INVESTIGATION OF TURBULENT HEAT TRANSFER AT HYPersonic SPEEDS

Volume III. The Laminar-Turbulent $\rho_r u_r$ Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs

R. T. Savage
C. L. Jaeck
J. R. Mitchell

THE BOEING COMPANY

TECHNICAL REPORT AFFDL-TR-67-144, VOLUME III

December 1967

This document has been approved for public release and sale; its distribution is unlimited.

Air Force Flight Dynamics Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

Reproduced by the CLEARINGHOUSE for Federal Scientific & Technical Information Springfield Va 22151

DDC
MAR 1 1968
C

232.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
INVESTIGATION OF TURBULENT
HEAT TRANSFER AT HYPERSONIC SPEEDS

Volume III. The Laminar-Turbulent $\rho_r\mu_r$ Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs

R. T. Savage
C. L. Jaeck
J. R. Mitchell

This document has been approved for public release and sale; its distribution is unlimited.
NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned to the Air Force Flight Dynamics Laboratory unless return is required by security considerations, contractual obligation, or notice on a specific document.
FOREWORD

This report was prepared by the Space Division, Aerospace Group of The Boeing Company, Seattle, Washington, under direction of Messrs. A. L. Nagel and V. Derigun, program managers. The contract was initiated under BPSN 5(611366-62405334), Project 1366 Hypersonic Gas Dynamic Heating, Task 136607 Aerodynamics and Flight Mechanics, USAF Contract AF33(615)-2372, Investigation of Turbulent Heat Transfer at Hypersonic Speeds. The work was administered by the Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. Mr. Richard D. Neumann (FDMG) was the Air Force project engineer.

Results obtained during this program are published in three volumes. Volume I, Analytical Methods; Volume II, Analysis of Heat Transfer and Pressure Data on a Flat Plate, Cone, Ogive, Cylindrical Leading Edge, Blunt Delta Wing, and X-15 Aircraft; and Volume III, The Laminar-Turbulent Turbulent Momentum Integral and Turbulent Nonsimilar Boundary Layer Computer Programs. Boeing document numbers assigned to these volumes are D2-113531-1, -2, and -3, respectively.

This report covers work conducted between March 1965 and March 1967. The report was submitted by the authors in May 1967.

The authors acknowledge Barbara J. Safley for her exceptional effort in generating and editing the figures contained in this report.

This technical report has been reviewed and is approved.
This report presents a combined analytical and experimental investigation of turbulent heat transfer on basic and composite configurations at hypersonic speeds. The analytical results are presented in Volume I, the experimental results, including data-theory comparisons, are presented in Volume II, and computer programs incorporating the analytical methods described herein are presented in Volume III.

The two heat-transfer prediction methods programmed are the laminar-turbulent $\rho_r \mu_r$ momentum integral method and the turbulent nonsimilar boundary layer method. This volume of the report describes the numerical method and presents flow charts, program listings, input forms, and a description of the output for each program. The programs are written in Fortran IV language for operation on the IBM 7094.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>I</th>
<th>INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>THE LAMINAR-TURBULENT MOMENTUM INTEGRAL COMPUTER PROGRAM ($\rho_r \mu_r$)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1. DISCUSSION</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>a. Method of Solution</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>b. $\rho_r \mu_r$ Program Schematic</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>c. Nomenclature</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2. $\rho_r \mu_r$ PROGRAM DESCRIPTION</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>a. $\rho_r \mu_r$ Flow Chart</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>b. Subroutines</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>1) ATMOS</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2) AXI2D (Option D-1)</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3) BEQI</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4) CYLIND (Option D-3)</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>5) DBTP</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>6) DELTA (Option D-4)</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>7) DERV</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>8) DIVERG</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>9) DSIRCH</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>10) EBAR</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>11) FIND</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>12) FLOW (Program B)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>13) HEMI (Option D-2)</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>14) INITIAL</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>15) JWALL</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>16) JAYELL</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>17) MUZERO</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>18) QTRANS (Program A)</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>19) RFF</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>20) RORMUR</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>21) SIRCH</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>22) SOMEGA</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>23) SONENT</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>24) STACON</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>25) STREAM (Program C)</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>26) TABLE2</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>27) TABLE6</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>28) TABLE7</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>29) TABL14</td>
<td>66</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

30) TAB118 67
31) TAB119 67
32) TAB120 68
33) TBLP 68

3. INPUT-OUTPUT DESCRIPTION 69
   a. Data Input Preparation 69
      1) Axisymmetric or two-dimensional bodies
         (Option D-1) 69
      2) Hemisphere (Option D-2) 74
      3) Swept infinite cylinder (Option D-3) 77
      4) Sharp delta wing (Option D-4) 81
      5) Streamline calculations (Program C) 87
      6) Reference calculations (Program REF) 93
      7) Boundary layer calculations (Program B) 95
      8) Heat transfer calculations (Program A) 99
   b. Output Description 104
   c. Error Statements 108

4. PROGRAMMING INFORMATION 110
   a. Program Listing 110
   b. Program Listing 182

III THE TURBULENT NONSIMILAR BOUNDARY LAYER PROGRAM (NSBL) 186

1. NOMENCLATURE 186

2. METHOD OF SOLUTION 189
   a. Forward Integration 189
   b. Numerical Method of Solution 191
      1) Calculation of tabular values 192
      2) Values obtained from tables 192
      3) Computation at initial x = x0 192
      4) Calculation of δη 192
      5) Calculation of y-derivatives and v-profile 193
      6) Calculation of x derivatives 196
      7) Calculation of δ*, θ, Qw, T, and τw, L, and τw, T at x 196
      8) Calculation of u and l-profiles at x + Δx 196
   c. Stagnation Region Calculation 197
TABLE OF CONTENTS (Concluded)

3. INPUT-OUTPUT DESCRIPTION
   a. Data Input Preparation
      1) Data card formats 198
      2) Case dependence 198
   b. Output Description 203
   c. NSBL Input Form 204

4. PROGRAMMING INFORMATION
   a. NSBL Flow Chart 206
   b. NSBL Program Listing 207

REFERENCES 222

LIST OF ILLUSTRATIONS

Figure

1. Pressure distributions on a hemisphere and unswept infinite cylinder 6
2. Boeing modified Newtonian hypersonic pressure coefficients 7
3. Streamline correlation for sharp delta wings 24
4. Correction factor for the velocity gradient at a hemisphere stagnation point or cylinder stagnation line 62
5. Streamlines near the apex of a sharp delta wing 82
SECTION I

INTRODUCTION

Two methods for the calculation of turbulent heating rates are discussed in Volume I of this report, the laminar-turbulent momentum integral method ($\rho_r \mu_r$) and the nonsimilar method for turbulent boundary layers (NSBL).

This volume describes the computer program for each method. A description of the numerical method is given as well as flow charts, program listings, input preparation, and output description.

The first computer program was written incorporating the $\rho_r \mu_r$ heat transfer prediction method developed by R. A. Hanks of The Boeing Company. The method was developed in the course of the X-20 program and finalized under NASA Research Contract No. NAS 8-11321 (Reference 1). In addition to the $\rho_r \mu_r$ method calculations, the program also contains subprograms for calculating pressures, gas properties, and streamline patterns for several special cases. The SRU 1108 version of the program was written during Boeing Company-funded research and converted to the IBM 7094 during this Air Force Flight Dynamics-funded study.

The program has four major sections. Program A contains the method per se. Given the external flow properties and wall condition, this section of the program computes laminar or turbulent heating rates. All required gas properties are computed by the program from the given pressure and edge velocity. The effect of transition is estimated by matching laminar and turbulent boundary layer momentum thicknesses at the transition point. The point of transition is determined by the program on the basis of a transition Reynolds number selected by the user. Program A can be applied to any geometry for which the external flow properties including nonisothermal wall effects can be defined by the user.

The second section, Program B, computes the local velocities and static temperatures along a streamline from a given pressure distribution.

If the streamline divergence is not known, Program C can be used to obtain this information. Program C calculates the path of two streamlines, the streamline of interest and one below, given a two-dimensional array of streamline angles. With two streamline paths known, the divergence parameter can be computed. This calculation, however, assumes zero divergence due to body geometry.

The fourth section of the computer program, called D, calculates or provides information for the three previous subprograms for four special cases; the axisymmetric and two-dimensional body, the hemisphere, the swept cylinder, and the sharp delta wing.
The second computer program, the turbulent nonsimilar boundary layer, integrates the boundary layer partial differential equations (described in Appendix C of Volume I) using finite difference methods. The program can calculate the turbulent boundary layer on unyawed sharp or blunt axisymmetric or two-dimensional bodies. Three-dimensional flow effects are calculated using the zero crossflow pressure gradient approximation which implies no rotation of the velocity vectors within the boundary layer. The program is also limited to attached flow in air, which is considered as ideal gas.

The program is capable of initiating its own boundary layer solutions, given only external flow properties, for the stagnation point of either sharp or blunt tip cones and plates.

Special inputs required are: pressure, wall enthalpy, a three-dimensional flow parameter and its derivative, the velocity gradient at the boundary layer edge and the normal velocity at the surface, all functions of the streamwise distance x. The user must also specify an initial and final value of x, the value of the x increments, and printout information.

Output includes streamwise and normal velocity, temperature, total and static enthalpy, shear, heat transfer rate, a similarity parameter, and x derivatives of velocity and total enthalpy. These values are tabulated as functions of y (normal to surface) at each output position. Also printed in the output are the shear and heat transfer rate at the wall, and the boundary layer displacement and momentum thickness.

The $Pr^1$ $Pr^2$ program requires about 0.5 to 1 minute of computer time per case, while the nonsimilar program requires 2 to 3 minutes for a flat plate case. The time estimate can vary and depends largely on the type of case.
THE LAMINAR-TURBULENT MOMENTUM INTEGRAL COMPUTER PROGRAM (P_r, u_r)

1. DISCUSSION

The computer program described in this report is based on the Hanks $P_r, u_r$ method of solution to the boundary layer momentum integral equation; and, as such, contains all of the assumptions pertaining to the method as discussed in References 1, 2, 3, and Volume I. The program calculates aerodynamic heating on arbitrary bodies, with or without streamwise pressure gradients in laminar or turbulent flow. Three-dimensional flow effects are included in the form of streamline divergence due to body geometry ($r$) and crossflow or transverse pressure gradients ($f$). The effect of a nonisothermal wall on aerodynamic heating is also calculated with modified methods of Lighthill and Seban. This is discussed in further detail in Reference 1. The program described herein is applicable only to air in chemical equilibrium (References 4-7).

a. Method of Solution

The $P_r, u_r$ integral program is actually four programs, any of which can be operated separately. The first program, Program A, calculates heating rates for any body shape at any flight or ground-facility test condition given the external flow properties, gas properties at the wall, gas properties at the stagnation condition, total streamline divergence and the streamline divergence due to body geometry alone.

Program A has an additional capability of calculating transition effects based on a transition Reynolds number selected by the user. The program matches laminar and turbulent boundary layer momentum thicknesses at the transition point.

In addition, the program calculates reference heat transfer coefficients which are used to normalize the calculated local heat transfer coefficients. The laminar heat transfer coefficients are all normalized by the value of the hemisphere stagnation-point heat transfer coefficient evaluated at the same free stream conditions as the body of interest. The heat transfer coefficient on the stagnation line of a 60° swept infinite cylinder is used to normalize all turbulent heat transfer coefficients. The hemisphere and cylinder radius are items of input.

Program B calculates the boundary edge velocity and temperature given the pressure distribution along the streamline or body.
Program C computes streamwise pressure and divergence parameters given a two-dimensional field of pressures and streamline angles. Since pressure data and streamline angles are often most easily available as spanwise plots at various stations, considerable crossplotting is required to provide specific values along the streamline passing over any specified point on the body. This work is performed by Program C, which begins at the point for which the user desires to calculate the heating rate, and traces out the streamline, proceeding upstream to a specified boundary (e.g., the boundary layer origin). The streamline angle is determined by double interpolation of a two-dimensional array of angles which is input. The streamline angle is then used to determine the coordinates of a point on the streamline a distance $dx$ upstream. This procedure is repeated until the upstream boundary is reached.

It also interpolates as required to obtain initial values at the upstream boundary of the input array $(\xi, \eta)$ thus providing all information required for Program B.

Program D provides the pressure and streamline arrays and initial values required by Programs A, B, and C for several specific bodies. These optional configurations are:

- **D-1** Arbitrary two-dimensional bodies at an angle of attack and arbitrary axisymmetric bodies at zero angle of attack. The bodies may be sharp or blunt and composed of wedges and plates or cones and cylinders.
- **D-2** Hemisphere
- **D-3** Swept infinite cylinder ($0 < \Lambda < 90^\circ$)
- **D-4** Sharp delta wing ($60^\circ < \Lambda < 80^\circ$ and $\alpha > 0$)
For options D-1 and D-2, the user provides a table of body width or radius at various locations along the axis. For options D-2 and D-3, the cylinder or hemisphere diameter must be input and for options D-3 and D-4, the cylinder or delta wing sweep angle must be provided. For option D-4 the angle of attack must be stated. In all options, the user provides the free stream conditions.

Options D-2 and D-3 (hemisphere and swept infinite cylinder) use pressure distributions stored within the program. These distributions are shown in Figure 1. Options D-1 and D-4 calculate pressures from the local flow deflection angle and a modified Newtonian pressure expression. For $\delta > 0$

$$\frac{C_p}{\sin^2 \delta} = 1.05 \cdot \left[ 1, 1025 - \frac{4}{(M \sin \delta)^2} \right]^{1/2} - \frac{1.278 \sin^2 \delta}{M^2_{0,6}}$$  (1)

and for $\delta \leq 0$

$$\frac{C_p}{\sin^2 \delta} = 0$$  (2)

This approximate analytical method was devised for predicting pressures from the Newtonian relationship in which $K$ is allowed to vary.

$$C_p = K \sin^2 \delta$$  (3)

The relationship for $K$ was chosen to obtain the best agreement with the exact solutions for a wedge, cone, and blunt body stagnation point. A plot of the modified pressure coefficients is shown in Figure 2.

The D options also provide velocity and streamline angle distributions along a streamline. In option D-1, the velocity is computed from

$$u_e = u_\infty \left[ 1 - \left( \frac{p}{p_\infty} - 1 \right) \frac{1}{\gamma M^2_{\infty}} \right]^{1/2} \text{ if } p_{\text{initial}} < 0.528 P_0$$  (4)

and

$$u_e = a_x \frac{x}{x_{\text{sonic point}}} \text{ if } p_{\text{initial}} > 0.528 P_0$$  (5)

$$x < x_{\text{sonic point}}$$
Figure 1: PRESSURE DISTRIBUTION ON A HEMISPHERE AND UNSWEEPED INFINITE CYLINDER (REF. 3)
Figure 2: BOEING MODIFIED NEWTONIAN HYPERSONIC PRESSURE COEFFICIENTS
and

\[ u_e = a^* \left\{ 6 \left[ 1 - \left( \frac{P}{P_0} \right)^{2/7} \right] \right\}^{1/2} \quad x > x_{\text{sonic point}} \]  

where \( a^* \) is the velocity at the sonic point.

Program options D-2 and D-3 obtain the edge velocity by integration of the pressure gradient along the streamline, while D-4 calculates \( u_e \) from

\[ u_e = u_\infty \left[ 1 - \frac{\delta^2}{5600} \right] \]  

The development of Equation (7) is presented in detail in Reference 3.

In addition, options D-3 and D-4 for the swept cylinder and delta wing also provide a two-dimensional array of streamline angles. The streamlines angles on the swept cylinder are obtained from

\[ \tan \theta_e = \frac{v_e}{u_\infty \sin \Lambda} \]  

where the spanwise velocity \( v_e \) is obtained by integration of the pressure gradient normal to the leading edge. Delta wing streamline angles are obtained from a correlation curve and

\[ \theta_e = (90 - \Lambda) \left( \frac{d \theta}{d \epsilon} \right)_{\text{CL}} \bar{R} + \left[ \theta_{\text{LE}} - (90 - \Lambda) \left( \frac{d \theta}{d \epsilon} \right)_{\text{CL}} \right] \bar{R}^9 \]  

where

\[ \bar{R} = \frac{\tan \left( \frac{\eta}{\xi} \right)}{\tan \left( \frac{90 - \Lambda}{57.3} \right)} \]

\[ \left( \frac{d \theta}{d \epsilon} \right)_{\text{CL}} \] is the gradient of the streamline angle with respect to the ray angle \( \epsilon \).

The streamline angle distribution across the wing span is a curve fit obtained by matching the boundary conditions given by the angle and the gradient at the centerline and the angle at the leading edge. The calculations of program D are necessarily somewhat approximate since exact flow field calculations are possible for only a few simple geometries. The approximations used in program D are those discussed in Reference 1 and are simple yet reasonably accurate.
Program D is easily modified and extended as more information becomes available, or as certain specific shapes become important.

b. \( \rho, \mu, \tau \) Program Schematic

The relation of the four programs is illustrated on the following page; however, it should be understood that all four programs are coupled and function together as a single program when desired. The symbol OR indicates that the required data can come from either of the indicated sources.

c. Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a^* )</td>
<td>sonic velocity</td>
</tr>
<tr>
<td>( C_P )</td>
<td>pressure coefficient</td>
</tr>
<tr>
<td>( f )</td>
<td>streamline divergence due to crossflow pressure gradients</td>
</tr>
<tr>
<td>( M_\infty )</td>
<td>free stream Mach number</td>
</tr>
<tr>
<td>( P )</td>
<td>local pressure</td>
</tr>
<tr>
<td>( P_0 )</td>
<td>stagnation point pressure</td>
</tr>
<tr>
<td>( P_{T_2} )</td>
<td>stagnation point or stagnation line pressure</td>
</tr>
<tr>
<td>( r )</td>
<td>streamline divergence due to body geometry</td>
</tr>
<tr>
<td>( u_e )</td>
<td>boundary layer edge velocity</td>
</tr>
<tr>
<td>( u_\infty )</td>
<td>free stream velocity</td>
</tr>
<tr>
<td>( v_e )</td>
<td>spanwise velocity at the boundary layer edge</td>
</tr>
<tr>
<td>( x )</td>
<td>distance along a streamline</td>
</tr>
<tr>
<td>( \delta )</td>
<td>local flow deflection angle measured from the free stream velocity vector</td>
</tr>
<tr>
<td>( \Delta )</td>
<td>total streamline divergence, includes body geometry and crossflow pressure gradients (rf)</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>ray angle on a delta wing</td>
</tr>
<tr>
<td>( \theta )</td>
<td>angular location</td>
</tr>
</tbody>
</table>
INPUT

- Geometry
- Test condition
- Wall temperature
- Transition Reynolds number
- Printout information

PROGRAM D

D-1 Axisymmetric or two-dimensional
D-2 Hemisphere
D-3 Swept infinite cylinder
D-4 Sharp delta wing

Output from Program D OR input to Program C

- Test condition
- Wall temperature
- Pressure
- Streamline angle
- Streamline divergence due to body geometry
- Transition Reynolds number

PROGRAM C

Output from Program C OR input to Program B

- Test condition
- Wall temperature
- Pressure
- Streamline angle
- Streamline divergence due to body geometry
- Streamline divergence due to crossflow pressure gradients
- Transition Reynolds number

PROGRAM B

Output from Program B OR input to Program A

- Wall temperature
- Pressure
- Streamline angle
- Streamline divergence due to body geometry
- Streamline divergence due to crossflow pressure gradients
- Transition Reynolds number
- Edge velocity
- Edge enthalpy
- Edge temperature
- Edge viscosity

PROGRAM A

Output from Program A

- Heat transfer coefficient
- Skin-friction coefficient
- Heating rate
- Shear stress
$\theta_e$  local streamline angle

$\Lambda$  leading edge sweep angle

$(\xi, \eta)$  coordinate system

$\rho_r \mu_r$  density-viscosity product evaluated at a reference condition

2. **PROGRAM DESCRIPTION**

Presented in the following section is a description of the program and subroutines. Equations, nomenclature input, and output from each subroutine are presented in alphabetical order by the subroutine title.

Before proceeding, a description of the program coordinate system is required. The program operates in a two-dimensional curvilinear coordinate system where the coordinate axes are designated as $\eta$ and $\xi$. Restrictions on this coordinate system are:

1. $\eta$ and $\xi$ are orthogonal.

2. Both $\eta$ and $\xi$ must be parallel to the surface over which the boundary layer flow is being considered, and

3. $\xi$ is taken to be in the general direction of the fluid flow.

Thus, no coordinate axis projects outward or normal from the surface area. In the case of a flat surface, $\eta$ and $\xi$ will lie in the same plane.

**CYLINDER**

![Cylinder Diagram](image)

$x$, streamline distance

**FLAT PLATE**

![Flat Plate Diagram](image)
a. $\rho, \mu$ Program Flow Chart

![Program Flow Chart Diagram]
The main program of the laminar-turbulent $\rho_r \mu_r$ momentum integral computer program — II (AS2419) — controls the input and the logic flow to the required mathematical subroutines. The input logic is arranged to minimize the modification required to change the input formats or the source of inputs (e.g., tape input or direct coupling to another program).

1) ATMOS (ALT, SONIC, PINF, RHO, TEMP)

Given the altitude, ATMOS calculates the free stream conditions. This subroutine is program AS1772, "Standard Atmosphere Properties, 1962".

**Input**

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>altitude, ft</td>
</tr>
</tbody>
</table>

**Output**

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SONIC</td>
<td>$a_\infty$ free-stream velocity of sound, ft/sec</td>
</tr>
<tr>
<td>PINF</td>
<td>$P_\infty$ free-stream pressure, lb/ft$^2$</td>
</tr>
<tr>
<td>RHO</td>
<td>$\rho_\infty$ free-stream density, slug/ft$^3$</td>
</tr>
<tr>
<td>TEMP</td>
<td>$T_\infty$ free-stream temperature, °R</td>
</tr>
</tbody>
</table>

2) AXI2D (Option D-1)

AXI2D calculates the flow conditions for arbitrary axisymmetric and two-dimensional geometries using the following procedure:

(a) With the geometry specified by the coordinates $(\xi_i, y_i)$, the slopes and pressures are assumed to occur midway between the input geometry points.

$$\delta_{i+1/2} = \tan \frac{y_{i+1} - y_i}{\xi_{i+1} - \xi_i}$$
Where $y$ is the local body coordinate:

and

$$x_{i+1/2} = x_{i-1/2} + \left[ (y_{i+1/2} - y_{i-1/2})^2 + (\xi_{i+1/2} - \xi_{i-1/2})^2 \right]$$

$$p_{i+1/2} = p_\infty \left[ \frac{\gamma}{2} M_\infty \frac{2}{\sin \delta} \sin^2 \delta + 1 \right]$$

where

$$\frac{C_P}{\sin^2 \delta} = 0 \text{ for } \delta \leq 0$$

$$\frac{C_P}{\sin^2 \delta} = 1.05 + \left[ 1.1025 + \frac{4}{M_\infty^2 \sin^2 \delta} \right]^{1/2} - \frac{1.278 \sin^2 \delta}{M_\infty^6}$$

for $\delta > 0$

The end points $x_l$ and $x_f$ and the pressures at the end points are found by linear extrapolation.

(b) If $P$ at either end point is less than $p_\infty$, set the respective $P = p_\infty$. If $P$ at either end point is greater than $p_0$, set the respective $P = p_0$.

(c) Initialize $x^* = x_l$ and calculate the flow conditions at each increment $dx$ along the streamline

$$x^*_i = x^*_i - dx \quad i \geq 2$$
Linear interpolation is performed to find \( P/P_{SL}(x^*) \) at each \( x^* \) using the tables of \( P \) and \( x \) calculated previously.

If \( P(x_1) \leq 0.528 P_0 \)

\[
u_e(x^*) = u_\infty \left[ 1 - \left( \frac{P}{P_\infty} - 1 \right) \frac{1}{\gamma M_\infty^2} \right]^{1/2}
\]

If \( P(x_1) > 0.528 P_0 \) find the sonic location \( X^* \) where \( P = 0.528 P_0 \) from interpolation of the \( x \) and \( P \) tables. Then

\[
u_e(x^*) = \frac{a^* x}{X^*} \quad \text{when} \quad x \leq X^*
\]

\[
u_e(x^*) = a^* \left\{ 6 \left[ 1 - \left( \frac{P}{P_0} \right)^{2/7} \right] \right\}^{1/2} \quad \text{when} \quad x > X^*
\]

\[a^* = \left( \frac{H}{3} \right)^{1/2} \quad \text{Sonic velocity}
\]

The remaining flow conditions are calculated from

\[
\begin{align*}
\frac{\Delta}{\Delta_i}(x^*) &= 1.0 \\
\frac{r}{r_i}(x^*) &= 1.0 \\
\frac{\Delta}{\Delta_i}(x^*) &= y(x^*) \\
\frac{r}{r_i}(x^*) &= y(x^*)
\end{align*}
\]

Two-dimensional body

Axisymmetric body

\[
i_e(x^*) = H - \frac{1}{2} u_e^2(x^*)
\]

\[
\omega_e(x^*) = f \left( \frac{P}{P_{SL}(x^*)}, i_e(x^*) \right)
\]

\[
ZT_e(x^*) = f \left( \frac{P}{P_{SL}(x^*)}, i_e(x^*) \right) \quad \text{[SOMEGA]}
\]
(d) Step (c) is repeated until the flow properties have been calculated for all points on the streamline.

**Input**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>total enthalpy, $\text{ft}^2/\text{sec}^2$</td>
</tr>
<tr>
<td>$\xi$</td>
<td>coordinate in free-stream flow direction, ft</td>
</tr>
<tr>
<td>$y(\xi)$</td>
<td>geometry coordinate orthogonal to $\xi$, ft</td>
</tr>
<tr>
<td>$x_I$</td>
<td>streamline coordinate at which calculations are to begin, ft</td>
</tr>
<tr>
<td>$dx$</td>
<td>streamline coordinate increment, ft</td>
</tr>
<tr>
<td>$P_{0}/P_{SL}$</td>
<td>stagnation pressure, atm</td>
</tr>
<tr>
<td>$M_{\infty}$</td>
<td>free-stream Mach number</td>
</tr>
<tr>
<td>$P_{\infty}$</td>
<td>free-stream pressure, $\text{lb}/\text{ft}^2$</td>
</tr>
</tbody>
</table>

**Output**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>number of points calculated along streamline</td>
</tr>
<tr>
<td>$x^*$</td>
<td>streamline coordinate, ft</td>
</tr>
<tr>
<td>$P_{0}/P_{SL}(x^*)$</td>
<td>local pressure along streamline, atm</td>
</tr>
<tr>
<td>$u_e$</td>
<td>edge velocity, $\text{ft}/\text{sec}$</td>
</tr>
<tr>
<td>$l_e(x^*)$</td>
<td>edge enthalpy, $\text{ft}^2/\text{sec}^2$</td>
</tr>
<tr>
<td>$(ZT)_e(x^*)$</td>
<td>edge compressibility-temperature product, $^\circ R$</td>
</tr>
<tr>
<td>$\omega_e(x^*)$</td>
<td>edge viscosity parameter, slug/ft-sec-$^\circ R$</td>
</tr>
<tr>
<td>$(r/r_I)(x^*)$</td>
<td>divergence parameter due to body-shock layer geometry</td>
</tr>
<tr>
<td>$(\Delta/\Delta_I)(x^*)$</td>
<td>distance between streamlines, or total streamline divergence</td>
</tr>
</tbody>
</table>

16
3) **BEQI**

BEQI calculates the equivalent distance parameters at the initial streamline coordinate $x_i$. Following is the sequence of operations:

(a) Initialize \[ \frac{b_{eq,L}}{x} = \frac{b_{eq,T}}{x} = 1.0 \]

(b) Divide the first streamline increment $dx$ into 10 intervals, $dx/10$. Interpolate to find the flow properties at each interval.

(c) Interpolate for \[ i_w = f(T_w, P/P_{SL}) \] [IWALL]

\[
\begin{align*}
ZT_S &= f(H, P/P_{SL}) \quad \text{[SOMEGA]} \\
\omega_S &= f(i_w, P/P_{SL}) \quad \text{[SOMEGA]}
\end{align*}
\]

(d) Calculate the reference terms \[ \omega_r, \rho_r, \mu_r, ZT_r, i_r, \sigma_r \] [RORMUR]

(e) Calculate \[ \frac{d(\Delta/\Delta_r)}{dx} \] [DERV]

(f) Calculate $i_{e,SL}$ and EXPK from

\[
N = \frac{x^* d(\Delta/\Delta_r)}{(\Delta/\Delta_r) dx}
\]

If $N \leq 0.05$; $0.99 \leq N \leq 1.01$; \[ \text{EXPK} = 0, \quad i_{e,SL} = i_e \]

\[
\begin{align*}
0.05 < N < 0.99; \quad \text{EXPK} &= -0.194 e^{-\frac{2}{3}N(N-1)} \quad , \quad \theta_{SL} = (N-1) \theta_e \\
N > 1.01; \quad \text{EXPK} &= 0.194 e^{-\frac{2}{3}(N-1)} \quad , \quad \theta_{SL} = \left(\frac{N-1}{N}\right) \theta_e
\end{align*}
\]
\[ v_p = u_e \cos \delta_{SL}; \quad i_e, SL = H - \frac{v_p^2}{2} \]

(g) Obtain \( \overline{E_L} \) and \( \overline{E_T} \) [EBAR]

(h) Calculate the equivalent distance parameter

\[ \frac{b_{eq}}{x} \bigg|_{x^*} = \frac{1}{x^* G(x^*)} \left[ x^* G(x^*) \left( \frac{b_{eq}}{x} \right) \right]_{x^*} + \int_{x^*}^{x^*} G(x) dx \]

where

\[ G_L = \left( \rho_r \mu_r \right) u_e \left( \frac{r/r_i}{r_i} \right)^2 \left( t/t_i \right)^2 \overline{E_L} x_i \]

\[ G_T = \left( \rho_r \mu_r \right) u_e \left( \frac{r/r_i}{r_i} \right)^{5/4} \left( t/t_i \right)^{(5/4)} \overline{E_T} \]

and

\[ f/t_i = \frac{\Delta/\Delta_i}{r/r_i} \]

The integration is performed by applying Simpson's rule to the integrands \( G_L \) and \( G_T \) to find \( b_{eq, L/x} \) and \( b_{eq, T/x} \), respectively.

(i) \( b_{eq}/x \) is calculated for 10 intervals. The 10th \( b_{eq}/x \) is used as the new \( (b_{eq}/x)_{x_i} \), the first interval is divided into 100 intervals, and the above calculations are repeated for the first 10 intervals.

(j) The calculations are repeated for 5 iterations (i) where

\[ \frac{dx_i}{10} = \left[ \begin{array}{c} \frac{b_{eq}}{x} \end{array} \right]_{x_i} = \left[ \begin{array}{c} \frac{b_{eq}}{x} \\ x_{ii} \end{array} \right]_{i-1} \]

The final \( \left[ \frac{(b_{eq}/x)_{x_{ii}}}{x_{ii}} \right]_{i=5} \) is then used as the \( (b_{eq}/x)_{x_{ii}} \) for the general calculation (QTRANS).
4) CYLIND (Option D-3)

CYLIND calculates the coordinates \( \xi \) and \( \eta \) and the two-dimensional parameters \( \theta_e(\xi, \eta), P/P_o(\xi, \eta) \) and \( T_w(\xi, \eta) \) for an infinite swept cylinder. The properties are calculated as a function of \( \eta \) at \( \xi = 0 \) and are assumed to be constant in the \( \xi \) direction.

Pressure:

\[
\frac{P(\eta)}{P_o} = \left[ \frac{P(\theta)}{P_T} \right] \times \left[ \frac{P_T}{P_o} \right]
\]

where

\[
\theta = 57.3 - \frac{\eta}{R_{CYL}}
\]

\( P/P_T(\theta) \) is found by linear interpolation on a built-in table (Figure 1).
\[ P_{T_2} = P_\infty \left(1.2 M_N\right)^{3.5} \left(\frac{6}{7 M_N^2 - 1}\right)^{2.5} \]

Stagnation Line Pressure

\[ M_N = M_\infty \cos \Lambda \]

Wall temperature is found by linear interpolation on the input array.

Streamline angle:

\[ \theta_\xi(\eta) = \tan^{-1} \frac{v(\eta)}{u_\infty \sin \Lambda} \]

where \( v(\eta) \) is the tangential velocity calculated in subroutine FLOW.

Then

\[ \frac{P}{P_0}(\xi, \eta) = \frac{P}{P_0}(\eta) \quad \text{for all } \xi \]

\[ T_w(\xi, \eta) = T_w(\eta) \quad \text{for all } \xi \]

\[ \theta_\xi(\xi, \eta) = \theta_\xi(\eta) \quad \text{for all } \xi \]

\[ (u_\xi)^2 = u_\infty^2 \sin \Lambda \]

Input

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADIUS</td>
<td>( R_{CYL} )</td>
</tr>
<tr>
<td>SWEEP</td>
<td>( \Lambda )</td>
</tr>
<tr>
<td>ACH</td>
<td>( M_\infty )</td>
</tr>
<tr>
<td>VEL</td>
<td>( u_\infty )</td>
</tr>
<tr>
<td>ETAF</td>
<td>( \eta_f )</td>
</tr>
<tr>
<td>XSIF</td>
<td>( \xi_f )</td>
</tr>
<tr>
<td>TW</td>
<td>( T_w )</td>
</tr>
</tbody>
</table>
PINF $P_\infty$ free stream pressure, lb/ft$^2$

POPSI. $P_o / P_{SI}$ stagnation pressure, atm

Output

XSI $\xi$ coordinate along axis of cylinder, ft

ETA $\eta$ coordinate tangential to cylinder, ft

M number of $\xi$ values

N number of $\eta$ values

ETAMAX $\eta_{max}(\xi)$ maximum $\eta$ value, ft

THisman $\theta_{max}(\xi)$ streamline angle at $\eta_{max}$, degrees

PRESSR $P/P_o(\xi, \eta)$ pressure ratio array

TWALL $T_w(\xi, \eta)$ wall temperature array, °R

THETAE $\theta_e(\xi, \eta)$ streamline angle array, degrees

5) DBTP

DBTP is a linear double interpolation routine. Search time is reduced by starting the current search at the search result for the previous entry to this routine. DBTP uses TBLP to perform the interpolation.

Input

XX independent variable table

YY independent variable table

ZZ dependent variable table

X independent variable

Y independent variable

NX number of values in X table

NY number of values in Y table
NXS  location in X table at which search is begun

NYS  location in Y table at which search is begun

Output

\( Z \)  dependent variable

6) DELTA (Option D-4)

DELTA calculates the body geometry, the streamline angles, and the two-dimensional pressure array \( P/P_0(\xi, \eta) \) for a sharp delta wing. The following procedure is used:

(a) If \( R < 0 \) a sharp delta wing case is indicated and \( \delta = \alpha \) (degrees) for all \( \xi \).

(b) The pressure is calculated from

\[
P(\xi, \eta) = \frac{\gamma}{2} P_\infty M_\infty^2 \left( \frac{C_p}{\sin^2 \delta} \right) \sin^2 \delta + P_\infty
\]

where

\[
\frac{C_p}{\sin^2 \delta} = 1.05 + \left[ 1.1025 + \frac{4}{M_\infty^2 \sin^2 \delta} \right]^{1/2} - \frac{1.276 \sin^2 \delta}{M_\infty^6}
\]

(c) The initial velocity is calculated from

\[
(u_\infty)_{x_1} = u_\infty \left[ 1 - \delta^2/5600 \right]
\]

(d) The equations used to calculate \( \theta_e(\xi, \eta) \) are:

\[
\mathcal{L} = \left[ \delta + 1.22 \left( \frac{\rho_\infty}{\rho_2} \right) \left( \frac{\tan \delta}{\tan \Lambda} \right)^{566} \right]^2 + \left[ \sin^{-1} \left( \frac{1}{M_\infty} \right) \right]^2 \right]^{1/2}
\]

\[
M_N = M_\infty \sin \mathcal{L}
\]

\[
\bar{\phi}^{**} = \frac{1}{90 - \Lambda} \tan^{-1} \left[ \frac{1}{6} \left( 1 + \frac{5}{M_\infty} \left( \frac{M_N^2 - 1}{M_N^2} \right) \right) \frac{1}{(u/u_\infty)^2} + \frac{1}{6} \right]^{1/2}
\]
\[
\gamma = \frac{5(\bar{\phi}**)^4}{5(\bar{\phi}**)^4 + 1}
\]

\[\theta^* = (90 - \Lambda)(1 + \bar{\phi}**)^\gamma\]

\(N_{CL}\) is found by linear interpolation on a built-in table (Figure 3).

