1983

A three-dimensional transonic, potential flow computer program : its conversion to IBM fortran and utilization

Paschall, Jack.

https://hdl.handle.net/10945/20012
THESIS

A THREE-DIMENSIONAL TRANSONIC, POTENTIAL FLOW
COMPUTER PROGRAM, ITS CONVERSION TO IBM
FORTRAN AND UTILIZATION

by

Jack Paschall III

December 1983

Thesis Advisor: R. D. Zucker

Approved for Public Release, Distribution Unlimited
This thesis describes the conversion of a computer program from Fortran IV for the NOS 1.2 operating system of the CYBER 175 or CDC 6600 computer to Fortran IV compatible with the Naval Postgraduate School IBM 3033 system. The converted program, called FLO27, calculates the inviscid, three-dimensional transonic potential flow over wings or wing-body combinations. The data input to FLO27 is
extensive; therefore, an interactive program was developed to aid the user in building the required input data file.
A Three-Dimensional Transonic, Potential Flow Computer Program, Its Conversion to IBM Fortran and Utilization

by

Jack Easchall III
Commander, United States Navy
B.S., Oregon State University, 1965

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983
AESTRACT

This thesis describes the conversion of a computer program from Fortran IV for the NOS 1.2 operating system of the CYBER 175 or CDC 6600 computer to Fortran IV compatible with the Naval Postgraduate School IBM 3033 system. The converted program, called FLO27, calculates the inviscid, three-dimensional transonic potential flow over wings or wing-body combinations. The data input to FLO27 is extensive; therefore, an interactive program was developed to aid the user in building the required input data file.
TABLE OF CONTENTS

I. INTRODUCTION ................................................. 9
   A. BACKGROUND ............................................. 9
   B. VISCOUS/INVISCID SYSTEM OF PROGRAMS .................. 10

II. POTENTIAL FLOW PROGRAM FLO27 .............................. 14
   A. RE-FRAGMENTING ........................................ 14
   B. PROGRAM DESCRIPTION ................................... 19
      1. Program Input ...................................... 22
      2. Program Output ..................................... 32

III. INTERACTIVE INPUT PROGRAM FLO27IN .......................... 35

IV. FLO27 BATCH SYSTEM EXECUTION ............................... 38

V. PROGRAM TEST RESULTS ....................................... 40
   A. ACCEPTANCE TEST DATA .................................. 40
   B. COMPARISON WITH OTHER PROGRAMS ....................... 41
   C. AE-4501 CLASS PROJECT .................................. 41

APPENDIX A ...................................................... 44

APPENDIX B ...................................................... 46

APPENDIX C ...................................................... 47

APPENDIX D ...................................................... 48

APPENDIX E ...................................................... 70

APPENDIX F ...................................................... 132

LIST OF REFERENCES ............................................ 151

INITIAL DISTRIBUTION LIST .................................... 152
LIST OF TABLES

I. Viscous/Inviscid Wing System of Programs  
   11

II. CDC Magnetic Tape Files  
    15

III. FLO27 Re-Programing Changes  
    16

IV. Initialized Input Variables  
    18
**LIST OF FIGURES**

1.1. Viscous/Inviscid Interaction Procedure .... 12
2.1. Dihedral Angle ..................... 27
2.2. Wing Defining Geometry ................ 28
2.3. Section Defining Geometry ............ 31
2.4. Versatec Plot of Cp vs. X/C .......... 34
5.1. Program Calculated and Wind Tunnel Data ... 42
ACKNOWLEDGEMENT

I would like to acknowledge the most valuable assistance of the W. R. Church computer center personnel in the successful completion of this thesis. In particular, the guidance of Mr. Dennis Mar was instrumental in directing my initial efforts. I am grateful to Dr. Robert Zucker for providing the constant attention and encouragement which kept me working at an even pace. In addition, I am deeply appreciative of the love and support provided by my wife Tricia and our two children. Most of all I thank God for the strength, determination and peace which has sustained me and brought this project to a successful completion.
I. INTRODUCTION

In the Aeronautical Engineering curriculum graduate level aerodynamics course, AE-4501, the students are exposed to two computer programs. One of these, prepared by the Douglas Aircraft Company, analyzes the potential flow around three-dimensional wings but is limited to incompressible flow [Ref. 1]. The other program, prepared by Cebeci, calculates the friction drag for two-dimensional incompressible flow over airfoils [Ref. 2]. A serious defect of these programs is that they are not state-of-the-art computer programs. The Douglas program does not consider the effects of compressibility and the boundary layer program, in addition to being restricted to incompressible flow, does not predict the laminar to turbulent transition point.

A. BACKGROUND

In 1980 the Department of Aeronautics at the Naval Postgraduate School acquired an intricate computer program recently developed by the Boeing Commercial Airplane Company. This state-of-the-art program calculates three-dimensional transonic flow over wings and bodies in
both the outer-inviscid flow region governed by the transonic potential equation and the thin layer in which the first order, compressible boundary layer equations are assumed to be valid.

The Boeing program as received was designed to be executed on a CDC 6600 or a CYBER 175 computer and was written using CDC FORTRAN IV extended language. This thesis therefore was primarily concerned with the conversion of the program to FORTRAN IV extended compatible with the Naval Postgraduate School's (NPS) IBM 3033 system. The large modular program was divided so that the potential flow analysis portion could be run separately. Simplified instructions for use of the program were also prepared.

3. VISCOUS/INVISCID SYSTEM OF PROGRAMS

The Viscous/Inviscid Wing System (VIWS) of programs calculates three-dimensional transonic flow over wings and wing body combinations including details of the laminar or turbulent flow in the three-dimensional viscous boundary layer. The flow field is calculated in two overlapping regions: an outer inviscid flow region governed by the transonic potential equation, and a thin boundary layer in which the first order, three-dimensional, compressible
boundary layer equations are assumed to hold and in which the effects of surface heat and mass transfer can be computed. A list of the VIWS of programs is presented in Table I.

**TABLE I**

**Viscous/Inviscid Wing System of Programs**

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI027</td>
<td>Jameson-Caughey inviscid, transonic wing code</td>
</tr>
<tr>
<td>A411IN</td>
<td>Reads geometry &amp; velocity data, constructs coordinate system</td>
</tr>
<tr>
<td>VWIN</td>
<td>Potential flow boundary layer interface</td>
</tr>
<tr>
<td>A411AC1</td>
<td>Three-dimensional boundary layer program</td>
</tr>
<tr>
<td>INTERP</td>
<td>Boundary layer potential flow interface</td>
</tr>
<tr>
<td>A411F1, A411F2, A411FS</td>
<td>Graphics display programs</td>
</tr>
</tbody>
</table>

The basic sequence of calculations used by the VIWS to obtain matched viscous and inviscid solutions consists of an iterative loop in which the inviscid outer flow analysis and the boundary layer analysis are performed sequentially. The iterative sequence is continued until either convergence (satisfactory matching) is achieved, or the maximum number of iterations specified by the user has been performed. The VIWS programing sequence is shown schematically in Fig. 1.1.
Figure 1.1. Viscous/Inviscid Interaction Procedure

The potential flow is calculated for the bare wing during the first iteration. In subsequent iterations, the effect of the boundary layer flow on the outer inviscid flow is
felt as a modification to the wing shape through the addition of the boundary layer displacement thickness. Convergence is recognized, and the iterations are stopped, when the maximum change between the new and old displacement thickness, expressed as a fraction of the maximum displacement thickness, is less than the convergence tolerance chosen by the user.

The VIWS utilizes the Jameson-Caughey transonic inviscid wing program FLO27, to carry out the potential flow analysis. The boundary layer analysis is performed by a finite difference boundary layer prediction program developed by the Boeing Commercial Airplane Company. The basic theory behind the boundary layer program is contained in [Ref. 3]. A detailed description of the VIWS of programs (excluding the potential flow program FLO27) is contained in [Ref. 4]. A basic guide to the use of the VIWS of programs is contained in [Ref. 5].
II. POTENTIAL FLOW PROGRAM FLO27

Because of the extensive length and number of program modules in the VIWS, the Potential Flow Program, FLO27, was singled out for conversion. It was anticipated that FLO27 would be run separately at first and recombined with the other program modules at some later date when these modules were themselves translated for execution on the IBM 3033 computer.

A. RE-PGRAMING

The Potential Flow Program, hereafter called FLO27, was received on magnetic tape and loaded into the IBM 3033 mass storage system using the Jcb Control Language (JCL) routines presented in Appendix A. The magnetic tape contained twenty (20) total files in which the format was 9 track, 1600 CPI, unlabeled. The card image format for the sixteen (16) program files is 80 characters per record and the four (4) output files contain 150 characters per record. The program and output files on the original CDC tape are listed in Table II.

The FLO27 program was converted to FORTRAN IV extended suitable for execution on the IBM computer using the NPS CDC
### TABLE II

CDC Magnetic Tape Files

<table>
<thead>
<tr>
<th>File/Records</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 /2356</td>
<td>FLO27</td>
<td>Potential Flow Program</td>
</tr>
<tr>
<td>2 /3194</td>
<td>A411IN</td>
<td>Reads geometry &amp; velocity data constructs coordinate system</td>
</tr>
<tr>
<td>3 / 378</td>
<td>VWIN</td>
<td>Potential flow boundary-layer interface</td>
</tr>
<tr>
<td>4 /6611</td>
<td>A411A01</td>
<td>Three-dimensional boundary layer program</td>
</tr>
<tr>
<td>5 /1977</td>
<td>INTERP</td>
<td>Boundary-layer potential flow interface</td>
</tr>
<tr>
<td>6 / 668</td>
<td>A411PS</td>
<td>Streamline plots</td>
</tr>
<tr>
<td>7 / 211</td>
<td>A411P1</td>
<td>One-dimensional plots</td>
</tr>
<tr>
<td>8 / 586</td>
<td>A411P2</td>
<td>Contour plots</td>
</tr>
<tr>
<td>9 / 70</td>
<td>COUPLE</td>
<td>Procedure files</td>
</tr>
<tr>
<td>10 / 158</td>
<td>ITER</td>
<td></td>
</tr>
<tr>
<td>11 / 7</td>
<td>DATAIN</td>
<td></td>
</tr>
<tr>
<td>12 / 78</td>
<td>FINAL</td>
<td></td>
</tr>
<tr>
<td>13 / 434</td>
<td>BOEB1</td>
<td>Boeing McLean computer program</td>
</tr>
<tr>
<td>14 / 36</td>
<td>CONTIPLT</td>
<td>Contour plots</td>
</tr>
<tr>
<td>15 / 17</td>
<td>CORDPLT</td>
<td>One-dimensional plots</td>
</tr>
<tr>
<td>16 / 40</td>
<td>STREPLT</td>
<td>Streamwise plots</td>
</tr>
<tr>
<td>17</td>
<td>OUTF27</td>
<td>Output from FLO27</td>
</tr>
<tr>
<td>18</td>
<td>OUTIPC</td>
<td>Output from VWIN</td>
</tr>
<tr>
<td>19</td>
<td>OUT411L</td>
<td>Output from boundary-layer, lower surface</td>
</tr>
<tr>
<td>20</td>
<td>OUT411U</td>
<td>Output from boundary-layer, upper surface</td>
</tr>
</tbody>
</table>

The first step taken consisted of program compilation using the WATFIV compiler with its extended error messages. The listing which was
produced flagged all areas of the program which required revision. Program changes were accomplished utilizing this WATFIV listing. Some of the more general and repetitive changes are listed in Table III.

**TABLE III**

**FLO27 Re-Programming Changes**

<table>
<thead>
<tr>
<th>CDC Code</th>
<th>IBM Code Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables: FREAD, FREAD,</td>
<td>Eliminated from program</td>
</tr>
<tr>
<td>PWRT, EWRIF, IREAD,</td>
<td></td>
</tr>
<tr>
<td>IREAD, IWRIT, IWRIF</td>
<td></td>
</tr>
<tr>
<td>WRITE(IWRIT, 600)</td>
<td>WRITE(6, 600)</td>
</tr>
<tr>
<td>READ(IREAF, 500)</td>
<td>READ(5, 500)</td>
</tr>
<tr>
<td>READ 7, WRITE 7 or REWIND 7</td>
<td>Changed to READ 14, WRITE 14 or REWIND 14</td>
</tr>
<tr>
<td>Call SECCNT(T)</td>
<td>Step eliminated</td>
</tr>
<tr>
<td>Call SSWITCH(1, ISTCP)</td>
<td>Call SLITET(1, ISTOP)</td>
</tr>
<tr>
<td>Delimiter of form *</td>
<td>Replaced by '</td>
</tr>
<tr>
<td>Comment cards with *</td>
<td>Replaced by C</td>
</tr>
<tr>
<td>LEVEL statement</td>
<td>Step eliminated</td>
</tr>
<tr>
<td>If(UNIT(N), GT.0.) GO TC</td>
<td>All of this type eliminated</td>
</tr>
</tbody>
</table>

The most difficult change to make occurred with the CDC Buffer IN or Buffer OUT statements which were used in the program to transfer portions of a three-dimensional array into and out of main memory. The Buffer routines reduce the memory size required to execute the program. This statement type occurred in the main program and several of the subroutines.
The change required to translate this statement is presented below with the CDC code preceding the IBM FORTRAN.

BUFFER OUT (N3,1) (G(1,1,1),G(MX,MY,1)) changed to
WRITE(N3) ((G(I,J,1),I=1,MX),J=1,MY) and
BUFFER IN (N1,1) (G(1,1,M),G(MX,MY,M)) changed to
READ(N1,ERR= ) ((G(I,J,M),I=1,MX),J=1,MY)

The variable ERR was assigned the GO TO statement number of the UNIT statement immediately following the BUFFER IN line of code. As an example, if the UNIT statement following the BUFFER IN code was - If(UNIT(N1).GT.0.) GO TO 151, then the number 151 was assigned to variable ERR following the equal sign. All CDC UNIT statements were eliminated from the FLO27 source code per Table II.

In addition to the program changes required to run FLO27 on the IBM computer, several lines of code were added to modify the output format to a more usable form. A subroutine, VERTEC, which calls the Versatec plotter was also added to enhance program usefulness. This plotting routine is user controlled through an input variable and is explained in the next section. The modified FLO27 program source code is presented in Appendix E.
To facilitate program data entry several input variables which had recommended values were initialized to these values within the Main program and the subroutine GEOM. The initialized input variables and their values are presented in Table IV.

**TABLE IV**

*Initialized Input Variables*

<table>
<thead>
<tr>
<th>AREA</th>
<th>VARIABLE NAME</th>
<th>INITIALIZED VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN Prgm.</td>
<td>XSCAL</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PSCAL</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PCONT</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>P20</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>P30</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>FSMCO</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PTMAP</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>BLCP</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>WEIG</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>PTCK</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>FIX</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>YSYM</td>
<td>0.0</td>
</tr>
<tr>
<td>Subrt. GEOM</td>
<td>FNB</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>PX</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PZ</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>TRL</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>SLT</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>XSING</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>YSING</td>
<td>0.0</td>
</tr>
</tbody>
</table>
A complete description of each input variable in Table IV can be found on pages 19 through 23 of [Ref. 5].

B. PROGRAM DESCRIPTION

The FLC27 program is a computer code written to analyze the transonic flow over a wing alone or a wing on a cylindrical fuselage. It uses a finite-volume formulation to solve the exact potential flow equation in conservative form. In the development of the equations, the basic assumptions are: steady flow, no heat or work transfer, isentropic flow, irrotational flow, no body forces and a perfect gas. The velocity vector in cartesian coordinates is

\[
\mathbf{v} = u \hat{i} + v \hat{j} + w \hat{k}
\]  

(2.1)

where \(u\), \(v\) and \(w\) are the velocity components. The continuity equation, assuming steady flow, is

\[
\frac{\partial}{\partial x} (\rho u) + \frac{\partial}{\partial y} (\rho v) + \frac{\partial}{\partial z} (\rho w) = 0
\]  

(2.2)

Next a velocity potential is introduced such that the velocity components are calculated as the gradient of this potential.

\[
u = \phi_x, \quad v = \phi_y, \quad w = \phi_z
\]  

(2.3)
With the introduction of the velocity potential, the continuity equation 2.2 becomes

$$\frac{\partial}{\partial x}(\rho \phi_x) + \frac{\partial}{\partial y}(\rho \phi_y) + \frac{\partial}{\partial z}(\rho \phi_z) = 0 \quad (2.4)$$

Assuming no heat or work transfer, the energy equation can be written as

$$T \left[ 1 + \frac{(\gamma - 1)}{2} M^2 \right] = T_\infty \left[ 1 + \frac{(\gamma - 1)}{2} M_\infty^2 \right] \quad (2.5)$$

The flow is assumed to be uniform in the far field. On the surface of the body, the normal velocity component is zero. The velocities and densities of the near field are normalized using the free stream velocity and density, thus \( V_\infty = 1 \) and \( \rho_\infty = 1 \). Using the assumptions that the flow is isentropic and a perfect gas, the energy equation 2.5 can be shown to be

$$\rho = \left[ 1 + \frac{(\gamma - 1)}{2} M^2 \right] \left( 1 - v^2 \right) \frac{1}{\gamma - 1} \quad (2.6)$$

With equations 2.5 and 2.6 there are only two unknowns, \( \phi \) and \( \rho \). They can be solved, subject to the boundary condition of flow tangency, using a finite volume technique. The basic numerical scheme for the solution is the
construction of a mesh from small volume elements (cubes) which are packed around the wing or wing body configuration. The cubes in the computational domain are separately mapped to distorted cubes in the physical domain by independent transformations from local coordinates \( X, Y \) and \( Z \) to Cartesian coordinates \( x, y \) and \( z \). The mesh points are the vertices (corners) of the mapped cubes. The velocity potential and density are calculated at each vertex in the mesh. The pressure distribution can then be calculated from

\[
P = \frac{\rho \sigma}{\gamma M_\infty^2}
\] (2.7)

In the event that the local flow velocity becomes supersonic and shocks occur, these are handled in the usual manner by insuring that:

1) The tangential velocity components are equal on each side of the shock.

2) Continuity is maintained by keeping the product of \( \rho U_n \) constant across the shock (where \( U_n \) is the normal velocity component).

3) Discontinuous expansions (corresponding to an "expansion shock") are excluded from the flow field.
The assumption of isentropic flow along with the existence of shocks presents a contradiction which can only be resolved by limiting the flow to very weak shocks for which entropy and vorticity generation may be ignored. Thus, solutions will be valid only for subsonic free stream velocities.

The main three-dimensional array containing the potential function data is stored on disk, and special unformatted input/output statements are used to bring planes of data into central computer memory and to store updated planes of data back on the disk. In the construction of the computational coordinate system, a Joukowski transformation is used to transform the cylindrical fuselage to a vertical slit and then a sheared parabolic transformation is used in planes containing the airfoil sections. A detailed mathematical formulation of the potential flow analysis is contained in [Ref. 7].

1. Program Input

The input to FLO27 consists of variables which are read with an 8F10.6 FORMAT. Each input card has a title card which precedes it. This title card contains the input variable name and effectively labels the input data for easy
reference. The title for each input data is placed in the same column as the input data it labels. The title cards are read with a 20A4 FORMAT. All numerical input values are real numbers. The following data deck, listed card by card, is the minimum input data required for a simple wing analysis. Each "card" can be interpreted as one line of data on your terminal. A complete sample data set is presented in Appendix C.

CARD 1 The Run Title (64 characters maximum)

CARD 2 Title card for the input variables

   FNX, FNY, FNZ, FMESH and FPILOT

CARD 3

Cols. 1-10 FNX - Number of computational cells in the chordwise direction for the initial mesh.

MAX = 160/2**n, where n = FMESH - 1. (See Cols. 31-40 for FMESH)

Cols. 11-20 FNY - Number of computational cells in the normal direction from the airfoil surface for the initial mesh.

MAX = 16/2**n, where n = FMESH - 1.

Cols. 21-30 FNZ - Number of computational cells in the spanwise direction for the initial mesh.

MAX = 32/2**n, where n = FMESH - 1.
Cols. 31-40 FMESH - Determines the number of times a program generated computational mesh is refined. Enter only 1.0, 2.0 or 3.0 for coarse, medium or fine mesh. If 3.0 is selected the program will calculate flow over the wing for the coarse mesh then half the mesh size (medium), recalculate, then half the mesh again (fine) and do a final potential flow calculation. Output parameters are printed for each mesh size for which calculations were performed.

Cols. 41-50 PPLOT - Output flag

0.0 = Normal output without printer-plot of Cp
1.0 = Normal output with printer-plot of Cp at each computational mesh point for each wing section.
2.0 = Normal output with Versatec plots of Cp versus X/C for each wing section of the final mesh.

CARD 4 Title card for the input variables FIT, COVO and P10

CARD 5-M One card for each computational mesh. Total number of cards equal to M = FMESH.
Cols. 1-10 FIT - A parameter which fixes the maximum number of iterations the program will use to converge the velocity potential to a specified tolerance (COVO). This parameter must be repeated for each mesh refinement.

Cols. 11-20 COVO - Velocity potential convergence criteria. This input variable is also entered for each selected mesh. A value of 0.000001 is recommended.

Cols. 21-30 P10 - This parameter determines the subsonic point relaxation factor for the specified mesh size. A value of less than 2.0 must be entered for each designated mesh. Recommended values are: 1.6 for coarse, 1.3 for medium and 1.2 for the fine mesh.

CARD 6 Title card for the input variables
FMACH, YA, AL and CDO

CARD 7
Cols. 1-10 FMACH - Free stream Mach number
Cols. 11-20 YA - Yaw angle in degrees
Cols. 21-30 AL - Angle of attack in degrees
Cols. 31-40 CDO - Drag coefficient due to skin friction. Unless known, an estimated value of 0.01 is recommended.

CARD 8 Title card for the input variables
ZSYM, FNS, SWEEP, DIHED and FUS

CARD 9

Cols. 1-10 ZSYM - The wing planform symmetry trigger.

0.0 = Yawed wing, has no spanwise symmetry
1.0 = Swept wing, has spanwise symmetry

Cols. 11-20 FNS - This input variable tells the program the total number of wing sections you have selected to define the wing half span. The number must be at least three (3) but not more than eleven (11) sections.

Cols. 21-30 SWEEP - Leading edge sweep angle in degrees.

Cols. 31-40 DIHED - Dihedral angle in degrees. See Fig. 2.1.

Cols. 41-50 FUS - Input the fuselage radius. Enter 0.0 for a wing-alone case.
Data input cards from 10 through 15 are used for defining wing planforms and section geometrics. For the first wing section, all data cards from card 10 through card 15 must be used. For the second and subsequent sections there is an option for skipping the wing section defining data (cards 12 through 15) and copying the data from that of the previous section. This option is controlled by the input variable FSEC. If this option is not used, data cards from 10 through 15 must be repeated for each wing defining section. The number of wing sections which are defined is input with the variable FNS. Remember, up to 11 sections may be defined, and a minimum of 3 sections must be defined. All wing planform and section defining geometrics must be in consistent units. Wing planform and section defining quantities are presented in Fig. 2.2.
CARD 10  Title card for the input variables

ZS, XI, YL, CHORD, THICK, AT and PSEC

CARD 11

Cols. 1-10 ZS - The section spanwise coordinate
(Start at the centerline and work outboard)

Cols. 11-20 XL - Section leading edge X coordinate
Cols. 21-30 YL - Section leading edge Y coordinate
Cols. 31-40 CHORD - Section chord length

Figure 2.2. Wing Defining Geometry

FNS = 5 defining sections
Cols. 41-50 THICK - The thickness scaling factor can be used to scale all Y coordinates of the wing section. Thus percent thickness and camber are increased (or decreased) accordingly. Use 1.0 if no scaling is desired.

Cols. 51-60 AT - The twist angle of each section (geometric twist) measured from the X axis to the chord line. A positive twist angle reduces the section angle of attack and gives "washout". Use 0.0 for no twist.

Cols. 61-70 FSEC - This is a flag which determines whether or not the program reads wing section defining geometry from a previous wing section or from new defining geometry. For the first section defined you must set FSEC to 1.0. Following the first section, if you define new section geometry then use FSEC = 1.0. If you want the program to read the section geometry defined from the previous section then set FSEC = 0.0.

CARD 12 Title card for the input variable FN
CARD 13

Ccls. 1-10 FN - This variable contains the number of points which define the upper and lower surface of the section. A maximum of 161 points may be used.

CARD 14 Title card for the input variables

XF(I) and YP(I)

CARDS 15-1 to 15-N Total number of cards equals N, where N = integer part of (FN+2)/3.

