficient matrix. MATXSOL then calls routine GIVENS to effect a solution.

2. GIVENS - Called by MATXSOL to partition the work storage arrays into rows and columns.

3. BLOCKR - Called by GIVENS to compute the size of, and the number of rows in each partition of the triangularized matrix.

4. TRIANG - Called by GIVENS to triangularize the augmented matrix using planar rotations.

5. SOLVER - Called by TRIANG to perform the back substitution on the matrix and overstore the results back onto mass storage.

6. BUFFIN - Called by TRIANG to transfer data into the work array from mass storage.

3.2.3 AERODYNAMIC ANALYSIS

This node consists of five routines and is used to compute the linear aerodynamic characteristics of the configuration. These routines are:

1. AERODYN - Called by WINGAL, AERODYN is the main routine in this node. AERODYN computes the linear aerodynamic lifting pressures and overall forces and moments.

2. FLOWFL - Determines the flow field characteristics off the wing. FLOWFL reads the field line definition when the flow field data is required. FLOWFL also uses routine FTLUP.

3. HEAPSRT - Called by FLOWFL to sort the X-Y values of the panel centers. FLOWFL compares these values to the X-Y locations of the flow line points in determining the relative position of the flow line with respect to any given planform. HEAPSRT uses a Heap Sort algorithm, which is described in reference 5.

4. SIFT - Called by HEAPSRT to swap values in the sorting process.

5. CDICLS - Computes the far-field induced drag for simple configurations (wing-body) with no dihedral.
3.2.4 NEAR-FIELD INDUCED DRAG SOLUTIONS

This node consists of the single routine, CDRAGNF, (called by WINGAL) that computes the near-field induced drag values. CDRAGNF uses routine FTLUP in determining the near-field chord force properties.

3.2.5 SIDE-EDGE SUCTION ANALYSIS

This node consists of two routines, TIPSUCT and WRTANS. The principal routine, TIPSUCT, called by WINGAL, computes the side-edge force and values for KV,se for each planform. WRTANS is then called by TIPSUCT to compute the values of Kp and KV,le for each planform. If the configuration is cambered/twisted so that the option for vortex flow computation on warped wings is exercised, WRTANS is not called and Kp and KV,le are not calculated. In this situation, TIPSUCT makes numerous calls to the LRCGOSF routines for producing the graphics output. This section of TIPSUCT will have to be revised considerably by users if the VLM program is installed on any other computer. The LRCGOSF routines used by VLM are listed in the Appendix A.
3.2.6 VORTEX FLOW ANALYSIS

This node consists of the single routine, VORTEX, called by WINGAL, to perform the vortex flow analysis for cambered configurations. VORTEX also uses routine FTLUP.

3.2.7 LONGITUDINAL LOAD DISTRIBUTION

This node consists of 18 routines, the principal one being CNLONG, that called by WINGAL. These routines compute the longitudinal load distribution of the configuration. Most of them are associated with the Delta Cp or Net Cp surface interpolation from constant Y to constant X values. They are (by name only) as follows:

- INTERP
- IQHSCV
- IQHSD
- IQHSE
- IQHSF
- IQHSG
- IQSH
- UERTST
- UGETIO
- SUTS
- CURVIN
- CURV12
- CURVI
- CEEZ
- TERMS
- SNHCSH
- INTRVL

The routines beginning with the letters "IQ" provide smooth curve fitting over randomly distributed data points. IQHSCV calculates the interpolating function that is a fifth degree polynomial, and is continuous and has continuous first order partial derivatives. This
technique is detailed in reference 6. Routines IQHSD through UGETIO are support routines for this process. Routines SUTS through INTRVL are used to evaluate the integral of the spline approximation to a function of a single variable. These routines were adapted from current routines on the NASA - Langley Research Center FORTRAN Math Library which is documented in reference 7. This documentation is available from the NASA - LaRC Analysis and Computation Division (ACD) User Support Office.

3.3 FILES

There are nine files associated with the VLM program and these are as follows:

<table>
<thead>
<tr>
<th>Input</th>
<th>TAPE6</th>
<th>TAPE20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>TAPE10</td>
<td>TAPE30</td>
</tr>
<tr>
<td>TAPE5</td>
<td>TAPE11</td>
<td>TAPE81</td>
</tr>
</tbody>
</table>

TAPE81 is equated to INPUT and TAPE6 to OUTPUT, and all formatted INPUT/OUTPUT is performed on these files. Routine READIN reads the input from TAPE81 and prints all of it except the title card over onto TAPE5. For the remainder of the program, TAPE5, serves as the primary input file. The remaining files are scratch files used by the program for auxiliary storage. TAPE10 and TAPE11 are used in the matrix solver routines and are treated as random access backing store. The CDC routines READMS, WRITEMS, OPENMS, and CLOSEMS provide this capability.
TAPE10 is reused as a scratch file in the vortex analysis routine. There it is employed along with TAPE20 and TAPE30, and subsequently, TAPE30 is used to pass information from routine VORTEX back to routine TIPSUCT for the final coefficient reporting.

4 LOAD STRUCTURE

The tree diagram shown earlier can now be represented as follows:

```
GEOMETRY MATXSOL AERODYN CDRAGNF TIPSUCT VORTEX CNLONG


PROGRAM
WINGAL
```

where the nodes are main routines instead of functional descriptors.

On CDC equipment, this method of loading is called Segmentation and is done by the CDC SEGRES Loader Program. The SEGRES program reads a directives file, shown in figure 1, and in effect structures the relocatable binary file into a set of movable pieces (here, each branch of the tree). When the main routine in these branches is called by WINGAL, the Loader brings into memory the entire branch, leaving the others out on mass storage. When a branch finishes processing and returns control back to the main program (WINGAL), that branch is moved back out to mass storage, and another one brought in and loaded into
the same space occupied by the previous one. Hence, the same space in memory is reused, which reduces the core requirements of the program. Complete details regarding the syntax of these directives and the actions of the SEGRES Loader Program are given in reference 8.
APPENDIX A

LRCGOSF Routines Used by VLM

PSEUDO
CALPLT
INFOPLT
NOTATE
PNTPLT

Documentation on these routines is given in reference 9, which is available from the NASA - LaRC ACD User Support Office.
PROGRAM WINGAL(INPUT,OUTPUT,TAPE6=INPUT,TAPE5=OUTPUT,TAPE10=INPUT,TAPE11=OUTPUT,TAPE20=INPUT,TAPE30=OUTPUT)

COMMON /ALL/ BOT, BOTS(4), M, BETA, PTEST, QTEST,
$ STA(4), TBLSW(100), YYCP(4),
$ Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),
$ PHI(100), ZH(100), CP(400), STLIND(4)

COMMON /TOTHREE/ CIR(400, 2)

COMMON /THREEFOR/ CCAV(2, 100), CLT, CLNT, NSSW, ALPD

COMMON /ONETHREE/ TWIST(4), CREF, SREF, CAVE, CLOES, STRUE, AR,
$ ARTRUE, RTCDHT(4), CONFIG(2), NSSWV(4),
$ MSV(4), KBOT, PLAN, IPLAN, MACH,
$ SSW(100), XL(4), XT(4), CLWD, CMCL, CLA(4), BLAIR(100),
$ CLAMAR(4), CLWIN(4), CLWNG(4), XLOCNT,
$ YINIER(4), YOUTER(4)

INTEGER CONFIG

COMMON /MAINONE/ CODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCST(4),
$ XREG(25, 4), YREG(25, 4), AREG(25, 4), DIH(25, 4), MCO(25, 4),
$ XX(25, 4), YY(25, 4), AS(25, 4), TTWD(25, 4), MMC(25, 4), AN(4),
$ ZI(25, 4), ITIPCODE, ICAMTST

PRODUCTION CODE
MARK IV VERSION
SEPTEMBER, 1981

PLEASE DIRECT ALL QUESTIONS, COMMENTS, ETC. TO:
DR. JOHN E. LAMAR
NASA - LANGLEY RESEARCH CENTER
HAMPTON, VA. 23666 M.S. 287

SOFTWARE SUPPORT ON THIS VERSION PROVIDED BY:
HENRY E. HERBERT
COMPUTER SCIENCES CORPORATION
HAMPTON, VA. 23666

--- NOTES TO THE USERS ---

1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING
EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF
ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE
CALCULATED FROM CLDES = 100, AND CLDES = 1.0

2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOMMENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE MORE MEANINGFUL SIDE EDGE RESULTS

3. SILDIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO 0.1 IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO 1.0 IF THIS LOADING IS TO BE NON-ZERO

COMMON/CRRD8, TSPAN(), TSPANA, KBIT, CTILDA, XTILDA, DISTALE
DIMENSION INDEX(2)
VORTEX LATTICE AERODYNAMIC COMPUTATION
NASA-LRC PROGRAM NO. A2794

CALL OPENMS11, INDEX, 1, 0
CALL READIN

ICODEOF=TOTAL=0

CALL GEOMETRY
IF (M .EQ. -1) GO TO 70
IF (IACMST .EQ. 3) GO TO 70
IF (ICODEOF.GT.0) GO TO 70
IF (M .GT. 400) GO TO 70
NSW = 0
GO TO 15 IT = 1, IPLAN
NSW = NSW + NNSWSV(IT)

CONTINUE
IF (NSW .GT. 100) GO TO 30
ITSV = 0
GO TO 20 IT = 1, IPLAN
IF (AN(IT),LE.1.25.) GO TO 20
WRITE (6,100) IT, AN(IT)
ITSV = 1

CONTINUE
IF (ITSV.GT.0) GO TO 60
GO TO 50
30 WRITE (6,90) NSW
  GO TO 60
40 WRITE (6,80) M
  GO TO 60
50 CALL MATHSOL
   CALL AERODYN
   IF (ITIPCOD.EQ.3) CALL CNLONQ
   IF (CLDSC.EQ.100.) GO TO 55
   IF (PTEST.EQ.1..OR.QTEST.EQ.1..OR.ITIPCOD.EQ.2) GO TO 60
   CALL CDRAQNX
   GO TO 57
55 CALL VORTEX
C
   TEST FOR ERROR CONDITIONS DETECTED IN VORTEX
   IF (M.EQ.1) GO TO 70
   C
57 CONTINUE
   IF (ITIPCOD.EQ.1) CALL TIPSYCT
   TOTAL=TOTAL+1.
   GO TO 10
70 CONTINUE
C
80 FORMAT (1H1/10X,4S99HORSESHOE VORTICES LAIDOUT, THIS IS MORE THAN)
   1AN THE 400 MAXIMUM. THIS CONFIGURATION IS ABOR TED.
90 FORMAT (1H1/10X,16,10H ROWS OF HORSESHOE VORTICES LAIDOUT. THIS)
   11S MORE THAN THE 100 MAXIMUM. THIS CONFIGURATION IS ABOR TED.
100 FORMAT (1H1/10X,8HPLANFORM,16,4H HAS,16,74H BREAKPOINTS. THE MAXI)
   MU DIMENSIONED IS 25. THE CONFIGURATION IS ABOR TED.)
       END
       SUBROUTINE READIN
C
       THIS SUBROUTINE READS IN THE DATA OFF -TAPE8-, AND
       PRINTS IT TO -OUTPUT-, AND TO -TAPE3-.
C
       DIMENSION ICARD(8)
       LINE = 0
       WRITE(6,100)
10 READ(6,200) ICARD
   IF (EOF(6)) 30, 20
20 LINE = LINE + 1
   WRITE(6,300) LINE, ICARD
   IF (LINE.EQ.1) GO TO 10
   WRITE(5,200) ICARD
   GO TO 10
C
30 ENDFILE 5
       REWIND 5

19
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>READIN20</td>
</tr>
<tr>
<td>200</td>
<td>READIN21</td>
</tr>
<tr>
<td>300</td>
<td>READIN22</td>
</tr>
<tr>
<td></td>
<td>READIN23</td>
</tr>
<tr>
<td></td>
<td>READIN24</td>
</tr>
<tr>
<td></td>
<td>READIN25</td>
</tr>
<tr>
<td></td>
<td>LOADINGS2</td>
</tr>
<tr>
<td></td>
<td>LOADINGS3</td>
</tr>
<tr>
<td></td>
<td>LOADINGS4</td>
</tr>
<tr>
<td></td>
<td>LOADINGS5</td>
</tr>
<tr>
<td></td>
<td>LOADINGS6</td>
</tr>
<tr>
<td></td>
<td>LOADINGS7</td>
</tr>
<tr>
<td></td>
<td>LOADINGS8</td>
</tr>
<tr>
<td></td>
<td>LOADINGS9</td>
</tr>
<tr>
<td></td>
<td>LOADINGS10</td>
</tr>
<tr>
<td></td>
<td>INFSUB 2</td>
</tr>
<tr>
<td></td>
<td>INFSUB 3</td>
</tr>
<tr>
<td></td>
<td>INFSUB 4</td>
</tr>
<tr>
<td></td>
<td>INFSUB 5</td>
</tr>
<tr>
<td></td>
<td>INFSUB 6</td>
</tr>
<tr>
<td></td>
<td>INFSUB 7</td>
</tr>
<tr>
<td></td>
<td>INFSUB 8</td>
</tr>
<tr>
<td></td>
<td>INFSUB 9</td>
</tr>
<tr>
<td></td>
<td>INFSUB 10</td>
</tr>
<tr>
<td></td>
<td>INFSUB 11</td>
</tr>
<tr>
<td></td>
<td>INFSUB 12</td>
</tr>
<tr>
<td></td>
<td>INFSUB 13</td>
</tr>
<tr>
<td></td>
<td>INFSUB 14</td>
</tr>
<tr>
<td></td>
<td>INFSUB 15</td>
</tr>
<tr>
<td></td>
<td>INFSUB 16</td>
</tr>
<tr>
<td></td>
<td>INFSUB 17</td>
</tr>
<tr>
<td></td>
<td>INFSUB 18</td>
</tr>
<tr>
<td></td>
<td>INFSUB 19</td>
</tr>
<tr>
<td></td>
<td>INFSUB 20</td>
</tr>
<tr>
<td></td>
<td>INFSUB 21</td>
</tr>
<tr>
<td></td>
<td>INFSUB 22</td>
</tr>
<tr>
<td></td>
<td>INFSUB 23</td>
</tr>
<tr>
<td></td>
<td>INFSUB 24</td>
</tr>
<tr>
<td></td>
<td>INFSUB 25</td>
</tr>
<tr>
<td></td>
<td>INFSUB 26</td>
</tr>
<tr>
<td></td>
<td>INFSUB 27</td>
</tr>
<tr>
<td></td>
<td>INFSUB 28</td>
</tr>
<tr>
<td></td>
<td>INFSUB 29</td>
</tr>
<tr>
<td></td>
<td>INFSUB 30</td>
</tr>
<tr>
<td></td>
<td>INFSUB 31</td>
</tr>
<tr>
<td></td>
<td>INFSUB 32</td>
</tr>
<tr>
<td></td>
<td>INFSUB 33</td>
</tr>
<tr>
<td></td>
<td>INFSUB 34</td>
</tr>
<tr>
<td></td>
<td>INFSUB 35</td>
</tr>
</tbody>
</table>

100 FORMAT(14H1,30X,11HINPUT DATA ,//)  
200 FORMAT(8A10)  
300 FORMAT(11O,1H++1X,8A10)  
RETURN  
END  
SUBROUTINE LOADING  
  
THIS IS A DUMMY ROUTINE USED ONLY TO ENSURE THE  
PROPER LOADING OF THE ITEMS NAMED BELOW.  
  
CALL STINDEX(11,INDEX=1+0)  
CALL WRITM(11,INDEX=1+0)  
RETURN  
END  
SUBROUTINE INFSUB (B0T,CF1,CFI,FUI)  
CMJMN /INSUB23/ PSII,APHII,XXX,YYY,ZZZ,SNN,TOLRNC  
FC*COS(PSSII)  
FS*SSIN(PSSII)  
FT*FS/FC  
  
FPC*COS(APHII)  
FP5*SSIN(APHII)  
FPT*FS/FC  
F1*XXX*SNN*FT*FPC  
F2*YYY*SNN*FPC  
F3*ZZZ*SNN*FPS  
F4*XXX*SNN*FT*FPC  
F5*YYY*SNN*FPC  
F6*ZZZ*SNN*FPS  
FFF=((XXX*2+(YYY*FPS)**2+FPC**2*((YYY*FT)**2+(ZZZ/FC)**2-2)*XXX*YYY)INFSUB 14  
1Y*FT)-2.*ZZZ*FPC*(YYY*FPS*XXX*FT*FPS))  
INFSUB 19  
FFF=((F3#1+F2+F2+F3)**.5  
FPC*(F4+F4+F5+F5+F6+F6)**.5  
FFD=F5+F5+F6  
FFE=F2+F2+F2  
FFF=(F1+FPC*FT+F2*FPC+F3*FPS)/FFB-(F4+FPC*FT+F5+FPC+F6*FPS)/FFC  
INFSUB 24  
INFSUB 25  
INFSUB 26  
INFSUB 27  
INFSUB 28  
INFSUB 29  
INFSUB 30  
INFSUB 31  
INFSUB 32  
INFSUB 33  
INFSUB 34  
INFSUB 35  

THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE  
CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES.
10   FUONE = FVONE = FWONE = 0.

C
20   IF (ABS(FFD), LT, TOLRNC) GO TO 30
    FVTWO = F6* (1., -F4/FFC)/FFD
    FWTHRE = F3* (1., -F4/FFC)/FFE
    GO TO 40

30   FVTWO = FWTHRE = 0.

C
40   IF (ABS(FFE), LT, TOLRNC) GO TO 50
    FWTHRE = F3* (1., -F1/FFB)/FFE
    GO TO 60

50   FWTHRE = FWTHRE = 0.

C
60   FWONE = FWTHRE = FWONE
    FVONE = FWONE = FWTHRE
    RETURN
END

SUBROUTINE GEOMTR
DIMENSION NUMBER(4)
DIMENSION XREF(50), YREF(50), SAR(50), A(50), RSAR(50),
$ X(50), Y(50), S(4), VBORD(102), SPY(50, 4),
$ KFX(4), IYL(50, 4), IYT(50, 4)
COMMON /ALL/, B0T, BOTS(4), M, BETA, TETST, GTEST,
$ STA(4), TBLINCW(100), YYCP(4),
$ Q(400), PRW(400), PV(400), ALP(400), S(400), PSI(400),
$ PHI(100), ZH(100), CP(400), STLIND(4)

COMMON /ONETHRE/, TIST(4), CREF, SREP, CAVE, CLDES, STRUE, AR,
$ ATRUE, KTCHT(4), CONFIG(2), NSSW(4),
$ MSV(4), KBDT, PLAN, IPLAN, MACH
$ SWSA(100), X(4), X(4), CL8, CMCL, CLA(4), BLAIR(100),
$ CLAR(4), CLIM(4), CLNWNG(4), XLOCN,
$ YNNER(4), YNTER(4)

C
INTEGER CONFIG

C COMMON /MAINONE/, ICODEOF, TOTAL, AAN(4), XS(4), VS(4), KFCT(4),
$ XREG(25, 4), YREG(25, 4), AREG(25, 4), DIM(25, 4), MCD(25, 4),
$ XX(25, 4), YY(25, 4), AS(25, 4), TWD(25, 4), MMD(25, 4), AN(4),
$ ZI(25, 4), ITIPCOD, ICAMSTT

--- NOTES TO THE USERS ---

C
1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING
EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF
ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE
CALculated FROM CLDES = 100. AND CLDES = 1.0

2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOMMENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE MORE MEANINGFUL SIDE EDGE RESULTS

3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO 0. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO 1.0 IF THIS LOADING IS TO BE NON-ZERO

COMMON/CCTDDD/ TSPAN(4), TSPANA, KBIT, CTILA, XTILA, DISTALE
REAL MACH
INTEGER PRICON
DIMENSION TEMP(8), INDEXES(4,3)
DATA INDEXES / 12*1 /
DATA NUMBER/5HFIRST,6HSECOND,5HTHIRD,6HFORTH/

PART ONE - GEOMETRY COMPUTATION
SECTION ONE - INPUT OF REFERENCE WING POSITION

IF(TOTAL .NE. 0.0) GO TO 5
DO 4 I=1,4
PTCMTH(I) = 0.0
XT(I) = 0.0
XL(I) = 0.0
4 CONTINUE
5 CONTINUE
YTOIL=1.E-10
AZY=1.E+13
PI = 4. * ATAN(1.)
PIT = PI / 2.
RAD = 180. / PI
IF (TOTAL.GT.0.) GO TO 70

SET PLAN EQUAL TO 1. FOR A WING ALONE COMPUTATION - EVEN FOR A VARIABLE SWEEP WING
C SET PLAN EQUAL TO 2. FOR A WING - TAIL COMBINATION
C SET PLAN = 3 OR 4 FOR CANARDS ETC.
C SET TOTAL EQUAL TO THE NUMBER OF SETS
C OF GROUP TWO DATA PROVIDED
C
READ(5,085)
$ PLAN, TOTAL, CREF, SREF, XLOCTN, CTILDA, XTILDA, DISTAL
IF(EQF(5)) 830,10
10 IPLAN=PLAN
XTILDA=XTILDA-XLOCTN
C
C SET AAN(IT) EQUAL TO THE MAXIMUM NUMBER OF CURVES REQUIRED TO
C DEFINE THE PLAINFORM PERIMETER OF THE (IT) PLAINFORM.
C SET RTCHIT(IT) EQUAL TO THE ROOT CHORD HEIGHT OF THE LIFTING
C SURFACE (IT), Whose PERIMETER POINTS ARE BEING READ IN, WITH
C RESPECT TO THE WING ROOT CHORD HEIGHT
C
WRITE (6,860)
PRTCON = 10H
DO 60 IT=1,IPLEN
STLOIND(IT) = 0.0
READ(5,085)
$ AAN(IT), XS(IT), YS(IT), RTCHIT(IT), STLOIND(IT)
N=AAN(IT)
N1=N+1
M=0
IF(PLAN.GT.1) PRTCON = NUMBER(IT)
WRITE (6,870) PRTCON,N,RTCHIT(IT),XS(IT),YS(IT)
WRITE (6,990)
DO 50 I=1,N1
READ(5,085)
$ XREG(I,IT), YREG(I,IT), DIH(I,IT), AMCD
C
IF (AMCD .NE. 2.) AMCD = 1.0
XREG(I,IT)=XREG(I,IT)-XLOCTN
MCD(I,IT)=AMCD
IF (I.EQ.1) GO TO 50
IF (M=0.OR.MCD(I-1,IT).NE.2) GO TO 20
M=I-1
20 IF (ASY(YREG(I-1,IT)-YREG(I,IT),LT,YTOL) GO TO 30
APER(I-1,IT)=(XREG(I-1,IT)-XREG(I,IT))/((YREG(I-1,IT)-YREG(I,IT)))
ASWP=ATAN(AREG(I-1,IT))*RAD
GO TO 30
30 YREG(I,IT)=YREG(I-1,IT)
APER(I-1,IT)=ASY
ASWP=90.
40 J=I-1
C WRITE PLANFORM PERIMETER POINTS AND ANGLES
C WRITE (6,960) J,XREG(J,IT),YREG(J,IT),ASWP,DIM(J,IT),MCD(J,IT)
50 CONTINUE
C KFTETS(IT)=MAX
C WRITE (6,960) N1,XREG(N1,IT),YREG(N1,IT)
CONTINUE

PART 1 - SECTION 2
READ GROUP 2 DATA AND COMPUTE DESIRED WING POSITION

SET S4(1-IPLAN) EQUAL TO SWEEP ANGLE IN DEGREES
FOR THE FIRST
CURVE(S) THAT CAN CHANGE SWEEP FOR EACH PLANFORM
IF A PARTICULAR VALUE OF CL IS DESIRED AT WHICH THE LOADINGS ARE
TO BE COMPUTED, SET CLDES EQUAL TO THIS VALUE
SET CLDES EQUAL TO 11. FOR A DRAG POLAR AT CL VALUES OF -.1 TO 1.0
IF PTEST IS SET EQUAL TO ONE THE PROGRAM WILL COMPUTE CLP
IF QTEST IS SET EQUAL TO ONE THE PROGRAM WILL COMPUTE CMQ AND CLP
DO NOT SET BOTH PTEST AND QTEST TO ONE FOR A SINGLE CONFIGURATION
SET TWIST(1-IPLAN) EQUAL TO 0. FOR A FLAT PLANFORM
AND TO 1.
FOR A PLANFORM THAT HAS TWIST AND/OR CAMBER
SET ATPCOD TO ONE IF THE CONTRIBUTIONS TO LIFT, DRAG AND MOMENT
FROM SEPARATED FLOW AROUND THE LEADING AND/OR SIDE EDGES IS
DESIRED. OTHERWISE SET ATPCOD TO ZERO.

70 READ(5,950) CONFIG(1), CONFIG(2), SCW, VIC, MACH, CLDES,
$ (SA(1),I=1,4),(TWIST(J),J=1,4),PTEST,
$ QTEST,ATPCOD
C
C IF (EOF(5)) 830, 71
71 IF (VIC .GE. 10.) GO TO 72
WRITE(6,1070)
M = -1
GO TO 2000
72 IF (CLDES .NE. 100. .OR. ATPCOD .NE. 1.) GO TO 73
IF (CLDES .EQ. 100. .AND. ATPCOD .EQ. 1. .AND.
$ SCW .GE. 2. ) GO TO 73
WRITE(6,1080)
M = -1
GO TO 2000
73 ICAMTS = 0
DO 74 IT = 1, 4
IF(TWIST(IT).NE. 0.) ICAMTS = 1
74 CONTINUE
IF(ATPCOD .EQ. 1.) ICAMTS = ICAMTS + 1
IF(CLDES .NE. 100.) ICAMTS = ICAMTS + 1
IF(ICAMTS .NE. 3) GO TO 75
WRITE(6,1060)
GO TO 2000
75 ITIPCOD = ATPCOD
IF (ATPCOD .NE. 2.) GO TO 90
IF (CLDES .NE. 11. OR. CLDES .NE. 100.) GO TO 90
M = -1
WRITE(6,1100) CLDES
GO TO 2000
C
90 IF (ITIPCOD .NE. 1) GO TO 110
C
READ IN THE LIMITS OF INTEGRATION ON
THE SIDE EDGES, AND ON THE LEADING/TRAILING
EDGES
(IF ATPCOD = 1)
C
READ(5,885) (INNER(IT), OUTER(IT), IT=1, IPLAN)
READ(5,885) (XL(IT), XT(IT), IT=1, IPLAN)
DO 100 IT = 1, IPLAN
XL(IT) = XL(IT) - XLOCTN
XT(IT) = XT(IT) - XLOCTN
100 CONTINUE
IF (SCW .NE. 0.) GO TO 110
I = 1
DO 107 IIT = 1, IPLAN
IF (XL(IIT).NE. XT(IIT)) I = I + 1
107 CONTINUE
IF (I .EQ. 0) GO TO 110
M = -1
WRITE(6,1110)
GO TO 2000
C
110 CONTINUE
WRITE (6,890) CONFIG
IF (PTEST .EQ. 1. AND. QTEST .EQ. 1.) GO TO 850
IF (SCW .EQ.0.) GO TO 140
DO 125 I = 1, 100
TWS(CW(I)) = SCW
125 CONTINUE
GO TO 150
READ IN THE -STA- AND -TBLSCW- VALUES

ISTART = 0
IEND = 0
NSTA = 0
DO 145 ITT = 1, IPLAN
READ(5,880) STA(ITT)
NSTA = NSTA + STA(ITT)
ISTART = IEND + 1
IEND = NSTA
READ(5,890) (TBLSCW(IKK),IKK=ISTART,IEND)

CONTINUE

DO 410 IT=1, IPLAN
N = AN(I)
N1 = N + 1
DO 150 I = 1, N
XREF(I) = XREG(I, IT)
YREF(I) = YREG(I, IT)
A(I) = REG(I, IT)
RSAR(I) = ATAN(A(I))
IF (A(I).EQ.AZY) RSAR(I) = PIT

CONTINUE
XREF(N1) = XREG(N1, IT)
YREF(N1) = YREG(N1, IT)
IF (<FCTS(IT), GT, 0) GO TO 170
K = 1
S(A(I)) = RSAR(I) * RAD
GO TO 180

K = <FCTS(IT)
WRITE (6,920) K, SA(IT), IT
SB = SA(IT) / RAD
IF (ARS(SB-RSAR(K)), GT, (.1 / RAD)) GO TO 210

REFERENCE PLANFORM COORDINATES ARE STORED UNCHANGED FOR WINGS

WITHOUT CHANGE IN SWEET

DO 200 I = 1, N
X(I) = XREF(I)
Y(I) = YREF(I)
IF (RSAR(I).EQ.PIT) GO TO 190
A(I) = TAN(RSAR(I))
GO TO 200

A(I) = AYZ

SAR(K) = RSAR(I)
X(N1) = XREF(N1)
Y(N1) = YREF(N1)
GO TO 390

CHANGES IN WING SWEET ARE MADE HERE
C 210 IF (MCD(K,IT) .NE. 2) GO TO 840
   KA=K-1
   GO 220 I=1,KA
   X(I)=XREF(I)
   Y(I)=YREF(I)
220   SAR(I)=RSAR(I)
C DETERMINE LEADING EDGE INTERSECTION BETWEEN FIXED AND VARIABLE
C SWEEP WING SECTIONS
   SAR(K)=SB
   A(K)=TAN(S3)
   SSI=SB-RSAR(K)
   X(K+1)=X(S(I))+(XREF(K+1)-X(S(I)))*COS(SAI)+(YREF(K+1)-YS(I)))*SIN(SG(EOMT251)
1AI)
   Y(K+1)=YS(I)+(YREF(K+1)-YS(I))*COS(SAI)-(XREF(K+1)-X(S(I)))*SIN(SG(EOMT253)
1AI)
   IF (ANS(SB-RSAR(K-1)) .LT. (1/RAI)) GO TO 230
   Y(K)=X(K+1)-X(K-1)+A(K)*Y(K+1)+A(K-1)*Y(K-1)
   Y(K)=Y(K)/((A(K-1)-A(K))
   X(K)=A(K)*X(K-1)-A(K-1)*X(K+1)+A(K-1)*A(K)*Y(K)-Y(K-1))
   X(K)=X(K)/((A(K-1)-A(K-1))
   GO TO 260
C ELIMINATE EXTRANEOUS BREAKPOINTS
230   X(K)=XREF(K-1)
   Y(K)=YREF(K-1)
   SAR(K)=SAR(K-1)
240   K=K+1
C SWEEP THE BREAKPOINTS ON THE VARIABLE SWEEP PANEL
   (IT ALSO KEEPS SWEEP ANGLES IN FIRST OR FOURTH QUADRANTS)
   C K=K+1
   SAR(K-1)=SAI+RSAR(K-1)
260   IF (SAR(K-1) .LE. PIT) GO TO 270
   SAR(K-1)=SAR(K-1)+3.1415927
   GO TO 260
270   IF (SAR(K-1) .GE. (-PIT)) GO TO 280
   SAR(K-1)=SAR(K-1)+3.1415927
   GO TO 270
280   IF ((SAR(K-1)) .LT. 0) GO TO 290
   IF (SAR(K-1)=PIT) 320,300,300
290   IF (SAR(K-1)=PIT) 310,310,320
300   A(K-1)=AYZ
   GO TO 330
310   A(K-1)=AYZ
   GO TO 330
320   A(K-1)=TAN(SAR(K-1))
330   K=MCD(K,IT)
   IF (K .EQ. 1) GO TO 350
340   Y(K)=YS(I)+(YREF(K)-YS(I))*COS(SAI)-(XREF(K)-X(S(I)))*SIN(SAI)
   X(K)=X(S(I)+(XREF(K)-X(S(I)))*COS(SAI)+(YREF(K)-YS(I)))*SIN(SAI)
   GEOMT239
   GEOMT240
   GEOMT241
   GEOMT242
   GEOMT243
   GEOMT244
   GEOMT245
   GEOMT246
   GEOMT247
   GEOMT248
   GEOMT249
   GEOMT250
   GEOMT251
   GEOMT252
   GEOMT253
   GEOMT254
   GEOMT255
   GEOMT256
   GEOMT257
   GEOMT258
   GEOMT259
   GEOMT260
   GEOMT261
   GEOMT262
   GEOMT263
   GEOMT264
   GEOMT265
   GEOMT266
   GEOMT267
   GEOMT268
   GEOMT269
   GEOMT270
   GEOMT271
   GEOMT272
   GEOMT273
   GEOMT274
   GEOMT275
   GEOMT276
   GEOMT277
   GEOMT278
   GEOMT279
   GEOMT280
   GEOMT281
   GEOMT282
   GEOMT283
   GEOMT284
   GEOMT285
   GEOMT286
   GEOMT287
   GEOMT288
   GEOMT289
   GEOMT290
   GEOMT291
   GEOMT292
   GEOMT293
   GEOMT294
   GEOMT295
   GEOMT296
   GEOMT297
   GEOMT298
   GEOMT299
   GEOMT300
   GEOMT301
   GEOMT302
   GEOMT303
   GEOMT304
GO TO 250
C DETERMINE THE TRAILING EDGE INTERSECTION
C BETWEEN FIXED AND VARIABLE SWEEP WING SECTIONS
350 IF (ABS(RSAR(K)-SAP(K-1)) .LT.(.1/RAD)) GO TO 360
   Y(K)=XREF(K-1)-X(K-1)-A(K)*XREF(K+1)+A(K-1)*Y(K-1)
   Y(K)=Y(K)/(A(K-1)-A(K))
   X(K)=A(K)*X(K-1)-A(K-1)*XREF(K+1)+A(K-1)*A(K)*YREF(K+1)-Y(K-1)
   X(K)=X(K)/(A(K)-A(K-1))
   GO TO 370
360 X(K)=XREF(K+1)
   Y(K)=YREF(K+1)
370 K=K+1
C STORE REFERENCE PLANFORM COORDINATES ON INBOARD FIXED TRAILING
C EDGE
DO 380 I=K,N1
   X(I)=XREF(I)
   Y(I)=YREF(I)
380 SAP(I-1)=RSAR(I-1)
390 DO 400 I=1,N
   XX(I,IT)=X(I)
   YY(I,IT)=Y(I)
   MCO(I,IT)=MCD(I,IT)
   TTNO(I,IT)=DIH(I,IT)
   AS(I,IT)=A(I)
   XX(N1,IT)=X(N1)
   YY(N1,IT)=Y(N1)
   AN(IT)=AN(IT)
400 CONTINUE
C LINE UP BREAKPOINTS AMONG PLANFORMS
C
BOTS(1) = 0.0
BOTS(2) = 0.0
BOTS(3) = 0.0
BOTS(4) = 0.0
WRITE (6,980)
DO 530 IT=1,IPLAN
   NIT=AN(IT)+1
   DO 470 IIT=1,IPLAN
      IF (ITT.EQ.IT) GO TO 470
      NITT=AN(IIT)+1
   DD 470 IIT=1,IPLAN
   IF (ITT.EQ.IIT) GO TO 470
530 CONTINUE
C
980 CONTINUE
C
GO TO 430 JP=1,NIT
IF (YY(JP,IT),EQ.YY(I,IT)) GO TO 460
420 CONTINUE
DO 430 JP=1,NIT
IF (YY(JP,IT),LT.YY(I,IT)) GO TO 440
430 CONTINUE
GO TO 460

IF(JP EQ 1) GO TO 460

JPSV = JP

IND-NIT-(JPSV-1)

DO 450 JP=1,IND

K2=NIT-JP+2

K1=NIT-JP+1

XX(K2,II)=XX(K1,II)

YY(K2,II)=YY(K1,II)

MCD(K2,II)=MCD(K1,II)

AS(K2,II)=AS(K1,II)

450 TTW(K2,II)=TTW(K1,II)

YY(JPSV,IT)=YY(I,IT)

AS(JPSV,IT)=AS(JPSV-1,IT)

TTW(JPSV,IT)=TTW(JPSV-1,IT)

XX(JPSV,IT)=(YY(JPSV,IT)-YY(JPSV-1,IT)) * AS(JPSV-1,IT) + XX(JPSV-1,IT)

GEOMT352

MCD(JPSV,IT)=MCD(JPSV-1,IT)

AN(IT)=AN(IT)+1

NIT=NIT+1

460 CONTINUE

470 CONTINUE

C

SEQUENCE WING COORDINATES FROM TIP TO ROOT

C

N1=AN(IT)+1

DO 480 I=1,N1

480 O(I)=YY(I,IT)

DO 520 J=1,N1

HIGH=1

DO 490 I=1,N1

IF ((O(I)-HIGH).GE.0.) GO TO 490

HIGH=O(I)

IH=I

490 CONTINUE

IF (J.NE.1) GO TO 500

BOTSV(IT)=HIGH

KFX(IT)=IH

500 Q1(IH)=1

SPV(J,IT)=HIGH

IF (IH.GT.KFX(IT)) GO TO 510

IYL(J,IT)=1

IYT(J,IT)=0

GO TO 520

510 IYL(J,IT)=0

IYT(J,IT)=1

520 CONTINUE

530 CONTINUE

C

GEOMT360

GEOMT369

GEOMT370

GEOMT371

GEOMT372

GEOMT373

GEOMT374

GEOMT375

GEOMT376

GEOMT377

GEOMT378

GEOMT379

GEOMT380

GEOMT381

GEOMT382

GEOMT383

GEOMT384

GEOMT385
SELECT MAXIMUM B/2 AS THE WING SEMISPAN. IF BOTH FIRST AND
SECOND PLANFORMS HAVE SAME SEMISPAN THEN THE SECOND PLANFORM IS
TAKEN TO BE THE WING.