If \(\bar{\phi}** \leq 1.2\)

\(N_{CL} = f(\bar{\phi}**)\)

If \(\bar{\phi}** > 1.2\)

\(N_{CL} = f(\Lambda, \bar{\phi}**)\)

\[
N_{CL} = \frac{1 + \bar{\phi}**}{M_{C}^2 + 1} N_{CL}
\]

\[
\eta_{max} = \xi \tan (90 - \Lambda)
\]

\[
\theta = (90 - \Lambda) N_{CL} \bar{R} + \left[\theta^* - (90 - \Lambda) N_{CL}\right] \bar{R}^9
\]

where

\[
\bar{R} = \frac{\tan (\eta/\xi)}{\tan (90 - \Lambda)}
\]

Then

\[
\theta_{max} = \theta^*
\]

\[
\xi_i = \xi_{i-1} + \Delta \xi
\]

\[
\eta_i = \eta_{i-1} + \Delta \eta
\]

(e) The delta wing option D-4 assumes zero streamline divergence due to body geometry

\[
\frac{r}{r'} = 1.0
\]
Figure 3: STREAMLINE CORRELATION FOR SHARP DELTA WINGS
<table>
<thead>
<tr>
<th>Input</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACH</td>
<td>$M_\infty$</td>
<td>free stream Mach number</td>
</tr>
<tr>
<td>PINF</td>
<td>$P_\infty$</td>
<td>free stream pressure, lb/ft²</td>
</tr>
<tr>
<td>POPSIL</td>
<td>$P_o/P_{SL}$</td>
<td>stagnation pressure, atm</td>
</tr>
<tr>
<td>VEL</td>
<td>$u_\infty$</td>
<td>free stream velocity, ft/sec</td>
</tr>
<tr>
<td>ALPHA</td>
<td>$\alpha$</td>
<td>angle of attack, degrees</td>
</tr>
<tr>
<td>SWEEP</td>
<td>$\Lambda$</td>
<td>sweep angle, degrees</td>
</tr>
<tr>
<td>RADIUS</td>
<td>$R$</td>
<td>leading edge radius, ft</td>
</tr>
<tr>
<td>XSII</td>
<td>$\xi_i$</td>
<td>initial streamline $\xi$ coordinate, ft</td>
</tr>
<tr>
<td>ETAI</td>
<td>$\eta_i$</td>
<td>initial streamline $\eta$ coordinate, ft</td>
</tr>
<tr>
<td>XSIF</td>
<td>$\xi_f$</td>
<td>final streamline $\xi$ coordinate, ft</td>
</tr>
<tr>
<td>ETAF</td>
<td>$\eta_f$</td>
<td>final streamline $\eta$ coordinate, ft</td>
</tr>
<tr>
<td>DELXSI</td>
<td>$\Delta \xi$</td>
<td>increment used to calculate $\xi$ coordinate, ft</td>
</tr>
<tr>
<td>DELETA</td>
<td>$\Delta \eta$</td>
<td>increment used to calculate $\eta$ coordinate, ft</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>$M$</td>
<td>number of $\xi$ values</td>
</tr>
<tr>
<td>N</td>
<td>$N$</td>
<td>number of $\eta$ values</td>
</tr>
<tr>
<td>XSI</td>
<td>$\xi$</td>
<td>geometry coordinate, ft</td>
</tr>
<tr>
<td>ETA</td>
<td>$\eta$</td>
<td>geometry coordinate, ft</td>
</tr>
<tr>
<td>ETAMAX</td>
<td>$\eta_{max}$</td>
<td>leading edge coordinate, ft</td>
</tr>
<tr>
<td>THSMX</td>
<td>$\theta_{max}$</td>
<td>streamline angle at leading edge, degrees</td>
</tr>
<tr>
<td>THETAE</td>
<td>$\theta_e$</td>
<td>streamline angle, degrees</td>
</tr>
<tr>
<td>PRESSR</td>
<td>$P/P_o$</td>
<td>pressure ratio</td>
</tr>
<tr>
<td>URATIO</td>
<td>$u/u_\infty$</td>
<td>velocity ratio</td>
</tr>
</tbody>
</table>
7) DERV (V, W, X, Y, Z, DBYDX, DX, ITYPE)

DERV calculates the derivative of a tabulated function \( f \) by Langrangian five-point formulas. Given a sequence of points \( x_{-2}, x_{-1}, x_0, x_1, x_2 \) separated by a common increment \( h \), the corresponding derivative of \( f \) at each point is given by:

\[
\begin{align*}
\frac{f'}{-2} &= \frac{1}{h} (-f_{-2} + f_{-1}) ; \\
\frac{f'}{-1} &= \frac{1}{2h} (-f_{-2} + f_{0}) ; \\
\frac{f'}{0} &= \frac{1}{2h} (-f_{-1} + f_{1}) ; \\
\frac{f'}{1} &= \frac{1}{2h} (-f_{0} + f_{2}) ; \\
\frac{f'}{2} &= \frac{1}{h} (-f_{1} + f_{2})
\end{align*}
\]

**Input**

| \( V \) | \( f_{-2} \) |
| \( W \) | \( f_{-1} \) |
| \( X \) | \( f_0 \) |
| \( Y \) | \( f_1 \) |
| \( Z \) | \( f_2 \) |

**ITYPE** determines which derivative formula is to be used

**DX** \( h \) function increment

**Output**

| DBYDX | \( f' \) | function derivative |

8) DIVERG

DIVERG calculates \( P/P_0, T_w, \theta_e, r/r_i \) and \( \Delta/\Delta_i \) along a given streamline by the following method:

(a) Store the original streamline (1) data and calculate an auxiliary streamline (2) such that

\[
\eta_{F_2} = \eta_{F_1} \pm \Delta \eta_{\text{max}}
\]
(b) Calculate the following at each $dx$ location on the streamline of interest

\[ \theta_e(x^*) \quad \text{linear interpolation on } \theta_e(x) \]

\[ T_w(x^*) \quad \text{double linear interpolation on } T_w(\xi, \eta) \]

\[ P/P_0(x^*) \quad \text{double linear interpolation on } P/P_0(\xi, \eta) \]

\[ \eta_{\text{i}1} = 1.0 \]

\[ \frac{\eta_{\text{i}1}(x^*) - \eta_{\text{i}2}(x^*)}{\eta_{\text{i}1}(x^*) - \eta_{\text{i}2}(x^*)} \frac{\cos \theta_e(x^*)}{\cos \theta_e(x^*)} \]

Where the subscripts 1, 2, and i indicate the original and auxiliary streamlines and the point at which calculations are begun, respectively. If the streamlines cross at any point, the above calculations are begun a distance $dx$ downstream from the intersection point.

Input:

| XI  | $x_1$ | initial streamline distance, ft |
| DX  | $dx$  | streamline coordinate increment, ft |
| ETA | $\eta_f$ | final streamline coordinate, ft |
| XSIF | $\xi_f$ | final streamline coordinate, ft |
| M  | | number of $\xi$ input values |
| N  | | number of $\eta$ input values |
| XSI | $\xi$ | streamline coordinate array, ft |
| ETA | $\eta$ | streamline coordinate array, ft |
| PRESSR | $P/P_0(\xi, \eta)$ | pressure ratio array |
| TWALL | $T_w(\xi, \eta)$ | wall temperature array, °R |
| XSIX | $\xi(x)$ | streamline coordinate, ft |
| ETAX | $\eta(x)$ | streamline coordinate, ft |
streamline coordinate, ft

\( \eta_{\text{max}} \) coordinate for geometry leading edge, ft

\( \theta_{e}(x) \) streamline angle, degrees

\( \Delta/\Delta_{i}(x^*) \) dimensionless distance between streamlines

\( (r/r_{i})(x^*) \) divergence parameter

\( T_{w} \) wall temperature, \(^{\circ}\)R

\( x_{f} \) final streamline x-coordinate, ft

II number of points on streamline

9) DSIRCH \((Z, X, Y, ZA, XA, YA, NZ, NX, NY, IX, IY)\)

DSIRCH uses DTAB to perform double interpolation on a table of the form

\[ Z = f(x, y) \]

**Input**

<table>
<thead>
<tr>
<th>( X )</th>
<th>independent variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td></td>
</tr>
<tr>
<td>( XA )</td>
<td>table of X values</td>
</tr>
<tr>
<td>( YA )</td>
<td>table of Y values</td>
</tr>
<tr>
<td>( ZA )</td>
<td>table of Z values</td>
</tr>
<tr>
<td>( NX )</td>
<td>number of values in X table</td>
</tr>
<tr>
<td>( NY )</td>
<td>number of values in Y table</td>
</tr>
<tr>
<td>( NZ )</td>
<td>number of values in Z table</td>
</tr>
</tbody>
</table>
order of interpolation in X direction

order of interpolation in Y direction

Output

dependent variable

K3AR calculates the crossflow pressure gradient parameters $E_L$ and $E_T$ and the pressure gradient correlation parameters $r_c$ and $r_o$ using the following method:

(a) Interpolate to find

$$Z_{T\ e, SL} = f\left(\frac{P}{P_{SL}}, e, SL\right) \quad [\text{SOMEGA}]$$

(b) Calculate the local pressure in lb/ft$^2$ and the $\rho \mu$ terms

$$P_r = \frac{P}{P_{SL}}$$

$$\rho_{T V} = \frac{P}{R \ \omega_s}$$

$$\rho_{e \mu_e} = \frac{P}{R \ \omega_e}$$

(c) Calculate the mean enthalpies

$$i_{avg} = \frac{1}{2} (i_w - i_{e, SL})$$

$$i_{m,c} = i_{avg} + 0.2 \ (H - i_{e, SL}) \ \sigma_r$$

(d) Interpolate to find

$$Z_{T\ m,c} = f\left(\frac{P}{P_{SL}}, i_{m,c}\right) \quad [\text{SOMEGA}]$$

$$Z_{T\ m,o} = f\left(\frac{P}{P_{SL}}, i_{avg}\right) \quad [\text{SOMEGA}]$$
(e) Compute:

\[ F_{\Sigma_c} = \frac{(\Sigma_c - 0.294) \sigma \cdot 355}{0.4018} \]

and

\[ F_{\Sigma_o} = \frac{(\Sigma_o - 0.294) \sigma \cdot 355}{0.4018} \]

\[ F_{K_c} = (2 \Sigma_c)^{\text{EXPK}} \]

\[ F_{K_o} = (2 \Sigma_o)^{\text{EXPK}} \]

where

\[ \Sigma_c = \frac{ZT_{m,c}}{ZT_{e,SL}} \]

\[ \Sigma_o = \frac{ZT_{m,o}}{ZT_{e,SL}} \]

(f) The pressure gradient correlation parameters are calculated from:

\[ \Gamma_c = 0.71764 F_{K_c} \left( \sqrt{1 + F_{\Sigma_c}} - 1 \right) \]

\[ \Gamma_o = 0.71764 F_{K_o} \left( \sqrt{1 + F_{\Sigma_o}} - 1 \right) \]

(g) Compute the crossflow pressure gradient parameters:

\[ \bar{E}_L = 1 + \Gamma_c \]

\[ \bar{E}_T = (1 + 0.76519 \Gamma_o) \left( \frac{1 + \Gamma_c}{1 + \Gamma_o} \right)^4 \]
Input

AYEW \( i_w \) \( \text{wall enthalpy, ft}^2/\text{sec}^2 \)

AYESL \( i_{e,SL} \) \( \text{edge enthalpy at flow line of symmetry, ft}^2/\text{sec}^2 \)

\( \Pi \) \( \Pi \) \( \text{total enthalpy, ft}^2/\text{sec}^2 \)

SIGR \( \sigma_r \) \( \text{Prandtl number} \)

POPSLX \( P/P_{SL} \) \( \text{local pressure along streamline, atm} \)

OMEGE \( \omega_e \) \( \text{edge viscosity parameter, slug/ft-sec-'R} \)

OMEGS \( \omega_s \) \( \text{stagnation viscosity parameter, slug/ft-sec-'R} \)

Output

GAMC \( \Gamma_c \) \( \text{pressure gradient correlation parameter} \)

GAMO \( \Gamma_o \) \( \text{pressure gradient correlation parameter} \)

EBARL \( \bar{F}_L \) \( \text{laminar crossflow pressure gradient parameter} \)

EBART \( \bar{F}_T \) \( \text{turbulent crossflow pressure gradient parameter} \)

FIND looks up \( Z \) in an array \( Z_A \), dimensioned \( NZ \), and stores the location of \( Z \) relative to \( Z_A(1) \) in \( NZ_S \). If \( Z \) is not in the range of \( Z_A \), the message "VARIABLE NOT IN RANGE OF TABLE" is printed, followed by the input values of \( Z \), \( Z_A(1) \), and \( Z_A(NZ) \), respectively. \( NZ_S \) is then set to 1 if \( Z > Z_A(1) \) and to \( NZ \) if \( Z < Z_A(NZ) \). Upon subsequent entries to FIND the search is begun at \( Z_A(NZ_S) \) where \( NZ_S \) is now the location of \( Z \) from the previous entry to FIND. For a normal exit \( NZ_S = i \) where \( Z_A_i = Z < Z_A_{i+1} \).

Input

\( Z_A \) array to be searched

\( Z \) variable for which location is desired
12) FLOW (Program B)

FLOW calculates the external flow parameters at the boundary layer edge as a function of streamline distance given the pressure and the stagnation conditions.

At the starting point on the streamline, edge enthalpy is obtained from:

\[ \left( \frac{u_e}{x_1} \right) = H - \frac{v^2}{2} \]

For the special case of the swept cylinder tangential velocity calculation

\[ \left( \frac{u_e}{x_1} \right) = H - \frac{\left( \frac{v^2}{x_1} \right)}{2} - \frac{\left( u_\infty \sin \Lambda \right)^2}{2} \]

Then

\[ ZT_e = f \left( i_e, \frac{P}{P_{SL}} \right) \quad \text{[SOMEGA]} \]

\[ \omega_e = f \left( i_e, \frac{P}{P_{SL}} \right) \quad \text{[SOMEGA]} \]

Let

\[ u = \frac{1}{2} \left( \frac{u_e}{u_\infty} \right)^2 \]

and define

\[ \left| u_{x^*} \right|_K = \left\{ \frac{1}{2} \left( \frac{u_e}{u_\infty} \right)^2 \right\}_{x^*, K} \]

as the Kth iteration for \( u \) at the point \( x^* \).
Similarly define \((ZT_e)_{x*}^K\) as the Kth iteration for \(ZT_e\) at \(x^*\).

The iterative equation to be satisfied at each point \(x^*\) on the streamline is

\[
[u_{x*}^K] = u_{x*} - \frac{\text{d}R}{u_\infty} \left[ (ZT_e)_{x*}^K \right] \frac{\left[ \frac{\text{d}}{\text{d}x} \left( \frac{P}{P_0} \right) \right]}{\text{avg}} \left( \frac{P}{P_0} \right)_{\text{avg}}
\]

where

\[
\frac{\text{d}(P/P_0)}{\text{d}x} = \frac{1}{2} \left[ \frac{\text{d}}{\text{d}x} \left( \frac{P}{P_0} \right) \right]_{x*} + \frac{1}{\text{d}x} \left( \frac{P}{P_0} \right)_{x* - \text{d}x}
\]

and

\[
\left( \frac{P}{P_0} \right)_{\text{avg}} = \frac{1}{2} \left[ \frac{P}{P_0} \right]_{x*} + \frac{P}{P_0} \left( x^* - \text{d}x \right)
\]

The derivatives are calculated numerically using subroutine DERV.

For each iteration:

\[
\left[ (i_e)_{x*}^K \right] = H - u_\infty^2 \left[ u_{x*}^K \right]
\]

For the swept cylinder tangential velocity

\[
\left[ (i_e)_{x*}^K \right] = H - u_\infty^2 \left[ u_{x*}^K \right] - \frac{1}{2} (u_\infty \sin \Lambda)^2
\]

Then

\[
\begin{bmatrix}
(ZT_e)_{x*}^K \\
(\omega_e)_{x*}^K
\end{bmatrix} =
\begin{bmatrix}
[i_e]_{x*}^K \\
\frac{P}{P_{\text{SL}}}(x^*)
\end{bmatrix} [\text{SOMEGA}]
\]
This value of $ZT_e$ is then used in the above iterative equation to yield a new $u_{x^*}$. These calculations are repeated at every $x^*$ until

\[
\left| \frac{u_{x^*}}{K} - \frac{u_{x^*}}{K-1} \right| < \epsilon
\]

where $\epsilon = 10^{-6}$. A maximum of ten iterations ($K = 10$) is fixed by the program. If the relative error criteria has not been satisfied after ten iterations at any point $x^*$, an appropriate error comment will be printed. If convergence at $x^*$ has been established in the $I$th step, then set $u_{x^*} = \{u_{x^*}\}_I$ and

\[
\begin{align*}
(u_e)_{x^*} &= u_\infty (2u_{x^*})^{1/2} \\
(i_e)_{x^*} &= (i_e)_{x^*} \quad I \\
(ZT_e)_{x^*} &= (ZT_e)_{x^*} \quad I \\
(\omega_e)_{x^*} &= (\omega_e)_{x^*} \quad I
\end{align*}
\]

A special case results when $(u_e)_{x^*} = 0$. This corresponds to a stagnation point where $x_1 = 0$ and $\left. \frac{d}{dx} \left( \frac{P}{P_0} \right) \right|_{x_1} = 0$. The iterative equation for $u_e$ becomes $u_e(x^*) = u_e(x^* - dx)$ or $du_e/dx = 0$. Because this result is incorrect, the first step is calculated using the following:

The velocity gradient at the stagnation point $x_1$ is calculated from:

\[
\left[ \frac{d}{dx} \frac{u_e}{u_\infty} \right]_{x_1} = \frac{1}{u_\infty} \left[ -R(ZT_e)_{x_1} \frac{d^2}{dx^2} \left( \frac{P}{P_0} \right)_{x_1} \right]^{1/2}
\]

where the pressure derivative is obtained from the numerical derivative equation

\[
\frac{d^2}{dx^2} \left( \frac{P}{P_0} \right) = \frac{p(\xi_1, \eta) - 2 \frac{p}{P_0}(\xi_2, \eta) + \frac{p}{P_0}(\xi_3, \eta)}{\Delta \xi^2}
\]

34
The use of this equation requires that the first three input points be equidistant. Then

\[
\left( \frac{u_e}{x_{l+dx}} \right)_{x_{l+dx}} = u_e \left[ \frac{d}{dx} \left( \frac{u_e}{u_{\infty}} \right) \right]_{x_{l}}
\]

\[
\left( \frac{i_e}{x_{l+dx}} \right)_{x_{l+dx}} = H - \frac{1}{2} \left( \frac{u_e}{x_{l+dx}} \right)^2
\]

\[
\left( \frac{ZT_e}{x_{l+dx}} \right)_{x_{l+dx}} = \int \left( \frac{i_e}{x_{l+dx}} \right)_{x_{l+dx}} \left( \frac{P}{P_{SI}} \right)_{x_{l+dx}} [\text{SOMEGA}]
\]

The calculations continue, using the previously described procedure where the starting point is now \( x_{l+dx} \).

**Input**
- PRESRNX \( P/P_0(x^*) \) ratio of local to stagnation pressure along streamline
- \( H \) number of values in pressure array
- URATIO \( u_e/u_{\infty} \) velocity ratio at \( x_l \)
- \( H \) \( H \) total enthalpy, ft\(^2\)/sec\(^2\)
- VEL \( u_{\infty} \) free stream velocity, ft/sec
- DX \( dx \) streamline coordinate increment, ft
- POPSL \( P_0/P_{SI} \) stagnation pressure, atm

**Output**

The following outputs are a function of streamline distance:
- UE \( u_e \) edge velocity, ft/sec
- POPSLX \( P/P_{SI}(x^*) \) local pressure along streamline, atm
HEMI calculates the streamline and the pressure and divergence parameters along the streamline for a hemisphere geometry.

(a) Set \( x_1^* = x_1 \)

(b) Calculate \( x^*, P/P_0(x^*), r/r_1(x^*), \) and \( \Delta/\Delta_1(x^*) \) for all \( dx \) increments along the streamline until \( \theta = 90^\circ \)

\[
\theta = \frac{57.3 \times x^*}{R_{HEMI}}
\]

Interpolate to find \( P/P_0(x^*) = f(\theta) \)

\[
r/r_1(x^*) = \sin \left( \frac{x^*}{R_{HEMI}} \right)
\]

\[
\Delta/\Delta_1(x^*) = \sin \left( \frac{x^*}{R_{HEMI}} \right)
\]

(c) \( URATIO = u_{ex_1}/u_\infty \)

**Input**

- \( RH = R_{HEMI} \) hemisphere radius, ft
- \( XI = x_1 \) x-value at which calculations are to begin, ft
- \( DX = dx \) streamline increment, ft

**Output**

- \( XSTAR = x^* \) streamline coordinate, ft
- \( PRESRX = P/P_0(x^*) \) ratio of local to stagnation pressure along streamline
RORI \( r/r_i(x^*) \) divergence parameter due to body-shock layer geometry

DODI \( \Delta/\Delta_i(x^*) \) distance between streamlines, total streamline divergence

14) INTIAL (I1, NTYPE)

INTIAL calculates the free stream conditions for wind tunnel and flight type inputs.

(a) Wind Tunnel Class I

\[
\begin{align*}
\mathbf{u}_\infty &= 49 \, M_\infty \sqrt{T_\infty} \\
H &= 6000 \, T_\infty + \frac{u_\infty^2}{2} \\
q_\infty &= 0.7 \, P_\infty M_\infty^2 \\
\rho_\infty &= P_\infty/(1716 \, T_\infty)
\end{align*}
\]

(b) Wind Tunnel Class II

\[
\begin{align*}
i_\infty &= \frac{u_\infty^2}{2} \\
a_\infty &= f(i_\infty) \quad \text{when } 5.93 \times 10^6 \leq i_\infty \leq 20.3 \times 10^6 \text{ ft}^2/\text{sec}^2 \quad [\text{SONENT}] \\
a_\infty &= 0.6325 \sqrt{i_\infty} \quad \text{when } i_\infty < 5.93 \times 10^6 \text{ ft}^2/\text{sec}^2 \\
M_\infty &= \frac{u_\infty}{a_\infty} \\
T_\infty &= \left( \frac{u_\infty}{49 \, M_\infty} \right)^2 \\
\rho_\infty &= P_\infty/(1716 \, T_\infty)
\end{align*}
\]
\[ P_0/P_\infty \text{ from [STACON]} \]

\[
P_\infty = \left( \frac{P_0}{P_{SL}} \right) P_{SL} \left( \frac{P_\infty}{P_0} \right) \]

\[ q_\infty = \frac{P_\infty u_\infty^2}{2RT_\infty} \]

(c) Flight

\[
a_\infty = f(\text{altitude}) \]
\[
P_\infty = f(\text{altitude}) \]
\[
T_\infty = f(\text{altitude}) \quad \text{[ATMOS]} \]

\[ H = 6000 T_\infty + \frac{u_\infty^2}{2} \]

\[ M_\infty = \frac{u_\infty}{a_\infty} \]

\[ q_\infty = 0.7 \ P_\infty \ M_\infty^2 \]

For wind tunnel I and flight

\[ P_0/P_\infty = f(M_\infty, H, P_\infty, u_\infty) \quad \text{[STACON]} \]

\[ P_0/P_{SL} = \left( \frac{P_0}{P_\infty} \right) \frac{P_\infty}{P_{SL}} \]

Input

Wind Tunnel Class I, NTYPE = WT1

ACH \quad M_\infty \quad \text{free stream Mach number}

TEMP \quad T_\infty \quad \text{free stream temperature,} \quad ^\circ \text{R}

PINF \quad P_\infty \quad \text{free stream pressure, lb/ft}^2
Wind Tunnel Class II, NTYPE = WT2

VEL \( u_\infty \) free stream velocity, ft/sec

H \( H \) total enthalpy, ft\(^2\)/sec\(^2\)

POPSL \( P_o/P_{SL} \) stagnation pressure, atm

PINF \( P_\infty \) free stream pressure, lb/ft\(^2\)

Flight, NTYPE = FLIGHT

ALT \( \text{flight altitude, } \theta \)

VEL \( u_\infty \) free stream velocity, ft/sec

Output

ACH \( M_\infty \) free stream Mach number

H \( H \) total enthalpy, ft\(^2\)/sec\(^2\)

PINF \( P_\infty \) free stream pressure, lb/ft\(^2\)

POPINF \( \rho_o/P_\infty \) stagnation to free stream pressure ratio

POPSL \( P_o/P_{SL} \) stagnation pressure, atm

QINF \( q_\infty \) free stream dynamic pressure, lb/ft\(^2\)

SONIC \( a_\infty \) speed of sound, ft/sec

VEL \( u_\infty \) free stream velocity, ft/sec

15) IWALL (TW, POPSIX, AYEW)

IWALL calculates the enthalpy at the wall as a function of the wall temperature and the local pressure.

(a) Set \( T_w^* = T_w/1.8 \) (conversion to °K)
(b) Calculate wall enthalpy

\[ i_w = .432 \frac{T_w}{K} \text{ when } T_w < 300 \text{K} \]

\[ i_w = .432 \frac{T_w}{K} + 3.82 \times 10^{-5} (T_w - 300)^2 \text{ when } 300 < T_w < 1800 \text{K} \]

\[ i_w = f(\log_{10} \frac{P}{P_{SL}}, T_w) \text{ [TABLE 6] when } T_w > 1800 \text{K} \]

(c) Convert \( i_w \) to \( \text{ft}^2/\text{sec}^2 \)

\[ i_w \left( \frac{\text{ft}}{\text{sec}^2} \right) = 25031 i_w \left( \frac{\text{Btu}}{\text{lb}} \right) \]

**Input**

TW \( T_w \) wall temperature, °R

POPSLX \( \frac{P}{P_{SL}}(x) \) local pressure along streamline, atm

**Output**

AYEW \( i_w \) wall enthalpy, \( \text{ft}^2/\text{sec}^2 \)

16) JAYELL (AYEW, H, AYEE, ZTE, ZTS, SIGR, BETAS, POPSLX, XJL)

JAYELL calculates the laminar pressure gradient effect correlation parameter \( J_L \) from the following:

(a) Calculate the mean enthalpy

\[ i_{m,s} = \frac{1}{2} (i_w + H) \]

(b) Interpolate to find:

\[ ZT_{m,s} = f(\frac{P}{P_{SL}}, i_{m,s}) \text{ [SOMEGA]} \]

(c) Calculate

\[ F \Sigma_s = \frac{(\Sigma_s - .294)}{.4018} \left( \frac{i_w}{H} \right) \sigma_r^3 \cdot 355 \]
where

\[
\Sigma_s = \frac{ZT_m, s}{ZT_s}
\]

\[
i_{aw} = H \left[ \frac{i_e}{H} + \sigma \frac{1}{2} \left( 1 - \frac{i_e}{H} \right) \right]
\]

(d) If \( \beta_s < 0 \) set \( j = -1 \)
If \( \beta_s \geq 0 \) set \( j = 1 \)

See subroutine QTRANS for \( \beta_s \) calculation.

(e) Calculate the streamwise pressure gradient function:

\[
F_{\beta_s} = \frac{(1 + 2 \, CPES) |\beta_s|}{2 \, CPES + (j + 1) \beta_s}
\]

where

\[
CPES = \frac{ZT_s}{ZT_e}
\]

(f) Calculate \( J_L \)

\[
J_L = \left[ 1 + .7176 \left( \sqrt{1 + F_{\beta_s} \Sigma_s} - 1 \right) \right]^j
\]

**Input**

- **AYEW** \( i_w \) wall enthalpy, ft\(^2\)/sec\(^2\)
- **H** \( H \) total enthalpy, ft\(^2\)/sec\(^2\)
- **AYEE** \( i_e \) edge enthalpy, ft\(^2\)/sec\(^2\)
- **ZTE** \( ZT_e \) edge compressibility-temperature product, °R
- **ZTS** \( ZT_s \) stagnation compressibility-temperature product, °R
SIGR $\sigma_r$ Prandtl number

BETAS $\beta_s$ streamwise pressure gradient parameter

POPSLX $P/P_{SL}(x^*)$ local pressure along streamline, atm

Output

XJL $J_L$ laminar pressure gradient effect correlation parameter

17) MUZERO (OMEGR, ZTR, AYER, AYEE, H, POPS, XMUO, XL)

MUZERO computes the reference stagnation viscosity $\mu_o$ and the diffusion effect parameter $\zeta$.

(a) Interpolate to find:

$$\frac{i_D}{i_e} = f(\log_{10} P/P_{SL}, i_e) \quad \text{[TABLE 7]}$$

If

$$i_D/i_e < 0, \text{ set } i_D/i_e = 0$$

(b) Calculate $\mu_o$ and $\zeta$

$$\mu_o = \omega_r (ZT_r)^{3/2} \left[ \frac{ZT_r + 200}{(H_r/ZT_r)^{200}} \right]$$

$$\zeta = 1 + 19 \left( \frac{i_D}{i_e} \right) \frac{i_e}{H}$$

Input

OMEGR $\omega_r$ reference viscosity-temperature ratio, slug/ft-sec-$^\circ$R

ZTR $ZT_r$ reference compressibility-temperature product, $^\circ$R

AYER $i_r$ reference enthalpy, ft$^2$/sec$^2$
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i_e$</td>
<td>edge enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>$H$</td>
<td>total enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>$P_{SL}(x^*)$</td>
<td>local pressure along streamline, atm</td>
</tr>
<tr>
<td>$\mu_0$</td>
<td>reference stagnation viscosity, slug/ft·sec</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>parameter indicating diffusion effect on heat transfer</td>
</tr>
</tbody>
</table>

18) **QTRANS (Program A)**

QTRANS calculates heat transfer data for any body shape given the external flow properties, gas properties at the wall, gas properties at the stagnation conditions, and the streamline divergence parameters. The sequence of calculations is as follows (names in brackets are subroutines):

(a) Set up $x$-printout array

(b) Initialize $\rho_s = 0.5$

(c) Interpolate for

$$ i_w = f(T_w, P/P_{SL}) \quad [IWALL] $$

(d) Interpolate to find

$$ \begin{align*}
ZT_s \\
\omega_s
\end{align*} = f(H, P/P_{SL}) \quad [SOMEGA] $$

$$ \begin{align*}
ZT_w \\
\omega_w
\end{align*} = f(i_w, P/P_{SL}) \quad [SOMEGA] $$

(e) Calculate

$$ \omega_r, \rho_r, \mu_r, ZT_r, i_r, \sigma_r \quad [RORMUR] $$
(f) Calculate
\[ \frac{du_e}{dx}, \quad \frac{d(\Delta / \Delta_i)}{dx} \quad \text{[DERV]} \]

(g) Calculate \( i_e, SL \) and \( EXPK \) from
\[ N = \frac{x^*}{\Delta} \frac{d(\Delta / \Delta_i)}{dx} \]
If \( N \leq .05 \); \( .99 \leq N \leq 1.01 \); \( EXPK = 0 \), \( i_e, SL = i_e \)
\[ .05 < N < .99 ; \quad EXPK = -.194e^{\frac{2}{3}N(N-1)}, \quad \bar{\theta}_SL = (N-1) \theta_e \]
\[ N > 1.01 ; \quad EXPK = .194e^{\frac{2}{3}(N-1)}, \quad \bar{\theta}_SL = \left( \frac{N-1}{N} \right) \theta_e \]

(h) Obtain \( \bar{E}_L \) and \( \bar{E}_T \) \quad \text{[EBAR]} \]

(i) Calculate the equivalent distance parameters defined by:
\[ \frac{b}{x} = \frac{1}{x^* G(x^*)} \left[ x_i G(x_i) \left( \frac{b}{x} \right) x_i + \int_{x_i}^{x^*} G(x)dx \right] \]
\[ = \frac{1}{x^* G(x^*)} \left[ x_i G(x_i) \left( \frac{b}{x} \right) x_i + \int_{x_i}^{x^*-2dx} G(x)dx + \int_{x^*-2dx}^{x^*} G(x)dx \right] \]
where the integrand \( G \) is calculated from
\[ G_L = (\rho_T \mu_T)u_e \left( \frac{r}{r_i} \right)^2 \left( \frac{f}{f_i} \right)^2 \bar{E}_L \]
\[ G_T = (\rho_T \mu_T)u_e \left( \frac{r}{r_i} \right)^{5/4} \left( \frac{f}{f_i} \right)^{(5/4)} \bar{E}_T \]
and
\[ \frac{f}{f_i} = \frac{\Delta / \Delta_i}{\tau \tau_i} \]
Thus, for $x^* = x_i + (2K)dx$ (INTEGER $x \geq 1$) the integration is performed by applying Simpson’s rule using $G_L$ or $G_T$ as the integrand to find $(b_{eq}/x)_L$ and $(b_{eq}/x)_T$, respectively.

\[
\left( \frac{b_{eq}}{x} \right)_{x^*} = \frac{1}{x^* G(x^*)} \left[ x_i G(x_i) \left( \frac{b_{eq}}{x} \right)_{x_i} + \int_{x_i}^{x^*-2dx} G(x)dx \right.
\]
\[
+ \frac{dx}{3} \left[ G(x^*-2dx) + 4G(x^*-dx) + G(x^*) \right] \right]
\]

where

\[
\int_{x_i}^{x^* - 2dx} G(x)dx = \frac{dx}{3} \sum_{j=1}^{K-1} \left[ G \left( x_i + (2j - 2)dx \right) + 4\left( x_i + (2j - 1)dx \right) \right. \\
+ G(x_i + 2jdx) \right]
\]

This last sum is the result of successive application of the three point integration as the calculations move along the streamline. When $x^* = x_i + (2K+1)dx$ an extra point is required to compute $(b_{eq}/x)$. The calculation is made by repeating the above sequence of operations (c) through (i) with $x^*$ incremented by $dx$.

(j) Calculate the streamwise pressure gradient parameter $\beta_s$ from the following:

\[
\beta_s(x^*) = \beta_s(x^* - 2dx) + F \frac{d\beta_s}{dx} \left( \frac{\beta_s}{x^* - 2dx} \right)
\]

where

\[
F_x = \frac{71/(70 + 1/c_x)}{G_L \left( \frac{b_{eq}}{x} \right)_{x^*-dx} \left[ G_L \left( \frac{b_{eq}}{x} \right) \right]_{x^*}}
\]

\[
c_x = 1 - \frac{\left[ G_L \left( \frac{b_{eq}}{x} \right) \right]_{x^*-dx}}{\left[ G_L \left( \frac{b_{eq}}{x} \right) \right]_{x^*}}
\]
and

\[
\left( \frac{d \mu}{x} \right)_{x^* - 2dx} = \left[ 2 \left( \frac{H}{i_0} \right) \left( \frac{b_{eq}}{x} \right)_L \right]_{x^* - 2dx} \quad u_e = 0
\]

\[
= \left[ 2 \left( \frac{H}{i_0} \right) \left( \frac{b_{eq}}{x} \right)_L \frac{du_e}{dx} \right]_{x^* - 2dx} \quad u_e \neq 0
\]

(k) Calculate the reference Reynolds number \( R_{r,Q} \): calculate \( J_L \) [JAYELL] and \( \mu_0, \mathcal{Z} \) [MUZERO]

then

\[
\left( \frac{S_{eq}}{x} \right)_L = \frac{(b_{eq}/x)_L}{J_L^2}; \quad \frac{F_{x,Q}}{x} = \left[ \frac{J_L^{17/20} \left( \frac{b_{eq}}{x} \right)_T}{x} \right]^{1/3}
\]

\[
\left( \frac{S_{eq}}{x} \right)_T = \frac{(b_{eq}/x)_L}{J_L^{9/16}}; \quad R_{r,Q} = \frac{\rho_r \mu_r u_e \left( \frac{x_{eq}}{x} \right)_{x^*}}{\mu_0^2 F_{x,Q}^2}
\]

\[
\left( \frac{x_{eq}}{x} \right)_L = \frac{(b_{eq}/x)_L}{J_L^{4/10}}
\]

If \( R_{r,Q} \geq R_{transition} \) (input) a new value of \( (b_{eq}/x)_T \) is calculated using the following equations:

\[
R_{L,S} = \frac{\rho_r \mu_r u_e S_{eq,L}}{\mu_0^2}
\]

\[
R_{T,S} = \frac{\rho_r \mu_r u_e S_{eq,T}}{\mu_0^2}
\]
\[ R_{\theta, L} = \frac{\mu_0}{\mu_e} \left[ 0.664 \left( R_{L, S} \right)^{1/2} \right] = \frac{\rho u e \theta L}{\mu_e} \]

\[ R_{\theta, T} = \frac{\mu_0 \left( 0.2135 R_{T, S} \right)}{\mu_e \left( \log_{10} R_{T, S} - 0.407 \right)^{2.64}} = \frac{\rho u e \theta T}{\mu_e} \]

\[ C_{\text{transition}} = \left[ 1 - \left( \frac{R_{\theta, L}}{R_{\theta, T}} \right)^{5/4} \right] \int_0^{x_{\text{transition}}} G_T dx \]

\[ \left( \frac{b_{\text{eq}}}{x} \right)_T = \left( \frac{b_{\text{eq}}}{x} \right)_T - \frac{C_{TR}}{G_{T,x^*}} \]

If \( R_{\theta, Q} < R_{\text{transition}} \) (input), a check is made at this point for \( x \)-printout values. If a printout point does not lie in the open interval \( (x^* - 2dx, x^* + 2dx) \), where \( x^* \) is the current value of \( x \), the sequence (c) through (k) is repeated until one is encountered.

(l) When a printout point lies in the above interval and when the transition point is found the desired heat transfer data is calculated using the following equations:

Calculate \( J_L \) and \( \mu_0 \) \( [\text{JAYELL}] \) and \( [\text{MUZERO}] \)

Then

\[ \left( \frac{b_{\text{eq}}}{x} \right)_L = \left( \frac{b_{\text{eq}}}{x} \right)_L - \frac{C_{TR}}{J_L^{4/10}} \]

\[ \left( \frac{b_{\text{eq}}}{x} \right)_L = \left( \frac{b_{\text{eq}}}{x} \right)_L - \frac{C_{TR}}{J_L^{4/10}} \]

47
\[
F_{x,S} = \left[ J_L \frac{17/4}{\left( \frac{b_{eq}}{x} \right)_T} \right]^{1/3} \\
F_{x,Q} = \left[ J_L \frac{17/20}{\left( \frac{b_{eq}}{x} \right)_T} \right]^{1/3}
\]

Reference Reynolds numbers:

\[
R_{r,Q} = \frac{\rho_R \mu_R \left( \frac{x}{x_{eq}} \right) x^*}{\mu_o^2 \left( \frac{x}{x_{eq}} \right)_L} \\
R_{r,S} = \frac{\rho_R \mu_R \left( \frac{x}{x_{eq}} \right) x^*}{\mu_o^2 \left( \frac{x}{x_{eq}} \right)_S}
\]

\[
\frac{\tau_T}{u_e} = \frac{\mu_o^2 F_{x,S} J_L^{3/2}}{(S_{eq}/x)_L} \left[ \frac{.185 R_{r,S}}{\log_{10} (R_{r,S} + 5000)^{2.584}} \right] \\
\frac{\tau_L}{u_e} = .332 J_L^{3/2} \frac{\mu_o^2 F_{x,S} (R_{r,S})}{(S_{eq}/x)_L}^{1/2}
\]

Skin friction coefficients:

\[
C_{f,e,L} = 2 \frac{\tau_L}{u_e} \frac{1}{\rho_o u_e} \\
C_{f,e,T} = 2 \frac{\tau_T}{u_e} \frac{1}{\rho_o u_e}
\]

where

\[
\rho_o = 1,232 \frac{P/P_{SL}}{ZT_e}
\]
Heat transfer coefficients based on enthalpy:

\[
H_L = \frac{0.332 g \mathcal{L} L ^{3/10}}{\sigma \cdot 0.645} \frac{\mu_o F_{x,Q} (R_{r,Q})^{1/2}}{x eq / x_L} \frac{L o x Q}{x^*}
\]

\[
H_T = \frac{g \mathcal{L} L ^{3/10}}{\sigma \cdot 0.645} \frac{\mu_o F_{x,Q}}{x eq / x_L} \frac{.185 R_{r,Q}}{\log_{10} (R_{r,Q} + 3000)} \frac{x^*}{2.584}
\]

\[
H_L \frac{H_L}{H_o}
\]

\[
H_T = \frac{H_T}{H_{ref,T}}
\]

Nonisothermal wall parameters:

\[
\phi_{local} = \left( i_{w_{local}} - i_{w_o} \right) + \sum_{i=1}^{n} \left( i_{w_i} - i_{w_{i-1}} \right) F_{S_i}
\]

where

\[
F_{S_i} = \left[ 1 - \left( S_L^{*} \right)^{3/4} \right]^{-1/3}
\]

laminar

\[
F_{S_i} = \left[ 1 - \left( S_T^{*} \right)^{9/10} \right]^{-1/9}
\]

turbulent

and

\[
\bar{S}_L^{*} = \frac{S_L - \frac{1}{3} (S_L - S_{L_{i-1}})}{(S_L)_{local}}; \quad S_L = \int_0^x G_L \, dx
\]

\[
\bar{S}_T^{*} = \frac{S_T - \frac{1}{3} (S_T - S_{T_{i-1}})}{(S_T)_{local}}; \quad S_T = \int_0^x G_T \, dx
\]
Adiabatic wall enthalpy:

\[ i_{aw, L} = i_e - \sigma_r^{1/2} (t_0 - t_e) \]

\[ i_{aw, T} = i_e + \sigma_r^{1/2} (t_0 - t_e) \]

Nonisothermal heat transfer rates:

\[ \dot{q}_L = -H_L \left( i_{aw, L} - i_w - \Phi_{L_{local}} \right) \]

\[ \dot{q}_T = -H_T \left( i_{aw, T} - i_w - \Phi_{T_{local}} \right) \]

Isothermal heat transfer rates:

\[ \dot{q}_{L, iso} = -H_L \left( i_{aw, L} - i_w \right) \]

\[ \dot{q}_{T, iso} = -H_T \left( i_{aw, T} - i_w \right) \]

(m) Print out heat transfer data.

In order for the sequence of calculations to reach this point, at least one printout point \( x_{PO} \) must lie in the interval \( (x^* - 2dx), (x^* + 2dx) \) where \( x^* = x_i + (2k)dx \) (INTEGER \( k \geq 0 \)) is the current value of \( x \). Answers of interest are now printed using linear interpolation for those printout points satisfying \( x^* - 2dx < x_{PO} < x^* \). If a printout point satisfies the criterion \( x^* < x_{PO} < (x^* + 2dx) \) then all quantities are saved for the current value of \( x \) to use for the interpolation scheme at \( x + dx \). Steps (a) through (m) are repeated until all printout points have been exhausted.