The X and Y coordinates at each point are entered in pairs, three points to a card. (See Appendix C for sample input)

Ccls. 1-10 XP(I) - X coordinate of the wing section point

11-20 YP(I) - Y coordinate of the wing section point

21-30 defining X coordinate for next point

31-40 defining Y coordinate for next point

41-50 defining X coordinate for following point
defining Y coordinate for following point

The X and Y coordinates of the wing section defining points must be entered starting with the upper surface trailing edge point and proceeding along the upper surface to the leading edge, and returning along the lower surface to the lower surface trailing edge point. It is very important to define the section leading edge with a large number of closely spaced points. Suggest at least 0.05 spacing or less between X coordinate values from 0.1 X/C to the leading edge, X/C = 0.0. Each X and Y coordinate point is normalized using the chord length for that section. Section defining geometrics are illustrated in Fig. 2.3.

![Diagram of wing section with X and Y coordinates](image)

Figure 2.3. Section Defining Geometry
CARD 16  Title card containing the words in Cols. 1-80

END OF CALCULATIONS

CARD 17  Title card for the input variable

FNX

CARD 18

Cols. 1-10 FNX - This variable indicates the end of a set of calculations and must be set equal to 0.0. Its purpose is to indicate that the program has run to completion.

2. Program Output

Output from the FLC27 program varies with the value of the input variable FPLCT. When FPLCT is set equal to 0.0 a normal output is produced. This normal output contains (in order of occurrence): refined input geometry data including trailing edge slope and angle calculations; iterative solution of the potential flow mesh; section characteristics and wing characteristics. The iterative solution, section and wing characteristic data are repeated for each mesh refinement requested. Thus, if the input variable FMESH is set equal to 3, these data are calculated and output three times. The last data in the normal output consists of the non-dimensionalized chord (X/C) and pressure...
coefficient (Cp) data at each computational mesh point for each wing section calculated during the final mesh. A sample of the normal output data is presented in Appendix D and represents the output data from Appendix C input data.

If variable FPLT is set equal to 1.0, the output data is increased considerably. This output contains the normal output plus a line printer-plot of the pressure coefficient at each computational mesh point for each wing section. The line printer-plot is produced for each wing section of each mesh refinement. The length of the output data with FPLT set equal to 1.0 can approach 6000 records depending on the number of mesh refinements requested. These plots are of questionable value and, therefore, an alternate plotting program was developed.

When the variable FPLT is set equal to 2.0, the normal output data is produced plus a Versatec plotting subroutine (VERTEC) is called. The subroutine outputs, via the Versatec plotter, plots of Cp versus X/C for each wing section of the final mesh calculations. This routine is simply putting into plot form the Cp and X/C numerical data contained in the normal output. A sample of the Versatec plot is presented in Fig. 2.4.
SECTION CP DATA

* = UPPER SURFACE
+ = LOWER SURFACE

Figure 2.4. Versatec Plot of Cp vs. X/C
III. INTERACTIVE INPUT PROGRAM FLO27IN

The input data file required for the FLO27 program is extensive. Errors in input data FORMAT will cause program errors at execution time. In order to eliminate these errors and reduce the input data workload, a computer terminal interactive program was written. This interactive program, called FLO27IN, is a user-friendly way of creating an input data file for the potential flow wing analysis program FLO27. The FLO27IN program source code is presented in Appendix F.

The interactive program, FLO27IN, when executed asks the user questions in order to construct and write to the user's "A" disk the required FLO27 input data file. The following presents the step-by-step procedure for executing the interactive program FLO27IN.

STEP #1---Log on to any IBM 3033 interactive terminal with your user number and password.

STEP #2---Once logged on and in the CMS operation mode type:

    CP LINK 0247P 191 120 RR  then hit ENTER

STEP #3---The word PASSWORD will appear, Type and ENTER

AERO

35
STEP #4 --- Type and ENTER

ACC 120 D

STEP #5 --- Type and ENTER

LOAD FLO27IN (START)

The screen will display the header for the interactive program. Answer each question presented. At the end of each question in parenthesis is the input data variable associated with that question and whether the input parameter is a real number (R) or an integer (I). Example:

==> Enter the free stream Mach number (FMACH): (R). FMACH is the input data variable for the question. As you proceed through the FLC27IN program, opportunities to review and change input data will be presented. Should it become necessary to change your input data after completing the FLO27IN program, you can simply XEDIT the created data file.

The FLO27IN program also incorporates a library which contains the wing-section defining data for a number of current wing shapes. A copy of this library is presented in Appendix B. This feature will be displayed during program execution by the use of a menu from which the user can select a pre-defined wing section or define his own.
Upon completion of user inputs to the interactive program three additional data lines are automatically written to the bottom of the input file. They are:

END OF CALCULATION

FNX

0.0

In addition, Job Control Language (JCL) cards are written to the top and bottom of the file. All JCL cards start with a // format. After FLO27IN has run to completion type and enter RELEASE 191 to release the aero disk which was linked while executing the FLO27IN program. The created data file is written to the user's "A" disk with <filename> <filetype> of FLO27 DATAIN. Additional changes can be made simply by entering the XEDIT mode and editing the file.
IV. FLO27 BATCH SYSTEM EXECUTION

The potential flow program FLO27 can be executed after the input data file has been created. The batch processor is required for FLO27 execution because of the extensive CPU time needed to run the program. While in the XEDIT mode, a standard JOB card must be added to the top of the FLO27 DATAIN file prior to submission for job execution. The JOB card has the form:

//jobname JOB (nnnn, pppp), 'ident', CLASS=J

jobname = may contain up to 8 alphanumeric characters,
the first of which must be alphabetic.

nnnn = your user number
pppp = project number, assigned by professor
'ident' = contains the user's own identification information. A maximum of 20 characters may be contained within the single quotation marks.

After adding the JOB card to your data file, you are ready to execute the program. Type SUBMIT FLO27 DATAIN and press ENTER. Batch runs are normally not worth waiting for. To inquire about the status of the job, enter INQ and the job name used on the JOB card or "logoff". If the system is busy and the maximum mesh size was selected, it may be several hours before your job is run.
When the job is run the output will be spooled to the batch printer located next to the VM printer in the main computer building. The title at the top of the printout for batch jobs is the name entered on the JOB card. If it is desired to have the program output data spooled directly to the terminal, it will be necessary to add one additional JCL card to the input data set. This card must be placed immediately following the JOB card and has the form:

//*MAIN ORG=NPGVM1.nnnnP

where nnnn = your user number

Inserting this card in the input data will cause all program output to be spooled to the user's virtual reader where it may be looked at, printed or transferred to his "A" disk. To enquire as to whether information is in the reader simply type RDR and hit enter, then follow the instructions on the screen.
V. PROGRAM TEST RESULTS

The FLC27 program was tested in three stages; (1) during the reprogramming phase for conversion completeness, (2) after successful conversion with suitable wing data for program accuracy and (3) during an AE-4501 class project.

A. ACCEPTANCE TEST DATA

To test and ensure that the FLO27 program was converted to IEM compatible Fortran without error, an acceptance test data set was used. The acceptance test input and output data was supplied with the original CDC program source code. After conversion of the FLO27 program to Fortran suitable for the NPS IBM system, the acceptance test input data were run and the output results compared to the output generated by the CDC system.

Both output data sets were numerically exact when the FLO27 program was run in double precision on the IBM system. If the program was run in single precision, the numerical output values were exact to the third decimal place. The difference in single precision accuracy occurs because the CDC system uses a 64 bit word length while the IBM system word length in single precision is only 32 bits. It was
decided that the IBM single precision accuracy was satisfactory.

B. COMPARISON WITH OTHER PROGRAMS

The FLO27 program was also tested for accuracy by using the wing planform and section data from a NACA 572 wing. The data were run on both the FLO27 program and the Douglas potential flow program [Ref. 1]. The data generated by both programs was compared to wind tunnel data for the NACA 572 wing [Ref. 8]. The results are presented in Fig. 5.1 as plots of lift coefficient versus angle-of-attack. The results show that for the NACA 572 wing the FLO27 program more accurately predicts the wing lift coefficient than does the Douglas program.

C. AE-4501 CLASS PROJECT

The final test phase was conducted by introducing the FLO27 program into the AE-4501 course as a class project. This was accomplished to determine student problems/comments concerning the data input program FLO27IN and to test an additional wing shape. The wing chosen for study was that of the A-7 airplane. The A-7 wing has a distinct leading edge notch at the approximate mid-span. When the planform geometry was run with the notch included the FLO27 program
Figure 5.1. Program Calculated and Wind Tunnel Data

ran to completion but gave negative values for section and total induced drag coefficient. The value for the lift coefficient was low for the freestream Mach and angle-of-attack used. It was found that if the notch was
excluded from the wing geometry input data the program results were satisfactory both for induced drag and lift coefficient.

From the AE-4501 class experience it was determined that sharp wing planform discontinuities cannot be handled by the program. If however, the changes in shape are gradual, such as a wing glove, the program output appears to be satisfactory. Such was the case with the acceptance test case data where the wing geometry was that of the F-8 supercritical wing which incorporates a wing glove.
This JCL routine allocates sufficient space on the mass storage system to store the entire tape contents.

```
//JACK JOB (3266,0178), PASCHALL-2759, CLASS=A
//*MAIN ORG=NEGVM1.3266P
//EXEC PGM=IEFBFR14
//DD1 DD UNIT=3330V, MSGVP=PU84C, DISP=(NEW,CATLG),
// DSN=MSS.S3266.WFLOW.DATA, SPACE=(CYL, (16,4,2))
/*

This JCL routine is used to transfer all tape files to a partitioned data set in the mass storage system.