K80 = 1
DO 535 IT = 1, I80
IF (ABS(BOTS(V)) .GE. ABS(BOTS(V(K80)))) K80 = IT
CONTINUE
END

535
BOTS = BOTS(V(K80))

DO 570 IT = 1, I80
TSPAN(IT) = 0.0
ISAVE = KF8(IT) - 1
I = KF8(IT) - 2
IF (I .EQ. 0) GOTO 570
IF (I .EQ. 0) GOTO 550
IF (ITWD(I, IT) .EQ. TTWD(ISAVE, IT)) GOTO 560
CTWD = COS(ATAN(TTWD(ISAVE, IT)))
TLG8 = ((Y(I, I) - Y(I + 1, IT)) / CTWD
TSPAN(IT) = TSPAN(IT) + TLG8
IF (I .EQ. 0) GO TO 570
ISAVE = I
I = I - 1
GO TO 540
570 CONTINUE
VI = TSPAN(K80) / VIC
VSTOL = VI / 2

ELIMINATE PLANFORM BREAKPOINTS WHICH ARE WITHIN (B/2)/2000 UNITS
LATERALLY

DO 630 IT = 1, I80
N = AN(I, IT)
N1 = N + 1
DO 630 J = 1, N
AA = ABS(SPY(J, IT) - SPY(J + 1, IT))
IF (AA .EQ. 0.0) OR AA .GE. ABS(TSPAN(K80)) / 2000.0) GOTO 630
IF (AA .GE. Y20L) WRITE (6, 1010) SPY(J + 1, IT), SPY(J, IT)
DO 620 I = 1, N1
IF (Y(I + 1, IT) .NE. SPY(J + 1, IT)) GO TO 620
Y(I, IT) = SPY(J, IT)
620 CONTINUE
END

630 CONTINUE

COMPUTE Z COORDINATES

670 CONTINUE
```plaintext
JM=NL+AN(IT)+1.
DO 640 JZ=1,NL
640 ZZ(JZ,IT)=RTC(DHT(IT))
JZ=JZ+1
650 JM=JM+1
IF (JZ.GT.KFX(IT)) GO TO 660
ZZ(JZ,IT)=ZZ(JZ-1,IT)+YY(JZ,IT)-YY(JZ-1,IT))*TWD(JZ-1,IT)
GO TO 650
660 JM=JM-1
IF (JM.EQ.KFX(IT)) GO TO 670
ZZ(JM,IT)=ZZ(JM+1,IT)+YY(JM,IT)-YY(JM+1,IT))*TWD(JM,IT)
GO TO 660
670 CONTINUE
C
WRITE PLANFORM PERIMETER POINTS ACTUALLY USED IN THE COMPUTATIONS
C
WRITE (6,900) N
DO 690 IT=1,IPLAN
N=AN(IT)
N1=N+1
IF (IPLAN.GT.1) WRITE(6,1000) NUMBER(IT)
DO 680 KX=1,N
TOUT=ATAN(TWDD(KX,IT))*RAD
ADQD=ATAN(QX(KX,IT))*RAD
IF (AS(KX,IT).EQ.AZY) ADQT=90.
WRITE (6,910) KX,XX(KX,IT),YY(KX,IT),ZZ(KX,IT),ADQD,TOUT,HCMD(KK,IT)
1T
680 CONTINUE
WRITE (6,910) N1,XX(N1,IT),YY(N1,IT),ZZ(N1,IT)
690 CONTINUE
C
PART ONE - SECTION THREE - LAY OUT YAWED HORSESHOE VORTICES
C
STRTX=0.
DO 695 IT=1,IPLAN
MSWSV(IT) = 0
MSV(IT) = 0
695 CONTINUE
MSTD = 0.
NS= JTJ = 0
DO 780 IT=1,IPLAN
N1=AN(IT)+1.
I=0
J=1
YJ=JOTSV(IT)
ILE=LITE(KFX(IT))
C
DETERMINE SPANWISE BORDERS OF HORSESHOE VORTICES
700 IX=I+0
I=I+1
```

---

```
GEMOT435
GEMOT436
GEMOT437
GEMOT438
GEMOT439
GEMOT440
GEMOT441
GEMOT442
GEMOT443
GEMOT444
GEMOT445
GEMOT446
GEMOT447
GEMOT448
GEMOT449
GEMOT450
GEMOT451
GEMOT452
GEMOT453
GEMOT454
GEMOT455
GEMOT456
GEMOT457
GEMOT458
GEMOT459
GEMOT460
GEMOT461
GEMOT462
GEMOT463
GEMOT464
GEMOT465
GEMOT466
GEMOT467
GEMOT468
GEMOT469
GEMOT470
GEMOT471
GEMOT472
GEMOT473
GEMOT474
GEMOT475
GEMOT476
GEMOT477
GEMOT478
GEMOT479
GEMOT480
GEMOT481
GEMOT482
GEMOT483
```
CPIH = \cos(\arctan(\text{TWDO}(ILE,IT)))

IF (YIN,GE,(SPY(J,IT)+VSTOL*CPHI)) GO TO 710
C
BORER IS WITHIN VORTEX SPACING TOLERANCE (VSTOL) OF BREAKPOINT
C
THUS THERE USE THE NEXT BREAKPOINT INBOARD FOR THE BORDER

VBDRT(1)=YIN
GO TO 74C
C
USE NOMINAL VORTEX SPACING TO DETERMINE THE BORDER

710 VBDRT(1)=SPY(J,IT)
C
COMPUTE SUBSCRIPTS ILE AND ITE TO INDICATE WHICH
C
BREAKPOINT ARE ADJACENT AND WHETHER THEY ARE ON THE WING LEADING
C
EDGE OR THE TRAILING EDGE

720 IF (J,GE,N1) GO TO 730
IF (SPY(J,IT),NE,SPY(J+1,IT)) GO TO 730
IXL=IXL+IYL(J,IT)
IXT=IXT+IYT(J,IT)
J=J+1
GO TO 720
730 YIN=SPY(J,IT)
IXL=IXL+IYL(J,IT)
IXT=IXT+IYT(J,IT)
J=J+1
740 CONTINUE
IPHI=ILE-IXL
IF (J,GE,N1) IPHI=1
YIN=YIN-VI*\cos(\arctan(\text{TWDO}(IPHI,IT)))
IF (I,NE,1) GO TO 760
750 ILE=ILE-IXL
ITE=ITE+IXT
GO TO 760
C
COMPUTE COORDINATES FOR CHORDWISE ROW OF HORSESHOE VORTICES

760 YO=(VBDRT(1-1)+VBDRT(1))/2.
YH=(VBDRT(1)-VBDRT(1-1))/2.
IM1 = I - 1 + NSWRT
ZI(1) = ZI(ILE,IT)+(YO-YY(ILE,IT))*TWDO(ILE,IT)
PHI(1) = TWDO(ILE,IT)
SYYA(1) = AS(ILE,IT)
XLE=XLE(ILE,IT)+AS(ILE,IT)*(YO-YY(ILE,IT))
XTE=XTE(ILE,IT)+AS(ITE,IT)*(YO-YY(ITE,IT))
XLOCAL=(XLE-XTE)/TBLSCW(IM1)
C
COMPUTE WING AREA PROJECTED TO THE X - Y PLANE

C
STRUE=STRUE+XLOCAL*TBLSCW(IM1)*(HW*Z1)*Z1.

C
NSCC=TBLSCW(IM1)
DD 770 JCCW+1,NSCC
AJCCW=JCCW-1
XLEL=XLE-ALL*XLOCAL
NTS = JCCW + MSRT
PN(NTS) = XLEL - 25 * XLOCAL
P1(NTS) = XLEL - 75 * XLOCAL
PSI(NTS) = (XLEL - PN(NTS)) * AS(I1E, IT) + (PN(NTS) - XTE) * AS(I1E, IT) / (XLEL - XTE)

S(NTS) = H/C PHI
Q(NTS) = V

CONTINUE

MSV(IT) = MSV(IT) * NSCW
MSTOT = MSTOT + NSCW

C

TEST TO DETERMINE WHEN WING ROOT IS REACHED
IF (VBORED(I1).LT.YREG(I1, IT)) GO TO 750

C

NSSW5(V(IT)) = 1
NSWTOT = NSWTOT + NSSW5(V(IT))

CONTINUE

M = 0
DO 781 IT = 1, IPN(1)
M = M + MSV(IT)

CONTINUE

C

COMPUTE ASPECT RATIO AND AVERAGE CHORD

B = BJB
AR = 4. * BJB / ROT/SPEF
ARTRUE = 4. * BJB / ROT/STRU
CAVE = STRUE / (12. * BJB)
BETA = (1. - MACH * MACH)**.5

C

READ IN THE -ALP- VALUES. IF TWIST(ITT) = 0,
THEN BYPASS THE READ AND SIMPLY SET THEM TO 0.

C

DEGRAD = 1. / RAD
ISTART = 0
IEND = 0
INDXBL = 0
DO 810 ITT = 1, IPN(1)
J = J INNER
INDXBL = INDXBL + 1
NUMOALP = TCLSIDC(INDXBL)
ISTART = IEND + 1
IEND = IEND + NUMOALP
IF (TWIST(ITT) .NE. 0.) GO TO 795
READ(5, 85) (ALP(1TOE), ITOE = ISTART, IEND)
If (TWIST(ITT) .EQ. 0.) GO TO 805
DO 790 ITOE = ISTART, IEND
ALP(1TOE) = ALP(1TOE) * DEGRAD

CONTINUE
GO TO 805
795 DD 800 ITOE = ISTART, IEND
     ALP(ITOE) = 0.0
800 CONTINUE
805 CONTINUE
810 CONTINUE

C
C WRITE (6,1040) M
WRITE (6,1050) (IT, MSV(IT), NSSWSV(IT), IT=1, IPLAN)
NUMHORV = SCW
IF ( SCW .NE. 0.) GO TO 815

C SCW = 0, SO THE USER INPUT THE VALUES OF -TBLSCW-
C CHECK THE NUMBER OF INPUT STATIONS, -STA-, AGAINST
C THE NUMBER COMPUTED IN -NSSWSV-
C WRITE(6,1030) (TBLSCW(I), I=1,NSTA)
I = 0
DO 813 ITT = 1, IPLAN
     J = STA(ITT)
     IF ( J .NE. NSSWSV(ITT)) I = 1
813 CONTINUE
IF ( I .EQ. 0) GO TO 817
M = -1
WRITE(6,1090)
GO TO 2000

C 915 WRITE(6,1020) NUMHORV
817 CONTINUE

C PLOT PLANFORM CONFIGURATIONS ON THE
C LINE PRINTER
C CALL PLANPLT(IPLAN,XX,YY,AN,O,O)

C APPLY PRANDTL-GLAUERT CORRECTION
C DD 820 NV=1,M
PSI(NV)=ATAN(PSI(NV)/BETA)
P(NV)=P(NV)/BETA
820 PV(NV)=PV(NV)/BETA
GO TO 2000
830 ICDEOF=1
WRITE (6,930) CONFIG
GO TO 2G00
840 ICDEOF=2
WRITE (6,940) K, IT

GDM582
GDM583
GDM584
GDM585
GDM586
GDM587
GDM588
GDM589
GDM590
GDM591
GDM592
GDM593
GDM594
GDM595
GDM596
GDM597
GDM598
GDM599
GDM600
GDM601
GDM602
GDM603
GDM604
GDM605
GDM606
GDM607
GDM608
GDM609
GDM610
GDM611
GDM612
GDM613
GDM614
GDM615
GDM616
GDM617
GDM618
GDM619
GDM620
GDM621
GDM622
GDM623
GDM624
GDM625
GDM626
GDM627
GDM628
GDM629
GDM630
GO TO 2000

ICODEDF=3

WRITE (6,970) PTEST,QTTEST

GO TO 2000

C

C

C

GO TO 2000

F80 FORMAT (1HI//63X,13HGEOMETRY DATA)

F80 FORMAT (I4,6X,102HREFERENCE PLANFORM HAS, I3, 7H CURVES//12X,19HGEOMET69

1ROOT CHORD HEIGHT =>F12.5, 4X, 29H VARIABLE SWEEP PIVOT POSITION, 4X, 6HGEOMET60

2X(S) =>F12.5, 5X, 6HY(S) =>F12.5//46X, 40H BREAK POINTS FOR THE REFERENCEGEOMET641

3ENCE PLANFORM //

F80 FORMAT(16F5.1)

F80 FORMAT(8F10.6)

F80 FORMAT(1HI//, 47X, 16H CONFIGURATION =, 2A10)

F80 FORMAT (22X, 5H POINT, 6X, 1HX, 11X, 1HY, 11X, 1HZ, 10X, 5HSWEEP, 7X, 8H DIHEDRAGEOMET646

1AL, 4X, 4HM OVE, 68X, 5HM ANGLE, 8X, 5H M A N G L E, 6X, 4H C O D E //)

F80 FORMAT (20X, 15, 3F12.5, 2F14.5, 16)

F80 FORMAT (/440X, 5H CURVE, 13, 9H IS SWEEPT, F12.5, 20H DEGREES ON PLANFORM, GEOMET649

1I3)

F80 FORMAT(1HI//, 41X, *

$ 43 END OF FILE ENCONTRENDED AFTER CONFIGURATION , 1X, 2A10)

F80 FORMAT (1HI//1RX, 45H THE FIRST VARIABLE SWEEP CURVE SPECIFIED (K =GEOMET653

1, 13, 44H ) DOES NOT HAVE AN M CODE OF 2 FOR PLANFORM, I4)

F80 FORMAT (2A10, 8F5.2, 7F2.0)

F80 FORMAT (26X, 15, 2F12.5, 2F16.5, 4X, 14)

F80 FORMAT (1HI//30X, 38H ERROR, PROGRAM CANNOT PROCESS PTEST =>F5.1, 126GEOMET657

1H AND QTTEST =>F5.1)

F80 FORMAT (/440X, 35H BREAK POINTS FOR THIS CONFIGURATION//)

F80 FORMAT (28X, 5H POINT, 6X, 1HX, 11X, 1HY, 11X, 1HZ, 10X, 5HSWEEP, 7X, 8H DIHEDRAGEOMET660

1HM OVE, 38X, 3HM ANGLE, 8X, 3HM ANGLE, 10X, 5HM ANGLE, 11X, 5HM ANGLE, 9X, 4H CODE//)

F80 FORMAT (52X, 2A10, 22H PLANFORM BREAK POINTS//)

F80 FORMAT (1///25X, 34H THE BREAKPOINT LOCATED SPANWISE AT, F11.5, 3X, 20HGEOMET663

1HAS BEEN ADJUSTED TO, F9.5, ///)

F80 FORMAT(1HI0, 43X, 15, *

$ 41H HOPSESHOE VORTICES IN EACH CHORDWISE ROW)

F80 FORMAT (1///23X, 9H TABLE OF HOPSESHOE VORTICES IN EACH CHORDWISE ROW GEOMET667

1FROM TIP TO ROOT BEGINNING WITH FIRST PLANFORM//25F5.0/25F5.0)

C

F80 FORMAT (///53X, *

$ 3D HOPSESHOE VORTEX SUMMARY TABLE //,

$ 4X, 15, 14H HOPSESHOE VORTICES USED ON THE LEFT,

$ 27H HALF OF THIS CONFIGURATION //,

$ 50X, 8H PLANFORM, 7X, 5H TOTAL, 8X, 8H SPANWISE//

C

F80 FORMAT (52X, 14, 10X, 13, 11X, 14)

F80 FORMAT (1HI, 9X, 22H+ A T A L E R R O R , //,

$ 10X, 46HYOU ARE REQUESTING A VORTEX LIFT SOLUTION FOR //,

$ 10X, 46H CAMBERED WING WITH -CLDES- NOT EQUAL TO 100, //,

35
36

$ 10X* 464THIS WILL NOT YIELD ACCURATE RESULTS. PLEASE ,/, )
$ 10X* 464RESUBMIT WITH *CLOES- * 100.
1070 FORMAT(1H1,9X,22HF A T A L E R R O R ,/, )
$ 10X* 344THE VALUE OF *VIC- IS TOO SMALL. PLEASE ,/, )
$ 10X*344MAKE CORRECTIONS AND RESUBMIT. )

C
1080 FORMAT(1H1,9X,22HF A T A L E R R O R ,/,
$ 10X*344THE VALUE OF -SCW- MUST BE GREATER ,/, )
$ 10X*284THAN 2 FOR THIS COMBINATION. )
C
1090 FORMAT(1H1,9X,22HF A T A L E R R O R ,/
$ 10X* YOUR INPUT VALUES FOR -STA- DID NOT */,
$ 10X* MATCH UP WITH THE NUMBER OF SPANWISE*/,
$ 10X*STATIONS COMPUTED. REFER TO THE HORSESHOE */,
$ 10X* VDPTEX SUMMARY TABLE ABOVE TO OBTAIN THE */,
$ 10X*APPROPER VALUES FOR -STA- PER PLANFORM*/)
C
1100 FORMAT(1H1,9X,22HF A T A L E R R O R ,/
$ 10X*YOU HAVE REQUESTED THE FLOWFIELD OPTION, BUT*/,
$ 10X* HAVE SPECIFIED -CLOES- INCORRECTLY (>*F5.1,*)*/,
$ 10X*PLEASE MAKE CORRECTIONS AND RESUBMIT. */,
C
1110 FORMAT(1H1,9X,22HF A T A L E R R O R ,/
$ 10X*THE SIDE-EDGE FORCE IS NOT PROPERLY COMPUTED */
2000 CONTINUE
C
END
C
SUBROUTINE PLANPLT(IPLAN,XX,YY,AN,ICON,IETHA)
C
C THIS ROUTINE PREPARES A PLOT DIAGRAM OF THE
C INPUT PLANFORM CONFIGURATION(S) FOR THE LINE
C PRINTER
C
C IPLAN- NUMBER OF PLANFORMS TO BE PLOTTED
C XX- X COORDINATE ARRAY DIMENSIONED 25 X 4
C YY- Y COORDINATE ARRAY DIMENSIONED 25 X 4
C AN- NUMBER OF POINTS IN PLANFORM LESS ONE
C ICON- 0 REQUESTS CONTOUR DRAWINGS,
C IETHA- ANGLE OF ROTATION DESIRED IN DEGREES,
C 0 FOR NO ROTATION
C
C ROBERT GRAY COMPUTER SCIENCES CORP 1980
C
C DIMENSION XX(25,4), YY(25,4), AN(4)
C DIMENSION NARRAY (100)
C DIMENSION XXX(25,4), YYY(25,4)
C DIMENSION NXX(26,4), NYY(26,4), NCHAR(4)
C DIMENSION IXPTS(50)
DATA NARRAY / 100*1H /  
DATA IPLUS / 1H/  
DATA IRLANK / 10H /  
DATA NCHAR / 1H, 1H=1H$1H/  
DATA YMIN / -9999.0 /, YMAX / -9999.0 /  
DATA XMIN / -9999.0 /, XMAX / -9999.0 /  

C

ROTATE PLANFORM(S)

THETA = ITHETA + (3.14159/180)
CANS = COS (THETA)
SANS = SIN (THETA)
DO 5 I = 1, IPLAN
M = AN(I) + 1
DO 5 N = 1, M
XXX(I,N) = XX(N, I) * CANS - YY(N, I) * SANS
YYY(I,N) = XX(N, I) * SANS + YY(N, I) * CANS
5 CONTINUE

RESCALE ALL X AND Y COORDINATES
Y IS 6 PER INCH ON THE LINE PRINTER, MAX 7 INCHES
X IS 10 PER INCH, MAX 10 INCHES

FIND MINIMUM AND MAXIMUM COORDINATES
DO 10 I = 1, IPLAN
M = AN(I) + 1
DO 20 N = 1, M
IF (YYY(N, I) .LT. YMIN) YMIN = YYY(N, I)
IF (YYY(N, I) .GT. YMAX) YMAX = YYY(N, I)
IF (XXX(N, I) .LT. XMIN) XMIN = XXX(N, I)
IF (XXX(N, I) .GT. XMAX) XMAX = XXX(N, I)
20 CONTINUE
10 CONTINUE

VALUES USED AS SUBSCRIPTS SHOULD BE GREATER THAN OR EQUAL TO 1

IF (YMIN .GT. 0.0) YUP = 0.0 - YMIN
IF (YMIN .LE. 0.0) YUP = ABS(YMIN)
DO 70 I = 1, IPLAN
M = AN(I) + 1
DO 80 N = 1, M
YYY(N, I) = YYY(N, I) + YUP + 1.0
80 CONTINUE
70 CONTINUE

YMAX = YMAX + YUP + 1
YMIN = YMIN + YUP + 1.0
C

IF (XMIN .GT. 0.0) XUP = 0.0 - XMIN
IF (XMIN .LE. 0.0) XUP = ABS(XMIN)
DO 85 I = 1, PLAN
   M = AN(I) + 1
   DO 90 N = 1, M
      XXX(N,I) = XXX(N,I) + XUP + 1.0
85 CONTINUE
XMAX = XMAX + XUP + 1.0
XMIN = XMIN + XUP + 1.0
C
C   FIND Y SCALING FACTOR---MAXIMUM 42 LINES
YDIM = YMAX
YScale = 42.0 / YDIM
C
C   FIND X SCALING FACTOR---10 CHARACTERS PER INCH
AND IN PROPORTION TO Y INCHES
XScale = YSCALE * 1.666
XDIM = XMAX * XSCALE
C
C   IF (XDIM .LE. 100.49) GO TO 30
C
C   SCALING X DIMENSION IS GREATER THAN 100, RESCALE
C
RESCALE = 100.0 / XDIM
YScale = YSCALE * RESCALE
XScale = XSCALE * RESCALE
C
C   COMPUTE ALL RESCALED X AND Y COORDINATES,
C
C   Rounding Each
DO 40 I = 1, PLAN
   M = AN(I) + 1
   DO 50 N = 1, M
      RX = XXX(N,I) * XSCALE + 10.0 + 5.0
      RXX = RX * RX / 10.0
      RYY = YYY(N,I) * YSCALE + 10.0 + 5.0
      NYY(N,I) = RYY / 10.0
50 CONTINUE
40 CONTINUE
C
C   FIND INTEGER MAXIMA AND MINIMA
YMAX = YMAX * YSCALE + 10.0 + 5.0
MAXY = YMAX / 10.0
YMIN = YMIN * YSCALE + 10.0 + 5.0
MINY = YMIN / 10.0
XMAX = XMAX * XSCALE + 10.0 + 5.0
MAXX = XMAX / 10.0
XMIN = XMIN * XSCALE + 10.0 + 5.0
MINX = XMIN / 10.0
C
CENTER THE DATA

ICLINE = (100 - (MAXX - MINXX)) / 2 - MINXX
DO 65 I = 1, IPLAN
   M = AN(I) + 1
   DO 65 N = 1, M
      NXX(N, I) = NXX(N, I) + ICINTER
   CONTINUE

WRITE HEADING
60 WRITE(*,802) (I,NCHAR(I),I=1,IPLAN)
802 FORMAT(1H1,5LX,34HAPPROXIMATE PLANFORM CONFIGURATION,/
   *(4X,9HPLANFORM,12,6H IS A1))

DRAW THE INPUT PLANFORM CONFIGURATIONS
FOR EACH SCAN LINE
DO 500 ISCAN = MINYY, MAXXY
   IF (ISCAN .EQ. ISCANE) HARRAY (NXX(N, I)) = NCHAR(I)
   CONTINUE

CARRIAGE CONTROL FOR THE FIRST PLANFORM ON A LINE IS BLANK, AFTERWARDS IT IS A PLUS
ICC = IBLANK
FOR EACH PLANFORM
FIND POINTS ON THIS SCAN LINE
DO 400 I = 1, IPLAN
   M = AN(I) + 1
   FIND BREAKPOINTS ON THIS LINE
   DO 110 N = 1, M
      IF (NYY(N, I) .EQ. ISCANE) HARRAY (NXX(N, I)) = NCHAR(I)
   CONTINUE

INSURE A CLOSED PLANFORM
NXX(M+1, I) = NXX(1, I)
NYY(M+1, I) = NYY(1, I)
OFFSET SCAN LINE TO INSURE VERTICES ARE NOT INTERSECTED
SCAN = ISCAN + 0.1
FIND POINTS AT WHICH SCAN LINE INTERSECTS LINE SEGMENTS
ICOUNT = 0
FOR EACH LINE SEGMENT...
DO 300 N = 1, M
   IF THIS SEGMENT IS HORIZONTAL, IT MUST BE
C FILLED SEPARATELY.
C
C IF (ISCAN .NE. NYY(N,I)) GO TO 120
C IF (NYY(N+1,I) .NE. NYY(N,I)) GO TO 120
C ILEFT = MINO(NXX(N,I),NXX(N+1,I))
C IRIGHT = MAXO(NXX(N,I),NXX(N+1,I))
C DO 130 J = ILEFT, IRIGHT
C NAPRAY(J) = NCHAR(I)
C CONTINUE
C ICOUNT = ICOUNT + 1
C MAXNX = MAXO(NXX(N,I),NXX(N+1,I))
C MINNX = MINO(NXX(N,I),NXX(N+1,I))
C IXPTS(ICOUNT) = MINNX
C ICOUNT = ICOUNT + 1
C IXPTS(ICOUNT) = MAXNX
C
C 120
C RYMAX = AMAXO(NYY(N,I),NYY((N+1),I))
C RYMIN = AMINO(NYY(N,I),NYY((N+1),I))
C IF (SCAN .LT. RYMIN .OR. SCAN .GT. RYMAX) GO TO 300
C
C THE SCAN LINE WILL INTERSECTION
C THIS SEGMENT
C
C A = SCAN - NYY((N+1),I)
C B = NXX((N+1),I) - NXX(N,I)
C C = NYY((N+1),I) - NYY(N,I)
C IF SEGMENT IS HORIZONTAL,
C IGNORE IT
C IF (NYY((N+1),I) .EQ. NYY(N,I)) GO TO 300
C D = A * B/C
C
C THE POINT OF INTERSECTION IS
C (IPTINT,SCAN)
C
C RPTINT = (D + NXX(N+1,I)) * 10.0 + 5.0
C IPTINT = RPTINT/10.0
C MAXNX = MAXO(NXX(N,I),NXX((N+1),I))
C MINNX = MINO(NXX(N,I),NXX((N+1),I))
C
C IF INTERSECTION IS TO THE LEFT OR
C RIGHT OF ENDPOINTS, IGNORE IT
C
C IF (IPTINT .GT. MAXNX .OR. IPTINT .LT. MINNX) GO TO 300
C
C STORE AND COUNT IPTINT
C
C ICOUNT = ICOUNT + 1
C IXPTS(ICOUNT) = IPTINT
C CONTINUE
C
C 300
C IF ONLY ONE INTERSECTION FOUND, GO TO NEXT PLANFORM OR SCANLINE
C IF (ICOUNT .LE. 1) GO TO 400
C SORT THE INTERSECTIONS. THEY WILL BE FILLED LEFT TO RIGHT.
C ICT = ICOUNT - 1
DO 310 K = 1, ICT
   INDEX = ICT - K + 1
   DO 310 J = 1, INDEX
      IF (IXPTS(J) .LT. IXPTS(J+1)) GO TO 310
      TEMP = IXPTS(J)
      IXPTS(J) = IXPTS(J+1)
      IXPTS(J+1) = TEMP
   CONTINUE
C CONTOUR DRAWING Requested; PLACE INTERSECTION POINTS IN NARRAY
C IF (ICON .NE. 0) GO TO 320
DO 315 N = 1, ICOUNT
   NARRAY(IXPTS(N)) = NCHAR(I)
CONTINUE
GO TO 340
C CONTRAST DRAWING Requested; FILL IN BETWEEN INTERSECTION POINTS
C DO 320 N = 1, ICOUNT, 2
   ILEFT = IXPTS(N)
   IRIGHT = IXPTS(N+1)
   DO 330 IX = ILEFT, IRIGHT
      NARRAY(IX) = NCHAR(I)
   CONTINUE
C WRITE THE LINE
WRITE(*,204) ICC, NARRAY
204 FORMAT (11, 10X, 100A1)
C PREPARE TO OVERWRITE THE NEXT PLANFORM
ICC = IPLUS
C CLEAR NARRAY FOR NEXT PLANFORM
DO 350 N = 1, 160
   NARRAY(N) = IBLANK
CONTINUE
C 400 CONTINUE
C
PLANP220
PLANP221
PLANP222
PLANP223
PLANP224
PLANP225
PLANP226
PLANP227
PLANP228
PLANP229
PLANP230
PLANP231
PLANP232
PLANP233
PLANP234
PLANP235
PLANP236
PLANP237
PLANP238
PLANP239
PLANP240
PLANP241
PLANP242
PLANP243
PLANP244
PLANP245
PLANP246
PLANP247
PLANP248
PLANP249
PLANP250
PLANP251
PLANP252
PLANP253
PLANP254
PLANP255
PLANP256
PLANP257
PLANP258
PLANP259
PLANP260
PLANP261
PLANP262
PLANP263
PLANP264
PLANP265
PLANP266
PLANP267
PLANP268
500 CONTINUE
C
RETURN
END
SUBROUTINE MATXSOL
DIMENSION YY(2), FV(2), FW(2), FWN(400)
COMMON / ALL/ BOT, BOTS(4), M, BETA, PTEST, QTEST,
$ $ STA(N), TBLSCW(100), YYP(4),
$ $ N(400), PN(400), PV(400), ALP(400), S(400), PSI(400),
$ $ PHI(100), ZH(100), CP(400), STLOIND(4)
C
COMMON /TOTHREE/ CIR(400,2)
C
COMMON /INSUB23/ APSI, APHI, XX, YYY, ZZ, SN, TOLC
DIMENSION WA(20000), INDEX(2)
C
C
PART 2 - COMPUTE CIRCULATION TERMS

THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE
CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES
C
C
TOLC = (BOT*15.E-05)**2
DO 10 NV=1,M
CIR(NV,1)=12.5663704*ALP(NV)
CIR(NV,2)=12.5663704
IF (PTEST.NE.0.) CIR(NV,2)=1.0964155*Q(NV)/BOT
IF (QTEST.NE.0.) CIR(NV,2)=1.0964155*PV(NV)*BETA
10 CONTINUE

IZZ=1
NNV=TBLSCW(IZZ)
REFER 10
DO 20 NV=1,M
DO 20 I=1,M
20 FWN(I)=0.