Input

Streamline Functions

<table>
<thead>
<tr>
<th>XSTAR</th>
<th>( x^* )</th>
<th>streamline coordinate at ( dx ) intervals, ft</th>
</tr>
</thead>
<tbody>
<tr>
<td>POPSLX</td>
<td>( P/P_{SL}(x^*) )</td>
<td>local pressure along streamline, atm</td>
</tr>
<tr>
<td>TW</td>
<td>( T_w(x^*) )</td>
<td>wall temperature, °R</td>
</tr>
<tr>
<td>AIEEE</td>
<td>( i_e(x^*) )</td>
<td>edge enthalpy, ( \text{ft}^2/\text{sec}^2 )</td>
</tr>
</tbody>
</table>
THETAS \( \theta_e(x^*) \) streamline edge angle, radians

DODI \( \Delta/\Delta_i(x^*) \) divergence parameters

RORI \( r/r_i(x^*) \) 

UE \( u_e(x^*) \) edge velocity, ft/sec

ZTEX \( ZT_e(x^*) \) edge compressibility-temperature product, *R

OMEGAE \( \omega_e(x^*) \) edge viscosity temperature ratio, slug/ft-sec-*R

Initial and Reference Conditions

H \( H \) total enthalpy, ft\(^2\)/sec\(^2\)

XI \( x_I \) initial streamline value, ft

RTRANS \( R_{\text{transition}} \) transition Reynolds number

SIGR \( \sigma_r \) partial Prandtl evaluated at \( ZT_r \)

DX \( dx \) streamline coordinate increment, ft

XPO \( x_{PO} \) printout array, ft

KF \( K_{PO} \) number of values in printout array

AYEWO \( i_w^0 \) reference wall enthalpy, ft\(^2\)/sec\(^2\) (needed only if \( T_w \neq \text{constant} \))

BEQXLI \( \left( \frac{b_{eq}}{x} \right)_{x_I} \) initial value of \( b_{eq}/x \) (laminar and turbulent)

Output

XSTAR \( x^* \) streamline coordinate, ft

POPSLX \( P/P_{SL_x} \) local pressure along streamline, atm

JE \( u_e(x^*) \) edge velocity, ft/sec

\( \Delta/\Delta_i \) \( \Delta/\Delta_{i}(x^*) \) distance between streamlines

51
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROHI</td>
<td>$r/r_i(x^*)$ divergence parameter due to body-shock layer geometry</td>
</tr>
<tr>
<td>TW</td>
<td>$T_w(x^*)$ wall temperature, °R</td>
</tr>
<tr>
<td>ZTEX</td>
<td>$ZT_e(x^*)$ edge compressibility-temperature product, °R</td>
</tr>
<tr>
<td>OMEGAF</td>
<td>$\omega_e(x^*)$ edge viscosity temperature ratio, slug/ft-sec-°R</td>
</tr>
<tr>
<td>AIEE</td>
<td>$i_e(x^*)$ edge enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>AYEW</td>
<td>$i_w(x^*)$ wall enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>BEQXL</td>
<td>$b_{eq, L}/x(x^*)$ ratio of laminar equivalent distance parameter to distance $x$ along streamline</td>
</tr>
<tr>
<td>BEQXT</td>
<td>$b_{eq, T}/x(x^*)$ ratio of turbulent equivalent distance parameter to distance $x$ along streamline</td>
</tr>
<tr>
<td>AYEAWL</td>
<td>$i_{aw, L}(x^*)$ laminar adiabatic wall enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>AYEAWT</td>
<td>$i_{aw, T}(x^*)$ turbulent adiabatic wall enthalpy, ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>HL</td>
<td>$H_L(x^*)/H_0$ laminar heat transfer coefficient ratio, $\dot{h}_m/ft^2/sec$</td>
</tr>
<tr>
<td>HT</td>
<td>$H_T(x^*)/H_{ref, T}$ turbulent heat transfer coefficient ratio, $\dot{h}_m/ft^2/sec$</td>
</tr>
<tr>
<td>QLISO</td>
<td>$\dot{q}_{iso, L}(x^*)$ laminar isothermal heat transfer rate, Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>QTISO</td>
<td>$\dot{q}_{iso, T}(x^*)$ turbulent isothermal heat transfer rate, Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>QDOTL</td>
<td>$\dot{q}_L(x^*)$ laminar heat transfer rate, Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>QDOTT</td>
<td>$\dot{q}_T(x^*)$ turbulent heat transfer rate, Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>CFEL</td>
<td>$C_f, e, L(x^*)$ laminar skin friction coefficient</td>
</tr>
<tr>
<td>CFET</td>
<td>$C_f, e, T(x^*)$ turbulent skin friction coefficient</td>
</tr>
<tr>
<td>RRQ</td>
<td>$R_{r, Q}$ heat transfer reference Reynolds number</td>
</tr>
<tr>
<td>RRS</td>
<td>$R_{r, S}$ skin friction reference Reynolds number</td>
</tr>
</tbody>
</table>
REF calculates the stagnation heat transfer coefficient based on a reference hemisphere of radius $R_0$, and the reference heat transfer coefficient based on a 60° infinite swept cylinder given the reference cylinder radius. The following procedure is used to calculate the stagnation conditions:

(a) Initialize $i_e = H$

(b) Obtain the following from the indicated subroutines:

\[
\begin{align*}
   i_w, ZT_w, ZT_e, ZT_s, \omega_w, \omega_e, \omega_s & \quad \text{[IWALL]} \\
   \omega_r, \rho_r \mu_r, ZT_r, i_r, \sigma_r & \quad \text{[RORMUR]} \\
   J_L & \quad \text{[JAYEL]} \\
   \mu_0, \mathcal{L} & \quad \text{[MUZERO]}
\end{align*}
\]

(c) Calculate the stagnation heat transfer coefficient and heat transfer rate:

\[
H_0 = \frac{0.664 g \mathcal{L}}{\sigma_r \cdot 645} \left[ \frac{\rho_r \mu_r u' I_{\text{MIN}}}{R_{\text{HEMI, ref}}} \right]^{1/2}
\]

\[
\dot{q}_0 = H_0 (H - i_{w_0})
\]

The reference conditions are calculated from the following:

(d) Reinitialize $M_N = 1/2 M_\infty$

(e) Obtain the following from the indicated subroutine:

\[
\begin{align*}
   P_{ST}/P_\infty, i_{ST}, u'_{MN} & \quad \text{[STACON]} \\
   i_w, \text{ref} & \quad \text{[IWALL]} \\
   ZT_w, ZT_e, ZT_s, \omega_w, \omega_e, \omega_s & \quad \text{[SOMEGA]}
\end{align*}
\]
(f) Calculate the following:

\[
\omega_r, \mu_r, \mu_{r'}, Z T_r', i_c, \sigma_r \quad \text{[ROMUR]}
\]

\[
\Gamma_c, \Gamma_0 \quad \text{[EBAR]}
\]

\[
\mu_o, \mathcal{L} \quad \text{[MUZER0]}
\]

\[
\frac{\omega_r}{\mu_r} \frac{\mu_{r'}}{Z T_r'} \frac{i_c}{\sigma_r} = \frac{\omega_r}{\mu_r} \frac{\mu_{r'}}{Z T_r'} \frac{i_c}{\sigma_r}
\]

\[
\Gamma_c, \Gamma_0
\]

\[
\mu_o, \mathcal{L}
\]

\[
\frac{\omega_r}{\mu_r} \frac{\mu_{r'}}{Z T_r'} \frac{i_c}{\sigma_r} = \frac{\omega_r}{\mu_r} \frac{\mu_{r'}}{Z T_r'} \frac{i_c}{\sigma_r}
\]

\[
\Gamma_c, \Gamma_0
\]

\[
\mu_o, \mathcal{L}
\]

\[
x_{eq, L} = \frac{0.866 R_{\text{CYL}, \text{ref}}}{(1 + \Gamma_c) u_{\text{MN}}}
\]

\[
F_{x, Q} = \left[ 1.6 \left( \frac{1 + \Gamma_0}{1 + 1.7625 \Gamma_0} \right) \right]^{1/3} \frac{1 + \Gamma_0}{1 + \Gamma_c}
\]

\[
R_{r, Q, \text{ref}} = \frac{0.866 R_{\text{CYL}, \text{ref}} u_{eq, L}}{x^{2/3} F_{x, Q}}
\]

\[
H_{\text{ref, L}} = \frac{0.332 g \mathcal{L} \mu_o F_{x, Q}}{x_{eq, L}^{1/2}} \left( R_{r, Q, \text{ref}} \right)^{1/2}
\]

\[
H_{\text{ref, T}} = \frac{0.185 g \mathcal{L} \mu_o F_{x, Q}}{x_{eq, L}^{1/2}} \left( R_{r, Q, \text{ref}} \right)^{1/2} \log_{10} \left( \frac{R_{r, Q, \text{ref}}}{1000} \right)^{2.584}
\]

\[
i_{aw, \text{ref, L}} = H - (1 - \sigma_r^{1/2})(H - i_e)
\]

\[
i_{aw, \text{ref, T}} = H - (1 - \sigma_r^{1/3})(H - i_e)
\]

\[
q_{\text{ref, L}} = H_{\text{ref, L}} (i_{aw, \text{L}} - i_w, \text{ref})
\]

\[
q_{\text{ref, T}} = H_{\text{ref, T}} (i_{aw, \text{T}} - i_w, \text{ref})
\]
\[
\tau_{\text{ref, } L} = \frac{.866 u_\infty \sigma_r}{g \xi} \theta_{\text{ref}, L}
\]

\[
\tau_{\text{ref, } T} = \frac{.866 u_\infty \sigma_r}{g \xi} \theta_{\text{ref}, T}
\]

\[
C_{f, c, \text{ref}} = \frac{\tau_{\text{ref}}}{q_\infty}
\]

**Input**

- PINF: \(P_\infty\) free stream pressure, lb/ft\(^2\)
- ACH: \(M_\infty\) free stream Mach number
- VEL: \(u_\infty\) free stream velocity, ft/sec
- VELP: \(u'_\text{MN}\) modified Newtonian velocity gradient, 1/sec
- H: \(H\) total enthalpy, ft\(^2\)/sec\(^2\)
- POPSL: \(P_o/P_{SL}\) stagnation pressure, atm
- TWREF: \(T_w, \text{ref}\) reference wall temperature, °R
- RHMREF: \(R_{\text{HEMI, ref}}\) reference hemisphere radius, ft
- RCLREF: \(R_{\text{CYL, ref}}\) reference cylinder radius, ft

**Output**

- AYEWO: \(i_{w_o}\) stagnation wall enthalpy, ft\(^2\)/sec\(^2\)
- HO: \(H_o\) stagnation heat transfer coefficient, lbm/ft\(^2\)-sec
- QDOTO: \(\dot{q}_o\) stagnation heat transfer rate, Btu/ft\(^2\)-sec
- HREFL: \(H_{\text{ref, L}}\) laminar reference heat transfer coefficient, lbm/ft\(^2\)-sec
- HREFT: \(H_{\text{ref, T}}\) turbulent reference heat transfer coefficient, lbm/ft\(^2\)-sec
AYAWRL \( i_{aw, \text{ref}, L} \) \text{laminar reference adiabatic wall enthalpy, ft}^2/\text{sec}^2

AYAWRT \( i_{aw, \text{ref}, T} \) \text{turbulent reference adiabatic wall enthalpy, ft}^2/\text{sec}^2

TAURL \( \tau_{\text{ref}, L} \) \text{laminar reference shear stress, lb/ft}^2

TAURT \( \tau_{\text{ref}, T} \) \text{turbulent reference shear stress, lb/ft}^2

QREFL \( \dot{q}_{\text{ref}, L} \) \text{laminar reference heat transfer rate, lb}_m/\text{ft}^2\text{-sec}

QREFT \( \dot{q}_{\text{ref}, T} \) \text{turbulent reference heat transfer rate, lb}_m/\text{ft}^2\text{-sec}

CFREFL \( C_{f, e, \text{ref}, L} \) \text{laminar reference skin friction coefficient}

CFREFT \( C_{f, e, \text{ref}, T} \) \text{turbulent reference skin friction coefficient}

AYEREF \( i_w, \text{ref} \) \text{reference wall enthalpy, ft}^2/\text{sec}^2

20) RORMUR

RORMUR calculates the following reference properties:

(a) Viscosity parameter

\[
K^* = \left( \frac{\omega_s}{\omega_w} \right)^{1/14} \left[ \frac{1,005}{0.005 + \left( \frac{\omega_s}{\omega_e} \right)^7} \right]^{1/14} \\
K = (K^*)^{13/16} \left[ \frac{13}{16} + \frac{3}{16} e^{-\left( K^* \frac{\omega_e}{\omega_w} \right)} \right]^{1/10}
\]

\[
F_1 = K \frac{\omega_e}{\omega_w}
\]

\[
\omega_r = \omega_w F_1^{7/8} \left( \frac{1.2}{2 + F_1^{5/2}} \right)^{1/10}
\]
(b) Density-viscosity product

\[ \varrho \mu = \frac{\varrho}{\varrho_{SL}} \frac{\mu_{r}}{\mu} \]

\[ \varrho \mu_{r} = \frac{\mu_{r}}{\mu_{r}/R} \]

(c) Compressibility-temperature product \( ZT_r \)

Change units of \( \omega_r \) to slug/ft-sec-°K

\[ \omega_{r, R} = 1.8 \omega_{r, K} \]

For \( 3.585 \times 10^{-10} \leq \omega_r \leq 8.03 \times 10^{-10} \)

\[ ZT_{r, K} = \frac{23213.13 \times 10^{-20}}{\omega_{r, K}^{2}} \left[ 1 + \left( 1 - 0.478656 \times 10^{18} \omega_{r, K}^{2} \right)^{1/2} \right]^{2} \]

Otherwise \( ZT_{r, K} \) is obtained from interpolation

\[ ZT_{r, K} = f(\log_{10} P, \omega_{r, K}) \] [TABL 19]

\( ZT_{r, K} \) is converted to °R by \( ZT_{r, R} = 1.8 ZT_{r, K} \)

(d) Enthalpy \( i_r \)

\[ \frac{i_r}{RT_o} = f(\log_{10} P, ZT_{r, K}) \] [TABL 20]

\[ i_r = 847559.8 \left( \frac{i_r}{RT_o} \right) \]

(e) Prandtl number \( \sigma_r \)

\[ \sigma_r = f(ZT_r) \] [TABL 14]
POPSIX p / p_{SL} \quad \text{local pressure along streamline, atm}

OMEGS \omega_s \quad \text{stagnation viscosity-temperature ratio, slug-ft-sec-}^h\text{R}

OMEGE \omega_e \quad \text{edge viscosity-temperature ratio, slug/ft-sec-}^h\text{R}

OMEGW \omega_w \quad \text{wall viscosity-temperature ratio, slug/ft-sec-}^h\text{R}

OMEGR \omega_r \quad \text{reference viscosity-temperature ratio, slug/ft-sec-}^h\text{R}

ROMUR \rho_r \mu_r \quad \text{reference density-viscosity product, slug}^2/ft^4\text{-sec}

ZTR \quad ZT_r \quad \text{reference compressibility-temperature product, }^h\text{R}

AYER i_r \quad \text{reference enthalpy, ft}^2/sec^2

SIGR \sigma_r \quad \text{reference Prandtl number}

21) \text{SIRCH (Y, X, YA, XA, N, IO)}

\text{SIRCH uses TAB to perform single interpolation on a table of } y = f(x).

\begin{array}{ll}
\text{Input} & \\
X & x \quad \text{independent variable} \\
YA & \quad \text{dependent array} \\
XA & \quad \text{independent array} \\
N & \quad \text{number of values in XA and YA arrays} \\
IO & \quad \text{order of interpolation} \\
\text{Output} & \\
Y & y \quad \text{dependent variable}
\end{array}
22) **OMEGA (ENTH, POPSIX, ZT, OMEGA)**

OMEGA calculates the viscosity-temperature ratio $\omega$ and the compressibility-temperature product $ZT$ as functions of enthalpy and pressure.

(a) Convert enthalpy to Btu/lbm

$$i(\text{Btu/lbm}) = 25301 \text{ ft}^2/\text{sec}^2$$

(b) Calculate $ZT$

For $i \leq 180$

$$ZT = i/432$$

For $i > 180$

$$ZT = f(\log_{10} P/\text{PSL}, i) \quad [\text{TABLE2}]$$

Calculate $\omega$

$$ZT \leq 111 \quad \omega = 14.454 \times 10^{-10}$$

$$111 < ZT \leq 2000 \quad \omega = \left(\frac{1.8 ZT}{200}\right)^{1/2} \left(\frac{14.454}{1.8 ZT + 200}\right) 14.45 \times 10^{-10}$$

$$ZT > 2000 \quad \omega = f(\log_{10} P/\text{PSL}, ZT) \quad [\text{TABL18}]$$

(c) Convert $ZT$ and $\omega$ to °R

$$ZT_{°R} = 1.8 ZT_{°K}$$

$$\frac{\omega_{\text{slug}}}{\text{ft-sec-°R}} = \frac{\omega}{1.8}$$

**Input**

ENTH $i$ enthalpy

POPSIX $P/\text{PSL}$ local pressure along streamline, atm
23) SONENT (ENTH, SONIC)

SONENT calculates the speed of sound as a function of free stream enthalpy. The result is obtained by linear interpolation on tabular arrays constructed from the equation:

\[ a_\infty = \left( \gamma R T_\infty \right)^{1/2} \]

where

\[ T_\infty = f(i_\infty / RT_o) \]
\[ \gamma_\infty = f(T_\infty) \]

Input
ENTH \[ i_\infty \] free stream enthalpy, ft$^2$/sec$^2$

Output
SONIC \[ a_\infty \] speed of sound, ft/sec

24) STACON (PSTPNF, AYEST, VELP, ACH, ACHN, H, PINF, VEL)

STACON calculates the stagnation-free stream pressure ratio, the stagnation enthalpy, and the modified Newtonian pressure gradient.

(a) Stagnation enthalpy

\[ i_{ST} = H - \frac{u_\infty^2}{2} \left[ 1 - \left( \frac{M_N}{M_\infty} \right)^2 \right] \]

(b) Pressure ratio \( P_0 / P_\infty \)

\[ \left( \frac{P_{ST}}{P_{SL}} \right)_o = \left( \frac{P_\infty}{P_{SL}} \right) \left( \frac{1,2 M_N^2}{7M_N^2-1} \right)^{6.5} \left( \frac{6}{7M_N^2-1} \right)^{2.5} M_N \geq 1 \]
\[
\left( p_{ST} \right)_o = \frac{p_{\infty}}{P_{SL}} \left( 1 + \frac{M_N^2}{5} \right)^{3.5} \quad M_N < 1
\]

\[
\frac{\rho_2}{\rho_1} = \frac{6 M_N^2}{M_N^2 + 5}
\]

Interpolate for

\[
(ZT_{ST})_o = f \left( p_{ST}, i_{ST} \right) \quad \text{[SOMEGA]}
\]

\[
\left( p_{ST} \right)_1 = \frac{p_{\infty}}{P_{SL}} + \left( p_{ST}_o - \frac{p_{\infty}}{P_{SL}} \right) \left[ 2 \left( \frac{P_2}{P_1} \right) \frac{6006 (ZT_{ST})_o}{i_{ST}} - 1 \right]
\]

Interpolate for

\[
(ZT_{ST})_1 = f \left( p_{ST}, i_{ST} \right) \quad \text{[SOMEGA]}
\]

\[
\left( p_{ST} \right)_2 = \frac{p_{\infty}}{P_{SL}} + \left( p_{ST}_o - \frac{p_{\infty}}{P_{SL}} \right) \left[ 2 \left( \frac{P_2}{P_1} \right) \frac{6006 ZT_{ST}_1}{2 (P_2/P_1) - 1} \right]
\]

\[
\frac{p_0}{p_{\infty}} = \left( \frac{p_{ST}_o}{P_{SL}} \right)
\]

(c) Modified Newtonian pressure gradient

\[
u_{MN}^i = \left[ \frac{2}{7} \left( 1 + \frac{5}{M_N^2} \right) \left( 1 - \frac{p_{\infty}}{p_0} \right) \left[ \frac{6006 (ZT_{ST})_1}{i_{ST}} \right] \right]^{1/2}
\]

Interpolate to find the empirical correction factor to the modified Newtonian velocity gradient as a function of \( M_\infty \cos \Lambda \) (Swept cylinder) or \( M_\infty \) (Hemisphere).

\[
u_{MN}^i = (\nu_{MN}^i) \cdot \text{ (CORRECTION FACTOR)}^5
\]

For the correction factor see Figure 4.
Figure 4: CORRECTION FACTOR FOR THE VELOCITY GRADIENT AT A HEMISPHERE STAGNATION POINT OR CYLINDER STAGNATION LINE.
The coordinates $(\xi_f, \eta_f)$ determine a point on a particular streamline. The streamline calculation begins at $(\xi_f, \eta_f)$ and proceeds upstream until the geometry boundary is reached. The calculation of the streamline is done by the following procedure:

Let $(\xi_X)_1 = \xi_t$, $(\eta_X)_1 = \eta_t$, and $(x)_1 = 0$. Linear interpolation is performed on $\theta_e(\xi, \eta)$ and $\theta_{\text{max}}(\xi)$ if necessary, to determine $(\theta_X)_1$. Then

$$\left( d\eta_X \right)_1 = dx \sin \left( \theta_X \right)_1$$

$$\left( d\xi_X \right)_1 = dx \cos \left( \theta_X \right)_1$$
Then the coordinates of the new point are:

\[
(\xi_2) = (\xi_1) - (\text{d}\xi_1)
\]

\[
(\eta_2) = (\eta_1) - (\text{d}\eta_1)
\]

\[
(x_2) = \text{d}x
\]

Continuing this procedure, the general equations are:

\[
(\theta_x)_{K-1} = f \left( (\xi_x)_{K-1}, (\eta_x)_{K-1} \right)
\]

\[
(\text{d}\xi_x)_{K-1} = \text{d}x \sin (\theta_x)_{K-1}
\]

\[
(\text{d}\eta_x)_{K-1} = \text{d}x \cos (\theta_x)_{K-1}
\]

The streamline calculation is completed by rearranging the \( \xi_x, \eta_x \) and \( x \) arrays such that \( \xi_x, \eta_x \) and \( x \) are measured from the upstream end of the streamline.

The subroutine STREAM assumes zero streamline divergence due to body geometry

\[
\frac{r}{r_i} = 1, 0
\]
TABLE2 obtains by table interpolation the compressibility-temperature product as a function of pressure and enthalpy.
Output

TS ZT compressibility-temperature product, °R
CHECK program check to determine success of interpolation (if the interpolation is not successful, an error statement is printed)

27) TABLE6 (TW, PPLOG, AYEW)

TABLE6 interpolates to find wall enthalpy as a function of pressure and wall temperature tables.

Input

TW T_w wall temperature, °K
PPLOG log_{10} \frac{P}{P_{SL}} log of ratio of pressure to sea level pressure, pressure in atm

Output

AYEW i_w wall enthalpy, Btu/lbm

28) TABLE7 (AYEE, PPLOG, AYEAYE)

TABLE7 interpolates on a table of pressure and edge enthalpy to find the dissociation on reaction enthalpy ratio \frac{i_D}{i_e}.

Input

AYEE i_e edge enthalpy, ft^2/sec^2
PPLOG log_{10} \frac{P}{P_{SL}} log of ratio of pressure to sea level pressure, pressure in atm

Output

AYEAYE \frac{i_D}{i_e} dissociation reaction enthalpy ratio

29) TABL14 (ZTW, SIGT)

TABL14 obtains by table interpolation the partial Prandtl number as a function of the compressibility-temperature product.
### Input

| ZTW | ZT | compressibility-temperature product, °R |

### Output

| SIGT | σ | Prandtl number |

#### 30) TABL18

TABL18 determines by table interpolation the viscosity-temperature ratio as a function of the pressure and compressibility-temperature product.

### Input

| ZT | compressibility-temperature product, °K |
| PPLOG | $\log_{10} \frac{P}{P_{SL}}$ | log of ratio of pressure to sea level pressure, pressure in atm |

### Output

| OMEGA | $\omega$ | viscosity-temperature ratio, slug/ft·sec·°K |
| CHECK | program check to determine success of interpolation (if interpolation is unsuccessful, an error comment is printed) |

#### 31) TABL19

TABL19 performs table interpolation to determine the compressibility-temperature product as a function of pressure and viscosity-temperature ratio. The table is composed of data in TABLE2 and TABL18.

### Input

| OMEGA | $\omega$ | viscosity-temperature ratio, slug/ft·sec·°K |
| PPLOG | $\log_{10} \frac{P}{P_{SL}}$ | log of ratio of pressure to sea level pressure, pressure in atm |

### Output

| ZT | ZT | compressibility-temperature product, °K |
32) TABL20

TABL20 performs table interpolation to find \( \frac{i}{RT_0} \) as a function of pressure and compressibility-temperature product.

**Input**

- **ZT** compressibility-temperature product, °K
- **PPLOG** \( \log_{10} \frac{P}{P_{SL}} \) log of ratio of pressure to sea level pressure, pressure in atm

**Output**

- **AYERTO** \( \frac{i}{RT_0} \) dimensionless enthalpy

33) TBLP (XT, YT, X, NTAB, N)

TBLP is a linear single interpolation routine. To save search time in finding the location of the independent \( x \) value in the \( x \) table, the search is begun at the previously located \( x \) value.

**Input**

- **XT** independent variable table
- **YT** dependent variable table
- **X** independent variable
- **NTAB** number of values in the \( x \) table
- **N** location in \( x \) table at which the search is begun. This value should be set = 1 upon the first entry to this routine. Thereafter, it stores the location of the previous search.
3. **INPUT-OUTPUT DESCRIPTION**

a. **Data Input Preparation**

Data are input to this program via punched cards. The purpose of this section is to define the required input and the form it takes on the cards. Input sheets for each option or section of the program are shown. Use of this type of form greatly reduces the amount of effort required to obtain results from the program. Data may be key punched directly from the forms.

1) **AXISYMMETRIC OR TWO-DIMENSIONAL BODIES (Option D-1)**

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-6</td>
<td>A6</td>
<td>Type of case: FLIGHT, WT1, or WT2. This card indicates the type of free-stream conditions to be input.</td>
</tr>
<tr>
<td>1</td>
<td>7-10</td>
<td>****</td>
<td>Leave blank.</td>
</tr>
<tr>
<td>1</td>
<td>11-72</td>
<td>10A6</td>
<td>Title or some description of case.</td>
</tr>
</tbody>
</table>

Card =2 will consist of only one of the following types of input: Flight, Wind Tunnel 1 (WT1), or Wind Tunnel 2 (WT2). Wind Tunnel 1 case should be used for tunnels operating in the ideal gas region, while Wind Tunnel 2 case should be used for high energy tunnels such as shock tubes.

**FLIGHT**

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10</td>
<td>Flight altitude. ft.</td>
</tr>
<tr>
<td>2</td>
<td>11-20</td>
<td>F10</td>
<td>Flight velocity. ft/sec.</td>
</tr>
</tbody>
</table>

**WT1**

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10</td>
<td>Free-stream Mach number.</td>
</tr>
<tr>
<td>2</td>
<td>11-20</td>
<td>F10</td>
<td>Free-stream static temperature. R.</td>
</tr>
<tr>
<td>2</td>
<td>21-30</td>
<td>F10</td>
<td>Free-stream static pressure. lb/ft².</td>
</tr>
</tbody>
</table>

**WT2**

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10</td>
<td>Free-stream velocity. ft/sec.</td>
</tr>
<tr>
<td>2</td>
<td>11-20</td>
<td>F10</td>
<td>Total enthalpy. ft²/sec².</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>2</td>
<td>21-30</td>
<td>F10</td>
<td>Ratio of stagnation pressure to sea level pressure.</td>
</tr>
<tr>
<td>2</td>
<td>41-50</td>
<td>F10</td>
<td>Free-stream dynamic pressure, lb/ft².</td>
</tr>
<tr>
<td>3</td>
<td>1-10</td>
<td>F10</td>
<td>Radius of 60° swept infinite cylinder for the turbulent reference heating rate, ft.</td>
</tr>
<tr>
<td>3</td>
<td>11-20</td>
<td>F10</td>
<td>Radius of hemisphere for the laminar reference heating rate, ft.</td>
</tr>
<tr>
<td>3</td>
<td>21-30</td>
<td>F10</td>
<td>Reference wall temperature, °R.</td>
</tr>
<tr>
<td>4</td>
<td>1-14</td>
<td>A6</td>
<td>Contains the word AXISYMMETRIC or TWODIMENSIONAL.</td>
</tr>
<tr>
<td>5</td>
<td>1-5</td>
<td>I5</td>
<td>M is the number of values in the geometry tables y(ξ) and ξ. 3 ≤ M ≤ 50.</td>
</tr>
<tr>
<td>6</td>
<td>----</td>
<td>8F10</td>
<td>This table contains M values of ξ, ft.</td>
</tr>
<tr>
<td>7</td>
<td>----</td>
<td>8F10</td>
<td>This table contains M values of y(ξ) corresponding to ξ in the preceding table, ft.</td>
</tr>
</tbody>
</table>

Note: A minimum of three points must be input into the preceding two tables. If the body is very slender, more points should be used.

<p>| 8    | 1-10    | F10    | Initial x location, point at which calculation begins, ft. |
| 8    | 11-20   | F10    | Increment in x, ft. dx &gt; (x_i - x_o)/750. |
| 8    | 21-30   | F10    | Edge velocity at x_i, ft/sec. |
| 9    | 1-5     | I5     | ITC is the number of values in the wall temperature versus surface distance tables. 2 ≤ ITC ≤ 50. If ITC ≤ 3 then the wall temperature must be nonisothermal. |</p>
<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>----</td>
<td>8F10</td>
<td>This table contains ITC values of surface distance, ft.</td>
</tr>
<tr>
<td>11</td>
<td>----</td>
<td>8F10</td>
<td>This table contains ITC values of wall or surface temperature corresponding to the preceding distance table. If $T_w$ = constant then enter two values.</td>
</tr>
<tr>
<td>12</td>
<td>1-5</td>
<td>I5</td>
<td>$K_{PO}$ is the number of printout locations desired. $K_{PO} \leq 500$.</td>
</tr>
<tr>
<td>12</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of $R_{Tr,Q}$ in Program A, labeled QTRANS in section 2. b. 18.</td>
</tr>
<tr>
<td>13</td>
<td>----</td>
<td>8F10</td>
<td>This table contains $K_{PO}$ locations at which printout is desired.</td>
</tr>
<tr>
<td>TYPE</td>
<td>TITLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MACH NO.</th>
<th>STATIC TEMP.</th>
<th>STATIC PRESS.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VELOCITY</th>
<th>TOTAL ENTHALPY</th>
<th>P₀/P Sl</th>
<th>STATIC PRESS.</th>
<th>DYNAMIC PRESS.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALTITUDE</th>
<th>VELOCITY</th>
<th>FLIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>r₁Cyl</th>
<th>r₂HEMI</th>
<th>T_w</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NPROG</th>
<th>CROSS OUT ONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AXISYMMETRIC</th>
<th>TWO DIMENSIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ξ TABLE (x VALUES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>η (ξ) TABLE (m VALUES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

|                     |
### HEMISPHERE CALCULATION (Option D-2)

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>A6</td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10</td>
<td>Contains title of program - HEMISPHERE.</td>
</tr>
<tr>
<td>5</td>
<td>1-10</td>
<td>F10</td>
<td>Hemisphere radius, ft.</td>
</tr>
<tr>
<td>6</td>
<td>1-5</td>
<td>I5</td>
<td>ITC is the number of points in the $T_w$ versus $x$ tables. $2 \leq ITC \leq 50$. If $ITC \geq 3$ then the wall temperature must be nonisothermal.</td>
</tr>
<tr>
<td>7</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of $ITC$ values of $x$ where $T_w$ are input, ft.</td>
</tr>
<tr>
<td>8</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of $ITC$ values of $T_w$ which correspond to locations given in the preceding $x$ table, or if $T_w = \text{constant}$, enter two values of $T_w$.</td>
</tr>
<tr>
<td>9</td>
<td>1-5</td>
<td>I5</td>
<td>$K_{PO}$ is the number of printout locations desired. $K_{PO} &lt; 500$.</td>
</tr>
<tr>
<td>9</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of $R_{Tr,Q}$ in Program A, labeled QTRANS in section 2. b. 18.</td>
</tr>
<tr>
<td>10</td>
<td>----</td>
<td>8F10</td>
<td>This table contains $K_{PO}$ surface distance locations around the hemisphere at which printout is desired.</td>
</tr>
</tbody>
</table>
### 3) SWEEP INFINITE CYLINDER (Option D-3)

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>This card contains the word CYLINDER to allow the program to select the D-3 options.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>Cylinder radius, ft.</td>
</tr>
<tr>
<td>5</td>
<td>1-10</td>
<td>F10</td>
<td>Cylinder sweep angle, measured from a line normal to the flow, $0 &lt; \Lambda &lt; 90$.</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10</td>
<td>$\xi_f$ is the abscissa at which the streamline calculation begins, ft. $\xi_f$ must be large enough so that the streamline has very nearly zero slope at the stagnation line. Cases have been run by the authors using $\xi_f = 10$ and 20 ft for $\Lambda = 10^\circ$ and 60$^\circ$.</td>
</tr>
<tr>
<td>6</td>
<td>11-20</td>
<td>F10</td>
<td>$\eta_f$ is the ordinate at which the streamline calculation begins, ft.</td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10</td>
<td>$dx$ is the increment in distance along the streamline used in the boundary layer calculations (Program A), ft. $dx \geq (\xi_f^2 + \eta_f^2)^{1/2}/600$.</td>
</tr>
<tr>
<td>7</td>
<td>1-5</td>
<td>I5</td>
<td>ITC is the number of values in $T_W$ and $\eta$ tables. $2 \leq$ ITC $\leq 50$. If ITC $\geq 5$ the wall temperature is nonisothermal.</td>
</tr>
<tr>
<td>8</td>
<td>-----</td>
<td>8F10</td>
<td>This table contains ITC values of $\eta$, where $\eta$ is the distance around the cylinder, ft.</td>
</tr>
<tr>
<td>9</td>
<td>-----</td>
<td>8F10</td>
<td>This table contains ITC values of $T_W$ corresponding to the $\eta$ table, °R.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>10</td>
<td>1-5</td>
<td>I5</td>
<td>( k_{P0} ) is the number of ( x ) printouts desired along the streamline. ( k_{P0} &lt; 500 ).</td>
</tr>
<tr>
<td>10</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of ( R_{TQ} ) in Program A, labeled QTRANS in section 2. b. 18.</td>
</tr>
</tbody>
</table>
| 10   | 21-30   | F10    | \( \Delta \eta_f \) is an increment used in Program C to locate the starting location of second streamline, ft.  
If \( \Delta \eta_f \) is small, in some cases the resolution between streamlines near the streamlines origin or apex may be very poor. The poor resolution leads to heat transfer distributions that are not smooth. In the event this occurs \( \Delta \eta_f \) should be increased, ft. |
| 11   | ----    | 8F10   | \( x_{P0} \) is a table containing locations along the streamline at which printout is desired. |
### η TABLE (ITC VALUES)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

### T_{w} TABLE (ITC VALUES)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

### K_{P0} TABLE

<table>
<thead>
<tr>
<th>K_{P0}</th>
<th>R_{f, transition}</th>
<th>Δ\eta_{f}</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### x TABLE (K_{P0} TABLE)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cards</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-6</td>
<td>A-6</td>
<td>Contains the word DELTA.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1-5</td>
<td>I5</td>
<td>ITC is an indicator for wall temperature gradient. If the wall is at constant temperature input a 1 and one value of wall temperature in table.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10</td>
<td>Angle of attack, degrees.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11-20</td>
<td>F10</td>
<td>Wing sweep angle measured from a line normal to the flow, degrees.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10</td>
<td>$\xi_f$ is the final or end coordinate along the streamline at which heating data is desired, ft.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>31-40</td>
<td>F10</td>
<td>$\eta_f$ is the final or end coordinate along the streamline at which heating data is desired, ft.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>41-50</td>
<td>F10</td>
<td>$x_1$ is the point along the streamline at which the calculations begin and may have to be greater than zero as indicated in Figure 5. The resulting change in streamline curvature, due to interpolation, near the apex produces a discontinuity in the heating calculations, ft.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1-10</td>
<td>F10</td>
<td>$\Delta \xi$ is the increment in $\xi$ which the program uses to setup an $(\xi, \eta)$ array of streamline angles, velocity, and pressures, ft.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$\Delta \xi \geq \frac{\xi_f}{50}$$
Figure 5: STREAMLINES NEAR THE APEX OF A SHARP DELTA WING
<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
| 7    | 11-20   | F10    | \( \Delta \eta \) is the increment in \( \eta \) the program uses to setup an \((\xi, \eta)\) array of streamline angles, velocity and pressures, ft. \[
\Delta \eta \geq \frac{\eta_f}{50}
\]
| 7    | 21-30   | F10    | \( dx \) is the increment in distance along the streamline at which calculations will be performed, ft. \[
dx > \frac{(x_f - x_1)}{750}.
\]

Note: If \( T_w \) is a constant value, enter one value into card 11, and delete cards 8, 9, and 10.