```

```
//JACK JOB (3266,0178), PASCHALL-2759, CLASS=J
//*MAIN ORG=NEGVM1.3266P
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=3400-6, VCL=SER=WFLOW, DISP=(OLD,PASS),
// LABEL=(&FILE, BLK, IN)
// DCB=(RECFM=F, BLKSIZE=80, DEN=3, OPTCD=Q)
// SYSUT2 DD DISP=(OLD, KEEP),
// DSN=MSS.S3266.WFLOW SOURCE(&MEM),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=6400)
//PEN
//EXEC COPY, FILE=1, MEM=FL027
//EXEC COPY, FILE=2, MEM=A411N
//EXEC COPY, FILE=3, MEM=WIN
//EXEC COPY, FILE=4, MEM=A411A01
//EXEC COPY, FILE=5, MEM=INTER
//EXEC COPY, FILE=6, MEM=A411P2
//EXEC COPY, FILE=7, MEM=A411P1
//EXEC COPY, FILE=8, MEM=A411P1
//EXEC COPY, FILE=9, MEM=COUPLE
//EXEC COPY, FILE=10, MEM=ITER
//EXEC COPY, FILE=11, MEM=DATAIN
//EXEC COPY, FILE=12, MEM=FINAL
//EXEC COPY, FILE=13, MEM=BOEB1
//EXEC COPY, FILE=14, MEM=CONTPLT
//EXEC COPY, FILE=15, MEM=CORDPLT
//EXEC COPY, FILE=16, MEM=STREPLT
//COPY2 PROC FILE=, MEM=, LRECL=80, BLK=6400
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=3400-6, VOL=SER=WFLOW, DISP=(OLD,PASS),
// LABEL=(&FILE, BLK, IN)
// DCB=(RECFM=F, BLKSIZE=&LRECL, DEN=3, OPTCD=Q)
// SYSUT2 DD DISP=(OLD, KEEP), DSN=MSS.S3266.WFLOW DATA(&MEM),
// DCB=(RECFM=FB, LRECL=&LRECL, BLKSIZE=&BLK)
//PEND
//EXEC COPY2, FILE=17, LRECL=150, BLK=5000, MEM=OUTF27
//EXEC COPY2, FILE=18, LRECL=150, BLK=5000, MEM=OUTIPC
//EXEC COPY2, FILE=19, LRECL=150, BLK=5000, MEM=OUT411L
//EXEC COPY2, FILE=20, LRECL=150, BLK=5000, MEM=OUT411U
/*/```
C This JCL routine moves all source code files from mass storage to the MVS 004 disk which can be accessed by entering GET MVS then following the screen instructions to move source files to your disk. If you want to move the data files to MVS 004 then change the word SOURCE to DATA in the JCL program below.

//JACK JOB (3266.0178), 'FASCHALL=2759', CLASS=A
//*MAIN ORG=NPGVM1.3266P
// EXEC PGM=IEBCOPY
// SYSPRINT DD SYSOUT=A
// FROM DD DISP=SHR, DSN=MSS.S3266.WFLOW.SOURCE
// INTO DD UNIT=3350, VOL=SER=MVS.004, DISP=(NEW,KEEP),
// SPACE=(CYL, (16, 4, 10), RLSE), DSN=S3266.SOURCE
// SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (2,2))
// SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (2,2))
// SYSIN DD *
// COPY OUTDD=INTC, INDD=PFM
//
//
APPENDIX B

LIBRARY OF AIRFOIL SECTION GEOMETRIES

0 = user input section coordinate data
1 = flat plate data
2 = symmetrical wing (11% thickness at 30% chord)
3 = supercritical wing (cambered, 12% thickness at 32% chord)
4 = NACA 24-30-0 (cambered, 12% thickness at 30% chord)
5 = F-14 wing (cambered, 9.5% thickness at 37% chord)
6 = A-7 wing (7 deg droop at 20% chord, 7% thickness at 43% chord)
7 = LISSAMAN 7769 Airfoil (cambered, 11% thickness at 30% chord)
8 = NACA 0010 (symmetrical, 10% thickness at 30% chord)
9 = NACA 0010-34 (symmetrical, 10% thickness at 40% chord)
10 = NACA 0010-35 (symmetrical, 10% thickness at 50% chord)
11 = NACA 0010-64 (symmetrical, 10% thickness at 40% chord)
12 = NACA 0010-66 (symmetrical, 10% thickness at 60% chord)
13 = NACA 16-009 (symmetrical, 9% thickness at 50% chord)
14 = NACA 63-010 (symmetrical, 10% thickness at 35% chord)
15 = NACA 63A010 (symmetrical, 10% thickness at 35% chord)
16 = NACA 64-010 (symmetrical, 10% thickness at 40% chord)
17 = NACA 64A010 (symmetrical, 10% thickness at 40% chord)
18 = NACA 65-010 (symmetrical, 10% thickness at 40% chord)
19 = NACA 65A010 (symmetrical, 10% thickness at 40% chord)
20 = NACA 66-010 (symmetrical, 10% thickness at 45% chord)
APPENDIX C

THIS APPENDIX PRESENTS A COMPLETE INPUT DATA SET INCLUDING THE JCL CARDS REQUIRED TO EXECUTE THE PROGRAM FLC27

// (STANDARD JOB CARD - SEE MVS USER'S GUIDE NO. MVS-01)
// EXEC FLC27
// GO SYSIN CC *
SAMPLER DATA (NACA 572 WING SECTION)

<table>
<thead>
<tr>
<th>FNX</th>
<th>FNY</th>
<th>FNZ</th>
<th>FMESH</th>
<th>FLOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0</td>
<td>4.0</td>
<td>6.0</td>
<td>3.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

FIT

<table>
<thead>
<tr>
<th>100.0</th>
<th>O</th>
<th>Q</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCC001</td>
<td>1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20.0</th>
<th>O</th>
<th>Q</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCC001</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5.0</th>
<th>O</th>
<th>Q</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCC001</td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FMACF

<table>
<thead>
<tr>
<th>YA</th>
<th>AL</th>
<th>CD0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

ZSYM

<table>
<thead>
<tr>
<th>FNS</th>
<th>SLEEP</th>
<th>DIFED</th>
<th>FUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>3.0</td>
<td>3.2</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZS</th>
<th>XL</th>
<th>YL</th>
<th>CHORD</th>
<th>THICK</th>
<th>AT</th>
<th>FSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

FN

<table>
<thead>
<tr>
<th>41.0</th>
<th>XP(I)</th>
<th>YP(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.000000</td>
<td>0.00001300</td>
</tr>
<tr>
<td>0.8</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>0.6</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>0.2</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>0.1</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>0.0</td>
<td>0.000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

ZS

<table>
<thead>
<tr>
<th>XL</th>
<th>YL</th>
<th>CHORD</th>
<th>THICK</th>
<th>AT</th>
<th>FSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.0</td>
<td>6.0</td>
<td>6.1307</td>
<td>0.0</td>
<td>6.4500</td>
<td>1.0000</td>
</tr>
<tr>
<td>19.0</td>
<td>12.2614</td>
<td>0.0</td>
<td>4.3000</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

END OF CALCULATION
APPENDIX D

THIS APPENDIX PRESENTS THE FLG27 OUTPUT DATA PRODUCED FROM THE INPUT DATA OF THE PREVIOUS APPENDIX.

A46C MODIFIED FROM FLG27 OF ANTICNY JAMESON, CURANT INSTITUTE
THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW USING FINITE VOLUME SCHEME
NACA 572 WING SECTION
FUSELAGE RAC

<table>
<thead>
<tr>
<th>PROFILE AT Z = 0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWEEP</td>
</tr>
<tr>
<td>LIHED</td>
</tr>
<tr>
<td>32.3275</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TE ANGLE</th>
<th>TE SLOPE</th>
<th>X SING</th>
<th>Y SING</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.0633</td>
<td>-0.0720</td>
<td>0.0024</td>
<td>-0.0000</td>
</tr>
</tbody>
</table>

NL = 21, X(FNL) = 0.0

<table>
<thead>
<tr>
<th>X(P), Y(P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00000  -0.00130</td>
</tr>
<tr>
<td>0.60000  -0.12606</td>
</tr>
<tr>
<td>0.20000  -0.04220</td>
</tr>
<tr>
<td>0.03000  -0.02270</td>
</tr>
<tr>
<td>0.0      0.0</td>
</tr>
<tr>
<td>0.03000  0.02900</td>
</tr>
<tr>
<td>0.20000  0.07260</td>
</tr>
<tr>
<td>0.60000  0.06420</td>
</tr>
<tr>
<td>1.00000  0.00130</td>
</tr>
</tbody>
</table>

SECTION DEFINITION AT Z = 0.0

<table>
<thead>
<tr>
<th>XLE</th>
<th>YLE</th>
<th>CHORD</th>
<th>THICKNESS RATIO</th>
<th>TWIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>8.6000</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>YMIN</td>
<td>JMIN</td>
<td>YMAX</td>
<td>JMAX</td>
<td>YDIFF</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>6.4500</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>-0.4220E-01</td>
<td>10</td>
<td>0.7880E-01</td>
<td>33</td>
<td>.1210</td>
</tr>
</tbody>
</table>

SECTION DEFINITION AT Z = 9.68750

<table>
<thead>
<tr>
<th>XLE</th>
<th>YLE</th>
<th>CHORD</th>
<th>THICKNESS RATIO</th>
<th>TWIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>6.4500</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>YMIN</td>
<td>JMIN</td>
<td>YMAX</td>
<td>JMAX</td>
<td>YDIFF</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>6.4500</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>-0.4220E-01</td>
<td>10</td>
<td>0.7880E-01</td>
<td>33</td>
<td>.1210</td>
</tr>
</tbody>
</table>

SECTION DEFINITION AT Z = 19.37500

<table>
<thead>
<tr>
<th>XLE</th>
<th>YLE</th>
<th>CHORD</th>
<th>THICKNESS RATIO</th>
<th>TWIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>4.3000</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>YMIN</td>
<td>JMIN</td>
<td>YMAX</td>
<td>JMAX</td>
<td>YDIFF</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>4.3000</td>
<td>1.0000</td>
<td>0.0</td>
</tr>
<tr>
<td>-0.4220E-01</td>
<td>10</td>
<td>0.7880E-01</td>
<td>33</td>
<td>.1210</td>
</tr>
</tbody>
</table>
|   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | four
### SECTION CHARACTERISTICS

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>YAW</th>
<th>ANG OF ATTACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10000</td>
<td>0</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPAN STATION</th>
<th>CL</th>
<th>CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.20238</td>
<td>-7.1088E-02</td>
</tr>
<tr>
<td>3.87499</td>
<td>0.24033</td>
<td>-2.2462E-02</td>
</tr>
<tr>
<td>7.74999</td>
<td>0.26515</td>
<td>-4.8115E-02</td>
</tr>
<tr>
<td>11.62498</td>
<td>0.28127</td>
<td>-5.7074E-02</td>
</tr>
<tr>
<td>15.4998</td>
<td>0.28695</td>
<td>-6.5431E-02</td>
</tr>
<tr>
<td>19.37497</td>
<td>0.24168</td>
<td>-7.0588E-02</td>
</tr>
</tbody>
</table>

### WING CHARACTERISTICS

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>YAW</th>
<th>ANG OF ATTACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10000</td>
<td>0</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>YAW</th>
<th>CD FCRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25575</td>
<td>-3.2534E-02</td>
<td>1.0000E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CM YAW</th>
<th>CM ROLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.00467</td>
<td>0.23919</td>
</tr>
<tr>
<td>ITERATIVE SOLUTION</td>
<td>ANG CF ATTACK</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>MACH NO     YAW</td>
<td>2.0000</td>
</tr>
<tr>
<td>0.10000     0.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NX</th>
<th>NY</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RELAX FCT 1</th>
<th>RELAX FCT 2</th>
<th>RELAX FCT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAX CORRENC</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>AVG CORRENC</th>
<th>MAX RESIDAL</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.70000</td>
<td>64</td>
<td>4</td>
<td>0.10046E-03</td>
<td>0.16464E-02</td>
<td>2</td>
<td>8</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>0.1421E-02</td>
<td>2</td>
<td>2</td>
<td>0.10463E-04</td>
<td>0.12200E-02</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.5853E-03</td>
<td>2</td>
<td>2</td>
<td>0.10529E-04</td>
<td>0.10263E-02</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.3194E-03</td>
<td>2</td>
<td>2</td>
<td>0.10421E-04</td>
<td>0.84352E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.3258E-03</td>
<td>2</td>
<td>2</td>
<td>0.36229E-04</td>
<td>0.72680E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1848E-03</td>
<td>2</td>
<td>2</td>
<td>0.3175E-04</td>
<td>0.63685E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1643E-03</td>
<td>2</td>
<td>2</td>
<td>0.2867E-04</td>
<td>0.58517E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1491E-03</td>
<td>2</td>
<td>2</td>
<td>0.2630E-04</td>
<td>0.5727E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1368E-03</td>
<td>2</td>
<td>2</td>
<td>0.2461E-04</td>
<td>0.5367E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1268E-03</td>
<td>2</td>
<td>2</td>
<td>0.2310E-04</td>
<td>0.5033E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1748E-03</td>
<td>2</td>
<td>2</td>
<td>0.2151E-04</td>
<td>0.4643E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.1083E-03</td>
<td>2</td>
<td>2</td>
<td>0.2062E-04</td>
<td>0.4264E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.890E-04</td>
<td>2</td>
<td>2</td>
<td>0.1553E-04</td>
<td>0.4039E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.9236E-04</td>
<td>2</td>
<td>2</td>
<td>0.1529E-04</td>
<td>0.3847E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.8513E-04</td>
<td>2</td>
<td>2</td>
<td>0.1759E-04</td>
<td>0.3674E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.7899E-04</td>
<td>2</td>
<td>2</td>
<td>0.1672E-04</td>
<td>0.3470E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.7356E-04</td>
<td>2</td>
<td>2</td>
<td>0.1517E-04</td>
<td>0.3283E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.6547E-04</td>
<td>2</td>
<td>2</td>
<td>0.1515E-04</td>
<td>0.3097E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.6559E-04</td>
<td>2</td>
<td>2</td>
<td>0.1445E-04</td>
<td>0.2963E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0.6196E-04</td>
<td>2</td>
<td>2</td>
<td>0.1378E-04</td>
<td>0.2872E-03</td>
<td>80</td>
<td>7</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAX RESIDAL 1</th>
<th>MAX RESIDAL 2</th>
<th>WORK</th>
<th>RELUCTCYCLE</th>
<th>CONV</th>
<th>TOLERANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1646E-02</td>
<td>-0.2872E-03</td>
<td>19.0000</td>
<td>0.9122</td>
<td>0.1000E-05</td>
<td></td>
</tr>
<tr>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SECTION CHARACTERISTICS

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>YAW</th>
<th>ANG OF ATTACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10000</td>
<td>0</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPAN STATION</th>
<th>CL</th>
<th>CC</th>
<th>GM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>22.093</td>
<td>-1.815E-01</td>
<td>-1.0667</td>
</tr>
<tr>
<td>1.93750</td>
<td>24.319</td>
<td>-1.533E-02</td>
<td>-1.0667</td>
</tr>
<tr>
<td>3.87499</td>
<td>26.053</td>
<td>-1.4154E-02</td>
<td>-1.0667</td>
</tr>
<tr>
<td>5.81249</td>
<td>27.508</td>
<td>-1.44517E-02</td>
<td>-1.0667</td>
</tr>
<tr>
<td>7.74995</td>
<td>28.683</td>
<td>-7.2064E-03</td>
<td>-1.1125</td>
</tr>
<tr>
<td>9.68749</td>
<td>29.643</td>
<td>-1.1954E-02</td>
<td>-1.1175</td>
</tr>
<tr>
<td>11.62499</td>
<td>30.359</td>
<td>-1.6262E-02</td>
<td>-1.1192</td>
</tr>
<tr>
<td>13.56248</td>
<td>30.863</td>
<td>-2.0637E-02</td>
<td>-1.1192</td>
</tr>
<tr>
<td>15.50998</td>
<td>30.746</td>
<td>-2.7385E-02</td>
<td>-1.1181</td>
</tr>
<tr>
<td>17.43747</td>
<td>29.141</td>
<td>-4.1055E-02</td>
<td>-1.1090</td>
</tr>
</tbody>
</table>

### WING CHARACTERISTICS

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>YAW</th>
<th>ANG OF ATTACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10000</td>
<td>0</td>
<td>2.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CL</th>
<th>CD FORM</th>
<th>CD FRICTION</th>
<th>CD</th>
<th>L/D FORM</th>
<th>L/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27663</td>
<td>8715E-03</td>
<td>10000E-01</td>
<td>1.0872E-01</td>
<td>317.42</td>
<td>25.0</td>
</tr>
<tr>
<td>CM YAW</td>
<td>CM RCLL</td>
<td>CM PITCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.00145</td>
<td>0.25615</td>
<td>-0.35963</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITERATIVE SOLUTION</td>
<td>ANG OF ATTACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MACH NO</td>
<td>YAW</td>
<td>NY</td>
<td>N2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>160</td>
<td>32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RELAX FCT 1</th>
<th>RELAX FCT 2</th>
<th>RELAX FCT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.20000</td>
<td>0.70000</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION</th>
<th>MAX CORRECNI</th>
<th>J</th>
<th>K</th>
<th>AVG CORRECNI</th>
<th>MAX RESIDAL</th>
<th>i</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.95314E-03</td>
<td>17</td>
<td>24</td>
<td>0.20565E-04</td>
<td>-0.13855E-02</td>
<td>2</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>-0.96428E-03</td>
<td>17</td>
<td>23</td>
<td>0.12764E-04</td>
<td>0.98673E-03</td>
<td>160</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0.42692E-03</td>
<td>17</td>
<td>23</td>
<td>0.96247E-05</td>
<td>0.87373E-03</td>
<td>160</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>-0.24824E-03</td>
<td>15</td>
<td>15</td>
<td>0.75284E-05</td>
<td>0.73407E-03</td>
<td>160</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>-0.21125E-03</td>
<td>15</td>
<td>16</td>
<td>0.71403E-05</td>
<td>0.62240E-03</td>
<td>160</td>
<td>15</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAX RESIDAL 1</th>
<th>MAX RESIDAL 2</th>
<th>WORK</th>
<th>RELUCTN/CYCLE</th>
<th>CONV TOLERANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1385E-02</td>
<td>0.62240E-03</td>
<td>4.0000</td>
<td>0.8197</td>
<td>0.1000E-05</td>
</tr>
<tr>
<td>SPAN STATION</td>
<td>CD</td>
<td>CM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.56875</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.93750</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.90625</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.87499</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.84374</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.81249</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.78124</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.74999</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.71874</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.68749</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.65624</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.62499</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.59373</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.56624</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.53123</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.49998</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.46872</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.43747</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.40622</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.37497</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A46C MODIFIED FROM FL27 OF ANT CNY JAMESON, CURRENT INSTITUTE
THREE DIMENSIONAL WING ANALYSIS IN TRANSSONIC FLOW USING FINITE VOLUME SCHEME
BOEING VERSION, PREPARED BY DR. HAI-CHOW CHEN STANDARD BOEING INPUT FORMAT
END OF CALCULATION
<table>
<thead>
<tr>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000</td>
<td>0.372240</td>
<td>0.208949</td>
<td>0.517071</td>
<td>-0.341032</td>
<td>0.546485</td>
<td>-0.32254</td>
<td></td>
</tr>
<tr>
<td>0.846730</td>
<td>0.301548</td>
<td>0.607694</td>
<td>0.725998</td>
<td>0.639486</td>
<td>0.237831</td>
<td>0.672056</td>
<td>-0.26293</td>
</tr>
<tr>
<td>0.706152</td>
<td>0.317394</td>
<td>0.739502</td>
<td>0.190774</td>
<td>0.774368</td>
<td>-0.124863</td>
<td>0.810659</td>
<td>0.06446</td>
</tr>
<tr>
<td>0.846511</td>
<td>-0.360272</td>
<td>0.883730</td>
<td>0.014850</td>
<td>0.921712</td>
<td>0.068296</td>
<td>0.960470</td>
<td>0.14686</td>
</tr>
</tbody>
</table>

SPAN STATION = 4.84374
NO. OF DATA POINTS = 101

<table>
<thead>
<tr>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000</td>
<td>0.231922</td>
<td>0.960492</td>
<td>0.208446</td>
<td>0.521777</td>
<td>0.163078</td>
</tr>
<tr>
<td>0.846730</td>
<td>0.108152</td>
<td>0.810396</td>
<td>0.083923</td>
<td>0.774854</td>
<td>0.063487</td>
</tr>
<tr>
<td>0.706152</td>
<td>0.032375</td>
<td>0.725558</td>
<td>0.030516</td>
<td>0.640641</td>
<td>0.032970</td>
</tr>
<tr>
<td>0.578250</td>
<td>-0.016888</td>
<td>0.468305</td>
<td>0.006876</td>
<td>0.410269</td>
<td>0.000563</td>
</tr>
<tr>
<td>0.420550</td>
<td>-0.016888</td>
<td>0.468305</td>
<td>0.006876</td>
<td>0.410269</td>
<td>0.000563</td>
</tr>
<tr>
<td>0.360542</td>
<td>-0.016888</td>
<td>0.468305</td>
<td>0.006876</td>
<td>0.410269</td>
<td>0.000563</td>
</tr>
<tr>
<td>0.270527</td>
<td>-0.116851</td>
<td>0.250026</td>
<td>-0.117327</td>
<td>0.213941</td>
<td>-0.011536</td>
</tr>
<tr>
<td>0.193235</td>
<td>-0.137934</td>
<td>0.175822</td>
<td>-0.165175</td>
<td>0.159224</td>
<td>-0.197249</td>
</tr>
<tr>
<td>0.128543</td>
<td>-0.188053</td>
<td>0.114441</td>
<td>-0.166050</td>
<td>0.101126</td>
<td>-0.156833</td>
</tr>
<tr>
<td>0.076568</td>
<td>-0.206717</td>
<td>0.065637</td>
<td>-0.229674</td>
<td>0.055557</td>
<td>-0.219107</td>
</tr>
<tr>
<td>0.038050</td>
<td>-0.093051</td>
<td>0.030435</td>
<td>-0.057772</td>
<td>0.024313</td>
<td>-0.133514</td>
</tr>
<tr>
<td>0.013378</td>
<td>-0.261988</td>
<td>0.009184</td>
<td>-0.249590</td>
<td>0.006139</td>
<td>-0.430500</td>
</tr>
<tr>
<td>0.001527</td>
<td>-0.613352</td>
<td>0.002626</td>
<td>-0.625460</td>
<td>0.002116</td>
<td>-0.695311</td>
</tr>
<tr>
<td>0.000000</td>
<td>-0.452094</td>
<td>0.003592</td>
<td>-0.456903</td>
<td>0.006166</td>
<td>-0.318244</td>
</tr>
<tr>
<td>0.127560</td>
<td>-0.174114</td>
<td>0.087041</td>
<td>-0.392090</td>
<td>0.022359</td>
<td>-0.025596</td>
</tr>
<tr>
<td>0.035734</td>
<td>-0.155529</td>
<td>0.091073</td>
<td>-0.329758</td>
<td>0.035080</td>
<td>-0.500977</td>
</tr>
<tr>
<td>0.072512</td>
<td>-0.613255</td>
<td>0.084319</td>
<td>-0.588666</td>
<td>0.096642</td>
<td>-0.455938</td>
</tr>
<tr>
<td>0.124444</td>
<td>-0.481384</td>
<td>0.139316</td>
<td>-0.505073</td>
<td>0.155014</td>
<td>-0.524044</td>
</tr>
<tr>
<td>0.188946</td>
<td>-0.524989</td>
<td>0.207137</td>
<td>-0.520740</td>
<td>0.226132</td>
<td>-0.513344</td>
</tr>
<tr>
<td>0.266567</td>
<td>-0.510275</td>
<td>0.288015</td>
<td>-0.497852</td>
<td>0.310273</td>
<td>-0.481921</td>
</tr>
<tr>
<td>0.357205</td>
<td>-0.445434</td>
<td>0.381869</td>
<td>-0.426582</td>
<td>0.407327</td>
<td>-0.403531</td>
</tr>
<tr>
<td>0.460618</td>
<td>-0.374505</td>
<td>0.488450</td>
<td>-0.358377</td>
<td>0.517012</td>
<td>-0.342105</td>
</tr>
<tr>
<td>0.576962</td>
<td>-0.301668</td>
<td>0.607694</td>
<td>-0.272325</td>
<td>0.639466</td>
<td>-0.237235</td>
</tr>
<tr>
<td>0.705355</td>
<td>-0.172896</td>
<td>0.395093</td>
<td>-0.149667</td>
<td>0.774368</td>
<td>-0.123731</td>
</tr>
<tr>
<td>0.846611</td>
<td>-0.034986</td>
<td>0.883730</td>
<td>0.016212</td>
<td>0.921712</td>
<td>0.068296</td>
</tr>
<tr>
<td>1.000000</td>
<td>0.217983</td>
<td>1.000000</td>
<td>0.217983</td>
<td>1.000000</td>
<td>0.217983</td>
</tr>
</tbody>
</table>

SPAN STATION = 5.81249
NO. OF DATA POINTS = 101
<table>
<thead>
<tr>
<th>SPAN STATION</th>
<th>S.68749</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. OF DATA POINTS</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>C.237601</td>
<td>0.960492</td>
<td>0.254040</td>
<td>0.521777</td>
<td>0.164577</td>
<td>0.883857</td>
<td>0.13210</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.846730</td>
<td>C.106719</td>
<td>0.610396</td>
<td>0.607738</td>
<td>0.774854</td>
<td>0.067702</td>
<td>0.740105</td>
<td>0.04595</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.706152</td>
<td>C.391011</td>
<td>0.723558</td>
<td>0.036240</td>
<td>0.640642</td>
<td>0.038692</td>
<td>0.609077</td>
<td>0.02814</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.705355</td>
<td>C.175008</td>
<td>0.379502</td>
<td>0.151621</td>
<td>0.774386</td>
<td>0.124948</td>
<td>0.810059</td>
<td>0.08611</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.846511</td>
<td>C.035192</td>
<td>0.883750</td>
<td>0.016212</td>
<td>0.521712</td>
<td>0.069618</td>
<td>0.960470</td>
<td>0.14531</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPAN STATION</th>
<th>10.65624</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. CF DATA POINTS</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
<th>X/C</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000</td>
<td>C.237601</td>
<td>0.960492</td>
<td>0.254040</td>
<td>0.521777</td>
<td>0.164577</td>
<td>0.883857</td>
<td>0.13210</td>
</tr>
<tr>
<td>0.846730</td>
<td>C.106719</td>
<td>0.610396</td>
<td>0.607738</td>
<td>0.774854</td>
<td>0.067702</td>
<td>0.740105</td>
<td>0.04595</td>
</tr>
<tr>
<td>0.706152</td>
<td>C.391011</td>
<td>0.723558</td>
<td>0.036240</td>
<td>0.640642</td>
<td>0.038692</td>
<td>0.609077</td>
<td>0.02814</td>
</tr>
</tbody>
</table>
APPENDIX E

THIS APPENDIX PRESENTS THE SOURCE CODE FOR THE POTENTIAL FLOW PROGRAM FLO27.

C******************************************************************************FLG27******************************************************************************9/12/83******************************************************************************
C**
C** THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW USING
C** FINITE VOLUME SCHEME WITH SHEARED PARABOLIC COORDINATES
C** PROGRAMMED BY ANTHONY JAMESON, JANUARY-APRIL 1977
C**
C******************************************************************************
C** THE BOEING VERSION OF FLO27 WAS PREPARED BY CR. HAI-CHOW
C** CHENG WITH THE FOLLOWING MODIFICATIONS
C** 1) TEMPORARY STORAGE OF THE LARGE CORE MEMORY REQUIREMENTS
C** HAS BEEN IMPLEMENTED TO REDUCE THE COMPUTING COSTS
C** BY BUFFERING DATA IN AND OUT OF CORE.
C**
C** 2) STANDARD BOEING INPUT FORMAT FOR THE WING SECTION
C** HAS BEEN USED.
C**
C** 3) SUBPROGRAM BLIN HAS BEEN IMPLEMENTED TO ADD THE
C** DISPLACEMENT THICKNESS TO THE ORIGINAL WING SECTIONS
C**
C** 4) WING SECTION LEADING EDGE SINGULAR POINT IS FOUND BY
C** COMPUTING THE FOCUS OF A PARABOLA BY 2N+1 POINTS
C** LEAST-SQUARE FIT CENTERED AT THE LEADING EDGE POINT.
C** N IS SUPPLIED BY THE USER THROUGH INPUT CARD.
C**
C** 5) TRAILING EDGE CLOSURE ANGLE AND BISECTOR SLOPE ARE
C** COMPUTED BASED ON BACKWARD DIFFERENCE.
C**
C** 6) PRINT FOR PRINTER-Plotting CF THE UNWRAPPED
C** WING SECTIONS IS AVAILABLE
C**
C******************************************************************************
C******************************************************************************

THE FOLLOWING FILES ARE USED TO EXECUTE FLO27. SOME OF THESE
FILES ARE USED SUBSEQUENTLY IN OTHER MODULES OF THE VISCOUS/
INVISICL INTERACTIVE WING SYSTEM

FILE1 IS USED TO BUFFER DATA IN AND OUT OF CORE
FILE2 IS USED TO BUFFER DATA IN AND OUT OF CORE
FILE3 IS USED TO BUFFER DATA IN AND OUT OF CORE
FILE4 IS USED TO READ IN THE VELOCITY POTENTIAL
GENERATED PREVIOUSLY

FILE8 IS WRITTEN FOR DATA TRANSFER TO BEEING
TURBULENT BOUNDARY LAYER PROGRAM A411.

FILES IS USED TO SAVE SECTION SURFACE PRESSURE
AGAINST X/C.

FILE10 IS USED TO SAVE THE SECTION X, Y, Z
LOCATION FOR A411 IN WHICH CALCULATES THE CORRESPONDING
DISPLACEMENT THICKNESS
WHERE X, Y, Z SHOULD BE THE WING SURFACE LOCATION
FOR THE CURRENT RUN

FILE11 IS USED TO READ IN THE DISPLACEMENT
THICKNESS FROM A411 IN

FILE12 IS USED TO REPLACE PART OF THE INPUT CARDS
BY CARD IMAGES

TAPE13 IS USED TO REPLACE PART OF THE CUTOFF
SKIPPED FROM THE LINE PRINTER

TAPE14 IS USED TO SAVE THE VELOCITY POTENTIAL
FOR FUTURE USE

**************************************************************************************
**************************************************************************************
COMMON G(161,18,3), S0(161,35), VORT(115), ZV(115),
1 IV(161,35), ITE1(35), ITE2(35),
2 AO(161), BO(161), XO(35), YC(35), ZO(35), SCAL(35),
3 NX, NY, N2, KTE1, KTE2, ISYM, KSYM, FUS,
4 YAW, CYAW, SYAW, ALPHA, CA, SA, FMACH, N1, N2, N3, 10
COMMON /CPF/ NL
COMMON /FCR/ NC
COMMON /FLO/ P1, P2, P3, FRES, IRES, JRES, KRES, ARES, CG, IG, JG, KG, AG, NSUP
COMMON /PARMT3/ XI3(161), Y3(161), Z3(161),
1 UT3(161), VT3(161), WT3(161), N01, N2
COMMON /PRS/ X0CDI(161)
DIMENSION XS(161,11), YS(161,11), ZS(11), XLE(11), YLE(11),
1 SLOFT(11), TRAIL(11), NP(11), EL(11), E2(11), E3(11),
2 DI(161), D2(161), D3(161), SN(161),
3 SV(161), SM(161), CP(161), XP(161), YP(161),
4 YP0(35), ZP0(35), XM(35), YMIN(35), YMAX(35), YMIN(35),
5 CHORD1(35), SCL1(35), SCG1(35), SCC1(35), TITLE(20),
DIMENSION Fit(2), COV(3), PI0(3), P20(3), P30(3), FSMOD(3),
1 RES(201), CCUNI(201), FLMAP(3),

DUMX(161), DUMY(161), DELR(161)

** INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **

```
XSCAL = 0.0
PSCL = 0.0
FCENT = 0.0
BLCP = 0.0
WEIG = 1.0
PTCK = 0.0
```

**

```
ND = 161
NE = 161
KPLCT = 0
IPLCT = 1
ISTCP = 2
N1 = 1
N2 = 2
N3 = 3
REIND 1
REIND 2
REIND 4
REIND 10
REIND 13
REIND 14
J0 = 0
RAD = 57.295779513082
```

1 WRITE (6,600)
WRITE (6,601)

2 FORMAT(*CA460 MODIFIED FROM FLC27 CF ANTONY JAMESON*,
1 'Courant Institute')/
1 FOR THREE DIMENSIONAL WING ANALYSIS IN TRANS sonic FLOW,
1 'USING FINITE VOLUME SCHEME/
1 'SCHENG VERSION, PREPARED BY DR. HAI-CHENG CHEN',
1 * STANDARD BOEING INPUT FORMAT FOR WING SECTION DATA IS USED*)
READ (5,530) TITLE
WRITE (6,630) TITLE
READ (5,500)
READ (5,510) FNX, FNY, FNZ, FMESH, FPLCT
AK = FNX
NY = FNY
NZ = FNZ
MMESH = FMESH
IF (NX .LT. 1) GO TO 301
KPLCT = ABS(FPLCT)
READ (5,500)
```
DO 12 NP=1,PMESH
C*******************************************************************************
C** INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **
C*******************************************************************************
P2C(NM)= 0.7
P3C(NM)= 1.0
FSGO(NM)= 0.0
PT inflatable(NM)= 0.0
C*******************************************************************************
12 READ (5,510) FIT(NM),CCVO(NM),PI0(NM)
READ (5,506)
READ (5,510) FMACH,YA,AL,COO
CALL GECM (NO,NS,NP,XS,YS,ZS,XLE,YLE,SLOPT,TRAIL,XP,YP,
1 FUS,XTEG,CHORDO,ZTIP,SWEEP,CIHED,
2 FIX,FX,P2,ISYMO,ksym)
ALPHA = AL/RAD
IF (BLCM.LE.0.0) GO TO 44
IF (PTCK .GE. 1.) WRITE (6,600)
READ (11) (TITLE(I),I=1,8),FMACH,ALPHA,NS
DO 40 K = 1,NS
READ (11) NFCK
NP(K) = NFCK
NPCK1 = NFCK + 1
READ (11) (DUMX(NFCK1-I),DUMZ,DUMY(NFCK1-I),CEL(R(NPCK1-I),
1 I=1,NPCK)
IF (PTCK .LE. 0.0) GO TO 30
WRITE (6,52) K,NPCK,NPCK1
WRITE (6,54) (DUMX(NPCK1-I),DUMZ,DUMY(NFCK1-I),
1 CELR(NPCK1-I),I=1,NPCK)
WRITE (6,54) (XS(I,K),YS(I,K),I=1,NPCK)
30 CONTINUE
CALL BLIN (XS(1,K),YS(1,K),DEL,R,WEIG,NFCK,NL)
40 CONTINUE
44 CONTINUE
IF (PTCK .LT. 0.0) GO TO 56
IF (PTCK .GE. 1.) WRITE (6,600)
WRITE (10) (TITLE(I),I=1,8),FMACH,ALPHA,NS
DO 50 K=1,NS
NPCK = NP(K)
NPCK1 = NPCK + 1
DO 48 I = 1,NPCK
DUMX(NFCK1-I) = XS(I,K) + XLE(K)
DUMY(NFCK1-I) = YS(I,K) + YLE(K)
48 CONTINUE
WRITE (10) NPCK
WRITE (10) (DUMX(I),ZS(K),DUMY(I),I=1,NFCK)
IF (PTCK .LE. 0.0) GO TO 49
WRITE (6,52) K, NFK, NPCK1, ZS(K)
WRITE (6,54) (DUMX(I), ZS(K), DUMY(I), I=1, NPCK)
CONTINUE
ENDFILE 10
FORMAT (1HO, 5X, 3I5, F11.4)
FORMAT (12F11.4)
CONTINUE
IF (KSYM .NE. 0) YA = 0.
ISYM = ISYM0
IF (AL .NE. 0.) ISYM = 0
YA = YA/RAD
CYA = COS(YAW)
SYA = SIN(YAW)
CA = CYA*COS(ALPHA)
SA = CYA*SIN(ALPHA)
IF (FCCNT .LT. 1.) GC TC 91
REAL (4) NX, NY, NZ, NM, K1, K2, NIT
MX = NX + 1
MY = NY + 1
HZ = NZ + 3
DO 62 K = 1, HZ
READ (4) ((G(I,J,1), I=1,MX), J=1,MY)
BUFFER CLT(N3, I) (G(I,1,1), G(MX,MY,1))
WRITE (N3) ((G(I,J,1), I=1,MX), J=1,MY)
IF (UNIT(N3).GT.0.) GO TO 1
BUFFER CLT(N1, I) (G(I,1,1), G(MX,MY,1))
WRITE (N1) ((G(I,J,1), I=1,MX), J=1,MY)
IF (UNIT(N1).GT.0.) GO TO 1
CONTINUE
READ (4) (VORT(K), K=K1, K2)
REWIND N3
REWIND N1
REWIND 4
CALL CCRD (NX, NY, NZ, KSYM, ZTIP, XLIM, ZLIM,
SY, AX, AY, AZ, PX, PZ, AC, BC, ZO)
CALL SINGL (NS, NZ, KSYM, KTE1, KTE2, FUS, CHCRDO, ZS, XLE, YLE,
SWEEP, DIHED, X0, Y0, Z0, YPO, ZPO, E1, E2, E3, IND)
CALL SUFF (ND, NE, NS, NX, NZ, ISYM, KSYM, KTE1, KTE2,
YAW, KTEO, XLM, FIX, XP, XS, YS, ZS, SLOP, TRAIL,
AO, AX, ZC, SO, SCAL, ZV, IV, ITE1, ITE2,
XP, YP, SN, D1, D2, D3, IND)
IF (INC .EQ. 0) GC TC 291
IF (FCCNT .GE. 1.) GO TO 101
NIT = 1
NIT = 0
CALL ESTIM
IF (IG.EQ.0) GO TO 1
REWIND N2
REWIND N1
101 IF (PTCK .GE. 1.) WRITE (6,600)
  FCCNT = 0.
  COV = COVG(NM)
  P1 = P10(NM)
  P2 = P20(NM)
  P3 = P30(NM)
  MIT = FIT(NM) + NIT
  KIT = MIT
  IF (KIT.LT.2) KIT=2
  J1T = NIT
  LRES = 0.
  MRES = (MIT - NIT - 2)/200 + 2
  NRES = 0.
  NX = 0.
  MY = 0.
  MZ = 0.
  NY = 0.
  NY = 0.
  NZ = 1.
  IF (KSYM.EQ.0) GC TC 103
  K1 = 3.
  K2 = 1.
  IF (KSYM.NE.0) LZ = 3
  IF (PTCK .LE. 0.) GC TC 108
  WRITE (6,104)
104 FORMAT (48HINDICATION OF LOCATION OF WING AND VORTEX SHEET,
    1Z7H IN COORDINATE PLANE Y = 0. /,
    2Z7H((I(V1(I,K),K=K1,K2),1=1,MX))
  DO 106 I=1,MX
106 WRITE (6,650) (IV1(I,K),K=K1,K2)
106 CONTINUE
  IMAF = IMAF(NM)
  IF (IMAF .EQ. 0) GC TC 830
  WRITE (6,600)
  WRITE (6,112)
112 FORMAT (46HCHORDWISE CELL DISTRIBUTION IN SQUARE ROOT PLANE,
    1Z4H AND MAPPED SURFACE COORDINATES AT CENTER LINE AND TIP)
  DO 820 ISEC = L2,KTEZ,IMAP
    WRITE (6,812) ISEC, ZP0(ISEC)
812 FORMAT (15H0 I 1 )
820 CALL PP(Y (2,NX,AO,SO(1,1SEC))
830 CONTINUE
  IF (PTCK .LE. 0.) GC TC 130
  WRITE (6,116)
116 FORMAT(15F0, TE LOCATION ,15H POWER LAW )
WRITE (6,610) XLIM,AX
WRITE (6,600)
WRITE (6,118)
116 FORMAT(46H NORMAL CELL DISTRIBUTION IN SQUARE ROOT PLANE/ 
1   15F0 
DO 120 =1,KY
120 WRITE (6,610) BC(J)
WRITE (6,122)
122 FORMAT(15F0 SCALE FACTOR,15H POWER LAW )
WRITE (6,610) SY,AY
WRITE (6,600)
WRITE (6,124)
124 FORMAT(45H SPANWISE CELL DISTRIBUTION AND SINGULAR LINE/ 
1   15H0 
   Z ,15H X SING ,15H Y SING )
DO 126 K=K1,K2
126 WRITE (6,610) Z0(K),X0(K),Y0(K)
WRITE (6,128)
128 FORMAT(15H0 TIP LOCATION,15H POWER LAW )
WRITE (6,610) ZLIM,AZ
130 CONTINUE
WRITE (6,600)
WRITE (6,132)
132 FORMAT(15H0 ITERATIVE SOLUTION)
WRITE (6,134)
134 FORMAT(15H0 MACH NO ,15H YAW ,15H ANG OF ATTACK)
WRITE (6,610) FMACH,YA,AL
WRITE (6,136)
136 FORMAT(15H0 MX ,15H NY ,15H NZ )
WRITE (6,640) MX,NY,NZ
WRITE (6,138)
138 FORMAT(15H0 RELAX FCT 1 ,15H RELAX FCT 2 ,15H RELAX FCT 3 )
WRITE (6,610) P10(NM),P20(NM),P30(NM)
WRITE (6,140)
140 FORMAT(10H ITERATION, 
1   15H MAX CORRECTION ,4H I ,4F J ,4H K ;15H AVG CORRECTION ; 
2   15H MAX RESIDUAL ,4H I ,4F J ,4H K ;15H AVG RESIDUAL ; 
3   12H CIRCULATION,15H SONIC PIS)
141 NIT = NIT +1
JIT = JIT +1
CALL MIXFLG
IF (10.EC.0) GO TO 151
JO = 0
REWIND A1
REWIND A2
N = N1
N1 = N2
N2 = N3
N3 = N
WRITE (6,66C) NIT, DG, IG, JG, KG, AG, FRES, IFES, JRES, KRES, ARE,
1 1
19C(T[1Z]), NSUP
LRES = LRES + 1
IF (LRES.EQ.MRES) LRES = 1
IF (LRES.NE.1) GO TO 143
NRES = NRES + 1
COUNT(NRES) = NIT - 1
RES(NRES) = FRES
143 IF (JIT.GE.KIT) GO TO 251
IF (NM.LE.1 OR NM.LT.MMESH) GO TO 148
IF (ABS(CG).LE.COV) GO TO 251
148 CONTINUE
IF (NIT.LT.MIT.AND.ABS(DG).GT.COV.AND.ABS(DG).LT.10.) GO TO 141
GO TO 161
151 IF (JQ.EQ.1) GO TO 1
REIND N1
REIND N2
JQ = 1
N1 = N3
N2 = N2
N3 = N1
N1 = N
GO TO 141
160 RATE = 0.
IF (NRES.GT.1) RATE = (ABS(RES(NRES)/RES(1))
1 1
**1/(COUNT(NRES) - COUNT(1)))
WRITE (6,162)
162 FORMAT(15HO MAX RESIDAL 1,15H MAX RESIDAL 2,15H WORK 1
15H REDUCTN/CYCLE, 15H CONV TOLERANCE)
WRITE (6,670) RES(1), RES(NRES), COUNT(NRES), RATE
1,COV
WRITE (6,600)
DO 164 K = 1,3
C BUFFER IN (N1,1) (G1,J,M),G(MX,MY,MJ)
REAL (N1,ERROR) 151 (G(I,J,M),I=1,MX,J=1,MY)
C
164 CONTINUE
LX = NX/2 + 1
K = 2
KKK = 0
C WRITE HEADER ON TAPE 8
IF (NM.LT.MMESH) GO TO 17C
REIND E
REIND E
REIND E
NRC = KIE2 - KTE1 + 1
WRITE (E) (TITLE(I), I=1,8), FMACH, ALPHA, NRC
17C CONTINUE
171  K = K + 1
    IF (K.EQ.M2) GO TO 191
    DO 172 J=1,MY
    DO 172 I=1,MX
      G(I,J,1) = G(I,J,3)
    172   G(I,J,2) = G(I,J,3)
    C    BUFFER IN (NI,11) (G(I,J,3),G(MX,MY,3))
    C    READ (NI,ERR=151) ((G(I,J,3),I=1,MX),J=1,MY)
    C    IF (UNIT(IN),GT.0.) GO TO 151
    IF (K.LT.KTE1 .OR. K.GT.KTE2) GO TO 171
    CALL VELG (K, SV, SM, CP, XP, YP, XMAX(K), XMIN(K), YMAX(K), YMIN(K))
      II = ITEL(K)
      I2 = ITE2(K)
      CHORD(K) = XP(II) - XP(LX)
      CALL FCFCF (II, I2, XP, YP, CP, AL, CHORD(K), XO(K), YPO(K),
             1 SCL(K), SCD(K), SCM(K))
      KKK = KKK + 1
      IF (KPLCT.GT.1 .AND. K.GT.KTE1) GC TC 185
      IF (KPLCT.EQ.0 .AND. KKK .GT. 1) GO TO 185
    WRITE (6,600)
    WRITE (6,182)
182 FORMAT (24HOSECTON CHARACTERISTICS/
           1 15H0 MACH NO ,15H YAW ,15H ANG OF ATTACK)
    WRITE (6,610) FMACH,YA,AL
    WRITE (6,184)
184 FORMAT (15H SPAN STATIC)
    Z = ZPC(K)
    IF (M1.LE.0) GC TC 850
    IF (KPLCT.LE.2) CALL CPOZ (2,NX,FMACH,XP,YP,CP,SM,II,12,KPLOT)
850  CONTINUE
    C    WRITE CNE FILE ON TAPE 8
    IF (NM.LT.MMESH) GO TO 186
    WRITE (E) NCI
    WRITE (E) (XT3(I),ZT3(I),YT3(I),UT3(I),HT3(I),VT3(I),I=1,NCI)
    NRC=I2-I1+1
    C    WRITE CP VS X/C SECTION DATA FOR FINAL MESH ON TAPE 9
    WRITE (6,900) ZPO(K)
    WRITE (6,910) NRD
    WRITE (6,920)
    WRITE (6,950) (XOCD(J),CP(J),J=I1,12)
900  FORMAT (/1HSpan Station =,F12.5)
910  FORMAT (1X,20HNC, CF DATA POINTS =,I5)
920  FORMAT (1X,6H X/C,8X,2HCP,7X,3H/C,8X,2HCP,7X,3H/C,8X,2HCP,
           17X,3H/C,8X,2HCP,3X)
950  FORMAT (8F10.6)
186  CONTINUE
C WHEN KPLOT = 2 CALL SUBROUTINE VERTEG WHICH PLOTS CP VS X/C
C FOR EACH SECTION OF THE FINAL MESH
C
IF (KPLCT.EQ.2.AND.NM.EQ.MMESH) CALL VERTEG(I1,I2,XQCD,CP,ARD,
   1   ZPO,FMACH,YA,AL,SCL,SCD,SCM,K)
GO TO 171
171 CONTINUE
IF (NM.LT.MMESH) GO TO 200
ENCFILE 8
REWRITE 8
ENDFILE 8
REWRITE 5
REWRITE 9
CONTINUE
200 CONTINUE
CALL TCFOR (KTE1,KTE2,CHORD,SCL,SCD,SCM,XC,YPO,ZPO,
   1   CL,CD0,CMF,CMR,CMY)
CD1 = CYAW+CD1
CD = CD0 + CD1
VLC = 0
IF (ABS(CD1).GT.1.E-6) VLD1 = CL/CD
VLC = 0
IF (ABS(CD).GT.1.E-6) VLD = CL/CD
WRITE (*,6,600)
WRITE (*,6,192)
192 FORMAT(21H01ING CHARACTERISTICS/
   1   15H MACH NO, 15H YAW, 15H ANG OF AITACK)
   1   WRITE (6,610) FMACH,YA,AL
   1   WRITE (6,154)
   1   WRITE (6,194)
194 FORMAT(15H0 MACH NO, 15H CD FORM, 15H CD FRICTION )
   1   WRITE (6,610) CL,CC0,CC1,CD,CD,CD,VC1,VLC
   1   WRITE (6,196)
196 FORMAT(15H0 MACH NO, 15H CM RCLL, 15H CM PITCH )
   1   WRITE (6,610) CMY,CMR,CMF
REWRITE 1
IF (KPLOT.LT.1) GO TO 201
C CALL RPLTT (IPLCT,FFRES,CONT,TITLE,FMACH,YA,AL,NX,NY,NZ)
C CALL DRAW (IPLCT,XMAX,XMIN,YMAX,YMIN,ZPO,FUS,TITLE,NZ,KTE1,KTE2)
C CALL TFREED (IPLCT,SV,SM,CP,XP,YP,ZPO,TITLE,YA,AL,
C   1   VLC,CL,CD,CHORD,XSCAL,PSCAL)
   1   IF (I0.LT.0) GO TO 151
201 IF (1STCP.EQ.1) GO TO 301
IF (NM.LT.MMESH) GO TO 203
GO TO 1
203 CONTINUE
NX = NX +NX
NY = NY +NY
NZ = NZ +NZ
CALL CCCRD (NX, NY, NZ, KSYM, ZTIP, XLIN, ZLIN,
  1
CALL SINGL (NS, NZ, KSYM, KTE1, KTE2, FUS, CHCROD, ZS, XLE, YLE,
  1
CALL SUFF (NO, NE, NS, NX, NZ, KSYM, ZKTE1, ZKTE2,
  1
YAW, XTEO, XLIN, FIX, NP, XS, YS, ZS, SLOPT, TRAIL,
  2
AO, XO, ZO, SO, SCAL, LV, IV, ITE1, ITE2,
  3
XP, YP, SN, D1, D2, D3, IND)
IF (INC.EQ.0) GC TO 291
CALL REFIN
IF (ID.EQ.0) GO TO 221
REIND N1
REIND N2
NSMOD = FSMCC(NM)
IF (NSMCC.LT.1) GO TO 211
DO 202 K=1,NSMCC
REIND N1
REIND N2
221 NX = NX/2
NY = NY/2
NZ = NZ/2
CALL CCCRD (NX, NY, NZ, KSYM, ZTIP, XLIN, ZLIN,
  1
CALL SINGL (NS, NZ, KSYM, KTE1, KTE2, FUS, CHCROD, ZS, XLE, YLE,
  1
CALL SUFF (NO, NE, NS, NX, NZ, KSYM, ZKTE1, ZKTE2,
  1
YAW, XTEO, XLIN, FIX, NP, XS, YS, ZS, SLOPT, TRAIL,
  2
AO, XO, ZO, SO, SCAL, LV, IV, ITE1, ITE2,
  3
XP, YP, SN, D1, D2, D3, IND)
GO TO 1C1
GO TO 151
251 K1 = KTE1 - 1
K2 = KTE2 + ITE2(KTE2) - NX/2
WRITE (14) NX, NY, NZ, NM, K1, K2, NIT
DO 262 K=1,NZ
C BUFFER IA (N1, 1) (G(I, J, I), GIMX, MY, 1)
READ (N1, ERH=281) ((G(I, I, I), I=1, MX), J=1, MY)
C
IF (UNIT(N1).GT.0.) GO TO 281
C
262 WRITE (14) ((G(I, J, J), I=1, MX), J=1, MY)
REMAIN 11
WRITE (14) (VORT(K), K=K1,K2)
ENDFILE 14
REMAIN 14
CALL SSWITCH(1, ISTOP)
CALL SLISET(1, ISTOP)
IF (ISTCP.EQ.1) GO TO 161
JIT
IF (NIT.LT.MIT.AND.ABS(DG).GT.COVC.AND.AES(DG).LT.10.) GC TO 141
GO TO 161
281 REMAIN 4
GO TO 151
291 WRITE (6,600)
WRITE (6,292)
292 FORMAT(24X,OBAD DATA, SPLINE FAILURE)
GO TO 1
301 CONTINUE
REMAIN 10
REMAIN 11
C IF (KPLCT.GT.0) CALL PLOT(0.,0.,955)
STCP
50C FORMAT(1X)
51C FORMAT(E10.6)
53C FORMAT(2CA4)
60C FORMAT(1H1)
61C FORMAT(F15.5,7G15.5)
63C FORMAT(1H0,20A4)
64C FORMAT(18,7I15)
65C FORMAT(1X,33I3)
66C FORMAT(110,E15.5,3I4,2E15.5,3I4,E15.5,F10.5,I10,F10.3)
67C FORMAT(2E15.4,2F15.4,E15.4)
END
C**SUBROUTINE BLIN**

SUBROUTINE BLIN (XT, YT, DELR, WEIG, NL)

SUBPROGRAM FOR NORMALLY ADDING THE DISPLACEMENT
THICKNESS TO THE ORIGINAL WING SECTIONS

XT: CONTAIN THE X COORDINATES OF THE ORIGINAL
WING SECTION WHEN CALLED
CONTAIN THE X COORDINATES OF THE DISPLACED
WING SECTION ON RETURN

YT: CONTAIN THE Y COORDINATES OF THE ORIGINAL
WING SECTION WHEN CALLED
CONTAIN THE Y COORDINATES OF THE DISPLACED
WING SECTION ON RETURN

DELR: THE DISPLACEMENT THICKNESS

COMMON /FLCR/ PTCK
DIMENSION XT(1), YT(1), DELR(1)
WRITE (6, 1000)
I = 1
X2 = XT(I)
Y2 = YT(I)
20C IF (I.EQ.N) GO TO 300
X3 = XT(I+1)
Y3 = YT(I+1)
30C IF (I.EQ.1) GO TO 400
X1 = 2.*X2 - X3
Y1 = 2.*Y2 - Y3
40C IF (I.EQ.N) GO TO 500
X3 = 2.*X2 - X1
Y3 = 2.*Y2 - Y1
50C CONTINUE
X12 = X1 - X2
X23 = X2 - X3
X31 = X3 - X1
IF (ABS(X31).LE.1.E-6) GO TO 600
DYX = -X23/(X12*X31)*Y1
DYX = DYX + (X12-X23)/(X12*X23)*Y2
DYX = DYX + X12/(X31*X23)*Y3
IF (ABS(DYX).LE.1.E-6) GO TO 820
DYX = -1./DYX
GO TO 700
60C DYYN = C
70C CONTINUE
IF (I.EQ.NL) GO TO 80
800 CONTINUE
SI = 1.
IF (I .LT. NL) SI = -1.
GO TO 850

820 CONTINUE
S = 0.
DL = DELR(I)
DX = 0.
DY = DL*SI
GO TO 850

850 CONTINUE
S = S + DYN**2
SS = SQRT(S)
SF = 1.
IF (DYN .LT. 0.) F = -1.
DL = DELR(I)
DX = S*F
DY = ABS(DYN)*S*SI
DX = DX + DL
DY = DY + DL
XT(I) = X2 + DX*WEIG
YT(I) = Y2 + DY*WEIG

880 CONTINUE
IF (PTCK .LE. 1.) GO TO 890
WRITE (6,1000) I,F,S,DL,DY,DX,CYXN,XT(I),YT(I),CYX,WEIG

890 CONTINUE
IF (I .EQ. N) GO TO 900
X1 = X2
Y1 = Y2
X2 = X3
Y2 = Y3
I = I+1
GO TO 240

900 CONTINUE
RETURN

1000 FORMAT (1H ,I5,F7.2,9G13.5)
END
**Subroutine**: PPXY

**Purpose**: FOR LINE PRINTER PLOTTING OF THE UNWRAPPED WINE SECTIONS

**Code**:

```fortran
C**SUBROUTINE PPXY******************************************************************
SUBROUTINE PPXY(I1,I2,X,Y)
C C SUBPROGRAM FOR LINE PRINTER PLOTTING OF THE UNWRAPPED C WINE SECTIONS C
C COMMON /PCKR/ PCK
COMMON /SFARE/ LINE(1GC)
DIMENSION X(I1),Y(I1)
DATA IB /1H/, IP /1H++, KMAX /100/, ACC /1.5/
1 IZ /1F++, ICONST /0/
DO 10 I=1,100
LINE(I) = IB
1C CONTINUE
YMAX = -1.0E35
YMIN = -YMAX
WIDTH = KMAX - 5
DO 20 I = 11,I2
YMAX = AMAX1(YMAX,Y(I))
YMIN = AMIN1(YMIN,Y(I))
2C CONTINUE
VAL = ABS(YMAX) + ABS(YMIN)
S = WIDTH/VAL
KK = 0
IF (ICCAST.LE.YMAX AND ICONST.GE.YMIN) KK = S*(YMAX-CONST)+ACC
IF (KK.LE.0) LINE(KK)=IZ
DO 30 I=11,I2
K = S*(YMAX-Y(I)) + ACC
IF (K.LT.1) K = 1
IF (K.GT.KMAX) K = KMAX
LINE(K) = IP
WRITE (6,100) I,X(I),Y(I),LINE
LINE(K) = IB
IF (K.EQ.KK) LINE(KK) = IZ
30 CONTINUE
RETURN
100 FORMAT (1X,13,2F10.4,4X,100A1)
END
```
SUBROUTINE LSQR (NL,NB,XP,YP,XSING,YSING)

SUBPROGRAM FOR WING SECTION LEADING EDGE SINGULAR POINT
CALCULATION BY MEANS OF COMPUTING THE FOCUS OF A
PARABOLA BY NB*2+1 POINTS LEAST-SQUARE FIT CENTERED AT
THE LEADING EDGE

NB: SUPPLY BY THE CALLING PROGRAM GEOM

COMMON /FCKR/ FCK
DIMENSION XP(1),YP(1)
N1 = NL - NB
N2 = NL + NB
N = N2 - N1 + 1
A1 = N
B1 = 0.
C1 = 0.
A2 = 0.
B2 = 0.
C2 = 0.
A3 = C.
B3 = 0.
C3 = 0.
D1 = 0.
D2 = 0.
D3 = 0.
SCALE = 100.
SCALE2 = 500.
DO 300 I = N1,N2
YY = (YP(I) - YP(NL))*SCALE
B1 = B1 + YY
YP2 = YP*YY
C1 = C1 + YP2
YP3 = YP2*YY
C2 = C2 + YP3
YP4 = YP3*YY
C3 = C3 + YP4
XX = XP(I)*SCALE2
D1 = D1 + XX
YX = YX + XX
D2 = D2 + YX
Y2X = YF2*XX
300   D3 = D3 + Y2X
A2 = B1
B2 = C1
A3 = C2
B3 = C2
FA1 = B2*C3 - B3*C2
FA2 = A3*C2 - A2*C3
FA3 = A2*B3 - A3*B2
DET = A1*FA1 + E1*FA2 + C1*FA3
DI = 1/DET
FA1 = FA1*DI
FA2 = FA2*DI
FA3 = FA3*DI
FB1 = (E3*C1 - E1*C3)*C1
FB2 = (A1*C3 - A3*C1)*DI
FB3 = (A3*B1 - A1*B3)*DI
FC1 = (E1*C2 - E2*C1)*C1
FC2 = (A2*C1 - A1*C2)*DI
FC3 = (A1*B2 - A2*B1)*DI
A = FA1*C1 + FB1*D2 + FC1*D3
B = FA2*D1 + FB2*D2 + FC2*D3
C = FA3*C1 + FB3*D2 + FC3*C3
B = B*SCALE
C = C*SCALE**2
A = A/SCALE2
B = B/SCALE2
C = C/SCALE2
XINC = A + (1. - E**2)*.25/C
YINC = -B*.5/C + YP(NL)
IF (Ptck .LE. 0.) G0 TO 520
WRITE (6,400) N
400 FORMAT (1H0,5X,'LEAST SQUARE MATRIX FOR LEADING EDGE',
1 I3,' FCINTS LEAST-SQUARE FIT'/)
WRITE (6,500) A1,B1,C1,D1
WRITE (6,500) A2,B2,C2,D2
WRITE (6,500) A3,B3,C3,D3
WRITE (6,500) A1,FA1,E1,FB1,FC1
WRITE (6,500) A2,FA2,E2,FB2,FC2
WRITE (6,500) A3,FA3,E3,FB3,FC3
WRITE (6,500) XINC,YINC,A,B,C,DET,DI
500 FORMAT (10G13.5)
520 CONTINUE
R2 = 0.
DXAM = C.
DX2M = C.
DO 650 I = N1,N2
  Y = YP(I) - YP(NL)
  X = A + B*Y + C*Y**2
  DX = X - XP(I)
  DX2 = DX*DX
  R2 = R2 + DX2
  DXA = ABS(DX)
  IF (DXAM .GE. DXA) GC TO 600
DXAM = CXA
DXAM = CXA
DX2M = DXAM**2

60C CONTINUE
IF (PTCK .LE. 0.) GC TC 65C
WRITE (6,70C) I, X, Y, DX, DX2, R2, CXAM, DX2M, XP(I), YP(I)

65C CONTINUE
IF (PTCK .LE. 0.) GC TC 75C
RA = R2/A1
WRITE (6,70C) N, RA, DXAM

70C FORMAT (113,9G13.5)
75C CONTINUE
RETURN
ENTRY LSC
IF (DXAM .LE. 1. E-4) RETURN
WRITE (6,80C) DXAM
80C FORMAT (1H0.5X, 'WARNING ??? DEVIATION OF THE LEADING EDGE POINTS',
1 ' FROM PARABOLA IS GREATER THAN 0.001'/6X,'DXAM = ',G13.4/
RETURN
END
SUBROUTINE GEOM (ND,NS,NP,XS,YS,ZS,XLE,YLE,SLOPT,TRAIL,XP,YP, 
   3 FUS,XTEO,CHCKEO,2TIP,Sweep,DIHED, 
   2 FIX,XP,PL,ISYM,KSYM)

GEOMETRIC DEFINITION OF WING

STANDARD BOEING INPUT FORMAT FOR WING SECTION DATA IS USED

OPTION FOR WING SECTION TRAILING EDGE CLOSURE ANGLE

AND BASELOR SLCP AUTOBATIC COMPUTED IS AVAILABLE

LEADING EDGE SINGULAR POINT CAN BE AUTOMATIC COMPUTED

BY INVOKING THE OPTION TO CALL LSQR

COMMON /PCKR/ PICK
COMMON /CPF/ NL

DIMENSION XS(ND,1),YS(ND,1),ZS(1),XLE(1),YLE(1), 
  1 SLOP(1),TRAIL(1),XP(ND),YP(ND),NP(1)

C** INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **

C**

FNB  = 2.0
PX   = 0.0
PZ   = 0.0
FIX  = 0.0
TRL  = 0.0
SLT  = 0.0
XSIW = 0.0
YSIW = 0.0
YSYM = 0.0

C**

RAD  = 57.295779513082
READ (6,50C)
READ (6,51O) ZSYM,FNS,SWEEP,DIHED,FUS
IF (FNS.LT.3) RETURN
KSYM  = ZSYM
IF (FUS.LE.0.) KSYM = 1
NS    = FNS
WRITE (6,2)
2 FORMAT(15H0,FUSELAGE RAD )
WRITE (6,61O) FLS
WRITE (6,4)
4 FORMAT(15H0, SWEEP ,15H CIHED )
WRITE (6,61O) SWEEP,DIHED
SWEEP = SWEEP/RAD
DIHED  = DIHED/RAD
ISYMO = 1
XTEC = 0.
CHORDO = 0.
K = 1

READ (5,500)
READ (5,510) ZS(K),XL,YL,CHORD,THICK,AL,FSEC
IF (K.EQ.1) ZS1 = ZS(1)
ALPHA = AL/RAD
IF (K.EQ.1 .AND. FSEC.EQ.0.) GO TO 31
READ (5,500)
READ (5,510) FN
N = FN
READ (5,500)
N1 = N + 1
READ (5,520) (XP(N1-I),YP(N1-I),I=1,N)

52C FORMAT (6F1C.0)
DO 26 I=1,N
IF (XP(I+1).LT.XP(I)) GC TC 26
NL = 1
GO TO 860
26 CONTINUE
860 CONTINUE
IF (FNE .LE. 0.) GC TO 21
DYL = YF(1) - YF(2)
DXL = XP(1) - XP(2)
DYU = YF(N) - YF(N-1)
DXU = XP(N) - XP(N-1)
TSU = EYL/DXU
TSL = EYL/DXL
TRL = ATAN2(DYL,DXL) - ATAN2(DYU,DXU)
SLT = TRL*RAD
NB = FAE
CALL LSCER (N,NE,XP,YP,XSING,YSING)
21 WRITE (6,600)
WRITE (6,622) ZS(K)
22 FORMAT(160PROFILE AT Z =,'F10.5/
1 15HO TE ANGLE ,15H TE SLCPE ,15H X SING ,
1 15H Y SING )
WRITE (6,610) TRL,SLT,XSING,YSING
27 WRITE (6,620) NL,XP(NL),YP(N),XP(N),YP(N),I=1,N
62C FORMAT ('/// NL = ' , F10.5 , ' /// (XP,YP) ///
1 (ZX,XSING,YSING) )
31 SCALE = CHORD/(