IZ=1
NNN=TBLSCW(IZ)
DO 60 NN=1,M
APHI=ATAN(PHI(IZ))
APSI=PSI(NN)
XX=PV(NV)-PN(NN)
YY(1)=Q(YY)-Q(NN)
YY(2)=G(NV)+Q(NN)
ZZ=2.(IZZ)-2.(IZ)
SN=5.*NN
DO 30 I=1,2
YYY=YY(I)
30 CONTINUE

PLANP269
PLANP270
PLANP271
PLANP272
MATXSOL2
MATXSOL3
MATXSOL4
MATXSOL5
MATXSOL6
MATXSOL7
MATXSOL8
MATXSOL9
MATXSOL10
MATXSOL11
MATXSOL12
MATXSOL13
MATXSOL14
MATXSOL15
MATXSOL16
MATXSOL17
MATXSOL18
MATXSOL19
MATXSOL20
MATXSOL21
MATXSOL22
MATXSOL23
MATXSOL24
MATXSOL25
MATXSOL26
MATXSOL27
MATXSOL28
MATXSOL29
MATXSOL30
MATXSOL31
MATXSOL32
MATXSOL33
MATXSOL34
MATXSOL35
MATXSOL36
MATXSOL37
MATXSOL38
MATXSOL39
MATXSOL40
MATXSOL41
CALL INFSUB (BOT,FV(1),FW(1),FUI)
APHI=APHI
APSI=APSI
30 CONTINUE
IF (PTEST.NE.0.) GO TO 40
FVN(NN)=FV(1)+FV(2)-(FV(1)+FV(2))*PHI(IZZ)
GO TO 50
40 FVN(NN)=FV(1)-FV(2)-(FV(1)-FV(2))*PHI(IZZ)
IF (NN.LT.NNN,.OR.,NN.EQ.M) GO TO 60
IZ=IZ+1
NN=NN+TBLSCW(IZ)
60 CONTINUE
DUMB=CMZ(NV,1)
DUMY=CMZ(NV,2)
WRITE(10) (FVN(I),I=1,M)
IF (NV.LT.NNV,.OR.,NV.EQ.M) GO TO 70
IZ=IZ+1
NNV=NNV+TBLSCW(IZZ)
70 CONTINUE
LWA=20000
CALL GIVENS(M,M,2,WA,LWA,CIR,10,11,IERR)
CALL STINDX(11,INDEX,1,0)
END
SUBROUTINE GIVENS(NR,NC,NS,W,KORE,B,LEQ,LRAF,FLAG)
C
C SOLVE A LARGE SET OF LINEAR EQUATIONS OF THE FORM AX = B
C IN A SMALL CENTRAL MEMORY WORKING AREA USING GIVENS
C TRIANGULARIZATION. THE SET MAY BE OVERDETERMINED. THE
C A-MATRIX IS STORED BY ROWS ON A SEQUENTIAL FILE, THE WORK-
C ING AREA IS DYNAMICALLY MANAGED USING A RANDOM ACCESS
C FILE CREATED WITHIN THE MODULE.
C
C ON ENTRY
C NR = NUMBER OF ROWS OF A.
C NC = NUMBER OF COLUMNS OF A.
C NS = NUMBER OF RIGHT HAND SIDES.
C W = WORKING AREA ARRAY.
C KORE = LENGTH OF THE WORKING AREA, GE 4 * (NC + NS)
C IF NEGATIVE, KORE IS SET TO FL - W(11) >
C B = WP * NS MATRIX OF RIGHT HAND SIDES.
C LEQ = LOGICAL UNIT NUMBER OF FILE HOLDING ROWS OF A.
C LRAF = LOGICAL UNIT NUMBER OF RANDOM ACCESS FILE.
C (GIVENS SUBINDEXXES THIS FILE.) THE USER
C MUST RESTORE THE MASTER INDEX UPON RETURN)
C
C UPON RETURN
C B HOLDS THE NS SOLUTION VECTORS, EACH STORED OVER THE
C FIRST NC WORDS OF THE CORRESPONDING COLUMN.
C FLAG = 0, EXECUTION PROCEEDED NORMALLY.
C = -- WORKING AREA TOO SMALL, (FLAG) WORDS NEEDED.
C = ++ MATRIX SINGULAR, FLAG = NUMBER OF SINGULAR ROW
C
C DIMENSION W(KORE),R(400,2)
COMMON/MKDAT/ KM*NBLK,NRAF,NROWS,NSEQ,TOL
INTEGER FLAG
C
C INITIALIZE
DO 4 I=1,KORE
  4 W(I)=0.0
FLAG=0
NSEQ=LSEQ
NRAF=LRAF
NCS=4C+NS
TOL=1.0E-100
C
C SIZE THE WORKING AREA.
NWA=**(NC+NS)
IF(KORE.GE.NWA) GO TO 1
FLAG=-NWA
RETURN
C
C PARTITION THE WORKING AREA. NROWS IS NUMBER OF ROWS TO BE
C PROCESSED PER REDUCTION PASS. KM IS PORTION OF WORKING
C AREA AVAILABLE FOR HOUSEKEEPING ARRAYS AND SCRATCH STORAGE
C NBLK IS NUMBER OF BLOCKS OF THE TRIANGULARIZED MATRIX.
   1 NROWS*KORE/(2*NCS)+1
   2 KM*KORE-NROWS*NCS
   3 NBLK=NC*NCS/(2*KM)+1
   4 KM=KM-3*NBLK-2
   5 IF(KM.GE.2*NCS-1) GO TO 3
C
C NROWS OVERESTIMATED, REDUCE.
   6 NROWS=NROWS-1
   7 GO TO 2
C
C COMPUTE PARTITION (BLOCK) SIZES AND NUMBER OF ROWS IN EACH
   3 NB=NBLK
   4 CALL BLOCKR(W,NB,NC,NCS,KORE)
C
C NOW DYNAMICALLY ALLOCATE THE WORKING AREA...
   5 LB=1
   6 LI=1+2*NBLK
   7 LR=LI+NBLK+2
   8 LW=LR+X*NDS*NCS
C
C SUBINDEX THE RANDOM ACCESS FILE.
   9 CALL STINDX(NRAF,W(IL),NRLK+1,0)
C
C AND SOLVE USING GIVENS TRIANGULARIZATION.
CALL TRIANG(W(LB),W(LR),W(LW),B,NR,NC,NS,NCS,NBLK,
       NROWS,KM,FLAG)

C
RETURN
END
SUBROUTINE BLOCK(LBLK,NB,NC,NCS,KDRE)
C
C COMPUTE SIZE OF, NUMBER OF ROWS IN, EACH PARTITION
C OF THE TRIANGULARIZED MATRIX.
C
C R.W. HAMM, COMPUTER SCIENCES CORP., 1974
C
DIMENSION LBLK(2,NB)
COMMON/MOATAF/ KM,NBLK,NRAF,NROWS,NSEQ,TOL
1 NT=NCS
IC=NC
NBLK=0
MAXW=0
C
C LOOP ON BLOCKS OF MATRIX.
2 LBLK=NBLK+1
KW=KM
LNG=0
IRW=0
C
C COMPUTE ROWS WITHIN BLOCK. IRW = NO. OF ROWS, LNG = LENGTH
C OF BLOCK, NT = LENGTH OF ROW I, KW = WORDS REMAINING.
DO 3 I=1,IC
   KW=KW-N
   IF(KW.LT.0) GO TO 4
   LNG=LNG+N
   IRW=IRW+1
   NT=NT-1
3 CONTINUE
4 LBLK(1,NBLK)=LNG
   LBLK(2,NBLK)=IRW
   MAXW=MAXW(LNG,MAXW)
C
C NEXT BLOCK
IC=IC-IRW
IF(IC.GT.0) GO TO 2
C
C TEST IF WKING AREA OVERRUN. MAXW = MAX. PARTITION SIZE.
       NW=ROWS*NMAX=NCS*NROWS*3*NBLK+2
       NDF=NDF*CRE-*NOPS
       IF(NDF.GT.6) GO TO 2
C WORKING AREA OVERRUN. REDUCE BLOCK SIZE AND REITERATE.
5 KM=KM+NDIF
   GO TO 1
C UNDETERMINED, CAN WE INCREASE NROWS...
6 NROWS=NROWS+NDIF/NCS
KMAX=MAXW
7 RETURN

SUBROUTINE TRIANG(LBLK,ROWS,M,NR,NC,NS,NCS,NB,
. . . NRW,KMAX,FLAG)
C
C TRIANGULARIZE THE AUGMENTED MATRIX USING PLANE
C ROTATIONS, THEN BACK-SUBSTITUTE FOR SOLUTIONS.
C
C
DIMENSION LBLK(2,NB),ROWS(NCS,NRW),M(KMAX),B(400,2)
COMMON/HDATAR/ KM,NBLK,NRAF,NROWS,NSEQ,TOL
INTEGER FLAG
C
C INITIALIZE SCRATCH FILE NRAF TO SIMPLIFY LATER LOGIC.
DO 1 I=1,KM
1 W(I)=0.0
DO 2 N=1,NBLK
LNG=LBLK(1,N)
CALL WPMTSH(NRAF,W,LNG,N)
2 CONTINUE
NCSP=NC+1
IR=0
C
C TRIANGULARIZATION LOOP. READ NROWS ROWS PER PASS, AUGMENT.
REWIND NSEQ
DO 15 ISET=1,N,NROWS
DO 4 JR=1,NROWS
   CALL BUFFIN(NSEQ,ROWS(I,JRR),NC,IEOF)
   IF(IEOF.EQ.0) GO TO 5
   JR=JRR
   IS=IR+JR
   DO 3 J=1,NS
      NCJ=NC+J
      3 ROWS(NCJ,JR)=B(IS,J)
   CONTINUE
   5 IR=IR+1
   NRW=0
C
C READ IN A BLOCK AND SET ITS ROW INDICES.
DO 13 K=1,NBLK
   LNG=LBLK(1,K)
   NW=LNG+1
   NRWZ=LBRK(2,N)+NRWZ
   IF(IWLT,NPW) GO TO 14
CALL READMS(NRAF,W,LNG,N)

C
C LOOP THROUGH ROWS READ IN THIS PASS.
LP=0
DO 12 I=IR,IRR
   LR=LR+1
   JJ=1
C AND PROCESS ROWS OF THIS BLOCK.
DO 11 J=NRW1,MRW2
   IF(I-J) 12,9,6
   AB=ROWS(JJ,LR)
   IF(ABS(AB).LT.TOL) GO TO 8
   6 R=SQR(AB*AB+W(JJ)*W(JJ))
   IF(R.LT.TOL) GO TO 8
   C=W(JJ)/R
   S=AB/R
   W(JJ)*R
   JJ=JJ+1
   JP=J+1
   KK=JJ+1
C
C ELIMINATE ELEMENT J OF INPUT ROW.
DO 7 K=JP,NCS
   T=C*W(KK)+S*ROWS(K,KR)
   7 CONTINUE
C
C ELEMNTS J TO NCS OF ROWS TO WORKING AREA.
   K=JJ
   P=SIGN(1,0,ROWS(JJ,LR))
   DO 10 K=J,NCS
      W(KK)=R*ROWS(K,KR)
      KK=KK+1
   10 CONTINUE
C
C CALL WITMS(NRAF,W,LNG,N,1)
C
C BACK-SUBSTITUTE FOR SOLUTION
CALL SOLVER(LBLK,W,B,NR,NC,NS,NB,KMAX,FLAG)
RETURN
END
SUBROUTINE SOLVER(LBLK,W,B,MR,NC,NS,NB,KMAX,FLAG)
C BACK SUBSTITUTION FOR X. STORE OVER B.
C
C
DIMENSION LBLK(2,NB), W(KMAX), B(N400*2)
COMMON/HDATAR/KM, NBLK, NMAXB, NROWS, NSEQ, TOL
INTEGER FLAG
C
LE = 0
NK = NBLK + 1
NSP = NS + 1
MR = NC
C
C BLOCK LOOP. LAST BLOCK FIRST.
DO 6 NN = 1, NBLK
   N = NK - NN
   LNG = LBLK(N, N)
   CALL READS(NMAXB, LNG, N)
   NROW = LBLK(2, N)
   KK = LNG - NS
C
C ROWS IN THIS BLOCK.
DO 5 I = 1, NROW
   DO 4 JS = 1, NS
      KK = KK + JS
      B(MR, JS) = W(KK)
C
C SOLUTIONS
   DO 4 JS = 1, NS
      TERM = 0.0
      LN = LE
      LR = NC
      JJ = KK
      C SUBSTITUTION
      2 IF(LN .EQ. 0) GO TO 3
      TERM = TERM + W(JJ) * B(LR, JS)
      LR = LR - 1
      LN = LN - 1
      JJ = JJ - 1
      GO TO 2
   C TEST FOR SINGULAR ROW
   3 IF(W(JJ) .LT. TOL) GO TO 7
   4 B(LP, JS) = B(LP, JS) - TERM / W(JJ)
      MP = MP - 1
      LE = LE + 1
      KK = JJ - NSP
      CONTINUE
   CONTINUE
SOLVER 3
SOLVER 4
SOLVER 5
SOLVER 6
SOLVER 7
SOLVER 8
SOLVER 9
SOLVER 10
SOLVER 11
SOLVER 12
SOLVER 13
SOLVER 14
SOLVER 15
SOLVER 16
SOLVER 17
SOLVER 18
SOLVER 19
SOLVER 20
SOLVER 21
SOLVER 22
SOLVER 23
SOLVER 24
SOLVER 25
SOLVER 26
SOLVER 27
SOLVER 28
SOLVER 29
SOLVER 30
SOLVER 31
SOLVER 32
SOLVER 33
SOLVER 34
SOLVER 35
SOLVER 36
SOLVER 37
SOLVER 38
SOLVER 39
SOLVER 40
SOLVER 41
SOLVER 42
SOLVER 43
SOLVER 44
SOLVER 45
SOLVER 46
SOLVER 47
SOLVER 48
SOLVER 49
SOLVER 50
SOLVER 51
SUBROUTINE AERODYN

DIMENSION CLCC(400,2), CH(2,100), SUM(3), YCP(4), CROLL(4),
$ AC(4), CLCL(2,100), P(400), SMDAD(2,100),
$ SLDR(100), SM(2,100), SCT(5), SAT(5)
COMMON /ALL/ BOT, BOTS(4), M, BETA, PTEST, QTEST,
$ STA(4), TBLSCW(100), YYCP(4),
$ Q(400), PM(100), PV(400), ALP(400), S(100), PSI(400),
$ PHI(100), ZH(100), CP(400), STQIND(4)
COMMON /TOTHREE/ CIR(400,2)
COMMON /THREFO2/ CCAV(2,100), CLT, CNT, NSSW, ALP
COMMON /ONETHRE/ T4IST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,
$ ARTRE, RTCOPT(4), CONFIG(2), NSSWSV(4),
$ MSV(4), K50T, PLAN, IPLAN, MACH,
$ SSWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),
$ CLAMAR(4), CLWIN(4), CLWNG(4), XLOCIN,
$ YINNER(4), YOUTER(4)

INTEGER CONFIG

COMMON /THRECOD/ SLOAD(3,100)
COMMON /INSUB3/ APSI, APHI, XXX, YYY, ZZZ, SNN, TDLG
COMMON /MAINONE/ ICODEOF, TOTAL, AAN(4), XS(4), YS(4), XFTS(4),
$ XR(25,4), YR(25,4), AR(25,4), DIH(25,4), MCD(25,4),
$ XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),
$ ZZ(25,4), ITIPCOD, ICMTST

DIMENSION NUMBER(4)
DATA NUMBER/5HFIRST, 6HSECOND, 5HTHIRD, 6HFORTH/

PART 3 - COMPUTE OUTPUT TERMS

CLSAVE=CLDES
IF (CLDES.EQ.100,-) CLDES=1.
PART 3 - SECTION 1

COMPUTE LIFT AND PITCHING MOMENT HERE

IZ=1
NNN=TBLSWC(IZ)
DO 10 I=1,M
PI=S(I)*COS(ATAN(PHI(IZ)))
IF (I.LT.NNN.OR.I.EQ.M) GO TO 10
IZ=IZ+1
NNN=NNN+TBLSWC(IZ)
10 CONTINUE
IT = 1
SUM(1) = 0.0
SUM(2) = 0.0
SUM(3) = 0.0
MSUM = MSV(IT)
DO 20 I=1,M
SUM(1) = SUM(1) + CIR(I,1) * PI
SUM(2) = SUM(2) + CIR(I,2) * PI
SUM(3) = SUM(3) + CIR(I,2) * PI * Q(I)
IF (I.EQ.MSUM) GOTO 15
GOTO 20
15 CLWNG(IT) = SUM(1) * 8.0 / SREF
CLWIN(IT) = SUM(2) * 8.0 / SREF
CPOLL(IT) = SUM(3) * 8.0 / (SREF*BOT)
IT = IT + 1
MSUM = MSUM + MSV(IT)
20 CONTINUE
CLT=0.*SUM(1)/SREF
CLNT=3.*SUM(2)/SREF
IF (KBT.EQ.1) GO TO 30
CLWNGT = CLWNGK(KBT) - CLWNGK(KBT-1)
CLWING = CLWINK(KBT) - CLWINK(KBT-1)
GOTO 35
30 CLWNG = CLWNG(1)
   CLWNG = CLWNG(1)
35 CRL = 0.0
   DO 40 I=1,M
50 CLCC(I,1) = CIR(I,1)*2/CAVE
   CLCC(I,2) = CIR(I,2)*2/CAVE
C
C
CLP = CRL/(SREF*BOT*0.08725)
CLA(2)*CLNT
DO 120 IXX=1,2
SA = SB = SC = 0.
I = 0
IT = 0
JB = NSWSV(I)
JA = 1
CONTINUE
50 DO 70 JSSW = JA, JB
   SD = SE = 0.
60 SLOAD(IXX,JSSW) = 0
   NSC = TBLSCW(JSSW)
   DO 70 JSCW = 1, NSC
70 IF (TWS.T.EQ.0 ..AND. IXX.EQ.1) GO TO 60
    I = I + 1
   SA = SA + CIR(I,IXX)*P(I)
   SB = SB + CIR(I,IXX)*Q(I)*P(I)
   SC = SC + CIR(I,IXX)*P(N(I))*P(I)*BETA
   SLOAD(IXX,JSSW) = SLOAD(IXX,JSSW) + (BOT*CIR(I,IXX)*1.)/(2.*SUM)
   L(IXX)
   AA = P(I)/S(I)
   SD = SD + CIR(I,IXX) * AA
   SE = SE + CIR(I,IXX) * PN(I) * BETA * AA
   IF (JSCW.NE.NSCW) GO TO 60
   SMOAD(IXX,JSSW) = SE
   SMLD(IXX,JSSW) = SD
   GO TO 70
70 CONTINUE
60 SLOAD(1,JSSW) = SMOAD(1,JSSW) + SMLD(1,JSSW) = 0.
   CONTINUE
   IT = IT + 1
   IF (JSSW.GE.NSSW) GO TO 80
   JA = JA + NSWSV(IT)
   JB = JB + NSWSV(IT + 1)
   IF (IXX.EQ.1) GO TO 50
   IF (IT.EQ.11) GOTO 75
   SCT(IT) = SC
   SAT(IT) = SA
   AERODY73
   AERODY74
   AERODY75
   AERODY76
   AERODY77
   AERODY78
   AERODY79
   AERODY80
   AERODY81
   AERODY82
   AERODY83
   AERODY84
   AERODY85
   AERODY86
   AERODY87
   AERODY88
   AERODY89
   AERODY90
   AERODY91
   AERODY92
   AERODY93
   AERODY94
   AERODY95
   AERODY96
   AERODY97
   AERODY98
   AERODY99
   AERODY100
   AERODY101
   AERODY102
   AERODY103
   AERODY104
   AERODY105
   AERODY106
   AERODY107
   AERODY108
   AERODY109
   AERODY110
   AERODY111
   AERODY112
   AERODY113
   AERODY114
   AERODY115
   AERODY116
   AERODY117
   AERODY118
   AERODY119
   AERODY120
   AERODY121
CLAMAR(IT) = (SCT(IT) - SCT(IT-1)) / ((SAT(IT) - SAT(IT-1)) * CREF)  AEROD122
GOTO 50  AEROD123
GOTO 50  AEROD124
SCT(IT) = SC  AEROD125
SAT(IT) = SA  AEROD126
CLAMAR(IT) = SCT(IT) / (SAT(IT) * CREF)  AEROD127
GO TO 50  AEROD128
CONTINUE  AEROD129
IF (IXX.EQ.1) GO TO 100  AEROD130
IF (IPLAN.EQ.1) GO TO 90  AEROD131
SCT(IT) = SC  AEROD132
SAT(IT) = SA  AEROD133
CLAMAR(IT) = (SCT(IT) - SCT(IT-1)) / ((SAT(IT) - SAT(IT-1)) * CREF)  AEROD134
GO TO 100  AEROD135
CLAMAP(1)=SC/(SA*CREF)  AEROD136
CONTINUE  AEROD137
IF (TWST.FQ.0,.AND.IXX.EQ.1) GO TO 110  AEROD138
YCP(IXX)=SB/(SA*BOT)  AEROD139
AC(IXX)=SC/(SA*CREF)  AEROD140
GO TO 120  AEROD141
YCP(1)=AC(1)=0.  AEROD142
CONTINUE  AEROD143
CMCL=AC(2)  AEROD144
CMOD=(AC(1)-AC(2))*CLT  AEROD145

PART 3 - SECTION 2

COMPUTE OTHER- AND PRINT ALL FINAL- OUTPUT DATA HERE

C

DO 140 IXX=1,2  AEROD146
JN=0  AEROD147
DO 140 JSSW=1,NSSW  AEROD148
CH(IXX,JSSW)=0  AEROD149
NSCW=TBLSW(JSSW)  AEROD150
DO 130 JSCW=1,NSCW  AEROD151
JN=JN+1  AEROD152
CH(IXX,JSSW)=(-2.0)*(PV(JN)-PN(JN))*BETA+CH(IXX,JSSW)  AEROD153
CONTINUE  AEROD154
CCAV(IXX,JSSW)=CH(IXX,JSSW)/CAVE  AEROD155
CLCCL(IXX,JSSW)=SLOAD(IXX,JSSW)/CCAV(IXX,JSSW)  AEROD156
CONTINUE  AEROD157
CLD=CLDES  AEROD158
IF (CLDES.EQ.11) CLD=1.  AEROD159
GO TO 156  AEROD160
CP(I) = (CLCCL(1,1)+CLCCL(I,2)*(CLD-CLT)/CLNT)*CAVE/(2.*PN(I)-PV(I))  AEROD161
1BETA  AEROD162
CONTINUE  AEROD163
WRITE (6,240) CP(I)  AEROD164
WRITE (6,340) CP(I)  AEROD165
AEROD166
AEROD167
AEROD168
AEROD169
AEROD170
WRITE (6,360) CLD
HEAD=$HDEP$
IF (CLDES.EQ.11.) HEAD=$H$
IEND=11
IF (CLDES.NE.11.) IEND=1
DO 190 IUTK=1,IEND
IF (IEND.EQ.11.) CLDES=(FLOAT(IUTK)-1.)/10.
IF (CLDES.EQ.0.) CLDES=-1.
NR=0
DO 160 NV=1,NSSW
NSCW=TVLSCW(NV)
NP=NR+1
NR=NR+NSCW
PHIPR=ATAN(PHI(NV))*RAD
SLOAD(3,NV)=0.
IF (IUTK.NE.1.) GO TO 155
NVS = 1
DO 154 IT = 1,IPLAN
JJ = IT - 1
IF (JJ.EQ.0) GOTO 153
IF (INV.NE.NVS) GOTO 154
WRITE(6,230) NUMBER(IT)
GOTO 155
CONTINUE
155 CONTINUE
DO 160 I=NP,NR
IF (IUTK.GT.1) GOTO 160
PNPR=PN(I)*BETA
PVPR=PV(I)*BETA
PSIPR=ATAN(BETA*TAN(PSI(I)))*RAD
WRITE (6,370) PNPR,PWPR,O(I),ZHV(NV),S(I),PSIPR,PHIPR,ALP(I),CP(I)
SLOAD(3,NV)=SLOAD(3,NV)+CLCC(I$2$)*CLDES/CLNT+CLCC(I$1$)*CLCC(I$2$)*C
1LT/CLNT
IF (IUTK.GT.1) GOTO 170
WRITE (6,270)
WRITE (6,290) CREF,CAVE,TRUE,SREF,BOT,AR,ARTRUE,MACH
170 CONTINUE
C
C
IF (PTEST.NE.0.) WRITE (6,380) CLP
IF (PTEST.NE.0.) GOTO 220
C
C
CMQ=2.*CMCL*CLNT/($0.08725*CREF)
CLO=CMCL*CLNT/($0.08725*CREF)
IF (QTEST.NE.0.) WRITE (6,390) CMQ,CLO
IF (QTEST.NE.0.) GOTO 220
C
C
COMPUTE INDUCED DRAG FOR FLAT WING-BODY WITH NO DIHEDRAL

NSV = 1
MTOT = 1
DO 190 IT = 1, KBOT
MTOT = MTOT + MSV(IT)
NSV = NSV + NSSMSV(IT)
190 CONTINUE
CALL CDCLS (AR, ARTRUE, NSSMSV(KBOT), MTOT, NSV, CDI, CDIT)
CLAPD = CLA(2)/57.29578
ALPD = (CLT/CLA(2))*57.29578
ALPC = CLDES/CLAPD+ALPD
ALPW = 1./CLAPD
CLW = CLWING*ALPD/57.29578*CLWNG
CDIV = CDI/(CLW+CLWB)
IF (IUTK, EQ, 1) WRITE (6, 250) HEAD, CDIT
WRITE (6, 260) CLDES, ALPD, CLWB, CDI, CDIW
WRITE (6, 290) CLA(2), CLAPD, CLT, ALPD, YCP(2), CMCL, CMD
YVCP(1) = YCP(2)
IF (IPLAN, EQ, 1) GOTO 194
DO 193 IT = 1, IPLAN
IF (IT, GT, 1) GOTO 191
CLTWST = CLWNG(1)
CLALPHA = CLWIN(1)
YVCP(1) = CROLL(1) / CLALPHA
GOTO 192
191 CLTWST = CLWNG(IT) - CLWNG(IT-1)
CLALPHA = CLWIN(IT) - CLWIN(IT-1)
YVCP(IT) = (CROLL(IT) - CROLL(IT-1)) / CLALPHA
192 CLAPDIT = CLALPHA / 57.29578
ALPOIT = - (CLTWST / CLALPHA) * 57.29578
WRITE (6, 400) NUMBER(IT), CLALPHA, CLAPDIT, CLTWST,
ALPOIT, YVCP(IT)
193 CONTINUE
194 CONTINUE
WRITE (6, 300) CLT
NV = J = D
DO 210 NV = 1, NSSW
BCLCC = 3ADLAE + BASLD + 0.
NSC = TBLSCW(NV)
NP = NV + 1
NR = NP
DO 230 I = NP, NR
ADLAE = CCLCC(I, 2)/CLT/CLNT
BASLD = CLC(I, 1) + ADLAE
RCLCC = BCLCC + CCLCC(I, 1)
BADLAE = BADLAE + ADLAE
BASLD = BASLD + BASLD
AEROD220
AEROD221
AEROD222
AEROD223
AEROD224
AEROD225
AEROD226
AEROD227
AEROD228
AEROD229
AEROD230
AEROD231
AEROD232
AEROD233
AEROD234
AEROD235
AEROD236
AEROD237
AEROD238
AEROD239
AEROD240
AEROD241
AEROD242
AEROD243
AEROD244
AEROD245
AEROD246
AEROD247
AEROD248
AEROD249
AEROD250
AEROD251
AEROD252
AEROD253
AEROD254
AEROD255
AEROD256
AEROD257
AEROD258
AEROD259
AEROD260
AEROD261
AEROD262
AEROD263
AEROD264
AEROD265
AEROD266
AEROD267
AEROD268
200 CONTINUE
SLOT(NV)?SMOAD(1,NV)+SMOAD(2,NV)+(CLDES-CLT)/CLNT)/(SMLD(1,NV)+SMAEDD270
1LD(2,NV)*(CLDES-CLT)/CLNT)
AEROD269
J=J+NSCw
YQ*G(J)/BGT
NVS = 1
AEROD271
DO 205 IT = 1,IPLAN
JJ = IT - 1
AEROD272
IF(JJ.EQ.0) GOTO 204
AEROD273
NVS = NVS + SSSWV(JJ)
AEROD274
204 IF(INKX,NE,NVS) GOTO 205
AEROD275
WRITE(6,310) NUMBER(1T)
AEROD276
GOTO 206
AEROD277
205 CONTINUE
AEROD278
206 CONTINUE
AEROD279
210 WRITE (6,320) NV,YQ,LOAD(2,NV),CLCL(2,NV),CCAV(2,NV),BCLC,BADELAEEDR280
&18ASLD,LOAD3(3,NV),SLOT(NV)
AEROD281
220 CONTINUE
AEROD282
CLDES=CLSAVE
AEROD283
IF(ITIPCOD.EQ.2) CALL FLOWFL
AEROD284
C
AEROD285
C
AEROD289
C
AEROD290
C
AEROD291
230 FORMAT(/12X,AL,39H PLANFORM HORSESHOE VORTEX DESCRIPTIONS/)
AEROD292
240 FORMAT(1H1,///,55X,16H AERODYNAMIC DATA ///,54X,)
AEROD293
$ 16H CONFIGURATION : /3ZAL0 /
AEROD294
250 FORMAT (1H1,14X,22H COMPLETE CONFIGURATION + 31X,25H WING-BODY CHARACTERISTIC)
AEROD295
1ERISTICS/64X,4HLIFT,9X,33INDUCED DRAG (FAR FIELD SOLUTION)///16X ABEROD296
021H CL COMPUTED ALPHA,19X,6HCL(CLW),7X,13HCDI AT CL(CLW),4X,15HCAEDR297
0VCL(ClW)**2///,8X,16H (1/P*AR REF) = ,8F.5,11H)
AEROD298
260 FORMAT (11X,2F15.5,15X,3F15.5)
AEROD299
270 FORMAT (///,4X,11H REF. CHORD 6X,25H AVERAGE TRUE AREA ,2X,1AEROD300
&14HREFERENCE AREA,9X,3M8/2,9X,7HREF. AR,8X,7HTRUE AR,4X,11MACH NUMBER 4
2AE/) AEROD301
280 FORMAT (9F15.5)
AEROD302
290 FORMAT (///,47X,38H COMPLETE CONFIGURATION CHARACTERISTICS///36X,CMLC AEROD303
&01 ALPHA,18X,5HCL(TEST) ALPHA AT CL=0 Y CP CM/CL CMDE AEDP304
2/27X,23MPER RADIUS PER DEGREE/24X,7F12,5)
AEROD305
300 FORMAT (///,25X,1MM ADDITIONAL LOADING,24X,24H WITH CL BASED ON SITRUE AEROD307
17X,11H AT CL DES-67X,34LOAD DUE ADD. LOAD AT BASIC LOAD3X,27AEROD308
02HSPAN LOAD AT X LOCATION OF/S8 STATIONS,5H 2Y/B9SX,9H SL COEF. ,4X AEDRD309
03,8HCL RATIO4X,7H RATIO,7X,14HT0 TWIST CM=,9F.5,3X,7HMAT CL=05X,2AE AEROD310
46HOSCIPED CL LOCAL CEN PR/) AEROD311
310 FORMAT (/4,47X,Al,32H PLANFORM SPAN LOAD DISTRIBUTION ,/)
AEROD312
320 FORMAT (4X,14,F12.5,5X,3F12.5,3X,2F12.5)
AEROD313
330 FORMAT (///,54X,24H MCQ AND CLQ ARE COMPUTED ,/)
AEROD314
340 FORMAT (///,38X,57H Static LONGITUDINAL AERODYNAMIC COEFFICIENTS ARE CALROD315
&1MPUTED ,/)
AEROD316
350 FORMAT (///,59X,15H CLP IS COMPUTED ,/)
AEROD317
360 FORMAT (/20X,1HX,11X,1HX,11X,1HX,11X,1HX,12X,1HS,5X,9HC/4 SWEEP,4X AEROD318
  16x46DERAL,2X,11X LOCAL ALPHA,2X,19HDELA CP AT DESIRED/19X,3HC/4, AEROD319
  29X,4HC/4,142X,9HANGLE,7X,9HANGLE,4X,10HIN RADIANS,4X,4MCL =F10.3 AEROD320
  3)
370 FORMAT (12X,9F12.5) AEROD321
380 FORMAT (/1X, 1X, A10, 6HPLANFORM, 5X, 5F12.5) AEROD325
END AEROD326

SUBROUTINE FLOWFL
COMMON /ALL/ BOT, 90TSV(4), M, BETA, PTEST, QTEST,
  $ STA(4), TBLSCW(10), YYCP(4),
  $ Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),
  $ PHI(100), ZH(100), CP(400), STLIND(4)
  \$ ALL 2
  \$ ALL 3
  \$ ALL 4
  $ ALL 5
C
COMMOM /MAINONE, ICODEOF, TOTAL, AAN(4), XS(4), YS(4), KFCTS(4),
  $ XREG(25,4), YREG(25,4), AREG(25,4), DIH(25,4), MCD(25,4),
  $ XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),
  $ ZZ(25,4), ITIPCOD, ICAMTS
  \$ MAINONE2
  \$ MAINONE3
  \$ MAINONE4
  $ MAINONE5
C
COMMON /CNETHE/ T4IST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,
  $ ARTRUE, RTOCMT(4), CONFIG(2), NSSSV(4),
  $ HSV(4), KBDT, PLAN, IPLAN, MACH,
  $ SSSWWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),
  $ CMAR(4), CLWIN(4), CLWNG(4), XLOCTN,
  $ YINER(4), YOUTER(4)
  \$ ONETHRE2
  \$ ONETHRE3
  \$ ONETHRE4
  $ ONETHRE5
C
INTEGER CONFIG
C
COMMON /TOTHREE/ CIR(400,2)
C
COMMON /THREFDOR/ CCAV2(2,100), CLT, CLNT, NSSW, ALPD
C
COMMON /HEAP/ BREAKPT(75,9), INDEXL, INDEXR, NUMSTA, NUMSTOT
C
DIMENSION YFL(75), XCVLE(75), XCVTE(75)
EQUIVALENCE (BREAKPT(1,1), YFL(1)),
$ (BREAKPT(1,2), XCVLE(1)),
$ (BREAKPT(1,3), XCVTE(1))
C
COL1 COL2 COL3 COL4 COL5 COL6 COL7 COL8 COL9
YFL XLE1 XTE1 XLE2 XTE2 XLE3 XTE3 XLE4 XTE4
C
COMMON /INSUB23/ AP5I, APHI, XXX, YYY, ZZZ, SNN, TOLC
LOGICAL SKIPIT
DIMENSION GAMP(400,4), YA(2), FV(2), FW(2), FU(2)
LOGICAL OFFPLAN(75)

FLOWFL2
FLOWFL3
FLOWFL4
FLOWFL5
FLOWFL6
FLOWFL7
FLOWFL8
FLOWFL9
FLOWFL10
FLOWFL11
FLOWFL12
FLOWFL13
FLOWFL14
DIMENSION M(4), NUM(4)
DIMENSION INDEXES(400,2), YFLOW(75), XFLOW(75)

REAL MACH
DATA NUMBER/5HFIRST,5HSECOND,5HTHIRD,5HFORTH/

NUMSTA = 75
NUMSLOT = 75

DO 15 J = 1,9
15 CONTINUE

DO 10 I = 1, NUMSTA
3REAKPT(I,J) = 0.
10 CONTINUE

TOLC = (80T * 15.E - 05) ** 2

WRITE (6,630)
WRITE (6,620)

PI = 4. * ATAN(1.1)
RAD = 180. / PI
CONST = 4. * PI

ALPW AND ALPD USED HEREIN IN RADIANS

ALPW = 1. / CLNT
ALPD = -(CLT / CLNT)
ALPW = ALPW + CLDES
ALPD = ALPD + ALPD

GAMP(J,1) IS THE BASIC LOAD INITIALLY
GAMP(J,2) IS THE ADDITIONAL LOAD AT CL = 1.
GAMP(J,3) IS THE TOTAL LOAD AT CLDES
GAMP(J,4) IS THE ADDITIONAL LOAD AT CLDES