<p>| 8    | 1-5     | I5     | ( M ) is the number of points in the ( \xi ) array. ( 2 \leq M \leq 50 ). |
| 8    | 6-10    | I5     | ( N ) is the number of points in the ( \eta ) array. ( 2 \leq N \leq 50 ). |
| 9    | ----    | 8F10   | This is a table of ( M ) values of ( \xi ) where wall temperature will be input if ( ITC &gt; 1 ). |
| 10   | ----    | 8F10   | This is a table of ( N ) values of ( \eta ) where wall temperature will be input if ( ITC &gt; 1 ). |
| 11   | ----    | 8F10   | This is a table containing ((N) \cdot (M)) values of wall temperature, input so ( \xi ) varies with each ( \eta ). |
| 12   | 1-5     | I5     | ( K_{P0} ) is the number of ( x ) printouts desired along the streamline. ( K_{P0} &lt; 500 ). |
| 12   | 11-20   | F10    | Transition Reynolds number, see definition of ( R_f, Q ) in Program A, labeled QTRANS in section 2.b.18. |
| 12   | 21-30   | F10    | See card 10, swept infinite cylinder, option D-3. |
| 13   | ----    | 8F10   | ( X_{P0} ) is a table of ( K_{P0} ) locations along the streamline where printout is desired. |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \xi )</td>
<td>( \Delta \eta )</td>
<td>( dx )</td>
<td>( \xi ) TABLE (M VALUES)</td>
<td>( \eta ) TABLE (IN VALUES)</td>
<td>( \tau_{mn} ) TABLE (M ( \times ) N VALUES)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5) **STREAMLINE CALCULATIONS (PROGRAM C)**

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Must contain the word STREAM.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-6</td>
<td>A6</td>
<td>M is the number of points in the array $\xi$. $2 \leq M \leq 50$.</td>
</tr>
<tr>
<td>5</td>
<td>1-5</td>
<td>I5</td>
<td>N is the number of points in the array $\eta$. $2 \leq N \leq 50$.</td>
</tr>
<tr>
<td>5</td>
<td>6-10</td>
<td>I5</td>
<td>ITC is an indicator used to test whether the wall temperature is a constant or a variable. If $T^\infty$ is a constant $ITC = 1$ and enter one value of $T^\infty$ into card 16 and delete cards 13, 14, and 15.</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10</td>
<td>$\xi_f$ is $\xi$ at the final or end point, ft.</td>
</tr>
<tr>
<td>6</td>
<td>11-20</td>
<td>F10</td>
<td>$\eta_f$ is $\eta$ at the final or end point, ft.</td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10</td>
<td>$dx$ is the increment in distance along the streamline, ft. $dx \geq \left(\frac{\xi_f^2 + \eta_f^2}{2}\right)^{1/2} / 750$</td>
</tr>
<tr>
<td>6</td>
<td>31-40</td>
<td>F10</td>
<td>$x_f$ is the point along the streamline at which the calculations will begin, ft.</td>
</tr>
<tr>
<td>6</td>
<td>41-50</td>
<td>F10</td>
<td>$u_e_x_f$ is the edge velocity at the initial or start of calculations, ft/sec.</td>
</tr>
<tr>
<td>7</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of $M$ values of $\xi$. $2 \leq N \leq 50$. $\theta_e$, $P/P_0$ and $T_w$ will be input as functions of $\xi$ and $\eta$, ft.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 8    | ----    | 8F10   | This is a table of N values of $\eta$
|      |         |        | $2 \leq M \leq 50$. $\theta_0$, $P/P_0$ and $T_w$ will be
|      |         |        | input as functions of $\xi$ and $\eta$, ft. |
| 9    | ----    | 8F10   | This is a table of M values of $\eta_{\text{max}}$ as a
|      |         |        | function of $\xi$. $\eta_{\text{max}}$ defines the upper or
|      |         |        | outer boundary of the body, ft. |
| 10   | ----    | 8F10   | This is a table of M values of the streamline angle $\theta_{\text{max}}$
|      |         |        | along the boundary as a function of $\xi$, deg. |
| 11   | ----    | 8F10   | This is a table of $(M) \cdot (N)$ values of the local streamline angle $\theta_e$
|      |         |        | and should be input along lines of constant $\eta$ and $\xi$, deg. |
| 12   | ----    | 8F10   | This is a table of $(M) \cdot (N)$ values the local pressure divided by the stagnation
|      |         |        | pressure and should be input along lines of constant $\eta$ and varying $\xi$. |
| 13   | 1-5     | I5     | MT is the number of $\xi$ values entered into
|      |         |        | the table $T_w = f(\xi, \eta)$. Also see ITC above. |
| 13   | 6-10    | I5     | NT is the number of $\eta$ values entered into
|      |         |        | the table $T_w = f(\xi, \eta)$. Also see ITC above. |
| 14   | ----    | 8F10   | This is a table of MT values of $\xi$ at which $T_w$ will be input. Also see ITC above. |
| 15   | ----    | 8F10   | This is a table of NT values of $\eta$ at which $T_w$ will be input. Also see ITC above. |
| 16   | ----    | 8F10   | This is a table of $(MT) \cdot (NT)$ values of $T_w$ as a function of $\xi$ and $\eta$, °R. $T_w$
|      |         |        | should be input as a function of $\xi$ at constant $\eta$. |
| 17   | 1-5     | I5     | $K_{PO}$ is the number of $x$ locations along
<p>|      |         |        | the streamline at which printout is desired. $K_{PO} &lt; 500$. |</p>
<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of $R_e, Q$ in Program A, labeled QTRANS in section 2.b. 18.</td>
</tr>
<tr>
<td>17</td>
<td>21-30</td>
<td>F10</td>
<td>See card 10. Swept infinite cylinder, option D-3.</td>
</tr>
<tr>
<td>18</td>
<td>1-10</td>
<td>F10</td>
<td>$x_{p(i)}$ is a table of $K_{p(i)}$ locations along the streamline at which printout is desired.</td>
</tr>
<tr>
<td>TYPE</td>
<td>TITLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WT1</th>
<th>WT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{\infty}$</td>
<td>$T_{\infty}$</td>
</tr>
<tr>
<td>$u_{\infty}$</td>
<td>$\rho_{\infty}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALTITUDE</th>
<th>FLIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{CYL}$</td>
<td>$R_{HEMI}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NPROG</th>
<th>STREAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N$</th>
<th>$ITC$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

| $\xi_1$ | $\eta_1$ | $d_x$ | $x_1$ | $u_x x_1$ |

<table>
<thead>
<tr>
<th>$\xi$ TABLE (M VALUES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\eta$ TABLE (N VALUES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
6) REFERENCE CALCULATIONS (PROGRAM REF)

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>A6</td>
<td>Must contain the word REFERENCE.</td>
</tr>
</tbody>
</table>
7) BOUNDARY LAYER CALCULATIONS (PROGRAM B)

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Must contain the word FLOW.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-6</td>
<td>A6</td>
<td>IS is the number of values in the ( P/P_0 ), ( \theta_e ), ( \Delta/\Delta_l ) and ( r/r_l ) tables as a function of streamline distance. ( 2 \leq IS \leq 50 ).</td>
</tr>
<tr>
<td>5</td>
<td>1-5</td>
<td>IS</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>--------</td>
<td>8F10</td>
<td>This is a table containing IS number of x locations, ft, along the streamline at which ( P/P_0 ), ( \theta_e ), ( \Delta/\Delta_l ) and ( r/r_l ) will be input.</td>
</tr>
<tr>
<td>7</td>
<td>--------</td>
<td>8F10</td>
<td>This is a table containing IS values of local pressure each normalized by the model or vehicle stagnation pressure. The ( P/P_0 ) correspond to the above x table.</td>
</tr>
<tr>
<td>8</td>
<td>--------</td>
<td>8F10</td>
<td>This table contains IS values of the local streamline angle relative to the axis of symmetry, degrees. The values of ( \theta_e ) correspond to the above x table, degrees.</td>
</tr>
<tr>
<td>9</td>
<td>--------</td>
<td>8F10</td>
<td>This table contains IS values of the streamline divergence parameter ( \Delta/\Delta_l ). The values of ( \Delta/\Delta_l ) correspond to the above x table.</td>
</tr>
<tr>
<td>10</td>
<td>--------</td>
<td>8F10</td>
<td>This table contains IS values of the body shock geometry parameter ( r/r_l ). The values of ( r/r_l ) correspond to the above x table.</td>
</tr>
<tr>
<td>11</td>
<td>1-10</td>
<td>F10</td>
<td>( x_1 ) is point at which the calculation will begin, ft.</td>
</tr>
<tr>
<td>Card</td>
<td>Column</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>11</td>
<td>11-20</td>
<td>F10</td>
<td>(dx) is the increment in distance along the streamline, ft &lt;br&gt;[dx &gt; \frac{x_{FINAL} - x_I}{750}]</td>
</tr>
<tr>
<td>11</td>
<td>21-30</td>
<td>F10</td>
<td>(u_{e,x}) is the edge velocity at (x) ft/sec.</td>
</tr>
<tr>
<td>12</td>
<td>1-5</td>
<td>I5</td>
<td>ITC is the number of values in the (T_w) versus (x) table. (2 \leq ITC \leq 50). () If (T_w = ) constant then (ITC = 2).</td>
</tr>
<tr>
<td>13</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of (x) locations (ITC) along the streamline where (T_w) will be input.</td>
</tr>
<tr>
<td>14</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of ITC values of (T_w) along the streamline corresponding to the (x) table. If ITC = 2 enter only two values of (T_w).</td>
</tr>
<tr>
<td>15</td>
<td>1-5</td>
<td>I5</td>
<td>(Kp_1) is the number of (x) locations along the streamline at which printout is desired. (Kp_1 &lt; 500).</td>
</tr>
<tr>
<td>16</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of (R_{r,Q}) in Program A. labeled QTRANS in section 2. b. 18.</td>
</tr>
<tr>
<td>16</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of (Kp_1) locations along the streamline where printout is desired.</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>21</td>
<td>31</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(x_1)</td>
<td>(dx)</td>
<td>(u_{x_1})</td>
</tr>
<tr>
<td>12</td>
<td>ITC</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(K_{PO})</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Heat Transfer Calculations (Program A)

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Same as Option D-1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Must contain the word QTRANS.</td>
</tr>
<tr>
<td>3</td>
<td>1-6</td>
<td>A6</td>
<td>IS is the number of values in each of the following tables: x, P/P&lt;sub&gt;SL&lt;/sub&gt;, u&lt;sub&gt;e&lt;/sub&gt;, ( \theta_e ), ( \Delta / \Delta_i ) and ( r / r_i ). 2 ≤ IS ≤ 50.</td>
</tr>
<tr>
<td>4</td>
<td>1-5</td>
<td>IS</td>
<td>X&lt;sub&gt;j&lt;/sub&gt; is the initial x location along the streamline at which the calculations will start, ft.</td>
</tr>
<tr>
<td>5</td>
<td>11-20</td>
<td>F10</td>
<td>dx is the increment in distance along the streamline. (ft.) ( dx &gt; \frac{x_{FINAL} - x_1}{750} )</td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10</td>
<td>u&lt;sub&gt;e&lt;/sub&gt; is the boundary layer edge velocity at x&lt;sub&gt;j&lt;/sub&gt;, ft/sec.</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>8F10</td>
<td>This is a table containing x locations (IS) along the streamline at which P/P&lt;sub&gt;SL&lt;/sub&gt;, u&lt;sub&gt;e&lt;/sub&gt;, ( \theta_e ), ( \Delta / \Delta_i ) and ( r / r_i ) will be specified, ft.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8F10</td>
<td>This is a table containing IS values of the local pressure normalized with the sea level pressure, and at locations corresponding to the x table.</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>8F10</td>
<td>This is a table containing IS values of the boundary layer edge velocity corresponding to locations specified in the x table, ft/sec.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>8F10</td>
<td>This is a table containing IS values of the local streamline angle ( \theta_c ) relative to the axis of symmetry. The angles correspond to x locations specified in the x table, deg.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>11</td>
<td>----</td>
<td>8F10</td>
<td>This is a table containing IS values of the streamline divergence parameter ( \Delta / \Delta_i ) and correspond to ( x ) locations specified in the ( x ) table.</td>
</tr>
<tr>
<td>12</td>
<td>----</td>
<td>8F10</td>
<td>This is a table containing IS values of the body-shock parameter ( r/r_i ) and correspond to ( x ) locations specified in the ( x ) table.</td>
</tr>
<tr>
<td>13</td>
<td>1-5</td>
<td>I5</td>
<td>ITC is the number of values in the ( T_w ) versus ( x ) table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[ 2 \leq ITC \leq 50 ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If ( T_w ) is a constant, ( ITC = 2 ). Input two values of ( T_w ) into card 15.</td>
</tr>
<tr>
<td>14</td>
<td>----</td>
<td>8F10</td>
<td>This is a table containing ITC values of ( x ) along the streamline, ft.</td>
</tr>
<tr>
<td>15</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of ITC values of ( T_w ) along the streamline and corresponding to the ( x ) locations in the preceding table. (°R)</td>
</tr>
<tr>
<td>16</td>
<td>1-5</td>
<td>I5</td>
<td>( K_pO ) is the number of ( x ) printout locations desired along the streamline. ( K_pO &lt; 500 ).</td>
</tr>
<tr>
<td>16</td>
<td>11-20</td>
<td>F10</td>
<td>Transition Reynolds number, see definition of ( Rr_Q ) in Program A, labeled QTRANS in section 2. b. 18.</td>
</tr>
<tr>
<td>16</td>
<td>21-30</td>
<td>F10</td>
<td>BEQL is the ratio of the laminar equivalent distance parameter to the distance along the streamline. If ( x_I &gt; 0 ) then a value of BEQL is required. If ( x_I = 0 ) input BEQL=0.</td>
</tr>
<tr>
<td>16</td>
<td>31-40</td>
<td>F10</td>
<td>BEQT is the turbulent equivalent distance parameter. See BEQL above.</td>
</tr>
<tr>
<td>17</td>
<td>----</td>
<td>8F10</td>
<td>This is a table of ( K_pO ) locations along the streamline where printout is desired.</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>S</td>
<td>ITC</td>
<td>x TABLE (ITC VALUES)</td>
<td>x TABLE (ITC VALUES)</td>
</tr>
<tr>
<td>Kₚₒ</td>
<td>Sₜ</td>
<td>Pₜ,transition</td>
<td>(bₑ₋₁/ₚ)ₗ</td>
</tr>
</tbody>
</table>
b. Output Description

**INITIAL CONDITIONS**

**VELOCITY**
free stream velocity, ft/sec

**MACH**
free stream Mach number

**ENTHALPY**
free stream enthalpy, ft$^2$/sec$^2$

**TINF**
free stream static temperature, °R

**PINF**
free stream static pressure, lb/ft$^2$

**QINF**
free stream dynamic pressure, lb/ft$^2$

**PO/PSL**
(model total pressure)/(sea level static pressure)

**PO/PINF**
(model total pressure)/(free stream static pressure)

**REFERENCE CONDITIONS**

**HO**
\[ \frac{\dot{q}}{i_{aw} - i_w} = h_0/C_p \] where \( h_0 \) is the hemisphere stagnation point heat transfer coefficient, lb$\text{m}/\text{ft}^2\cdot\text{sec}$

\[ .24HO \]
\[ .24 h_0/C_p \text{, Btu/ft}^2\cdot\text{sec}^{-\circ F} \]

**QDOTO**
hemisphere stagnation point heating rate, Btu/ft$^2\cdot$sec

**HREF, L**
\[ \frac{\dot{q}_{REF, L}}{i_{aw, L} - i_w, REF} = h_{REF, L}/C_p \] where \( h_{REF, L} \) is the laminar heat transfer coefficient based on enthalpy at the stagnation line of a 60° swept infinite cylinder, lb$\text{m}/\text{ft}^2\cdot\text{sec}$

**RHEM, REF**
hemisphere radius, ft

**RCYL, REF**
cylinder radius, ft

**SWEEEP**
sweep angle of the swept cylinder used for the turbulent reference condition, degrees

\[ .24HREF, L \]
\[ .24 h_{REF, L}/C_p \]

**QDOT, REF, L**
laminar heating rate at the stagnation line of a 60° swept infinite cylinder, Btu/ft$^2\cdot$sec

104
\( \frac{q_{\text{REF, T}}}{(I_{\text{aw, T}} - I_{w, \text{REF}})} = \frac{h_{\text{REF, T}}}{C_p} \)

where \( h_{\text{REF, T}} \) is the turbulent heat transfer coefficient at the stagnation line of a 60° swept infinite cylinder based on enthalpy, lb\text{m}/ft\text{2}-sec

\( .24 h_{\text{REF, T}} \)

\( Q_{\text{DOT, REF, T}} \)

turbulent heating rate at the stagnation line of a 60° swept infinite cylinder, Btu/ft\text{2}-sec-R

\( I_{\text{AW, REF, L}} \)

laminar adiabatic wall enthalpy on a 60° swept infinite cylinder, ft\text{2}/sec\text{2}

\( I_{\text{AW, REF, T}} \)

turbulent adiabatic wall enthalpy on a 60° swept infinite cylinder, ft\text{2}/sec\text{2}

\( I_{w, \text{REF}} \)

reference wall enthalpy, ft\text{2}/sec\text{2}

\( T_{\text{AUREF, L}} \)

laminar shear stress on a 60° swept infinite cylinder, lb/ft\text{2}

\( C_{\text{INF, REF, L}} \)

\( 2 \tau_{\text{ref, L}}/\left( \rho_{\infty} u_{\infty}^2 \right) \). laminar skin friction coefficient based on free stream conditions

\( I_{w, O} \)

stagnation wall enthalpy, ft\text{2}/sec\text{2}

\( T_{\text{AUREF, T}} \)

turbulent shear stress on a 60° swept infinite cylinder, lb/ft\text{2}

\( C_{\text{INF, REF, T}} \)

\( 2 \tau_{\text{ref, T}}/\left( \rho_{\infty} u_{\infty}^2 \right) \). turbulent skin friction coefficient based on free stream conditions

\( T_{w, \text{REF}} \)

wall temperature for 60° swept infinite cylinder heating calculations, °R

**GEOMETRIC PARAMETERS**

| SWEEP CYL | cylinder sweep angle, degrees |
| SWEEP DELTA WING | delta wing sweep angle, degrees |
| ALPHA | angle of attack, degrees |
| RCYL | cylinder radius, ft |
| RHEMI | hemisphere radius, ft |
STREAMLINE

XSI coordinate, ft
ETA coordinate, ft
THETAE streamline angle, degrees

BOUNDARY LAYER CALCULATIONS

X streamline distance, ft
ETA ordinate of coordinate system, ft
XSI abscissa of coordinate system, ft
BEQXL \( \left( \frac{b_{eq}}{x} \right)_L \), laminar equivalent distance parameter
BEQXT \( \left( \frac{b_{eq}}{x} \right)_T \), turbulent equivalent distance parameter
POPSL local pressure in atmospheres
AYEAWL \( i_{aw,L} \), adiabatic wall enthalpy in laminar flow, ft\(^2\)/sec\(^2\)
AYEAWT \( i_{aw,T} \), adiabatic wall enthalpy in turbulent flow, ft\(^2\)/sec\(^2\)
UE edge velocity, ft/sec
HL laminar heat transfer coefficient divided by the reference hemisphere stagnation point heat transfer coefficient
HT turbulent heat transfer coefficient divided by the reference turbulent heat transfer coefficient at the stagnation line of a 60° swept infinite cylinder
DODI \( \Delta / \Delta_1 \), total streamline divergence
N \( x / (\Delta / \Delta_1) \ [d(\Delta / \Delta_1)/dx] \), streamline divergence parameter
QLISO laminar isothermal heat transfer rate, Btu/ft\(^2\)-sec-°R
QTISO turbulent isothermal heat transfer rate, Btu/ft\(^2\)-sec-°R
RORI \( r / r_1 \), streamline divergence due to body-shock geometry
QDOTL  laminar heating rate, Btu/ft^2-sece R
QDOTT  turbulent heating rate, Btu/ft^2-sece R
TW  wall temperature.  R
CFEL  laminar skin friction coefficient
CFET  turbulent skin friction coefficient
ZTEX  ZT_e, compressibility factor times the edge temperature, R
RRQ  heat transfer reference Reynolds number
RRS  skin friction reference Reynolds number
OMEGAE  edge viscosity divided by ZT_e, lb_f-sece/ft^2 R
AIEE  e_i, edge enthalpy, ft^2/sec^2
AYEW  w_i, wall enthalpy, ft^2/sec^2
THETA  e_0, local streamline angle, degrees
MUO  u_0, absolute viscosity at the stagnation reference condition, lb_f-sece/ft^2
ZTR  compressibility factor-temperature product evaluated at the reference enthalpy, R
EBARL  laminar crossflow pressure gradient parameter
EBART  turbulent crossflow pressure gradient parameter
JL  laminar streamwise pressure gradient profile parameter
XEQL  laminar heat transfer equivalent distance divided by the local distance along the streamline
FNQ  see definition of F_x,Q in QTRANS
FXS  see definition of F_x,S in QTRANS
OMEGAR  reference viscosity divided by reference temperature, lb_f-sece/ft^2 R
c. Error Statements

ERROR IN CONTROL CARD (NTYPE = name)

ERROR IN CONTROL CARD (NPROG = name)

The above errors are caused by an input error of the code words NTYPE or NPROG. "Name" is the control word that was input. This error will terminate the run.

ALTITUDE INPUT TO ATMOS EXCEEDED RANGE - ASSIGN VALUE OF 2,300.000 FT.

ALTITUDE INPUT TO ATMOS BELOW ROUTINE RANGE - ASSIGN VALUE OF 0 FT.

Input altitude out of range of ATMOS subroutine. FLIGHT case only.

STREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SUBROUTINE AXI2D

The input value of dx is too small for an axisymmetric or two-dimensional case. Also printed with this comment is the x-value at which the array dimension was exceeded, the maximum x defined by the geometry input, and the input dx. The run is terminated by this error.

DBTP ERROR

The independent variable is outside the range of the program tables. The independent variable and the table limits are printed. The program will use the table limit and proceed.

STREAMLINE CROSS AT XSI =

When the two divergence calculation streamlines cross, the above comment is printed with the value of the x coordinate at that point. The program moves downstream, by dx increments, until the streamlines are not crossed and begins calculations at that point.
CONVERGENCE NOT ESTABLISHED IN VELOCITY CALCULATION

The velocity iteration routine in subroutine FLOW failed to converge on a solution in 10 iterations. The program proceeds using the final calculated velocity.

SIRCH ERROR

Table interpolation error in the general interpolation routine SIRCH.

ERROR IN TABLE2 LOOK-UP

ERROR IN TABLE18 LOOK-UP

An error occurred during table interpolation in subroutine TABLE2 or TABLE18.

FREE STREAM ENTHALPY EXCEEDS TABLE VALUES IN SONENT

The enthalpy value used for table interpolation in subroutine SONENT is outside the table limits.

ERROR IN STREAMLINE CALCULATION

This comment is printed when a calculated streamline point lies outside the $\theta_0(\xi, \eta)$ grid in subroutine STREAM.
**SUBFC AS2419 DECK**

**MAIN PROGRAM FOR RHO-MU PROGRAM (AS2419)**

**COMMON/Q(A(11600),IA(10))**

**EQUIVALENCE**

1(A(1),ACH) (A(2),ALPHA) (A(3),ALT)*
2(A(4),AYEO) (A(5),BEQXL) (A(6),BEQAT)*
3(A(7),DELETA) (A(8),DELXSI) (A(9),DETA)*
4(A(10),DETA) (A(11),DX) (A(12),DXP)*
5(A(13),ETAIF) (A(14),ETAI) (A(15),E)*
6(A(16),HO) (A(17),HREFT) (A(18),PINF)*
7(A(19),POPINF) (A(20),POPSL) (A(21),QINF)*
8(A(22),RADIUS) (A(23),RCLREF) (A(24),RHMREF)*
9(A(25),RHO) (A(26),RTRANS) (A(27),SWEEP)*
10(A(28),TEMP) (A(29),TWREF) (A(30),URATIO)*
2(A(31),VEG) (A(32),VELP) (A(33),XF)*
3(A(34),XI) (A(35),XSI) (A(36),XSII)*
4(A(37),XSI) (A(38),ETA) (A(39),Y)*
5(A(137),ETAMAX) (A(187),THBMX) (A(237),THETA)*
6(A(237),ETAMAX) (A(187),THBMX) (A(237),THETA)*
7(A(237),ETAMAX) (A(187),THBMX) (A(237),THETA)*
8(A(237),ETAMAX) (A(187),THBMX) (A(237),THETA)*
9(A(237),ETAMAX) (A(187),THBMX) (A(237),THETA)*

**EQUIVALENCE**

1(A(8487),PRESRX) (A(9237),THETAS) (A(9987),UE)*
2(A(10737),TW) (A(11487),XPO) (A(11587),P)*

**EQUIVALENCE**

1(A(1),M) (IA(2),N) (IA(3),KF)*
3(A(4),II) (IA(8),NPR) (IA(9),ITC)*

**DIMENSION** X(50)*ETA(50)*Y(50)*ETAMAX(50)*THBMX(50)*
1THETA(50)*Y(50)*ETAMAX(50)*THBMX(50)*
1THETA(50)*Y(50)*ETAMAX(50)*THBMX(50)*
2U(700)*AEE(700)*TZX(750)*OMEGA(750)*THETAS(750)*DODI(750)*
3RO(1750)*TW(750)*XPO(1000)*TW(50)*
DIMENSION P(5)*ETAX(750)*X(750)*
DIMENSION DODS(50)*IPROG(9)*ITYPE(3)*PSI(50)*RORS(50)*TWS(50)*
1UES(50)*XSI(50)*THETS(50)*
DIMENSION XSIT(50)*ETAT(50)*
DIMENSION TITLE(10)*

**DATA** (ITYPE(11))=1.3/6HWT1/6HWT2/6HFLIGHT/
DATA(IPROGII)=1/9/6HREFERE/6HAXSYM/6HWTODIM/6HHEMISP,
16HEDELA/6HHELE/6HSTREAM/6HFLOW/6HQTANS/

**5000 FORMAT(1515)**
5010 FORMAT(8F10.0)
5020 FORMAT(A6,A6)
5030 FORMAT(15A5,F10.0)
5060 FORMAT(2F10.0*1015)
1000 CONTINUE
   IERROR=0
   NXSV=1
   READ(5,5020)NTYPE,TITLE(I),I=1,10
   WRITE(6,6001)(TITLE(I),I=1,10)
6001 FORMAT(1H1*10A6)
   DO 1 I=1,3
       LI=I
       IF(NTYPE-ITYPE(I))3,1,1
   1 CONTINUE
   WRITE(6,6010)NTYPE
6010 FORMAT(1H0*29HERROR IN CONTROL CARD (NTYPE=A6,2M ))
   STOP
   3 CONTINUE
   C
   GO TO (10+20+30)*LI
   10 CONTINUE
      READ(5,5010)ACH,TEMP,PINF
      GO TO 40
   20 CONTINUE
      READ(5,5010)VEL,H,POPSL,PINF
      GO TO 40
   30 CONTINUE
      READ(5,5010)ALT,VEL
   40 CONTINUE
      READ(5,5010)RCLREF,RHMR,F,TWREF
      READ(5,5020)NPROG
      WRITE(6,6015)NPROG
6015 FORMAT(1H0*A6)
   CALL INITIAL(L1,NTYPE)
   C
   CALL RFF
   DO 60 I=1,9
      IF(NPROG-IPOG(I))60,50,60
   50 NPR=1
      GO TO 70
   60 CONTINUE
      WRITE(6,6020)NPROG
   ...
6020 FORMAT (3OH0ERROR IN CONTROL CARD (NPROG=,A6*1H))
STOP
C
70 CONTINUE
GO TO (100*200*200*400*500*300*600*700*800)*NPR
C
100 CONTINUE
GO TO 1000
C
200 CONTINUE
C AXISYMMETRIC AND TWO-DIMENSIONAL
READ(5*5000)M
READ(5*5010)(XSI(I),I=1,M)
READ(5*5010)(Y(I),I=1,M)
READ(5*5010)XI,DX
CALL AXI2D
READ(5*5000)ITC
READ(5*5010)(XS(I),I=1,ITC)
READ(5*5010)(TWS(I),I=1,ITC)
READ(5*5050)KF,RTRANS
READ(5*5010)(XPO(I),I=1,KF)
CALL WALLT1(XS,TWS)
CALL BEOI
CALL QTRAN
GO TO 1000
C
300 CONTINUE
C SWEEP CYLINDER
READ(5*5010)RADIUS,Sweep
READ(5*5010)XSIF,ETAF,DX
WRITE(6*3300)RADIUS,Sweep,XSIF,ETAF,DX
6300 FORMAT(1H0,14HSWEEP CYLINDER //3X,6HRADIUS,6X,5HSweep,7X,4HXSIF,8X24190015
1,4HETAF,8X2HDIX//5E12,5)
READ(5*5000)N
ITC=N
READ(5*5010)(ETAT(I),I=1,ITC)
READ(5*5010)(TWS(I),I=1,ITC)
READ(5*5050)KF,RTRANS,DETA
READ(5*5010)(XPO(I),I=1,KF)
CALL CYLIND
CALL STREAM
CALL DIVERG
CALL FLOW
NXT=1
DO 350 I=1,II
350 TW(I)=TBLP(ETAT,TWS,ETAX(I),ITC,NXT)
CALL BEGI
CALL QTRAN
GO TO 1000
C
400 CONTINUE
C HEMISPHERE
READ(5,5010)RADIUS
WRITE(6,6400)RADIUS
6400 FORMAT(1H0,19HHEMISPHERE RADIUS = *E12.5)
CALL HEMI
CALL FLOW
410 CONTINUE
READ(5,5000)ITC
READ(5,5010)(XS(I),I=1,ITC)
READ(5,5010)(TWS(I),I=1,ITC)
READ(5,5050)KFRTRANS
READ(5,5010)(XPO(I),I=1,KF)
CALL WALLTI(XS,TWS)
CALL BEGI
CALL QTRAN
GO TO 1000
C
500 CONTINUE
C DELTA WING
READ(5,5000)ITC
XSII=0
ETAI=0
READ(5,5010)ALPHA,SWEEP,XTSIF,ETAF, XI
READ(5,5010)DELXSIFDELETA,DX
WRITE(6,6500)ALPHA,SWEEP,XSII,ETAI,XTSIF,ETAF,DELXSIF, DELLETA,DX
6500 FORMAT(11HODELTA WING //3X*5HALPHA,7X*5HSWEEP,7X*4HXSI,8X,4HETAI
18X*4HXSIF,8X,4HETAF,8X,6HDELXSIF,6X,6HDELET,6X,2HDX//9E12.5)
CALL DELTA
CALL STREAM
IF(ITC *EQ. 1)GO TO 510
READ(5,5000)MT*NT
24190125
24190126
24190127
24190128
24190129
24190130
24190131
24190132
24190133
24190134
24190135
24190136
24190137
24190138
24190139
24190140
24190141
24190142
24190143
24190144
24190145
24190146
24190147
24190148
24190149
24190150
24190151
24190152
24190153
24190154
24190155
24190156
24190157
24190158
24190159
24190160
24190161
24190162
24190163
24190164
24190165
24190166
READ(5,5010)(XSiT(I),I=1,MT)
READ(5,5010)(ETA(I),I=1,NT)
READ(5,5010)((TWALL(I,J),I=1,MT),J=1,NT)
GO TO 520
510 CONTINUE
READ(5,5010)TWC
DO 515 I=1,II
TWC(I)=TWC
515 CONTINUE
520 CONTINUE
READ(5,5050)KF,RTRAN,DE,AS
READ(5,5010)(XPO(I),I=1,KF)
CALL DIVERG
IF(ITC.EQ.1) GO TO 505
CALL WALL2(MTS,NT,XSiT,ETAT)
505 CONTINUE
CALL FLOW
CALL BEOI
CALL QTRAN
GO TO 1000
C
600 CONTINUE
C
STREAMLINE
READ(5,5000)M,NT,ITC
READ(5,5010)XSIF,ETAF,DX,X1,URATIO
READ(5,5010)(XSI(I),I=1,M)
READ(5,5010)(ETA(J),J=1,N)
READ(5,5010)(ETAMAX(J),J=1,M)
READ(5,5010)(THBMX(J),J=1,M)
READ(5,5010)((THETA(I,J),I=1,M),J=1,N)
READ(5,5010)((PRESSR(I,J),I=1,N),J=1,N)
URATIO=URATIO/VEL
IF(ITC.EQ.1) GO TO 610
READ(5,5000)MT,NT
READ(5,5010)(XSI(T),T=1,MT)
READ(5,5010)(ETA(T),T=1,NT)
READ(5,5010)((TWALL(T,J),T=1,MT),J=1,NT)
GO TO 620
610 CONTINUE
READ(5,5010)TWC
620 CONTINUE
READ(5,5000)ITC
READ(5,5010)(XS(I),I=1,ITC)
READ(5,5010)(TWS(I),I=1,ITC)
CALL WALLTI(XS,TWS)
READ(5,5050)KF*TTRANS
READ(5,5010)(XPO(I),I=1,KF)
CALL FLOW
CALL BEQI
CALL QTAN
GO TO 1000
C
800 CONTINUE
C
QTRANSFER
READ(5,5000)IS
READ(5,5010)XI,DX,URATIO
READ(5,5010)(XS(I),I=1,IS)
READ(5,5010)(PS(I),I=1,IS)
READ(5,5010)(UES(I),I=1,IS)
READ(5,5010)(THETS(I),I=1,IS)
READ(5,5010)(DODS(I),I=1,IS)
READ(5,5010)(RORS(I),I=1,IS)
II=XS(IS)-XS(1)/DX
II=II+1
XSTAR(I)=XS(1)
X(1)=XSTAR(1)
XSIX(1)=0.
ETAX(1)=0.
POPSSLX(1)=PS(1)
UE(1)=UES(1)
AIEE(1)=H-UE(1)**2/2.
CALL SONEGA(AIEE(1),POPSSLX(1),ZTEX(1),OMEGA(1))
THETS(1)=THETS(1)
DODI(1)=DODS(1)
RORI(1)=RORS(1)
DO 820 I=2,II
XSTAR(I)=XSTAR(I-1)+DX
X(I)=XSTAR(I)
XSIX(I)=0.
ETAX(I)=0.
POPSSLX(I)=TBLP(XS,PS,XSTAR(I),IS,NXSV)
UE(I)=TBLP(XS,UES,XSTAR(I),IS,NXSV)
820 CONTINUE
AIEE(I) = H-UE(I) * 2/2.
CALL SOMECA(AIEE(I), P0PSLX(I), ZTEX(I), OMEGA(I))
THETAS(I) = TBLP(XS, THETE, XTAR(I) * IS * XSV)
DODI(I) = TBLP(XS, DOSD, XTAR(I) * IS * XSV)
RORI(I) = TBLP(XS, ROR, XTAR(I) * IS * XSV)

820 CONTINUE
URATIO = UROATIO / VEL
READ(5, 5000) ITC
READ(5, 5010)(XS(I), I = 1, ITC)
READ(5, 5010)(TWS(I), I = 1, ITC)
CALL WALL(TI(XS, TWS))
READ(5, 5050) KF, RTRANS, BEQXL, BEQXT
READ(5, 5010)(XP(O), I = 1, KF)
IF(BEQX) 825, 825, 830

825 CONTINUE
CALL BEQI

830 CONTINUE
CALL QTRAN
GO TO 1000
END

$IBFTC DBTP DECK
C FUNCTION DBTP USES TBLP TO PERFORM DOUBLE LINEAR INTERPOLATION
FUNCTION DBTP(XS, YS, ZS, XX, YY, ZZ, NX, NY, NSX, NSY)
DIMENSION XX(NX), YY(NY), ZZ(50, 50)

C
ML = 0
IF(X - XX(1)) 10 * 20 * 30
10 ML = 1
20 NXS = 1
X1 = XX(1)
X2 = 0
GO TO 170
30 CONTINUE
IF(X - XX(NX)) 60 * 50 * 40
40 ML = 1
50 CONTINUE
NYS = NY
X1 = XX(NX)
X2 = 0
GO TO 170
60 CONTINUE
IF(X - XX(NXS)) 80 * 70 * 120

24190291
24190292
24190293
24190294
24190295
24190296
24190297
24190298
24190299
24190300
24190301
24190302
24190303
24190304
24190305
24190306
24190307
24190308
24190309
24190310
24190311
24190312
24190313
24190314
24190315
24190316
24190317
24190318
24190319
24190320
24190321
24190322
24190323
24190324
24190325
24190326
24190327
24190328
24190329
24190330
24190331
24190332
70 X1=XX(NXS)
    X2=0.
    GO TO 170
80 N=NXS-1
90 CONTINUE
    IF(X=XX(N))100*160*110
100 N=N-1
    GO TO 90
110 X1=XX(N)
    X2=XX(N+1)
    NXS=N
    GO TO 170
120 CONTINUE
    N=NXS
130 CONTINUE
    IF(X=XX(N))150*160*140
140 N=N+1
    GO TO 130
150 X2=XX(N)
    X1=XX(N-1)
    NXS=N-1
    GO TO 170
160 X1=XX(N)
    X2=0.
    NXS=N
170 CONTINUE
    IF(ML)=112*175*172
172 WRITE(6*6010)N,X,XX(N),NX,XX(NX)
6010 FORMAT(1H0,10DBTP ERROR/5H N= ,I3,5H X = ,E12*5,8H X(1) = ,E12*5)
13H X(*I3=4H) = *E12*5)
175 CONTINUE
    W=TBLP(YY=ZZ(1=NXS)*Y*NY*NYS)
    IF(X2)=190*190*180
180 R=TBLP(YY=ZZ(1=NXS+1)*Y*NY*NYS)
    Z=R-(R-W)*(X2-X)/(X2-X1)
    GO TO 200
190 Z=W
200 DBTP=Z
    RETURN
END
$1BFTCTTBLP   DECK
FUNCTION TBLP(XT,YT,XNTAB,N)
DIMENSION XT(NTAB),YT(NTAB)

C THIS IS A MODIFICATION OF THE UNIVAC 1107 ROUTINE
C
C DELY=0
M=0
IF(X-XT(1))1*2*3
1 M=-1
2 N=1
GO TO 100
3 IF(X-XT(NTAB))6*5*4
4 M=1
5 N=NTAB
GO TO 100
6 I=N
 IF(X-XT(N))7*100*9
7 I=I-1
8 I=I+1
9 IF(X-XT(I+1))11*10*8
10 N=I+1
GO TO 100
11 DELY=(YT(I+1)-YT(I))/(XT(I+1)-XT(I))*(X-XT(I))
12 N=I
100 TBLP=YT(N)+DELY
RETURN
END