! Scale calculations for chord
!!
! XLE(K) = XL +(XSING -XP(NL))*THICK*SCALE
! YLE(K) = YL +(YSING -YP(NL))*THICK*SCALE
! XX = XP(NL) +(XSING -XP(NL))*THICK
! YY = YP(NL) +(YSING -YP(NL))*THICK

}
CA = COS(ALPHA)
SA = SIN(ALPHA)
DO 32 I=1,N
XS(I,K) = SCALE*(XP(I) - XX)*CA + THICK*(YP(I) - YY)*SA
32 YS(I,K) = SCALE*THICK*(YP(I) - YY)*CA - (XP(I) - XX)*SA
SLOPE(K) = THICK*SLT - TAN(ALPHA)
TRAIL(K) = THICK*TRI/RAD
NP(K) = N
CHORDO = AMAX1(CHORDO,CHORD)
IF (YSYP*LE.0.0*CR.*ALPHA*NE.0.*0.*1SYM = C
WRITE (6,42) ZS(K)
42 FORMAT (2) THOSECTION DEFINITION AT Z = F10.5/
1 = 15H0 XLE 15H YLE 15H CHORD ,
2 = 15HTHICKNESS RATIO, 15H TWIST } WRITE (6,610) XLE,YL,CHRD,THICK,AL
YM1N = YP(NL)
YM2X = YM1N
DO 44 I = 1,N
IF (YP(I) * GE. YM1N) GO TO 43
JMIN = I
YM1N = YP(I)
43 IF (YP(I) * LE. YM2X) GO TO 44
JMAX = I
YM2X = YP(I)
44 CONTINUE
YDIF = YM2X - YM1N
NN = N - 1
SUM = C.
DO 46 I = NL,NN
SUM = SUM + .5*(YP(I)+YP(I+1))*(XP(I+1)-XP(I))
46 CONTINUE
NM = NL - 1
DO 48 I = 1,NM
SUM = SUM + .5*(YP(I)+YP(I+1))*(XP(I+1)-XP(I))
48 CONTINUE
WRITE (6,300)
300 FORMAT (15H0 YMIN ,15H JMIN ,15H YMAX ,
1 15H JMAX ,15H YG1F ,15H AREA ) WRITE (6,320) YMIN,JMIN,YMAX,JMAX,YDIF,SUM
320 FORMAT (IH ,G12.4,111,G19.4,111,G19.4,G15.4)
CALL LSC
IF (FUS*LE.0.*) GO TO 61
R = AMAX1(0.,(FUS**2 - YLE(I)**2))
Z = ZS(K) - ZS1 + SCR(R)
R = FUS**2/(YLE(K)**2 + Z**2)
ZS(K) = Z*(1. - R)
YLE(K) = YLE(K)*(1. + R)
S = R*XS(NL,K)
C***SLBROU1NE COORD

SUBROUTINE COORD (NX, NY, NZ, KSYM, ZLIM, XLIM, ZLIM, 
SY, AX, AY, AZ, PX, PZ, AO, BO, 20)

C SETS UP STRETCHED PARABOLIC AND SPANWISE COORDINATES

DIMENSION AO(1), EO(1), ZO(1)

PI = 3.1415926535898
BOUND = .95
AX = .5
AY = .5
AZ = .5
XLIM = 2.25*BOUND
ZLIM = 6.25*BOUND
SY = .5
SCALZ = ZLIM/(1.000001*ZLIM)
LX = NX/2 + 1
MX = NX + 1
DX = 2.*EOUNC/NX
Q = PI/XLIM
R = PX/C

DO 12 I=1,MX
  D = (I - LX)*DX
  D = D + R*SIN(Q)*D
  IF (ABS(D).LE.*XLIM) GO TO 12
  B = 1.
  IF (D.LT.0.) B = -1.
  A = 1. - (C - B*XLIM)*E**2
  A = A**AX
  D = B*XLIM + (D - B*XLIM)/C

12 AO(I) = D
  KY = NY + 1
  DY = BOUND/NY
  DO 22 J=1,KY
    D = (KY - J)*DY
    D = 1. - D*D
    A = A**AY
    C

22 BO(J) = SY*D/C
  LZ = NZ/2 + 1
  K1 = 1
  K2 = NZ + 1
  DZ = 2.*EOUNC/NZ
  Q = PI/ZLIM
  R = PZ/C
  IF (KSYM.EQ.0) GO TO 31
  LZ = 3
  K1 = 2
  K2 = NZ + 3
  DZ = BOUND/NZ
31 RED
32 K = K1, K2
D = (K - L2)*DZ
D = D + R*SIN(Q*D)
IF (ABS(C) .LE. ZLIM) GO TO 32
B = 1.
IF (D .LT. 0.) B = -1.
A = 1. - (C - B*ZLIM)*E**2
C = A**A2
D = B*ZLIM + (D - B*ZLIM)/C
32 ZO(K) = SCALZ*C
RETURN
END
C**SLBROUTEIN SINGL*******************************************************************************
SUBROUTINE SINGL(NS,NZ,KSYM,KTE1,KTE2,FUS,CHRD0, ZS,XLE,YLE,
  1 SLEEP,DIHED,XO,Y0,ZC,YP0,ZP0,E1,E2,E3,IND)
C GENERATES SINGULAR LINE FOR SQUARE ROOT TRANSFORMATION
DIMENSION ZS(1),XLE(1),YLE(1),XO(1),YC(1),Z0(1),YPO(1),ZPO(1),
  1 E1(1),E2(1),E3(1)
K1 = 1
K2 = NZ +1
IF (KSYM*EC.0) GO TO 11
K1 = 2
K2 = NZ +3
KTE1 = 3
11 DO 12 K=K1,K2
IF (ZO(K).LT.ZS(1)) KTE1 = K +1
IF (Z0(K).LE.ZS(NS)) KTE2 = K
12 CONTINUE
CALL SLFIL (1,NS,ZS,XLE,E1,E2,E3,2,0,2,0,0,0,IND)
CALL INTPL (KTE1,KTE2,ZO,XO,1,NS,ZS,XLE,E1,E2,E3,0)
S = CHRD0*TAN(SLEEP)
S1 = CHRD0*E1(1)
S2 = CHRD0*E1(NS)
CALL SLFIL (1,NS,ZS,YLE,E1,E2,E3,2,0,2,0,0,0,IND)
CALL INTPL (KTE1,KTE2,Y0,YC,1,NS,ZS,YLE,E1,E2,E3,0)
T = CHRD0*TAN(DIHED)
T1 = CHRD0*E1(1)
T2 = CHRD0*E1(NS)
X0(KTE1-1) = XO(KTE1) +X0(KTE1) -X0(KTE1+1)
YO(KTE1-1) = Y0(KTE1) +Y0(KTE1) -Y0(KTE1+1)
IF (KSYM*EC.0) GO TO 31
N = KTE1 -1
DO 22 K=K1,N
ZZ = (ZO(K) -Z0(KTE1))/CHRD0
A = EXP(ZZ)
X0(K) = XO(KTE1) +S*ZZ -(S1 -S)*(1. -A)
YO(K) = Y0(KTE1) +T*ZZ -(T1 -T)*(1. -A)
22 N = KTE2 +1
DO 32 K=N,K2
ZZ = (ZO(K) -Z0(KTE2))/CHRD0
A = EXP(-ZZ)
X0(K) = XO(KTE2) +S*ZZ +(S2 -S)*(1. -A)
YO(K) = Y0(KTE2) +T*ZZ +(T2 -T)*(1. -A)
32 DO 42 K=K1,K2
YPO(K) = Y0(K)
ZPO(K) = Z0(K)
IF (FUS*EC.0) GO TO 42
A = .25*(ZO(K)**2 - Y0(K)**2) +FUS**2
B = .5*Z0(K)*Y0(K)
S = SQRT(A**2 +B**2)
T = 0
IF (S.GT.0.) T = 5*ATAN2(B,A)
S = SQRT(S)
YP0(K) = 5*YO(K) + S*SIN(T)
ZP0(K) = 5*Z0(K) + S*COS(T)
42 CONTINUE
RETURN
END
SUBROUTINE SURF (NC,NE,NS,NX,NZ,ISYM,KSYM,KTE1,KTE2,
1 YAW,XTE0,XLIM,FIX,NP,XS,YZ,YS,SLCPT,TRAIL, 
2 AO,XO,ZO,SO,SCAL,2V,IV,ITE1,ITE2, 
3 XP,YP,SN,D1,D2,E3,1NE)
C
INTERPLATES MAPPED WING SURFACE AT MESH POINTS
DIMENSION SO(NE,1),XS(NX,1),YS(NY,1),YZ(1),SLOPT(1),TRAIL(1),
1 AO(11),XO(11),ZO(11),SCAL(11),2V(11),
2 XP(1),YP(1),SN(1),D1(1),D2(1),D2(1),
3 IV(INE,1),NP(1),ITE1(1),ITE2(1)
PI = 3.1415926535898
TYAW = TAN(YAW)
SSO = XTE0/XLIM**2
DX = 2./NX
LX = NX/2 + 1
MX = NX + 1
MZ = NZ + 3
IV0 = 1 -ISYM -ISYM -ISYM
IV1 = -1 -ISYM
DO 2 K=1,MZ
ITE1(K) = MX
ITE2(K) = MX
DO 2 I=1,MX
IV(I,K) = -2
2 SO(I,K) = 0.
K = KTE1
K2 = 1
K1 = 1.
R2 = ZS(K2) -Z0(K2) 21,25,23
R2 = (Z0(K) -ZS(K2))/(ZS(K2) -ZS(K1))
R1 = 1. -R2
I2 = (13*NX)/16 +1
I1 = NX +2 -12
IF (FIX.EQ.C.) GO TO 31
C = R1*XS(1,K1) +R2*XS(1,K2)
CC = SQRT(C/SS0)
DO 26 I=2,NX
IF ((AC(I)) +.5*DX*LX.-CC) I1 = I +1
IF ((AC(I)) -.5*DX*LX.CC) I2 = I
26 CONTINUE
31 KK = K1
RR = R1
41 N = NP(KK)
ANGL = PI +PI
U = 1.
V = 0.
DO 42 I=1,N 
  R   =  SQRT(XS(I,KK)**2 +YS(I,KK)**2) 
 IF (R.EQ.0.) GO TO 43 
  ANGL = ANGL +ATAN2((U*YS(I,KK) -V*XS(I,KK)), 
                        (U*XS(I,KK) +V*YS(I,KK))) 
  U = XS(I,KK) 
  V = YS(I,KK) 
  R = SQRT(R +R) 
  XP(I) = R*CCS(.5*ANGL) 
  YP(I) = R*SIN(.5*ANGL) 
  GO TO 42 
43 ANGL = PI 
  U = 0. 
  V = 0. 
  XP(I) = 0. 
  YP(I) = 0. 
42 CONTINUE 
  S  = A0(I2)/AMIN1(ABS(XP(I)),ABS(XP(N))) 
  SS = .5/S**2 
DO 44 I=1,N 
  XP(I) = S*XP(I) 
  YP(I) = S*YP(I) 
  ANGL1 = ATAN(SLQPT(KK)) 
  ANGL2 = ATAN(YS(N,KK)/XS(N,KK)) 
  ANGL1 = ANGL -5*(ANGL1 -TRAIL(KK)) 
  ANGL2 = ANGL -5*(ANGL2 +TRAIL(KK)) 
  T1 = TAN(ANGL1) 
  T2 = TAN(ANGL2) 
CALL SFLIF (I,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0,IND) 
IF (INC.EQ.0) WRITE (6,500) KK,K1,K2,N,FR,R1,R2,KS(KK) 
500 FORMAT (12H0BAC MAPPING,4110,4G13,47) 
CALL INTPL (I1,I2,A0,SA,1,A,XP,YP,B1,D1,D2,D3,C) 
  X1 = .25*XS(I,KK) 
  A = SLQPT(KK)*XS(1,KK) -X1 
  B = 1./XS(1,KK) -X1 
  ANGL = PI +PI 
  U = 1. 
  V = 0. 
DO 52 I=2,M 
  XX = SS*A0(I)**2 
  D = B*(XX -X1) 
  YY = YS(1,KK) +A*ALOG(D)/D 
  R = SQRT(XX**2 +YY**2) 
  ANGL = ANGL +ATAN2((U*YY -V*XX),(U*X) +V*YY)) 
  U = XX 
  V = YY
<table>
<thead>
<tr>
<th>Code</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>53</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>54</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>55</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>56</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>57</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>58</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>59</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>60</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>61</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>62</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>63</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>64</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>65</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>66</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>67</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>68</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>69</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>70</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>71</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>72</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>73</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>74</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>75</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>76</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>77</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>78</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>79</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>80</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>81</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>82</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>83</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>84</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>85</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>86</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>87</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>88</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>89</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>90</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>91</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>92</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>93</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>94</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>95</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>96</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>97</td>
<td>R:ANGL</td>
</tr>
<tr>
<td>98</td>
<td>R:ANGL</td>
</tr>
</tbody>
</table>

Note: The code numbers are placeholders for the actual instructions or operations that are not visible in the image.
IF (KSYM.EQ.0) GC TC 81
K1  =  3
K2  = NZ +2
81 SCAL(K)  =  SCAL(KTE2)
   DO 82  I=1, MX
      ZZ  =  ZO(K) - TYAW*(X0(K) + SS*A0(I)*AO(I))
   IF (ZZ.LE.ZS(NS).AND.ZZ.GE.ZV(KTE1)) IV(I,K) = IV0
   CONTINUE
      K  =  K +1
   IF (K.LE.K2) GO TO 81
   SCAL(K)  =  SCAL(KTE2)
   N  =  KTE2
   IF (YAH.LE.0.0) GO TO 93
      IO  =  ITE1(KTE2) +1
   DO 92  I=IO,LX
      N  =  N +1
      ZV(N)  =  ZO(KTE2) - TYAW*(X0(KTE2) + SS*A0(I)*AO(I))
      ZV(KTE1-1)  =  ZO(KTE1-1) - TYAW*(X0(KTE1) + SS*A0(I)*AO(I))
      ZV(N+1)  =  ZO(KTE2+1)
   DO 102  K=K1,K2
   DO 104  I=2,NX
      IF (IV(I,K) .LT. 0) GC TC 104
      IF (IV(I+1,K+1).GT.C) OR IV(I-1,K+1).GT.C) IV(I,K) = IV1
      IF (IV(I+1,K-1).GT.C) OR IV(I-1,K-1).GT.C) IV(I,K) = IV1
   CONTINUE
   102 IF (SO(LX,K).LT.1.E-05) IV(LX,K) = 0
   IF (KSYM.NE.0) RETURN
   N  =  KTE1 -1
   DO 112  K=1,N
5 SCAL(K)  =  SCAL(KTE1)
   RETURN
END
COMMON  G(161,18,3),SO(161,35),VORI(115),ZV(115),
           IV(161,35),ITE1(35),ITE2(35),
           AO(161),BO(18),XO(35),YV(35),ZV(35),SCAL(35),
           NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
MX = NX +1
MY = NY +2
MZ = NZ +3
DO 12 I=1,161
DO 12 J=1,18
DO 12 K=1,3
12 G(I,J,K) = 0.
DO 22 K=1,MZ
WRITE (N3) ((G(I,J,1),I=MX),J=MY)
WRITE (N1) ((G(I,J,1),I=1,MX),J=MY)
22 CONTINUE
K1 = KTE1 -1
K2 = KTE2 +ITE2(KTE2) -NX/2
DO 32 K=K1,K2
32 VORI(K) = 0.
IO = 1
RETURN
END
C**SUBROUTINE MIXFLO******************************************************************************
SUBROUTINE MIXFLO
C SOLUTION OF EQUATIONS FOR MIXED SUBSONIC AND SUPERSONIC FLOW
C USING FINITE VOLUME SCHEME
C
COMMON G(161,18,3),S0(161,35),VORI(115),ZV(115),
1 IV(161,35),ITE1(35),ITE2(35),
2 A0(161),B0(18),X0(35),Y0(35),Z0(35),SCAL(35),
3 NX,NY,NZ,KTE1,KTE2,ISYM,KSYP,FUS,
4 YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
COMMON/SPA/
1 GL(161,18),QL(161,18),FL(161,18),
2 UL(161,18),VL(161,18),WL(161,18),
3 AL(161,18),BL(161,18),CL(161,18),
COMMON/FLC/ P1,P2,P3,FRES,JRES,KRES,ARES,DG,IG,JG,KG,AG,NSUP
COMMON/SLF/
K,NV,LX,MX,KY,MY,J1,K1,FMACH2,AAO,Q1,Q2,RV,TYAW,TOT
LX = NX/2 + 1
MX = NX + 1
KY = NY + 1
MY = NY + 2
J1 = 2
IF (FMACH*GE.1) J1 = 3
TYAW = SYAW/CYAW
FMACH2 = FMACH**2
AAO = 1./FMACH**2 + 2
Q1 = 2./P1
Q2 = 1./P2 - 1.
TOT = 0.
FRES = 0.
ARES = 0.
DG = 0.
AG = 0.
NSUP = 0.
K1 = 3
K2 = NZ + 2
IF (KSYP.EQ.1) GO TO 1
K1 = 2
IF (FMACH*GE.1) K1 = 3
K2 = NZ

DO 2 M=2,3
READ (N1,ERR=101) ((G(I,J,M),I=1,MX),J=1,MY)
2 CONTINUE
K = 1
NV = KTE1 - 1
RV = 2.
DO 12 I=1,MX
G(I,J,1) = G(I,J,2)
GL(I,J) = G(I,J,2)
QQL(I, J) = 0.
FL(I, J) = 0.
UL(I, J) = 0.
VL(I, J) = 0.
WL(I, J) = 0.
AL(I, J) = 0.
BL(I, J) = 0.
CL(I, J) = 0.
12 RESL(I, J) = 0.
IF (KSEMNE = 0) GO TO 21
CALL YSWEPE
RV = 1.
GO TO 51
21 DO 22 J = 1, MY
DO 22 I = 1, MX
G(I, J, 2) = G(I, J, 3)
22 GL(I, J) = G(I, J, 3)
REAL (A1, ERR=101) ((G(I, J, 3), I = 1, MX), J = 1, MY)
WRITE (A2) ((G(I, J, 1), I = 1, MX), J = 1, MY)
K = K + 1
GO TO 51
31 CALL YSWEPE
RV = 1.
IF (KSEMNE = 0) GO TO 51
IO = $TET1(K) + 1$
DO 42 I = IO, LX
M = NX + 2 -1
V = G(M, KY, 1) - G(I, KY, 1)
NV = NV + 1
42 VORT(NV) = VORT(NV) + P3*(V - VORT(NV))
51 IF (K.EQ.K2) GO TO 61
DO 52 J = 1, MY
DO 52 I = 1, MX
G(I, J, 1) = G(I, J, 2)
52 G(I, J, 2) = G(I, J, 3)
REAL (A1, ERR=101) ((G(I, J, 3), I = 1, MX), J = 1, MY)
WRITE (A2) ((G(I, J, 1), I = 1, MX), J = 1, MY)
K = K + 1
GO TO 21
61 DO 62 K = 2, 3
WRITE (A2) ((G(I, J, M), I = 1, MX), J = 1, MY)
62 CONTINUE
FRES = FRES/64.
ARES = ARES/(64.*TOT)
AG = AG/CTT
IO = 1
RETURN
101 IO = 0
SUBROUTINE YSWEEP
C
ROWE RELAXATION
C
FINITE VOLUME SCHEME
C
COMMON
O(161, 18, 3), S0(161, 35), VORT(115), ZV(115),
1
IV(161, 18, 3), ITE(135), ITE2(35),
2
AO(161), BO(18), XO(35), YC(35), ZO(35), SCAL(35),
3
NX, NY, NZ, KTE1, KTE2, ISY, KSYM, FUS,
4
COMMON/SPA/
GL(161, 18), QQL(161, 18), FL(161, 18),
1
UL(161, 18), VL(161, 18), WL(161, 18),
2
AL(161, 18), BL(161, 18), CL(161, 18),
3
COMMON/FLG/
P1, P2, P3, FRES, IRRES, IRES, IRES, KRES, ARES, LG, IG, JG, KG, AG, NSUP
COMMON/SPw/
K, NV, LX, MX, KY, MY, J, K1, FMACH2, AAO, Q1, Q2, RV, TWA, TOT
DIMENSION
X(161), Y(161), Z(161), XP(161), YM(161), ZM(161),
1
XR(161), YR(161), ZR(161), XR(161), YR(161), ZR(161),
2
QCM(161), QCP(161), QCM(161), QCP(161),
3
UM(161), UP(161), VM(161), VP(161), WM(161), WP(161),
4
AM(161), AP(161), BM(161), BP(161), CM(161), CP(161),
5
ABM(161), AEP(161), BCM(161), ECP(161),
DIMENSION
CAM(161), CAP(161), ABM(161), ABCF(161),
1
RE(161), QA(161), PA(161), QA(161), PA(161),
2
FU(161), FV(161), FW(161), FU(161), FV(161), FV(161),
3
RES(161), D(161), E(161), CG(161),

PI = 3.1415926535898
BV = .125*RV
CV = .125*(Z - RV)
D(1) = 0.
E(1) = 0.
J = 1
XS = X0(K)/SCAL(KTE1)
YS = Y0(K)/SCAL(KTE1)
ZS = Z0(K)/SCAL(KTE1)
XR S = X0(K+1)/SCAL(KTE1)
YRS = Y0(K+1)/SCAL(KTE1)
ZRS = Z0(K+1)/SCAL(KTE1)
S2 = SCAL(K)/SCAL(KTE1)
S1 = .5*S2
S2 = SCAL(K+1)/SCAL(KTE1)
S1 = .5*S2
FS = FUS/SCAL(KTE1)

DO 12 I=1, MX
RES0(I) = 1.
QA(I) = 0.
P(I) = 0.
Q(I) = 0.
R(I) = 0.
QP(I) = 0.
FU(I) = 0.
FW(I) = 0.
CG(I) = 0.
XM(I) = XS + S1*(AO(I)**2 - (BO(J) + SO(I,K))**2)
YM(I) = YS + S2*AO(I)*(BO(J) + SO(I,K))
ZM(I) = ZS
XR(I) = XRS + S1*(AO(I)**2 - (BO(J) + SO(I,K+1))**2)
YR(I) = YRS + S2*AO(I)*(BO(J) + SO(I,K+1))
ZR(I) = ZRS

IF (FUS .LE. C.) GO TO 21
DO 14 I = 1, MX
A = .25*(ZM(I)**2 - YM(I)**2) + FS**2
B = .5*ZM(I)*YM(I)
S = SQRT(A**2 + B**2)
T = 0.
IF (S .GT. 0.) T = .5*ATAN2(B,A)
IF (B .EQ. 0.) AND.YM(I) .GT. (FS + FS)) T = .5*PI
S = SQRT(S)
YM(I) = .5*YM(I) + S*SIN(T)
ZM(I) = .5*ZM(I) + S*COS(T)
A = .25*(ZM(I)**2 - YM(I)**2) + FS**2
B = .5*ZM(I)*YM(I)
T = 0.
S = SQRT(A**2 + B**2)
IF (S .GT. 0.) T = .5*ATAN2(B,A)
S = SQRT(S)
YR(I) = .5*YM(I) + S*SIN(T)
ZR(I) = .5*ZM(I) + S*COS(T)

DO 22 I = 1, MX
X(I) = XM(I)
Y(I) = YM(I)
Z(I) = ZM(I)
XR(I) = XRS
YR(I) = YRS
ZR(I) = ZRS
XM(I) = XS + S1*(AO(I)**2 - (BO(J+1) + SO(I,K))**2)
YM(I) = YS + S2*AO(I)*(BO(J+1) + SO(I,K))
ZM(I) = ZS
XR(I) = XRS + S1*(AO(I)**2 - (BO(J+1) + SO(I,K+1))**2)
YR(I) = YRS + S2*AO(I)*(BO(J+1) + SO(I,K+1))
ZR(I) = ZRS