DO 20 I = 1,M
GAMP(I,1) = CIR(I,1) + CIR(I,2) * (- CLT / CLNT)
GAMP(I,2) = CIR(I,2) / CLNT
20 CONTINUE

DO 30 K = 1,M
GAMP(K,4) = GAMP(K,2) * CLDES
GAMP(K,3) = GAMP(K,1) + GAMP(K,4)
30 CONTINUE
WRITE (6, 640) CONFIG
WRITE (6, 670) CLDES, CLDES
C
NN = 1
LMA = TBLSCW(1)
M1M(1) = 1
M1M(2) = 1 * MSV(1)
M1M(3) = 1 * MSV(1) + MSV(2)
M1M(4) = 1 * MSV(1) + MSV(2) + MSV(3)
DO 60 I = 1, NsSW
J = NN
J1 = J + 1
C
DO 40 IT = 1, ITPLAN
IF (J .EQ. M1M(IT)) WRITE (6, 720) NUMBER(IT)
CONTINUE
WRITE (6, 680) I(J), GAMP(J, 1), GAMP(J, 2), GAMP(J, 3), GAMP(J, 4)
C
IF (TBLSCW(I), E0, 1) GO TO 50
C
WRITE (6, 690) (GAMP(L, 1), GAMP(L, 2), GAMP(L, 3), GAMP(L, 4), L = J1, LMA)
C
IF (I .EQ. NsSW) GO TO 60
LMA = TBLSCW(I) + LMA
50 NN = TBLSCW(I) + NN
60 CONTINUE
C
LL = 1
MM = 0
K = 0
C
DO 80 ITT = 1, ITPLAN
JANGE = NsSWSV(ITT)
DO 70 J = 1, JANGE
K = K + 1
MM = MM + TBLSCW(K)
INDEXES(K, 2) = MM
XCV(E(K)) = .5 * BETA * (3 * P(L) - P(L))
XCV(E(K)) = .5 * BETA * (3 * P(M) - P(M))
INDEXES(K, 1) = LL
YFL(K) = Q(LL)
LL = LL + TBLSCW(K)
70 CONTINUE
80 CONTINUE
C
SORT THE VALUES IN -YFL- AND REMOVE THE DUPLICATES. NOTE
FOR DUPLICATES, THEIR XLE AND XTE VALUES ARE STORED IN THE
NEXT AVAILABLE COLUMNS OF THE -BREAKPT- MATRIX
C
FLOW111
NUMSTA = K
CALL HEAPSRT
NUMPTS = NUMSTA
REMAIN = NUMSTOT - NUMPTS
DELTAY = 2.*BOT/REMAIN
READ IN TOTAL NUMBER OF FIELD LINES
READ(5,610) TOTFL
NTOTFL = TOTFL
DO 160 NTOT = 1,NTOTFL
RSTA = NTOT
SKIPIT = .FALSE.
FIELD LINE DESCRIPTION
XDOWN IS THE X LOCATION AT THE PLANE OF SYMMETRY
SWEP IS THE SWEEP ANGLE IN DEGREES
ZREF IS THE Z LOCATION AT THE PLANE OF SYMMETRY
DIHED IS THE DIHEDRAL ANGLE IN DEGREES
READ (5,610) XDOWN,SWEP,ZREF,DIHED
XDOWN = XDOWN-XLOCXY
TANFL = TAN(SWEP / RAD)
TDIHED = TAN(DIHED / RAD)
WRITE (6,600)
ALPWW = ALPD * RAD
WRITE(6,660) RSTA,XDOWN,SWEP,ZREF,DIHED,ALPWW,CLDES
WRITE (6,700)
NOTE THAT THE PLANFORM OF MAXIMUM
SEMI-SPAN MUST EXTEND TO THE PLANE
OF SYMMETRY.
DO 110 I = 1,NUMPTS
OFFPLAN(I) = .TRUE.
YFLOW(I) = YFL(I)
XFLOW(I) = YFL(I) * TANFL + XDOWN
DO 100 J=1,4
   JT2 = J * 2
   JT2P1 = J + 2 + 1
   XLE = BREAKPT(I,JT2)
   XTE = BREAKPT(I,JT2P1)
   IF(ABS(XLE) .LT. ABS(XFLOW(I))) .AND.
      (ABS(XFLOW(I)) .LT. ABS(XTE))) OFFPLAN(I) = .FALSE.
   $  
100 CONTINUE
110 CONTINUE
115 J = 0
   K = NUMPTS + 1
   DO 120 I = K,NUMSTOT
      J = J + 1
      OFFPLAN(I) = .TRUE.
      YFLOW(I) = -BOT - J * DELTAY
      XFLOW(I) = YFLOW(I) * TANFL + XDOWN
   120 CONTINUE
C
C
C
DO 150 II = 1,NUMSTOT
   IF (OFFPLAN(II)) GO TO 125
   ZFL = ZREF + YFLOW(II) * TDIMED
   WRITE(6,730) II, XFLOW(II), YFLOW(II), ZFL
   SKIPIT = .TRUE.
   GO TO 150
125 WOU = 0.
   ZFL = ZREF + YFLOW(II) * TDIMED
   UOU = 0
   VOU = 0
   DWUADA = 0
   DUADA = 0
   XFLOW(II) = XFLOW(II) / BETA
   IZ = 1
   NNN = TBSLCW(II)
   DO 140 NN = 1,M
      SNN = S(INN)
      XXX = XFLOW(II) - PNN(NN)
      ZZZ = ZFL - ZH(IZ)
      APHI = ATAN(PHI(IZ))
      APSI = PSI(NN)
      YA(1) = YFLOW(II) - J(NN)
      YA(2) = YFLOW(II) + Q(NN)
   140 CONTINUE
   DO 130 I = 1,2
      FLOWF162
      FLOWF163
      FLOWF164
      FLOWF165
      FLOWF166
      FLOWF167
      FLOWF168
      FLOWF169
      FLOWF170
      FLOWF171
      FLOWF172
      FLOWF173
      FLOWF174
      FLOWF175
      FLOWF176
      FLOWF177
      FLOWF178
      FLOWF179
      FLOWF180
      FLOWF181
      FLOWF182
      FLOWF183
      FLOWF184
      FLOWF185
      FLOWF186
      FLOWF187
      FLOWF188
      FLOWF189
      FLOWF190
      FLOWF191
      FLOWF192
      FLOWF193
      FLOWF194
      FLOWF195
      FLOWF196
      FLOWF197
      FLOWF198
      FLOWF199
      FLOWF200
      FLOWF201
      FLOWF202
      FLOWF203
      FLOWF204
      FLOWF205
      FLOWF206
      FLOWF207
      FLOWF208
      FLOWF209
      FLOWF210
YYY = YA(I)
CALL INFSUB (BOT, FV(I), FW(I), FU(I))
APHI = -APHI
APSI = -APSI

130 CONTINUE
WOU = WOU + (FW(1) + FW(2)) * GAMP(NN,3) / CONST
VOU = VOU + (FW(1) + FW(2)) * GAMP(NN,3) / CONST
UOU = UOU + (FU(1) + FU(2)) * GAMP(NN,3) / CONST
DWOUDA = DWOUDA + (FW(1) + FW(2)) / (CONST * ALPW)
DUUDA = DUUDA + (FU(1) + FU(2)) / (CONST * ALPW)
IF ((NN .LT. NN .OR. NN .EQ. M) GO TO 140
IZ = IZ + 1
NN = NNN + TBLSW(IZ)

140 CONTINUE
CA = COS(ALPD)
SA = SIN(ALPD)
EPSILN = ATAN(WOU * CA - UOU * SA) / (1. - WOU * SA - UOU * CA)
1) * RAD
DEPDAL = (WOU ** 2 + UOU ** 2 - (WOU + DUUDA) * SA + (DWOUA - UOUFLW230)
1U) * CA + WOU + DUUDA - DUUDA / (1. - 2. * (WOU + SA) + UOUFLW230)
2U + CA) * UOU ** 2 + WOU ** 2)
ORATI = 1. - 2. * (UOU + CA + WOU) + UOU ** 2 + VGU ** 2 + WOFLW232
1U ** 2
SMACH = MACH * MACH
ORATI = (1. + 2 * SMACH * (1. - ORATI)) ** 2.5

C
WRITE(6,710)II,XFLOW(ll),YFLOW(ll),ZFLW,VOU,OUU,DWNWH,EPSTIN
1,DEPDAL,ORATI,SIGMA

150 CONTINUE
IF (SKIPIT) WRITE(6,740)

160 CONTINUE
RETURN

C
610 FORMAT (8F10.6)
620 FORMAT (1H1)
630 FORMAT (1H1,150X,8H4HFLO FIELD I) FLOWF230
640 FORMAT (1H1,24X,16HCONFIGURATION I) FLOWF251
650 FORMAT (1H1,36X,15HFIELD LINE DATA/2OX,10HFIELD LINE,4X,1H1X,5X) FLOWF250
13H5X,EEDEGREES,5X,1H1X,4X,16HDIHEDRAL=DEGREES,2X,13HALPHA=DEGREES,FLW255
2,3X,7X,1H2)
660 FORMAT (15X,F12.0,4F12.5,6X) FLOWF235
670 FORMAT (15X,F12,7HSTATION,4X,3H Y,26X,7HGMMA/A/5X,41HBASIC ADDITION FLOWF257
1NAL TOTAL ADDITIONAL/48X,31HAT CL = 0. AT CL = 1.0 AT CLFLOWF258
2,F5.2,2X,7HAT CL,*,F5.2)

FLOWF211
680 FORMAT (29X,15,5F12.5) FLOWF260
690 FORMAT (46X,4F12.5) FLOWF261
700 FORMAT (//,8X,8H DOWNWATCH/,1X,5H FIELD,6X,1HX,11X,1HY,11X,1HZ,11X,X,1F7.4)
710 FORMAT (1X,15,6F12.5,5F11.5) FLOWF262
720 FORMAT (//,42X,32H INPUT VALUES OF CIRCULATION FOR A10,
* 8H PLANFORM//) FLOWF263
730 FORMAT (1X,15,6F12.5,12X,5H***** ) FLOWF264
740 FORMAT (1H0,5X,2EH***** - COMPUTATION OMITTED ,
$ 41H DUE TO THE POINT PROJECTION Lying WITHIN 
$ 26H THE CONFIGURATION BOUNDARY ,//) FLOWF265

C END
SUBROUTINE HEAPSRT

C THIS ROUTINE PERFORMS A HEAPSORT ON THE VALUES IN
C THE YFL ARRAY, AND MOVES THE CORRESPONDING XLE AND XTE
C VALUES WHEN A SWAP IS MADE. THIS ALGORITHM IS DESCRIBED
C IN M. WIRTH'S "ALGORITHMS + DATA STRUCTURES = PROGRAMS"
C BOOK, 1976, PRENTICE-HALL, INC.

C COMMON / HEAP / BREAKPT(75,9), INDEXL, INDEXR, NUMSTA, NUMSTOT

C DIMENSION YFL(75), XCVELE(75), XCVTE(75)
C EQUIVALENCE (BREAKPT(1,1),YFL(1)),
C $ (BREAKPT(1,2),XCVELE(1)),
C $ (BREAKPT(1,3),XCVTE(1))

C COL1 COL2 COL3 COL4 COL5 COL6 COL7 COL8 COL9
C YFL XLE1 XTE1 XLE2 XTE2 XLE3 XTE3 XLE4 XTE4

C N = NUMSTA
INDEXL = ( N / 2 ) + 1
INDEXR = N
10 CONTINUE
IF ( INDEXL .LE. 1 ) GO TO 20
INDEXL = INDEXL - 1
CALL SIFT
GO TO 10

20 CONTINUE
IF ( INDEXR .LE. 1 ) GO TO 30
Y1 = YFL(1)
X1 = XCVTE(1)
X2 = XCVAE(1)
YFL(I) = YFL(INDEXR)
XCVAE(I) = XCVAE(INDEXR)
XCVAE(I) = XCVAE(INDEXR)
YFL(INDEXR) = Y1
XCVAE(INDEXR) = X1
XCVAE(INDEXR) = X2
INDEXR = INDEXR - 1
CALL SIFT
GO TO 20
30 CONTINUE
C REMOVE THE DUPLICATE -YFL- VALUES BY COMBINING THEIR
C XLE AND XTE VALUES IN -BREAKPT-
C
N = 1
KK = 4
TEMP1 = YFL(1)
DO 50 J = 2, NUMSTA
   IF(TEMP1 .EQ. YFL(J)) GO TO 40
   KK = 4
   N = N + 1
   YFL(N) = YFL(J)
   XCVAE(N) = XCVAE(J)
   XCVAE(N) = XCVAE(J)
   TEMP1 = YFL(J)
   GO TO 50
40 CONTINUE
BREAKPT(N, KK) = XCVAE(J)
BREAKPT(N, KK+1) = XCVAE(J)
GO TO 50
50 CONTINUE
NUMSTA = N
RETURN
END
SUBROUTINE SIFT
COMMON / HEAP / BREAKPT(75, 9), INDEXL, INDEXR, NUMSTA, NUMSTOT
C
DIMENSION YFL(75), XCVAE(75), XCVAE(75)
EQUivalence (BREAKPT(1,1), YFL(1)),
$ (BREAKPT(1,2), XCVAE(1)),
$ (BREAKPT(1,3), XCVAE(1))
C
COL1 COL2 COL3 COL4 COL5 COL6 COL7 COL8 COL9
YFL XLE1 XTE1 XLE2 XTE2 XLE3 XTE3 XLE4 XTE4
C
INDEX = INDEXL
INDEX = 2 * INDEX
Y1 = YFL(INDEX)
X1 = XCVAE(INDEX)
X2 = XCVTE(INDEXI)
10 CONTINUE
   IF (INDEXJ .GT. INDEXR) GO TO 30
   IF (INDEXJ .EQ. INDEXR) GO TO 20
      JP1 = INDEXJ + 1
      ARSYJ = ABS(YFL(INDEXJ))
      ARSYJP1 = ABS(YFL(JP1))
   IF (ARYSJ .LT. ARSYJP1) INDEXJ = JP1
20 CONTINUE
   IF (ARSYJ .LT. ARSYJP1) GO TO 30
      YFL(INDEXI) = YFL(INDEXJ)
      XCVLE(INDEXI) = XCVLE(INDEXJ)
      XCVTE(INDEXI) = XCVTE(INDEXJ)
   IF (INDEXJ .EQ. INDEXR) INDEXJ = INDEXR
   INDEXJ = INDEXJ + 2
   GO TO 10
C
30 YFL(INDEXI) = Y1
   XCVLE(INDEXI) = X1
   XCVTE(INDEXI) = X2
   RETURN
END

SUBROUTINE FTLUP (X, Y, M, N, VARI, VARD)
C
***DOCUMENT DATE 09-12-69 MODIFICATION OF LIBRARY INTERPOLATION SUBROUTINE FTLUP***
DIMENSION VARI(1), VARD(1), V(3), YY(2)
DIMENSION II(43)
C
C INITIALIZE ALL INTERVAL POINTERS TO -1.0 FOR MONOTONICITY CHECK
C
DATA (II(IJ)), J = 1, 43) / 43 * -1/
MA = 1ABS(M)
C
C THE SAME POINTER WILL BE USED ON A GIVEN VARI TABLE
C
LI = MOD(LCFL(VARI(1)), 43) + 1
I = II(LI)
IF (I.GE.0) GO TO 50
IF (N.LT.2) GO TO 50
C
C MONOTONICITY CHECK
C
IF (VAPI(2) - VARI(1)) 20, 20, 40
C
ERRORS IN MONOTONICITY
C
K = LCFL(VARI(1))
POINT 17G, J, X, (VARI(J), J = 1, N), (VARD(J), J = 1, N)
STOP
C
C MONOTONIC DECREASING
C
20 DO 30 J = 2, N
   IF (VARI(J) - VARI(J-1)) 30, 10, 10
30 CONTINUE
GO TO 60
C
60 DO 50 J=2,N
IF (VARI(J)-VARI(J-1)) 10,10,50
50 CONTINUE
C
C INTERPOLATION
60 IF (I.LE.0) I=1
IF (I.GE.N) I=N-1
IF (N.LE.1) GO TO 70
IF (M.A.NE.0) GO TO 80
C
70 Y=VARD(I)
GO TO 160
C
LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
80 IF ((VARI(I)-X)*(VARI(I+1)-X)) 110,10,90
C IN GIVES DIRECTION FOR SEARCH OF INTERVALS
90 IN=SIGN(1.0, (VARI(I+1)-VARI(I))*(X-VARI(I)))
C IF X OUTSIDE ENDPONT, EXTRAPOLATE FROM END INTERVAL
100 IF ((I+IN).LE.0) GO TO 110
IF ((I+IN).GE.N) GO TO 110
I=I+IN
110 IF ((VARI(I)-X)*(VARI(I+1)-X)) 110,110,100
110 IF (M.A.EQ.2) GO TO 120
C
C FIRST ORDER
Y=(VAPD(I)*((VARI(I)+X)-VARD(I+1))*(VARI(I)-X))/(VARI(I+1)-VARI(I))
1)
GO TO 160
C
C SECOND ORDER
120 IF (N.EQ.2) GO TO 10
IF (I.EQ.(N-1)) GO TO 140
IF (I.EQ.1) GO TO 130
C
C PICK THIRD POINT
130 L=I
GO TO 150
140 L=I-1
150 V(1)=VARI(L)-X
V(2)=VARI(L+1)-X
V(3)=VARI(L+2)-X
YY(1)=(VAPD(L)*V(2)-VARD(L+1)*V(1))/(VARD(L+1)-VARI(L))
YY(2)=(VAPD(L+1)*V(3)-VARD(L+2)*V(2))/(VARD(L+2)-VARI(L+1))
Y=(YY(1)*V(3)-YY(2)*V(1))/(VARI(L+2)-VARI(L))
160 I=I+1
RETURN
C
FORMAT (1HI,50H TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION FTLUP178
170 /13W X TABLE IS STORED IN LOCATION 00B,//(18G15.B))
179 FTLUP180
END
FTLUP181
SUBROUTINE CDICL5 (AR,ARTRUE,ISEMSP,MTOT,NST,NSV,CDI,COI)
CDICL5 2
DIMENSION ETAN(51), GAMPR(51,1), ETA(41), GAMMA(41), VE(41), B(41)
CDICL5 3
1, FVN(41,41)
CDICL5 4
COMMON ALL, BOT, BOTSV(4), M, BETA, PTEST, QTEST,
ALL 2
$ STA(4), TBLSCW(100), YYCP(4),
ALL 3
$ Q(400), PN(400), PV(400), ALP(400), SI(400), PSI(400),
ALL 4
$ PHI(100), ZH(100), CP(400), STLQIND(4)
ALL 5
COMMON /THREC1/ SLOAD(3,100)
CDICL5 6
DO 10 I=1,41
DO 10 J=1,41
10 FVN(I,J)=0
CDICL5 9
SPAN=2.*BOT
CDICL5 10
CAVB=SPAN/ARTRUE
CDICL5 11
PI=3.14159265E+01
CDICL5 12
NST=ISEMSP+1
CDICL5 13
NN=MTOT
CDICL5 14
DO 20 N=1,ISEMSP
CDICL5 15
NM=NSV+N
CDICL5 16
NSCW=TBLSCW(NM)
CDICL5 17
NN=NN-NSCW
CDICL5 18
ETAN(N)=ASIN(-0.0)*Z/SPAN)
CDICL5 19
GAMPR(N,1)=SLOAD(3,NM)*CAVB/(2.*SPAN)
CDICL5 20
CONTINUE
CDICL5 21
ETAN(NST)=PI/2.
CDICL5 22
GAMPR(NST,1)=0
CDICL5 23
DO 30 NP=1,41
CDICL5 24
ANP=NP
CDICL5 25
30 ETAN(NP)=(ANP-21.)*PI/42.
CDICL5 26
C
DO 40 JK=21,41
CALL FTLUP (ETA(JK),GAMMA(JK),1,NST,ETAN,GMAPR)
CDICL5 27
40 CONTINUE
CDICL5 28
DO 50 NY=22,41
CDICL5 29
ETA(NY)=SIN(ETA(NY))
CDICL5 30
NP=42-NY
CDICL5 31
ETAN(NP)=ETA(NY)
CDICL5 32
GAMMA(NP)=GAMMA(NY)
CDICL5 33
DO 50 NU=21,41
CDICL5 34
ANU=NU
CDICL5 35
DO 50 N=1,41
CDICL5 36
CDICL5 37
AN=N
CDICL5 38
NNUD=IA85(N-NU)
CDICL5 39
VE(N)=COS((AN-21.)*PI/42.)
CDICL5 40
IF (NNUD NE 0) GO TO 60
CDICL5 41

B(N)*(42.)*(4.0*COS(11*PI)/42.)
GO TO 80
60 IF(MOD(NNUD,2),EQ.0)GO TO 70
B(N)*VE(N)/(42.)*(ETA(N)-ETA(NU))**2
GO TO 80
70 B(N)*1.0
80 CONTINUE
DO 90 NP=21,41
NUST=IABS(NU-21)
IF(NUST.EQ.0)GO TO 90
IF(MOD(NUST,2),EQ.0)GO TO 90
NPST=IABS(NP-20)
IF(MOD(NPST,2),EQ.0)GO TO 90
NPNUD=IABS(NP-NU)
IF(NPNUD.EQ.0)GO TO 90
IF(MOD(NPNUD,2),EQ.0)GO TO 90
FVN(NU,NU)*2.0*8(NP)/21.*COS((ANU-21.)*PI/42.)
IT=42-NU
ITT=42-NP
FVN(NU,ITT)*2.0*8(ITT)/21.*COS((ANU-21.)*PI/42.)
FVN(ITN,NU)*FVN(NU,ITT)
FVN(IT,ITT)*FVN(NU,NP)
90 CONTINUE
CCC=0.0
DO 110 N=1,41
CCC=CCC+(GAMMA(N)*GAMMA(N))
CCD=0.4
DO 110 NUP=1,41
DO 110 N=1,41
CCD=CCD-2.0*FVN(NUP,N)*GAMMA(NUP)*GAMMA(N)
110 CONTINUE
COD=(2.0*AR/PI)**(CCC+CDC)
CODIT=1./((PI*AR)
RETUN
END
SUBROUTINE CDAGNF
DIMENSION GAM(1100),XCC(1100),YQ(1100),CCR(40)
$ FW(1),PV(1),XCC(140),CCC(400),CRR(400)
$ YB(1),CR(1),MA(4),XCC(400),ACHI(100)
$ XCC(100),YY(3),PH(100),Z(1100)
$ PH1(100),SA(100),SSA(1100),ALU(400),ALLP(100)
$ APD(1100),APD(400),YP(102),YQ(1100),BOLT(4),SUMS(4),
$ SUMSV(4),SUMS(4),NUMBER(4)
COND, ALL, POT, ROTS(4), M, HETA, PIEST, GTES,
$ STA(4), T1LSCw(100), YC(4),
$ 4000, 4000, 4000, 4000, 4000, 4000,
$ PHI(100), ZH(100), CP(400), STLDQ(4)
C
COMMON /ONETHREE/ TWIST(4), CREF, SRC, CAVE, CLDES, STRUES, AR,
$ SARTH, RICDMHT(4), CONCFG(2), NSSWSV(4),
$ MSV(4), KBOT, PLAN, IPLAN, MACH,
$ SSW(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),
$ CLAR(4), CLWIN(4), CLWNG(4), XLOCTN,
$ YINNER(4), YOUTER(4)

C

INTEGER CONFG

C

COMMON /TOTHRE3/ CIR(400;2)

C

COMMON /INSUB23/ APSI, APHI, XX, YYY, ZZ, SNN, TOLCSO

C

COMMON /THERFOR/ CCAV(2,100), CLT, CLNT, NSSW, ALPD

C

COMMON/CCRDDD/ TSPAN(4), TSPANA, KBIT, CTIlda, XTILDA, DISTALE

C

DATA NUMER/HFIRST,6HSECOND,5HTHIRD,6HFORTH/

C

WRITE (6,250)

APSI=TOLCSO*TBL$=0.

PI=4.*ATAN(1.,1.)

PI=4.*PI

DO 5 IT=1,IPLAN

BOTL(IT) = ABS(TSPAN(IT))

5 CONTINUE

SNN = BOTL(KBOT) / ( 2. * NSSWSV(KBOT))

DELTB = 2.*SNN

NSMS = 0

NSMS = 0

NMAX = 0

DO 7 IT=1,IPLAN

NMS(IT) = BOTL(IT) / DELTB

NMS = NSMS + NSSWSV(IT)

NMS = NSMS + NSSWSV(IT)

NMS = NSMS + NSSWSV(IT)

NMS = NSMS + MSSMS(IT)

NMS = NSMS + MSSMS(IT)

NMS = NSMS + MSSMS(IT)

NMS = NSMS + MSSMS(IT)

NMS = NSMS + MSSMS(IT)

7 CONTINUE

NMAX = NMSMS(IPLAN)

DO 10 I=1,N

CRRI(I)=CIR(I,1)+CIR(I,2)*(CLDES-CLT)/CLNT

10 CONTINUE

SCWMIN=20.

DO 20 I=1,NSSW

20 SCWMIN=MIN(SCWMIN,TBLSCW(I))

MSCWMIN=SCWMIN

C

ONETHREE2

ONETHREE3

ONETHREE4

ONETHREE5

ONETHREE6

ONETHREE7

ONETHREE8

ONETHREE9

ONETHREE10

TOTHREE 2

TOTHREE 3

CORAGN13

THREFOR2

THREFOR3

CCRRDD 2

CCRRDD 3

CCRRDD 4

CCRRDD 5

CCRRDD 6

CCRRDD 7

CCRRDD 8

CCRRDD 9

CCRRDD 10

CCRRDD 11

CCRRDD 12

CCRRDD 13

CCRRDD 14

CCRRDD 15

CCRRDD 16

CCRRDD 17

CCRRDD 18

CCRRDD 19

CCRRDD 20

CCRRDD 21

CCRRDD 22

CCRRDD 23

CCRRDD 24

CCRRDD 25

CCRRDD 26

CCRRDD 27

CCRRDD 28

CCRRDD 29

CCRRDD 30

CCRRDD 31

CCRRDD 32

CCRRDD 33

CCRRDD 34

CCRRDD 35

CCRRDD 36

CCRRDD 37

CCRRDD 38

CCRRDD 39

CCRRDD 40

CCRRDD 41

CCRRDD 42

CCRRDD 43

CCRRDD 44

CCRRDD 45

CCRRDD 46

CCRRDD 47

CCRRDD 48
MN=NSCWMIN*NMAX
DELTXOC=1./SCWMIN
DO 100 LA=1,NSSW
CHO(JA)*CCAV(2,LA)*CAVE/BETA
DELTXX=1./TBLSCW(LA)
XG=+.75*DELTXX
ITBL=TBLSCW(LA)
DO 30 LB=1,ITBL
XG=XG*DELTXX
XXCC(LB)=XG
LC=L4+TBLS
30
ALO(LB)=ALP(LC)
XLE=NL(LC)*CHO(LA)*(1.-.75/TBLSCW(LA))
XGC=+.75*DELTXOC
KCODE=LB=0
DO 90 K=1,NSCWMIN
J=L(LA)-1-NSCWMIN
XGC=XGC+DELTXOC
XCC(J)=XGC*CHO(LA)*XLE
CALL FILLP (XGC,ALO(J)+1,ITBL,XXCC,ALO)
AXMN=K*DELTXOC
CAT=0.
IF (KCODE.EQ.2) CAT=CCCR(LB)-CUT
KCODE=0.
40
LB=L8+1
LC=L8+TBLS
CCP(LB)=CCP(LC)
AXITBL=LB*DELTXX
IF (AXMN-AXITBL) 50,60,70
50
CUT=CCP(LB)*(AXMN-(LB-1)*DELTXX)/DELTXX
*CODE=2
GO TO 80
60
KCODE=1
70
CUT=CCP(LB)
80
CAT=CAT+CUT
IF (KCODE.GE.1) GO TO 90
IF (LB-LT,ITBL) GO TO 40
90
CCP(J)=CAT
TBLS=TBLS+TBLSCW(LA)
100 CONTINUE
II=1
DO 150 I=1,IPLAN
IUZ=NSSV(I)
IUX=IUZ
I=IUZ+1
IF (STC(J).EQ.1.) IUX=IUZ
IC=PSL*SV(I)
IO=IC+1
II=NSSV(I)
YCAT=0.
CRAGN49
CRAGN30
CRAGN51
CRAGN52
CRAGN53
CRAGN54
CRAGN55
CRAGN56
CRAGN57
CRAGN58
CRAGN59
CRAGN60
CRAGN61
CRAGN62
CRAGN63
CRAGN64
CRAGN65
CRAGN66
CRAGN67
CRAGN68
CRAGN69
CRAGN70
CRAGN71
CRAGN72
CRAGN73
CRAGN74
CRAGN75
CRAGN76
CRAGN77
CRAGN78
CRAGN79
CRAGN80
CRAGN81
CRAGN82
CRAGN83
CRAGN84
CRAGN85
CRAGN86
CRAGN87
CRAGN88
CRAGN89
CRAGN90
CRAGN91
CRAGN92
CRAGN93
CRAGN94
CRAGN95
CRAGN96
CRAGN97
IAMD=MMA(I)
IF(I.EQ.1) GOTO 142
NST = NSUMSV(I-1)
NSTT = NSUMS(I-1)
GOTO 143
142 NST = 0
NSTT = 0
143 CONTINUE
DO 140 LA=1,NSCWMIN
   IF (STLOIND(I) .EQ. 1.) GO TO 144
   YC(I)=-PI/2.
   CR1(I)=0.
144 DO 120 J=1,IUZ
   L*J+1
   IF (STLOIND(I) .EQ. 1.) L = J
   LU = LA + (J-1 + NST) * NSCWMIN
   ALLP(J)=ALDP(LU)
   XC44(J)=XCC4(LU)
   CR1(L)=CCC(LU)
   IF (LA .NE. 1) GO TO 120
   JJ = J + NST
   ZH(JJ) = ZH(JJ)
   SA(J) = SSHA(JJ)
   PPHI(J) = PHI(JJ)
   YQQ(J) = 0(JJ)
   IL=II+TBLS(CW(JJ)
   IE=IUZ+J+1
   ITL=TBLS(CW(II)
   ID=ID+ITL
   IA=ID+ITL
   IF (IA,GT,1C) YCAT=YCAT-S(ID)
   IF (IA,GT,1C) GO TO 110
   YCAT=YCAT-S(ID)-S(IA)
110 IZ=II-1
   YB(I)=YCAT
120 CONTINUE
DO 130 JP=1,IUZ
   JZ=JP+1
   IF (STLOIND(I) .EQ. 1.) JZ = JP
   YC(JZ) = ASIN(YB(JP)/BOLT(I))
130 CONTINUE
Y08=MMA(I)2.*SNN-SNM
DO 140 K=1,IAMD
   KP = LA + (K-1 + NSTT) * NSCWMIN
   Y08=Y08+DELYR
   Y0C = ASIN(Y08/BOLT(I))
   CALL FTLUP (Y08,Y0(KP),+1,IUZ,YB,Y0C)
   CALL FTLUP (Y08,ALPDP(KP),+1,IUZ,YB,ALLP)
   CALL FTLUP (Y08,SSA(KP),+1,IUZ,YB,SA)
   CALL FTLUP (Y08,SSA(KP),+1,IUZ,YB,SA)
CALL FTLUP (YOB, X4(KP), +1, IU2, YB, X44)
CALL FTLUP (YOB, Z(KP), +1, IU2, YB, ZH)
CALL FTLUP (YOB, PHI(KP), +1, IU2, YB, PP)
CALL FTLUP (YOB, GAM(KP), +1, IU2, YC, CRI)
IF (YOB.GT.YB(IU2)) GOTO 225
GAM(KP) = CPI(IU2)
CONTINUE

140 CONTINUE
CORAG = CTHRST * CSUCT * O,
CONST = 14 * SNN * BOT / SREF
DD 190 LI = 1, NMAX
LA = (LI-1)*NSWXM=1
LB = LI * NSWXM
CDAGIT = CTT * O,
DO 190 NV = LA, LB
CPT = COS(ATAN(PHI(NV)))
VELIN = O,
DO 170 NN = 1, MM
XX = XC4(NV) - XC4(NN)
YY(NV) - YO(NV)
YY(NV) - YO(NV)
ZZ(NV) - 2(NN)
APHI = ATAN(PHI(NV))
DO 160 I = 1, 2
YII = YY(NV)
call INF SUB (BOT, FV(I), FW(I), FUI)
APHI = APH
160 CONTINUE
VELIN = ((FW(I) + FW(I)) - (FV(I) + FV(I)) * PHI(NV)) * GAM(NV) / FPI + VELIN
170 CONTINUE
CTT = CTT + GAM(NV) * (ALP/57.29578 + ALPD(NV)) * CPT / (2. * BDT)
180 CORAGIT = CORAGIT + VELIN + GAM(NV) * CPT / (2. * BDT)
CTT = CTT + CORAGIT
SWLE = ATAN(SSF, LA)
CST = CTT * COS(SWLE)
CCC(LI) = CORAGIT
CRR(LI) = CST
XC4(LI) = CST
190 CONTINUE
TABLE = I * 0
LI = O
LNR = 0
DO 220 I = 1, IPLAN
LAM = NH(NI)
IF (NN(N) = 0) GOTO 225
NST = NSUM(NV) * (I - 1)
NSTT = NSUMS(NV) + (I - 1)
GOTO 226

225 NST = 0
NSTT = 0
GOTO 226

226 CONTINUE
DO 200 J=1,IAMM
J = J + NSTT
LA = 1 + (J-1 + NSTT) * NSCWMIN
GAM(J) = CCC(JJ)
XC4(J) = CRR(JJ)
Z(J) = XCC4(JJ)
PHII(J) = YO(LA)
200 CONTINUE

IUZ = NLSWV(I)
DO 210 LBLAIR = 1, IZU
LI = LI + 1
LU = L + TBL
LBLR = LBLR + 1
YR = Q(LU)
II = II + 1
TBL = TBL + TBSW(II)
YDBB = YBB/BDT
CALL FLPUP (YBB, CDRAGT + 1, IAMM, PHII, GAM)
CALL FLPUP (YBB, CTT + 1, IAMM, PHII, XC4)
CALL FLPUP (YBB, CST + 1, IAMM, PHII, Z)
LL = LBLAIR + NST
SWALE = ATAN (NSWA(LLL)) * 57.29578
DO 300 IT = 1, IPLAN
IF (IT .GE. 1) GOTO 310
MOD = 1
GOTO 320
310 MOD = NLSWV(IT-1) + 1
320 IF (II.EQ.MOD) WRITE (6, 240) NUMBER
300 CONTINUE
WRITE (6, 260) LI, YOBB, SWALE, CDRAGT, CTT, CST
LBLR = LBLR + CST
210 CONTINUE
220 CONTINUE
230 CDCL2 = CDRAG/CLDES**2
WRITE (6, 270) CDCL2, CTHRU, CSUCT
C
C
C
240 FORMAT (37X, 20HCONTRIBUTION OF THE A10,
* 35HPLANFORM TO THE CHORD OR DRAG FORCE)!
250 FORMAT (18X, 13HINDUCED DRAG, LEADING EDGE THRUST AND SUCTION CDRAG420
10EFFECTIVE CHARACTERISTICS/40X, 93HCOMPUTED AT THE DESIRED CL FROM CDRAG421
2A NEAR FIELD SOLUTION/59X, 20HSECTION COEFFICIENTS/48X, 11HL. E. SWCDRAG422
3EEP/25X, 7HSTATION, 9X, 5X 27/8, 5X, 3HANGLE, 7X, 9HCDII C/2B, 5X, 7HCT C/2CDRAG423
48, 5X, 7HCS C/28)
260 FORMAT (20X,110,5X,5F12.5)  
270 FORMAT (/10,5X,5F12.5)  
310 FORMAT (13HCT=90,7X,5F12.5)  
330 FORMAT (13X,5F12.5)  
350 FORMAT (13X,5F12.5)  
370 FORMAT (13X,5F12.5)  
390 FORMAT (13X,5F12.5)  
410 FORMAT (13X,5F12.5)  
430 FORMAT (13X,5F12.5)  
450 FORMAT (13X,5F12.5)  
470 FORMAT (13X,5F12.5)  
490 FORMAT (13X,5F12.5)  
510 FORMAT (13X,5F12.5)  
530 FORMAT (13X,5F12.5)  
550 FORMAT (13X,5F12.5)  
570 FORMAT (13X,5F12.5)  
590 FORMAT (13X,5F12.5)  
610 FORMAT (13X,5F12.5)  
630 FORMAT (13X,5F12.5)  
650 FORMAT (13X,5F12.5)  
670 FORMAT (13X,5F12.5)  
690 FORMAT (13X,5F12.5)  
710 FORMAT (13X,5F12.5)  
730 FORMAT (13X,5F12.5)  
750 FORMAT (13X,5F12.5)  
770 FORMAT (13X,5F12.5)  
790 FORMAT (13X,5F12.5)  
810 FORMAT (13X,5F12.5)  
830 FORMAT (13X,5F12.5)  
850 FORMAT (13X,5F12.5)  
870 FORMAT (13X,5F12.5)  
890 FORMAT (13X,5F12.5)  
910 FORMAT (13X,5F12.5)  
930 FORMAT (13X,5F12.5)  
950 FORMAT (13X,5F12.5)  
970 FORMAT (13X,5F12.5)  
990 FORMAT (13X,5F12.5)  

--- NOTES TO THE USERS ---

1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING
   EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF
   ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE
   CALCULATED FROM CLDES = 1.0, AND CLDES = 1.0

2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOMMENDED
   THAT THE WING BE INPUT AS TWO PLAINWORMS TO PROVIDE
   MORE MEANINGFUL SIDE EDGE RESULTS

3. STLOIND = "STREAMWISE LOAD INDICATOR" ARRAY; SET TO
   0. IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE
   EDGE OF THIS PLAINWORM IS TO BE 0.0; OTHERWISE, SET TO
   1.0 IF THIS LOADING IS TO BE NON-ZERO