$IBFTC DIVERG DECK
SUBROUTINE DIVERG
C DIVERG CALCULATES D/DI&P,T,THETA AT DX INCREMENTS
C ALONG STREAMLINE
C
COMMON/G/A(11600)*IA(10)
EQUIVALENCE
1(A( 10)*DETAS ) (A( 11)*DX ) (A( 12)*DXP )
2(A( 13)*ETAF ) (A( 30)*URATIO) (A( 33)*XF )
3(A( 34)*XI ) (A( 35)*XSIF ) (A( 37)*XS1 )
4(A( 87)*ETA ) (A( 137)*ETAMAX) (A( 187)*THBMA )
5(A( 987)*ODI ) (A( 1737)*ROI ) (A( 2737)*PRESSR )
6(A( 5237)*X ) (A( 5987)*XSI ) (A( 6737)*ETAX )
7(A( 7737)*XSTAR ) (A( 8487)*PRESSX ) (A( 9237)*THETAX )
7(IA( 11)*M )+IA( 2)*N )*(IA( 4)*II ),
7(IA( 8)*NPR )
8(A(11587)*P   ) + (A(2737)*THETAD)
EQUIVALENCE (THETAX,THETAS)
C
DIMENSION XSID(750*2),ETAD(750*2),XD(750*2),THETAD(750*2),XSIX(750*2)
1) *ETAX(750)*X(750), THETAX(750)*ETA(50)*XSI(50)
2) PRESSR(750), PRESSR(50)*XSTAR(750)
DIMENSION DIDI(750), RORI(750), THETAS(750), P(5)
DIMENSION ETAMAX(50)
C
REWIND 3
WRITE(3)((PRESSR(I)*J),I=1*M),J=1*N)
NX=0
C
SAVE ORIGINAL STREAMLINE DATA
I2=II
DO 10 I=1,12
XSID(I,2)=XSIX(I)
ETAD(I,2)=ETAX(I)
XD(I,2)=X(I)
THETAD(I,2)=THETAX(I)
XSIS2=XF
10 CONTINUE
C
CALCULATE STREAMLINE 1
ETAZ=ETAF
NXSV=1
ETAMX=TBX(1,ETAMAX,XSIF,M,NXSV)
ETAF=ETAZ-DETA
IF(ETAF *GT* ETA(I)) GO TO 15
ETAF=ETAZ+DETA
NX=1
15 CONTINUE
CALL STREAM
I1=II
DO 20 I=1,11
XSID(I,1)=XSIX(I)
ETAD(I,1)=ETAX(I)
XD(I,1)=X(I)
THETAD(I,1)=THETAX(I)
XSIS=1 XF
20 CONTINUE
IF(XSIS* GT* XSIS2 ) GO TO 22
GO TO 45
22 CONTINUE
I1=I1+1
I=I+1
30 CONTINUE
XSID(I+1)=XSID(I-1+1)
ETAD(I+1)=ETAD(I-1+1)
IF(I*LE*2) GO TO 35
I=I+1
GO TO 30
35 CONTINUE
XSID(I+1)=XSID(I+2)
ETAD(I+1)=ETAD(I+2)
45 CONTINUE
WRITE(6,6100)
6100 FORMAT(1H0,37HSTREAMLINE COORDINATES AND FLOW ANGLE //19H PRIMARY 24190471
1STREAMLINE //36X,20HAUXILIARY STREAMLINE //10X,4M,3XSI,9X,3M,9X,1M//24190472
2X11X,5H,THETA,27X,9X,3M,9X,9X,1M//24190473
DO 12 I=1,12
WRITE(6,6200)I,XSID(I+2),ETAD(I+2),XD(I+2),THETAD(I+2),I,XSID(I+1)
12 CONTINUE
6200 FORMAT(1H,6S15,2X,4E12,5S,10X,15,2X,4E12,5)
44 CONTINUE
C CALCULATE P*T*THETA*D/DI AT DX INCREMENTS
NXS=1
NYS=1
NS2=1
NS1=1
XSTAR(I)=XI
XF=XD(2)+1-3*DX
44 CONTINUE
DO 51 I=1,12
THETAS(I)=TBLP(XD(I+2),THETAD(I+2),XSTAR(I+1),12,NS2)
51 XSTAR(I)=XSTAR(I)+DX
NS2=1
NI=0
DO 50 I=1,1749
XSIS=TBLP(XD(I+2),XSID(I+2),XSTAR(I+1),12,NS2)
ETAS=TBLP(XD(I+2),ETAD(I+2),XSTAR(I+1),12,NS2)
ETA1=TBLP(XSID(I+1),ETAD(I+1),XSID(I+1),NS1)
ETA2=TBLP(XSID(I+1),ETAD(I+1),XS1+1,NS2)
IF(NX*GT*0) GO TO 41
IF(ETA1*LT*ETA2) GO TO 42
99 CONTINUE
WRITE(*,6099)XSI
6099 FORMAT(13H0*26HSTREAMLINES CROSS AT XSI = *E12.5)
XSTAR(1)=XSTAR(1)+DX
XI=XSTAR(1)
GO TO 44
41 CONTINUE
IF(ETA1 *GT* ETA2) GO TO 42
GO TO 99
42 CONTINUE
IF(NI *GT* 0) GO TO 43
REWIND 3
READ(3)((PRESSR(K,J),K=1*M),J=1*N)
REWIND 3
NI=1
COST1=COS(THETAS(1)/57.2958)
ETA12=ETA2
ETA11=ETA1
43 CONTINUE
RORI(1)=1.0
DODI(1)=COS(THETAS(1)/57.2958)/COST1
IF(NX *GT* 0) GO TO 46
DODI(1)=DODI(1)*(ETA2-ETA1)/(ETA12-ETA11)
GO TO 47
46 CONTINUE
DODI(1)=DODI(1)*(ETA1-ETA2)/(ETA11-ETA12)
47 CONTINUE
PRESR(1)=DBTP(ETA*XSI*PRESSR*ETAS*XIS*N*M*NYS,NXS)
IF(XSTAR(I) *GT* XF) GO TO 60
XSTAR(I+1)=XSTAR(I)+DX
II=1
50 CONTINUE
60 CONTINUE
C
RESTORE ORIGINAL STREAMLINE
DO 100 I=1,I2
XSI(I)=XSID(I+2)
ETAX(I)=ETAD(I+2)
X(I)=XD(I+2)
100 CONTINUE
IF(NPR *EQ* 6) GO TO 1890
IF(URATIO *GT* 0.0) GO TO 1890
P(1)=PRESRX(1)
P(2)=PRESRX(2)
P(3)=PRESRX(3)
DXP=DX
1890 CONTINUE
RETURN
END
$IBFIC FIND DECK
SUBROUTINE FIND(ZA,Z,NZ,NZS)
DIMENSION ZA(NZ)
ML=0
C
      IF(Z-ZA(1))10*20*30
10  ML=1
   20 NZS=1
   30 IF(Z-ZA(NZS))60*50*40
   40 ML=1
   50 NZS=NZ
   60 IF(Z-ZA(NZ))180*170*120
   80 N=NZS-1
  90 IF(Z-ZA(N))100*160*160
 160 N=N-1
   GO TO 90
 120 N=NZS
 130 IF(Z-ZA(N))150*160*140
 140 N=N+1
   GO TO 130
 150 NZS=N-1
   GO TO 170
 160 NZS=N
 170 CONTINUE
     IF(ML)175*180*175
175 CONTINUE
      WRITE(6*6010)Z,ZA(1),NZ,ZA(NZ)
5010 FORMAT(1HO*32HVARIBLE NOT IN RANGE OF TABLE. /5H Z = E12.5*9H ZA E12.5)
1(1) = E12.5*4H ZA(13*4H) = E12.5
180 CONTINUE
RETURN
END
$IBFIC STREAM DECK
SUBROUTINE STREAMLINE
CALCULATES STREAMLINE COORDINATES, PRESSURE AND DIVERGENCE
PARAMETERS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(AI 11)*DX , (A( 13)*ETA , (A( 33)*XF ),
2(AI 34)*XI , (A( 35)*XSIF , (A( 37)*XS1 ,
3(AI 87)*ETAX , (A(137)*ETAMAX , (A( 187)*TH8MX ,
4(AI 237)*THETA , (A( 5237)*X , (A( 5987)*XSIX ,
5(AI 6737)*ETAX , (A(9237)*THETAS , (A( 8)*DELXS1 ,
6(AI 8487)*ETAS , (A(10737)*XSIS , (A( 9987)*XS ,
7(AI 7737)*THETAS ,
6(IA( 1)*M ) , (IA( 2)*N ) , (IA( 4)*II ,
EQUIVALENCE (THETAS,THETAS)
DIMENSION XS1(50),ETA(50),ETAMAX(50),TH8MX(50),THETA(5),50)
DIMENSION ETAS(750),XSIS(750),XS(750),ETAX(750),XSIX(750),X(750),
1THETAS(750),THETAS(750),THETAS(750)
DIMENSION D(18)
DATA (D(I),I=1:18)/18*6H******/
RADIAN=57.2958
C
IF(IT .EQ. 1) GO TO 90
WRITE(6*6000) (XS1(I),I=1,9),D
6000 FORMAT(19HOSTREAMLINE ANGLES //16H THETA2)XSI,ETA) //19x3HXSI/16x2
1 9E12*5/16*18A6)
DO 10 J=1,N
WRITE(6*6010) ETA(J),(THETA2(I*,J),I=1,9)
6010 FORMAT(4H ETA,E11,E4,2H *,9E12*5)
10 CONTINUE
M1=1
24190610
M2=9
24190612
15 CONTINUE
IF(M .LE. M2) GO TO 80
M1=M1+9
24190614
M2=M2+9
24190615
WRITE(6*6020) (XS1(I),I=M1,M2),D
6020 FORMAT(19x3HXSI/16x9E12*5/16x18A6)
DO 20 J=1,N
WRITE(6*6010) ETA(J),(THETA2(I*,J),I=M1,M2)
20 CONTINUE
GO TO 15
24190620
24190621
24190622
80 CONTINUE
IT=1
GO TO 100
90 CONTINUE
IT=0
C
100 CONTINUE
IX=0
NX=1
NY=1
XSV=XSI(1)
NXSV=1
XSAV=XSI(1)
YSAV=ETA(1)
NXS=1
NYS=1
I=1
XS1=0.
XSI1=XSIF
ETAS1=ETAF
C
CALL FIND(XSI*XSIS1*M*NX)
CALL FIND(ETA*ETAS1*N*NY)
THET1=THETAE(NX*NY+1)
THET2=THETAE(NX+1*NY+1)
THET3=THETAE(NX+1*NY)
THET4=THETAE(NX*NY)
THBS1=DBTP(ETA*XSI*THETA*ETAS1*XSIS1*N*M*NYS*NXS)
THET5(1)=THBS1
XSIS(1)=XSIF
ETAS(1)=ETAF
XS1=0.
105 CONTINUE
DXSI=DX*COS(THBS1/RADIAN)
DETA=DX*SIN(THBS1/RADIAN)
IF(THBS1 *LE* 0.) DXSI=DX
IF(THBS1 *LE* 0.) DETA=0.
ETAS2=ETAS1-DETA
IF(THBS1 *LE* 0.) ETAS2=ETAS1
XSIS2=XSIS1-DXSI
XS2=XS1+DX
ETAD=XSIS2
152 CONTINUE
IF(ETA2-ETAMX2)1100,1100,154
154 CONTINUE
IF(ETA1-ETAMX1)1156,156,158
156 CONTINUE
IF(ETA3-ETAMX2)1600,1600,1700
158 CONTINUE
IF(ETA4-ETAMX1)160,160,999
160 CONTINUE
IF(ETA3-ETAMX2)1400,1400,1800
C
200 CONTINUE
IF(ETA1-ETAMX1)1100,1100,210
210 CONTINUE
IF(ETA2-ETAMX2)1220,220,230
220 CONTINUE
IF(ETA4-ETAMX1)1200,1200,1300
230 CONTINUE
IF(ETA3-ETAMX2)1240,240,999
240 CONTINUE
IF(ETA4-ETAMX1)1400,1400,1500
C
C Routines to Calculate THETAe At XSIS2, ETAS2
C
1100 CONTINUE
ALL POINTS ON BODY
THAS2=DBTP(ETA*XS1,THETA,ETAS2*XSIS2*N*M*NY*NXS)
GO TO 2000
C
1200 CONTINUE
POINTS 2-3-4 ON BODY - POSITIVE SLOPE
THB3=THET4+(XSIS2-XS14)/DE*(THET3-THET4)
IF(ETAMXS*LE=ETA1) GO TO 1210
THM1=TBLP(ETAMAX,XBM2*ETAMAX*NXS)
THS1=TBLP(ETAMAX,XS1,ETA2*MX*NXS)
THB1=THM1+(XSIS2-XS13)/(XS12-XS13)*(THET2-THM1)
THS2=THB3+(ETAS2-ETA3)/DE*(THB1-THB3)
GO TO 1220
C
1210 CONTINUE
THB1=TBLP(XS1,THB3+XM*XSIS2*MX*NXS)
ETAM=TBLP(XS1,ETAMAX*XSIS2*MX*NXS)
THS2=THB3+(ETAS2-ETA3)/(ETAM-ETA3)*(THB1-THB3)
1220 CONTINUE
GO TO 2000
C
1300 CONTINUE
POINTS 2–3 ON BODY – POSITIVE SLOPE
THM=TBLP(ETAMAX,TH8MX,ETA3,M,NXSV)
XSIM=TBLP(ETAMAX,XSI,ETA1,M,NXSV)
TH83=THM+((XSI2–XSI)/(XSI3–XSI))*(THET3–THM)
IF(ETAMX*LE*. ETA1) GO TO 1310
THM=TBLP(ETAMAX,TH8MX,ETA1,M,NXSV)
XSIM=TBLP(ETAMAX,XSI,ETA1,M,NXSV)
TH83=THM+((XSI2–XSI)/(XSI3–XSI))*(THET2–THM)
TH82=TH83+(ETA2–ETA3)/(ETAM–ETA3)*(TH81–TH83)
GO TO 1320
1310 CONTINUE
TH81=TBLP(XSI,TH8MX,XSI2,M,NXSV)
ETAM=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
TH82=TH83+(ETA2–ETA3)/(ETAM–ETA3)*(TH81–TH83)
1320 CONTINUE
GO TO 2000
C
1400 CONTINUE
POINTS 3–4 ON BODY – POSITIVE SLOPE
TH83=THET4+((XSI2–XSI4)/(DC*(THET3–THET4))
TH81=TBLP(XSI,TH8MX,XSI2,M,NXSV)
ETAM=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
TH82=TH83+(ETA2–ETA3)/(ETAM–ETA3)*(TH81–TH83)
GO TO 2000
1500 CONTINUE
C
ONE POINT ON BODY (POINT 3)
C
POSITIVE SLOPE
TH81=TBLP(ETAMAX,TH8MX,ETA52,M,NXSV)
TH82=TBLP(XSI,TH8MX,XSI2,M,NXSV)
ALPHA=ATAN((ETA2–ETA3)/(XSI3–XSI2))
ETAMX=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
XSIMX=TBLP(ETAMAX,XSI,ETA2,M,NXSV)
CD=XSI2–XSIMX
BD=ETAMX–ETA2
GAM=ATAN(BD/CD)
DEL=3*14159–ALPHA–GAM
CE=SIN(ALPHA)*CD/SIN(DEL)
24190748 24190749 24190750 24190751 24190752 24190753 24190754 24190755 24190756 24190757 24190758 24190759 24190760 24190761 24190762 24190763 24190764 24190765 24190766 24190767 24190768 24190769 24190770 24190771 24190772 24190773 24190774 24190775 24190776 24190777 24190778 24190779 24190780 24190781 24190782 24190783 24190784 24190785 24190786 24190787 24190788
BC=SQR((CD**2+BD**2)
TH83=TH81+CE/BC*(TH82-TH81)
ED=SIGNAM)/SIGNDEL*CD
AD=(XS1-ZXS2)/COS(ALPHA)
EA=ED+AD
TH84=THETAE(NX+NY)
TH85=TH83+ED/EA*(TH84-TH83)
GO TO 2000
C
1600 CONTINUE
TH81=THET4+(XS1-SZ14)/DC*(THET3-THET4)
TH82=TBLP(XS1,TH8MX*XS12*M+NXSV)
TH852=TH81+ETAS2-ETA3/(ETAMXS-ETA3)*(TH82-TH81)
GO TO 2000
C
1700 CONTINUE
C NEGATIVE SLOPE
C TWO POINTS ON BODY (POINTS 1 AND 4)
TH81=THET4+(ETAS2-ETA4)/DE*(THET1-THET4)
TH82=TBLP(ETAMAX,TH8MX*ETAMXS*M+NXSV)
XSIM=XBLP(ETAMAX,TH8MX*ETAMXS*M+NXSV)
TH852=TH81+(XS12-SZ14)/(XS1MX-SZ14)*(TH82-TH81)
GO TO 2000
C
1800 CONTINUE
C ONE POINT ON BODY (POINT 4)
TH81=TBLP(ETAMAX,TH8MX*ETAS2*M+NXSV)
TH82=TBLP(XS1,TH8MX*XS12*M+NXSV)
ALPHA=ATAN((ETAS2-ETA4)/(XS12-SZ14))
ETAMX=TBLP(XS1,ETAMAX,TH8MX*ETAMXS*M+NXSV)
XSIM=TBLP(ETAMAX,TH8MX*ETAS2*M+NXSV)
CD=XSIM*XS12
BD=ETAMX-ETAS2
GAM=ATAN(BD/CD)
DEL=3.14159-ALPHA-GAM
BC=SQR((BD**2+CD**2)
CE=SIN(ALPHA)/SIGNDEL*CD
TH83=TH81+CE/BC*(TH82-TH81)
ED=SIGNAM)/SIGNDEL*CD
AD=(XS12-SZ14)/COS(ALPHA)
EA=ED+AD
TH84=THETAE(NX+NY)
TH8S2=TH83+ED/EA*(TH84-TH83)
GO TO 2000

C
1900 CONTINUE
* POINT IS OFF OF BODY
C
1710 CONTINUE
XSIMX=TBLP(ETAMAX,XS1,ETAS2,M,NXSV)
ETAMX=ETAS2+(XSIMX-XSIS2)*TAN(TH8S1/RADIAN)
IF(ABS(ETAMX-ETAS2)-1.E-06)*1750*1750*1720
1720 CONTINUE
ETAS2=ETAMX
GO TO 1710
1750 CONTINUE
XSIS2=XSIMX
ETAS2=ETAMX
TH8S2=TBLP(ETAMAX,THBMX,ETAMX,M,NXSV)
I=I+1
XS(I)=XS1+(ETAS1-ETAS2)/SIN(TH8S1/RADIAN)
XSIS(I)=XSIS2
ETAS(I)=ETAS2
THETS(I)=TH8S1
IF(ETAD LE ETA(I)+.00001)XSIS(I)=0*
IF(ETAD LE ETA(I)+.00001)ETAS(I)=0*
GO TO 2100

C
2000 CONTINUE
I=I+1
XSIS(I)=XSIS2
XS(I)=XS2
ETAS(I)=ETAS2
THETS(I)=TH8S2
IF(IX GT 0) THETS(I)=TH8S1
IF(IX GT 0) GO TO 2100

C
XSIS1=XSIS2
XS1=XS2
ETAS1=ETAS2
XSIS1=XSIS2
TH8S1=TH8S2
GO TO 105
C 2100 CONTINUE
C REORDER STREAMLINE COORDINATES
X$1(X) = XS(IS(I))
ETAX(I) = ETAS(I)
X(N) = 0
THETAX(I) = THETS(I)
J = 1
DO 2200 J = 2, I
K = I - J + 1
X$1(J) = XS(IS(K))
ETAX(J) = ETAS(K)
X(N) = XS(IS(I)) - XS(IS(K))
THETAX(J) = THETS(K)
2200 CONTINUE
3000 CONTINUE
XF = XS(IS(I))
I = 1
C GO TO 5000
999 CONTINUE
WRITE(6, 999) ETAMX1, ETAMX2, ETA3, ETA4
999 FORMAT(1H0, 3HERROR IN STREAMLINE CALCULATION, /1H ETAMX1 = individual, /1H ETAMX2 = individual, /1H ETA3 = individual, /1H ETA4 = individual)
STOP
5000 CONTINUE
RETURN
END
$IBFC WALL1 DECK
SUBROUTINE WALL1(XS, TWS)
C CALCULATES TW(X)
COMMON/Q/A(11600), IA(10)
EQUIVALENCE
1(A(7737), XSTAR), (A(10737), TW), (IA(4), II)
2(A(9), ITC)
DIMENSION XS(50), TWS(50), TW(750), XSTAR(750)
NSV = 1
DO 10 I = 1, II
TW(I) = TBLP(XS, TWS, XSTAR(I), ITC, NSV)
10 CONTINUE
RETURN
END
$IBFTCTWALL2 DECK
SUBROUTINE WALLT2(MT,NT*XSIT,ETAT)
C CALCULATES TW(XS1,ETA)
COMMON/Q(A(11600),I(10)
EQUIVALENCE
1(A(5237),TWALL) * (A(6737),ETAX) * (A(5987),XSIX)
2(A(10737),TW) * (IA(4),II) * (IA(9),ITC)
DIMENSION XSIT(50),ETAT(50),TW(750)
DIMENSION ETAX(750),XSIX(750)
NXS=1
NYS=1
IF(1TC.LT.2) GO TO 20
DO 10 I=1,II
TW(I)=DBTPEAT,XSIT,TWALL,ETAX(I),XSIX(I),NT,MT,NYS,NXS
10 CONTINUE
GO TO 30
20 CONTINUE
WRITE(*,6010)
6010 FORMAT(23HOERROR IN WALLT2, ITC =I2)
30 CONTINUE RETURN
END
$IBFTC BLKDATA DECK
BLOCK DATA
COMMON/BLKDAT/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
COMMON/BLKCON/R*PSL,ACCG,RADIAN,EPS
C
DATA R*PSL,ACCG,RADIAN,EPS/
11716,2116,2,32,1739,57,29578,000001/
DATA(PRESST(I),I=1,7)/
1-6=-3,-2,-1,0,1,2,3/
DATA(ZTT(I,J),I=1,7,J=1,26)/
17*111,7*223,7*333,7*444,7*555,7*833,7*
27*1110,7*1387,7*1665,7*1944,7*2222,7*2500,7*
37*2778,7*3056,7*3333,7*3889,7*4444,7*5000,7*
47*5555,7*6667,7*7778,7*8889,7*10000,7*11111,7*
57*13889,7*16667,7*
DATA ((AYRTOT(I,J),I=1,7),J=1,9)/
17*1,42,7*2,84,7*4,26,7*5,68,7*7,10,7*11,0,7*14,8,7*19,1,7*23,7/
DATA ((AYRTOT(I,J),I=1,7),J=10,18)/
130,0,4,28,73,28,0,2,28,0,2,28,0,2,28,0,2,28,0,2,28,0,2,28,0,2,28,0,2,28,0,2,
20 DBYDX=-(V+X)/(2*DX)
GO TO 60
30 DBYDX=-(W+Y)/(2*DX)
GO TO 60
40 DBYDX=-(X+Z)/(2*DX)
GO TO 60
50 DBYDX=-(Y+Z)/DX
60 RETURN
END
$IBFTC EBAR DECK
SUBROUTINE EBAR(AYEW,AYESL,H,SIGR,EXPK,POPSL,OMEGE,OMEGS,GAMC,GAMO)
1*EBARL*EBART)
COMMON/BLKCON/R,POSL,ACCG,RADIAN,EPS
CALL SOMEGA(AYESL,POPSL,ZTESL,OMESL)
AYEV=5*(AYEW*AYESL)
PR=POSL*POPSL
F:OMUT=PR*OMEGS/R
ROMUE=PR*OMEGE/R
AYEMC=AYEV+/2*(H-AYESL)*SIGR*SQRT(ROMUT/ROMUE)
CALL SOMEGA(AYEMC,POSL,ZTMC,OMMC)
SIGC=ZTMC/ZTESL
CALL SOMEGA(AYEAV,POSL,ZTMO,OMMO)
SIG0=ZTMO/ZTESL
C
FSC=(SIGC-294)*SIGR**355/4018
FSO=(SIGO-294)*SIGR**355/4018
FKC=(2*SIGC)**EXPK
FKD=(2*SIGO)**EXPK
GAMC=.71764*FKC*(SQRT(1+FSC)-1)
GAMO=.71764*FKO*(SQRT(1+FSO)-1)

C
EBARL=1*GAMC
EBART=(1+76519*GAMO*)((1+FAMC)/(1+GAMO))**4
RETURN
END
$IBFTC I WALL DECK
SUBROUTINE I WALL(TW,POPSL,AYEW)
1105 CONTINUE
PPLOG=ALOG10(POPSL)
TWK=TW/1.8
IF(TWK-300*)1110*1110*1115
1110 CONTINUE
AYEW=432*TWK
GO TO 1130
1115 CONTINUE
IF(TWK-1800*)1120*1120*1125
1120 CONTINUE
AYEW=432*TWK+0000382*(TWK-300*)**2
GO TO 1130
1125 CONTINUE
CALL TABLE6(TWK,PPLOG,AYEW)
1130 CONTINUE
AYEW=25031*3*AYEW
RETURN
END
$IBFTC JAYELL DECK
SUBROUTINE JAYELL(AYEW,H,AYEE,ZTE,ZTS,SIGR,BETAS,POPSL,XJL)
C
AYEMS=5*(AYEW+H)
CALL SOMEGA(AYEMS,POPSL,ZTMS,OMMS)
SIGS=ZTMS/ZTS
AYEWH=AYEE/H+SQRT(SIGR)*(1-AYEE/H)
FSS=(SIGS-294)*SIGR**355*AYEWH/*4018
CPES=(ZTE/ZTS)/AYEE/H
IF(BETAS<16*20*20
10 CONTINUE
FPM=1*
J=-1
GO TO 30
20 CONTINUE
FPM=1*
J=1
30 CONTINUE
FBETAS=(1+2*CPES)*ABS(BETAS)/(2*CPES+BETAS*FPM)
XJL=(1+71764*SQRT(1+FBETAS*FSS)-71764)**J
RETURN
END
$IBFTC MUZERO DECK
SUBROUTINE MUZERO(OMEGR,ZTR,AYER,AYEE,H,PSTPSL,XMUO,XL)
XMUO=OMEGR*ZTR*(H/AYER)**1.5*(ZTR+200)/(1/(H/AYER)*ZTR+200)
AYEED=AYEE/25031.6*33.86
PPLOG=ALOG10(PSTPSL)
CALL TABLE7(AYEC,PPLOG,AYEDIE)
IF(AYEDIE)10*20*20
10 CONTINUE
AYEDIE=0
20 CONTINUE
XL=1.0+19*AYEDIE*AYEE/H
RETURN
END
$IBFTC RORUM DECK
SUBROUTINE RORUM(P,OMEGS,OMEG,OMEGW,OMEGR,ROMUR,ZTR,AYER,SIGR)
COMMON/BLKCON/R*PSL,ACCG,RADIUS,EPS
PPLOG= ALOG10(P)
OMEGE= OMEGS/OMEG
XKS=OMEGE**(1.005/(0.005+OMEGE**7))**(1.0/14*)
XKEXP=1825*1873*EXP(-XK*OMEGE/OMEGW)
XX=XKS**XKEXP
F1=XX*OMEGE/OMEGW
OMRW=F1**1.875*(1.2/(1.2+F1**5))**1
PR=P*PSL
OMEGR=OMRW*OMEGW
ROMUR=OMEGR/R*PR
OMR=1.8*OMEGR
IF(OMR-3.585E-10)30,10,10
10 CONTINUE
IF(OMR-8.03E-10)20,20,30
20 CONTINUE
ZTR=23213.13E-20/OMRK**2*(1.0+SQRT(1.0-478656E18*OMRK**2))**2
GO TO 40
30 CONTINUE
.ALL TAL19(OMRK,PPLOG,ZTRK)
40 CONTINUE
CALL TABL20(ZTRK,PPLOG,AYER)
AYER=847559.8*AYER
ZTR=ZTRK*1.8
CALL TABL14(ZTR,SIGR)
RETURN
END
$IBFTC SOMECA DECK
C ROUTINE TO OBTAIN OMEGA AND ZT
C ENTHALPY IN FT**2/SEC**2 AND PRESSURE IN ATMOSPHERES
SUBROUTINE SOMECA(ENTH,PRESS,ZT,OMEGA)

ENTHC=ENTH/25031.3
PPLOG=4.34294*ALOG(PRESS)
20 IF(ENTHC-180.0*21.21.22
21 ZT=ENTHC/432
   GO TO 40
22 ENTH=ENTHC/33.86
   CALL TABLE2(ENTHC,PPLOG,ZT,CHECK)
   IF(CHECK=1.0)30*40*30
30 CONTINUE
   WRITE(6,6000)PPLOG,ENTHC,ZT
6000 FORMAT(1H0*23ERROR IN TABLE2 LOOK-UP,5X,3E12.5)
40 IF(ZT-111.11)41,41,42
41 OMEGA=14.45+1*10
   GO TO 60
42 IF(ZT-2000.143,43,44
43 OMEGA=SQR(D.18*ZT/200.)*(40.0.*(1.8*ZT+200.))*14.454*10
   GO TO 60
44 CALL TABLE18(ZT,PPLOG,OMEGA,CHECK)
   IF(CHECK=1.0)150,60,50
50 WRITE(6,6100)PPLOG,ZT,OMEGA
6100 FORMAT(1H0*23ERROR IN TABLE18 LOOK-UP,5X,3E12.5)
60 ZT=ZT*1.8
   OMEGA=OMEGA/1.8
70 RETURN
$/IBFTC TABLE2 DECK
C
ZT=F1/RT, F
SUBROUTINE TABLE2(AYES,SPGEL,TS,CHECK)
COMMON/BLKDAT/PRESST(7),ZT(7,26),AYRTOT(7,26),OMEGAT(7,26)
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
DIMENSION L(13)
DATA KS2/1/
   IF(KS2=1)1*1*10
1 CONTINUE
   L(1)=LOC(L(1))
   L(2)=LOC(PRESST(1))
   L(3)=1
   L(4)=7
   L(5)=2
   L(6)=2
   L(7)=XTAB(0)
   L(9)=LOC(AYRTOT(1,1))
L(10)=LOC(ZTT(1,1))
L(11)=7
L(12)=7
L(13)=26
KS2=2
10 L(8)=0
TS=DATAB(AYES,PEDGEL,L(1))
CHECK=L(8)
RETURN
END
$IBTCT E A D E
C
TABLE6 I=F(T,P)
SUBROUTINE TABLE6(TW,PEDGEL,AYES)
DIMENSION TWT(19),AYE(8*19),PRESS(8),L(13)
DATA KS6/1/
DATA(TWT(J),J=1:19)/
10**500*1000*1500*1800*2000*2200*2400*2600*2800*3000*/
23200*3400*3600*3800*4000*4200*4300*4400*/
DATA(PRESS(I),I=1:8)/
1-5=-4,-3=-2=-1*0*1*2/
DATA(AYE(I,J),J=1:19,I=1:4)/
10**240**450**700*870*1320**2067**2668**2893**3044**/
23180**3374**3680*4253*5376**7350**10000**11600**12800**/
10**240**450**700*870*1097**1510**2183**2730**2982**/
23140**3285**3468**3735**4176**4940**6205**7150**8150**/
10**240**450**700*870*1022**1242**1630**2202**2728**/
23044**3227**3385**3555**3778**4105**4607**4990**5399**/
10**240**450**700*870*1003**1150**1360**1688**2156**/
22655**3030**3276**3456**3627**3823**4068**4220**4405*/
DATA(AYE(I,J),J=1:8,I=1:19)**5:*8)/
10**240**450**700*870*994**1120**1267**1464**1727**/
22080**2496**2895**3220**3467**3659**3847**3990**4051**/
10**240**450**700*870*992**1108**1237**1381**1554**/
21764**2026**2339**2683**3032**3323**3582**3860**3805**/
10**240**450**700*870*990**1105**1225**1355**1494**/
21647**1822**2023**2259**2512**2783**3063**3230**3343**/
10**240**450**700*870*988**1102**1221**1346**1475**/
21607**1751**1903**2068**2245**2436**2637**2750**2857*/
IF(KS6=1),1=10
1 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(PRESS(1))
L(3)=1
L(4)=8
L(5)=1
L(6)=1
L(7)=XTAB(0)
L(9)=LOC(TWT(1))
L(10)=LOC(AYE(1,1))
L(11)=1
L(12)=8
L(13)=19
K56=1
10     AYE=XTAB(TW,PEDEG,L(1))
RETURN
END

$IBFCT TABLE7 DECK

C    TABLE7 ID/IE=F(I,RT,P)
SUBROUTINE TABLE7(AYE,PEDEG,AYEAYE)
DIMENSION AYDAYE(7,26)*L(13)
COMMON/BLKDAT/PRESP(7)*ZT(7,26)*AYRTOT(7,26)*OMEGAT(7,26)
DATA K57/1/
DATA ([AYDAYE(I,J)],I=1:7),J=1:16)/
17*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0, 7*0
2 0,20, 0,20, 0,18, 0,17, 0,24, 0,29, 0,26
3 0,45, 0,32, 0,28, 0,33, 0,36, 0,36, 0,37
4 0,29, 0,34, 0,91, 0,58, 0,49, 0,48, 0,47
5 0,32, 0,31, 0,26, 0,19, 0,58, 0,53, 0,53
6 0,37, 0,37, 0,37, 0,16, 0,84, 0,67, 0,63
7 0,12, 0,42, 0,36, 0,23, 0,12, 0,087, 0,071
8 0,459, 0,459, 0,396, 0,294, 0,178, 0,113, 0,083
9 0,466, 0,485, 0,435, 0,367, 0,283, 0,182, 0,126
DATA ([AYDAYE(I,J)],I=1:7),J=1:25)/
1 0,499, 0,475, 0,452, 0,397, 0,338, 0,259, 0,173
2 0,537, 0,477, 0,447, 0,409, 0,365, 0,301, 0,216
3 0,607, 0,517, 0,452, 0,411, 0,375, 0,323, 0,251
4 0,724, 0,647, 0,546, 0,460, 0,400, 0,354, 0,281
5 0,757, 0,719, 0,660, 0,555, 0,462, 0,392, 0,314
6 0,761, 0,744, 0,708, 0,635, 0,526, 0,438, 0,347
7 0,760, 0,754, 0,728, 0,673, 0,588, 0,484, 0,387
8 0,757, 0,750, 0,736, 0,690, 0,624, 0,532, 0,430
9 0,745, 0,731, 0,721, 0,697, 0,661, 0,596, 0,523
DATA(AYDAYE(1:26),I=1:7)/
1.748, .723, .704, .685, .664, .618, .559/
IF(KS7=1) I=1:10
1 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(PRESST(1))
L(3)=1
L(4)=7
L(5)=2
L(6)=2
L(7)=XTAB(0)
L(9)=LOC(AYRTOT(1:1))
L(10)=LOC(AYDAYE(1:1))
L(11)=7
L(12)=7
L(13)=26
KS7=2
10 L(8)=0
AYEYE 'AB(AYEE, PEDGEL, L(1))
RETURN
END
$IBFTC
TABL14 DECK
C
TABL14 SIG=F(ZT)
SUBROUTINE TABL14(ZT,W.SIGT)
DIMENSION SIGW(35)*ZTT(35)*L(3)
DATA KS14/1/
DATA(ZTT(I),I=1:35)/
1 0,  200,  400,  600,  800,  1000,  1200,  1400,
2 1600, 1800, 2000, 2500, 3000, 3500, 4000, 4200,
3 4600, 5000, 6000, 7000, 8000, 9000, 10000, 11000,
4 12000, 13000, 14000, 15000, 16000, 17000, 18000, 19000,
5 20000, 21000, 22000,
DATA(SIGW(J),J=1:35)/
1 .770, .770, .728, .699, .684, .680, .682, .687, .695, .706, .716, .716,
2 .740, .758, .770, .776, .777, .774, .776, .761, .750, .742, .733, .733,
3 .724, .715, .708, .701, .695, .690, .687, .685, .684, .685, .689, .689,
4 .694, .704, .704,
IF(KS14-1) I=1:10
1 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(ZTT(1))
L(3)=LOC(SIGW(1))
L(4)=1
L(5)=1
L(6)=2
L(7)=35
KS14=2
10 L(8)=0
SIGT=TAB(ZTW*L(1))
RETURN
END
$IBFTC TABL18 DECK
C TABL18 OMEGA=F(ZT*P)
SUBROUTINE TABL18(ZT*PRESSL*OMEGA*CHECK)
COMMON/BLKDAT/PRESSST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
COMMON/BLKCON/R*PSL*ACCG,RADIAN,EPS
DIMENSION L(13)
DATA KS18/I/
IF(KS18-1)*1,1*10
1 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(PRESSST(1))
L(3)=1
L(4)=7
L(5)=1
L(6)=1
L(7)=XTAB(0)
L(9)=LOC(ZT(1,1))
L(10)=LOC(OMEGAT(1,1))
L(11)=7
L(12)=7
L(13)=26
KS18=2
10 L(8)=0
OMEGA=(DTAB(ZT*PRESSL*L(1)))*1*E=10
RETURN
END
$IBFTC TABL19 DECK
C TABL19 ZT=F(OMEGA*P)
SUBROUTINE TABL19(OMEGA*PRESSL*ZT)
COMMON/BLKDAT/PRESSST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
COMMON/BLKCON/R*PSL*ACCG,RADIAN,EPS
DIMENSION L(13)
DATA KS19/1/
  IF(KS19-1)1,1,10
 1 CONTINUE
  L(1)=LOC(L(1))
  L(2)=LOC(PRESSST(1))
  L(3)=1
  L(4)=7
  L(5)=1
  L(6)=1
  L(7)=XTAB(0)
  L(9)=LOC(OMEGAT(1,1))
  L(10)=LOC(ZTT(1,1))
  L(11)=7
  L(12)=7
  L(13)=26
  KS19=2
  L(8)=0
  OMEGA = OMEGA/1.E-10
  ZT=DTAB(OMEGA,PRESSL,L(1))
  RETURN
END

$IBFTCTTBL20  DECK
C
  TABL20 1/RT=F(ZT,P)
  SUBROUTINE TABL20(ZT,PRESSL,AYERTO)
  COMMON/BLKDAT/PRESSST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
  COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
  DIMENSION L(13)
  DATA KS20/1/
  IF(KS20-1)1,1,10
 1 CONTINUE
  L(1)=LOC(L(1))
  L(2)=LOC(PRESSST(1))
  L(3)=1
  L(4)=7
  L(5)=1
  L(6)=1
  L(7)=XTAB(0)
  L(9)=LOC(ZTT(1,1))
  L(10)=LOC(AYRTOT(1,1))
  L(11)=7
  L(12)=7
L(13)=26
KS20=2
10 L(8)=0
AYERTO=DTAB(ZT*PRESSL*L(1))
RETURN
END
$IBFC Delta Deck
SUBROUTINE DELTA
C
CALCULATES INPUTS TO STREAM FOR DELTA WING
C
RESTRICTIONS
C
60 GE SWEEP LE 80
C
6 GE MACH LE 22
C
ANGLE OF ATTACK GT 0.
C
EQUIVALENCE
1(A(1),ACH ) *(A(2),ALPHA ) *(A(7),DELETA ),
2(A(8),DELXSI ) *(A(13),ETAF ) *(A(14),ETAI ),
3(A(15),H ) *(A(18),PINF ) *(A(20),POPSL ),
4(A(22),RADIUS ) *(A(25),RHO ) *(A(27),SWEEN ),
5(A(30),URATIO ) *(A(31),VEL ) *(A(35),XSIF ),
6(A(36),XSII ) *(A(37),XSI ) *(A(87),ETA ),
7(A(137),ETAMAX ) *(A(187),THBMX ) *(A(237),THEETA ),
8(A(273),PRESSR ) *(IA(1),M ) *(IA(2),N ),

COMMON/BLKCON/R,PSL,ACCG,RADIAN,EP
DIMENSION XSI(50),ETAMAX(50),THBMX(50),ETA(50),THEETA(50,50),
PRESSR(50,50),PHIT1(25),XN1(25),SWEEN(50),XNT2(50,50),PHIT2(50)

DATA NT1*(PHIT1(1),XN1(1),I=1,21)/ 21,
1.10, .095, .15, .125, .20, .143, .25, .182, .30, .222,
2.15, .258, .40, .292, .45, .327, .50, .388, .55, .450,
3.60, .473, .65, .480, .70, .479, .75, .475, .80, .467,
4.85, .457, .90, .449, .95, .439, 1.0, .430, 1.1, .413,
5.0, .4, 398 /

DATA NT1,NNTJ*(PHIT2(J),J=1,50)/ 5,50,
11.2, 1.9, 1.6, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2,
22.5, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3,
33.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4,
44.5, 4.6, 4.7, 4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5,
55.6, 5.8, 6.0, 6.2, 6.4, 6.6/

DATA SWEEN(1),XN2(1),J=1,50)/ 60
1.398, .390, .395, .421, .480, .550, .645, .760, 1.00, 41.0, 1.00 /

DATA SWEEN(2),XN2(2),J=1,50)/ 65
1.398, .387, .377, .380, .400, .426, .460, .498, .542, .597, 663,

...
NX=1
NY=1
XNCL=DBTP(PHI2,SWEET,XT2,PHI,SWEET,NTJ,NTI,NX,NY)
30 CONTINUE
XNCL=(1.*PHI)*XNCL/((ACHN2+1.)/ACHN2)
DELN=(90.-SWEET)*XNCL
THDN=THET-DELN
TAND=TAN((90.-SWEET)/RADIaN)
I=1
XSII(I)=XSII
40 CONTINUE
J=1
ETAM2(I)=XSII(I)*TAND
THMX(I)=DELN+THDN
ETA(J)=ETA(J)
50 CONTINUE
AOAM=TAN(ETA(J)/XSII(I))/TAND
THETE(I,J)=DELN+AOAM+THDN*AOAM**9
PRESSR(I,J)=P2/PSL/POPSL
J=J+1
ETA(J)=ETA(J-1)+DELETA
IF(ETA(J) .LE. ETAM2(I)+DELETA) GO TO 50
I=I+1
XSII(I)=XSII(I-1)+DELSII
60 CONTINUE
IF(XSII(I) .GT. XSIF+DELSII) GO TO 70
GO TO 40
70 CONTINUE
M=M-1
N=N-1
100 CONTINUE
RETURN
END

$IBFTC CYLIND DECK
SUBROUTINE CYLIND
C
CALCULATES INPUTS TO STREAM FOR SWEPT CYLINDER
COMMON/Q/A11600),IA(10)
COMMON/BLCKCON/R,PSL,ACCG,RADIAN,EPS
EQUIVALENCE
1(A( 1)*ACH ) , (A( 11)*DX ) , (A( 12)*DXP ) ,
2(A( 18)*PINF ) , (A( 20)*POPSL ) , (A( 22)*RADIUS),
3(A( 27)*SWEET ) , (A( 30)*URATIO) , (A( 31)*VEL ) ,
C

DIMENSION XSI(50), ETA(50), ETAMAX(50), TH8MX(50), THETA(50,50)
1PRESR(50,50), UE(1750), V(1750), PRESR(750), TW(750), TWALL(50,50)
DIMENSION TH8T(50), PPT2T(50)
DIMENSION P(5)
DIMENSION XSTAR(750), TXW(750)