IF (FUS .LE. C.) GO TO 31
DO 24 I = 1, MX
A = .25*(ZM(I)**2 - YM(I)**2) + FS**2
\[ B = 5 \times 2M(I) \times YM(I) \]
\[ S = \text{SORT}(A**2 + B**2) \]
\[ T = 0 \]
\[ \text{IF} (B < 1.0 \times T) \quad T = 0.5 \times \text{ATAN2}(B, A) \]
\[ \text{IF} (B < 1.0 \times YM(I) \times GT((FS + FS) + FS)) \quad T = 0.5 \times R \]
\[ \text{IF} (B < 1.0 \times \text{AND}(YM(I), GT((FS + FS))) \quad T = 0.5 \times P \]
\[ S = \text{SORT}(S) \]
\[ YM(I) = 0.5 \times YM(I) + 0.5 \times \text{SIN}(T) \]
\[ ZM(I) = 0.5 \times ZM(I) + 0.5 \times \text{COS}(T) \]
\[ A = 0.25 \times (ZRM(I))**2 - YRM(I)**2 + FS**2 \]
\[ B = 0.5 \times ZRM(I)**2 \times YRM(I) \]
\[ T = 0 \]
\[ S = \text{SORT}(A**2 + B**2) \]
\[ \text{IF} (S < 1.0 \times T) \quad T = 0.5 \times \text{ATAN2}(B, A) \]
\[ S = \text{SORT}(S) \]
\[ ZRM(I) = 0.5 \times ZRM(I) + 0.5 \times \text{SIN}(T) \]
\[ ZRM(I) = 0.5 \times ZRM(I) + 0.5 \times \text{COS}(I) \]
\[ \text{DO } 32 \text{ I=1,NX} \]
\[ XX = XR(I+1) - XR(I) + XRM(I+1) - XRM(I) \]
\[ XY = X(I+1) + X(I) + XM(I+1) - XM(I) \]
\[ XZ = XR(I+1) + XR(I) + XRM(I+1) - XRM(I) \]
\[ YX = YR(I+1) - YR(I) + YRM(I+1) - YRM(I) \]
\[ YY = YR(I+1) + YR(I) + YRM(I+1) - YRM(I) \]
\[ YZ = YR(I+1) + YR(I) + YRM(I+1) - YRM(I) \]
\[ ZX = ZR(I+1) - ZR(I) + ZRM(I+1) - ZRM(I) \]
\[ ZY = ZR(I+1) + ZR(I) + ZRM(I+1) - ZRM(I) \]
\[ ZZ = ZR(I+1) + ZR(I) + ZRM(I+1) + ZRM(I) \]
\[ FXX = YY**ZZ - YZ**ZY \]
\[ FYX = YZ**ZX - YX**ZZ \]
\[ FZX = YX**ZY - YY**ZX \]
\[ FXY = ZY**XZ - ZZ**XY \]
\[ FYZ = ZZ**XX - ZX**XX \]
\[ FYZ = ZX**XY - ZY**XX \]
\[ FXX = XY**ZV - XZ**YV \]
\[ FYZ = XZ**XY - XX**YZ \]
\[ FZZ = XX**YY - XY**YY \]
\[ FM(I) = FXX**XY + FYX**XY + FXY**XY \]
\[ A = 1.0 / FM(I) \]
\[ GX = G(I+1, J, 2) - G(I, J, 2) + G(I+1, J+1, 2) - G(I, J+1, 2) \]
U = BV*(UR +UL(I,J))
V = BV*(VR +VL(I,J))
W = CV*(WR +WL(I,J))
Q(I,J) = QQ*(Q(I,J))
F(I,J) = FR
UL(I,J) = UR
VL(I,J) = VR
WL(I,J) = WR
AA = AA - .2*CC
QA(I) = QQ/AA
A = 16*RESO(I)/(F*AA)
QP(I) = Q(I)
P(I) = O.
G(I) = O.
R(I) = O.
F = A*AMAX1(ABS(U),ABS(V),ABS(W))
FU(I) = F*U
FV(I) = F*ABS(V)
FW(I) = F*ABS(W)
IF (QA(I).LE.1.) GC TC 42
NSUP = NSUP +1
F = A*(1. - 1./QA(I))
FUL(I) = F*U*U
FVV(I) = F*V*V
FWW(I) = F*W*W
GXX = G(I+1,J,2) - G(I,J,2) - G(I-1,J,2) + G(I-1,J+1,2)
GYY = G(I,J-1,2) - G(I,J,2) - G(I-1,J-1,2) + G(I,J+1,2)
GZZ = G(I,J,3) - G(I,J,2) - G(I,J,2) + G(I,J+1,2)
GXY = G(I+1,J-1,2) - G(I-1,J-1,2) + G(I,J,J-1,2)
1
GYZ = G(I,J-1,3) - G(I,J+1,3) - G(I,J-1,3) + G(I,J,J+1,3)
GZX = G(I+1,J,3) - G(I-1,J,3) - G(I+1,J,3) + G(I-1,J+1,3)
F = .25*F
FXY = F*U*V*GXY
FYZ = F*V*W*GYZ
FZX = F*W*U*GZX
P(I) = FUU(I)*GXX +FXY +FZX
Q(I) = FVV(I)*GYY +FYZ +FXY
R(I) = FWI(I)*GZZ +FZX +FYZ

42 CONTINUE
PF = -P(2)
DO 52 I=2,NX
AV = RESC(I)*RV
PB = PF
PP = P(I)
IF ((FU(I) +FU(I+1)).LT.0.) PF = -P(I+1)
A = UP(I) -UP(I-1) +UM(I) -UM(I-1)
1 +UP(I) +UP(I-1) -VM(I) -VM(I-1)
\[
\begin{align*}
2 & \quad -ABP(I) + ABP(I-1) + ABM(I) - ABM(I-1) \\
B & \quad WP(I) + WP(I-1) + WM(I) + WM(I-1) \\
I & \quad -BCP(I) + BCP(I-1) + BCPM(I) + BCPM(I-1) \\
3 & \quad -ACP(I) + ACP(I-1) - CAM(I) + CAM(I-1) \\
RES(I) & \quad AV*(A + B + P + Q + QP(I) - Q(I)) + RESL(I, J) \\
RESL(I, J) & \quad A - B \\
AR & \quad AP(I) + AP(I-1) + AM(I) + AM(I-1) \\
BR & \quad BP(I) + BP(I-1) + BM(I) + BM(I-1) \\
CR & \quad CP(I) + CP(I-1) + CM(I) + CM(I-1) \\
A & \quad AV*AR + AL(I, J) \\
B & \quad AV*BR + BL(I, J) \\
C & \quad AV*CR + CL(I, J) \\
AL(I, J) & \quad AR \\
BL(I, J) & \quad BR \\
CL(I, J) & \quad CR \\
A & \quad A + A \\
B & \quad B + B \\
C & \quad C + C \\
TP & \quad A + A + Q1*(B + C) + Q2*(ABS(FU(I)) + FV(I) + Fw(I)) \\
TS & \quad (B + Q2*FV(I)*CG(I) + C + Q2*Fw(I))*(G(I, J, 1) - GL(I, J)) \\
F & \quad 0. \\
IF (QA(I) LE I) GC TG 53 \\
F & \quad FUU(I) \\
TP & \quad TP + F \\
T & \quad TP + T + 3.* (F + FV(I) + Fw(I)) + F \\
S & \quad S + 3.*(FV(I)*CG(I) + Fw(I)) + (G(I, J, 1) - GL(I, J)) \\
53 & \quad TP + T \\
IF (FU(I) GE 0) TM = TM + Q2*FU(I) + F + F \\
IF (FU(I) LT 0) TP = TP - Q2*FU(I) + F + F \\
TM & \quad TM*RES0(I-1) \\
B & \quad 1/(I - TM*E(I-1)) \\
D & \quad B*(RES(I) + S + TM*L(I-1)) \\
52 & \quad B*TP \\
IF (J, LT, J1 OR, K, LT, K1) GO TO 71 \\
I & \quad 12 \\
CG(I+1) & \quad 0. \\
DO 62 h = 2, 12 \\
GL(I, J) & \quad G(I, J, 2) \\
CG(I) & \quad DI(I) + E(I)*CG(I+1) \\
G(I+J, 2) & \quad G(I, J, 2) + CG(I) \\
TO1 & \quad TOT + 1 \\
ARES & \quad ARES + ABS(RES(I)) \\
IF (ABS(RES(I)) LE, ABS(FRES)) GO TO 63 \\
FRES & \quad RES(I) \\
IRES & \quad 1
\end{align*}
\]
GO CM

63 AG = AG + AES(CG(I))
IF (ABS(CG(I)) .LE. ABS(DG)) GO TO 62
DG = CG(I)
IG = I
JG = J
KG = K

62 I = 1 - 1
70 DO 72 I = 1, NX
QQP(I) = QQM(I)
DP(I) = DM(I)
FP(I) = FM(I)
UP(I) = UM(I)
VP(I) = VM(I)
WP(I) = WM(I)
AP(I) = AM(I)
BP(I) = BM(I)
CP(I) = CM(I)
ABP(I) = ABM(I)
BCP(I) = BCM(I)
CAP(I) = CAM(I)
ABCP(I) = ABCM(I)

72 J = J + 1
IF (J - KY) 21, 81, 101
81 I2 = ITEZ(K)
IF (ITEZ(K) .EQ. NX) I2 = LX
IF (ISYM .EQ. 1) I2 = NX
DO 82 I = 1, NX
IF (ISYM .EQ. 1) GO TO 83
IF (IV(I,K) .EQ. 2 .OR. IV(I+1,K) .EQ. 2) GO TO 83
M = NX + 1 - 1
QQM(I) = QQP(M)
DM(I) = DP(M)
FM(I) = FP(M)
UM(I) = -UP(M)
VM(I) = -VP(M)
WM(I) = WP(M)
AM(I) = AP(M)
BM(I) = BP(M)
CM(I) = CP(M)
ABM(I) = ABP(M)
BCM(I) = -BCP(M)
CAM(I) = -CAP(M)
ABCP(I) = ABCP(M)
GO TO 62
83 QQP(I) = QQP(I)
CM(I) = DP(I)
FM(I) = FP(I)
UM(I) = UP(I)
VM(I) = -VP(I)
WM(I) = AP(I)
BM(I) = BP(I)
CM(I) = CP(I)
ABM(I) = -ABP(I)
BCM(I) = -BCF(I)
CAM(I) = -CAP(I)
ABC(M) = -ABC(I)

82 CONTINUE
DO 92 I = 2, NX
IF (IAES(I*IV(I,K)) .GT. 1) GO TO 92
RESO(I) = 0.
A = 1. -AMAXO(O, IABS(I*IV(I,K)))
S = A*(G(I+1, J, 2) - G(I, J, 2) - G(I, J, 2) - G(I, J, 2) + G(I-1, J, 2))
RES(I, J) = S + S
AL(I, J) = A
CL(I, J) = 0.
92 CONTINUE
GO TO 41

101 S1 = .5*SCAL(K)
I1 = NX + 2 - 12
I1 = ITE1(K)
N1 = NV
IF (I .NE. I1 .OR. ISYM .EQ. 1) GO TO 103
V = G(12, KY, 2) - G(I1, KY, 2)
NV = NV + 1
VORT(NV) = VORT(NV) + P3*(V - VORT(NV))
N = NV
103 I = 1 - 1
V = 0.
IF (IV(I, K) .NE. 1) GO TC 109
ZZ = ZO(K) - TIAK*(XO(K) + S1*A0(1)*A0(1))
105 IF (ZZ .GE. ZV(N-1)) GO TO 107
N = N - 1
GO TO 105
107 A = (ZZ - 2*ZV(N-1))/(ZV(N) - 2*ZV(N-1))
V = A*VGR(N) + (1. - A)*VCF(N-1)
109 M = NX + 2 - 1
G(I, KY+1, 2) = G(M, KY-1, 2) - V
G(I, KY+1, 2) = G(I, KY-1, 2) + V
G(M, KY+1, 2) = G(I, KY, 2) + V
IF (I .EQ. 1) GO TO 103
G(I, KY, 2) = -.5*V
SUBROUTINE VELC (K, SV, SM, CP, XP, YP, XMAX, XMIN, YMAX, YMIN)
C CALCULATES SURFACE VELOCITY
COMMON G(161, 18, 3), SO(161, 35), VORT(115), ZV(115),
1 IV(161, 35), ITE1(35), ITE2(35),
2 A0(161), BO(181), XO(35), VC(35), ZC(35), SCAL(35),
3 NX, NY, N2, KTE1, KTE2, ISYP, KSYM, FUS,
4 YAW, CYAN, SYAW, ALPHA, CA, SA, FMACH, N1, N2, N3, IG
COMMON/SPA/
XL(161, 18), YL(161, 18), ZL(161, 18),
1 X(161, 1E), Y(161, 1E), Z(161, 1E),
2 XR(161, 18), YR(161, 18), ZR(161, 18),
3 RESL(161, 18)
COMMON /UVW/ UU(161), VV(161), WW(161)
DIMENSION SV(1), SM(1), CP(1), XP(1), YP(1)
PI = 3.1415926535898
PI1 = PI / (1.7 * FMACH**2)
MX = 1
NMAX = 1
AV = .5
IF (KSYM.EQ.1. AND. K.EQ.KTE1) GO TO 1
NMAX = 2
AV = .25
K = K
IF (K.EQ.KTE1) GO TO 11
N = KTE1 - 2 + KSYM
DO 2 J = 1, 2
DO 2 I = 1, MX
XL(I, J) = 0.
Y(I, J) = 0.
Z(I, J) = 0.
XR(I, J) = 0.
YR(I, J) = 0.
ZR(I, J) = 0.
2 N = N + 1
XR = XO(N) / SCAL(KTE1)
YR = YO(N) / SCAL(KTE1)
ZR = ZO(N) / SCAL(KTE1)
SZ = SCAL(N) / SCAL(KTE1)
S = .5 * S2
FUS = FUS / SCAL(KTE1)
DO 12 J = 1, 2
M = NY + 2 - J
DO 12 I = 1, MX
XL(I, J) = X(I, J)
Y(I, J) = Y(I, J)
Z(I, J) = Z(I, J)
X(I, J) = XR(I, J)
U = U + (F*Z*G) + (F*Y*G) + (F*Z*G)*F + G
V = V + (F*Z*G) + (F*Y*G) + (F*Z*G)*F + G
W = W + (F*Z*G) + (F*Y*G) + (F*Z*G)*F + G

IF (M.E.Q.2) GO TO 29
M = 2
XX = X(I,1) - X(I-1,1)
XY = Y(I,1) - Y(I-1,1)
XZ = Z(I,1) - Z(I-1,1)
GX = G(I,J,2) - G(I-1,J,2)
GO TO 23

25 IF (N.E.Q.NMAX) GO TO 27
M = 1
N = 2
XX = X(I+1,1) - X(I,1)
XY = Y(I+1,1) - Y(I,1)
XZ = Z(I+1,1) - Z(I,1)
GZ = G(I,J,2) - G(I,J,1)
GO TO 23

27 UU(I) = AV^L
VW(I) = AV^W
UU(I) = U + U + V + V + W + W
SV(1) = SQRT(Q)
Q = A * max(0.1 * (1.0 + Q1 + (1.0 - Q)))
SM(I) = FMACH * SV(I) / SQRT(Q)
CP(I) = T1 * (Q * Q * Q * Q * Q - 1.0)
XP(I) = SCAL(KTE1)^x(I,I)
YP(I) = SCAL(KTE1)^y(I,I)

22 II = ITE1(K)
I2 = ITE2(K)
XMAX = SCAL(KTE1)^x(I1,1)
XMIN = XMAX
YMAX = SCAL(KTE1)^y(I1,1)
YMIN = YMAX
DO 22 I1=I1,12
XMAX = AMAX1(XMAX,XP(I))
XMIN = AMIN1(XMIN,XP(I))
YMAX = AMAX1(YMAX,YP(I))
YMIN = AMIN1(YMIN,YP(I))

32 RETURN
END
C**SUBROUTINE CPLCT*****************************************************************
SUBROUTINE CPLCT (I3, I4, FMACH, XP, YP, CP, SM, I1, I2, KPLT)
C
PLCTS CF AT COMPUTATIONAL MESH POINTS
COMMON /PCKR/ PCTK
COMMON /PARM3/ XT3(161), YT3(161), ZT3(161),
1     UT3(161), VT3(161), WT3(161), NOI, Z
COMMON /UVW/ UU(161), VV(161), WW(161)
COMMON /PRES/ XOC0(161)
COMMON /SFAR/ LINE(90), DUMY(10)
DIMENSION KODE(1), XPI(1), YP(1), CP(1), SM(1)
DATA KODE/1H , 1H*/
DATA IST/1H*/
NOI = C
IMIN = I1 + (12 - I1)/2
CFC = XF(I1) - XP(IMIN)
2 FORMAT(4H0)PLOT CF CP AT COMPUTATIONAL MESH POINTS/
1     1CHO X , 1CH Y , 7HMACH N0, 8H CP ,
2    7H XCC , 2X, 5FCP* = F8, 4, 2X, 7HCP-CPD = F1C, 4)
CPC = 4F1 (1. + 2*FMACH**2)**3.5 - 1.0/(.7*FMACH**2)
DO 12 I=1, 90
12 LINE(1) = KODE(1)
CPS = (((5. + FMACH**21)/6.1)**3.5 - 1.0)/(.7*FMACH**2)
IF (KPLCT.EQ.0 .CR.KPLT.GT.1) GO TO 15
WRITE (6,2) CPS, CFC
15 CONTINUE
KS = 3C* (CP0 - CPS) + 4.5
IF (KS.LE.1. AND. KS.LE.90) LINE(KS)=IST
DO 22 I=13, I4
K = 30* (CP0 - CP(I)) + 4.5
K = MINC(9C, K)
LINE(K) = KODE(2)
XOC = (XP(I) - XPI(IMIN))/CHC
XOCD(I) = XCC
IF (KPLCT.EQ.0 .CR.KPLT.GT.1) GO TO 20
WRITE (6,16I) XP(I), YP(I), SM(I), CP(I), XCC, LINE
20 CONTINUE
LINE(K) = KODE(1)
IF (I.I.T.I1 .OR. I.GT.12) GO TO 22
NOI = ACI + 1
XT3(NOI) = XP(I)
YT3(NOI) = YP(I)
ZT3(NOI) = Z
UT3(NOI) = U(I)
VT3(NOI) = W(I)
WT3(NOI) = W(I)
22 IF (K.EQ.KS) LINE(KS) = IST
IF (KPLCT.EQ.0 .CR.KPLT.GT.1) GO TO 25
CALL IN\RT6
SUBROUTINE INVRT6

COMMON /FCKR/ FTCK
COMMON /PARMT3/ XT3(161), YT3(161), ZT3(161),
1 UT3(161), WT3(161), N01, Z

DIMENSION P(161,6), TEMP(161)

EQUIVALENCE (XT3(I),P(I,1))

DO 10 I=1,NCI
M=N01-I+1
TEMP(M)=P(I,J)
10 CONTINUE

DO 20 J=1,NCI
P(I,J)=TEMP(I)
20 CONTINUE

WRITE (6,100) (I,XT3(I),ZT3(I),YT3(I),UT3(I),
1 WT3(I),VT3(I),I=1,N01)

RETURN

10C FORMAT(4H0 1,1X,6HXT3(I),4X,6HZT3(I),4X,6HYT3(I),4X,6HUT3(I),
1 4X,6HWT3(I),4X,6HW3T(I),
2 6X,4H 1,1X,6HXT3(I),4X,6HZT3(I),4X,6HYT3(I),4X,6HUT3(I),
3 4X,6HWT3(I),4X,6HW3T(I)/
4 (I4,1X,6G10.3,2X,I4,1X,6G10.3))
SUBROUTINE FORCF

CALCULATES SECTION FORCE COEFFICIENTS

DIMENSION XP(I), YP(I), CP(I)

RAD = 57.295779513082

ALPHA = AL/RAD

CL = 0.

CD = 0.

CM = 0.

N = 12 - 1

DO 12 I=11, N

DX = (XP(I+1) - XP(I))/CHORD

DY = (YP(I+1) - YP(I))/CHORD

XA = (.5*(XF(I+1) + XP(I)) - XM)/CHORD

YA = (.5*(YP(I+1) + YP(I)) - YM)/CHORD

CPA = .5*(CP(I+1) + CP(I))

DCL = CPA*DX

CL = CL + DCL

CD = CD + DCL

CM = CM + DCD*YA - DCL*XA

CCL = CL*COS(ALPHA) - CD*SIN(ALPHA)

CD = CL*SIN(ALPHA) + CD*COS(ALPHA)

CL = DCL

RETURN
END
C**SUBROUTINE TCFOR**************************************************************************
SUBROUTINE TCFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,X0,YPO,ZPO,
1   CL,CD,CMP,CMR,CMY)
C CALCULATES TOTAL FORCE COEFFICIENTS
DIMENSION CHORD(K),SCL(K),SCD(K),SCM(K),X0(K),YPO(K),ZPO(K)
SPAN   = ZPO(KTE2) - ZPO(KTE1)
CL     = 0.
CD     = 0.
CMP    = 0.
CMR    = 0.
CMY    = 0.
S      = 0.
K      = KTE1
GL     = SCL(K) * CHORD(K)
OD     = SCD(K) * CHORD(K)
QM     = CHCFD(K) * (SCM(K) * CHORD(K)
   - SCL(K) * X0(K) + SCD(K) * YPO(K))
11  DZ  = .5 * (ZPO(K+1) - ZPO(K))
AZ     = .5 * (ZPO(K+1) + ZPO(K))
PL     = SCL(K+1) * CHORD(K+1)
PD     = SCD(K+1) * CHORD(K+1)
PM     = CHCFD(K+1) * (SCM(K+1) * CHORD(K+1)
   - SCL(K+1) * X0(K+1) + SCD(K+1) * YPO(K+1))
1  CLA  = DZ * (PL + CL)
CDLA  = DZ * (PD + OD)
CL    = CL + CLA
CD    = CD + CDLA
CMP   = CMP + DZ * (PM + QM)
CMR   = CMR + AZ * CLA
CMY   = CMY + AZ * CDLA
S     = S + DZ * (CHORD(K+1) + CHORD(K))
11 CL  = CL / S
CD    = CD / S
CMP   = CMP * SPAN / S**2
CMR   = (CMR + CR)/ (S*SPAN)
CMY   = (CMY + CMY)/ (S*SPAN)
RETURN
END
C**SLBROTLINE REFIN**

SUBROUTINE REFIN

HALVES MESH

COMMON

G(161,18,3),So(161,35),VORT(115),ZV(115),
IV(161,35),ITE1(35),ITE2(35),
1
AO(161),BO(18),XO(35),YO(35),ZO(35),SCAL(35),
NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
4
YAW,ZYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO

MX = NX + 1
KY = NY + 1
MY = NY + 2
MZ = NZ + 3
MXO = NX/2 + 1
MYC = NY/2 + 2
MZO = NZ/2 + 1
K = 1

IF (KSYM.EQ.0) GO TO 11
MZO = NZ/2 + 3
REAC (N1,ERR=401) (G(I,J,1),I=1,MXO),J=1,MYO)

K = 2
11 REAC (N1,ERR=401) (G(I,J,1),I=1,MXO),J=1,MYO)

JJ = KY

21 JJ = MXO

31 G(I1,J1,1) = G(I,J,1)

II = JJ - 1

IF (I.GT.0) GO TO 31

J = J - 1

JJ = JJ - 2

IF (J.GT.0) GO TO 21

DO 42 J=1,KY,2
   DO 42 J=1,MX,2

42 G(I,J,1) = 5*(G(I+1,J,1) + G(I-1,J,1))

DO 52 I=1,MX,2
   DO 52 I=1,MX,2

52 G(I,J,1) = 5*(G(I,J+1,1) + G(I,J-1,1))

GO TO 52

54 G(I1,MY,1) = 0.
G(I1,NY,1) = 0.
G(MX,NY,1) = 0.
WRITE (N2) ((G(I,J,1),I=1,MX),J=1,MY)

K = K + 1

IF (K.LE.MZO) GO TO 11

REIND N2
READ (N2,ERR=401) ((G(I,J,1),I=1,MX),J=1,MY)
READ (N2,ERR=401) ((G(I,J,3),I=1,MX),J=1,MY)
WRITE (N1) ((G(I,J,1), I=1,MX), J=1,MY)
K  =  1
IF (KSYK. NE.0) K = 2
111  K   =  K  + 1
DO 112   I=1,MX
DO 112   J=1,MX
112  G(I,J,2) = 5*(G(I,J,1) +G(I,J,3))
DO 112   K=2,3
WRITE (N1) ((G(I,J,K), I=1,MX), J=1,MY)
122  CONTINUE
IF (K. EQ. MZC) GC TC 201
DO 132   J=I,MX
132  G(I,J,1) = G(I,J,3)
READ (N2, ERR=401) ((G(I,J,3), I=1,MX), J=1,MY)
GO TO 111
201  REWIND N1
REWIND N2
DO 202   K=1,3
READ (N1, ERR=401) ((G(I,J,K), I=1,MX), J=1,MY)
202  CONTINUE
WRITE (N2) ((G(I,J,1), I=1,MX), J=1,MY)
TYAW  = SYAW/CYAW
NV   = KTE1 -1
VORT(NV) = 0.
K  =  2
IF (KSYK. NE.0) GO TO TC 251
111  S1  =  5*SCAL(K)
N   =  NV
I   =  MX0 +1
IF (K.LT.KTE1 OR K.GT.KTE2) GO TO 231
111  =  ITE1(K)
I2  =  ITE2(K)
DO 212   I=I1,I2
212  G(I,KY+1,2) = G(I,KY,2) +G(I,KY,2) -G(I,KY-1,2)
NV   = NV +1
VORT(NV) = G(I2,KY,2) -G(I1,KY,2)
N   =  NV
I   =  I1
IF (K.EQ.KTE2. OR .YAW.LE.0. ) GO TO 231
221  I   =  I  +1
M   =  NX +2 -1
NV   =  NV +1
VORT(NV) = G(M,KY,2) -G(I,KY,2)
IF (I.LT.MX0) GC TO 221
231  I   =  I  -1
V   =  0.
IF (IV(I,K) .NE. 1) GO TO 237