COMMON /TOTHREE/ CIR(400,2)
COMMON /ONE/ THREED, CREF, SREF, CAVE, CLDES, STRUE, AR,
$  ARTRUE, RICHTH(4), CONFIG(2), NSSW, NY4(4),
$  MSV(4), KBOT, PLAN, IPLAN, MACH,
$  SSWA(100), XL(4), XT(4), CLWB, CMCL, CLA(4), BLAIR(100),
$  CLRPA(4), CLWIN(4), CLNG(4), XLOCIN,
$  TINNER(4), YUTER(4)

INTEGER CONFIG

COMMON /THREED/ CCAV(2,100), CLT, CLNT, NSSW, ALPD

COMMON /THREED/ CCAV(2,100), CLT, CLNT, NSSW, ALPD
COMMON /INSUB23/ APHI, XX, YYY, ZZ, SNN, TOLCS0, TIPSUC12
DIMENSION CLLL(26), CDDD(26), CMNM(26), ALPH(26), CDCLZ(26), CDCLF(26), TIPSUC13
1    CDCLV(26)
REAL KVSE(4)
DIMENSION CENTR(4), TIPSUM(4)
DATA XLEG / 60*0.0, 60*0.0/
DATA ZLEGW, CIRSUM, YLEGW
S = 200*0.0, 200*0.0, 200*0.0 /
J = 0
DO 1 ITT = 1, IPLAN
   CENTR(ITT) = 0.0
   KVSE(ITT) = 0.0
   TIPSUM(ITT) = 0.0
   IF(XL(ITT).EQ. XT(ITT)) J = J + 1
1 CONTINUE
C IF(J .EQ. IPLAN) GO TO 540
C
C BLMAR = 1.0/BETA
C NSSL = 0
C DD 2 ITT = 1, IPLAN
C XT(ITT) = XT(ITT) * BLMAR
C XL(ITT) = XL(ITT) * BLMAR
C NSSL = NSSL + NSSW(NSSWSV(ITT))
2 CONTINUE
C
C THE TOLERANCE SET AT THIS POINT IN THE PROGRAM MAY NEED TO BE
C CHANGED FOR COMPUTERS OTHER THAN THE CDC 6000 SERIES
C
C TOLC = .0100*ROT
C TOLCSO = TOLC + TOLC
C TIPSI = 0.0
C TIPSU = PITCH = 0.
C
C GEOMETRY FOR TIP TRAILING LEGS
C
C IM = 0
C IMM = 0
C NSSL = 0
C NSSW2 = 0
C EPS = 1.e-6
C CTWS = 0.0
C CMW = 0.0
C CCIRS = 0.0
C
C
DO 535 ITT = 1,IPLAN
C
NSCW = MSVIT1/IT1 / NSSWST(1,ITT)
NSSW2 = NSSW2 + NSSWST(1,ITT)
IF (ITT .EQ. IPLAN .AND. XL(ITT) .EQ. XT(ITT)) GO TO 479
20 I = IM + 1
IMP = 0
J = JIM + 2
IUU = 2
APHI = ATAN(PHI(I+1))
SA = SIN(APHI)
CA = COS(APHI)
TLX1 = PN(I) - S(I) * TAN(PHI(I)) * CA
TLX2 = PN(J) - S(J) * TAN(PHI(J)) * CA
CLFTLG = TLX1 - TLX2
XTLEG(I) = TLX1/2. + TLX2/2.
YLEG = Q(I) - S(I) * CA
YLEG(V1,1,ITT) = YLEG
ZLEG = ZH(I+1) - S(I) * SA
ZLEG(V1,1,ITT) = ZLEG
IF (XL(ITT), EQ, XT(ITT)) GO TO 100
DO 30 NV = 2, NSCW
NVT = NV - 1
30 XTLFVG(NV) = XTLLEG(NVT) - CLFTLG
NCLT = 0
NA = 1
NB = NSCW
40 DO 70 NV = NA, NB
C
THE RATIO OF W/U IS INITIALIZED TO -1 BECAUSE IN THE TERM
-1*ALPHA/U, USED IN THIS SUMMATION, ALPHAI IS SET TO 1 RADIUS
SO THAT THE RESULTING TIP SUCTION CAN BE USED DIRECTLY TO FIND
THE SIDE EDGE
C
WVOUN(NV) = -1.
IZ = 1
NN = TRLSWC(IZ)
DO 60 NN = 1, M
APHI = ATAN(PHI(IZ))
APSI = PSI(NN)
PX = XTLLEG(NV) - PN(NN)
PY(I) = YLEG - Q(NN)
PY(I) = YLEG - Q(NN)
ZI = ZLEG + ZH(IZ)
SN = S(NN)
DO 50 I = 1, 2
YYY = YY(I)
50 CONTINUE
C
TIPSUC61
TIPSUC62
TIPSUC63
TIPSUC64
TIPSUC65
TIPSUC66
TIPSUC67
TIPSUC68
TIPSUC69
TIPSUC70
TIPSUC71
TIPSUC72
TIPSUC73
TIPSUC74
TIPSUC75
TIPSUC76
TIPSUC77
TIPSUC78
TIPSUC79
TIPSUC80
TIPSUC81
TIPSUC82
TIPSUC83
TIPSUC84
TIPSUC85
TIPSUC86
TIPSUC87
TIPSUC88
TIPSUC89
TIPSUC90
TIPSUC91
TIPSUC92
TIPSUC93
TIPSUC94
TIPSUC95
TIPSUC96
TIPSUC97
TIPSUC98
TIPSUC99
TIPSUC100
TIPSUC101
TIPSUC102
TIPSUC103
TIPSUC104
TIPSUC105
TIPSUC106
TIPSUC107
TIPSUC108
TIPSUC109
CALL INFSUB (BOT, FW(I), FW(I), FUI)
APHI = -APHI
APSI = -APSI
50 CONTINUE
D(VU0(NV)) = VU0(NV) + (FW(1) + FW(2)) * CIR(NV, 2) / 12.5663704
IF (NN, LT, NN, OR, NN.EQ.M) GO TO 60
IZ = IZ + 1
NN = NN + TBLSCW(IZ)
60 CONTINUE
70 CONTINUE
NCTL = NCTL + 1
IF (NCTL .EQ. 2) 80, 100, 150
C
GEOMETRY FOR SPANWISE BOUND VORTICES
C
80 NA = MSCW + 1
NB = 2 * MSCW
JA = IM + 1
YLEG = 0(JA)
ZLEG = ZH(IM + 1)
DO 90 J = 1, NSCW
JK = IM + J
NV = J * NSCW
YLEG(NV) = PN(JK)
GO TO 40
90 XLEG(NV) = PN(JK)
C
GEOMETRY ALONG RIGHT TRAILING LEGS
C
100 NA = 2 * MSCW + 1
NB = 3 * MSCW
CCIR = 0.
JK = IM + 1
APHI = ATAN(PHI(IM + 1))
SA = SIN(APHI)
CA = COS(APHI)
YLEG = Y(JK) = S(JK) * CA
YLEGSV(IIU, IITT) = YLEG
ZLEG = ZH(IM + 1) = S(JK) * SA
ZLEGSV(IIU, IITT) = ZLEG
IF (XL(IITT) .EQ. XT(IITT)) GO TO 150
TLX1 = PN(JK) = S(JK) * TAN(PSI(JK)) = CA
JK = JK + 1
TLX2 = PN(JK) = S(JK) * TAN(PSI(JK)) = CA
CVTLG = TLX1 - TLX2
XLEG(NA) = TLX1/2, + TLX2/2
NA = NA + 1
IF (IITT .EQ. 1) GO TO 130
LIM = IITT - 1
DO 125 ITLEG = 1, LIM
125
L = HSWSV(ITLEG) + 1
DO 120 I = 2,L
   IO = I - 1
   IF((ABS(YLEGSV(I,ITLEG)-YLEG).LT.TOLC)
      S = (ABS(ZLEGSV(I,ITLEG)-ZLEG).LT.TOLC))
   S = CCIRS + CCIRS + CIRSUM(I,ITLEG)
   IF(CCIRS .NE. 0) GO TO 129
120 CONTINUE
125 CONTINUE

C
130 DO 140 NV=NA,NR
   NVT=NV-1.
140 XTLEG(NV)=XTLEG(NVT)-CRTLG
   GO TO 40
C
C
150 CONTINUE
   IF (CCIRS .NE. 0) GO TO 160
   GO TO 270
160 IJ=2*NSCN+1
   XLT=XTLEG(IJ)+CLFTLG/2.
   XRT=XTLEG(IJ)+CRTTLG/2.
   XLL=XT+CLFTLG/4.
   XRL=XRT+CRTTLG/4.
   IF (XLL.GE.XL(ITT).AND.XLT.LE.XT(ITT)) GO TO 170
   IF (XLL.LE.XL(ITT).AND.XLT.GE.XT(ITT)) GO TO 190
   IF (XLL.GT.XL(ITT).AND.XLT.LE.XT(ITT)) GO TO 200
   IF (XLL.LE.XT(ITT)) GO TO 200
   IF (XLL.GT.XL(ITT).AND.XLT.LT.XL(ITT)) GO TO 180
   CON4=(XT(ITT)-XLL)/(XLT-XLL)
170 CON4=(XLL-XT(ITT))/(XLL-XLT)
   GO TO 210
180 CON4=(XLL-XT(ITT))/(XLL-XLT)
   GO TO 210
190 CON4=1.
   GO TO 210
200 CON4=0.0
210 CONTINUE
   IF (XFL.GE.XL(ITT).AND.XRT.LE.XT(ITT)) GO TO 220
   IF (XFL.LE.XL(ITT).AND.XRT.GE.XT(ITT)) GO TO 240
   IF (XFL.GT.XL(ITT).AND.XRT.LE.XT(ITT)) GO TO 250
   IF (XFL.LE.XT(ITT)) GO TO 250
   IF (XFL.GT.XL(ITT).AND.XRT.LT.XL(ITT)) GO TO 230
   CON5=(XT(ITT)-XRL)/(XRT-XRL)
220 CON5=(XT(ITT)-XT(ITT))/(XRL-XRT)
   GO TO 260
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>230</td>
<td>CON5=(XL(ITT)-XRT)/(XR-L-XRT)</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>GO TO 260</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>CON5=1.</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>CONTINUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TIPSI = CCIRS<em>0.25</em>(CON4*WVDU(I)<em>CLFTLG-CON5</em>WVDU(IJ)*CRTTLG)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*2./SKEF**BETA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF(TIPSI.gt.0 and SSMA(1) .lt. o) GO TO 270</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF(TIPSI.gt.0)TIPSUM(ITT) = TIPSUM(ITT) + TIPSI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TIPSU = TIPSU + TIPSI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PITCH+PITCH+CCIRS<em>0.25</em>(-CON4<em>WVDU(1)<em>CLFTLG</em>XLEG(1)+CON5</em>WVDU(1J)*CRTTLG)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>)CRFTLG</em>XLEG(ITT)<em>2./(SREF</em>CREF)*BETA**2</td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>CIRCUS=CCIRS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X+460)NPOS=1+NSCW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JK=IMM+KPOS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JN=2*NSCW+NSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPI=NSCW+NP沃尔玛</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CIRCU=CRCS*CIR(JK,2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XL(ITT).EQ.XT(ITT)) GO TO 460</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLEG=XLEG(NPOS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLEG=XLEG(NPOS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XRT-XLEG(JN)-CRFTLG/2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL.LE.XL(ITT) .AND. XLT.LE.XT(ITT)) GO TO 280</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL.LE.XL(ITT),AND.XLT.GE.XT(ITT)) GO TO 300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL.GT.XL(ITT),AND.XLT.GE.XT(ITT)) GO TO 310</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL.LT.XT(ITT)) GO TO 310</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL,G.HT.XL(ITT),AND.XLT.LE.XT(ITT)) GO TO 290</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLL,G.HT.XL(ITT),AND.XLT.LE.XT(ITT)) GO TO 290</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CON1=(XT(ITT)-XLL)/(XLL-XL)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLEG=XLEG(ITT)+CCN1*CLFTLG/2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 320</td>
<td></td>
</tr>
<tr>
<td>280</td>
<td>CON1=(XL(ITT)-XLT)/(XLL-XLT)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLEG=XLEG(ITT)+XLT(ITT)/2</td>
<td></td>
</tr>
<tr>
<td>290</td>
<td>CON1=(XL(ITT)-XLT)/(XLL-XLT)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLEG=XLEG+CCN1*CLFTLG/2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 320</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>CON1=1.</td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>CON1=0.0</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>CONTINUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XLE,L.GE.XL(ITT),AND.XRT.LE.XT(ITT)) GO TO 330</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XRL.LE.XL(ITT),AND.XRT.GE.XT(ITT)) GO TO 330</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XL.LE.XL(ITT),AND.XRT.GE.XL(ITT)) GO TO 330</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IF (XRL.LE.XT(ITT)) GO TO 370</td>
<td></td>
</tr>
</tbody>
</table>
TIPSU = TIPSI + TIPSU
PITCH + PITCH * (CIRCUS * (-WVOU(NPIS)) + CLFTLG * CON1 * XLLEG + WVOU(JN) * CON2 * TIPSU)
1RTTLG + XLRLEG - CIRCUS * (JK + 2) * (WVOU(NPIS) * CON3 + BVDLG * XTLEG(NPIS)) * 2. /(SRTPSU)
2EF * CREF * BETA**2

460 CONTINUE
IM = IM + 1
IMM = IMM + TBLSW(IM)
IMP = IMP + 1
CIRCUS(IM) * TTT = CIRCUS - CCIR
IUU = IMP + 2

465 IF (IM .EQ. NSSW) GO TO 475
IF (XL(ITT) .EQ. XT(ITT)) Go TO 100
C
IF (TBLSW(IM) .NE. TBLSW(IM)) GO TO 471
NCTL = 1
CLFTLG = CRTTLG
DO 470 NV = 1 NSCW
NY = NV + 2 NSCW
XTLEG(NV) = XTLEG(NY)
WVOU(NV) = WVOU(NY)

470 CONTINUE
GO TO 80

471 NSCW = TBLSW(IM) + 1
I = IMM + 1
J = I + 1
APHI = ATAN(THI(IM) + 1))
GO TO 75
C

475 CONTINUE
KVSE(ITT) = 2.0 * ABS(TIPSI - CTSW)
IF (KVSE(ITT) .LT. EPS) GO TO 510
CENTR(ITT) = (PITCH - CMW) * CREF / ABS(TIPSI - CTSW)
C
510 CMW = PITCH
CTSW = TIPSI
NSSW = NSSWV(ITT)
C
533 CONTINUE
C
540 IF (CLOES .EQ. 100.) GO TO 541
CALL WRTANS(KVSE, CENTR, TIPSUM)
GO TO 575

541 PAD = 5. * ATAN(1.) / 180.
WRITE(6, 690)
DO 590 IK = 1, IPN
WRITE(6, 700) IK

TIPSU306
TIPSU310
TIPSU311
TIPSU312
TIPSU313
TIPSU314
TIPSU315
TIPSU316
TIPSU317
TIPSU318
TIPSU319
TIPSU320
TIPSU321
TIPSU322
TIPSU323
TIPSU324
TIPSU325
TIPSU326
TIPSU327
TIPSU328
TIPSU329
TIPSU330
TIPSU331
TIPSU332
TIPSU333
TIPSU334
TIPSU335
TIPSU336
TIPSU337
TIPSU338
TIPSU339
TIPSU340
TIPSU341
TIPSU342
TIPSU343
TIPSU344
TIPSU345
TIPSU346
TIPSU347
TIPSU348
TIPSU349
TIPSU350
TIPSU351
TIPSU352
TIPSU353
TIPSU354
XL(IK) = XL(IK) + BETA
XT(IK) = XT(IK) + BETA
WRITE(6,710) XL(IK), XT(IK),
% KVESE(IK), CENTR(IK), BOTSNIK(IK)
580 CONTINUE
WRITE(6,560)
GO TO 600 IA*1,26
READ(30) MVDUU,IA),XTLEG(IA),CIRSUM(IA)
600 CONTINUE
GO TO 605 IA*1,26
READ(30) YLEGSV(IA),ZLEGSV(IA),TBLSW(IA)
605 CONTINUE
GO TO 610 IA*1,26
READ(30) Q(IA),PN(IA),PV(IA)
610 CONTINUE
GO TO 620 IA*1,26
READ(30) ALP(IA), S(IA), PSI(IA)
620 CONTINUE
REWIND 30
CLOZ = 0.0
COMINZ = XTLEG(1)
CLOF = 0.0
COMINF = ZLEGSV(1)
COMIV = 100.0
CLOV = 0.0
DO 549 IALPH = 1,26
ALPHA(IALPH) = (IALPH-6)*2.
ALPHA = ALPH(IALPH)*RAD
SA = SIN(ALPHA)
SA2 = SA*SA
SLFS = SA2*CA + CLVS
CDVS = KVSE(I) + SA2 + SLFS
CMVS = KVSE(I) + SA2 + CENTR(I)/CREF + CMVS
550 CONTINUE
WRITE(6,570) ALPH(IALPH), MVDUU(IALPH), XTLEG(IALPH), CIRSUM(IALPH)
1 YLEGSV(IALPH), ZLEGSV(IALPH), TBLSW(IALPH)
2 Q(IALPH), PN(IALPH), PV(IALPH), CLVS, CDVS, CMVS
IF (XTLEG(IALPH),GE, COMINZ) GO TO 900
CROF = XTLEG(IALPH)
GO TO 620
900 CONTINUE
IF (ZLEGSV(IALPH),GE,COMINF) GO TO 910
CLOF = ZLEGSV(IALPH)
GO TO 620
910 CONTINUE
WRITE(30) CLVS, CDVS, CMVS
CONTINUE
END FILE 30
REWRITE 30
WRITE(6,800)
DO 810 IALPH=1,26
READ (30) CLVSE,CDVSE,CMVSE
CALL QI1ALPH,CLVSE
CALL QD(AALPH),CDVSE
CALL QMCPV,CMVSE
CALL QLC,IALPH,ALP(IALPH)
CALL QCDA,PSI(IALPH)
CALL QCMCPMA,CLL(IALPH),AILPH
CALL QDDD,IALPH
CALL QCDD,IALPH
WRITE(6,570) ALP(IALPH),ALP(IALPH),S(IALPH),PSI(IALPH),AILPH
CALL QCOINV,IALPH,GE,COINV
GO TO 920
CONTINUE
END
CONTINUE
PIAR = 1./((3.14159263)*AR)
WRITE (6,831) PIAR,CLOZ,COMINZ,CLOF,COMINF,CLOW,COMINV
DO 820 IALPH=1,26
IF(WV0I(IALPH),EQ.,CLOZ) CDCLZ(IALPH) = 100.
IF(YLEGS(VIALPH),EQ.,CLOF) CDCLF(IALPH) = 100.
IF(CLLL(IALPH),EQ.,CLOW) CDCLV(IALPH) = 100.
IF(WV0I(IALPH),NE.,CLOZ) CDCLZ(IALPH) = (XTLEG(IALPH)-COMINZ)/(WV0I(IALPH)
1 (IALPH) = CLOZ)*2)
1 (YLEGS(VIALPH),NE.,CLOF) CDCLF(IALPH) = (YLEGS(VIALPH)-COMINF)/
1 (YLEG(S(VIALPH)-CLOF)*2)
IF(CLLL(IALPH),NE.,CLOW) CDCLV(IALPH) = (CDDD(IALPH)-COMINV)/
1 (CLLL(IALPH)-CLOW)*2)
WRITE(6,830) ALPH,IALPH,AVU(CL(IALPH),CDCLZ(IALPH),YLEG(S(VIALPH)
1 CDCLF(IALPH),CLL(IALPH),CDCLV(IALPH)
CONTINUE
CALL PSEUDO
C
PLOT CL VS ALPHA FOR ZERO SUCTION, FULL SUCTION, AND VORTEX LIFT

CALL INFOPLT(0.26, ALPH,1,YLEG5,1,-10,,40.0,-4,1.6,1.0,-5,
15ALPH=-2,2HCL,12.10,10,0.75,1.50)
15ALPH=-2,2HCL,12.10,10,0.75,1.50)
CALL INFOPLT(1.26, ALPH,1,YLEG5,1,-10,,40.0,-4,1.6,1.0,-5,
15ALPH=-2,2HCL,12.10,10,0.75,1.50)
15ALPH=-2,2HCL,12.10,10,0.75,1.50)
<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL INFOPLT(0.26, ALPH,1, CLLL1,10), 40.0, -4.1, 6.1, 10, -5,</td>
<td></td>
</tr>
<tr>
<td>15HALPHA=-2, 2MCL, 0.10, 0.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.7, 0.15, 12HZERO SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.7, 11.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.4, 0.15, 12HFULL SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.4, 12.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.1, 0.15, 16HPOTENTIAL+VORTEX, 0.0, 16)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.1, 13.2)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(1.26, ALPH,1, CLLL1,10), 40.0, -4.1, 6.1, 10, -5,</td>
<td></td>
</tr>
<tr>
<td>15HALPHA=-2, 2MCL, 10.10, 10.10, 0.75, 1.50)</td>
<td></td>
</tr>
</tbody>
</table>

### CL vs CD

For zero suction, full suction, and vortex lift:

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL INFOPLT(0.26, VWOU,1, XTLEG,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.10, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, YLEGSV,1, ZLEGSV,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.10, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, CLLL1,1, CDDO,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.10, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, VWOU,1, XTLEG,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.11, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, YLEGSV,1, ZLEGSV,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.12, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.7, 0.15, 12HZERO SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.7, 11.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.4, 0.15, 12HFULL SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.4, 12.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.1, 0.15, 16HPOTENTIAL+VORTEX, 0.0, 16)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.7, 2.1, 13.2)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(1.26, CLLL1,1, CDDO,1, -4.1, 6.0, 4.1, 1.0, -2, 2MCL, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCD, 0.13, 0.10, 0.75, 1.5)</td>
<td></td>
</tr>
</tbody>
</table>

### CL vs CM

For zero suction, full suction, and vortex lift:

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL INFOPLT(0.26, YLEGSV,1, TBLSCW,1, -4.1, 6.0, 0.10, 0.10, 0.10, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCL, -2, 2HCM, 0.10, 0.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, YLEGSV,1, TBLSCW,1, -4.1, 6.0, 0.10, 0.10, 0.10, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCL, -2, 2HCM, 0.10, 0.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, VWOU,1, CIRSUM,1, -4.1, 6.0, 0.10, 0.10, 0.10, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCL, -2, 2HCM, 11.10, 10.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, VWOU,1, CIRSUM,1, -4.1, 6.0, 0.10, 0.10, 0.10, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCL, -2, 2HCM, 0.10, 0.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL INFOPLT(0.26, CLLL1,1, CMH,1, -4.1, 6.0, 0.10, 0.10, 0.10, -2)</td>
<td></td>
</tr>
<tr>
<td>2HCL, -2, 2HCM, 0.10, 0.10, 0.75, 1.50)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.7, 0.15, 12HZERO SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.6, 2.7, 11.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.4, 0.15, 12HFULL SUCTION, 0.0, 12)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.6, 2.4, 12.2)</td>
<td></td>
</tr>
<tr>
<td>CALL NOTE (7.0, 2.1, 0.15, 16HPOTENTIAL+VORTEX, 0.0, 16)</td>
<td></td>
</tr>
<tr>
<td>CALL PNTPLT(0.6, 2.1, 13.2)</td>
<td></td>
</tr>
</tbody>
</table>
CALL PNTPLT ( 6,7,2,17,13,2)
CALL INFDPLT (1,26, CLL 1, CMM 1, -4,1,6,-0,1,0,1,0,1,0,-2)
1 2HCL -2, 2HEM 13,10,0,0,75,150)

PLOT CL VS CD/(CL**2) FOR ZERO SUCTION, FULL SUCTION, + VORTEX CLI

CALL INFDPLT ( 0,26, YLEGSV 1, CDCLF 1, -4,1,6,0,1,0,1,0,-2)
12HCL -2, 24H(CD-COMIN)/((CL-CLC)**2) 0,10,10,0,0,75,150)
CALL INFDPLT ( 0,26, YLEGSV 1, CDCLF 1, -4,1,6,0,1,0,1,0,-2)
12HCL -2, 24H/(CD-COMIN)/((CL-CLC)**2) 0,10,10,0,0,75,150)

CALL NOTATE ( 7,0, 2,7,0,15,12HZERO SUCTION, 0,0,12)
CALL PNTPLT ( 6,7,2,7,7,11,2)
CALL NOTATE ( 7,0, 2,7,0,15,12HFULL SUCTION, 0,0,12)
CALL PNTPLT ( 6,7,2,7,7,11,2)
CALL NOTATE ( 7,0, 2,7,0,15,16HPOTENTIAL+VORTEX, 0,0,16)
CALL PNTPLT ( 6,7,2,7,7,11,2)

12HCL -2, 24H/(CD-COMIN)/((CL-CLC)**2) 0,10,10,0,0,75,150)

CALL NOTATE ( 7,0, 2,7,0,15,12HZERO SUCTION, 0,0,12)
CALL PNTPLT ( 6,7,2,7,7,11,2)
CALL NOTATE ( 7,0, 2,7,0,15,12HFULL SUCTION, 0,0,12)
CALL PNTPLT ( 6,7,2,7,7,11,2)
CALL NOTATE ( 7,0, 2,7,0,15,16HPOTENTIAL+VORTEX, 0,0,16)
CALL PNTPLT ( 6,7,2,7,7,11,2)
CALL NOTATE ( 7,0, 2,7,0,15,12HZERO SUCTION, 0,0,12)
CALL PNTPLT ( 6,7,2,7,7,11,2)

830 FORMAT(7F15.5)
831 FORMAT(1H1//20X6HM I N U C D R A G POLAR S H A)
1 P E F A C T O R//4X12H1/(PI/AR) = S F 0,5//27X
1 15HZERO LE SUCTION; 15X15FU LL LE SUCTION;
2 10X/HZIPOTENT.+VORTEX(LE+SE+AEU1I//26X3HCL, 8X5HCDIN, 14X3HCDL, TIPS030
3 8X5HCDIN, 14X3HCDL, 8X5HCDIN//15X6F15.5 //9X5HALPHA//12X2HCL//9XIPS031
411HCD-COMIN// 8X2HCL, 9X11H(CD-COMIN) 8X2HCL, 9X11H(CD-COMIN)//TIPS032
5 //3X13H/(CL-CLC)**2, 16X13H/(CL-CLC)**2, 16X13H/(CL-CLC)**2 ) TIPS033
560 FORMAT(1H1//47X33H TOTAL PERFORMANCE CHARACTERISTICS//
110X/110HA T T A C H E D F L O W/ TIPS034
2 W S E PA R A T E D F L O W /
372X2HPLUS POTENTIAL CONTRIBUTIONS//
13X112HZERO TIPS035
4 LEADING EDGE SUCTION FULL LEADING EDGE SUCTION LEAD ITIPS036
5 N G E DGE S I O D E DGE 13X,127HALPHA CL TIPS040
6 C D CM CL C M CO V S E CM V S E ) TIPS041
7D CM CL V S E CO V S E CM V S E ) TIPS042
570 FORMAT(13F10.5)
690 FORMAT(1H1//=31X,X60KVSE AND RESPECTIVE CHORDWISE CENTROID TIPS044
1 FOR EACH PLANFORM)
C
700 FORMAT//6X14PLANFORM NO. //2//,
$ 90X, 8LOCATION, /
$ 33X, 21LIMTS OF INTEGRATION, 18X, 5HKV SE, TIPS045
$ 5X, 9CHORDWISE, 3X, 8SPANWISE)
TIPS050
710 FORMAT(23X, F10.5, 10H (LEADING), 2X, F10.5, 11H (TRAILING),
$ 4X, F10.5, 2X, F10.5, 2X, F10.5)

800 FORMAT(1H1, // /4X11H5 E P A R A T E D
F L O W C O N T I N U E
S P O T E N T I A L C O N T
S T I T U B I O N S / 15X1094 A U G U M E N T E D
L E 4 S E
5+ AUGMENTED/
6
3X125H ALPHA CL AUG CD ATIPSU561
7UG CM AUG CL CD CM CL CD
8 CM CL CD CM)

575 CONTINUE
END

SUBROUTINE WRTANS(KVSE, CENTR, TIPSUM)
COMMON / ALL / BOT, BDSV(4), M, BETA, PTEST, QTEST,
$ STA(4), TBLSCW(100), YYCP(4),
$ Q(400), PN(400), PV(400), ALP(400), S(400), PSI(400),
$ PHI(100), ZH(100), CP(400), STLOIND(4)

COMMON / ONETHRED / TWIST(4), CREF, SREF, CAVE, CLOES, STRUE, AR,
$ ARTRUE, RTCOMT(4), CONFIG(2), NSSWSV(4),
$ MSV(4), KBOT, PLAN, IPLAN, MACH,
$ SSSWA(100), XL(4), XT(4), CLMB, CMCL, CLA(4), BLAIR(100),
$ CLAMP(4), CLWIN(4), CLWNG(4), XLCIN,
$ YINN(4), YOUTER(4)

INTEGER CONFIG

COMMON / THREFOR / CAV(2, 100), CLT, CLNT, NSSW, ALPD

REAL KP(4), KVLE(4), KVSE(4)
DIMENSION CENTR(4), CENT(4), TIPSUM(4), CENTY(4)
LCM=1
LAMAP*NSSWSV(1)
CONV=3.1415926536/180.
CONV=1.(3.1415926536+AR)
DELTA=2.*CONV
CONST=16.*BOT/SREF
ALPHA=ALPD*CONV
S22=WALPHA**2
EPS = 1.*6
DO 10 IIT=1, IPLAN
CEN(T(IIT)) = 0.0
XT(IIT) = XT(IIT) + BETA
XL(IIT) = XL(IIT) + BETA
CENTY(IIT) = 0.0
KP(IIT) = 0.0
10 CONTINUE

C
KVL(ITT) = 0.0
10 CONTINUE
C
KP(1) = CLWIN(1)
IF(IPLAN .EQ. 1) GO TO 20
DO 15 IIT = 2, IPLAN
KP(IIT) = CLWIN(IIT) - CLWIN(IIT-1)
15 CONTINUE
20 CONTINUE
C
C
IEDGE = 1
INDEX = 0
DO 50 IIT = 1, IPLAN
MSPAN = NSSWSV(IIT)
PITCHMO = 0.0
FORCE = 0.0
ROLLMO = 0.0
DO 10 ISPN = 1, MSPAN
INDEX = INDEX + 1
IF(0.01< YINNER(IIT)) GO TO 25
INDEX = INDEX + 1
IF(IIT .EQ. YOUTER(IIT)) GO TO 25
PITCHMO = PITCHMO + (PN(IEDGE) - (PN(IEDGE-1))/4.0)* BETA *
B L A I R(INDEX) = S(IEDGE) = CONST
FORCE = FORCE + BLAIR(INDEX) = S(IEDGE) = CONST
ROLLMO = ROLLMO + O(IEDGE) = BLAIR(INDEX) = S(IEDGE) = CONST
10 CONTINUE
25 IEDGE = IEDGE + TBLSCW(INDEX)
30 CONTINUE
IF(0.01< FORCE .EQ. 0.0) GO TO 50
KVL(IIT) = FORCE / 522
IF(PITCHMO .NE. 0.0) CENT(IIT) = PITCHMO / FORCE
IF(ROLLMO .NE. 0.0) CENTY(IIT) = ROLLMO / FORCE / BOT
IF(KVL(IIT)) .LT. EPS) GO TO 50
CENT(IIT) = 0.0
CENTY(IIT) = 0.0
50 CONTINUE
C
100 CONTINUE
WRITE (6, 190)
DO 110 IK = 1, IPLAN
CENTP = CLAYAR(IK)*CREF
WRITE (6, 200) IK
WRITE (6, 210) KP(IK), CENTPM, SPANFK
WRITE (6, 220) YINNER(IK), YOUTER(IK),
$ KVLE(IK), CENT(IK), CENTYSP
WRITE(6,230) XL(IK), XT(IK),
$ KVSE(IK), CENTR(IK), BOTSV(IK)
WRITE(6,290) TIPSUM(IK)
110 CONTINUE

120 CONTINUE
DO 160 IK=1,IPLAN
IF (LCM.EQ.1) GO TO 130
WRITE (6,240) IK
130 WRITE (6,250)
ALPHA=0.0
DO 160 J=1,26
V=SIN(ALPHA)
C=COS(ALPHA)
C2=C**2
S2=V**2
IF (LCM.EQ.1) GO TO 140
C
INDIVIDUAL PLANFORM CHARACTERISTICS
C
CLP = KP(IK) * V * C2
CLVL = CLP + KVLE(IK) * S2 * C
CLSL = CLP + KVSE(IK) * S2 * C
CLTOT = CLVL + KVSE(IK) * S2 * C
CMP = CLAMAR(IK) * KP(IK) * V * C
CMPL = CMP + CENT(IK) * KVLE(IK) * S2/CREF
CMPS = CMP + KVSE(IK) * CENTR(IK) * S2/CREF
CMTOT = CMPL + KVSE(IK) * CENTR(IK) * S2/CREF
GO TO 150
C
TOTAL PLANFORM CHARACTERISTICS
140 CLP = SKP * V * C2
CLVL = CLP + SKVLE * S2 * C
CLSL = CLP + SKVSE * S2 * C
CLTOT = CLVL + SKVSE * S2 * C
CMP = V * C * SCMP
CMPL = CMP + S2 * SCMPL/CREF
CMPS = CMP + S2 * SCMPS/CREF
CMTOT = CMPL + CMPS - CMP
C
150 CDI=CLTOT*TAN(ALPHA)
CDII=(CLTOT**2)*CINV
ALPHA=ALPHA/CONV
CINT=CLTOT/C
WRITE (6,260) ALPHAI,CINT,CLPL,CLVL,CLSL,CLTOT,CMPS,CMPL,CMTOT,C
101,CDI
C
160 ALPHA=ALPHA+DELTA
IF (IPLAN.EQ.1) GO TO 170
IPLAN=1
LCM=1
WRITE (6, 280)
SKP = 0.0
SKVLE = 0.0
SKVSE = 0.0
SCMP = 0.0
SCMPL = 0.0
SCMPS = 0.0
C
K = IFIX(PLAN)
DO 165 ITT = 1, K
    SKP = SKP + KP(ITT)
    SKVLE = SKVLE + KVLE(ITT)
    SKVSE = SKVSE + KVSE(ITT)
    SCMP = SCMP + KP(ITT) * CLAMAR(ITT)
    SCMPL = SCMPL + KVLE(ITT) * CENT(ITT)
    SCMPS = SCMPS + KVSE(ITT) * CENT(ITT)
165 CONTINUE
C
170 WRITE (6, 270)
   IPLAN = PLAN
   RETURN
C
180 FORMAT (8F10.5)
190 FORMAT (1H1, 1/41X,
$ 50HKP > KV AND RESPECTIVE CENTROIDS FOR EACH PLANFORM )
C
200 FORMAT (/55X, 18HPLANFORM NUMBER ,12//,
$ 92X, 8HLOCATION/,
$ 77X, 5HVALUE, 3X, 8HCORRWISE, 4X, 8HPANWISE/,
$ 21X, 21HLIMITS OF INTEGRATION )
C
210 FORMAT (65X, 7HKP ,3(F10.5, 2X))
C
220 FORMAT (10X, F10.5, 2X, 7H(INNER), 6X, F10.5, 2X, 7H(OUTER),
$ 11X, 7HKV LE ,3(F10.5, 2X))
C
230 FORMAT (10X, F10.5, 2X, 9H(LEADING), 4X, F10.5, 2X, 10H(TRAILING),
$ 8X, 7HKV SE ,3(F10.5, 2X))
C
240 FORMAT (1H1, 1/43X, 40HPERFORMANCE CHARACTERISTICS FOR PLANFORM, I,WRITAN164'
12)
C
250 FORMAT (/7X, 5HALPHA, 6X, 2HCN, 8X, 3HCLP, 4X, 9HCLP+CLVLE, 1X, 9HCLP+CLV5,
$ 1E+4X, 2HCL, 8X, 3HCM, 4X, 9HCM+CMVLE, 1X, 9HCM+CMVSE, 4X, 2HCM, 8X, 2HCO, 3WRTAN167
2X+13HCL+2/(PI*AR/) )
C
260 FORMAT (3X, 12F10.5)
270 FORMAT (/***,50X,21H THIS CASE IS FINISHED)  
280 FORMAT (1HI,,/,,/,,40X,33H TOTAL PERFORMANCE CHARACTERISTICS)  
290 FORMAT (140,32X,  
$ 47HSUM OF THE POSITIVE SIDE EDGE CONTRIBUTIONS = #F10.5)  
END  
SUBROUTINE VORTEX  
DIMENSION GAM(1000), XCC(1000), YQ(1000), CCR(401),  
$ FW(2), FY(2), XCC(400), CCC(400), CUC(400), YR(100),  
$ BOTL(4), NUMBER(4), NMASSUM(4),  
$ CRII(100), NMAI(4), XCC(400), CHP(100), XCC(44), YQ(2),  
$ PPHI(100), ZIM(100), Z(100), PHI(100), SA(100),  
$ SSA(100), ALDP(100), ALDPD1(1000), AIDS(1001),  
$ YC(102), YQ0(100), CCR(401), CCR(402), YC(102), YCHLD(4), YCHHI(4),  
$ VELIN(26), CORAGIT(26), CCLF(26), GAQ(1000), CUTE(26),  
$ CM(26), CORAG(26), CLIFT(26), CPITCH(26),  
$ CSS(26), CSUCT(26), CLAUG(26), CDAG(26), CMIG(26), CLV(26)  
COMMON /ALL/, BOT, BOTS(4), M, BETA, PTEST, QTEST,  
$ STA(4), TBLSCW(100), YYCP(4),  
$ Q(400), PN(400), PTV(400), ALP(400), S(400), PSI(400),  
$ PHI(100), ZH(100), CP(400), STLOIND(4)  
C  
C --- NOTES TO THE USERS ---  
C  
1. BOTH TOTAL RESULTS AND THOSE FROM THE LEADING  
EDGE VORTEX SOLUTION WILL AGREE IF AND ONLY IF  
ALL PANELS ARE OF UNIFORM WIDTH, AND CAN BE  
CALCULATED FROM CLDES = 100, AND CLDES = 1.0  
C  
2. IF A WING HAS MORE THAN ONE STREAMWISE TIP, IT IS RECOMMENDED THAT THE WING BE INPUT AS TWO PLANFORMS TO PROVIDE  
MORE MEANINGFUL SIDE EDGE RESULTS  
C  
3. STLOIND - "STREAMWISE LOAD INDICATOR" ARRAY; SET TO  
0, IF THE LOADING ALONG THE ENTIRE OUTER STREAMWISE  
EDGE OF THIS PLANFORM IS TO BE 0.0; OTHERWISE, SET TO  
1.0 IF THIS LOADING IS TO BE NON-ZERO  
C  
COMMON /ONETHRE/, TWIST(4), CREF, SREF, CAVE, CLDES, STRUE, AR,  
$ ARTTRUE, RTCDHT(4), CONFIG(2), NSSWSV(4),  
$ -V41H, WATT, PTV, IFLEP, PACH,  
$ S1=AI(100), YL(4), YT(4), CLF, CM1, CIAR(4), ALAIR(100),  
$ CIAR(4), C1AR(4), C1AR(4), YQ(2), Y105,  
$ Y106, Y107, Y108  
C  
INTEGER CONFIG
COMMON /TOTHRED/ CIP(400,2)
COMMON /INSURI3/ ADRI,APH1,XY,YY,ZZ,SN,NOLCSQ
COMMON /THERMP/ CCAV(2,1CO), CLT, CLNT, KSS, ALPD
COMMON/CCRROD/ TSPAN(4), TSNA, KRIT, CTILDA, XTILDA, DISTALE

DIMENSION CROG(26,4), CLE(26,4), CPH(26,4), CST(26,4)

DATA NUMREF,6H1ST,...