C

DATA NT, (TH8T(1), PPT2T(1), 1=1, 44)/
1 0**1.00000, 2**99998782, 4**99995126, 6**9989034,
1 8**99980, 10**99970, 12**99956, 14**99940,
1 16**99922, 18**99902, 20**99878, 22**99726,
2 4**99512, 5**99239, 6**98907, 7**98516,
3 8**98063, 9**97553, 10**96985, 11**963597,
4 12**95677, 13**94937, 14**94148, 15**93302,
5 16**92402, 17**91451, 18**90452, 19**89401,
6 20**8887, 24**840, 28**795, 32**732, 36**,
2 670**40, 608, 44**546, 48**485, 52**426, 56**370, 60**317, 60**317,
5 365**258, 70**207, 80**124, 90**077, 100**05,
C

NXS=1
NXSV=1
XSV=Dx
USIN=VEL*SIN(SWEEP/RADIANS)
ACHN=ACH*COS(SWEEP/RADIANS)
DETA=**RADIUS
PT2=(L2**ACHN*2)**3.5*(6.0/(7**ACHN**2-1.0))*2.5*PIN
XSTAR(I)=0
DO 20 I=1,750
THETR=RADIANS*XSTAR(I)/RADIUS
IF(IHTR=100) 10,30,30
20 CONTINUE
POPT2=TBLP(TH8T, PPT2T, THETR, NT, NXS)
PRESR(I)=POPT2PT2/PPOSPL
TW(I)=TBLP(ETA, TXW, XSTAR(I), N, NXSV)
XSTAR(I+1)=XSTAR(I)+DETA

C

10 CONTINUE

POPT2=TBLP(TH8T, PPT2T, THETR, NT, NXS)
PRESR(I)=POPT2PT2/PPOSPL
TW(I)=TBLP(ETA, TXW, XSTAR(I), N, NXSV)
XSTAR(I+1)=XSTAR(I)+DETA

146
II=1
20 CONTINUE
WRITE(6,1000)THETR,ETA(II)
6100 FORMAT(1H0,67HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SUBROUTINE CYLIND /15H THETAR-ETA(II) ,5X,2E12.5//26H RUN TERMINATED 24191540
2 BY PROGRAM )
STOP
30 CONTINUE
II=II+1
XSTAR(II)=XSTAR(I-1)+DETA
PRESX(II)=PRESX(II-1)
TWX(II)=TWX(II-1)
DX=DETA;
URATIO=0.
P(1)=1.0
P(2)=99998782
P(3)=99995126
DXP=0.00349
CALL FLOW
M=2
N=II/4
J=1
DO 35 I=1,N
PRESX(I)=PRESX(J)
TWX(I)=TWX(J)
V(I)=V(J)
ETA(I)=XSTAR(J)
J=J+4
35 CONTINUE
DO 40 I=1,9
XSI(I)=0.
DO 40 J=1,N
THETA(I,J)=0.
40 CONTINUE
DO 60 I=1,M
DO 50 J=1,N
THETA(I,J)=ATAN(V(J)/USIN)*Radian
PRESR(I,J)=PRESX(J)
50 CONTINUE
60 CONTINUE
XSI(1)=0.
XSI(2)=1.E+6
ETAMAX(1)=ETA(N)
ETAMAX(2)=ETA(N)
THBMX(1)=THETAE(1*N)
THBMX(2)=THETAE(1*N)
DX=XSV
URATIO=SIN(SWEEP/57.2950)
RETURN

END

$IBFTC
FLOW DECK
SUBROUTINE FLOW
C FLOW CALCULATES THE EXTERNAL FLOW AND STAGNATION
C PROPERTIES
C
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A( 11)*DX ), (A( 12)*DXP ), (A( 15)*H )
2(A( 20)*PSL  ), (A( 30)*URATIO ), (A( 31)*VEL )
3(A( 27)*SWEEP ), (A( 237)*PSPLX ), (A( 2737)*AIEE )
4(A( 3487)*OMEGAE), (A( 4237)*ZTEX  ), (A( 7737)*XSTAR )
5(A( 8487)*PRESX ), (A( 9987)*UE  ), (A(11587)*P )
6(AI( 4)*II )
DIMENSION PRESX(750)*PSPLX(750)*UE(750)*AIEE(750)*
12TEX(750)*OMEGAE(750)*XSTAR(750)
DIMENSION P(5)
C
C DO 780 J=1,II
POP SLX(J)=PSL*PRESX(J)
CONTINUE
780 C CALCULATE VELOCITY, EDGE ENTHALPY, AND ZT ARRAYS
800 CONTINUE
KK=2
UEINF=5*URATIO**2
UE(1)=URATIO*VEL
U=UE(1)
IF(NPR.EQ.6) U=VEL*SIN(SWEEP/57.2950)
AIEE(1)=H=5*U**2
CALL SOME(4E1)*PSLX(1),ZTEX(1)*OMEGAE(1))
ZTE=ZTEX(1)
IF(U/RATIO)805*805*810
805 CONTINUE
SECDRV=(P(1)-2*P(2)+P(3))/DX**2
UEOUI=DX/VEL*SORT(-R*ZTE*SECDRV)
UEINF=5*UEOUI*2
UE(2)=VEL*UEOUI
U=UE(2)
AIEE(2)=H*5*U**2
IF(NPR*EQ.6)AIEE(2)=AIEE(2)+5*VEL*SIN(SWEEP/57.2958)**2
CALL Omega(AIEE(2),POPXL(2),ZTEX(2),OMEGA(2))
ZTE=ZTEX(2)
KK=KK+1
810 CONTINUE
DO 865 I=KK,II
IF(I-3)815*820*825
815 CONTINUE
CALL DERV(PRESX(I-1),PRESX(I),PRESX(I+1),PRESX(I+2),
1PRESX(I+3),DBYDX1,DX+1)
CALL DERV(PRESX(I-1),PRESX(I),PRESX(I+1),PRESX(I+2),
1PRESX(I+3),DBYDX2,DX+2)
GO TO 845
820 CONTINUE
CALL DERV(PRESX(I-2),PRESX(I-1),PRESX(I),PRESX(I+1),
1PRESX(I+2),DBYDX1,DX+2)
CALL DERV(PRESX(I-2),PRESX(I-1),0.,PRESX(I+1),PRESX(I+2),
1DBYDX2,DX+3)
GO TO 845
825 CONTINUE
IF(I-I+1)830*835*840
830 CONTINUE
CALL DERV(PRESX(I-3),PRESX(I-2),0.,PRESX(I),PRESX(I+1),
1DBYDX1,DX+3)
CALL DERV(PRESX(I-2),PRESX(I-1),0.,PRESX(I+1),PRESX(I+2),
1DBYDX2,DX+3)
GO TO 845
835 CONTINUE
CALL DERV(PRESX(I-3),PRESX(I-2),0.,PRESX(I),PRESX(I+1),
1DBYDX1,DX+3)
CALL DERV(PRESX(I-3),PRESX(I-2),PRESX(I-1),PRESX(I),
1PRESX(I+1),DBYDX2,DX+4)
GO TO 845
840 CONTINUE
CALL DERV(PRESX(I-4),PRESX(I-3),PRESX(I-2),PRESX(I-1),
1PRESX(I),DBYDX1,DX3)
845 CONTINUE
DBYDX=(DBYDX1+DBYDX2)/2.
JJ=1
DO 855 NO=1,10
UEINF2=UEINF1
UEINF1=UEINF1-R*ZTE*DBYDX*DX/(VEL**2*(PRESX(I-1)+PRESX(I))/2)
AIEE(I)=H-UEINF1*VEL**2
IF(NPR.EQ.6)AIEE(I)=AIEE(I)-0.5*(VEL*SIN(SWEEP/57.2958)**2
CALL SOMEGA(AIEE(I),POPSLX(I),ZTE,OMEGA)
IF(JJ)847*850*847
847 CONTINUE
JJ=0
GO TO 855
850 CONTINUE
IF(ABS(UEINF1-UEINF2)/UEINF1-EPS)860*855*855
855 CONTINUE
WRITE(6,6005)UEINF1,UEINF2
6005 FORMAT(1HO,62HCONVERGENCE NOT ESTABLISHED IN VELOCITY CALCULATION
1UEINF1 = .E12*5,10HUEINF2 = .E12*5)
860 CONTINUE
UEINF=UEINF1
UE(I)=VEL*SQR(2.*UEINF)
ZTEX(I)=ZTE
OMEAGA(I)=OMEGA
865 CONTINUE
IF(NPR.EQ.6)NPR=100
C
RETURN
END
$IBFTC AX12D DECK
SUBROUTINE AX12D
C CALCULATES INPUTS TO QTRAN FOR TWO-DIMENSIONAL AND
C AXISYMMETRIC BODIES
COMMON/BLKCON/RPSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A(1),ACH),(A(5),BEQXL),(A(6),BEQXT)
2(A(11),DX),(A(15),H),(A(16),HO)
3(A(18),PINF),(A(20),POPSL),(A(30),URATIO)
4(A(31),VEL),(A(34),XI),(A(37),XSI)
5(A(87),Y),(A(987),DODI),(A(1737),RORI)
6(A(237),AIEE),(A(3487),OMEGA),(A(4237),ZTEX)
7(A(5237),X),(A(5987),XSIX),(A(6737),ETAX)
8(A(7737),XSTAR),(A(9237),THETAX),(A(9987),UE)
9(A(237),POPSLX),((IA(1),M),(IA(4),II)
1(IA(8),NPR)
DIMENSION XSI(500),Y(500),XSTAR(750),POPSLX(750),UE(750),AIEE(750),
ZX(750),OMEGA(750),DODI(750),RORI(750)
DIMENSION YS(750),PRESS(750),XS(50)
DIMENSION DELS(50),XSI(50)
DIMENSION XST(50),PREST(50)
DIMENSION X(750),XSIX(750),ETAX(750),THETAX(750)
C
NXSV=1
XSI(1)=XSI(1)
YS(1)=Y(1)
XS(1)=XI
XSTAR(1)=XI
IF(NPR=2)120,120,110
110 CONTINUE
DODI(1)=1.0
RORI(1)=1.0
GO TO 130
120 CONTINUE
DODI(1)=Y(1)
RORI(1)=Y(1)
130 CONTINUE
DO 100 I=2,M
XSI(1)=0.5*(XSI(1)+XSI(I-1))
YS(1)=0.5*(Y(1)+Y(I-1))
XS(1)=XSI(I-1)+SORT(YS(I)-YS(I-1))*2+(XSI(I)-XSI(I-1))*2
DELS(1)=ATAN(Y(I)-Y(I-1))/(XSI(I)-XSI(I-1))
IF(XSI(I)=EQ.XSI(I-1)) DELS(1)=1.5707963
DEL=DELS(I)
IF(DELS(I))10,10,20
10 CONTINUE
CP50=0.0
55 CONTINUE
XSONIC=RBLP(PREST,XST,P&M)
SONIC=SORT(H/3.)
58 CONTINUE
IF(XSTAR(I)-XSONIC)*70*T0.*80
70 CONTINUE
UE(I)=SONIC/XSONIC*XSTAR(I)
GO TO 90
80 CONTINUE
UE(I)=SONIC*SQRT(6.*((1-POPSL(1)/POPSL)**2*285714))
90 CONTINUE
60 CONTINUE
AIEE(I)=H-UE(I)**2/2.
CALL SOME(AIEE(I)*POPXL(I),ZTEX(I),OMEGA(I))
IF(MPR=2)150*150*140
140 CONTINUE
DODI(1)=1.
RORI(1)=1.
GO TO 160
190 CONTINUE
DODI(I)=TBLP(XS*YS*XSTAR(I),M*XSV)
RORI(I)=DODI(I)
160 CONTINUE
I=1
200 CONTINUE
THETA(I)=THETA(2)
WRITE(6,6100)XSTAR(I),M*XS(M),DX
6100 FORMAT(HO*66HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SUBROUTINE AX1D /50H XZEX(I)-M-XXS(M)-DX *5XE12,5*13*2X*2E12,5)
WRITE(6,6120)
6120 FORMAT(HO*25HRUN TERMINATED BY PROGRAM)
STOP
210 CONTINUE
URATON=UE(1)/VEL
C
IF(MPR=2)224*224*221
224 CONTINUE
WRITE(6,6150)
6150 FORMAT(HO*19HAXISYMMETRIC OUTPUT)
GO TO 223
221 CONTINUE
WRITE(6,6160)
6160  FORMAT(1H0*22HTWO-DIMENSIONAL OUTPUT)
223  CONTINUE
WRITE(6*6200)XI*DX
6200  FORMAT(1H0*10X*5H1 = *E12.5,6H DX = *E12.5/12X*3HSGI,9X*1HY/) 
M=M+1
DO 222 I=1*M
WRITE(6*6300)XSI(I),Y(I)
222  CONTINUE
M=M+1
RETURN
END
$IBFTC HEMI  DECK
1 SUBROUTINE HEMI
C  CALCULATES INPUTS TO FLOW FOR HEMISPHERE
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A( 1),ACH ) , (A( 11)*DX ) , (A( 12)*DXP ) ,
2(A( 18),PINF ) , (A( 20),PSPS ) , (A( 22),RADIUS) ,
3(A( 34),XI ) , (A( 987),DODI ) , (A( 1737),ROI ) ,
4(A( 5237),K ) , (A( 5987),XSIX ) , (A( 6737),LTAX ) ,
5(A( 7737),XSTAR ) , (A( 9237),THETAX ) , (A(11587),P ) ,
6(A( 30),URATIO ) , (A( 8487),PRESX ) , (IA( 4),11 ) ,
DIMENSION XSTAR(750),PRESX(750),DODI(750),ROI(750)
DIMENSION TH87(50),PPT2(50)
DIMENSION P(5)
DIMENSION X(750),XSIX(750),E(750),THETAX(750)
C
DATA NT,(TH87(1),PPT2(1),I=1,44)/44,
1 0.0000, 0.9998782, 0.9995126, 0.99982034,
1 0.0000, 0.999900, 1.000, 0.99970, 0.99956, 1.000, 0.99940,
1 1.000, 0.999822, 1.800, 0.999202, 0.99878, 0.99726,
2 4.000, 9.99512, 5.000, 9.92399, 0.98907, 7.000, 0.98516,
3 8.000, 9.80639, 9.000, 9.75539, 0.96885, 11.000, 0.963597,
4 12.000, 9.56779, 13.000, 9.49397, 0.94148, 15.000, 0.93302,
5 16.000, 9.2402, 17.000, 9.1451, 0.90452, 19.000, 0.894021,
6 20.000, 8.8302, 24.000, 8.838, 0.785, 32.000, 0.720,
7 36.000, 6.53, 40.000, 5.85, 44.000, 5.17, 48.000, 4.50,
8 52.000, 3.84, 56.000, 3.20, 60.000, 2.62, 64.000, 2.00,
C 9 70** 0.147 80** 0.076 90** 0.045 100** 0.4 24191868
C
NXSV=1
XI=0
P(1)=1.0
P(2)=99998782
P(3)=99995126
DXP=0.00349
DX=0.03000 RADIUS
XSTAR(1)=XI
DO 100 1=1,750
THETA=RADIAN*XSTAR(I)/RADIUS
X(I)=XSTAR(I)
XSIX(I)=0
ETAX(I)=0
THETA(I)=90.0-THET
IF(THETA(I) < 0.0) THE1AX(I)=0.
IF(THETA(I)-100 50*50*110
50 CONTINUE
POPT2=TBLP(THET*PPT2*THET*NT/NXSV)
PRESRX(I)=POPT2
OR1(I)=SIN(XSTAR(I)/RADIUS)
DOD1(I)=OR1(I)
XSTAR(I+1)=XSTAR(I)+DX
I=I+1
100 CONTINUE
WRITE(6,6100)THETA,XSTAR(I),DX
6100 FORMAT(1H0,65HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SUB)24191895
1BROUNTE HEMI /15H THETA-XSTAR-DX ,5X,3E12,5) 24191896
WRITE(6,6120)
6120 FORMAT(1H0,25H RUN TERMINATED BY PROG(1A:1) 24191898
STOP
110 CONTINUE
URATIO=0.
C
RETURN
END
$IBFC QTRAN DECK
SUBROUTINE QTRAN
C QTRAN CALCULATES HEAT TRANSFER DATA GIVEN THE
C STREAMLINE FUNCTIONS
COMMON/G/ (11600)*IA(10) 24191909
C
COMMON/BLKCON/REST,ACCG,RADIAN, EPS 
EQUIVALENCE 
1(A( 4)•AYEWO )• (A( 5)•BEQXL )• (A( 6)•BEQXT )• 24191910 
2(A( 11)•DX )• (A( 15)•H )• (A( 16)•HO )• 24191911 
3(A( 17)•HEFT )• (A( 21)•QINF )• (A( 26)•RTRANS )• 24191912 
4(A( 31)•VEL )• (A( 32)•VELP )• (A( 34)•XI )• 24191913 
5(A( 231)•POPSLX )• (A( 987)•DO DI )• (A( 1737)•RO RI )• 24191914 
6(A( 2737)•AIEE )• (A( 3487)•OMEGA E )• (A( 4237)•ZTEX )• 24191915 
7(A( 5237)•X )• (A( 5987)•XSIX )• (A( 6737)•ETAX )• 24191916 
8(A( 7737)•XSTAR )• (A( 9237)•THETAS )• (A( 9987)•UE )• 24191917 
9(A( 10737)•TW )• (A(11487)•XPO )• (A( 21)•RADIUS )• 24191918 
11(A( 3)•KF )• (IA( 4)•I1 )• (IA( 8)•NPR )• 24191919 
21(A( 2730)•AYEW )• (A( 232)•SL )• (A( 980)•ST )• 24191920 
31(A( 9)•ITC )• 24191921 
DIMENSION AYEW (750)• SL(750)• ST(750)• 24191922 
DIMENSION XSTAR (750)• PO PSLX (750)• AIEE (750)• ZTEX (750)• 24191923 
OMEGA E (750)• THETAS (750)• DODI (750)• RORI (750)• TW (750)• XPO (100)• 24191924 
DIMENSION X (750)• XSIX (750)• ETAX (750)• 24191925 
INTEGER GSAVE • AA 24191926 
C 
FINTRP (Q0FL • Q1FL • Q2FL • Q3FL • Q4FL) = Q3FL + (Q4FL - Q3FL) / 
( Q1FL - Q0FL ) 24191927 
C 
WRITE (6, 6200) 24191928 
6200 FORMAT (1H1, 3X, 1HX, 11X, 5HP/PSL, 7X, 2HUE, 10X, 4HDODI, 8X, 4HRORI, 8X, 2HTW 24191929 
10X3HTZT, 9X, 6HOMEGA E, 6X, 4HAIEE, 8X, 4HAYEW/ 24191930 
24X3HXS1, 9X, 5HBEQXL, 7X, 7HAYEAWL, 6X, 2MHL, 10X, 5HEBARL, 7X, 24191931 
32HML, 10X, 5HQTISO, 7X, 7HODOTT, 7X, 4HCFEL/ 24191932 
44X, 3HETA, 9X, 5HBEQXT, 7X, 7HAYEAWT, 6X, 2HIN, 10X, 5HEBART, 7X, 24191933 
52HHT, 10X, 5HQTISO, 7X, 7HODOTT, 7X, 4HCET/ 24191934 
64X, 5HHTET4, 7X, 3HZTR, 9X, 3HMUC, 9X, 3HFXQ, 9X, 3HFXS, 9X, 3HRRQ, 9X, 3HKKS, 24191935 
79X, 6HOMEGAR, 6X, 6HTETAL, 6X, 6HTETAL ) 24191936 
DO 100 K = 1, KF 24191937 
K*K 24191938 
IF (XPO(K) < GT* (XSTAR(II) - 3*XDX)) GO TO 110 24191939 
100 CONTINUE 24191940 
GO TO 120 24191941 
110 CONTINUE 24191942 
KF = K*K 24191943 
120 CONTINUE 24191944 
C INITIALIZATION 24191945
900 CONTINUE
NXSV=1
MXSV=1
I=0
K=1
ITRAN=0
ITR=0
PRINT=0
INTERP=0
GSAVE=0
COUNT=0
PHIL=0
PHIT=0
BETAS=0.5
CTR=0
IP=I1/DX
IP=IP+1
XTRANS=0

C POINT FOR REPEATED LOOPING AS X IS INCREMENTED BY DX

1001 CONTINUE
I=I+1

1100 CONTINUE
CALL ISTALL(TW(I),POPLX(I),AYEW(I))
CALL SOMEA(H,POPLX(I),ZTS,OMEGAS)
CALL SOME(A(AYEW(I)),POPLX(I),ZT,OMEGAW)
CALL RORMU(POPLX(I),OMEGAS,OMEGES(I),OMEGAW,OMEGAR,KOMUR,ATK)
LAYER+SIGR)

C VELOCITY AND DELTA DERIVATIVE

1200 CONTINUE
IF(I=1210=1220=1230)
1210 CONTINUE
CALL DERV(DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DODI(I+4),DDELUX,D2
I1+1)
CALL DERV(UE(I),UE(I+1),UE(I+2),UE(I+3),UE(I+4),DDELUX,DX+1)
GO TO 1270

1220 CONTINUE
CALL DERV(DODI(I-1),DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DDELUX,D2
I+2)
CALL DERV(UE(I-1),UE(I),UE(I+1),UE(I+2),UE(I+3),DDELUX,DX+2)

...
GO TO 1270

1230 CONTINUE
IF(I-I+1)1240+1250+1260

1240 CONTINUE
CALL DERV(DODI(I-2),DODI(I-1)*0.,DODI(I+1),DODI(I+2),DDEL/DX,DX3)
CALL DERV(UE(I-2),UE(I-1)*0.,UE(I+1),UE(I+2),DULE/DX,DX3)
GO TO 1270

1250 CONTINUE
CALL DERV(DODI(I-3),DODI(I-2)*DODI(I-1),DODI(I),DODI(I+1),DDEL/DX,DX1
1X*4)
CALL DERV(UE(I-3),UE(I-2)*UE(I-1),UE(I),UE(I+1),DULE/DX,DX4)
GO TO 1270

1260 CONTINUE
CALL DERV(DODI(I-4),DODI(I-3)*DODI(I-2),DODI(I-1),DODI(I),DDEL/DX,DX1
1X*5)
CALL DERV(UE(I-4),UE(I-3)*UE(I-2),UE(I-1)*UE(I+1),DULE/DX,DX5)
GO TO 1270

1270 CONTINUE
IF(NPR*LE.4)GO TO 1320
XN=XSTAR(I)/DODI(I)*DDEL/DX

1300 CONTINUE
IF(XN*LT.0.)GO TO 1320
IF(ABS(XN)*LE.05)GO TO 1320
IF(XN*LT.99)GO TO 1340
IF(ABS(XN-1.)*LE.01)GO TO 1320
GO TO 1350

1320 CONTINUE
EXPK=0.
AYELE=IEE(I)
GO TO 1370

1340 CONTINUE
THETSL=(XN-1.)*THETAS(I)/Radian
XK=2*3.*XN*(XN-1.)
EXPK=-1.94*EXP(-XK)
GO TO 1360

1350 CONTINUE
THETSL=(1.1-1.)*THETAS(I)/Radian
XK=2*3.*(Y-1.)
EXPK=1.94*EXP(-XK)

1360 CONTINUE
VP=UE(I)*COS(THETSL)
AYELE=H-*5*VP**2
1370 CONTINUE
C
C     EQUIVALENT DISTANCE PARAMETERS
C
C
1500 CONTINUE
C
CALL EBAR(AYEW*A,YESL*H,SIGN*EXP,F,POPSLX(I),OMEGA(I),OMEGAS,GAMC,
1GAMO*EBARL,EBART)
G=ROMUR*UE(I)
FOI=DOD(I)/ROI(I)
GLAM=G/RORI(I)**2*FOI**2*EBARL
GTURB=G/RORI(I)**1.25*FOI**2*(1.25*EBART)
IF(GSAVE=1)1510,1520,1530

1510 CONTINUE
GSAVE=1
SML=GLAM*XI*BEQXL
SUMT=GTURB*XI*BEQXT
SL(1)=SML
ST(1)=SUMT
SAVEL=SML
GO TO 1540

1520 CONTINUE
GSAVE=2
GL2=GLAM
GT2=GTURB
GO TO 1001

1530 CONTINUE
GSAVE=1
SML=SML+DX/3**1(GL1+4*GL2+GLAM)
SUMT=SUMT+DX/3**1(GT1+4*GT2+GTURB)
SL(1)=SML
ST(1)=SUMT
BEQXL=SML/(GLAM*XSTAR(I))
BEQXT=SUMT/(GTURB*XSTAR(I))
CX=1.0*SAVEL/SML
FX=71.*CX/(70.*CX+1.)
BETAS=BETAS+FX*(DBS-BETAS)
SAVEL=SML
GO TO 1540

1540 CONTINUE
IF(UE(I))1560,1550,1560

1550 CONTINUE
DBS=2.*(H/AIEE(I))*BEQXL

24192034
24192035
24192036
24192037
24192038
24192039
24192040
24192041
24192042
24192043
24192044
24192045
24192046
24192047
24192048
24192049
24192050
24192051
24192052
24192053
24192054
24192055
24192056
24192057
24192058
24192059
24192060
24192061
24192062
24192063
24192064
24192065
24192066
24192067
24192068
24192069
24192070
24192071
24192072
24192073
24192074
24192075
GO TO 1570
1560 CONTINUE
   DBE=2.*H/AIEE(I)*XSTAR(1)/UE(I)*DUEDX*BEQXL
1570 CONTINUE
   GL1=GLM
   GT1=GTJRB
1580 CONTINUE
   BOUNDARY LAYER TRANSITION CHECK
   IF(RTRANS)1600+1583,1583
1583 CONTINUE
   IF(ITRAN)1585,1585,1600
1585 CONTINUE
   CALL JAYELL(AYEW(I)*H,AEE(I)*ZTEX(I)*ZTS*SIGR,BETAS,
          POPSLX(I)*XJL)
   CALL MZERO(OMEGAR,ZTR,AYER,AEE(I)*H,POPSLX(I)*XMUX*XJL)
   SEQX=BEQXL/XJL**2
   SEQX=BEQXT/XJL**2+5625
   SEQL=SEQXL*XSTAR(1)
   SEQT=SEQXT*XSTAR(1)
   XEQX=BEQXL/XJL**4
   XEQX=BEQXL/XJL**4
   FXQ=CBRT(XJL**85*BEQXT/BEQXL)
   RR=ROMUR*UE(I)*XEQL/XMUO**2/FXQ**2
   IF(RR-RTRANS)1600+1590,1590
1590 CONTINUE
   IF(UE(I))1600+1600,1591
1591 IF(XSTAR(I))1600+1600,1592
1592 ITRAN=1
   RUM=ROMUR*UE(I)/XMUX*2
   RLS=RUM*SEQL
   RTS=RUM*SEQT
   XMUE=OMEGAE(I)*ZTEX(I)
   RTHL=XMUG/XMUE**664*SQRT(RLS)
   ADEM=ALOG10(RTHL)-407
   IF(ADEM)1593+1593+1594
1593 RTHT=0*
   CTR=0*
   GO TO 1595
1594 RTHT=XMUX/XMUE**2.135*RTH/RTHT**2.64
   CTR=SUMT*(1-(RTHL/RTHT)**1.25)
1632  RTHT=XMUX/XMU*2135*RTS/ADEN**2*64
1633  THETAL=XMUE/RHOE/UEI*RTHL
    THETAT=XMUE/RHOE/UEI*RHTH
    CFEL=2*TUEL/RHOE/UEI
    CFET=2*TUET/RHOE/UEI
    HL=332*Q/SIGR**.645*XMUX*FXQ*SQRT(RRQ)/XEQL
    HL=ACCG*XL*HL
    IF(ITRAN*GE*1) HL=0*
    HT=XMUX*FXQ*Q/SIGR**.645/XEQL*(.185*RRQ/ALOG10(RRQ+)***2.584)
    HT=ACCG*XL*HT
    IF(ITRAN*LE*0) HT=0*
    AYEAWL=H-(1-SORT(SIGR))*H-AIEE(I))
    AYEAWT=H-(1-CBRT(SIGR))*H-AIEE(I))
    IF(I*LE*1) GO TO 1660
    IF(ITC*LE*2) GO TO 1660
    DO 1635 J=2*I+2
       SL(J)=(SL(J-1)+SL(J-I))/2*
       ST(J)=(ST(J-1)+ST(J+1))/2*
1635  CONTINUE
       PHISL=0*
       PHIST=0*
       AA=0
       DO 1640 J=2*I+1
          SLBS=(SL(J)-SL(J-I))/3*/SL(I)
          STBS=(ST(J)-ST(J-I))/3*/ST(I)
          FSBSL=1*/CBRT(1*-SLBS**.75)
          STBS9=1*-STBS**.9*
          IF(STBS9)1636,1639,1639
1636  IF(AA)1639,1637,1639
1637  WRITE(6,1638)
1638  FORMAT(1HO*5X,97: IN THE CALCULATION OF THE NON-ISOTHERMAL WALL PA24192190
1RAMETER STBS9 A NEGATIVE VALUE WAS ENCOUNTERED*//6X*23HABS(STBS9)24192191
2WAS ASSUMED* )
       AA=1
1639  FSBS=ABS(STBS9)**(-1*9*)
       PHISL=(AYEW(J)-AYEW(J-I))*FSBSL+PHISL
       PHIST=(AYEW(J)-AYEW(J-I))*FSBST+PHIST
1640  CONTINUE
       PHIL=AYEW(I)-AYEW(I)-PHISL
       PHIT=AYEW(I)-AYEW(I)-PHIST
       24192159
       24192160
       24192161
       24192162
       24192163
       24192164
       24192165
       24192166
       24192167
       24192168
       24192169
       24192170
       24192171
       24192172
       24192173
       24192174
       24192175
       24192176
       24192177
       24192178
       24192179
       24192180
       24192181
       24192182
       24192183
       24192184
       24192185
       24192186
       24192187
       24192188
       24192189
       24192190
       24192191
       24192192
       24192193
       24192194
       24192195
       24192196
       24192197
       24192198
       24192199
1640 CONTINUE
QDOTL=HL*(AYEAWL+PHIL-AYEW(I))/25031.3
QDOTT=HT*(AYEAWT+PHIT-AYEW(I))/25031.3
QLISO=HL*(AYEAWL-AYEW(I))/25031.3
QTISO=HT*(AYEAWT-AYEW(I))/25031.3
HL=HL/HO
HT=HT/HRFT
IF (ITR) 1680, 1680, 1670
1670 CONTINUE
ITR=0
GO TO 1660
1680 CONTINUE
C THIS COMPLETES THE CALCULATIONS FOR OBTAINING THE
C HEAT TRANSFER COEFFICIENT AND SKIN FRICTION COEFFICIENT
C FOR BOTH LAMINAR AND TURBULENT FLOW
C
C OBTAINING THE ABOVE QUANTITIES AT THE REQUESTED
C PRINT OUT POINTS
C
1700 CONTINUE
IF (INTERP) 1701, 1735, 1701
C INTERPOLATION NEEDED FOR PRINTOUTS
1701 CONTINUE
AYAHLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWL,AYEAWL)
AYAWTI=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWT,AYEAWT)
BEILINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXL,BEQXLT)
BEQXT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXS, BEQXT)
CFEL=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFELS,CFEL)
CFELT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFETS,CFET)
DODINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),DODI,DODI(I))
EBARLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),EBARLS,EBARL)
EBARTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),EBARTS,EBART)
ENTH=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AIEE, AIEE(I))
FXQINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXQSAV,FXQ)
FXSINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXSAV,FXS)
HLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HLSAV,HL)
HTSAV=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HTSAV,HT)
XLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XJLSAV,XJL)
XNSV=5
OMEGNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),OMEGA(I-2),OMEGA(I))
OMEGRT = FINTRP(XSTAR(1-2),XSTAR(I),XPO(K),OMEGRS,OMEGAR) 24192242
PPSL = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),PPSXLX(1-2),PPSXLX(I)) 24192243
QDOTL = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTLS,QDOTL) 24192244
QDOTT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTTS,QDOTT) 24192245
QLISOT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QLISOS,QLISO) 24192246
QTISOT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QTISOS,QTISO) 24192247
RORINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RORI(I-2),RORI(I)) 24192248
RRQINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRQSAV,RRQ) 24192249
RRSINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRSSAV,RRS) 24192250
THEPTL = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETLS,THETAL) 24192251
THETT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETTS,THETAT) 24192252
THSNT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETAS(I-2),THETAS(I)) 24192253
TWINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),TW(I-2),TW(I)) 24192254
UEDGE = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),UE(I-2),UE(I)) 24192255
WENHT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),WENHT(I-2),WENHT(I)) 24192256
XEOXLT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEOXLS,XEOXL) 24192257
XEOXTT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEOXTS,XEOXT) 24192258
XMUONT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XMUOSV,XMUO) 24192259
ZTEINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTEX(I-2),ZTEX(I)) 24192260
ZTRINT = FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTRSAV,ZTR) 24192261
COUNT = COUNT + 1
IF(COUNT = 9) 1720,1720,1702
1702 CONTINUE
WRITE(6,6200)
COUNT = 1
1720 CONTINUE
XSIPO = TBLP(1,XPX1X,XPO(K),IP,MXSV)
ETAPO = TBLP(1,ETAX,XPO(K),IP,NXSV)
WRITE(6,6400) XPO(K),PPSL,UEDGE,QDINT,KORINT,TWINT,ZTEINT,OMEGNT
1ENHT = WENHT,
2XSIPO,BGLINT,XEOXLT,AYAWLT,XJLINT,EBARLT,HLINT,QLISOT,QDOTLT,CFELT
3ETAPO,BGTINT,XEOXTT,AYAWTT,XNINT,EBARTT,HTINT,QTISOT,QDOTTT,CFETT
4THSNT,ZTRINT,XMUONT,FXQINT,FXSINT,RRQINT,RRSINT
5,OMEGRT,THEPTL,THETT
6400 FORMAT(1H0+4(10E12.4))
K = K+1
INTERP = 0
IF(K = KF) 1730,1730,1000
1730 CONTINUE
IF(XSTAR(I) = XPO(K)) 1760,1740,1701
C
C PRINT OUT AND SAVE FOR INTERPOLATION

C

1735 CONTINUE
IF(IPRINT)1740*1765*1740
C PRINT OUT CURRENT VALUES

1740 CONTINUE
COUNT=COUNT+1.
IF(COUNT=9*)1750*1750*1742

C

1742 CONTINUE
WRITE(6*6200)
COUNT=1.

1750 CONTINUE
WRITE(6*6400) XPO(K),POPPLX(I),UE(I),DODI(I),ROBI(I),TW(I),ZTEX(I)
1*OMEGAE(I),AIEE(I),AYER(I),
2*SEX(I),BEOXL,XEQLX,AYEWL,XJL,EBARL,HL,QLISO,QDOTL,CFEL,
3*ETAX(I),BEOXT,XEQX,AYEAWT,XN,EBART,HT,QUISO,QDOTT,CFET,
4*THETAS(I),ZTR,XMOU,FXG,FXS,RRQ,RRS
5*OMEGAR,THETAL,THETAT
K=K+1
IPRINT=0
IF(K=KF)1760*1760*1000