233 IF (ZZ .GE. ZV(N-1)) GO TO 235

235 A = (ZZ - ZV(N-1))/(ZV(N) - ZV(N-1))

237 M = NX + 2 - I
G(I,KY-1,1) = G(M, KY-1,2) - V
G(M, KY+1,2) = G(I,KY-1,2) + V
IF (IV(I,K) .NE. -1) GO TO 241
G(I,KY+1,3) = 5*G(I,KY,1) + .25*(G(I,KY,3) + G(M, KY, 3))
IF (IV(I,K+1) .LT. 1)
G(I,KY-1,3) = 5*G(I,KY,3) + .25*(G(I,KY,1) + G(M, KY, 1))
G(M, KY-1,3) = 5*G(M, KY, 3) + .25*(G(I,KY-1,2) + G(M, KY-2,2))

241 IF (I .GT. 1) GO TO 231
G(I,KY,2) = -5*V
G(M, KY, 2) = 5*V

251 K = K + 1
IF (K .LE. MZ) GO TO 261
DO 252 I = 1, KY
DO 252 I = 1, MX
G(I,J,1) = G(I,J,2)
G(I,J,2) = G(I,J,3)
WRITE (N2) ((G(I,J,1),I=1, MX), J=1, MY)
READ (N1), ERR=401 ! !(G(I,J,3),I=1, MX), J=1, MY
GO TO 251

261 VORT(N+1) = 0.
DO 262 K = 2, Z
WRITE (N2) ((G(I,J,K), I=1, MX), J=1, MY)
CONTINUE
RE WIND A1
READ A2
DO 302 K = 1, MZ
READ A2, ERR=401 ! !(G(I,J, K), I=1, MX), J=1, MY
WRITE (N1) ! !(G(I,J, K), I=1, MX), J=1, MY
CONTINUE
IO = 1
RETURN
IO = 0
RETURN
ENC
C**SUBROUTINE SMOC***********************************
SUBROUTINE SMOC
C
SMOOTHING POTENTIAL
C
COMMON
G(161,18,3), SQ(161,35), WOR (115), ZV(115),
1 IV(161,35), ITE1(35), ITE2(35),
2 AO(161), BO(18), XO(35), YC(35), Zo(35), SCAL(35),
3 NX,NY,NZ,KTE1,KTE2,ISYM,SY, FUS,
4 YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
MX = NX + 1
KY = NY + 1
MY = NY + 2
MZ = NZ + 3
K1 = 2
K2 = NZ + 2
IF (KSYM.EQ.G) GO TO 1
K1 = 3
K2 = NZ + 2
1 PX = 1./6.*
PY = 1./6.*
PZ = 1./6.*
DO 2 L=1,3
READ (K), ERR=51) ((G(I,J,L),I=1,MX),J=1,MY)
2 CONTINUE
WRITE (A2) ((G(I,J,1),I=1,MX),J=1,MY)
K = K1
11 K = K + 1
DO 12 J=3,NY
12 DO 14 I=1,NX
14 G(I,J,1) = (1. - FX - PY - PZ)*G(I,J,2)
1 + .5*PX*(G(I+1,J,2) + G(I-1,J,2))
2 + .5*PY*(G(I,J+1,2) + G(I,J-1,2))
3 + .5*PZ*(G(I,J,3) + G(I,J,1))
G(I,J,1) = G(I,J,2)
12 G(MX,J,1) = G(MX,J,2)
DO 16 I=1,MX
16 G(I,1,1) = G(I,1,2)
G(I,2,1) = G(I,2,2)
G(I,KY,1) = G(I,KY,2)
16 G(I,MY,1) = G(I,MY,2)
WRITE (A2) ((G(I,J,1),I=1,MX),J=1,MY)
IF (K.EQ.K2) GO TO 31
DO 22 J=1,MY
22 DO 22 I=1,MX
G(I,J,1) = G(I,J,2)
22 G(I,J,2) = G(I,J,3)
READ (K), ERR=51) ((G(I,J,3),I=1,MX),J=1,MY)
GO TO 11
31 WRITE (A2) ((G(I,J,3),I=1,MX),J=1,MY)
RE WIND N1
RE WIND N2
DO 42 K=1,MZ
READ (N2, ERR=51) ((G(I,J,1), I=1,MX), J=1,MY)
WRITE (N1) ((G(I,J,1), I=1,MX), J=1,MY)
42 CONTINUE
IO RETURN = 1
51 RETURN = 0
ENC
C**SUBROUTINE SPLIF***********************************************************
SUBROUTINE SPLIF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN, VN, MODE, FQM, IND)
C
C SPLINE FIT - JAMESON
C INTEGRAL PLACED IN FPPP IF MODE GREATER THAN 0
C IND SET TO ZERO IF DATA ILLEGAL
C
DIMENSION S(1),F(1),FP(1),FPP(1),FPPP(1)

INC = 0
K = IABS(N - M)
IF (K - 1) E1, E1, 1
1 K = (N - M)/K
J = M + K
DS = S(J) - S(I)
D = DS
IF (DS) 11, 81, 11
11 DF = (F(J) - F(I))/CS
IF (KM - 2) 12, 13, 14
12 U V W GOTO 25
13 U V W = 0.
14 U V W GOTO 25
15 U V W = -1.
21 I J = J + K
DS = S(J) - S(I)
IF (D*DS) 81, 81, 23
23 DF = (F(J) - F(I))/DS
B = 1./(DS + DS + U)
B = DS
V = B*DF - V
25 FP(I) = U
FPP(I) = V
U = (2. - U)*DS
V = 6.*DF + DS*V
IF (J - N) 21, 31, 21
31 IF (KN - 2) 32, 33, 34
32 V W GOTO 35
33 V W = VN
34 V W GOTO 35
35 V W = (DS*VN + FPP(I))/(1. + FP(I))
36 B D = V
41 CS = S(J) - S(I)
L = FPP(I) - FP(I)*V
FPP(I) = (V -U)/DS
FPF(I) = U
FP(I) = (F(J) -F(I))/DS -DS*(V +L +U)/6.
V = U
J = I
IF (J -K) 41, 51, 41
51 I = N -K
FPP(N) = FPPF(I)
FPP(N) = B
FP(N) = DF +D*(FPP(I) +B +B)/6.
INC = 1
IF (MOCE) 81, 81, 61
61 FPFP(J) = FQM
V = FPP(J)
71 I = J
J = J +K
DS = S(J) -S(I)
U = FPP(J)
FPPP(J) = FPPP(I) +.5*DS*(F(I) +F(J) -CS*DS*(U +V)/12.
V = U
IF (J -K) 71, 81, 71
81 IF (INC.EQ.1) GC TO 90
WRITE (6, 85) INC, MOCE, I, J, K, M, S(I), S(J), DS, C
85 FORMAT (6HOCHECK, 6110, 4G13.4)
WRITE (6, 86) (S(I), F(I), I=M, N)
86 FORMAT (10G13.4)
90 RETURN
END
SUBROUTINE INTPL(PI,N1,S1,F1,M,N,S,F,FP,FPPP,MODE)

INTERPOLATION USING TAYLOR SERIES - JAESC

ADCS CORRECTION FOR PIECEWISE CONSTANT FOURTH DERIVATIVE

IF MODE GREATER THAN 0
D I M E N S I O N S l(N), F I (N), 3 (1), 3 (I), 3 (1), FP (1), F P (1), F P P (1)

K = I A B S ( N - M )
K = ( N - M ) / K
I = M
M I N = M I
N I N = N I
D = S ( N ) - S ( M )
I F ( D * ( S 1 ( M I ) - S I ( M I ) ) ) 11, 13, 13

11 M I N = N I
N I N = M I
13 K I = I A B S ( N I N - M I N )
I F ( K I ) 21, 21, 1 5
15 K I = ( N I N - M I N ) / K I
21 I I = M I N - K I
C I F ( M O D E ) 31, 31, 23
23 C I I = 1 I
3 I S S = S I ( I I )
33 I I = I + K I
I F ( I - N ) 25, 3 7, 3 5
35 I F ( D * ( S 1 ) - S S ) 3 3, 3 3, 3 7
37 J = I
I = I - K
S S = S S - S 1
F P P P = C * ( F P P P ( I ) - F P P P ( I ) ) / S ( J ) - S ( I )
F F = F P P ( I ) + 2 * S S * F P P P
F F = F P P ( I ) + S S * F F / 3.
F F = F P P ( I ) + 0. 5 * S S * F F
F I ( I I ) = F ( I I ) + S S * F F
I F ( I I ) - N I N 3 1, 4 1, 3 1
41 R E T U R N
E N D
SUBROUTINE VERTEC(II,12,XCCD,CP,NRD,ZPC,FMAC,YA,AL,
  SCL,SCD,SCM,K)
C SUBROUTINE FOR VERSATEC PLOTTING OF THE PRESSURE COEFFICIENT
C VS NON-DIMENSIONAL CHORD (X/C) FOR EACH SECTION OF THE FINAL
C MESH
C REAL PX(165),PCP(165),XCL(85),XCLP(85),CPLO(85),CPUP(85),
  XCCD(161),CPI(161),ZP0(35),SCL(35),SCD(35),SCM(35),
  FMAC,YA,AL
INTEGER J,NUM,NUM1,II,12,NRD,K
C INITIALIZE ARRAYS AND DATA TO ZERO:
NUM = 0
NUM1 = 0
DO 10 I=1,165
  PX(I) = 0.0
  PCP(I) = 0.0
10 CONTINUE
DO 20 J=1,85
  XCL(J) = 0.0
  CPLO(J) = 0.0
  CPUP(J) = 0.0
20 CONTINUE
C READ IN X/C AND CP DATA INTO NEW ARRAY STARTING AT ARRAY
C ELEMENT NUMBER 1
DO 30 II=II,12
  PX(II-11) = XCCD(II)
  PCP(II-11) = CPI(II)
30 CONTINUE
PX(NRC) = 1.0
C PUT THE DATA INTO TWO ARRAYS, ONE FOR THE LOWER SURFACE
C AND ONE FOR THE UPPER SURFACE
NUM = (NRD-1)/2
NUM1 = NUM + 1
DO 40 I=1,NUM1
  XCL(I) = PX(I)
  CPLO(I) = PCP(I)
40 CONTINUE
DO 50 J=NUM1,NRD
  CPUP(J-NUM) = PX(J)
  CPUP(J-NUM) = PCP(J)
50 CONTINUE
C INITIALIZE THE VERSATEC PLITTER SYSTEM
CALL PICTS (0.0,0.0,0.0)

SCALE THE DATA TO AN 5.0 X 7.0 INCH SPACE
CALL SCALE (PCX,5.0,NDR+1)
CALL SCALE (PCP,5.0,NDR-1)

DRAW THE X AND Y AXIS
CALL AXIS (1.0,2.0,'X/C',-3.5,0.0,PCX(NDR+1),PCX(NDR+2))
CALL AXIS (1.0,2.0,'PRESSURE COEFFICIENT (CF)',25,
>7.0,90.0,PCP(NDR+1),PCP(NDR+2))

PUT SCALE FACTORS INTO TWO ARRAYS FOR UPPER AND LOWER SURFACE
XCLC(NL+1) = PCX(NDR+1)
XCLC(NL+2) = PCX(NDR+2)
CPQC(NL+1) = PCP(NDR+1)
CPQC(NL+2) = PCP(NDR+2)

PLCT THE DATA POINTS
CALL NEKPEN (2)
CALL PLCT (1.0,2.0,-3)
CALL LINE (XCLC,CPQC,NL1,1,-1,13)
CALL LINE (XCUP,CPUP,NL1,1,-1,11)

PLACE TITLE AT TOP OF PAGE
CALL NEKPEN (3)
CALL SYMBOL (1.25,7.5,0.2,'SECTION CP DATA ',0.0,16)
CALL NEKPEN (1)
CALL SYMBOL (1.25,7.25,0.1,'*', = UPPER SURFACE ',0.0,18)
CALL SYMBOL (1.25,7.0,0.1,'*', = LOWER SURFACE ',0.0,18)

PLACE THE ERECTING INFORMATION ON PLOT
CALL NEKPEN (2)
CALL SYMBOL (0.0,-0.75,0.1,'SPAN STATION = ',0.0,15)
CALL NUMBER (999.999,0.1,0.0,3)
CALL SYMBOL (0.0,0.0,0.1,'MACF ',0.0,6)
CALL NUMBER (999.999,0.1,0.0,3)
CALL SYMBOL (2.0,0.0,0.1,'YAW ',0.0,6)
CALL NUMBER (999.999,0.1,0.0,3)
CALL SYMBOL (4.0,0.0,0.1,'AOA ',0.0,5)
CALL NUMBER (999.999,0.1,0.0,3)
CALL SYMBOL (0.0,-0.25,0.1,'CL ',0.0,6)
CALL NUMBER (999.999,0.1,0.0,3)
CALL SYMBOL (2.0,-1.25,0.1,'CD ',0.0,5)
APPENDIX F

THIS APPENDIX PRESENTS THE SOURCE CODE FOR THE INTERACTIVE
PROGRAM FLO27IN

C*************************************************************************
C FLO27IN FORTRAN - 1C/05/83  JACK PASCHALL  AERO ENGINEERING
C A PROGRAM TO INTERACTIVELY DEFINE A POTENTIAL FLOW PROBLEM.
C THIS PROGRAM WRITES AN INPUT FILE "FLO27 INPUT A" TO THE USERS
C A DISK WHICH CAN BE SUBSEQUENTLY EXECUTED FOR JOB CARD ENTRY AND
C BATCH SYSTEM EXECUTION.
C*************************************************************************

REAL TLL(16), FMEF, FM, FNY, FNZ, FPL0, FIT(3), CWQ(3), P10L(3),
FMACH, YA, AL, CCO, ZSYM, FNS, SWEEP, DIRIF, FUS, ZS, XL, YL, CHOSC
TF0, FSEC, FN, YSYM, XP(I), YP(I), ZSYM, SCWH, SCWYP(67),
SCWYP(I-1), SYMFX, SYMXP(59), SYMYP(59), N572FN, N572XP(I-1),
X72YP(I+1), FPAM, FPX(31), FPAM(I), FL4PN, FL4XP(47), FL4YP(47),
A7FP, A7XP(31), A7YP(31), LISP, LISPX(49), LISPYP(49), N10FP,
N10XP(37), N10YP(37), N34FN, N34XP(45), N34YP(45), N35FN,
N35XP(45), N35YP(45), N64FN, N64XP(45), N64YP(45), N66FN,
N66XP(45), N66YP(45), N16FN, N16XP(45), N16YP(45), N63FN,
N63XP(51), N63YP(51), N63AFN, N63AXP(51), N63AYP(51), N64OFN,
N64OX(51), N640XP(51), N640XP(51), N640YP(51), N640YP(51), N65OFN,
N65OX(51), N65OX(51), N65AFN, N65AXP(51), N65AYP(51), N660FN,
N660XP(51), N660YP(51)
INTEGER I, IA, J, M, N1, FLAG, NUM, SCWNUM, SYMNUM, N57NUM, FPNUM,
FL14NUM, A7NUM, LISPNUM, N10NUM, N34NUM, N35NUM, N64NUM, N66NUM,
N16NUM, N63NUM, N63AN, N640N, N64AN, N650N, N65AN, N660N
C*************************************************************************

DATA ZEF0/0.0/, SCWPN/67.0/, SCWNUM/67.0/,
SCWXP/1.0, 9937.9883, 9683.6522, 9361.6508, 9713.6582,
9883.6569, 7335.6575, 6174.5764, 5322.4835,
4293.3683, 2587.2184, 1727.1216, 942.0648, 3335,
2104.0966, 0092.0028, 0055.0096, 0150.0204, 0335.0648,
1218.1727, 2184.2987, 3683.4293, 4835.5322, 5764.6174,
6575.6962, 7335.7694, 8043.6382, 8713.9038, 9361.9522,
9883.9937, 1.0/,
SCWYP/363926.0374, 0381.0414, 0438.0461, 0506.0544, 0579,
0611.0641, 0669.0695, 0725.0751, 0780.0805, 0827,
0847.0859, 0860.0843, 0826.0792, 0764.0723, 0654,
0607.0581, 0545.0567, 0547.0471, 0420.0378, 0326.0284,
0246.0206, 0167.0136, 0080.0000, -0045, -0073, -0109,
-0138, -0140, -0126, -0105, -0077, -0045, -0007, 0139,
0081, 0125, 0178, 0228, 0273, 031C, 0334, 0347, 0346,
0339, 0329, 0308, 0302, 0293/
### Symmetrical Wing Section Data

Data Symfn/59, 0/1 Symnum/59/

<table>
<thead>
<tr>
<th>Symxf</th>
<th>Symyf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.055</td>
<td>0.035</td>
</tr>
<tr>
<td>0.100</td>
<td>0.070</td>
</tr>
<tr>
<td>0.125</td>
<td>0.090</td>
</tr>
<tr>
<td>0.090</td>
<td>0.090</td>
</tr>
<tr>
<td>0.050</td>
<td>0.115</td>
</tr>
<tr>
<td>0.080</td>
<td>0.080</td>
</tr>
<tr>
<td>0.090</td>
<td>0.090</td>
</tr>
<tr>
<td>0.040</td>
<td>0.060</td>
</tr>
</tbody>
</table>

### NACA 24-30-0 Section Data

Data N572fn/41, 0/1 N57num/41/

<table>
<thead>
<tr>
<th>N57xf</th>
<th>N57yf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>0.050</td>
<td>0.035</td>
</tr>
<tr>
<td>0.100</td>
<td>0.090</td>
</tr>
<tr>
<td>0.090</td>
<td>0.090</td>
</tr>
<tr>
<td>0.050</td>
<td>0.115</td>
</tr>
<tr>
<td>0.080</td>
<td>0.080</td>
</tr>
<tr>
<td>0.090</td>
<td>0.090</td>
</tr>
<tr>
<td>0.040</td>
<td>0.060</td>
</tr>
<tr>
<td>0.020</td>
<td>0.020</td>
</tr>
</tbody>
</table>

### Flat Plate Data

Data Fpfn/3, 0/1 FPnum/3/1

<table>
<thead>
<tr>
<th>Fpxf</th>
<th>Fpyf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.050</td>
<td>0.030</td>
</tr>
<tr>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>0.025</td>
<td>0.030</td>
</tr>
<tr>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### F-14 Wing Section - Typical

Data F14fn/47, 0/1 F14num/47/

<table>
<thead>
<tr>
<th>F14xf</th>
<th>F14xf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.050</td>
<td>0.030</td>
</tr>
<tr>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>0.025</td>
<td>0.030</td>
</tr>
<tr>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.010</td>
<td>0.010</td>
</tr>
<tr>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>0.020</td>
<td>0.020</td>
</tr>
<tr>
<td>0.030</td>
<td>0.030</td>
</tr>
<tr>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>C</td>
<td>NACA 64-01C SECTION DATA</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Na40XF/1.0000, 0550, 0500, 0500, 0850, 0600, ...</td>
</tr>
<tr>
<td>2</td>
<td>0550, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>3</td>
<td>0500, 0500, 0500, 0500, 0850, 0600, 0850, 0600, ...</td>
</tr>
<tr>
<td>4</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>5</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>6</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>7</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>C</td>
<td>NACA 64010 SECTION DATA</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------</td>
</tr>
<tr>
<td>1</td>
<td>N64AXF/1.0000, 0550, 0500, 0500, 0850, 0600, ...</td>
</tr>
<tr>
<td>2</td>
<td>0550, 0500, 0500, 0500, 0500, 0500, 0850, 0600, ...</td>
</tr>
<tr>
<td>3</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>4</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
<tr>
<td>5</td>
<td>0500, 0500, 0500, 0500, 0500, 0500, 0500, 0500, ...</td>
</tr>
</tbody>
</table>
C TITLE PAGE AND INSTRUCTIONS
CALL FRICMS ('CLRSCRN ')

C FIRST LINE INPUT DATA--DEFINE COMPUTATIONAL GRID
WRITE (6,450)
WRITE (6,460)
READ (5,1000) TITLE
CONTINUE
WRITE (6,470)
READ (5,*) FMESH
WRITE (6,480)
READ (5,*) FNX
WRITE (6,490)
READ (5,*) FNY
CALL FRICMS ('CLRSCRN ')
WRITE (6,500)
READ (5,*) FNZ
WRITE (6,510)
READ (5,*) FPLOT

C SUMMARY OF FIRST LINE INPUT DATA
CALL FRICMS ('CLRSCRN ')
WRITE (6,520)
READ (5,*) ANS
IF (ANS.GE.2) GC TC 20
WRITE (6,521)
WRITE (6,522) FNX,FNY,FMZ,FMESH,FPLOT
WRITE (6,530)
READ (5,*) ANS
IF (ANS.EQ.1) GO TO 10

C SECOND, THIRD AND FORTH LINES INPUT DATA--ITERATION AND CONVERGENCE
C TOLERANCE FOR GRID. NUMBER OF LINES EQUAL TO M = FMESH
M = IFI(FMESH)
CALL FRICMS ('CLRSCRN ')

WRITE (6,450)
DO 30 I=1,N
  IF (I.EQ.1) WRITE (6,541)
  IF (I.EQ.2) WRITE (6,542)
  IF (I.EQ.3) WRITE (6,543)
  WRITE (6,550)
  REAL (5,*) FIT(I)
  WRITE (6,560)
  REAL (5,*) COVO(I)
  WRITE (6,570)
  REAL (5,*) PI0(I)
CALL FRICMS ('CLRSCHR')
30 CONTINUE

C SUMMARY OF SECOND, THIRD AND FOURTH LINES INPUT DATA

C CALL FRICMS ('CLRSCHR')
WRITE (6,580)
READ (5,*) ANS
IF (ANS.GE.2) GC TO 40
  WRITE (6,581)
  WRITE (6,582) (FIT(I),COVO(I),PI0(I)),I=1,N
  WRITE (6,590)
  REAL (5,*) ANS
  IF (ANS.EQ.1) GO TO 20
40 CONTINUE

C FIFTH LINE INPUT DATA--MACH NO., YAW ANGLE, AOA, SKIN FRICTION DRAG

C CALL FRICMS ('CLRSCHR ')
WRITE (6,450)
WRITE (6,600)
READ (5,*) FMACH
WRITE (6,610)
READ (5,*) YA
WRITE (6,620)
READ (5,*) AL
WRITE (6,630)
REAL (5,*) C00

C SUMMARY OF FIFTH LINE INPUT DATA

C CALL FRICMS ('CLRSCHR ')
WRITE (6,640)
READ (5,*) ANS
IF (ANS.GE.2) GC TO 50
  WRITE (6,641)
  WRITE (6,642) FMACH,YA,AL,C00
WRITE (6, 650)
READ (5, *) ANS
IF (ANS .EQ. 1) GC TO 40
50 CONTINUE

C SIXTH LINE INPUT DATA--WING PLANFORM SYMMETRY, NUMBER OF SECTIONS,
C SWEEP, DIHEDRAL ANGLE AND FUSELAGE RADIUS
C
CALL FRICMS ('CLCRSCRN *)
WRITE (6, 450)
WRITE (6, 660)
READ (5, *) ZSYM
WRITE (6, 670)
READ (5, *) FNS
WRITE (6, 680)
READ (5, *) SWEEP
WRITE (6, 690)
READ (5, *) DIHED
WRITE (6, 700)
READ (5, *) FUS

C SUMMARY OF SIXTH LINE INPUT DATA
C
CALL FRICMS ('CLCRSCRN *)
WRITE (6, 710)
READ (5, *) ANS
IF (ANS .GE. 2) GC TO 60
WRITE (6, 711)
WRITE (6, 712) ZSYM, FNS, SWEEP, DIHED, FLS
WRITE (6, 720)
READ (5, *) ANS
IF (ANS .EQ. 1) GO TO 50
60 CONTINUE

C WRITE JCL CARDS TO TOP OF FILE ON USER'S "A" DISK
C
WRITE (6, 1200)
WRITE (6, 1210)