ICOUNT=1
DO 1 ITT = 1, IPLAN
YYCHI(ITT) = -0.0
YCMT(ITT) = -0.0
1 CONTINUE
RETURN
5 WRITE (6,250)
DO 10 I = 1, IPLAN
10 WRITE(6,255) I, YCHI(I), YCMT(I)
IF (ICOUNT.EQ.1) WRITE(6,275)
IF (ICOUNT.EQ.2) WRITE(6,280)
IFAMP=20
API=TOLC50+TRLS=0.
PI=4.*ATAN(1.)
PHI=4.*DAT
DO 15 ITT = 1, IPLAN
ROT(IJT) = API(TSPAN(IJT))
15 CONTINUE
SNH = ROT(KBOT) / (2.0 * NNSWV(KBOT))
DELTY=2.*SNH
NMAY = C
DO 17 ITT = 1, IPLAN
NMIX(IJT) = AMIX(IJT) / DELTYB
IF (NMIX(IJT).LE.0.0) GO TO 18
NMAY= NMAY + NMIX(IJT)
NMIX(IJT) = NMAY
17 CONTINUE
GO TO 19
18 WRITE(6,790)
M = -1
GO TO 460
19 SXMIN = 20.
100 CONTINUE
II=1
IC = 0
IT = 0
NCT = 0
NST = 0
GO TO 150 I=1,IPLAN
IU2=M5SWV(I)
III=THI+1
IF (TLOIND(I) .EQ. 1.) II'X = IU2
IC = IC + MSV(I)
IT = IC+1
IZ = IZ + MSV(I)
YCAT=0.
JAMM=KMA(I)
C
105 CONTINUE
GO TO 140 LA=1,NSCWIN
IF (TLOIND(I) .EQ. 1.) GO TO 107
Ye(I)=-PZ2.
CRI(I)=0.
CXY(I)=0.
107 GO TO 170 J = 1,IU2
L=J+1
IF (TLOIND(I) .EQ. 1.) L = J
LU = L8 + (J-1 + NST) * NSCWIN
SCALE(J)=ALP(LU)
X(:4(J))=XCF4(LU)
CRI(L)=CCC(LU)
CXY(L)=CCU(LU)
IF (L,A,+.F.,1) GO TO 120
JJ = J + NST
ZXX(J)=Z+X(JJ)
CAX(J)=SSWAX(JJ)
PP=I(IJ)*PH(JJ)
YCO(IJ)=Y(IJ)
II=II+TPLSCW(JJ)
IF*II=II+1
ITL=TPLSCW(I2)
IT=II-ITL
JAM=I+ITL
IF (JG,GT,IC) YCAT=YCAT-S(ID)
IF (JG,GT,IC) GO TO 110
YCAT=YCAT-S(ID)-S1A)
110 1Z=17-I
YBF(IF)*YCAT
120 CONTINUE
GO TO 100 JP=1,1U7
J7=J+1
TE = (TTL*IND(I) *EQ. 1.) J7 = JP
YCS(J7) = ASIN(YR(JP)/RTIL(I))
CONTINUE
YCS = NYA(I)+2.*SNK-SNK
DO 140 K = 1, LAM
K0 = LA + (K-1) + NSTT)*NSCWMN
YCS=YNK+DELTY
YCS = ASIN(YR/RTIL(I))
CALL FLTLP (YPR, YQI(KP)+1, I1Z, YM, YQI)
CALL FLTLP (YPR, ALFPLD(KP)+1, I1Z, YM, ALL)
CALL FLTLP (YPR, SS(KP)+1, I1Z, YM, SA)
CALL FLTLP (YPR, XA(KP)+1, I1Z, YM, XC44)
CALL FLTLP (YPR, 7(KP)+1, I1Z, YM, 7KM)
CALL FLTLP (YPR, PHI(KP)+1, I1Z, YM, PPH)
CALL FLTLP (YPR, CAM(KP)+1, I1Z, YM, CPR)
CALL FLTLP (YPR, CAT(KP)+1, I1Z, YM, CKJ)
IF (YMGT,YR(I1Z)) CAM(KP) = CPR(KP)
IF (YMGT,YR(I1Z)) CAM(KP) = CPR(KP)
CONTINUE
140
NST = NSWSSV(I) + NST
NSTT = NMA(I) + NSTT
CONTINUE
DO 151 IALPH = 1, 26
COSIALPH = CLF(TIALPH)*CPICTHIALPH*CSUCTIALPH = 0.
SINIALPH = COTAIALPH*CPICTHIALPH*C YANGIALPH = O.
CONTINUE
COSIT = A.*SNK/SREF
CONTINUE
C
DO 1510 IALPH = 1, 26
COSIALPH = R0.0
SINIALPH = R0.0
CONTINUE
1510 CONTINUE
CONTINUE
CONTINUE
C
READING 10
DC 170 (1*1, NMAX)
IPOT = 1
ITPLN = 1
IN 155 IIT = 1*KPOT
IF (IIT.GT. N*SCLM(IIT)) IPOT = IPOT + 1
CONTINUE
L2= (L1-1)*NSCWMN+1
L2= L1*NSCWMN
DO 157 IALPH = 1, 26
152 CPACCT(IALPH)*CLL(IALPH)*CMPC(IALPH)*CSS(IALPH)*CLV(IALPH)*C.
CONTINUE
AAP=ATAN(ALPHD(LA))
OLYF=(YC4(LA)+0.25*(YC4(LA)-YC4(LA+1)))*RETA
ELE=ATAN(55A(LA))
DO 100 NV=1,LA
CPT=CS5(ATAN(PHI2(NV)))
DO 100 I=1,5 ALPH=1.26
ALPHA=(IALPH-6)*2.*PI/180.
GAM=GAM(LA) + SIN(ALPHA)*GAM(LA)
IF(GAM,F0,0.) GO TO 153
CUTELL(IALPH)=ABS(GAMM)/GAM
GO TO 154
153 NLTE(IALPH)=1.
154 NLTE=1:(IALPH)*0.
155 CONTINUE
IF (-COUNT .GT. 1) GO TO 171
DO 170 NN=1,MM
YY=YY(NV)-YY(NN)
YY(NV)=YY(NN)
YY(2)=YY(NV)+YY(NN)
YY(1)=YY(NV)-YY(NN)
IF(GAMM,F1,F0) GO TO 153
APHI=ATAN(PHI2(NN))
DO 160 I=1,2
YY=YY(I1)
CALL INSUR (8CT,FV(1)FV(2)FV(3)FV(4))
APHI=APHI
160 CONTINUE
DO 170 I=1,5 ALPH=1.26
ALPHA=(IALPH-6)*2.*PI/180.
GAM=GAM(NV) + SIN(ALPHA)*GAM(NV)
VEL=VEL(NV)*(FV(1)+FV(2))-(FV(1)+FV(2))*PHI2(NV)*GAMM/FPI*VELNV
11(IALPH)
170 CONTINUE
DO 170 I=1,7 ALPH=1.26
CONTINUE
DO 170 I=1,7 ALPH=1.26
CONTINUE
DO 170 I=T+170
PRED(0)VEL(NV)
CONTINUE
DO 170 I=1,7 ALPH=1.26
CONTINUE
DO 170 I=1,7 ALPH=1.26
ALPHA=(IALPH-6)*2.*PI/180.
GAM=GAM(NV)*SIN(ALPHA)*GAM(NV)
VEL=VEL(NV)*(FV(1)+FV(2))-(FV(1)+FV(2))*PHI2(NV)*GAMM/FPI*VELNV
CONTINUE
DO 170 I=1,7 ALPH=1.26
CONTINUE
DO 170 I=1,7 ALPH=1.26
ALPHA=(IALPH-6)*2.*PI/180.
GAM=GAM(NV)*SIN(ALPHA)*GAM(NV)
VEL=VEL(NV)*(FV(1)+FV(2))-(FV(1)+FV(2))*PHI2(NV)*GAMM/FPI*VELNV
CONTINUE
C

LEADING EDGE VOPTE FLOW ANALYSIS EMPLOYED HERE

C1 = CUV + COS(IALPH) * CUTE(IALPH)
C2 = CUV + SIN(IALPH) * CLTE(IALPH)
C3 = CUV + TAN(IALPH) * AAP + CLLE(IALPH)
C4 = CUV + SIN(IALPH) * CLLE(IALPH)

IF(YCNV).LT.YCHLD(IPOT),OR,YCNV).GT.YCHLD(IPOT)) GO TO 173

C

ATTACHED FLOW ANALYSIS EMPLOYED HERE

ONE HUNDRED PERCENT LEADING EDGE SUCTION

GO TO 177

C

ATTACHED FLOW ANALYSIS EMPLOYED HERE

ZERO PERCENT LEADING EDGE SUCTION

GO TO 174

C

CONTINUE

C

174 CONTINUE

CDRAGIT(IALPH) + CDRAGIT(IALPH) * GAM + CUV * COW + CN) + 2.
CLL (IALPH) + PLL(IALPH) * GAM + CLV * COW + CN) + 2.
CMG(IALPH) + CMG(IALPH) * GAM + CUV * COW (AAP) + YCNV * BETA + CN3) + 2.

C

175 CONTINUE

GP 1 = S + 1,26
WRITE(1G) CLL (IALPH), CMG(IALPH), CDRAGIT(IALPH)
IF (ICOUNT.EQ.3) WRITE(1G) CLV (IALPH)

C

180 CONTINUE

C

GO TO 185, IALPH = 1,26
WRITE(1G) CLL (IALPH), CMG(IALPH), CDRAGIT(IALPH)
IF (ICOUNT.EQ.3) WRITE(1G) CLV (IALPH)

C

185 CONTINUE

C

190 CONTINUE
END FILE 10
DO 104 IALPH = 1,26
DO 104 ITT = 1,1PLAN
CDAG(IALPH) = CDAG(IALPH) + CDG(IALPH,ITT)
CLIFT(IALPH) = CLIFT(IALPH) + CLF(IALPH,ITT)
CPTCH(IALPH) = CPTCH(IALPH) + CP(IALPH,ITT)
IF (ITCOUN.T, EQ. 3)
  CSUCT(IALPH) = CSUCT(IALPH) + CST(IALPH,ITT)
104 CONTINUE
C
IF (ITCOUN.T, EQ. 1) END FILE 2G
IPTU = 0
RE=IND. 10
IF (ITCOUN.T, NE. 3) GO TO 193
CSUCMN = 1.
60 192 JI=1,26
CSUCMN = AMIN1(CSUCMN, CSUCT(IJ))
IF (CSUCMN.EQ.0., CSUCT(IJ)) IALPSV=IJ
192 CONTINUE
193 DO 275 IALP=1,26
ALPHA = (IALP-6)*7.
WRITE(6,260) ALPHA
IF (ITCOUN.T, NE. 3, NP, DISTALF, EQ. 0.) GO TO 191
AAL = ALPHA**PI/180.
JSIGN = 1.
IF (AILP.LE., IALPSV) JSIGN = -1.
TERM = CSUCT(IALP) * CSILDA/DISTALF*JSIGN
CLUS(IALP) = TERM**COS(AAL)
COSL(IALP) = TERM**SIN(AAL)
CSUAL(IALP) = TERM**XILDA/CRFF
275 CONTINUE
C
191 TPLE=ITU=0.
L1 = 0
IPLT = IPTU + 1
* STT = G
DO 220 J = 1, IPLAN
IAM**N(M)(I)
C
DO 200 J = 1, IAMP
JJ = J + NSTT
LA = 1 + (J - 1 + NSTT) * NSCMIN
DO 195 IALPH=1,26
READ(10) CLL(IALPH), CMY(IALPH), COPLAG(IALPH)
IF (ITCOUN.T, EQ. 3)
  READ(10) CLV(IALPH)
195 CONTINUE
CCM(JJ) = CLV(IPLT)/CAVE
CCM(JJ) = CMY(IPLT)/(CAVE*CRFF)
XCM(JJ) = COPLAG(IPLT)/CAVE
CON(JJ) = CLV(IPLT)/CAVE
CON(JJ)*CCM(JJ)
200 CONTINUE

210 CONTINUE

220 CONTINUE

230 CONTINUE

300 CONTINUE

310 CCONT = ICONT + 1

320 CONTINUE

400 CONTINUE

END FILE 30
REWIND 30
C
C

VCRTF348
VCRTF349
VCRTF350
VCRTF351
VCRTF352
VCRTF353
VCRTF354
VCRTF355
VCRTF356
VCRTF357
VCRTF358
VCRTF359
VCRTF360
VCRTF361
VCRTF362
VCRTF363
VCRTF364
VCRTF365
VCRTF366
VCRTF367
VCRTF368
VCRTF369
VCRTF370
VCRTF371
VCRTF372
VCRTF373
VCRTF374
VCRTF375
VCRTF376
VCRTF377
VCRTF378
VCRTF379
VCRTF380
VCRTF381
VCRTF382
VCRTF383
VCRTF384
VCRTF385
VCRTF386
VCRTF387
VCRTF388
VCRTF389
VCRTF390
VCRTF391
VCRTF392
VCRTF393
VCRTF394
VCRTF395
VCRTF396
VCRTF397
VCRTF398
VCRTF399
VCRTF400
240 FORMAT(/37X,22H DISTRIBUTIONS FOR THE 416X9H PLANFORM /)
250 FORMAT(1H1/80X* AERODYNAMIC CHARACTERISTICS FOR CAMBERED AND TWISTED VORTICES)

1 40X* AERODYNAMIC CHARACTERISTICS FOR CAMBERED AND TWISTED VORTICES
20* WINGS/4X* WITH VORTICES AT VARIOUS ANGLES OF ATTACK*
255 FORMAT(/33X PLANFORM *12* HAS LEADING EDGE VORTEX FLOW ASSUMED FROM VORTICES)
11X *,F12.5, TO *,F17.5, //4X,
20* ATTACHED FLOWS /5X* VORTEX FLOW ACROSS THE SPAN*
260 FORMAT(13X 12X 10X 5(F0.5,1X))
265 FORMAT(1H1,///50X* ANGLE OF ATTACK * *F0.5,1X* DEGREES */60X* DEG VTIC F0.48"
10X CHARACTERISTICS*/ 14X* STATION*, 12X, 2Y/R*, 14X9HCL*C/CAVE,
201X9HCL*C/CAVE, 4X2H(C*M**2)/CAVE,CREF), 4X, 17HCL VORT LE*C/CAVE, VORT
3 90Y9HARNT C.G/

770 FORMAT(/35X,21H TOTAL CHARACTERISTICS ,///70X,
7 4HCL =*,F12.5,17X, 4HDD =*,F12.5,17X, 4HCM =*,F12.5)
771 FORMAT(/35X,21H PLANFORM CHARACTERISTICS ,///70X,
7 4HCL =*,F12.5,17X, 4HDD =*,F12.5,17X, 4HCM =*,F12.5)
775 FORMAT(////40X* ZER0 PERCENT LEADING EDGE SUCTION ASSUMED*)
780 FORMAT(////33X*40% HUMPED PERCENT LEADING EDGE SUCTION ASSUMED*)
785 FORMAT(////50X,11HC SUCTION *,F12.5)
790 FORMAT(////13X,3CH*** THE VALUE OF -V/C- IS TOO ,
7 39H5WALL, PLEASE CORRECT AND RESUBMIT. *** )

RETURN
END

SUBROUTINE CNLONG

THIS OVERLAY COMPUTES XNUM LOCAL CN VALUES AND TOTAL CN FOR EACH PLANFORM.

ROBERT GRAY COMPUTER SCIENCES CORP. 1980

COMMON /ALL/ BDT, BDTV(4), M, BETA, PTEST, QTEST,
* STA(4), TBL(100), YYCP(4),
* Q(400), RN(400), PV(400), ALP(400), S(400), PSI(400),
* PHI(100), ZH(100), CP(400), STLOIND(4)

COMMON /THH/ CIR(C400,2)

COMMON /THRE/ CLT, CLNT, NSSV, ALPD

COMMON /ONETH/ TWIST(4), CREF, SREF, CAVE, CLOES, STRU, AR,
* ARTRUE, RTCDMT(4), CONFIG(2), NSSWS(4),
* MSV(4), KDOT, PLAN, IPLAN, MACH,
INTEGER CONFIG

COMMON /MAINONE/,ICODEOF, TOTAL, AAN(4), X5(4), YS(4), KFCTS(4),
$ XREG(25,4), YREG(25,4), AREG(25,4), DTM(25,4), MCQ(25,4),
$ XX(25,4), YY(25,4), AS(25,4), TTWD(25,4), MMCD(25,4), AN(4),
$ ZI(25,4), ITIPCOD, ICMTST

DIMENSION DCPS1(1,25), YL(25), XXL(25),
$ WK1(49), XOVERL(25), CNL(25), YLOBLZ(25), IENSDW(2),
$ RPTS(2), CPX(400)

DATA XNUM /20.0/
DATA YNUM /10.0/
DATA IST /1/
DATA CNG /0.0/

WRITE(6,805)

FIND X COORDINATES FOR ALREADY COMPUTED
DELTA CP'S

DO 5 I = 1,M
CPX(I) = PN(I) * BETA
CONTINUE

WRITE(6,806) IP

WRITE(6,807)

CONTINUE

DO 1000 IP = 1,IPN
WRITE(6,806) IP
WRITE(6,807)

-IST- STARTING POSITION IN ARRAYS
CPX, Q, AND CP FOR EACH PLANFORM
-MAX- NUMBER OF POINTS IN ARRAYS CPX, Q,
AND CP USED BY EACH PLANFORM
-N- NUMBER OF BREAKPOINTS USED TO DEFINE
EACH PLANFORM

IF (IP GT 1) IST = IST + MSV(IP - 1)
N = AN(IP) + 1
MAX = MSV(IP)
XFORE = -999999
XAFF = -999999
YMIN = 999999
YROOT = -999999

C FIND LEADING AND TRAILING POINTS
DO 10 I = 1,N
  IF (XFORE .LT. XX(I,IP)) XFORE = XX(I,IP)
  IF (XAFF .GT. XX(I,IP)) XAFF = XX(I,IP)
  IF (YMIN .GT. YY(I,IP)) YMIN = YY(I,IP)
  IF (YROOT .LT. YY(I,IP)) YROOT = YY(I,IP)
10 CONTINUE

C FIND TOTAL LENGTH -RL-
RL = ABS(XFORE - XAFF)

C FIND THE X INCREMENT
XINC = RL/NUM

C USE MID POINT OF EACH X SEGMENT FOR EACH LOCAL VALUE OF X
XNEXT = XFORE - (XINC / 2.0)

C FOR EACH VALUE OF XXL...
NUMX = XNUM
N = AN(IP)

C DO 100 I = 1,NUMX
XXL(I) = XNEXT

C FIND INTERCEPT POINT AT THIS XXL
NPTS = 0
DO 30 J = 1,N
  RXMAX = AMAX1(XX(J,IP),XX(J+1,IP))
  RXMIN = AMIN1(XX(J,IP),XX(J+1,IP))
  IF (XXL(I) .LE. RXMIN .OR. XXL(I) .GE. RXMAX) GO TO 30
  C PERPENDICULAR TO XXL WILL INTERCEPT
  NPTS = NPTS + 1
  IF (NPTS .LE. 2) GO TO 25
  WRITE (6,990) XXL(I)
  GO TO 95
C C FIND Y COODINATE FOR INTERCEPT
25  RYMAX = AMAX1(YY(J,IP),YY(J+1,IP))
  RYMIN = AMIN1(YY(J,IP),YY(J+1,IP))
  A = ABS(RXMIN - RXMAX)
  AP = ABS(XXL(I) - RXMAX)

C
B = ABS(RYMAX - RYMIN)
BP = (B * AP) / A
C
RPTS(NPTS) = RYMAX - BP
30 CONTINUE
C
IF(NPTS .EQ. 0) GO TO 90
C
SET BL AND COMPUTE Y COORDINATES
ON NOTCHED PORTION OF THE WING
Y GOES FROM THE LEADING EDGE TO THE INNER
EDGE (I.E., FROM YLE TO RYMAX); OTHERWISE,
FROM YLE TO YROOT.
C
RYMAX = YROOT
YLE = PPTS(1)
IF (NPTS .EQ. 1) GO TO 35
RYMIN = AMIN1(RPTS(1),RPTS(2))
RYMAX = AMAX1(RPTS(1),RPTS(2))
YLE = RYMIN
35 BL = (RYMAX - YLE) / 2
WRITE(6,B600)XXL(I),BL
YINC = (RL / 2.) / (YNUM - 1)
NUMY = YNUM
YNEXT = 0.0
DO 40 J = 1,NUMY
YL(J) = YLE + YNEXT
YNEXT = YNEXT + YINC
40 CONTINUE
C
INTERPOLATE FOR NEW DELTA CP'S
C
CALL INTERP(CPX(IST),Q(IST),CP(IST),MAX,XXL(I),1,YL,NUMY,
5      DCP,1,ITER)
C
DO 45 J = 1,NUMY
WRITE(6,B601)XXL(I),YL(J),DCP(J)
45 CONTINUE
C
COMPUTE CNL AT THIS XXL
= RL/BL TIMES INTEGRAL FROM 2 * YLE / BL TO
2 * YROOT (OR INNER EDGE OF WING) / BL OF DELTA CP
TIMES DI(2Y/BL)
C
DO 50 J = 1,NUMY
YL0=YL2(J) = (YL(J) / BL) * 2.
50 CONTINUE
C
C
WRITE(*,803) IP,TCN
C
1000 CONTINUE
C
WRITE(*,804) CN
C
800 FORMAT(5X,F12.5,42X,F12.5)
801 FORMAT(5X,F12.5,5X,F12.5,5X,F12.5)
802 FORMAT(72X,6HCNL = ,F12.5)
803 FORMAT(1HO,17HCN FOR PLANFORM , I1,3H = ,F12.5)
804 FORMAT(1HO,11HTOTAL CN = ,F12.5)
805 FORMAT(1H1,50X,30HLONGITUDINAL LOAD DISTRIBUTION)
806 FORMAT(1HO,57X,17HPLANFORM NUMBER ,II)
807 FORMAT(1HO,10X,1HX,1X,1X,1HY,13X,12HINTERPOLATED,9X,5HBL(X),/
$ 44X,8HDELTA CP,//)
900 FORMAT (1H ,27HTN MANY INTERCEPTS AT X = , F12.5,
$ 2X,17HTWO ARE PERMITTED)
901 FORMAT(1H ,27HTD INTERCEPT FOR X STATION ,12,
$ 4HX = ,F12.7)
C
9000 CONTINUE
END

SUBROUTINE INTERP(CPX,Q,CP,MAX,XXL,NX,YL,NUMY,DCP,/
$ NO,IER)
C
THIS SUBROUTINE CALLS ROUTINE -IQHSCV- TO INTERPOLATE
VALUES FOR DELTA CP (DCP).
C
CPX--- X COORDINATES
Q--- Y COORDINATES
CP--- DELTA CP VALUES AT CPX,Q
MAX--- NUMBER OF POSITIONS TO BE USED IN CPX,Q, AND CP
XXL--- ARRAY CONTAINING NEW X COORDINATES
NX--- NUMBER OF ELEMENTS IN XXL
YL--- ARRAY CONTAINING NEW Y COORDINATES
NUMY--- NUMBER OF ELEMENTS IN YL
DCP--- OUTPUT ARRAY DIMENSIONED NX BY NY CONTAINING
INTERPOLATED DELTA CP'S
NO--- ROW DIMENSION OF ARRAY DCP
IWK--- INTEGER WORK ARRAY OF LENGTH
         311 + MAX + NX * NUMY
WK--- REAL WORK ARRAY OF LENGTH 6 * MAX
IER--- ERROR RETURN
   * 129, MAX IS LESS THAN 4 OR NX OR NUMY
   IS LESS THAN 1
   * 130, DATA POINTS ARE COLINEAR
   * 131, SOME DATA POINTS ARE IDENTICAL
INTKP27
DIMENSION CPX(1),Q(1),CP(1),XXL(1),YL(1),DCP(1),NUMY
DIMENSION IWK(12425),WK(2400)

CALL IQMSCV(CPX,Q,CP,MAX,XXL,NX,NX,DCP,ND,IWK,WK,IER)

RETURN

END

SUBROUTINE IQMSCV (XD,YD,ZD,ND,XI,NXI,YI,NYI,ZI,IZI,IWK,WK,IER)
INTEGER ND,XI,NXI,YI,NYI,ZI,IZI,IER,IWK(1)
REAL XD(1),YD(1),ZD(1),XI(1),YI(1),ZI(IZI),IWK(1)
INTEGER 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

CALL ICGS(NDPO,XD,YD,NT,IWK(JWPL),NL,IWK(JWPL),IWK(JWPL),)
1 IWK(JWPL),WK,IER)
2 IF (IEP.GE.129) GO TO 3000
3 IWK(5) = NT
4 IWK(5) = NL
5 IF (NT.EQ.0) GO TO 9000
6 IAK(JWCP1) = IWK(JWPL1)
7 CALL ICGS(NDPO,XD,YD,NT,IWK(JWPL),NL,IWK(JWPL),IWK(JWPL),)
8 IWK(JWPL),WK,IER)
9 ITPV = 0
10 JIGOMX = 0
11 JIGIMN = XI*NYI+1
nnnp = nt*2*nl

do 25 jngp=1,nnnp
    iti = jngp
    if (jngp.le.nt) go to 5
    il1 = (jngp-nt+1)/2
    il2 = (jngp-nt+2)/2
    if (il2.gt.nl) il2 = 1
    iti = il1*(nt+nl)+il2
    jwngp = jwngp+1
    ngp0 = iw(k(jwngp))
    if (ngp0.eq.0) go to 15
    jigmn = jigmn+1
    jigmx = jigmx-ngp0
    do 10 jigp=jigmn,jigmx
        jwigr = jwigr+jigp
        iz = iw(k(jwigr))
        iy = (iz-1)/nxio+1
        ix = iz-nxio*(iy-1)
        call i0hsf(xd,yd,zd,nt,iwk(jwftp),nl,iwk(jwipl),wk,iti,
        x(iy),y(iy),z(iy,iy))
    10 continue
    15 continue
    jwngp = jwngp+2*nnnp-1-jngp
    ngp1 = iw(k(jwngp))
    if (ngp1.eq.0) go to 25
    jigmn = jigmn-1
    jigmx = jigmx-ngp1
    do 20 jigp=jigmn,jigmx
        jwigr = jwigr+jigp
        iz = iw(k(jwigr))
        iy = (iz-1)/nxio+1
        ix = iz-nxio*(iy-1)
        call i0hsf(xd,yd,zd,nt,iwk(jwftp),nl,iwk(jwipl),wk,iti,
        x(iy),y(iy),z(iy,iy))
    20 continue
25 continue
    go to 9005
30 ier = 129
9000 continue
    call uertst (ier,6hiqmsc)
9005 return
end

integer function i0hsd (x,y,i1,i2,i3,i4)
integer i1,i2,i3,i4
real x(1),y(1)
integer idx
real a150, a250, a350, a450, b150, b250, b350, b450, c150,
    a250, a350, c450, epsln, s150, s250, s350, s450, tol,
    u1, u2, u3, u4, x1, x2, x3, x4, y1, y2, y3, y4
1 equivalence (c25, c150), (a350, b250), (b350, a150),
2 equivalence
(A450,B150),(B450,A250),(C450,C350)
TOL/164140000000000000000000

DATA
TOL = .71054E-14
EPSLN = TOL*100.0
X1 = X(I1)
Y1 = Y(I1)
X2 = X(I2)
Y2 = Y(I2)
X3 = X(I3)
Y3 = Y(I3)
X4 = X(I4)
Y4 = Y(I4)
IDX = 0
U3 = (Y2-Y3)*(X1-X3)-(X2-X3)*(Y1-Y3)
U4 = (Y1-Y4)*(X2-X4)-(X1-X4)*(Y2-Y4)
IF (U3*U4.LE.0.0) GO TO 5
U1 = (Y3-Y1)*(X4-X1)-(X3-X1)*(Y4-Y1)
U2 = (Y4-Y2)*(X3-X2)-(X4-X2)*(Y3-Y2)
A150 = (X1-X3)**2+(Y1-Y3)**2
B150 = (X4-X1)**2+(Y4-Y1)**2
C150 = (X3-X4)**2+(Y3-Y4)**2
A250 = (X2-X4)**2+(Y2-Y4)**2
B250 = (X3-X2)**2+(Y3-Y2)**2
C250 = (X2-X1)**2+(Y2-Y1)**2
S15Q = U1*U1/(C150*A15Q(A15Q,B15Q))
S25Q = U2*U2/(C250*A25Q(A25Q,B25Q))
S35Q = U3*U3/(C350*A35Q(A35Q,B35Q))
S45Q = U4*U4/(C450*A45Q(A45Q,B45Q))
IF ((A150(1350,450)-A150(1350,2350)).GT.EPSLN) IDX = 1
9 RETURN
END
SUBROUTINE IOMSE (NDP,XD,JD,NT,IPT,PD,WK)
INTEGER NDP,NT,IPT(1),
REAL XD(1),JD(1),PD(1),WK(1)
INTEGER IDP,IPT(3),IT,I,JPDQ,JPDMX,JPD,JPT,JPTO,NDPO;
1 NTO = 107
2 DATA
TOL = .71054E-14
EPSLN = TOL*100.0
NDPO = NDP
NTO = NT
JPDMX = 5*NDPO
DO 5 JPD=1,JPDMX
PD(JPD) = 0.0
5 CONTINUE
DO 10 IDP=1,NDP
     WK(IDP) = 0.0
10 CONTINUE
DO 25 IT=1,NTO
    JPTO = 3*(IT-1)
    DO 15 IV=1,3
        JPT = JPTO+IV
        IDP = IPTY(JPT)
        IPTI(IV) = IDP
        XV(IV) = XD(IDP)
        YV(IV) = YD(IDP)
        ZV(IV) = ZD(IDP)
15 CONTINUE
    DX1 = XV(2)-YV(1)
    DY1 = YV(2)-YV(1)
    DX2 = XV(3)-XV(1)
    DY2 = YV(3)-YV(1)
    DZ2 = ZV(3)-ZV(1)
    VPX = D1*DX2-DZ1*DYZ
    VPY = D1*DX2-DZ1*DZ2
    VPZ = D1*DY2-D1*DX2
    VPMN = ABS(DX1*DX2-DY1*DY2)*EPSLN
    IF (ABS(VPX).LE.VPMN) GO TO 25
    DO 20 IV=1,3
        IDP = IPTI(IV)
        JPDO = 5*(IDP-1)+1
        PD(JPDO) = PD(JPDO)+VPX
        PD(JPDO+1) = PD(JPDO+1)+VPY
        WK(IDP) = WK(IDP)+VPZ
20 CONTINUE
25 CONTINUE
DO 30 IDP=1,NDPO
    JPDO = 5*(IDP-1)+1
    PD(JPDO) = -PD(JPDO)/WK(IDP)
    PD(JPDO+1) = -PD(JPDO+1)/WK(IDP)
30 CONTINUE
DO 45 IT=1,NTO
    JPTO = 3*(IT-1)
    DO 35 IV=1,3
        JPT = JPTO+IV
        IDP = IPTY(JPT)
        IPTI(IV) = IDP
        XV(IV) = XD(IDP)
        YV(IV) = YD(IDP)
        JPDO = 5*(IDP-1)+1
        ZD(JV) = PD(JPDO)
        ZV(IV) = PD(JPDO+1)
35 CONTINUE