1760 CONTINUE
IF(XSTAR(I)+2.*DX-XPO(K))1001*1001*1765

C SAVE ALL QUANTITIES AT LEFT END OF INTERVAL

1765 CONTINUE
AYAWLS=AYEAWL
AYAWTS=AYEAWT
BEOXL=BEOXL
BEOXTS=BEOXT
CFELS=CFEL
CFETS=CFET
EBARLS=EBARL
EBARTS=EBART
FXQSAV=FXQ
FXSSAV=FXS
HLSAVE=HL
HTSAVE=HT
XJLSV=XJL
XNSV=XN
OMEGRS=OMEGAR
QDOTLS=QDOTL
QDOTT=QDOTT
QLISC$=GLISO
QTISO=QTISO
RRQSAV=RRQ
RNSSAV=RRS
THETL$=THETAL
THETT$=THETAT
XEQXL$=XEQXL
XEQX$=XEQXT
XMUOSV=XMUO
ZTRSAV=ZTR
INTERP=1
GO TO 1001
1000 CONTINUE
WRITE(6,6100)XTRANS
6100 FORMAT(1HO,26HTRANSITION OCCURRED AT X = *E12.5)
RETURN
END
$IBFCT BEQI DECK
SUBROUTINE BEQI
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A(  5),BEQXL) , (A(  6),BEQXT) , (A( 11),DX ) ,
2(A 15),H   ) , (A( 34),X 1 ) , (A( 237),P0SLX) ,
3(A  987),DODI ) , (A( 1737),R0RI ) , (A( 2737),AIJE ) ,
4(A 3487),OMEGA ) , (A( 7737),XSTAR ) , (A( 9237),THETAS) ,
5(A  9987),UE   ) , (A(10737),TW   ) , (IA(  8),NPR ) ,
DIMENSION XSTAR(750),POPSLX(750),UE(750),AIJE(750),
1OMEGA(750),THETAS(750),DODI(750),R0RI(750),TW(750),
1DIMENSION T(11),P(11),OMEG(11),DOD(11),RO0R(11),X(11),
1THBS(11),U(11),F0F(11),AE(11)
C
RADIUS=57.3
BEQXL=1.0
BEQXT=1.0
T(1)=TW(1)
P(1)=POPSLX(1)
OMEG(1)=OMEGA(1)
DOD(1)=DODI(1)
RO0R(1)=RO0R(1)
X(1)=XSTAR(1)
THBS(1)=THETAS(1)
U(1) = UE(1)
AE(1) = AEE(1)
DO 300 ITER = 1, 5
D* = DX/10 + 
B* = 1/10 + **ITER
DO 100 I = 2, 11
B = I - 1
B = BX * B
X(I) = X(I) + B * (XSTAR(2) - XSTAR(I))
P(I) = P(I) + B * (POPSLX(2) - POPSIX(I))
T(I) = T(I) + B * (TW(1) - TW(I))
U(I) = U(I) + B * (UE(2) - UE(I))
AE(I) = AE(I) + B * (AEE(2) - AEE(I))
OMEG(I) = OMEG(I) + B * (OMEGAE(2) - OMEGAE(I))
DOD(I) = DOD(I) + B * (DODI(2) - DODI(I))
ROR(I) = ROR(I) + B * (RORI(2) - RORI(I))
THBS(I) = THBS(I) + B * (THETAS(2) - THETAS(I))
100 CONTINUE
C
NG = 0
DO 200 I = 1, 11
CALL IWall(T(I), P(I), AYEW)
CALL SOMEA(H, P(I), ZT, OMEGA)
CALL SOMEA(AYEW, P(I), ZT, OMEGA)
CALL ROMURP(I), OMEGA, OMEG, OMEGAE, OMEGAE, ROMUR, ZTR, AYER
1500 CONTINUE
C
DELTA DERIVATIVE
1200 CONTINUE
IF(I-2), 1210, 1220, 1230
1210 CONTINUE
CALL DERV(DOD(I), DOD(I+1), DOD(I+2), DOD(I+3), DOD(I+4), DDELX, DX, 1)
GO TO 1270
1220 CONTINUE
CALL DERV(DOD(I-1), DOD(I), DOD(I+1), DOD(I+2), DOD(I+3), DDELX, DX, 2)
GO TO 1270
1230 CONTINUE
IF(I-10), 1240, 1250, 1260
1240 CONTINUE
CALL DERV(DOD(I-2), DOD(I-1), DOD(I+1), DOD(I+2), DDELX, DX, 3)
GO TO 1270
1250 CONTINUE
CALL DERV(DOD(I-3), DOD(I-2), DOD(I-1), DOD(I), DOD(I+1), DDELX, DX, 4)
24192366
24192367
24192368
24192369
24192370
24192371
24192372
24192373
24192374
24192375
24192376
24192377
24192378
24192379
24192380
24192381
24192382
24192383
24192384
24192385
24192386
24192387
24192388
24192389
24192390
24192391
24192392
24192393
24192394
24192395
24192396
24192397
24192398
24192399
24192400
24192401
24192402
24192403
24192404
24192405
24192406
24192407
GO TO 1270
1260 CONTINUE
CALL DERV(DOD(I-4),DOD(I-3),DOD(I-2),DOU(I-1),DOD(I),DDELDX,DX,5)
1270 CONTINUE
IF(NPR*LE.4) GO TO 1320
XN=X(I)/DOD(I)*DDELDX
1300 CONTINUE
IF(XN*LT.0.) GO TO 1320
IF(ABS(XN)*LE.*05) GO TO 1320
IF(XN*LT.*99) GO TO 1340
IF(ABS(XN-1.)*LE.*01) GO TO 1320
GO TO 1350
1320 CONTINUE
EXPK=0.
AYESL=AE(I)
GO TO 1370
1340 CONTINUE
THETSL=(XN-1.)*THBS(I)
XK=2.*3.*XN*(XN-1.)
EXPK=-.194*EXP(-XK)
GO TO 1360
1350 CONTINUE
THETSL=(1.-1./XN)*THBS(I)
XK=2.*3.*XN-1.)
EXPK=-.194*EXP(-XK)
1360 CONTINUE
VP=U(I)*COS(THETSL/RADIANT)
AYESL=H.-5.*VP**2
1370 CONTINUE
C EQUIVALENT DISTANCE PARAMETERS
C
CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,P(I),OMEG(I),OMEGAS,GAMC,GAMO,
1EBARL,EBART)
G=ROMUR*U(I)
FOF(I)=DOD(I)/ROR(I)
GLAM=G*ROR(I)**2*FOF(I)**(2.*EBARL)
GTURB=G*ROR(I)**1.2*FOF(I)**(1.2*EBART)
IF(NG=1)1151,152,153
151 CONTINUE
NG=1
24192408
24192409
24192410
24192411
24192412
24192413
24192414
24192415
24192416
24192417
24192418
24192419
24192420
24192421
24192422
24192423
24192424
24192425
24192426
24192427
24192428
24192429
24192430
24192431
24192432
24192433
24192434
24192435
24192436
24192437
24192438
24192439
24192440
24192441
24192442
24192443
24192444
24192445
24192446
24192447
24192448
SUML=GLAM*XI*BEQXI
SUMT=GTURB*XI*BEQXTI
GO TO 160
152 CONTINUE
NG=2
GL2=GLAM
GT2=GTURB
GO TO 200
153 CONTINUE
NG=1
SUML=SUML+DX/3*{(GL1+4*GL2+GLAM)}
SUMT=SUMT+DX/3*{(GT1+4*GT2+GTURB)}
BEQXL=SUML/(GLAM*X(I))
BEQXT=SUMT/(GTURB*X(I))
160 CONTINUE
GL1=GLAM
GT1=GTURB
200 CONTINUE
BEQXL=BEQXI
BEQXT=BEQXTI
300 CONTINUE
BEQXL=BEQXI
BEQXT=BEQXTI
C
310 CONTINUE
WRITE(6,6010)BEQXI, BEQXTI
6010 FORMAT(1H0, 18H INITIAL BEQ VALUES/10H BEQXI = "E12.5/10H BEQXTI = "E12.5/)
C
DX=1*E+5*DX
RETURN
END
$10FTC INITIAL DECK
SUBROUTINE INITIAL(L1, NTYPE)
C CALCULATES INITIAL CONDITIONS
COMMON/BLKCON/R*PSL, *ACCG, RADIUS, EPS
COMMON/LOAD(11600), IA(10)
EQUIVALENCE
1(A( 1)*ACH ) , (A( 3)*ALT ) , (A( 15)*H )
2(A( 18)*PL*F ) , (A( 19)*POPINF) , (A( 20)*POPSL )
3(A( 21)*QINF ) , (A( 25)*RHO ) , (A( 26)*TEMP )
4(A( 31)*VEL ) , (A( 32)*VELP )
WIND TUNNEL I
GO TO(10+20+30)*L1
10 CONTINUE
VEL=49*ACH*SORT(TEMP)
H=6000*TEMP+5*VEL**2
RHO=PINF/(1716*TEMP)
GO TO 40
WIND TUNNEL II
20 CONTINUE
AYEI=H-.5*VEL**2
IF(AYEI-9300000.21,22,22
21 CONTINUE
SONIC=.6325*SORT(AYEI)
GO TO 23
22 CONTINUE
CALL SONENT(AYEI,SONIC)
23 CONTINUE
ACH=VEL/SONIC
TEMP=(VEL/49./ACH)**2
RHO=PINF/(1716*TEMP)
GO TO 50
FLIGHT
30 CONTINUE
CALL ATMOS(ALT,SONIC,PINF,RHO,TEMP)
H=6000*TEMP+5*VEL**2
ACH=VEL/SONIC
40 CONTINUE
QINF=.7*PINF*ACH**2
50 CONTINUE
ACHN=ACH
CALL STACON(PSTPNF,AYEST,VELP,ACHN,H,PINF,VEL)
GO TO(60+70+30)*L1
60 CONTINUE
PINF=PSTPNF
POPSL=PINF*PSL
46H QDOTO,7X,10HQDOT,REF,L,2X,10HQDOT,REF,T,2X,6HIW,REF,6X,4HIW,O,824192615
5X,6HTW,REF,//
63(7E12.4/)1/)
RETURN
END
$IBFTC SONENT DECK
SUBROUTINE SONENT(ENTH,SONIC,IERRO)
C ROUTINE TO OBTAIN FREE STREAM SPEED OF SOUND
C ENTHALPY IN FT**2/SEC**2 AND SPEED OF SOUND IN FT/SEC
DIMENSION SONICT(10),AYE(10),LA(8)
INTEGER XLOC
DATA KSS/1/
DATA(AYE(I),I=1,9)/
15.936E8,475E6,10.17E6,11.86E6,13.56E6,15.26E6,16.95E6,
218.646E,20.346E/
DATA(SONICT(I),I=1,9)/
115400,1794,1941,2076,2200,2313,2421,2519,2609/
IF(ENTH*GT*20.34E6)GO TO 30
GO TO(10+20)*KSS
10 CONTINUE
LA(1)=XLOC(LA(1))
LA(2)=XLOC(AYE(1))
LA(3)=XLOC(SONICT(1))
LA(4)=1
LA(5)=1
LA(6)=1
LA(7)=10
KSS=2
20 LA(8)=0
SONIC=TB(ENTH,LA(1))
GO TO 40
30 CONTINUE
WRITE(6,6100)ENTH
6100 FORMATO(IH*63)FREE STREAM ENTHALPY EXCEEDS TABLE VALUE IN SONENT.
1ENTHALPY = *E12.5/25H MAXIMUM VALUE = 20.34E6* )
40 CONTINUE
RETURN
END
$IBFTC STACON DECK
SUBROUTINE STACON(FSTPNF,AYEST,VELP,AHM,ACHN,H,PINF,VEL)
DIMENSION L(8),ACOST(11),CORT(11)
DATA (ACOST(I),I=1,111/)1,2,3,5,7,9,11,13,15,17,21/2,1,5,2,0,2,5/
24192616
24192617
24192618
24192619
24192620
24192621
24192622
24192623
24192624
24192625
24192626
24192627
24192628
24192629
24192630
24192631
24192632
24192633
24192634
24192635
24192636
24192637
24192638
24192639
24192640
24192641
24192642
24192643
24192644
24192645
24192646
24192647
24192648
24192649
24192650
24192651
24192652
24192653
24192654
24192655
24192656
DATA (CORRT(1)), I=1,11)/1,122,1,118,1,113,1,0955,1,068,1,055,  24192657
11,033,1,021,1,0076,1,002,1,000/
COMMON/BLKCON/R*PSL+ACCG+RADIANT+EPS
DATA (L(1), I=4,7)/1,1,2,11/
AYEST=-H++.5*VEL**2*(1- (ACHN/ACH)**2)
IF (ACHN-1)=10,20,20
10 CONTINUE
PSTO=PINF/PSL* (1+ACHN**2/5.)***3.5
GO TO 30
20 CONTINUE
PSTO=PINF/PSL* (1+2*ACHN**2)***3.5*(6/(7*AHCN**2-1.))**2.5
30 CONTINUE
R2R10=6.+ACHN**2/(ACHN**2+5.)
CALL SOMEA (AYEST*PST*ZTST1+OMEG)
PT1*PINF/PSL+(PST0-PINF/PSL)*((2*R2R10+-6006.*ZTST0AYEST)/
1(2.*R2R10-1.))
CALL SOMEA (AYEST*PST1*ZTST0+OMEG)
PST2*PINF/PSL+(PST0-PINF/PSL)*((2*R2R10-6006.*ZTST1AYEST)/
1(2.+R2R10-1.))
PSTPNF=PST2*PSL/PINF
VELP=SORT(2.*7.*(1.5+ACHN**2)*(1-1/PSTPNF)*.6006**
1ZTST1AYEST))
IF (ACHN-2.5)=40,40,50
C
40 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(ACOAST(1))
L(3)=LOC(CORRT(1))
L(8)=0
CORRT=TAB (ACHN,L(1))
GO TO 60
C
50 CONTINUE
CORRT=1.0
C
60 CONTINUE
VELP=VELP*CORRT**5
RETURN
END
$IBFTC ATMOS DECK
SUBROUTINE ATMOS(ALT,CS,P,DENS,T)
C
SAVE SENSE LIGHT
CALL SLITET(1*K000FX)
  GO TO(4600+47001*K000FX
  4600 SAVF = 1*0
  GO TO 4800
  4700 SAVF = 0*0
  4800 CONTINUE
C SET ERROR INDICATOR
I = 0
C CONVERT TO METERS
C
Z = ALT*0.3048
C
COMPUTE GEOPOTENTIAL ALTITUDE
C
IF (Z-90000+0) 1000+ 1000+ 2000
ALITUDE LESS THAN NINETEEN THOUSAND METERS
1000 Z = 6356766+0*Z/(6356766+0+Z)
WEIMOL = 28.9644
IF (Z) 1, 12, 3
  I = -1
Z = 0*0
GO TO 12
3 IF (Z-70000+0) 5, 4, 4
  TMB = 180.65
  GRAD = 0+0
  ZB = 70000+0
  PB = 1+0377E-2
  GO TO 3000
  5 IF (Z-47000+0) 6, 7, 7
  6 IF (Z-20000+0) 8, 9, 9
  7 IF (Z-61000+0) 10, 11, 11
  8 IF (Z-11000+0) 12, 13, 13
  9 IF (Z-32000+0) 14, 15, 15
  10 IF (Z-52000+0) 16, 17, 17
  11 TMB = 252.65
  GRAD = -4.0
  ZB = 61000+0
  PB = 1+82099E-1
  GO TO 3000
  12 TMB = 288.15
  GRAD = -6.5
  ZB = 0*0
PB = 1.01325E3
GO TO 3000
13 TMB = 216.65
GRAD = 0.0
ZB = 11000.0
PB = 2.26320E2
GO TO 3000
14 TMB = 216.65
GRAD = 1.0
ZB = 20000.0
PB = 5.47487E1
GO TO 3000
15 TMB = 228.65
GRAD = 2.8
ZB = 32000.0
PB = 8.68014
GO TO 3000
16 TMB = 270.65
GRAD = 0.0
ZB = 47000.0
PB = 1.10905
GO TO 3000
17 TMB = 270.65
GRAD = -2.0
ZB = 52000.0
PB = 5.90005E-1
GO TO 3000
2000 IF (Z-700000.0) 20: 20: 18
18 I = 1
Z = 700000.0
20 IF (Z-170000.0) 21: 22: 22
21 IF (Z-110000.0) 23: 24: 24
22 IF (Z-40000.0) 25: 26: 26
23 IF (Z-100000.0) 27: 28: 28
24 IF (Z-150000.0) 29: 30: 30
25 IF (Z-230000.0) 31: 32: 32
26 IF (Z-500000.0) 33: 34: 34
27 TMB = 180.65
GRAD = 0.003
ZB = 90000.0
PB = 0.16437801
24192740
24192741
24192742
24192743
24192744
24192745
24192746
24192747
24192748
24192749
24192750
24192751
24192752
24192753
24192754
24192755
24192756
24192757
24192758
24192759
24192760
24192761
24192762
24192763
24192764
24192765
24192766
24192767
24192768
24192769
24192770
24192771
24192772
24192773
24192774
24192775
24192776
24192777
24192778
24192779
24192780
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>TMB = 210.65</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.005</td>
</tr>
<tr>
<td>29</td>
<td>IF1 = 120000.0</td>
</tr>
<tr>
<td>30</td>
<td>IF1 = 120000.0</td>
</tr>
<tr>
<td>31</td>
<td>IF1 = 120000.0</td>
</tr>
<tr>
<td>32</td>
<td>IF1 = 120000.0</td>
</tr>
<tr>
<td>33</td>
<td>TMB = 400000.05</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.0026</td>
</tr>
<tr>
<td>34</td>
<td>TMB = 20000.65</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.001</td>
</tr>
<tr>
<td>35</td>
<td>TMB = 100000.0</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.006</td>
</tr>
<tr>
<td>36</td>
<td>TMB = 20000.65</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.0026</td>
</tr>
<tr>
<td>37</td>
<td>TMB = 100000.0</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.001</td>
</tr>
<tr>
<td>38</td>
<td>TMB = 110000.0</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.001</td>
</tr>
<tr>
<td>39</td>
<td>TMB = 120000.0</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.001</td>
</tr>
<tr>
<td>40</td>
<td>TMB = 130000.65</td>
</tr>
<tr>
<td></td>
<td>GRAD = 0.001</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>GB = 100000.0</td>
</tr>
<tr>
<td>29</td>
<td>GB = 200000.0</td>
</tr>
<tr>
<td>30</td>
<td>GB = 300000.0</td>
</tr>
<tr>
<td>31</td>
<td>GB = 400000.0</td>
</tr>
<tr>
<td>32</td>
<td>GB = 500000.0</td>
</tr>
<tr>
<td>33</td>
<td>GB = 600000.0</td>
</tr>
<tr>
<td>34</td>
<td>GB = 700000.0</td>
</tr>
<tr>
<td>35</td>
<td>GB = 800000.0</td>
</tr>
<tr>
<td>36</td>
<td>GB = 900000.0</td>
</tr>
<tr>
<td>37</td>
<td>GB = 1000000.0</td>
</tr>
<tr>
<td>38</td>
<td>GB = 1100000.0</td>
</tr>
<tr>
<td>39</td>
<td>GB = 1200000.0</td>
</tr>
<tr>
<td>40</td>
<td>GB = 1300000.0</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>PB = 300000.0</td>
</tr>
<tr>
<td>29</td>
<td>PB = 400000.0</td>
</tr>
<tr>
<td>30</td>
<td>PB = 500000.0</td>
</tr>
<tr>
<td>31</td>
<td>PB = 600000.0</td>
</tr>
<tr>
<td>32</td>
<td>PB = 700000.0</td>
</tr>
<tr>
<td>33</td>
<td>PB = 800000.0</td>
</tr>
<tr>
<td>34</td>
<td>PB = 900000.0</td>
</tr>
<tr>
<td>35</td>
<td>PB = 1000000.0</td>
</tr>
<tr>
<td>36</td>
<td>PB = 1100000.0</td>
</tr>
<tr>
<td>37</td>
<td>PB = 1200000.0</td>
</tr>
<tr>
<td>38</td>
<td>PB = 1300000.0</td>
</tr>
<tr>
<td>39</td>
<td>PB = 1400000.0</td>
</tr>
<tr>
<td>40</td>
<td>PB = 1500000.0</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>ZB = 300000.0</td>
</tr>
<tr>
<td>29</td>
<td>ZB = 400000.0</td>
</tr>
<tr>
<td>30</td>
<td>ZB = 500000.0</td>
</tr>
<tr>
<td>31</td>
<td>ZB = 600000.0</td>
</tr>
<tr>
<td>32</td>
<td>ZB = 700000.0</td>
</tr>
<tr>
<td>33</td>
<td>ZB = 800000.0</td>
</tr>
<tr>
<td>34</td>
<td>ZB = 900000.0</td>
</tr>
<tr>
<td>35</td>
<td>ZB = 1000000.0</td>
</tr>
<tr>
<td>36</td>
<td>ZB = 1100000.0</td>
</tr>
<tr>
<td>37</td>
<td>ZB = 1200000.0</td>
</tr>
<tr>
<td>38</td>
<td>ZB = 1300000.0</td>
</tr>
<tr>
<td>39</td>
<td>ZB = 1400000.0</td>
</tr>
<tr>
<td>40</td>
<td>ZB = 1500000.0</td>
</tr>
</tbody>
</table>

177
GRAD = 0.005
ZB = 190000.0
PB = *16852213E-03
GO TO 4000
41 TMB = 1550.65
GRAD = 0.004
ZB = 230000.0
PB = *69604207E-04
GO TO 4000
42 TMB = 1830.65
GRAD = 0.0033
ZB = 300000.0
PB = *18838460E-04
GO TO 4000
43 TMB = 2420.65
GRAD = 0.0017
ZB = 500000.0
PB = *10956796E-05
GO TO 4000
44 TMB = 2590.65
GRAD = 0.0011
ZB = 600000.0
PB = *34502145E-06
GO TO 4000
3000 TM = TMB*GRAD*(Z-ZB)*0.001
T = TM
CS = SQRT(1.4*TM/0.00348368)
GO TO 5000
4000 TM = TMB*GRAD*(Z-ZB)
IF (Z-170000.0) 4400, 4500, 4500
4400 CONTINUE
C Z BETWEEN 90000.M AND 170000.M COMPUTE M
EM =(((1.41865090E-27)*Z +111458341E-21)*Z +359201416E-16)*Z -2191E2
WEIMOL = EM*28.9644/28.966
GO TO 4050
4500 CONTINUE
C Z GREATER THAN 170000.M COMPUTE M
EM =(((1.41605540E-33)*Z -3.42916205E-27)
1*Z +3.81596699E-21)*Z -2.0402054E-15)
2*A +5.642214*E-101*Z  -1.0700489E-4)  24192864
3*A +3.56299978E1  24192865
WEIMOL = EM*28*9644/28*966  24192866
4050 CONTINUE  24192867
C COMPUTE TEMPERATURE
T = TM*WEIMOL/28*9644  24192868
C SPEED OF SOUND IS CONSTANT
CS = 269.44  24192869
C COMPUTE PRESSURE
EMOR = 0.00348368  24192870
C = (-ZB+TM/GRAD)*0.000001  24192871
D=6*3675708  24192872
CMD=C-D  24192873
CMD2=CMD*CMD  24192874
CMD3=CMD2*CMD  24192875
CMD4=CMD2*CMD2  24192876
CMD5=CMD2*CMD3  24192877
CMD6=CMD3*CMD3  24192878
CMD7=CMD4*CMD3  24192879
CMD8=CMD4*CMD4  24192880
CMD9=CMD5*CMD4  24192881
CMD10=CMD5*CMD5  24192882
C2=C*C  24192883
C3=C2*C  24192884
D2=D*D  24192885
PBLN=ALOG(PB)  24192886
TEM14 = EMOR/GRAD  24192887
CALL SLITE (1)  24192888
C COMPUTE USING 2B
ZB=ZB*0.000001  24192889
4100 ZPD=Z6+D  24192890
ZPC=Z6+C  24192891
ZPD2=ZPD*ZPD  24192892
ZPD3=ZPD2*ZPD  24192893
ZPD4=ZPD2*ZPD2  24192894
ZPD5=ZPD2*ZPD3  24192895
ZPD6=ZPD3*ZPD3  24192896
ZPD7=ZPD3*ZPD4  24192897
ZPD8=ZPD4*ZPD4  24192898
ZPD9=ZPD5*ZPD4  24192899
ZPC2=ZPC*ZPC  24192900
ZPC3=ZPC2*ZPC  24192901
...
ZDLOG=ALOG(ABS(ZPD)) / 6 * ALOG(10)
ZCLOG=ALOG(ABS(ZPC)) / 6 * ALOG(10)
ZCDLOG=ALOG(ABS(ZPC/ZPD))
TEMP1=1 / (CMD * ZPD) + ZCDLOG / CMD2
TEMP2=-ZDLOG / CMD4 - 1 / (CMD3 * ZPD) + 1 / ((2 * CMD2 * ZPD2) - 1 / (3 * CMD * ZPD3))
1+ZCLOG / CMD4
TEMP3= D * ZDLOG / CMD4 - ZPD / CMD4 + ZDLOG / CMD3 + D / (ZPD * CMD3)
1+1 / (ZPD * CMD2) - D / (ZPD * CMD2) - 1 / (2 * ZPD2 * CMD4)
2+/ (3 * ZPD3 * CMD) + ZPC / CMD4 - C * ZCLOG / CMD4
TEMP4=-ZPD2 / (ZPC * CMD4) + 1 / (CMD3 * ZPD) - ZPC2 / (ZPD * CMD3)
2-/ZDLOG / CMD2 + 2 * UV / (ZPD * CMD2) + D / (2 * ZPD2 * CMD2)
3-1 / (ZPD * CMD) + D / (ZPD2 * CMD1) - 2 / (3 * ZPD3 * CMD)
4+ZPC2 / (2 * CMD4) - 5 * C * ZPC / CMD4 + C2 / ZCLOG / CMD4
TEMP5=ZDLOG / CMD4 - 1 / (CMD3 * ZPD) + 1 / (2 * ZPD2 * CMD2)
1-1 / (3 * ZPD3 * CMC)
TEMP6=ZPC-C * ZCLOG
TEMP7=ZPC2 / 2 - 2 * C * ZPC + C2 / ZCLOG
TEMP8=ZPC3 / 3 - 3 * C * ZPC2 / 2 * 3 * C2 / ZPC-C * C3 / ZCLOG
TEMP11=ZDLOG / CMD10 - 1 / (CMD9 * ZPD) + 1 / (2 * CMD8 * ZPD2)
1-1 / (3 * CMD7 * ZPD3) + 1 / (4 * CMD6 * ZPD4) + 1 / (5 * CMD5 * ZPD5)
2+1 / (6 * CMD4 * ZPD6) + 1 / (7 * CMD3 * ZPD7) + 1 / (8 * CMD2 * ZPD8)
3-1 / (9 * CMD1 * ZPD9) + ZCLOG / CMD10
TEM12=ZDLOG / CMD9 + ZCLOG / CMD4 - 1 / (2 * CMD3 * ZPD2)
1+1 / (3 * CMD2 * ZPD3) - 1 / (4 * CMD1 * ZPD4) - ZCLOG / CMD5
TEM13=398.62694 * TEMP13 + 39.22716 * TEMP2 * 24192907
1E-7 * TEM14 + 260372892 * TEM15 + 255864746 * 2 * TEM16 + 280749260 * 6 * TEM7 + 924192932
29336241E-12 * TEM3 + 0.17043125 * ZCLOG + 1.4388483E3 * TEM11 + 29060526 * TE24192933
3M12 + 1.4673439E-5 * ZCLOG
CALL SLITET(1, K000FX)
GO TO 4200, 4300, K000FX
4200 TEM15=TEM13
C REPEAT USING Z
Z6=Z * 0.000001
GO TO 4100
4300 P = EP(PLTN-TEM14 * (TEM13-TEM15))
C CONVERT TO MB
P = P * 0.01
GO TO 8000
5000 IF (GRAD) 6000, 7000, 6000
6000 P = PB *(TMB/1M) ** (9.80665 * 9.9644 / (5.31432 * GRAD))
GO TO 8000
7000 P = PB*EXP(-9.80665*28*9644*0.001*(Z-ZU)/(8.31432*TMB))
8000 CONTINUE
C DENS = 28*9644*P/(8.31432*TM)*0.1
C CONVERT DENSITY(KG/M3) TO LB/FT3
C DENS = DENS*(0.3048)**3/0.45359237
C CONVERT DENSITY TO LB(MASS)/FT3
C DENS = DENS/32*1741
C CONVERT TO P(MM HG)
C P = P*75061682E00
C CONVERT TO IN HG
C P = P*393700787E-01
C CONVERT TO LB/FT2
C P = P*70*7269
C CONVERT CS(M/SEC2) TO CS(FT/SEC2)
C CS = CS*3*2808399
C CONVERT T(K) TO T(R)
C T = T*1.8
C RESTORE SENSE LIGHT
8100 IF (SAVE) 8200, 8300, 8200
8200 CALL SLITE (1)
GO TO 8400
8300 CALL SLITE(1*K000FX)
GO TO(8400*8400)*K000Fx
8400 CONTINUE
C IF (1) 300* 9000, 200
200 WRITE (6*400)
GO TO 9000
300 WRITE (6*500)
9000 CONTINUE
RETURN
400 FORMAT(1HO///////5X*79HALITUDE INPUT TO ATMOS EXCEEDED ROUTINE RANG24192978
1E - ASSIGNED VALUE OF 2*300*000 FT////////)
500 FORMAT(1HO///////5X*68HALITUDE INPUT TO ATMOS BELOW ROUTINE RANGE -24192980
1 ASSIGNED VALUE OF 0 FT////////)
END
b. \( \rho_r \mu_r \) Deck Setup

The following diagrams indicate the deck setup and the core storage requirements for the \( \rho_r \mu_r \) program. Each link is composed of the subroutine decks (symbolic or object) listed in the allocation diagram. All control cards are indicated by a $ in column 1 and must be punched exactly as shown in the deck setup diagram. All items in parenthesis indicate subroutine or data card decks.
Notes:

1. Storage locations are on actual numbers.

2. Load points (LP) are indicated as follows:
   
   LP1 - 04076
   LP2 - 60005
   LP3 - 65623

SCHEMATIC DIAGRAM OF COMPUTER STORAGE ALLOCATION
The contents of each link are:

Link 0
AS2419 (main program)
TAB
DTAB
DBTP
TBLP
XTAB
LOC
XLOC

Link 1
DIVERG
FIND
STREAM

Link 2
WALLT 1
WALLT 2
BLKDTA
DERV
EBAR
IWALL
JAYELL
MUZERO
RORMUR
SOMECHA
TABLE 2
TABLE 6
TABLE 7
TABLE 14
TABLE 18
TABLE 19
TABLE 20

Link 3
DELTAY

Link 4
CYLIND
FLOW

Link 5
AXI2D
HEMI

Link 6
QTRAN
BEQI

Link 7
INITIAL
REF
SONENT
STACON
ATMOS
## 1. NOMENCLATURE

<table>
<thead>
<tr>
<th>Program Symbol</th>
<th>Math Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>( C_p )</td>
<td>specific heat of air at constant pressure, ( \delta_0 )</td>
<td>( \text{ft}^2/\text{sec}^2 \times \text{R} )</td>
</tr>
<tr>
<td>( \delta )</td>
<td></td>
<td>boundary layer thickness</td>
<td>( \text{ft} )</td>
</tr>
<tr>
<td>( \delta(\eta/\rho) )</td>
<td>( \delta^* )</td>
<td>increment in boundary layer parameter</td>
<td>( \text{ft}^3/\text{slug} )</td>
</tr>
<tr>
<td>( \delta^* )</td>
<td></td>
<td>boundary layer displacement thickness</td>
<td>( \text{ft} )</td>
</tr>
<tr>
<td>( \partial H/\partial x )</td>
<td>( \partial H/\partial y )</td>
<td>derivative of total enthalpy</td>
<td>( \text{ft}/\text{sec}^2 )</td>
</tr>
<tr>
<td>( \partial H/\partial y )</td>
<td>( \partial H/\partial x )</td>
<td>derivative of total enthalpy</td>
<td>( \text{ft}/\text{sec}^2 )</td>
</tr>
<tr>
<td>( \partial \mu/\partial y )</td>
<td>( \partial \mu/\partial x )</td>
<td>derivative of viscosity</td>
<td>( \text{lb}-\text{sec}/\text{ft}^3 )</td>
</tr>
<tr>
<td>( \partial \dot{q}/\partial y )</td>
<td>( \partial \dot{q}/\partial x )</td>
<td>derivative of the heating rate</td>
<td>( \text{Btu}/\text{ft}^3 \times \text{sec} )</td>
</tr>
<tr>
<td>( \partial r/\partial x )</td>
<td>( \partial r/\partial y )</td>
<td>derivative of three-dimensional flow parameter</td>
<td>---</td>
</tr>
<tr>
<td>( \Delta S )</td>
<td></td>
<td>increment for stagnation region problems</td>
<td>( \text{ft} )</td>
</tr>
<tr>
<td>( \partial u/\partial x )</td>
<td>( \partial u/\partial y )</td>
<td>derivative of velocity</td>
<td>( 1/\text{sec} )</td>
</tr>
<tr>
<td>( \partial u/\partial y )</td>
<td>( \partial u/\partial x )</td>
<td>derivative of velocity</td>
<td>( 1/\text{sec} )</td>
</tr>
<tr>
<td>( (\partial u/\partial y)_e )</td>
<td>( (\partial u/\partial x)_e )</td>
<td>velocity derivative at boundary layer edge</td>
<td>( 1/\text{sec} )</td>
</tr>
<tr>
<td>( \partial v/\partial y )</td>
<td>( \partial v/\partial x )</td>
<td>derivative of velocity normal to surface</td>
<td>( 1/\text{sec} )</td>
</tr>
<tr>
<td>( \Delta x )</td>
<td></td>
<td>increment in ( x )-direction</td>
<td>( \text{ft} )</td>
</tr>
</tbody>
</table>

* Those symbols marked with an asterisk are arrays (tables) in the program. Throughout this document, a subscript \( i \) on either the program symbol or the equivalent math symbol indicates that reference is being made to the \( i \)th position within the array.
<table>
<thead>
<tr>
<th>Program symbol</th>
<th>Math symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DY</td>
<td>$\Delta y$</td>
<td>increment in y-direction</td>
<td>ft</td>
</tr>
<tr>
<td>* D2HDY</td>
<td>$\partial^2 H/\partial y^2$</td>
<td>second derivative of total enthalpy</td>
<td>1/sec$^2$</td>
</tr>
<tr>
<td>* D2UDY</td>
<td>$\partial^2 u/\partial y^2$</td>
<td>second derivative of velocity</td>
<td>1/sec-ft</td>
</tr>
<tr>
<td>ENDX</td>
<td>$x_f$</td>
<td>final value of x</td>
<td>ft</td>
</tr>
<tr>
<td>* ENTH</td>
<td>$i$</td>
<td>static enthalpy</td>
<td>ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>EPS</td>
<td>$\epsilon$</td>
<td>eddy viscosity</td>
<td>lb-sec/ft$^2$</td>
</tr>
<tr>
<td>EPSY</td>
<td>$\partial \epsilon / \partial y$</td>
<td>derivative of eddy viscosity</td>
<td>lb-sec/ft$^3$</td>
</tr>
<tr>
<td>* ETA</td>
<td>$\eta$</td>
<td>similarity parameter</td>
<td>---</td>
</tr>
<tr>
<td>* H</td>
<td>$I$</td>
<td>total enthalpy</td>
<td>ft$^2$/sec$^2$</td>
</tr>
<tr>
<td>* PR</td>
<td>$\sigma$</td>
<td>Prandtl number</td>
<td>---</td>
</tr>
<tr>
<td>* PRESSURE</td>
<td>$P$</td>
<td>pressure</td>
<td>lb/ft$^2$</td>
</tr>
<tr>
<td>* QP</td>
<td>$\dot{q}$</td>
<td>heating rate</td>
<td>Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>QW</td>
<td>$\dot{q}_w$</td>
<td>heating rate at the wall</td>
<td>Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>QQW</td>
<td>QQW</td>
<td>heating rate at the wall</td>
<td>Btu/ft$^2$-sec</td>
</tr>
<tr>
<td>R</td>
<td>$r$</td>
<td>three-dimensional flow parameter</td>
<td>ft</td>
</tr>
<tr>
<td>$R$</td>
<td>$R$</td>
<td>universal gas constant, 1716</td>
<td>ft$^2$/sec$^2$-R</td>
</tr>
<tr>
<td>* RHO</td>
<td>$\rho$</td>
<td>density</td>
<td>lb-sec$^2$/ft$^4$</td>
</tr>
<tr>
<td>SI</td>
<td>$S_i$</td>
<td>initial value of S for the calculation of $\eta$</td>
<td>ft</td>
</tr>
<tr>
<td>STARTX</td>
<td>$x_o$</td>
<td>initial value of x</td>
<td>ft</td>
</tr>
<tr>
<td>* T</td>
<td>$T$</td>
<td>temperature</td>
<td>°R</td>
</tr>
<tr>
<td>* TAU</td>
<td>$\tau$</td>
<td>shear force</td>
<td>lb/ft$^2$</td>
</tr>
<tr>
<td>TAU</td>
<td>$\tau_L$</td>
<td>local laminar shear force</td>
<td>lb/ft$^2$</td>
</tr>
<tr>
<td>Program symbol</td>
<td>Math symbol</td>
<td>Description</td>
<td>Units</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>TAULW</td>
<td>$\tau_{w,L}$</td>
<td>laminar shear force at the wall</td>
<td>lb/ft&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>TAUŁY</td>
<td>$\partial\tau_L/\partial y$</td>
<td>derivative of the laminar shear force</td>
<td>lb/ft&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>TAUTP</td>
<td>$\tau_T$</td>
<td>local turbulent shear force</td>
<td>lb/ft&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>TAUTW</td>
<td>$\tau_{w,T}$</td>
<td>turbulent shear force at the wall</td>
<td>lb/ft&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>TAUTY</td>
<td>$\partial\tau_T/\partial y$</td>
<td>derivative of the turbulent shear force</td>
<td>lb/ft&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>TAUW</td>
<td>$\tau_w$</td>
<td>shear force at the wall</td>
<td>lb/ft&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>THETA</td>
<td>$\theta$</td>
<td>boundary layer momentum thickness</td>
<td>ft</td>
</tr>
<tr>
<td>TURBS</td>
<td>$\frac{\partial\tau_T/\partial y}{\partial \tau_L/\partial y}$</td>
<td>ratio of the turbulent to the laminar shear force derivative</td>
<td>---</td>
</tr>
<tr>
<td>* U</td>
<td>$u$</td>
<td>velocity in the x-direction</td>
<td>ft/sec</td>
</tr>
<tr>
<td>* V</td>
<td>$v$</td>
<td>velocity, $\mathbf{v}$ in Volume I (Appendix C)</td>
<td>ft/sec</td>
</tr>
<tr>
<td>X</td>
<td>$x$</td>
<td>coordinate tangent to the surface</td>
<td>ft</td>
</tr>
<tr>
<td>XI</td>
<td>$x_I$</td>
<td>initial value of $x$ for the calculation of $\eta$</td>
<td>ft</td>
</tr>
<tr>
<td>* XMU</td>
<td>$\mu$</td>
<td>absolute viscosity of air</td>
<td>lb·sec/ft&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>* Y</td>
<td>$y$</td>
<td>coordinate normal to the surface</td>
<td>ft</td>
</tr>
</tbody>
</table>

Subscript notation:

- **e**: evaluated at the boundary layer edge
- **i**: evaluated at the i<sup>th</sup> position within an array
- **I**: initial value
- **L**: laminar
- **o**: initial value
- **T**: turbulent
- **w**: evaluated at the surface
2. METHOD OF SOLUTION

a. Forward Integration

In the following discussion, the equation numbers in parenthesis refer to an appropriate equation in Appendix C of Volume I. The equations solved by this computer program are the equation of state (C-3).

\[ \rho = \frac{P}{RT} \]

Continuity (C-17),

\[
\frac{1}{\frac{\partial u}{\partial y} - \Delta y} \left( \frac{u_i^2}{\Delta y} \left( \frac{1}{P \frac{\partial P}{\partial x} + \frac{1}{r} \frac{\partial r}{\partial x} } \right) + \frac{1}{\rho_1} \left[ - \frac{\partial P}{\partial x} + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) - \frac{\partial \tau_T}{\partial y} \right] \right) \\
- \frac{u_i}{\rho} \left[ \frac{\partial}{\partial y} \left( \frac{\mu \frac{\partial u}{\partial y}}{\sigma} \right) + \frac{\partial \dot{q}_T}{\partial y} + \frac{\partial u}{\partial y} \left( \mu \frac{\partial u}{\partial y} + \tau_T \right) + \frac{\partial P}{\partial x} \right] \frac{\partial u}{\partial y} \\
\left( \frac{\Delta y}{2} \right) \frac{\partial v_{i-1}}{\partial y} \frac{\partial u}{\partial y} \\
\right)
\]

x-momentum (C-6),

\[
\frac{\partial u}{\partial x} = \frac{1}{\rho u} \left[ - \frac{\partial P}{\partial x} + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} + \tau_T \right) \right] \frac{\partial u}{\partial y}
\]

and Energy (C-7):

\[
\frac{\partial I}{\partial x} = \frac{1}{\rho u} \frac{\partial}{\partial y} \left[ \frac{\mu \frac{\partial i}{\partial y}}{\sigma} + \dot{q}_T + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} + \tau_T \right) \right] \frac{\partial I}{\partial y}
\]

In addition, the relationship for turbulent shear stress, Equation (C-39), is required for the computation.

\[
\frac{\partial \tau_T}{\partial y} = \left( \frac{\rho}{\rho_e} \right)^4 \left[ 0.054 \left( \frac{\mu}{\mu_e} \right)^{8.33} \left( \frac{u}{u_e} \right)^{12} \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) \right]
\]

\[
\epsilon_i = \frac{\tau_T}{\frac{\partial u}{\partial y}}
\]

\[
\dot{q}_T = \frac{\epsilon_i}{\sigma} \frac{\partial i}{\partial y}
\]
Since $v$ is expressed as a function of input data, it can be determined explicitly at each point in the boundary layer at the initial or start position.

The $y$ derivatives in the above expressions are calculated using three-point central differences at all positions except at $y = \Delta y$. Since the turbulent velocity and enthalpy profiles change very rapidly near the wall, an off-center method of calculation of derivatives is used at $y = \Delta y$.

\[
\left( \frac{\partial u}{\partial y} \right)_y = \frac{1}{6 \Delta y} (9u_3 - 2u_4 - 6u_2)
\]

\[
\left( \frac{\partial^2 u}{\partial y^2} \right)_y = \frac{1}{6 \Delta y^2} (u_4 - 3u_3 - 9u_2)
\]

With $v$ defined and with $u$, $I$, and $y$ derivatives, $\partial u / \partial x$ and $\partial I / \partial x$ can now be determined. With the $\partial u / \partial x$ and $\partial I / \partial x$ derivatives determined, the $u$ and $I$ profiles at the next station $(x_0 + \Delta x)$ can be obtained by forward integration using:

\[
u_{x+\Delta x} = u_x + \left( \frac{\partial u}{\partial x} \right)_x \Delta x
\]

\[I_{x+\Delta x} = I_x + \left( \frac{\partial I}{\partial x} \right)_x \Delta x
\]

The $y$ derivatives, $v$, $\partial u / \partial x$, and $\partial I / \partial x$ calculations move out from the wall along a line normal to the wall until a specified limit is reached. The profiles are then stepped forward and the calculations repeated.

At each point in the boundary layer, a similarity parameter $\eta$ is calculated as indicated below in subsection b., item (4). A value of $V_{\text{max}}$ is an item of input used to limit the calculation in the $y$-directions, which are to be printed out.