C WRITE FIRST SIX LINES OF DATA TO FILE ON USER'S "A" DISK
C
WRITE (6, 1010) TITLE
WRITE (6, 1020) FMCH, YA, AL, CDO
WRITE (6,1080)
WRITE (6,1090) ZSYM,FNS,SLEEP,CIHEL,FUS

C SECTION INLET DATA

N = IFIX(FNS)
CALL FRICMS ('CLRSCRN ')
WRITE (6,45C)
WRITE (6,730)
WRITE (6,410)
DO 200 I=1,N
WRITE (6,760) I
WRITE (6,770)
REAL (5,*) ZS
WRITE (6,780)
REAL (5,*) XL
WRITE (6,790)
REAL (5,*) YL
WRITE (6,800)
REAL (5,*) CHRLC
WRITE (6,810)
REAL (5,*) THICK
WRITE (6,820)
READ (5,*) AT
CALL FRICMS ('CLRSCRN ')
WRITE (6,830)
REAL (5,*) FSEC
WRITE (8,1100)
WRITE (8,1110) ZS, XL, YL, CHRLC, THICK, AT, FSEC
IF (FSEC .EQ. 0.0) GO TO 190
CALL FRICMS ('CLRSCRN ')
WRITE (6,550)
WRITE (6,740)
WRITE (6,740)
WRITE (6,750)
REAL (5,*) ANS
FLAG = ANS
IF (FLAG .GT. 0) GO TO 70
WRITE (6,840)
READ (5,*) YSYM
CALL FRICMS ('CLRSCRN ')
WRITE (6,850)
REAL (5,*) FN

C USER INPUT X AND Y COORDINATES FOR WING SECTION DEFINING GEOMETRY

NUM = IFIX(FN)
CALL FRICMS ('CLRSCRN ')
WRITE (6,450)
WRITE (6,860) I
DO 180 J=1,NUM
WRITE (6,870)
READ (5,*) XP(J),YP(J)
180 CONTINUE
IF (YSYM .EQ. 0.0) GC TC 185
N1 = NUM + NUM
J1 = NUM
DO 182 IA=1,J1
XP(N1-IA) = XP(IA)
YP(N1-IA) = -YP(IA)
183 CONTINUE
NUM = N1 - 1
FN = FN + (FN - 1.0)
185 CONTINUE
WRITE (8,1120)
WRITE (8,1130) FN
WRITE (8,1140)
WRITE (8,1150) ((XP(J),YP(J)),J=1,NUM)
GO TO 190
70 CONTINUE
C MENU INPUT X AND Y COORDINATES FOR WING SECTION DEFINING GEOMETRY
C
IF (FLAG .EQ. 1) GO TO 81
IF (FLAG .EQ. 2) GO TO 82
IF (FLAG .EQ. 3) GO TO 83
IF (FLAG .EQ. 4) GO TO 84
IF (FLAG .EQ. 5) GO TO 85
IF (FLAG .EQ. 6) GO TO 86
IF (FLAG .EQ. 7) GO TO 87
IF (FLAG .EQ. 8) GO TO 88
IF (FLAG .EQ. 9) GO TO 89
IF (FLAG .EQ. 10) GO TO 90
IF (FLAG .EQ. 11) GO TO 91
IF (FLAG .EQ. 12) GO TO 92
IF (FLAG .EQ. 13) GO TO 93
IF (FLAG .EQ. 14) GO TO 94
IF (FLAG .EQ. 15) GO TO 95
IF (FLAG .EQ. 16) GO TO 96
IF (FLAG .EQ. 17) GO TO 97
IF (FLAG .EQ. 18) GO TO 98
IF (FLAG .EQ. 19) GO TO 99
IF (FLAG .EQ. 20) GO TO 100
81 CONTINUE
WRITE (8,1120)
WRITE (8,1130) FFPN
WRITE (8,1140)
WRITE (8,1150) ((FPXP(J),FPYP(J)),J=1,FNUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) SYMFN
WRITE (8,1140)
WRITE (8,1150) ((SYMP(J),SYMP(J)),J=1,SYMNUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) SCMFN
WRITE (8,1140)
WRITE (8,1150) ((SCWM(J),SCWM(J)),J=1,SCWMNUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N572FN
WRITE (8,1140)
WRITE (8,1150) ((N572P(J),N572P(J)),J=1,N57NUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) F14FN
WRITE (8,1140)
WRITE (8,1150) ((F14XP(J),F14YP(J)),J=1,FNUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) A7FN
WRITE (8,1140)
WRITE (8,1150) ((A7XP(J),A7YP(J)),J=1,A7NUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) LISFN
WRITE (8,1140)
WRITE (8,1150) ((LISXP(J),LISYP(J)),J=1,LISNUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N1CFN
WRITE (8,1140)
WRITE (8,1150) ((N10XP(J),N10YP(J)),J=1,N10NUM)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N34FN
WRITE (8, 1140) (N34XP(J), N34YP(J)), J=1, N34NUM)
CONTINUE
90
WRITE (8, 1150) (N35XP(J), N35YP(J)), J=1, N35NUM)
CONTINUE
91
WRITE (8, 1120) N35FN
WRITE (8, 1130) N64FN
WRITE (8, 1140) N64YN
WRITE (8, 1150) (N64XP(J), N64YP(J)), J=1, N64NUM)
CONTINUE
92
WRITE (8, 1120) N66FN
WRITE (8, 1130) N66YN
WRITE (8, 1140) (N66XP(J), N66YP(J)), J=1, N66NUM)
CONTINUE
93
WRITE (8, 1120) N16FN
WRITE (8, 1130) N16YN
WRITE (8, 1140) (N16XP(J), N16YP(J)), J=1, N16NUM)
CONTINUE
94
WRITE (8, 1120) N63FN
WRITE (8, 1130) N63YN
WRITE (8, 1140) (N63XP(J), N63YP(J)), J=1, N63NUM)
CONTINUE
95
WRITE (8, 1120) N63AFN
WRITE (8, 1130) N63AFP(J), J=1, N63AN)
CONTINUE
96
WRITE (8, 1120) N64FN
WRITE (8, 1130) N64YP(J), J=1, N64ON)
CONTINUE
97
WRITE (8, 1130) N64AFN
```
WRITE (8,1140)
WRITE (8,1150) ((N64AXP(J),N64AYP(J)),J=1,N64AN)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N650FN
WRITE (8,1140)
WRITE (8,1150) ((N65AXP(J),N65AYP(J)),J=1,N65AN)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N65AFN
WRITE (8,1140)
WRITE (8,1150) ((N65AXP(J),N65AYP(J)),J=1,N65AN)
GO TO 190
CONTINUE
WRITE (8,1120)
WRITE (8,1130) N660FN
WRITE (8,1140)
WRITE (8,1150) ((N66AXP(J),N66AYP(J)),J=1,N66AN)
GO TO 190
CONTINUE
CALL FRTCMS (*CLRSCRN *)
190 CALL FRTCMS (*CLRSCRN *)
200 CONTINUE
C WRITE THREE LINES TO USER'S FILE INDICATING END OF CALCULATION
C WRITE (E,1160)
WRITE (E,1170)
WRITE (E,1180) ZERC
C WRITE JCL CARDS TO BOTTOM OF FILE CN USER'S "A" DISK
C WRITE (E,1220)
WRITE (E,1230)
C INDICATE TO USER THAT INPUT IS COMPLETE
C CALL FRTCMS (*CLRSCRN *)
WRITE (E,450)
WRITE (E,480)
WRITE (E,410)
WRITE (E,890)
C FORMAT STATEMENTS
C RETURN
410 FORMAT (1X,79H==================================================================
1======================================================================)
```
FORMAT (/,'25X,24H'FLO27 DATA INPUT PROGRAM,'/,'33X,7THAE=4501//)
430 FORMAT (8X,66H*PROGRAM INTERACTIVELY WRITES A DATA FILE TO YOU
1R A DISK WHICH, /,8X,66HCAN SUBSEQUENTLY BE USED TO RUN THE POTENTI
2AL FLOW PROGRAM"'/,25X//)
440 FORMAT (/,'8X,52HENTER DATA FOR THE PENTIAL FLOW PROGRAM IN FREE
1 FORMAT.',/,'8X,52HIF EACH QUESTION THE FORMAT IS GIVEN: (R) = RE
2AL ,/,'8X,14H(I) = INTEGER.*/,'8X,45HEXAMPLE: (R,R) INPUT 2.5,6.789
3OR (/,I) INPUT 5,//)
450 FORMAT (1X,79H****************************** FLO27 INPUT PARAMETERS
1*******************************)
460 FORMAT (1X,28H==> ENTER THE PROBLEM TITLE:,//,5X,20H64 LETTERS MAX
1HUM)*)
470 FORMAT (1X,76H==> ENTER THE NUMBER OF TIMES THE COMPUTATIONAL MESH
1 IS REFINED (FMESH): (R),/,'1X,22H1.0 = COARSE MESH ONLY,/*,1X,27H2.
20 = REFINE TO MEDIUM MESH,/*,1X,25H3.0 = REFINE TO FINE MESH)
480 FORMAT (1X,56H==> ENTER THE NUMBER OF CO COMPUTATIONAL CELLS IN CHORD
WISE,/*,5X,41HDIRECTION FOR THE INITIAL MESH (FNX): (R),/,'5X,47HIMA
2XI NUM NC. = 160/2**N, WHERE N = FMESH - 1.0))
490 FORMAT (1X,63H==> ENTER THE NUMBER OF COMPUTATIONAL CELLS IN NORMAL
1L DIRECTION,/*,5X,52HFRCM AIRFOIL SURFACE FOR THE INITIAL MESH (FNY
2): (R),/,'5X,46H(MAXIMUM NO. = 16/2**N, WHERE N = FMESH - 1.0)
500 FORMAT (1X,55H==> ENTER THE NUMBER OF COMPUTATIONAL CELLS IN SPAN
WISE,/*,5X,41HDIRECTION FOR THE INITIAL MESH (FNU): (R),/,'5X,46H(MAX
2NUM NC. = 32/2**N, WHERE N = FMESH - 1.0))
510 FORMAT (1X,54H==> ENTER FLAG FOR PRINTER-PLOTTING OF CP (FPLT): (R)
1R),/,'1X,5CH0.0 = PRINTER OUTPUT OF CP WITH/CT CP PRINTER-PLOT,/*,1X
2,47H1.C = PRINTER OUTPUT OF CP WITH CP PRINTER-PLOT,/*,1X,57H2.0 =
3PRINTER OUTPUT OF CP AND VERSATEC PLOT CF CP VS X/C,/*,7X,34HFOR EA
4CH SECTION CF THE FINAL MESH)
520 FORMAT (1X,33H SUMMARY OF FIRST LINE INPUT DATA,?/,'1X,25H===> ENTER
11 = YES; 2 = NO)
521 FORMAT (1X,3HFNX,7X,3HFNY,7X,3HFNZ,7X,5HFMECH,5X,5HFPLT)
522 FORMAT (1X,5F5.1,3F5.1,6X,F4.1,6X,F3.1,7X,F3.1,1//)
530 FORMAT (1X,29HCHANGE FIRST LINE INPUT DATA,?/,'1X,25H===> ENTER 1 =
1YES; 2 = NO)
540 FORMAT (26X,28H***COARSE MESH PARAMETERS***//)
542 FORMAT (26X,28H***MEDIUM MESH PARAMETERS***//)
543 FORMAT (26X,26H***FINE MESH PARAMETERS***//)
550 FORMAT (1X,58H==> ENTER MAXIMUM NUMBER OF ITERATIONS FOR MESH (FIT
1): (R),/,'5X,41HRECOMMEND: 100 ITERATIONS FOR COARSE MESH,/*,17X,29H
220 ITERATIONS FOR MEDIUM MESH,/*,17X,21H10 ITERATIONS FOR MEDIUM MESH
3)
560 FORMAT (1X,66H==> ENTER CONVERGENCE TOLERANCE FOR VELOCITY POTENTI
1AL (CGVC): (R),/,'5X,18HRECOMMEND: .003001)
570 FORMAT (1X,62H==> ENTER SUBSONIC POINT RELAXATION FACTOR FOR MESH
26 FOR COARSE MESH,/*,16X,19H1.3 FOR MEDIUM MESH,/*,16X,17H1.2 FOR FINE
3 MESH)
FORMAT (1X,55HSUMMARY CF SECONC, THIRD AND FCRTH LINES OF INPUT DATA,
1TA?; ,1X,25H===> ENTER 1 = YES; 2 = NO)
581 FORMAT (1X,3HFIT,7X,5HCVRG,6X,3HP10)
582 FORMAT (1X,F5.1,5X,F7.3,F3.11)
583 FORMAT (1X,5HCHANGE SECONC, THIRD AND FCRTH LINES INPUT DATA?,
1/ ,1X,25H===> ENTER 1 = YES; 2 = NO)
600 FORMAT (1X,46H===> ENTER FREE STREAM MACH NUMBER (FMACH): (R))
610 FORMAT (1X,4OH===> ENTER YAW ANGLE IN DEGREES (YA): (R))
620 FORMAT (1X,46H===> ENTER ANGLE OF ATTACK IN DEGREES (AL): (R))
630 FORMAT (1X,53H===> ENTER DRAG COEFFICIENT DUE TO SKIN FRICTION (CD0
1/ ,5X,48H(UNLESS OTHERWISE AVAILABLE CD0 IS RECOMMENDED))
640 FORMAT (1X,33HSUMMARY CF FIFTH LINE INPLT DATA?, ,1X,25H===> ENTER
11 = YES; 2 = NO)
641 FORMAT (1X,5FHMAC,5X,2HYA,8X,2HAL,8X,3HCD0)
642 FORMAT (1X,F4.2,EX,F3.1,7X,F3.1,7X,F8.6//)
650 FORMAT (1X,29HCHANGE FIFTH LINE INPUT DATA?, ,1X,25H===> ENTER 1 =
1YES; 2 = NO)
660 FORMAT (1X,52H===> ENTER WING PLANFORM SYMMETRY TRIGGER (ZSYM): (R)
1/ ,1X,41H0.0 = YAWED WING HAS NO SPANWISE SYMMETRY, ,1X,38H1.0 = S
2HEPT WING HAS SPANWISE SYMMETRY)
670 FORMAT (1X,78H===> ENTER NUMBER OF SECTIONS WHERE WING SECTION GEOM
1TRY IS DEFINED (FNS): (R), ,5X,53H(VALUE MUST BE > OR = 3.0, BUT
2NOT GREATER THAN 11.0))
680 FORMAT (1X,56H===> ENTER LEADING EDGE SWEET ANGLE IN DEGREES (SWEEP
1/ ,1: (R))
690 FORMAT (1X,48H===> ENTER DIHEDRAL ANGLE IN DEGREES (DIHED): (R))
700 FORMAT (1X,36H===> ENTER FUSELAGE RADIUS (FUS): (R), ,5X,33HNOTE: U
1SE = 0.0 FOR WING-ALONE CASE)
710 FORMAT (1X,33HSUMMARY OF SIXTH LINE INPLT DATA?, ,1X,25H===> ENTER
11 = YES; 2 = NO)
711 FORMAT (1X,4HZSYM,6X,3HFNS,7X,5HSWEEP,5X,5HC1HED,5X,3HFUS)
712 FORMAT (1X,F3.1,7X,F4.1,6X,F6.3,4XF6.2,4XF10.6//)
720 FORMAT (1X,29HCHANGE SIXTH LINE INPLT DATA?, ,1X,25H===> ENTER 1 =
1YES; 2 = NO)
730 FORMAT (5X,65HTHE NEXT SET OF INPUT DATA WILL BE REPEATED FOR EACH
1WING SECTION, ,5X,63HX AND Y COORDINATES DEFINING THE WING SECTI
2ON SHAPE ARE ENTERED, ,5X,67HSTARTING WITH THE UPPER SURFACE TRAIL
3ING EDGE AND PROCEEDING AROUND, ,5X,35HTO THE LOWER SURFACE TRAIL;
4NG EDGES, ,5X,57HX AND Y COORDINATES ARE NORMALIZED WITH THE CHORD
5LENGTH, ,5X,46HWHERE SECTION DEFINING COORDINATES CAN BE INPUT, /,
65X,36HYTHE USER OR SELECTED FROM A MENU.)
740 FORMAT (28X,23H**WING SECTION MENU**,, ,6X,34H0 = USER INPLT SECT
1ION COCFO, DATA,, ,6X,19H1 = FLAT PLATE DATA,, ,6X,49H2 = SYMMETRIC
2L WING (1% THICKNESS AT 30% CHORD), ,6X,61H3 = SUPERCRITICAL WING
3CAMBERED, 12% THICKNESS AT 32% CHORD), ,6X,51H4 = NACA 24-30 (CA
4MBER), 12% THICKNESS AT 30% CHORD), ,6X,52H5 = F14 WING (CAMBERED,
59, 5% THICKNESS AT 37% CHORD), ,6X,64H6 = A-7 WING (7 DEG DROOP AT
620% CHORD, 7% THICKNESS AT 43% CHORD), ,6X,64H7 = LISSAMAN 7769 AI
7RFClL (CAMBERED, 11% THICKNESS AT 30% CHORD), /, 6X, 55H8 = NACA 0010
8 (SYMMETRICAL, 10% THICKNESS AT 30% CHORD), /, 6X, 58H9 = NACA 0010-3
94 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD), /, 5X, 59H10 = NACA 00
110-35 (SYMMETRICAL, 10% THICKNESS AT 50% CHORD), /, 5X, 59H11 = NACA 00
110-64 (SYMMETRICAL, 10% THICKNESS AT 60% CHORD), /, 5X, 59H12 = NACA
200-6 (SYMMETRICAL, 10% THICKNESS AT 60% CHORD), /, 5X, 59H13 = NACA
3A 12-0CS (SYMMETRICAL, 9% THICKNESS AT 50% CHORD), /, 5X, 59H14 = NAC
4A 63-016 (SYMMETRICAL, 10% THICKNESS AT 35% CHORD), /, 5X, 59H15 = NA
5CA 63A010 (SYMMETRICAL, 10% THICKNESS AT 35% CHORD), /, 5X, 59H16 = NA
6ACA 66-010 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD), /, 5X, 59H17 = NA
7ACA 66A010 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD), /, 5X, 59H18 = NA
866-010 (SYMMETRICAL, 10% THICKNESS AT 45% 2 CHORD), /, 5X, 59H19 = NAC
9741 FORMAT (5X, 59H15 = NACA 65A010 (SYMMETRICAL, 10% THICKNESS AT 40%
1CHORD), /, 5X, 58H20 = NACA 66-010 (SYMMETRICAL, 10% THICKNESS AT 45%
2 CHORD), /)
750 FORMAT (1X, 35H=> ENTER DESIRED NUMBER FROM MENU.)
760 FORMAT (15X, 22H***WING SECTION NUMBER, 12, 1X, 13HPARAMETERS***, /, 5X,
142HNOTE: ALL DIMENSIONS MUST BE IN SAME UNITS, /)
770 FORMAT (1X, 40H=> ENTER THE SPANWISE CORDINATE FOR THIS SECTION (1
12S): (RF))
780 FORMAT (1X, 53H=> ENTER SECTION LEADING EDGE X COORDINATE (XL): (RF)
1))
790 FORMAT (1X, 53H=> ENTER SECTION LEADING EDGE Y COORDINATE (YL): (RF
1))
800 FORMAT (1X, 43H=> ENTER SECTION CHORD LENGTH (CHORD): (RF)
810 FORMAT (1X, 55H=> ENTER SECTION THICKNESS SCALING FACTOR (THICK): (RF
1))
820 FORMAT (1X, 50H=) ENTER SECTION TWIST ANGLE IN DEGREES (AT): (RF)
830 FORMAT (1X, 56H=> ENTER FLAG FOR DEFINING NEW WING SECTION GEOMETRY
1Y (YSEC): (RF), /, 1X, 40H1.0 = DEFINE A NEW WING SECTION GEOMETRY, /, 1
2X, 67H0. = COPY THE WING SECTION DEFINING GEOMETRY FROM PREVIOUS'S
3SECTION, /, 7X, 38HNOTE: FOR FIRST SECTION MUST ENTER 1.0)
840 FORMAT (1X, 40H=> ENTER FLAG FOR INICATING WHETHER OR NOT, /, 5X43H
1THE WING SECTION IS SYMMETRICAL (YSYM): (RF), /, 1X, 20H0.0 = NCSYMMETRICAL, /, 1X, 25H1.0 = SYMMETRICAL SECTION, /, 1X, 33HNOTE: IF SYMMETRICAL SELECTION, YOU ONLY HAVE TO INPUT, /, 1X, 46HDEFINING PCNTS FOR T
4HE SECTION UPPER SURFACE).
850 FORMAT (1X, 58H=> ENTER NUMBER OF WING SECTION DEFINING POINTS (FN
1): (RF), /, 1X, 38HNOTE: MAXIMUM NUMBER OF POINTS IS 161.)
860 FORMAT (17X, 22H***WING SECTION NUMBER, 12, 1X, 22HX AND Y COORDINATES
1***/)
870 FORMAT (1X, 67H=> ENTER WING SECTION DEFINING POINT X AND Y COORDIN
1 (XP, YP): (RF, RF))
880 FORMAT (5X, 64HTHREE ADDITIONAL DATA LINES WILL BE AUTOMATICALLY WR
ritten in the file, /, 5X, 43HBOTTOM OF YOUR INPUT FILE. THESE LINES ARE: /, 2
5X, 18HEND OF CALCULATION, /, 5X, 43HBOTTOM OF YOUR INPUT FILE. THESE LINES ARE: /, 5X, 30H1)
890 FORMAT (5X, 70HTHIS IS COMPLETED THE INPUT DATA. A FILE WITH <FILENAME
1> <FILETYPE> FLG27, /, 5X, 69HDATAIN HAS BEEN WRITTEN TO YOUR "A" DIS
2K. IF YOu WISH TO MAKE FURTHER,5X,62+CHANCES TO YOUR INPUT DATA
3SIMPLY XEDIT THE CREATED DATA FILE.,5X,62HTO RUN THE POTENTIAL
4FLCw PPCGRAM (FLC27) USING YOUR DATA FILE.,5X,61HXEDIT THE FILE
5AND ENTER THE ADDITIONAL CARDS (JOB CARD ETC.),5X,60HAS OUTLINED
6IN THE INSTRUCTIONS, THEN SUBMIT THE FILE TO THE.,5X,16HBATCH PR
7OCESSOR.,5X,4HBEYe.)

1000 FORMAT (1x,16(A4))
1010 FORMAT (1x,16(A4))
1020 FORMAT (3HFNX,7X,3HFNY,7X,3HFNZ,7X,5HF ,MESH,5X,5HFPLCT)
1030 FORMAT (5.1,5X,F4.1,6X,F4.1,6X,F3.1,7X,F3.1)
1040 FORMAT (3HFIT,7X,4HC0VC,6X,3HP10)
1050 FORMAT (5.1,5X,F7.6,3X,F3.1)
1060 FORMAT (5HEMACH,5X,2HYA,8X,2HAL,8X,3HCDC)
1070 FORMAT (F4.2,6X,F3.1,7X,F3.1,7X,F8.6)
1080 FORMAT (4H2SYH,6X,3HFNS,7X,5HSHEEP,5X,5HDIHED,5X,3HFUS)
1090 FORMAT (F3.1,7X,F4.1,6X,F6.3,4X,F6.3,4X,F10.6)
1100 FORMAT (2H2S,8X,2HAL,8X,2HYA,8X,5HCHCRD,5X,5THICK,5X,
1200 FORMAT (8X,41FSEC))
1110 FORMAT (6(F8.4,2X),F3.1)
1120 FORMAT (2FFN)
1130 FORMAT (F5.1)
1140 FORMAT (5HXP(1),5X,5HYP(1))
1150 FORMAT (6F10.7)
1160 FORMAT (1X,18HEAD OF CALCULATION)
1170 FORMAT (3FNX)
1180 FORMAT (F3.1)
1200 FORMAT (16H//EXEC FLC27)
1210 FORMAT (17H//GO SY SIN DD *)
1220 FORMAT (2H/*)
1230 FORMAT (2H//

ENC
### LIST OF REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Initial Distribution List</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Technical Information Center Cameron Station Alexandria, Virginia 22314</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>Library, Code 0142 Naval Postgraduate School Monterey, California 93943</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943</td>
<td>1</td>
</tr>
<tr>
<td>4.</td>
<td>Dr. Robert D. Zucker, Code 67Zu Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943</td>
<td>5</td>
</tr>
<tr>
<td>5.</td>
<td>Dr. Oscar Bitlarz, Code 67Bi Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943</td>
<td>2</td>
</tr>
<tr>
<td>6.</td>
<td>LCDR Lowell D. Boaz, Code 67Bo Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>CDR Jack Paschall III, Code 3107 Naval Weapons Center China Lake, California 93555</td>
<td>2</td>
</tr>
</tbody>
</table>
A three-dimensional transonic, potential flow computer program, its conversion to IBM Fortran and utilization.