NTL = NT+NL
IF (ILO.LE.NTL) GO TO 5
IL1 = ILO/NTL
IL2 = ILO-IL1*NTL
IF (IL1.EQ.IL2) GO TO 30
GO TO 55
5 IF (ILO.EQ.IPV) GO TO 25
JIPT = 3*(ILO-1)
JPD = 0
DO 15 I=1,3
JIPT = JIPT+1
IDP = JIPT(JIPT)
X(I) = XD(IDP)
Y(I) = YD(IDP)
Z(I) = ZD(IDP)
JPDD = 5*(IDP-1)
DO 10 KPD=1,5
JPD = JP0+1
JPDD = JPDD+1
PD(JPD) = PDD(JPDD)
10 CONTINUE
15 CONTINUE
X0 = X(1)
YO = Y(1)
A = X(2)-X0
B = X(3)-X0
C = Y(2)-YO
D = Y(3)-YO
AD = A*D
BC = B*C
DLT = AD-BC
AP = D/DLT
BP = -3/DLT
CP = -C/DLT
DP = A/DLT
AA = A*A
ACT2 = 2.0*A*C
CC = C*C
AA = A*B
ADBC = AD*BC
CD = C*D
BB = B*B
BDT2 = 2.0*B*D
DD = D*D
DO 20 I=1,3
JP0 = 5*I
20 ZU(I) = A*PD(JPD-4)+C*PD(JPD-3)
ZV(I) = B*PD(JPD-4)+D*PD(JPD-3)
ZU(I) = AA*PD(JPD-2)+ACT2*PD(JPD-1)+C*C*PD(JPD)
IOM 236
IOM 237
IOM 238
IOM 239
IOM 240
IOM 241
IOM 242
IOM 243
IOM 244
IOM 245
IOM 246
IOM 247
IOM 248
IOM 249
IOM 250
IOM 251
IOM 252
IOM 253
IOM 254
IOM 255
IOM 256
IOM 257
IOM 258
IOM 259
IOM 260
IOM 261
IOM 262
IOM 263
IOM 264
IOM 265
IOM 266
IOM 267
IOM 268
IOM 269
IOM 270
IOM 271
IOM 272
IOM 273
IOM 274
IOM 275
IOM 276
IOM 277
IOM 278
IOM 279
IOM 280
IOM 281
IOM 282
IOM 283
IOM 284
```
 JPDD = 56*(DPD-1)
 DO 60 KPD=1,5
     JPDD = JPDD+1
     PD(KPD) = PDD(JPDD)
 60 CONTINUE
 P00 = Z0
 P10 = PD(1)
 P20 = 0.5*PD(2)
 P30 = 0.5*PD(3)
 P40 = PD(4)
 P50 = 0.5*PD(5)
 ITPV = ITO
 65 U = XX-Y0
   V = YI-Y0
   PO = POO+V*(PO1+V*PO2)
   P1 = P10+V*P11
   ZII = PO+U*(P1+U*P20)
   RETURN
END
```
DSQI = DSQF(X1,Y1,YD(IP2),YD(IP2))
IF (DSQI.EQ.0.0) GO TO 160
IF (DSQI.GE.DSQMN) GO TO 9
DSQMN = DSQI
IPMN1 = IP1
IPMN2 = IP2
5 CONTINUE
10 CONTINUE
XQMP = (XD(IPMN1)+XD(IPMN2))/2.0
YQMP = (YD(IPMN1)+YD(IPMN2))/2.0
JP1 = 2
DO 15 JP1=1,NDPO
   IF (IP1.EQ.IP1.OR.IP1.EQ.IP1) GO TO 15
   JP1 = JP1+1
   IWP(JP1) = IP1
   WK(JP1) = DSQF(XQMP,YQMP,XD(IP1),YD(IP1))
15 CONTINUE
DO 20 JP1=3,NDPM1
   DSQMN = WK(JP1)
   JPMN = JP1
   DO 20 JP2=1,NDPO
      IF (WK(JP2).GE.DSQMN) GO TO 20
      DSQMN = WK(JP2)
   JPMN = JP2
20 CONTINUE
ITS = IWP(JP1)
IWP(JP1) = IWP(JPMN)
IWP(JPMN) = ITS
WK(JPMN) = WK(JP1)
25 CONTINUE
X1 = XD(IPMN1)
Y1 = YD(IPMN1)
X2 = XD(IPMN2)
Y2 = YD(IPMN2)
DO 30 JP=1,NDPO
   IP = IWP(IP)
   SP = SPT(DX(IP),YD(IP),X1,Y1,X2,Y2)
   VP = VPT(DX(IP),YD(IP),X1,Y1,X2,Y2)
   IF (ABS(VP).GT.(ABS(SP)+EPSL4)) GO TO 35
30 CONTINUE
GO TO 165
35 IF (JP.EQ.3) GO TO 45
   JPMX = JP
   DO 40 JPC=1,JPMX
      JP = JPMX+1-JPC
      IWP(JP) = IWP(JP-1)
40 CONTINUE
IWP(3) = IP
45 IP1 = IPMN1
IP2 = IPMN2
IP3 = IWP(3)
IF (VPDT(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3)).GE.0.0)
  GO TO 50
IP1 = IPMN1
IP2 = IPMN2
50 MTO = 1
MNT3 = 3
IPT(1) = IP1
IPT(2) = IP2
IPT(3) = IP3
NLO = 3
NLT3 = 9
IPL(1) = IP1
IPL(2) = IP2
IPL(3) = 1
IPL(4) = IP2
IPL(5) = IP3
IPL(6) = 1
IPL(7) = IP3
IPL(8) = IP1
IPL(9) = 1
DO 150 JP1=4, NDPO
IP1 = IWP(JP1)
X1 = XD(IP1)
Y1 = YD(IP1)
D0 65 IL=1,NLO
  IP2 = IP(3*IL-2)
  IP3 = IP(3*IL-1)
  X2 = XD(IP2)
  Y2 = YD(IP2)
  X3 = XD(IP3)
  Y3 = YD(IP3)
  SP = SPDT(X1,Y1,X2,Y2,X3,Y3)
  VP = VPDT(X1,Y1,X2,Y2,X3,Y3)
  IF (IL.NE.1) GO TO 55
  IXVS = 0
  IF (VP.LE.(ABS(SP)*(-EPSL)) ) IXVS = 1
  ILIV = 1
  ILV5 = 1
  GO TO 65
150 IXVSPV = IXVS
  IF (VP.GT.(ABS(SP)*(-EPSL)) ) GO TO 60
  IXVS = 1
  IF (IXVSPV.EQ.1) GO TO 65
  ILV5 = IL
  IF (ILIV.NE.1) GO TO 70
  GO TO 65
60 IXVS = 0
IF (IXVSPV.EQ.0) GO TO 65
ILIV = IL
IF (ILVS.NE.1) GO TO 70
65 CONTINUE
IF (ILIV.EQ.1.AND.ILVS.EQ.1) ILVS = NLO
70 IF (ILVS.LT.ILIV) ILVS = ILVS+NLO
IF (ILIV.EQ.1) GO TO 65
NLST = TLIV-1
NLST3 = NLST+3
DO 75 JL1=1,NLST3
JL2 = JL1+NLST3
IPL(JL2) = IPL(JL1)
75 CONTINUE
DO 80 JL1=1,NLST3
JL2 = JL1+NLST3
IPL(JL1) = IPL(JL2)
80 CONTINUE
ILVS = ILVS-NLST
85 JLW = 0
DO 105 JLW=1,ILVS-NLST
ILT = IL*N3
IPL1 = IPL(ILT-2)
IPL2 = IPL(ILT-1)
IT = IPL(ILT)
NTO = NTO+1
NLST3 = NLST+3
IPT(NLST3-2) = IPL2
IPT(NLST3-1) = IPL1
IPT(NLST3) = IP1
IF (ILW.NE.ILVS) GO TO 90
IPL(ILT-1) = IP1
IPL(ILT) = NTO
IF (ILW.NE.NLST) GO TO 95
NLN = ILVS+1
NLNT3 = NLN+3
IPL(NLNT3-2) = IP1
IPL(NLNT3-1) = IPL1
IPL(NLNT3) = NTO
90 II = IT-1
IPTF = IPT(II-3)
IF (IPTF.NE.IPLF.AND.IPTF.NE.IPL2) GO TO 100
IPTF = IPT(II-1)
IPTF = IPT(II-1)
IF (IPTF.NE.IPLF.AND.IPTF.NE.IPL2) GO TO 100
IPTF = IPT(II-1)
100 IF (ILV5.EQ.0) GO TO 105
IPTF(II-2) = IPTF
IPTF(II-1) = IPTF
IPTF(II-3) = IP1
105
IF (IL.EQ.ILVS) IPL(IIT3) = IT
IF (IL.EQ.NLO.AND.IPL(IIT3).EQ.IT) IPL(3) = NTO
JWL = JWL*4
IWL(JWL-3) = IPL1
IWL(JWL-2) = IPT1
IWL(JWL-1) = IPT2
IWL(JWL) = IPL2
CONTINUE
NLO = KLN
NLT3 = KLNT3
NFL = JWL/2
IF (NFL.EQ.0) GO TO 150
NTT3P3 = NTT3+3
DO 145 IREP=1,NREP
   DO 135 IIF=1,NLF
      IPL1 = IWL(2*IF-1)
      IPL2 = IWL(2*IF)
      NTF = 0
      DO 110 ITT3=3,NTT3+3
         ITT3 = NTT3P3-ITT3R
         IPT1 = IPT(ITT3-2)
         IPT2 = IPT(ITT3-1)
         IPT3 = IPT(ITT3)
         IF (IPL1.NE.IPT1.AND.IPL1.NE.IPT2.AND.IPL1.NE.IPT3) GO TO 1
         TO 110
      IF (IPL2.NE.IPT1.AND.IPL2.NE.IPT2.AND.IPL2.NE.IPT3) GO TO 1
      TO 110
      NTF = NTF+1
      IF (NTF) = (ITT3/3)
      IF (NTF.EQ.2) GO TO 115
   CONTINUE
   IF (NTF.LT.2) GO TO 135
   ITT3 = ITT((1)*9
   IPT1 = IPT(ITT3-2)
   IF (IPT1.NE.IPL1.AND.IPT1.NE.IPL2) GO TO 120
   IPT1 = IPT(ITT3-1)
   IF (IPT1.NE.IPL1.AND.IPT1.NE.IPL2) GO TO 120
   IPT1 = IPT(ITT3)
   ITT2 = ITF((2)*3
   IPT2 = IPT(ITT2-2)
   IF (IPT2.NE.IPL1.AND.IPT2.NE.IPL2) GO TO 125
   IPT2 = IPT(ITT2-1)
   IF (IPT2.NE.IPL1.AND.IPT2.NE.IPL2) GO TO 125
   IPT2 = IPT(ITT2)
   ITT3 = ITF((3)*9
   IPT3 = IPT(ITT3-2)
   IF (IPL1.NE.IPL1.AND.IPL1.NE.IPL2).EQ.0) GO TO 135
   IPT(ITT3-2) = IPT1
   IPT(ITT3-1) = IPT2
   IPT(ITT3) = IPT1
   IPT(ITT2-2) = IPT1
   IPT(ITT2-1) = IPT2
   IPT(ITT2) = IPT1
   IPT(ITT3) = IPT1
   IPT(ITT2) = IPT1
IPT(I2(T3-1) = IPT(I1)
IPT(I2(T3) = IPT(I2)
JWL = JLW+8
IWL(JWL-7) = IPT(I)
IWL(JWL-6) = IPT(I)
IWL(JWL-5) = IPT(I)
IWL(JWL-4) = IPT(I)
IWL(JWL-3) = IPT(I)
IWL(JWL-2) = IPT(I)
IWL(JWL-1) = IPT(I)
IWL(JWL) = IPT(I)
DO 130 JLT3=3,NL(T3,3)
   IPTJ1 = IPT(JLT3-2)
   IPTJ2 = IPT(JLT3-1)
   IF ((IPLJ1.EQ.IPL1.AND.IPLJ2.EQ.IPTI2).OR.(IPLJ2.EQ.IPL1)) IPT(JLT3)
   L1, AND.IPLJ1.EQ.IPTI2) IPT(JLT3)
   IF ((IPLJ2.EQ.IPL1).AND.(IPLJ1.EQ.IPTI1).OR.(IPLJ1.EQ.IPL1)) IPT(JLT3)
1   L2, AND.IPLJ1.EQ.IPTI1) IPT(JLT3)
2   = ITF(2)
130 CONTINUE
135 CONTINUE
   NLFC = NLFC
   NLF = JLW/2
   IF (NLFC.EQ.NLFC) GO TO 150
   JLW1MN = 2*NLFC+1
   NLFT2 = NLF+2
   DO 140 JLW1=JWL1MN,NLF+2
      JLW = JLW1+1-JWL1MN
      IWL(JWL) = IWL(JWL)
140 CONTINUE
145 CONTINUE
150 CONTINUE
   DO 155 I(T3)=3,NT(T3,3)
      IP1 = IPT(I(T3-2)
      IP2 = IPT(I(T3-1)
      IP3 = IPT(I(T3)
      IF ((VP(DT(XD(IP1),XD(IP2),XD(IP3),YD(IP3))GE.0))
1      IP(T(T3-2) = IP2
      IP(T(T3-1) = IP1
155 CONTINUE
   NT = NTO
   NL = NLO
   RETURN
160 IER = 131
   RETURN
165 IER = 130
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RETURN</td>
<td>END SUBROUTINE IQSH</td>
</tr>
<tr>
<td>2</td>
<td>INTEGER</td>
<td>NT,NL,NX1,NY1,IPT(1),IPL(1),NGP(1),IGP(1)</td>
</tr>
<tr>
<td>3</td>
<td>REAL</td>
<td>XD(1),YD(1),XI(1),YI(1)</td>
</tr>
<tr>
<td>4</td>
<td>INTEGER</td>
<td>ILOT3,IL0,ILP13,ILP1,INSD,IP1,IP2,IP3,ITOT3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>ITO,XIMN,XIMX,XI,YI,Y1,Y2,Y3,YY1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>JNGPO,JNGP1,L,NGPO,NGP1,NLO,NT0,NXIO,NXINY1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>NYIO</td>
</tr>
<tr>
<td>8</td>
<td>REAL</td>
<td>X1,X2,X3,X11,XIMN,XIMX,XMN,XMX,V1,Y2,Y3,YY1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>YIMN,YIMX,YMN,YMX</td>
</tr>
<tr>
<td>10</td>
<td>REAL</td>
<td>SPDT,VPD,T,U1,U2,U3,V1,V2,V3</td>
</tr>
<tr>
<td>11</td>
<td>SPDT</td>
<td>(U1-U2)<em>(U3-U2)+(V1-V2)</em>(V3-V2)</td>
</tr>
<tr>
<td>12</td>
<td>VPD</td>
<td>(U1-U3)<em>(V2-V3)-(V1-V3)</em>(U2-U3)</td>
</tr>
<tr>
<td>13</td>
<td>NTO</td>
<td>NT</td>
</tr>
<tr>
<td>14</td>
<td>NLO</td>
<td>NL</td>
</tr>
<tr>
<td>15</td>
<td>NXIO</td>
<td>NXI</td>
</tr>
<tr>
<td>16</td>
<td>NYIO</td>
<td>NYI</td>
</tr>
<tr>
<td>17</td>
<td>NXINY1</td>
<td>NXIO*NYIO</td>
</tr>
<tr>
<td>18</td>
<td>XIMN</td>
<td>AMIN1(XI(1),XI(NXIO))</td>
</tr>
<tr>
<td>19</td>
<td>XIMX</td>
<td>AMAX1(XI(1),XI(NXIO))</td>
</tr>
<tr>
<td>20</td>
<td>YIMN</td>
<td>AMIN1(YI(1),YI(NYIO))</td>
</tr>
<tr>
<td>21</td>
<td>YIMX</td>
<td>AMAX1(YI(1),YI(NYIO))</td>
</tr>
<tr>
<td>22</td>
<td>JNGP0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>JNGP1</td>
<td>2*(NTO+2*NLO)+1</td>
</tr>
<tr>
<td>24</td>
<td>JGPO</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>JGP1</td>
<td>NXINY1+1</td>
</tr>
<tr>
<td>26</td>
<td>DO 80</td>
<td>ITO=1,NT0</td>
</tr>
<tr>
<td>27</td>
<td>80</td>
<td>NGPO=0</td>
</tr>
<tr>
<td>28</td>
<td>ITO3</td>
<td>ITO=3</td>
</tr>
<tr>
<td>29</td>
<td>IP1</td>
<td>IPT(ITOT3-2)</td>
</tr>
<tr>
<td>30</td>
<td>IP2</td>
<td>IPT(ITOT3-1)</td>
</tr>
<tr>
<td>31</td>
<td>IP3</td>
<td>IPT(ITOT3)</td>
</tr>
<tr>
<td>32</td>
<td>X1</td>
<td>XD(IP1)</td>
</tr>
<tr>
<td>33</td>
<td>Y1</td>
<td>YD(IP1)</td>
</tr>
<tr>
<td>34</td>
<td>X2</td>
<td>XD(IP2)</td>
</tr>
<tr>
<td>35</td>
<td>Y2</td>
<td>YD(IP2)</td>
</tr>
<tr>
<td>36</td>
<td>EN</td>
<td>XD(IP3)</td>
</tr>
<tr>
<td>37</td>
<td>V3</td>
<td>YD(IP3)</td>
</tr>
<tr>
<td>38</td>
<td>XMN</td>
<td>AMIN1(X1,X2,X3)</td>
</tr>
<tr>
<td>39</td>
<td>XMX</td>
<td>AMAX1(X1,X2,X3)</td>
</tr>
<tr>
<td>40</td>
<td>YMN</td>
<td>AMIN1(Y1,Y2,Y3)</td>
</tr>
<tr>
<td>41</td>
<td>YMX</td>
<td>AMAX1(Y1,Y2,Y3)</td>
</tr>
<tr>
<td>42</td>
<td>INSD</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>DO 10</td>
<td>IX1=1,NXIO</td>
</tr>
<tr>
<td>44</td>
<td>10</td>
<td>IF (IX1.GE.XMN.AND.XI(IX1).LE.XMX) GO TO 5</td>
</tr>
<tr>
<td>45</td>
<td>IF (INSD.EQ.0) GO TO 10</td>
<td>770</td>
</tr>
<tr>
<td>46</td>
<td>IXIMX</td>
<td>IXX1-1</td>
</tr>
<tr>
<td>47</td>
<td>IXX1</td>
<td>IXX1+1</td>
</tr>
<tr>
<td>48</td>
<td>IXX2</td>
<td>IXX2+1</td>
</tr>
<tr>
<td>49</td>
<td>IXX3</td>
<td>IXX3+1</td>
</tr>
</tbody>
</table>
GO TO 15
   IF (INSO.EQ.1) GO TO 10
   INSD = 1
   IXIMN = IXI
10 CONTINUE
   IF (INSO.EQ.0) GO TO 75
   IXIMX = NXIO
15 DO 70 IYI=1,NYIO
   YII = YII(IYI)
   IF (YII.LT.YMNM.DR.YII.GT.YMX) GO TO 70
   DO 65 IXI=IXIMN,IXIMX
       XIIX = XIIX(IXI)
L = 0
   IF (VPDOT(X1,Y1,X2,Y2,XIIX,YII)) 65,20,25
20 L = 1
   IF (VPDOT(X2,Y2,X3,Y3,XIIX,YII)) 65,30,35
30 L = 1
   IF (VPDOT(X3,Y3,X1,Y1,XIIX,YII)) 65,40,45
45 IIZ = NXIO*(IYI-1)+IXI
   IF (L.EQ.1) GO TO 30
   NGPO = NGPO+1
   JIGPO = JIGPO+1
   IGP(JIGPO) = IIZ
   GO TO 65
50 IF (JIGPO.GT.NXINYI) GO TO 60
   DO 55 JIGPI=JIGPL,NXINYI
       IF (IIZ.EQ.IGP(JIGPI)) GO TO 65
55 CONTINUE
60 NGPI = NGPI+1
   JIGPI = JIGPI+1
   IGP(JIGPI) = IIZ
65 CONTINUE
70 CONTINUE
75 JMGPO = JMGPO+1
   NGP(JMGPO) = NGPO
   JMGPI = JMGPI+1
   NGP(JMGPI) = NGPI
80 CONTINUE
70 DO 225 ILO=1,NLO
   NSPO = 0
   NGPI = 0
   ILO3 = ILO+3
   IP1 = IPL(ILO3-2)
   IP2 = IPL(ILO3-1)
   X1 = X(IP1)
   Y1 = Y(IP1)
   X2 = X(IP2)
   Y2 = Y(IP2)
225 CONTINUE
IOM 823
123

JNGP1 = JNGP1-1
NGP1(JNGP1) = NGP1
NGP0 = 0
NGP1 = 0
ILP1 = MOD(ilo,NL0)+1
ILPIT3 = ILP1*3
IP3 = IPL(ILPIT3-1)
X3 = XD(IP3)
YW = YD(IP3)
XMM = XIMN
XMX = XIMX
YMN = YIMN
YMX = YIMX
IF (Y3.GE.Y2.AND.Y2.GE.Y1) XMM = X2
IF (Y3.LE.Y2.AND.Y2.LE.Y1) XMX = X2
IF (X3.GE.X2.AND.X2.GE.X1) YMN = Y2
IF (X3.LE.X2.AND.X2.LE.X1) YMX = Y2
INSD = 0
DO 165 IXI=1,NX10
   IF (X1(XI).GE.XMN.AND.X1(XI).LE.XMX) GO TO 160
   IF (INSD.EQ.0) GO TO 165
   IXIMX = IXI-1
   GO TO 170
160  IF (INSD.EQ.1) GO TO 165
    INS = 1
    IXIMN = IXI
165  CONTINUE
    IF (INSD.EQ.0) GO TO 220
    IXIMX = NX10
170  DO 215 IYI=1,NY10
YII = YI(IYI)
   IF (YII.LT.YMN.OR.YII.GT.YMX) GO TO 215
   DO 210 IXI=IXIMN,IXIMX
      X11 = X1(XI)
      L = 0
      IF (SPDT(X1,Y1,X2,Y2,X11,YII)) 180,175,210
      L = 1
175  IF (SPDT(X3,Y3,X2,Y2,X1,YII)) 190,185,210
180  L = 1
185  L = 1
190  IZI = NX10*(IYI-1)+XI
   IF (L.EQ.1) GO TO 199
   NGP0 = NGPO+1
   JIGPO = JIGPO+1
   IGP(JIGPO) = IZI
   GO TO 210
   IF (JIGP1.GT.NXY1) GO TO 205
   DO 200 JIGP1=JIGP1,NX1Y1
      IF (IZI.EQ.IGP(JIGP1)) GO TO 210
   200  CONTINUE
205  NGP1 = NGP1 + 1
    JIGP1 = JIGP1 - 1
    IGR(JIGP1) = I11
210  CONTINUE
215 CONTINUE
220  JNGPo = JNGPo + 1
    NGP(JNGPo) = NGPo
    JNGP1 = JNGP1 - 1
    NGP(JNGP1) = NGP1
225 CONTINUE
RETURN
END
SUBROUTINE UERST (IER, NAME)
INTEGER IER, NAME
INTEGER NAMESET, NAMEO
DATA NAMESET, NAMEO / 6, 6 /
DATA NAMEO(6) / 0, 0, 0, 0, 0, 0 /
DATA LEVEL/4/, IFQDF/0/, IEQ/1H=/
IF (IER .GT. 999) GO TO 25
IF (IER.LT.-32) GO TO 55
IF (IER.LE.128) GO TO 5
IF (LEVEL.LT.1) GO TO 30
CALL UGETIO(1, NIN, IUNIT)
IF (IEQDF.EQ.1) WRITE(IUNIT,35) IER, NAMEO, IEQ, NAME
IF (IEQDF.EQ.0) WRITE(IUNIT,35) IER, NAME
GO TO 30
5 IF (ISP.LE.64) GO TO 10
IF (LEVEL.LT.2) GO TO 30
CALL UGETIO(1, NIN, IUNIT)
IF (IEQDF.EQ.1) WRITE(IUNIT,40) IER, NAMEO, IEQ, NAME
IF (IEQDF.EQ.0) WRITE(IUNIT,40) IER, NAME
GO TO 30
10 IF (IER.LE.32) GO TO 15
IF (LEVEL.LE.3) GO TO 30
CALL UGETIO(1, NIN, IUNIT)
IF (IEQDF.EQ.1) WRITE(IUNIT,45) IER, NAMEO, IEQ, NAME
IF (IEQDF.EQ.0) WRITE(IUNIT,45) IER, NAME
GO TO 30
15 CONTINUE
IF (NAME.NE.NAMSET) GO TO 25
LEV0LD = LEVEL
LEVEL = IER
IF = LEV0LD
IF (LEVEL.LE.0) LEVEL = 4
IF (LEVEL.GT.4) LEVEL = 4
GO TO 30
25 CONTINUE
IF (LEVEL.LE.4) GO TO 30
CALL UGETIO(1, NIN, IUNIT)
IF (IEOOF.EQ.1) WRITE(IUNIT,50) IER,NAMEx,IER,NAME
IF (IEOOF.EQ.0) WRITE(IUNIT,50) IER,NAME
30 IEOOF = 0
RETURN
35 FORMAT(14H *** TERMINAL ERROR,10X,7H IER = $I3$, IBER = $I3$)
6 15H) FROM ROUTINE, A6, A1, A6
40 FORMAT(36H *** WARNING WITH FIX ERROR (IER = $I3$, $I3$)) FROM ROUTINE, A6, A1, A6
45 FORMAT(11H *** WARNING ERROR,11X,7H IER = $I3$, $I3$) FROM ROUTINE, A6, A1, A6
$ 15H) FROM ROUTINE, A6, A1, A6
50 FORMAT(20H *** UNDEFINED ERROR,9X,7H IER = $I5$, $I5$)
$ 15H) FROM ROUTINE, A6, A1, A6
55 IEOOF = 1
NAME = NAME
65 RETURN
END
SUBROUTINE UGETIO(LOPT, NIN, NOUT)
INTEGER IOPT, NIN, NOUT
INTEGER NIND, NOUTD
DATA NIND/5/NOUNP/NOUTD/6/LUNPUT/
IF (LOPT.EQ.3) GO TO 10
IF (LOPT.EQ.2) GO TO 5
IF (LOPT.NE.1) GO TO 9005
NIN = NIND
NOUT = NOUTD
GO TO 9005
9005 RETURN
END
SUBROUTINE SUTS(NX, Y, XL, XU, SIGMA, IENDSW, END, IW, PROXIN, WK, IERR)
SUTS 2
C DO 11
C
C***************************************************************************
SUTS 5
C*
C* PURPOSE - TO COMPUTE THE INTEGRAL OF A CUBIC SPLINE*
C* INTERPOLATION UNDER TENSION TO A SET OF*
C* DISCRETE NODES OF INDEPENDENT AND DEPENDENT*
C* REAL VARIABLES.
C*
C* USE - CALL SUTS(NX, Y, XL, XU, SIGMA, IENDSW, END, IW, PROXIN, WK, IERR)
C*
C* PARAMETERS N - AN INPUT INTEGER SPECIFYING THE NUMBER OF*
C* OF NODES OR VALUES FOR BOTH THE INPUT*
C* INDEPENDENT AND DEPENDENT VARIABLES. N > 1.*
C*
C* X - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *
N CONTAINING THE INDEPENDENT VARIABLES OR *SUTS
X-COORDINATES OF THE NODES. X MUST BE A *SUTS
STRICTLY INCREASING ARRAY. *SUTS
Y - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *SUTS
N CONTAINING THE DEPENDENT VARIABLES OR *SUTS
Y-COORDINATES OF THE NODES. Y MUST BE A *SUTS
STRICTLY INCREASING ARRAY. *SUTS
SIGMA - AN INPUT REAL NUMBER CONTAINING THE TENSION *SUTS
XL - AN INPUT REAL NUMBER SPECIFYING THE LOWER *SUTS
LIMIT OF INTEGRATION. *SUTS
XU - AN INPUT REAL NUMBER SPECIFYING THE UPPER *SUTS
LIMIT OF INTEGRATION. *SUTS
FACTOR, SIGMA SHOULD BE NON-NEGATIVE. *SUTS
IENDSW - A ONE-DIMENSIONAL INPUT INTEGER ARRAY OF *SUTS
LENGTH 2 SPECIFYING CONDITIONS ON THE CURVE *SUTS
AT THE TWO ENDPOINTS. IENDSW(1) REFERS TO *SUTS
(X(1), Y(1)) AND IENDSW(2) REFERS TO *SUTS
(X(N), Y(N)). *SUTS
IENDSW(1) = 0 MEANS THAT THE USER WILL INPUT *SUTS
A VALUE FOR Y' AT THE *SUTS
APPROPRIATE ENDPOINT IN *SUTS
END(1) = *SUTS
IENDSW(1) = 1 MEANS THAT Y' WILL BE ESTIMATED *SUTS
INTERNALLY FOR THE APPROPRIATE *SUTS
ENDPOINT. *SUTS
IENDSW(1) = 2 MEANS THAT Y' = ZERO (I.E., THE *SUTS
NATURAL END-CONDITION HOLDS) AT *SUTS
THE APPROPRIATE ENDPOINT. *SUTS
END - A ONE-DIMENSIONAL INPUT REAL ARRAY OF LENGTH *SUTS
2 CONTAINING DESIRED VALUES FOR Y' AT *SUTS
(X(1), Y(1)) IN END(1) AND AT (X(N), Y(N)) *SUTS
IN END(2). IF IENDSW(1) .NE. 0, THEN *SUTS
END(1) NEED NOT BE DEFINED. IF BOTH *SUTS
IENDSW(1) AND IENDSW(2) ARE NON-ZERO, THEN *SUTS
END MAY BE A DUMMY PARAMETER. *SUTS
IW - AN INPUT/OUTPUT INTEGER. *SUTS
INPUT - IF THE USER IS INPUTTING X, Y, *SUTS
N, SIGMA, IENDSW, AND END FOR *SUTS
THE FIRST TIME (AS A PACKAGE), THEN *SUTS
IW SHOULD BE SET TO 0. OTHERWISE, *SUTS
IW SHOULD BE SET TO 1. *SUTS
OUTPUT - IW IS AUTOMATICALLY SET TO 1. *SUTS
PROXIN - AN OUTPUT REAL NUMBER SPECIFYING THE VALUE OF *SUTS
THE INTEGRAL. *SUTS
WK - A ONE-DIMENSIONAL INPUT/OUTPUT/WORK REAL *SUTS
ARRAY OF LENGTH 2N - 1. *SUTS
INPUT - IF IW = 1, THEN THE FIRST N *SUTS
PLACES OF WK MUST BE OUTPUT FROM *SUTS
A PREVIOUS CALL TO STIUNI (E1.0), *SUTS
SUTER (04.3), OR SUTS (01.11). *SUTS 69
OUTPUT - THE FIRST N PLACES OF WK CONTAIN *SUTS 70
DERIVATIVE INFORMATION NECESSARY ON *SUTS 71
SUCCEEDING CALLS TO STUNI ; *SUTS 72
SUTER , OR SUTS USING THE SAME *SUTS 73
INPUT DATA. *SUTS 74
WORK - THE REST OF WK IS WORK STORAGE. *SUTS 75
IERR - AN OUTPUT INTEGER ERROR CODE. *SUTS 76
* 0 NORMAL RETURN. *SUTS 77
* 1 N < 2. *SUTS 78
* 2 X IS NOT AN INCREASING ARRAY. *SUTS 79
* 3 X IS A NEGATIVE ARRAY. *SUTS 80
* 4 X IS A REPEATED ARRAY. *SUTS 81
* 5 X IS A DECREASING ARRAY. *SUTS 82
* 6 X IS A ZERO ARRAY. *SUTS 83
PRECISION - SINGLE *SUTS 84
* 8 X IS A POSITIVE ARRAY. *SUTS 85
* 9 X IS A ONE ARRAY. *SUTS 86
REQUIRED ROUTINES - CEEZ,CURVI,CURVIN,CURVZ,INTRVL,SNHCSM,TERMS. *SUTS 87
DATE RELEASED - MARCH 1, 1979. *SUTS 88
LANGUAGE - FORTRAN *SUTS 89
LATEST REVISION - NONE *SUTS 90

FORMAL PARAMETERS

INTEGER IENSOW(2),IERR,IW,N
REAL ENO(2),PROXIN,SIENA,WK(1),X(1),XL,XU,Y(1)

INTERNAL VARIABLES

REAL CURVI,DELM,SIENAP,ZERO

CONSTRUCT THE CONSTANT ZERO

DATA ZERO/0.000/
DATA IUNS /0/
IUNS = (IENSOW(2) + 2 * IENSOW(1))
IUNS = 1

INITIALIZE IERR AND CHECK ERROR RETURNS

IERR = 1
IF (N .LT. 2) GO TO 9000
IERR = 2
DELX = X(N) - X(1)
IF (DELX .LE. ZERO) GO TO 9000

SUTS 91
SUTS 92
SUTS 93
SUTS 94
SUTS 95
SUTS 96
SUTS 97
SUTS 98
SUTS 99
SUTS 100
SUTS 101
SUTS 102
SUTS 103
SUTS 104
SUTS 105
SUTS 106
SUTS 107
SUTS 108
SUTS 109
SUTS 110
SUTS 111
SUTS 112
SUTS 113
SUTS 114
SUTS 115
SUTS 116
SUTS 117
IERR = 0

DENORMALIZE THE TENSION FACTOR SIGMA

SIGMAP = ABS(SIGMA) * (N - 1)/DELX

CALL CURVI IF NECESSARY

IF (IW .EQ. 1) GO TO 10

IF (N .GT. 2) CALL CURVIN(N,X,Y,IENDSW,END,SIGMAP,WK,WK(N+1),IERR)

IF (N .EQ. 2) CALL CURVI2(DELX,Y,IENDSW,END,SIGMAP,WK)

IF (IERR .NE. 0) GO TO 9000

SET IW = 1 FOR OUTPUT

IW = 1

COMPUTE THE INTEGRAL PROXIN

PROXIN = CURVI(XL,XU,N,X,Y,WK,SIGMAP)

9000 RETURN

END

SUBROUTINE CURVIN(N,X,Y,IENDSW,END,SIGMAP,YP,TEMP,IERR)

******************************************************************************

C* PURPOSE
- COMPUTE SECOND DERIVATIVES NECESSARY TO BE
ABLE TO INTERPOLATE POINTS ON A SPLINE
FUNCTION UNDER TENSION PASSING THROUGH AT LEAST THREE NODES. SEE STIUNI (EI.X).

C* USE
- CALL CURVIN(N,X,Y,IENDSW,END,SIGMAP,YP,TEMP,IERR)

C* PARAMETERS
N - AN INPUT INTEGER SPECIFYING THE NUMBER OF NODES. N MUST BE AT LEAST 3.
X - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N CONTAINING THE X-COORDINATES OF THE NODES.
Y - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N CONTAINING THE Y-COORDINATES OF THE NODES.
IENDSW - AN INPUT ONE-DIMENSIONAL INTEGER ARRAY OF LENGTH 2 SPECIFYING THE OPTIONS CHOSEN AT THE TWO ENDPOINTS OF THE SPLINE CURVE. IENDSW(1) REFER TO (X(1),Y(1)) AND IENDSW(2) REFER TO (X(N),Y(N)).
YP - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N CONTAINING THE NORMALIZED X-COORDINATES OF THE NODES.
TEMP - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH 100 CONTAINING THE INTERPOLATION DATA.
IERR - A RETURN CODE, 0 IF ERROR FREE.

******************************************************************************
C* IENDSW(I) = 0 THE USER HAS INPUT VALUES FOR SUTS 167
C* Y' AT THE APPROPRIATE ENDPOINT SUTS 168
C* IN END(I) . SUTS 169
C* = 1 Y' WILL BE ESTIMATED SUTS 170
C* INTERNALLY AT THE APPROPRIATE ENDPOINT. SUTS 171
C* = 2 THE NATURAL CONDITION (Y' = 0) SUTS 173
C* WILL BE IMPOSED ON THE SUTS 174
C* APPROPRIATE ENDPOINT. SUTS 175
C* END - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH SUTS 176
C* 2. END(I) CONTAINS THE DERIVATIVE OF THE SUTS 177
C* SPLINE FUNCTION AT (X(I),Y(I)). END(2) SUTS 179
C* CONTAINS THE DERIVATIVE OF THE SPLINE SUTS 179
C* FUNCTION AT (X(N),Y(N)). IF IENDSW(I) IS SUTS 180
C* NON-ZERO, THEN END(I) NEED NOT BE DEFINED. SUTS 181
C* IF BOTH IENDSW(1) AND IENDSW(2) ARE NON- SUTS 182
C* ZERO, THEN END MAY BE A DUMMY PARAMETER. SUTS 183
C* SIGMAP - AN INPUT REAL NUMBER SPECIFYING THE SUTS 184
C* DENORMALIZED TENSION FACTOR. SUTS 185
C* YP - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF SUTS 186
C* LENGTH N CONTAINING SECOND DERIVATIVE SUTS 187
C* INFORMATION NECESSARY TO INTERPOLATE THE SUTS 188
C* SPLINE FUNCTION. SUTS 189
C* TEMP - A WORK ONE-DIMENSIONAL REAL ARRAY OF LENGTH SUTS 190
C* N - 1. SUTS 191
C* IERR - AN OUTPUT INTEGER SPECIFYING THE ERROR SUTS 192
C* RETURN CODE. SUTS 193
C* = 0 NORMAL RETURN. SUTS 194
C* = 2 TWO CONSECUTIVE X-VALUES ARE THE SAME. SUTS 195
C* PRECISION - SINGLE. SUTS 196
C* REQUIRED ROUTINES - CEE2, SNHC5, TERMS. SUTS 197
C* DATE RELEASED - MARCH 1, 1979. SUTS 198
C* LANGUAGE - FORTRAN. SUTS 199
C* SOURCE - A. K. CLINE AND R. J. RENKA SUTS 200
C* UNIVERSITY OF TEXAS AT AUSTIN SUTS 201
C* NON-NATURAL OPTIONS ONLY SUTS 202
C* LATEST REVISION - NONE. SUTS 203
C* FORMAL PARAMETERS
C*
INTEGER IENDSW(2), IERR, N

REAL END(2), SIGMAP, TEMP(1), X(N), Y(N), YP(N)

INTERNAL VARIABLES

INTEGER I, IBAK, NM1, NP1

REAL C1, C2, C3, DELXN, DELXNM, DELX1, DELX2, DIAG, DIAG1, DIAG2, DX1, DX2

REAL SDIAG1, SDIAG2, SLPN, SLP1, ZERO

DEFINE THE CONSTANT ZERO

DATA ZERO/O, OEO/

INITIALIZE VARIABLES

NM1 = N - 1
NP1 = N + 1
DELX1 = X(2) - X(1)
IERR = 2

APPROXIMATE THE LEFT END SLOPE IF NEEDED

IF (DELX1 .LE. ZERO) GO TO 9000
IF (IENDSW(1) .EQ. 0) SLP1 = END(1)
IF (IENDSW(1) .NE. 1) GO TO 10
DELX2 = X(3) - X(1)
IF (DELX2 .LE. DELX1) GO TO 9000
CALL CEEZ (DELX1, DELX2, SIGMAP, C1, C2, C3, N)
SLP1 = C1*Y(1) + C2*Y(2) + C3*Y(3)

APPROXIMATE THE RIGHT END SLOPE IF NEEDED

10 IF (IENDSW(2) .EQ. 0) SLPN = END(2)
IF (IENDSW(2) .NE. 1) GO TO 20
DELXN = X(N) - X(NM1)
DELXNM = X(N) - X(N-2)
IF (DELXN .LE. ZERO .OR. DELXNM .LE. DELXN) GO TO 9000
CALL CEEZ (-DELXN, -DELXNM, SIGMAP, C1, C2, C3, N)
SLPN = C1*Y(N) + C2*Y(NM1) + C3*Y(N-2)

SET UP RIGHT HAND SIDE AND TRIDIAGONAL SYSTEM FOR YP AND
PERFORM FORWARD ELIMINATION

20 DX1 = (Y(2) - Y(1)) / DELX1
CALL TERMS (DIAG1, SDIAG1, SIGMAP, DELX1)

TEST WHETHER THE LEFT ENDPOINT IS NATURAL
IF (IENDSW(1) .EQ. 2) GO TO 30

THIS CODE IS FOR A NON-NATURAL LEFT ENDPOINT

YP(1) = (DX1 - SLP1) / DIAG1
TEMP(1) = SDIAG1/DIAG1
GO TO 40

THIS CODE IS FOR A NATURAL LEFT ENDPOINT

30 TEMP(1) = ZERO
YP(1) = ZERO

BEGIN THE ELIMINATION LOOP

40 DO 50 I=2,NM1
IM1 = I - 1
DELX2 = X(I+1) - X(I)
IF (DELX2 .LE. ZERO) GO TO 9000
DX2 = (Y(I+1) - Y(I)) / DELX2

CALL TERMS (DIAG2,SDIAG2,SIGMA2,DELX2)
DIAG = DIAG1 + DIAG2 - SDIAG1*TEMP(IM1)
YP(I) = (DX2 - DX1 - SDIAG1*YP(IM1)) / DIAG
TEMP(I) = SDIAG2/DIAG
DX1 = DX2
DIAG1 = DIAG2
SDIAG1 = SDIAG2
50 CONTINUE

TEST WHETHER THE RIGHT ENDPOINT IS NATURAL

IF (IENDSW(2) .EQ. 2) GO TO 60

THIS CODE IS FOR A NON-NATURAL RIGHT ENDPOINT

DIAG = DIAG1 - SDIAG1*TEMP(NM1)
YP(N) = (SLPN - DX1 - SDIAG1*YP(NM1)) / DIAG
GO TO 70

THIS CODE IS FOR A NATURAL RIGHT ENDPOINT

60 YP(N) = ZERO

PERFORM BACK SUBSTITUTION

70 DO 80 I = 2,N
IBAK = NP1 - I
YP(IBAK) = YP(IBAK) - TEMP(IBAK)*YP(IBAK+1)
**BO CONTINUE**

**IERR = 0**

**CONTINUE**

**9000 RETURN**

**END**

**SUBROUTINE CURV12,(X,Y, IENDSW, END, SIGMA, YP)**

**DESCRIPTION**

- **PURPOSE**
  - COMPUTE SECOND DERIVATIVES NECESSARY TO BE ABLE TO INTERPOLATE POINTS ON A SPLINE FUNCTION UNDER TENSION PASSING THROUGH EXACTLY TWO NODES. SEE STIUNI (E1,X).

- **USE**
  - CALL CURV12,(X,Y, IENDSW, END, SIGMA, YP)

- **PARAMETERS**
  - **DX** - AN INPUT REAL NUMBER = X(2) - X(1) WHERE THE X’S ARE THE X-COORDINATES OF THE NODES.
  - **Y** - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTHX.
  - **IENDSW** - AN INPUT ONE-DIMENSIONAL INTEGER ARRAY OF LENGTH 2 CONTAINING THE Y-COORDINATES OF THE NODES.
  - **END** - AN INPUT ONE-DIMENSIONAL INTEGER ARRAY OF LENGTH 2 SPECIFYING THE OPTIONS CHOSEN AT THE TWO ENDPOINTS OF THE SPLINE CURVE.
  - **SIGMA** - AN INPUT REAL NUMBER SPECIFYING THE DEOMALIZED TENSION FACTOR.
  - **YP** - AN OUTPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH 2 CONTAINING SECOND DERIVATIVE INFORMATION NECESSARY TO INTERPOLATE THE SPLINE FUNCTION.

- **PRECISION**
  - SINGLE

- **REQUIRED ROUTINES**
  - SNI5C5M, TERMS.
C* DATE RELEASED - MARCH 1, 1979.
C* LANGUAGE - FORTRAN.
C* SOURCE - A. K. CLINE AND R. J. RENKA
UNIVERSITY OF TEXAS AT AUSTIN
NON-NATURAL OPTIONS ONLY
C* LATEST REVISION - NONE.
C* *
C* FORMAL PARAMETERS
C* INTEGER IENDSW(2)
C* REAL DX,END(2),SIGMAP,Y(2),YP(2)
C* INTERNAL VARIABLES
C* REAL DIAG,DIAG1,DIAG2,DX1,ONE,SDIAG1,SLP1,SLPN,TEMP,ZERO
C* INITIALIZE CONSTANTS
C* DATA ZERO,ONE/0.0E0,1.0E0/
C* INITIALIZE YP
C* YP(1) = ZERO
Yp(2) = ZERO
C* CHECK TO SEE IF EACH ENDPOINT IS EITHER NATURAL OR
HAS A COMPUTER-CONSTRUCTED DERIVATIVE
C* IF (IENDSW(1)*IENDSW(2) .NE. 0) GO TO 9000
C* AT LEAST ONE ENDPOINT HAS A USER-DEFINED SLOPE
C* COMPUTE SOME NECESSARY CONSTANTS
C* DX1 = (Y(2) - Y(1))/DX
CALL TEPK(SDIAG1,SDIAG2,SIGMAP,DX)
DIAG1 = ONE/DIAG1
C* CHECK TO SEE IF THE LEFT ENDPOINT IS NATURAL
C* IF (IENDSW(1) .NE. 2) GO TO 10
YP(2) = (END(2) - DX1)*DIAG1
GO TO 9000
C
C CHECK TO SEE IF THE RIGHT ENDPOINT IS NATURAL
C
10 IF (IENDSW(2) .NE. 2) GO TO 20
YP(1) = (DX1 - END(1))*DIAG
GO TO 9000
C
C NOW AT LEAST ONE OF THE ENDPOINTS HAS A USER-SPECIFIED
C DERIVATIVE AND NEITHER IS NATURAL - THIS IS THE CODE
C FROM THE ORIGINAL CURVI WITH N = 2
C
20 SLP1 = END(1)
IF (IENDSW(1) .EQ. 1) SLP1 = DX1
SLPN = END(2)
IF (IENDSW(2) .EQ. 1) SLPN = DX1
TEMP = SDIAG1*DIAG
DIAG = SDIAG1 - SDIAG1*TEMP
YP(2) = (SLPN - DX1 - (DX1 - SLP1)*TEMP) / DIAG
YP(1) = (DX1 - SLP1)*DIAG - TEMP*YP(2)
C
9000 RETURN
END
FUNCTION CURVI(XL,XU,N,X,Y,YP,SIGMAP)
C
******************************************************************************************
C PURPOSE
- THE VALUE OF CURVI IS THE DEFINITE INTEGRAL OF A SPLINE FUNCTION UNDER TENSION FROM XL
OF A SPLINE FUNCTION UNDER TENSION FROM XL TO XU GIVEN NECESSARY INFORMATION FROM
CURVIN OR CURVII - SEE SUTS (D1.X).

C USE
- CALL CURVI(XL,XU,N,X,Y,YP,SIGMAP)

C PARAMETERS
XL - AN INPUT REAL NUMBER SPECIFYING THE LOWER LIMIT OF INTEGRATION.
XU - AN INPUT REAL NUMBER SPECIFYING THE UPPER LIMIT OF INTEGRATION.
N - AN INPUT INTEGER SPECIFYING THE NUMBER OF NODES. N MUST BE AT LEAST 2.
X - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N SPECIFYING THE X-COORDINATES OF THE NODES.
Y - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N SPECIFYING THE Y-COORDINATES OF THE NODES.
YP - AN INPUT ONE-DIMENSIONAL REAL ARRAY OF LENGTH N WHICH IS OUTPUT FROM CURVIN OR CURVII.
SIGMAP - AN INPUT REAL NUMBER SPECIFYING THE

C SUTS 412
C SUTS 413
C SUTS 414
C SUTS 415
C SUTS 416
C SUTS 417
C SUTS 418
C SUTS 419
C SUTS 420
C SUTS 421
C SUTS 422
C SUTS 423
C SUTS 424
C SUTS 425
C SUTS 426
C SUTS 427
C SUTS 428
C SUTS 429
C SUTS 430
C SUTS 431
C SUTS 432
C SUTS 433
C SUTS 434
C SUTS 435
C SUTS 436
C SUTS 437
C SUTS 438
C SUTS 439
C SUTS 440
C SUTS 441
C SUTS 442
C SUTS 443
C SUTS 444
C SUTS 445
C SUTS 446
C SUTS 447
C SUTS 448
C SUTS 449
C SUTS 450
C SUTS 451
C SUTS 452
C SUTS 453
C SUTS 454
C SUTS 455
C SUTS 456
C SUTS 457
C SUTS 458
C SUTS 459
C SUTS 460
DENORMALIZED TENSION FACTOR.

PRECISION - SINGLE.

REQUIRED ROUTINES - INTRVL, SNHC3H.

DATE RELEASED - MARCH 1, 1979.

LANGUAGE - FORTRAN.

SOURCE - A. K. CLINE AND R. J. RENKA
UNIVERSITY OF TEXAS AT AUSTIN

LATEST REVISION - NONE.

------------------------------------------------------------------

FORMAL PARAMETERS

INTEGER N

REAL SIGMAP, X(N), XL, XU, Y(N), YP(N)

INTERNAL VARIABLES

INTEGER IL, ILM1, ILP1, IM1, IU, IUM1

REAL CL1, CL2, CS, CU1, CU2, C1, C2, DEL1, DEL2, DELL1, DELL2, DELS, DELSI, DELU1

REAL DELU2, DEL1, DEL2, DUMMY, HALF, ONE, SIGCUBE, SIGDELS, SIXTH, SS

REAL SSIGN, SUM, THIRD, TWEIGHT, T1, T2, XXL, XXU, ZERO

INITIALIZE CONSTANTS

DATA ZERO, ONE, HALF, 0.000, 1.000, 5.000, 10.000

DATA THIRD, SIXTH, TWEIGHT/

$171652525252525255525538, 1715525252525252525252538, 171452525252525252538$

THIRD = 1/3.

SIXTH = 1/6.

TWEIGHT = 1/12.

CONSTRUCT THE STATEMENT FUNCTION TERM
TERM COMPUTES COEFFICIENTS FOR ANTI-DERIVATIVES

TERM(CMM1, CMM2, T) = (CMM1 - CMM2 - SIGMAP*T*SS) /

$$(SIGCUBE*(SS + SIGDELS))$$

SIGCUBE = SIGMAP*SIGMAP

DETERMINE ACTUAL UPPER AND LOWER LIMITS.
C
XXL = XL
XXU = XU
SSIGN = ONE
IF (XXL .LT. XXU) GO TO 10
XXL = XXU
XXU = XL
SSIGN = -ONE
IF (XXU .GT. XXL) GO TO 10

RETURN ZERO IF XL = XU

CURVI = ZERO
GO TO 9000

SEARCH FOR THE PROPER SUBINTERVALS
X(ILM1) .LE. XXL .LE. X(IL)

10 ILM1 = INTRVL (XXL,X,N)
IL = ILM1 + 1

X(IUM1) .LE. XXU .LE. X(IU)

IUM1 = INTRVL (XXU,X,N)
IU = IUM1 + 1

BRANCH IF XXU AND XXL ARE IN THE SAME SUBINTERVAL

IF (IL .EQ. IU) GO TO 70
SUM = ZERO

BRANCH IF XXL = X(IL)

IF (XXL .EQ. X(IL)) GO TO 20

INTEGRATE FROM XXL TO X(IL)
FOR THIS CODE XXL .LT. XX(IL) .LE. XX(IUM1)

DELI = XXL - X(ILM1)
DEL2 = X(IL) - XXL
DELS = DEL1 + DEL2
DELS1 = DEL1 + DEL2

T1 = (DELS1 .LE. DELS)*DELS1
T2 = DEL2*DELS1
SUM = T1*X(IL) + T2*X(ILM1)

IF (SIGMAP .EQ. ZERO) SUM = SUM - T1*T1*DELS*YP(ILM1)*SIXTH
$ = T2*(DEL1*(DEL2 + DELS) + DELS*DELS)*YP(ILM1)*TWELFTH

IF (SIGMAP .EQ. ZERO) GO TO 20
THIS CODE IS FOR A SPLINE UNDER TENSION

SIGDELS = SIGMAP*DELS
CALL SNMCSH(DUMMY,C1,SIGMAP,DELS1,2)
CALL SNMCSH(DUMMY,C2,SIGMAP,DELS2,2)
CALL SNMCSH(SS,C5,SIGDELS,3)
SUM = SUM + TERM(CS,C1,T1)*YP(IL1) + TERM(C2,ZERO,T2)*YP(ILM1)

BRANCH IF X(IL1) < X(IUM1)

20 IF (IL .EQ. IUM1) GO TO 60

INTEGRATE OVER THE INTERIOR INTERVALS WITH A LOOP
FOR THIS CODE X(IL1) < X(IUM1)

ILP1 = IL + 1
IF (SIGMAP .NE. ZERO) GO TO 40

THIS CODE IS FOR A CUBIC SPLINE

DO 30 I=ILP1,IUM1
IM1 = I - 1
DELS = (X(I) - X(IM1))*HALF
SUM = SUM + (Y(I) + Y(IM1))*DELS - (YP(I) + YP(IM1))*
DELS*DELS*DELS*THIRD
30 CONTINUE
GO TO 60

THIS CODE IS FOR A SPLINE UNDER TENSION

40 DO 50 I=ILP1,IUM1
IM1 = I - 1
DELS = (X(I) - X(IM1)
SIGDELS = SIGMAP*DELS
CALL SNMCSH(SS,C5,SIGDELS,3)
SUM = SUM + (Y(I) + Y(IM1))*DELS*HALF + (YP(I) + YP(IM1))*
(Cs - SS*SIGDELS*HALF)/(SIGCUBE*SS + SIGDELS)
50 CONTINUE

BRANCH IF X(IU-1) = XXU

60 IF (XXU .EQ. X(IUM1)) GO TO 80

INTEGRATE FROM X(IU-1) TO XXU
FOR THIS CODE X(IU-1) < XXU

DELS = XXU - X(IUM1)
DEL2 = X(IU) - XXU
DELS = X(IU) - X(IUM1)
DELS1 = HALF/DELS
T1 = DEL1*DELS1
T2 = (DEL2 + DELS)*DELS1
SUM = SUM + T1*Y(IU1) + T2*Y(IUM1)
IF (SIGMAP = .EQ. ZERO) SUM = SUM - T1*(DEL2*(DEL1 + DELS) + DELS*DEL2)*YP(IU1) + TWELFTH - T2*DEL2*YP(IUM1)*SIXTH
IF (SIGMAP = .EQ. ZERO) GO TO 80
SIGDELS = SIGMAP*DELS
CALL SNHCSH (DUMMY,C1,SIGMAP*DELS)
CALL SNHCSH (DUMMY,C2,SIGMAP*DELS)
CALL SNHCSH (S5,C5,SIGDELS, S3)
SUM = SUM + TERM(C1, ZERO, T1)*YP(IU1) + TERM(C5,C2,T2)*YP(IUM1)
GO TO 80

C
INTEGRATE FROM XXL TO XXU
C
FOR THIS CODE XXL .LE. XXU .LT. XXU .LE. X(IU) SUTS 623
C
70 DELU1 = XXU - X(IUM1)
DELU2 = X(IU) - XXU
DELL1 = XXL - X(IUM1)
DELL2 = X(IU) - XXL
DELS = X(IU) - X(IUM1)
DELI = XXL - XXU
DELSI = HALF/DELS
T1 = (DELU1 + DELL1)*DELI*DELSI
T2 = (DELU2 + DELL2)*DELI*DELSI
SUM = T1*Y(IU1) + T2*Y(IUM1)
IF (SIGMAP = .EQ. ZERO) SUM = SUM - T1*(DELU2*(DELS + DELU1) + DELL2)*YP(IU1) + TWELFTH - T2*(DELL1*(DELS + DELL2))
S + DELU1*(DELS + DELU2)*YP(IUM1)*TWELFTH
IF (SIGMAP = .EQ. ZERO) GO TO 80
C
THIS CODE IS FOR A SPLINE UNDER TENSION
C
SIGDELS = SIGMAP*DELS
CALL SNHCSH (DUMMY,C1,SIGMAP*DELU1,2)
CALL SNHCSH (DUMMY,C2,SIGMAP*DELU2,2)
CALL SNHCSH (DUMMY,C1,SIGMAP*DELL1,2)
CALL SNHCSH (DUMMY,C2,SIGMAP*DELL2,2)
CALL SNHCSH (S5,DUMMY, SIGDELS, S3)
SUM = SUM + TERM(C1,C1,T1)*YP(IU1) + TERM(C2,C1,T2)*YP(IUM1)
C
CORRECT SIGN AND RETURN
C
80 CURVI = SSIGN*SUM
C
9000 RETURN
END
SUBROUTINE CEEZ (DEL1,DEL2,SIGMA,C1,C2,C3,N)
C**************************************************************************************************
C  PURPOSE
- DETERMINE COEFFICIENTS C1, C2, AND C3 USED TO DETERMINE ENDPOINT SLOPES. GIVEN THE
  POINTS (X1,Y1), (X2,Y2), AND (X3,Y3), THE ENDPOINT SLOPE IS C1*Y1 + C2*Y2 + C3*Y3
C  IF N > 2 AND C1*Y1 + C2*Y2 IF N = 2.
C
C  USE
- CALL CEEZ(DEL1,DEL2,SIGMA,C1,C2,C3,N)

C  PARAMETERS
DELI - AN INPUT REAL NUMBER = X2 - X1. DEL1 > 0.
DEL2 - AN INPUT REAL NUMBER = X3 - X1. IF N = 2, THEN DEL2 MAY BE A DUMMY PARAMETER.
        DEL2 > 0.
SIGMA - AN INPUT REAL NUMBER SPECIFYING THE TENSION FACTOR.
C1 - AN OUTPUT REAL NUMBER.
C2 - AN OUTPUT REAL NUMBER.
C3 - AN OUTPUT REAL NUMBER. IF N = 2, THEN C3 IS A DUMMY PARAMETER.
N - AN INPUT INTEGER SPECIFYING THE NUMBER OF NODES.

C  PRECISION
- SINGLE.

C  REQUIRED Routines
- NONE.

C  DATE RELEASED
- MARCH 1, 1979.

C  LANGUAGE
- FORTRAN.

C  SOURCE
- A. K. CLINE AND R. J. RENKA
  UNIVERSITY OF TEXAS AT AUSTIN

C  LATEST REVISION
- NONE.

C**************************************************************************************************
REAL COSH1, DEL, DENOM, ONE, SIN, TWO, ZERO

C INITIALIZE CONSTANTS
C
DATA ZEPO, ONE, TWO/O, OEO, 1.0EO, 2.0EO/C
C
TEST WHETHER N = 2
C IF (N .EQ. 2) GO TO 20
C
DEL = DEL2 - DEL1
C
TEST WHETHER SIGMA = ZERO
C IF (SIGMA .NE. ZERO) GO TO 10
C
THIS CODE IS FOR A STANDARD CUBIC SPLINE
C
C1 = -(DEL1*DEL2)/(DEL1*DEL2)
C2 = DEL2/(DEL1*DEL)
C3 = -(DEL1/(DEL2*DEL)
GO TO 9000
C
THIS CODE IS FOR A SPLINE UNDER TENSION
C
10 CALL SNHCSH(DUMMY, COSH1, SIGMA*DEL1, 1)
CALL SNHCSH(DUMMY, COSH2, SIGMA*DEL2, 1)
SIN = TWO * SINH(SIGMA*(DEL2*DEL1)/TWO) * SINH(SIGMA*DEL/TWO)
DENOM = ONE / (COSH1*DEL - DEL1*SIN)
C1 = SIN * DENOM
C2 = -COSH2 * DENOM
C3 = COSH1 * DENOM
GO TO 9000
C
TWO COEFFICIENTS
C
20 C1 = -DEN/DDEL1
C2 = -C1
9000 RETURN
C
END

SUBROUTINE TERMS(DIAG, SDIAG, SIGMA, DEL)
C
C***********************************************
C
C* PURPOSE
C* - COMPUTE THE DIAGONAL AND SUPERDIAGONAL TERMS
C* OF THE TRIDIAGONAL LINEAR SYSTEM ASSOCIATED
C*
WITH SPLINE UNDER TENSION INTERPOLATION.

* SUTS 755

* USE
  - CALL TERMS(DIAG, SDIAG, SIGMA, DEL)

* PARAMETERS
  - DIAG - AN OUTPUT REAL NUMBER CONTAINING THE
            DIAGONAL TERM.
  - SDIAG - AN OUTPUT REAL NUMBER CONTAINING THE
            SUPERDIAGONAL TERM.
  - SIGMA - AN INPUT REAL NUMBER SPECIFYING THE TENSION FACTOR.
  - DEL - AN INPUT REAL NUMBER SPECIFYING THE STEP SIZE.

* PRECISION
  - SINGLE.

* REQUIRED ROUTINES
  - NONE.

* DATE RELEASED
  - MARCH 1, 1979.

* SOURCE
  - A. K. CLINE AND R. J. RENKA
  - UNIVERSITY OF TEXAS AT AUSTIN

* LATEST REVISION
  - NONE.

real del, diag, sdiag, sigma

internal variables

real coshm, denom, sigdel, sinh, sixth, third

initialize constants

data zero, sixth, third/
  0.000, 17.152525252525252525252538,
  17.152525252525252525252538/

  sixth = 1. / 6.
  twelfth = 1. / 12.

  test whether sigma is zero

  if (sigma .ne. zero) go to 10

  this code is for the standard cubic spline
DIAG = DEL * THIRD
SDIA = DEL * SIXTH
GO TO 9000
C
C
THIS CODE IS FOR A SPLINE UNDER NON-ZERO TENSION
C
10 SIGDEL = SIGMA * DEL
CALL SNHCISH(SINHM, COSHM, SIGDEL, O)
DENOM = DEL / (SINHM + SIGDEL) * SIGDEL * SIGDEL
DIAG = DENOM * (SIGDEL + COSHM - SINHM)
SDIA = SIGDEL * SINHM
C
9000 RETURN
END
SUBROUTINE SNHCISH(SINHM, COSHM, X, ISW)

C***************************************************************SUTS 820
C*   PURPOSE
C*   - APPROXIMATE SINHM(X) = SIN(X) - X
C*   C*   COSHM(X) = COS(X) - 1
C*   C*   AND COSHM(X) = COS(X) - 1 - X*X/2
C*   C*   WITH RELATIVE ERROR < 3.42E-14
C*   C*   USE
C*   - CALL SNHCISH(SINHM, COSHM, X, ISW)
C*   C*   PARAMETERS
C*   SINHM - AN OUTPUT REAL NUMBER CONTAINING SINHM(X)
C*   C*   IF ISW = -1, 0, OR 3, OTHERWISE SINHM IS UNCHANGED UPON RETURN.
C*   C*   COSHM - AN OUTPUT REAL NUMBER CONTAINING COSHM(X)
C*   C*   IF ISW = 0 OR 1 AND CONTAINING COSHM(X)
C*   C*   ISW = 2 OR 3, OTHERWISE COSHM IS UNCHANGED UPON RETURN.
C*   C*   X - AN INPUT REAL NUMBER CONTAINING THE INDEPENDENT VARIABLE.
C*   C*   ISW - AN INPUT INTEGER SPECIFYING THE FUNCTION DESIRED.
C*   C*   -1 (ONLY) SINHM IS DESIRED.
C*   C*   0 SINHM AND COSHM ARE DESIRED.
C*   C*   1 (ONLY) COSHM IS DESIRED.
C*   C*   2 (ONLY) COSHM IS DESIRED.
C*   C*   3 SINHM AND COSHM ARE DESIRED.
C*   C*   PRECISION
C*   - SINGLE.
C*   C*   REQUIRED ROUTINES
C*   - NONE.
C*   C*   DATE RELEASED
C*   - MARCH 1, 1979.
C*   SUTS 804
SUTS 805
SUTS 806
SUTS 807
SUTS 808
SUTS 809
SUTS 810
SUTS 811
SUTS 812
SUTS 813
SUTS 814
SUTS 815
SUTS 816
SUTS 817
SUTS 818
SUTS 819
SUTS 820
SUTS 821
SUTS 822
SUTS 823
SUTS 824
SUTS 825
SUTS 826
SUTS 827
SUTS 828
SUTS 829
SUTS 830
SUTS 831
SUTS 832
SUTS 833
SUTS 834
SUTS 835
SUTS 836
SUTS 837
SUTS 838
SUTS 839
SUTS 840
SUTS 841
SUTS 842
SUTS 843
SUTS 844
SUTS 845
SUTS 846
SUTS 847
SUTS 848
SUTS 849
SUTS 850
SUTS 851
SUTS 852
**LANGUAGE**  
- FORTRAN

**SOURCE**  
- A. K. CLINE AND R. J. RENKA  
- UNIVERSITY OF TEXAS AT AUSTIN

**LATEST REVISION**  
- NONE

**FORMAL PARAMETERS**

**INTEGER ISW**

**REAL COSH,M,SINH,X**

**INTERNAL PARAMETERS**

**REAL AX,CP1,CP2,CP3,CP4,CQ1,EXPX,SP1,SP2,SP3,SP4,SQ1,XX,ZP1,ZP2**

**REAL IP3,ZQ1,ZQ2,ZQ3,ZQ4**

**DATA SP4/4.50217693813133E-08/  
SP3/8.9528734163904E-06/  
SP2/8.72063789791702E-04/  
SP1/4.363115936891690E-02/  
SQ1/-6.38954430175110E-03/  
C4/1.7941956490190E-07/  
CP3/2.8727722979044E-05/  
CP2/2.1515159002028E-03/  
CP1/7.98181822756256E-02/  
CQ1/-7.5151510567447E-03/  
DATA ZP4/5.5929711526472E-07/  
ZP3/5.97943494030694E-04/  
ZP1/1.6900044194792E-02/  
ZQ4/1.3341253542375E-09/  
ZQ3/-5.00058944138663F-07/  
ZQ2/-1.2781496403863E-04/  
ZQ1/-1.65332871439181E-02/  
XX = X  
AX = ABS(XX)  
XS = XX*XX

**IF**  
((AX .GE. 2.70) .OR. (AX .GE. 1.15) .AND.  
ISW .NE. 2))  
EXPX = EXP(AX)

**APPROXIMATE SINHM**

**IF**  
(ISW .EQ. 1 .OR. ISW .EQ. 2)  
GO TO 2

**IF**  
(AX .GE. 1.15)  
GO TO 1
\[
\sinh = \left( (\sinh + \sinh) \right) \times \sinh
\]

GO TO 2
1 \ sinh = -\left( (1.0 + \exp(x)) / 2.0 \right).
2 IF (x < 0.0) \ sinh = -\sinh
3 \approx \cosh

GO TO 4
4 IF (x > 2.0) \ RETURN
5 \cosh = \left( (\cosh + \cosh) \right) \times \cosh
6 \approx \cosh
7 \cosh = \left( (1.0 + \exp(x)) / 2.0 \right).
8 \approx \cosh

RETURN
9 \cosh = \left( (1.0 + \exp(x)) / 2.0 \right).
10 \RETURN
11 FUNCTION \text{intrvl}(t, x, n)

Purpose:
- Determine the index of the interval
  (determined by a given increasing sequence)
  in which a given value lies.

Use:
- \( t = \text{intrvl}(t, x, n) \)

Parameters:
- \( t \): an input real number specifying the given value.
- \( x \): an input one-dimensional real array of length \( n \) specifying the increasing sequence. \( x \) must be strictly increasing.
- \( n \): an input integer specifying the length of \( x \). \( n > 1 \).

Output:
- \( i \): if \( t \leq x(1) \), \( i = 1 \).
- if \( t > x(n-1) \), \( i = n - 1 \).
- otherwise, \( x(i) \leq t \leq x(i+1) \).

Precision:
- single.
REQUIRED ROUTINES - NONE.
DATE RELEASED - MARCH 1, 1979.
SOURCE - A. K. CLINE AND R. J. RENKA
         UNIVERSITY OF TEXAS AT AUSTIN
LATEST REVISION - NONE.

FORMAL PARAMETERS
INTEGER N
REAL T, X(N)

INTERNAL VARIABLES
INTEGER IM, IL
REAL TT

IF (TT .LE. X(2)) GO TO 4
IF (TT .GE. X(N-1)) GO TO 5
IL = 2
IM = N-1

INTERPOLATE LINEARLY FOR I
1 I = IL+IFIX(FLOAT(IM-IL)*(TT-X(IL))/(X(IM)-X(IL)))
IF (TT .LT. X(I)) GO TO 2
IF (TT .LE. X(I+1)) GO TO 3

I IS TOO SMALL - ADJUST AND TRY AGAIN
IL = I+1
GO TO 1

I IS TOO LARGE - ADJUST AND TRY AGAIN
IM = I
GO TO 1

INTRVL = I
I IS JUST RIGHT - RETURN

RETURN

LEFT END

4 INTRVL = 1
RETURN

RIGHT END

5 INTRVL = N-1
RETURN

END
REFERENCES


**SEGMENTATION DIRECTIVES FOR THE**
**VORTEX LATTICE FORTRAN PROGRAM**

**H. E. HERBERT**
**COMPUTER SCIENCES CORPORATION**
**HAMPTON, VA.**
**JULY, 1981**

**ROOT TREE WINGAL- (GEOMETRY, MATXSOL, AERODYN, CDRAGNF, TIPSUCT, VORTEX, CNLONG ,)**

INCLUDE WINGAL, INFSUB, LOADING, FTLUP, READIN
GLOBAL ALL, TOTHREE, THREFOR, ONETHRE, MAINONE, CCRRDD, INSUB23

**GEOMETRY** INCLUDE GEOMETRY, PLANPLT
**MATXSOL** INCLUDE MATXSOL, GIVENS, BLOCKR, TRIANG, SOLVER, BUFFIN
**AERODYN** INCLUDE AERODYN, FLOWFL, CDICLS, HEAPSRT, SIFT
**CDRAGNF** INCLUDE CDRAGNF
**TIPSUCT** INCLUDE TIPSUCT, WRTANS
**VORTEX** INCLUDE VORTEX
**CNLONG** INCLUDE CNLONG, INTERP, IQHSCV, IQHSD, IQHSE, IQHSF, IQHSG
**CNLONG** INCLUDE IQHSH, UERTST, UGETIO, SUTS, CURVIN, CURVI2, CURVI
**CNLONG** INCLUDE CEEZ, TERMS, SNHCSH, INTRVL

END

**FIGURE 1 - SEGMENTATION DIRECTIVES**
This document presents the source code for the latest production version, MARK IV, of the NASA-Langley Vortex Lattice Computer Program. All viable subcritical aerodynamic features of previous versions have been retained. This version extends the previously documented program capabilities to four planforms, 400 panels, and enables the user to obtain vortex-flow aerodynamics on cambered planforms, flow-field properties off the configuration in attached flow, and planform longitudinal load distributions.