Also calculated at each station $x$ are the displacement and momentum thickness, heating rate, and the laminar and turbulent shear stress at the wall using:

\[
\delta^* = \int_0^\delta \left[ 1 - \frac{\rho u}{\rho_e^* u_e^*} \right] dy
\]

\[
\theta = \int_0^\delta \left[ \frac{\rho u}{\rho_e^* u_e^*} \left( 1 - \frac{u}{u_e^*} \right) \right] dy
\]

1 The notation $x_0$ will be used throughout this section to denote the initial value of $x$. 190
\[ \dot{q}_w = -\frac{1}{\theta_{\text{in}}} \int_0^\delta \left[ \rho u \frac{\partial I}{\partial y} + \rho v \frac{\partial I}{\partial y} - u \tau \right] \, dy \]

\[ \tau_L = \int_0^\delta \left[ \mu \frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right] \, dy \]

\[ \tau_T = \int_0^\delta \frac{\partial \tau_T}{\partial y} \, dy \]

The heating rate and shear at the wall are calculated by an integration over the boundary layer rather than from the definitions because of the greater accuracy which can be obtained with the equation.

b. Numerical Method of Solution

The purpose of this section is to give the procedure used in solving the partial differential equations defined previously including the required numerical approximations.

The following variables and tables are input. Tables are indicated as functions of some independent variable, e.g. \( P = f(x) \) indicates the table of pressure as a function of \( x \).

\[ u = f(y) \text{ at } x_o \]

\[ I = f(y) \text{ at } x_o \]

\[ P = f(x) \]

\[ r = f(x) \]

In addition, the following quantities must be specified:

\[ \Delta x \quad \Delta y \quad \eta_{\text{max}} \quad x_0 \quad x_i \quad s_i \quad x_f \]
(1) Calculation of tabular values

\[
\frac{\partial P}{\partial x_i} = \left[(P_{i+1} - P_i)x_{i+1} - x_i \right] + \left[(P_i - P_{i-1})x_i - x_{i-1} \right] \left[\frac{1}{x_{i+1} - x_i} - \frac{1}{x_i - x_{i-1}}\right]
\]

\[i = 2, 3, \ldots, (NP-1)\]

\[
\frac{\partial P}{\partial x_i} \bigg|_{i=1} = 0
\]

(2) Values obtained from tables (linear interpolation)

\[
P = f(x)
\]

\[
\left(\frac{\partial u}{\partial y}\right)_e = f(x)
\]

\[
\frac{\partial P}{\partial x} = f(x)
\]

\[v_1 = f(x)\]

\[r = f(x)\]

\[
\frac{\partial r}{\partial x} = f(x)
\]

(3) Computation at initial \( x = x_0 \)

\[y_i = (i - 1) \Delta y \quad \text{i = 1, 2, 3, \ldots, 200}\]

\[u_i = u_{i-1} + \left(\frac{\partial u}{\partial y}\right)_e \Delta y \quad \text{i = (LIST + 1) \ldots, 200}\]

\[I_i = I_{\text{LIST}}\]

LIST = number of input values in u and I tables.

\[T_i = \left(I - u^2/2\right)/C_p\]

\[\mu_i = \frac{(2.272 \times 10^{-8} T_i^{-3/2})}{(T_i + 198.6)}\]

\[\rho_i = P_i/(1716 T_i)\]

\[C_p = 6006 \text{ ft}^2/\text{sec}^2\text{R}\]

\[i = 2, 3, \ldots, 200.\]
(4) Calculation of $\delta \eta$

$$\delta \left( \frac{\eta}{\rho} \right) = \left( u_2 r \Delta y \mu e \right)^{3/5} / S^{4/5}$$

$$S = S_l + \rho \mu u \mu e r^2 x_l$$ initially. At $x + \Delta x$

$$S_{x+\Delta x} = \rho \mu u \mu e r^2 \Delta x$$

(5) Calculation of $y$-derivatives and $v$-profile

$$\left. \frac{\partial v}{\partial y} \right|_1 = \frac{v_1 \left( \frac{T_2}{T_1} - 1 \right)}{\Delta y}$$

$$\left. \frac{\partial \mu}{\partial y} \right|_1 = \frac{\mu_{i+1} - \mu_{i-1}}{2 \Delta y}$$

$$\left. \frac{\partial u}{\partial y} \right|_2 = \frac{1}{6\Delta y} (9u_3 - 2u_4 - 6u_2)$$

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_2 = \frac{1}{6\Delta y} (u_4 - 3u_3 - 9u_2)$$

$$\left. \frac{\partial l}{\partial y} \right|_2 = \frac{1}{6\Delta y} (9l_3 - 2l_4 - 6l_2 - l_1)$$

$$\left. \frac{\partial^2 l}{\partial y^2} \right|_2 = \frac{1}{6\Delta y} (l_4 - 3l_3 - 9l_2 + 5l_1)$$

$$\left. \frac{\partial u}{\partial y} \right|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta y}$$

$$\left. \frac{\partial l}{\partial y} \right|_i = \frac{l_{i+1} - l_{i-1}}{2\Delta y}$$

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_i = \frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta y^2}$$
\[ \frac{\partial^2 I_i}{\partial y^2} = \frac{I_{i+1} + I_{i-1} - 2I_i}{\Delta y^2} \]

\[ \frac{\partial \tau_L}{\partial y} = \mu \frac{\partial^2 u}{\partial y^2} + \frac{\partial \mu}{\partial y} \frac{\partial u}{\partial y} \]

\[ \frac{\partial \tau_T}{\partial y} = 0.654 \left( \frac{\mu u_{i+1}}{\mu_i} \right)^{0.833} \left( \frac{u_{i+1}}{u_e} \right)^{12} \left( \frac{T_e}{T_i} \right)^4 \left( \frac{\partial \tau_L}{\partial y} \right)_i \]

\[ \tau'_L = \tau'_L|_{i-1} + \left[ \left( \frac{\partial \tau_L}{\partial y} \right)_{i-1} + \left( \frac{\partial \tau'_L}{\partial y} \right)_i \right] \frac{\Delta y}{2} \]

\[ \tau'_T = \tau'_T|_{i-1} + \left[ \left( \frac{\partial \tau_T}{\partial y} \right)_{i-1} + \left( \frac{\partial \tau'_T}{\partial y} \right)_i \right] \frac{\Delta y}{2} \]

\[ \eta_i = \eta_{i-1} + (\rho_i + \rho_{i-1}) \delta \left( \frac{\eta}{\rho} \right)^{1/2} \]

\[ i = 2, 3, \ldots, \text{MIN}(j, 199) \]

The edge of the boundary layer is defined by \( j = i \) at which the following criteria is satisfied:

\[ \frac{\Delta y}{u} \left[ \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right] \leq 0.0001 \quad \frac{\Delta y}{1} \frac{\partial f}{\partial y} \leq 0.0005 \]

The following are calculated for the range \( - \leq i \leq j \)

\[ \tau_{T_i} = \tau'_{T_i}|_j - \tau'_{T_i}|_j \]

\[ \tau_{L_i} = \tau'_{L_i}|_j - \tau'_{L_i}|_j \]

\[ \tau' = \tau'_{L_i} - \tau'_{T_i} \]

\[ \tau'_L = - \tau'_{T_j} - \tau'_{L_j} \]

\[ \tau' = \tau_{T_i} / (\partial u / \partial y)_i \]
\[
\frac{\partial \epsilon}{\partial y} = \left\{ 0.05 \left( \frac{\rho u_{l+1} \epsilon}{\mu_i} \right)^{0.833} \left( \frac{u_i}{u_e} \right) \left( \frac{T_i}{T_e} \right)^{12} \frac{\partial T_i}{\partial y} \right\} + \epsilon_i \left( \frac{\partial^2 u}{\partial y^2} \right)_i \left( \frac{1}{(\partial u/\partial y)_i} \right)
\]

\[
\frac{d\dot{q}}{dy} = \left( \frac{1}{\sigma} \left( \mu_i + \epsilon_i \right) \frac{\partial T_i}{\partial y} + \left( \mu_i \frac{\partial u}{\partial y} + \epsilon_i \frac{\partial u}{\partial y} \right) \frac{\partial T_i}{\partial y} \right) \left( \frac{1}{\Delta y} \left( u_{i+1} T_{i+1} - u_{i-1} T_{i-1} \right) \right)
\]

\[
\dot{q}_i = \dot{q}_{i-1} + \left( \frac{\partial q}{\partial y}_{i-1} + \frac{\partial q}{\partial y}_i \right) \Delta y / 2
\]

**PART 1.**

\[
\text{u}_i = \frac{2}{\Delta y} \left( \frac{1}{r} \frac{d\rho}{dx} + \frac{1}{\rho} \frac{dp}{dx} \right)
\]

**PART 2.**

\[
\text{u}_i = \frac{1}{\rho u_i} \left( \frac{\partial T_i}{\partial y} + \frac{\partial T_i}{\partial x} - \frac{\partial q}{\partial x} \right)
\]

**PART 3.**

\[
\text{u}_i \left( \frac{2 v_{i-1}}{\Delta y} + \frac{\partial v}{\partial y}_{i-1} \right)
\]

**PART 4.**

\[
v_i = \frac{1}{u_i} \left[ \text{PART 1} + \text{PART 2} - \frac{u_i^2}{(1 - u_i^2/2)} \left( \text{PART 3} - u_i \text{PART 2} \right) - \text{PART 6} \right]
\]

\[
\frac{\partial v}{\partial y} = \frac{2}{\Delta y} \left( v_i - v_{i-1} \right) - \frac{\partial v}{\partial y}_{i-1}
\]

195
(6) Calculation of $x$ derivatives

$$\frac{\partial u}{\partial x} = \text{PART 2}, \quad -\frac{v_i}{u_i} \frac{\partial u}{\partial y_i}$$

$$\frac{\partial I}{\partial x} = \text{PART 3}, \quad -\frac{v_i}{u_i} \frac{\partial I}{\partial y_i}$$

(7) Calculation of $\delta^*$, $\theta$, $Q_w$, $\tau_{w,L}$, $\tau_{w,T}$ at $x$

$$\delta^* = \Delta y \sum_j \left[ 1 - \frac{\rho_i u_i + \rho_{i-1} u_{i-1}}{2 \rho_j u_j} \right]$$

$$\theta = \frac{\Delta y}{2} \sum_j \left[ \frac{\rho_i u_i}{\rho_j u_j} \left( 1 - \frac{u_i}{u_j} \right) + \frac{\rho_{i-1} u_{i-1}}{\rho_j u_j} \left( 1 - \frac{u_{i-1}}{u_j} \right) \right]$$

$$\dot{q}_w = \frac{-\Delta y}{778} \sum_j \left[ \frac{1}{2} \left[ \rho_i u_i \left( \frac{\partial I}{\partial x} \right)_i + \rho_{i-1} u_{i-1} \frac{\partial I}{\partial x}_{i-1} \right] + \rho_i v_i \frac{\partial I}{\partial y_i} - (u_i \tau_i - u_{i-1} \tau_{i-1}) \frac{1}{\Delta y} \right]$$

$$Q_{Q_w} = \frac{1}{778} \left\{ \frac{\tau_j^2}{\mu_i} \frac{\Delta y}{2} \right\}$$

$$\tau_{w,L} = -\tau_{Lj}$$

$$\tau_{w,T} = -\tau_{Tj}$$

(8) Calculation of $u$ and I-profiles at $x + \Delta x$

$$I_{x+\Delta x} = I_x + \frac{\partial I}{\partial x} \Delta x$$

$$u_{x+\Delta x} = u_x + \frac{\partial u}{\partial x} \Delta x$$

$$\{ i = 1, 2, \ldots, j \}$$
\[ u_i|_{x+\Delta x} = u_j|_{x+\Delta x} \]
\[ I_i|_{x+\Delta x} = I_j|_{x+\Delta x} \]  \( i = j + 1, j + 2, \ldots, 200 \)

c. Stagnation Region Calculation

A slight modification to the previous discussed procedure is used to obtain stagnation point profiles. Input profiles are corrected by integrating the \( u \) and \( I \) profiles, but \( x \) (not equal to zero) is not increased during the integration. The profiles are assumed correct when

\[ \left| \frac{\partial u}{\partial x} \right| \frac{u_i}{x} < 0.1 u_{LIST} \text{ for all } i \]

This convergence criteria is obtained with the velocity similarity stated below.

\[ \left( \frac{\partial \rho}{\partial x} \right) = \frac{u e}{u} \frac{\partial u}{\partial x} \]
\[ \frac{\partial I}{\partial x} = 0 \]

Experience has indicated that an enthalpy profile convergence criterion is not necessary.

For the stagnation region, the equations for the velocity and enthalpy profiles are changed to

\[ I_{1,S} = I_{1,S-\Delta S} + \frac{\partial I}{\partial x}_{i} \Delta S \]
\[ u_{1,S} = \left[ u_{1,S-\Delta S} + \frac{\partial u}{\partial x}_{i} \Delta S \right] \frac{x_0}{x_0 + \Delta S} \]

If \( I_i \) or \( u_i \) become negative for some \( i \), \( \Delta S \) is halved and the case is restarted. This may be repeated a maximum of two times for any combination of \( I_i \) or \( u_i < 0 \).

Additional input quantities are \( \Delta S \) and \( MITR \).

MITR is the maximum number of iterations that the program will make in trying to converge on a solution to the stagnation problem.
\( \Delta S \) is a pseudo length similar to \( \Delta x \). The program also uses \( \Delta S \) as a test to determine whether the stagnation point calculation is to be performed or bypassed. For a flat plate case \( \Delta S = 0 \).

Calculations for a hemisphere may be performed near the stagnation point and also along the surface. This is accomplished by the input of \( \Delta S > 0 \) and \( \Delta x > 0 \).

3. **INPUT-OUTPUT DESCRIPTION**

   a. **Data Input Preparation**

      Data are input through the medium of IBM cards. The purpose of this section is to define the required inputs and the form they take on the cards.

      A sample data sheet may be found at the end of this section. Use of this type of form greatly reduces the amount of effort required to obtain results from the non-similar boundary layer program. The data may be keypunched directly from the form.

      All cards except cards 1 and 4 must be keypunched with decimal points and power of ten indicators as follows: if only a decimal point is needed, the number may be punched anywhere within the respective ten column field; however, if the power of ten indicator \( E \) is used, the exponent must be right-adjusted in the field.

      The information on card 1 is alphanumeric and may be keypunched as desired.

      The data on card 4 are all integers and are keypunched as fixed point numbers (i.e., no decimal or power of ten indicator \( E \) is to be used). Each number on card 4 must be right-adjusted in its respective five column field. For example, if \( JOT1 = 1 \) then a one must be keypunched in column five.

      1) **Data Card Formats**

      | Card | Columns | Format | Description |
      |------|---------|--------|-------------|
      | 1    | 1-48    | 8A6    | Any information may be given on this card. It is used to identify the case. |
      | 2    | 1-10    | 7F10   | \( \Delta x \), the increment in the x direction. See the discussion on numerical stability in Appendix C (Volume I) for the criteria to be used in choosing a value. |
      | 2    | 11-20   | 7F10   | \( \Delta y \), the increment in the y direction. Should be chosen to provide at least ten points in the boundary layer for accurate results. |

198
<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21-30</td>
<td>7F10</td>
<td>$\eta_{\text{max}}$, the limiting value on the number of calculations in the y direction.</td>
</tr>
<tr>
<td>2</td>
<td>31-40</td>
<td>7F10</td>
<td>$x_0$, the initial value of $x$. $x_0 &gt; 0$.</td>
</tr>
<tr>
<td>2</td>
<td>41-50</td>
<td>7F10</td>
<td>$x_f$, the final value of $x$.</td>
</tr>
<tr>
<td>2</td>
<td>51-60</td>
<td>7F10</td>
<td>$\sigma$, the turbulent Prandtl number.</td>
</tr>
<tr>
<td>2</td>
<td>61-70</td>
<td>7F10</td>
<td>CHECK, case dependence criteria. If CHECK = 0 then the current case is not dependent on previous calculations. If CHECK = 1 then the current case is dependent on the previous case.</td>
</tr>
<tr>
<td>3</td>
<td>1-10</td>
<td>4F10</td>
<td>$\Delta S$. Calculations may be made for a blunted body. This involves calculating profiles near the stagnation point as well as along the body. This type of problem is indicated by $\Delta S &gt; 0$ and MITR &gt; 0 (card 4). For nonstagnation point calculations, $\Delta S = 0$.</td>
</tr>
<tr>
<td>3</td>
<td>11-20</td>
<td>4F10</td>
<td>Turbulent Prandtl number - must be equal to Prandtl number on card 2.</td>
</tr>
<tr>
<td>3</td>
<td>21-30</td>
<td>4F10</td>
<td>$S_1$, value of $S$ used to initialize $S = S_1 + \frac{\rho_u r^2}{\epsilon} x_1$</td>
</tr>
<tr>
<td>3</td>
<td>41-50</td>
<td>4F10</td>
<td>$x_1$, value of $x$ used to initialize $S = S_1 + \frac{\rho u r^2}{\epsilon} x_1$</td>
</tr>
</tbody>
</table>

The data on card 4 control the number of values to be input to each of the succeeding tables. The values must be greater than zero. There is an upper limit on the number of values that may be input to each table. This limit is given with the description of the table. The limit may be changed by altering DIMENSION statements in the program.

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1-5</td>
<td>1415</td>
<td>JOT1, the number of tabular values in the DJOT and XJOTT tables. $0 &lt; \text{JOT1} \leq 10$.</td>
</tr>
<tr>
<td>4</td>
<td>6-10</td>
<td>1415</td>
<td>LIST, the number of tabular values in the u and I profile tables. $2 \leq \text{LIST} \leq 50$.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>4</td>
<td>11-15</td>
<td>1415</td>
<td>NR, the number of tabular values in the three-dimensional flow parameter table (r), the table of its x-derivatives, and the tables of their corresponding values of x. $2 \leq NR \leq 50$.</td>
</tr>
<tr>
<td>4</td>
<td>16-20</td>
<td>1415</td>
<td>NP, the number of tabular values in the pressure table and the table of corresponding values of x. $2 \leq NP \leq 50$.</td>
</tr>
<tr>
<td>4</td>
<td>21-25</td>
<td>1415</td>
<td>NW, the number of tabular values in the wall enthalpy table and the table of corresponding values of x. $2 \leq NW \leq 50$.</td>
</tr>
<tr>
<td>4</td>
<td>26-30</td>
<td>1415</td>
<td>NDU, the number of tabular values in the $(\partial u/\partial y)_e$ table and the table of corresponding values of x. $2 \leq NDU \leq 50$.</td>
</tr>
<tr>
<td>4</td>
<td>31-35</td>
<td>1415</td>
<td>NV1, the number of tabular values of velocity normal to the surface and the table of corresponding values of x. $2 \leq NV1 \leq 50$.</td>
</tr>
<tr>
<td>4</td>
<td>36-40</td>
<td>1415</td>
<td>MITR, the maximum number of iterations allowed to calculate the profiles for the stagnation region.</td>
</tr>
</tbody>
</table>

Cards 5 and 6 are used to specify different frequencies of output and the corresponding ranges. The maximum number of values in these tables is 10.

2 In the case of tables with more than seven input values, the card number refers to a set of cards (e.g., if ten entries are made in the DJOT table, then card 5 actually refers to two cards).
Example: \( x_0 = 0.5 \) and \( x_f = 1.0 \). Printout is desired every .05 (value of \( x \)) until \( x = .25 \), every .01 until \( x = .3 \), and every .1 until \( x = x_f \). The required input would be:

<table>
<thead>
<tr>
<th>Card</th>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1-5</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1-10</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>.1</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>.25</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>.3</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Card</th>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>---</td>
<td>7F10</td>
<td>Table u-profile = ( f(y) ). The purpose of this table is to provide the program with an initial velocity profile at ( x = x_0 ). Normally, this table must be input and contain at least three values. Exception is made when a case is dependent on a previous case (see Section 2). This table contains LIST tabular values. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>7</td>
<td>1-10</td>
<td>7F10</td>
<td>Velocity at wall.</td>
</tr>
<tr>
<td>7</td>
<td>11-20</td>
<td>7F10</td>
<td>Velocity at ( \Delta y ) from wall.</td>
</tr>
<tr>
<td>7</td>
<td>21-30</td>
<td>7F10</td>
<td>Velocity at ( 2\Delta y ) from wall.</td>
</tr>
<tr>
<td>8</td>
<td>---</td>
<td>7F10</td>
<td>Table ( l )-profile = ( g(y) ). The purpose of this table is to provide the program with an initial total enthalpy profile at ( x = x_0 ). Normally, this table must be input and contain at least three values. Exception is made when a case is dependent on a previous case (see Section 2). This table must contain the same number of values (LIST) as the u-profile table. The maximum number of values in this table is 50. Entries are assumed to be consistent with ( \Delta y ).</td>
</tr>
<tr>
<td>8</td>
<td>1-16</td>
<td>7F10</td>
<td>Total enthalpy at the wall.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>8</td>
<td>11-20</td>
<td>7F10</td>
<td>Total enthalpy at $\Delta y$ from the wall.</td>
</tr>
<tr>
<td>8</td>
<td>21-30</td>
<td>7F10</td>
<td>Total enthalpy at $2\Delta y$ from the wall.</td>
</tr>
<tr>
<td>9</td>
<td>-----</td>
<td>7F10</td>
<td>Table $r = f(x)$. This table contains NR tabular values of $r$, the three-dimensional flow parameter (streamline divergence due to body geometry). The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>10</td>
<td>-----</td>
<td>7F10</td>
<td>Table $(\partial r/\partial x) = f(x)$. This table contains NR tabular values of $(\partial r/\partial x)$. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>11</td>
<td>-----</td>
<td>7F10</td>
<td>Table $x$. This table contains NR tabular values of $x$ corresponding to entries made in the $r$ and $(\partial r/\partial x)$ tables (cards 9 and 10).</td>
</tr>
<tr>
<td>12</td>
<td>-----</td>
<td>7F10</td>
<td>Table $P(x) = f(x)$. This table contains NP tabular values of pressure. This table is numerically differentiated for $(\partial P/\partial x)$. The first value of $(\partial P/\partial x)$ is set to zero. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>13</td>
<td>-----</td>
<td>7F10</td>
<td>Table $x$. This table contains NP tabular values of $x$ corresponding to the entries made in the $P$ table (card 12).</td>
</tr>
<tr>
<td>14</td>
<td>-----</td>
<td>7F10</td>
<td>Table wall enthalpy $= f(x)$. This table contains NW tabular values of the total enthalpy at the wall. This table is used for nonisothermal wall problems. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>15</td>
<td>-----</td>
<td>7F10</td>
<td>Table $x$. This table contains NW tabular values of $x$ corresponding to the entries made in the wall enthalpy table (card 14).</td>
</tr>
<tr>
<td>16</td>
<td>-----</td>
<td>7F10</td>
<td>Table $(\partial u/\partial y)_{e} = f(x)$. This table contains NDU tabular values of $(\partial u/\partial x)$ evaluated at the edge of the boundary layer. For all problems not involving vorticity $(\partial u/\partial x) = 0$. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>17</td>
<td>-----</td>
<td>7F10</td>
<td>Table (x). This table contains ND1 tabular values of (x) corresponding to the entries made in the ((\partial u/\partial y)_e) table (card 16).</td>
</tr>
<tr>
<td>18</td>
<td>-----</td>
<td>7F10</td>
<td>Table (V_w = f(x)). This table contains NV1 tabular values of the velocity normal to the wall. This table is used for mass injection or leakage problems. The maximum number of values in this table is 50.</td>
</tr>
<tr>
<td>19</td>
<td>-----</td>
<td>7F:0</td>
<td>Table (x). This table contains NV1 tabular values of (x) corresponding to the entries made in the (V_w) table (card 18).</td>
</tr>
</tbody>
</table>

20 **NOTE:** In all cases where the values in the above tables are zero, a zero must be entered on the input sheet.

Example: the normal velocity at the wall is zero:

\[
\begin{align*}
V_w \text{ table (card 14)} &: 0.0 \quad 0.0 \quad 0.0 \\
x \text{ table (card 15)} &: 0.0 \quad 1.0 \quad 2.0
\end{align*}
\]

2) **Case Dependence**

The NSBL program has a criterion which allows a case to be dependent on the success or failure of the previous case. This involves the input CHECK and LIST and the test for ERROR.

<table>
<thead>
<tr>
<th>LIST</th>
<th>CHECK</th>
<th>ERROR</th>
<th>Then:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>---</td>
<td>Not dependent on the previous case. program continues.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Dependent on the previous case which worked, program continues, using final profile from previous case.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Dependent on the previous case which failed, so present case stops and program continues to next case.</td>
</tr>
<tr>
<td>0</td>
<td>---</td>
<td>0</td>
<td>Dependent on the previous case which worked, program continues, using final profile from previous case.</td>
</tr>
<tr>
<td>0</td>
<td>---</td>
<td>1</td>
<td>Dependent on the previous case which failed, so present case stops and program continues to next case.</td>
</tr>
</tbody>
</table>
b. Output Description

The printed output consists of the input data and at each printout interval the following are printed as a function of y:

\[
\begin{array}{ccccccc}
y & u & \dot{q} & \partial v / \partial y & v & \partial u / \partial x & T \\
\tau & I & \eta & \partial I / \partial x & \epsilon
\end{array}
\]

The following are printed as a function of the x at the end of each printout interval:

\[
\begin{array}{c}
\dot{q}_w \\
\tau_{w, L} \\
\tau_{w, T} \\
w^* \\
\theta \\
\rho_1 \\
\partial P / \partial x \\
QQW \\
S
\end{array}
\]
c. NSBL Input Form

<table>
<thead>
<tr>
<th>CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE CARD FORMAT 8A6 ALPHANUMERIC ANY SPACE 1-48</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Δx</td>
<td>Δy</td>
<td>η_{max}</td>
<td>x₀</td>
<td>x₁</td>
<td>Prandtl no.</td>
</tr>
<tr>
<td>ΔS</td>
<td>Prandtl no.</td>
<td>S₁</td>
<td>x₁</td>
<td></td>
<td>50 60 70</td>
</tr>
<tr>
<td>JOT1</td>
<td>LIST</td>
<td>NR</td>
<td>NP</td>
<td>NW</td>
<td>NDU</td>
</tr>
<tr>
<td>DJOT 5</td>
<td>10 15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35 40</td>
</tr>
<tr>
<td>XJOTT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u-profile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-profile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dr/dx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pressure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wall enthalpy</td>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(du/dy)_x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v_w</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 1 | 11 | 21 | 31 | 41 | 51 | 61 | 71 |
4. PROGRAMMING INFORMATION

a. Program Flow Chart

```
INPUT
PRINT INPUT

LOOK UP
P, \( \frac{\partial p}{\partial x}, \frac{\partial r}{\partial x}, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial y} \)
AS FUNCTIONS OF \( x \)

CALCULATE
\( \eta, \kappa, \frac{\partial \mu}{\partial \eta}, \frac{\partial u}{\partial \eta}, \frac{\partial v}{\partial \eta}, \frac{\partial w}{\partial \eta}, \frac{\partial T}{\partial \eta}, \frac{\partial T}{\partial r}, \frac{\partial T}{\partial \kappa} \)
FOR ALL POINTS IN THE PROFILE (\( \eta \leq \eta_{\text{max}} \))

CALCULATE
\( \tau_L, \tau_T, \tau, \epsilon, \frac{\partial e}{\partial \tau}, \Delta Q, Q, \text{PART 1, PART 2, PART 3} \)
PART 5, \( v, \frac{\partial v}{\partial y}, \frac{\partial \lambda}{\partial x} \)
FOR ALL POINTS IN PROFILE

PRINT

YES

CALCULATE \( \dot{\alpha}, \tau_w, \delta, \theta \)

NO

PRINT OUTPUT

YES

LAST \( x \)

NO

INCREASE \( x_{\text{old}} \) \( + \Delta x \)

LOOK UP
P, \( \frac{\partial p}{\partial x}, \frac{\partial r}{\partial x}, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial y} \)
AS FUNCTION OF \( x + \Delta x \)

CALCULATE \( u \) AND \( v \) PROFILE
\( \mu, T, \rho \) AT \( x + \Delta x \)
```
SIBFT C /A1781T Deck
C Non-Similar Boundary Layer Problem - Ideal Gas

DIMENSION Y(200), U(200), T(200), V(200), UDPY(200), D2UDY(200), DHYD(201)
10, D2HDY(200), HDX(200), ETA(200), H(200), PT(50), XT(50), DPT(50), LDP(1781004)
28, LP(8), LVIT(8), XVIT(50), XHIT(50), HRE(200), XMU(200), DMUDY(200)
30, WALL(50), XWH(50), LHI(8), JXHI(10), JXHIT(10), DUX(200),
40, DUXY(50), LR(8), RT(50), XT(50), LDR(8), XRT(50), LIT(10), LDU(8), DUT(1781007)
50, ADUT(50), USVE(200), HUSVE(200), TSVE(200), XMUSVE(200), HRO(200),
60, TAU(200), TURB5(200), TAU7(200), TAK(200), TAULP(200), TAU(200)
10, EPS(200), EPSY(200), EP(200), QP(200), Q(200)

INTEGER XLOC
1 FORMAT (7F10.0)
2 FORMAT (29H PRESSURE TABLE LOOK-UP ERROR)
5 FORMAT (25H DPDX TABLE LOOK-UP ERROR)
6 FORMAT (34H WALL-ENTHALPY TABLE LOOK-UP ERROR)
7 FORMAT (1415)
8 FORMAT (15H0 PRESSURE TABLE/19X, 1X, 1X, 1X, 1X, 1X, 1X, 8H PRESSURE)
9 FORMAT (20H WALL-ENTHALPY TABLE/19X, 1X, 1X, 1X, 1X, 1X, 7X, 1X, 13H WALL-ENTHALPY)
10 FORMAT (50M10N-ONE SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS *10X, *10A6, *3H0X=F8.5)
11 FORMAT (1H0, 8X, 1H, 19X, 1H, 19X, 1H, 19X, 1H, 18X, 4DUXY, 17X, 1H, 18X, 4HUX, 9X)
12 FORMAT (1H0, 8X, 1H, 19X, 1H, 19X, 1H, 19X, 1H, 18X, 4DUXY, 17X, 1H, 18X, 4HUX, 9X)
13 FORMAT (2(6E20.6/))
14 FORMAT (27H SOME T(I) IS LESS THAN 0.0)
15 FORMAT (31H SOME U(I) IS LESS THAN (-U(J)))
16 FORMAT (39H SOME H(I) IS LESS THAN OR EQUAL TO 0.0)
17 FORMAT (1H0, 11HTRY NUMBER 11, 30H WILL BE MADE USING DX EQUALS F8.5/1781028)

17 FORMAT (12F20.8)
19 FORMAT (31H R FUNCTION TABLE LOOK-UP ERROR)
200 FORMAT (25H DRDX TABLE LOOK-UP ERROR)
201 FORMAT (50M10N-ONE SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS *10X, *10A6)
202 FORMAT (34H INPUT DATA FOR THE FOLLOWING CASE.../77H (ANY QUANTITY IS NOT SHOWN HERE RETAIN THE VALUES USED IN THE PRECEDING CASE))
203 FORMAT (15H0 CASE CONSTANTS/*5X, 3HUX=E12, 5X, 3HXY=E12, 5X, 6X, 7HETAMA1781037)
204 FORMAT (13H PRINT EVERY 12, 5, 9H UNTIL X=E12, 5)
205 FORMAT (8H case table/19X, 1H, 19X, 1H, 15X, 5HDR/DX)
206 FORMAT (23HOC=WALL FOR THIS PROFILE IS E12, 5, 39H TAU-WALL (LAMINAR))
1 FOR THIS PROFILE IS E12.5/41H TAU-WALL (TURBULENT) FOR THIS PROF1781T042
2LE IS E12.5/32H DELTA STAR FOR THIS PROFILE IS E12.5/ 1781T043
226H THE MOMENTUM THICKNESS IS E12.5/ 47H THE1781T044
2CONVERGENCE CRITERION WAS SATISFIED AFTER 13,9H PROFILES) 1781T045
207 FORMAT(1HO*26X*65H*** THIS CASE WAS NOT RUN DUE TO DEPENDENCE ON P1781T046
1RECEIVING CASE ***} 1781T047
208 FORMAT(3F20.5) 1781T048
209 FORMAT(1HO*11HDA/DY TABLE/19X*1HX*15X*5HDU/DY) 1781T049
210 FORMAT(1H *24HDU/DY TABLE LOOK-UP ERROR) 1781T050
211 FORMAT(3A6) 1781T051
216 FORMAT(1H *24HV(1) TABLE LOOK UP ERROR) 1781T052
217 FORMAT(1HO*10HV(1) TABLE/19X*1HX*16X*4HV(1)) 1781T053
2

PROGRAM CONSTANTS
D jot(1)=0.0
Xjot(1)=0.0
CP = 6006.
DMUDY(1)=0.0
ETA(1)=0.

TABF CALLING SEQUENCES

***PRESSURE

LP(1)=XLOC(LP(1))
LP(2)=XLOC(XT(1))
LP(3)=XLOC(PT(1))
LP(4)=1
LP(5)=1
LP(6)=1
LP(7)=50

***DPDX

LDP(1)=XLOC(LDP(1))
LDP(2)=XLOC(XT(1))
LDP(3)=XLOC(DPT(1))
LDP(4)=1
LDP(5)=1
LDP(6)=1
LDP(7)=50
***R FUNCTION

LR(1)=XLOC(LR(1))
LR(2)=XLOC(XRT(1))
LR(3)=XLOC(RT(1))
LR(4)=1
LR(5)=1
LR(6)=1
LR(7)=50

***DRDX

LDR(1)=XLOC(LDR(1))
LDR(2)=XLOC(XRT(1))
LDR(3)=XLOC(DRT(1))
LDR(4)=1
LDR(5)=1
LDR(6)=1
LDR(7)=50

***WALL ENTHALPY

LH(1)=XLOC(LH(1))
LH(2)=XLOC(XWH(1))
LH(3)=XLOC(WALLH(1))
LH(4)=1
LH(5)=1
LH(6)=1
LH(7)=50

***DUDX

LDU(1)=XLOC(LDU(1))
LDU(2)=XLOC(XDUT(1))
LDU(3)=XLOC(DUT(1))
LDU(4)=1
LDU(5)=1
LDU(6)=1
LDU(7)=50
```c
###V(1)
LV1(1)=XLOC(LV1(1))
LV1(2)=XLOC(XVIT(1))
LV1(3)=XLOC(VIT(1))
LV1(4)=1
LV1(5)=1
LV1(6)=1
LV1(7)=50

18 READ (5*211)(TITLE(I),I=1,8)
READ(5*1)DX,DY,ETAMAX,STARTX,ENDX,PR,CHECK,DS,PRT,S1,X1
X=STARTX
LENGTH=ENDX/DX+1
LX=1
XJOT=STARTX
TIMES=0.0
NTRY=1
READ (5*7)JOT,LIST,NR,NP,NW,NDU,NV1,MITR
KK=200
LIST#4#LIST
IF(LIST)185*185*180

180 J=LIST4
185 JOT=JOT+1
READ (5*1)(DJOT(I),I=2#JOT)
READ (5*1)(XJOT(I),I=2#JOT)
WRITE (6#201)(TITLE(I),I=1,8)
WRITE (6#202)
WRITE (6#203)DX,DY,ETAMAX,STARTX,ENDX,PR
WRITE (6#204)(DJOT(I),XJOTT(I),I=2#JOT)
IF(LIST)1252*252*93

93 READ (5*1)(U(I),I=1#LIST)
READ (3*1)(H(I),I=1#LIST)
IF(NR)81*81*80

C
R TABLE AND DR/DX TABLE

80 READ (5*1)(RT(I),I=1#NR)
READ (5*1)(DRT(I),I=1#NR)
READ (5*1)(XRT(I),I=1#NR)
WRITE (6#205)
WRITE (6#208)(XRT(I),RT(I),DRT(I),I=1#NR)

81 IF(NP) 94*94*96
```

```
1781T125
1781T126
1781T127
1781T128
1781T129
1781T130
1781T131
1781T132
1781T133
1781T134
1781T135
1781T136
1781T137
1781T138
1781T139
1781T140
1781T141
1781T142
1781T143
1781T144
1781T145
1781T146
1781T147
1781T148
1781T149
1781T150
1781T151
1781T152
1781T153
1781T154
1781T155
1781T156
1781T157
1781T158
1781T159
1781T160
1781T161
1781T162
1781T163
1781T164
1781T165
1781T166
```
C
C PRESSURE TABLE
C
96 READ (5*1)(PT(I),I=1,NP)
97 READ (5*1)(WALLH(I),I=1,NW)
98 READ (5*1)(DUT(I),I=1,NDU)
99 READ (5*1)(V1T(I),I=1,NV1)
C
96 READ (5*1)(PT(I),I=1,NP)
97 READ (5*1)(WALLH(I),I=1,NW)
98 READ (5*1)(DUT(I),I=1,NDU)
99 READ (5*1)(V1T(I),I=1,NV1)
C
96 READ (5*1)(PT(I),I=1,NP)
97 READ (5*1)(WALLH(I),I=1,NW)
98 READ (5*1)(DUT(I),I=1,NDU)
99 READ (5*1)(V1T(I),I=1,NV1)
C
96 READ (5*1)(PT(I),I=1,NP)
97 READ (5*1)(WALLH(I),I=1,NW)
98 READ (5*1)(DUT(I),I=1,NDU)
99 READ (5*1)(V1T(I),I=1,NV1)
C
C CALCULATE DPDX TABLE
C
DPT(I)=0.0
DO 100 I=2,NP
DELI=XT(I+1)-XT(I)
DEL2=XT(I)-XT(I-1)
DPT(I)=((PT(I+1)-PT(I))*DELI2/DELI1+(PT(I)-PT(I-1))*DELI/DELI2)/DEL3
100 CONTINUE
C
C WALL ENTHALPY TABLE
C
94 IF(NW)98,98,97
94 IF(NW)98,98,97
94 IF(NW)98,98,97
94 IF(NW)98,98,97
C
C DU/DY TABLE
C
95 IF(NDU)985,985,980
96 WRITE (6*9)
97 WRITE (6*17)(WALLH(I),I=1,NW)
98 WRITE (6*209)
C
C DU/DY TABLE
C
95 IF(NDU)985,985,980
96 WRITE (6*9)
97 WRITE (6*17)(WALLH(I),I=1,NW)
98 WRITE (6*209)
C
C DU/DY TABLE
C
95 IF(NDU)985,985,980
96 WRITE (6*9)
97 WRITE (6*17)(WALLH(I),I=1,NW)
98 WRITE (6*209)
C
C DU/DY TABLE
C
95 IF(NDU)985,985,980
96 WRITE (6*9)
97 WRITE (6*17)(WALLH(I),I=1,NW)
98 WRITE (6*209)
WRITE (6,17) (V1T(I),V1T(I),I=1,IV1)

WRITE OUT VELOCITY AND ENTHALPY PROFILES

WRITE (6,218) (U(I),H(I),I=1,LIST)

218 FORMAT (31H0VELOCITY AND ENTHALPY PROFILES/19X,1HU,19X,1HH/(2F2.)

READ TABLES AND/OR SET UP INITIAL PROFILE

——AS FAR AS READ IN

TEST CHECK AND ERROR****LIST IS POSITIVE

LOUP=0

ITTER=0

99 IF(CHECK) 260,260,251

251 IF(ERROR) 260,260,253

TEST ERROR****LIST IS ZERO

252 IF(ERROR) 953,953,253

253 WRITE (6,207)

ERROR=1.

GO TO 18

260 ERROR=0*0

LP(8)=0

PRESS=TAB(X,LP(1))

IF(LP(8)-1)990,101,990

101 LDP(8)=0

DPDX=TAB(X,LDP(1))

IF(LDP(8)-1) 993,102,993

102 LR(8)=0

R=TAB(X,LR(1))

IF(LR(8)-1) 996,122,996

122 LDR(8)=0

DRDX=TAB(X,LDR(1))

IF(LDR(8)-1) 997,89,997

89 LDU(8)=0

DUDY=TAB(X,LDU(1))

IF(LDU(8)-1) 998,87,998

87 LV1(8)=0

V(1)=TAB(X,LV1(1))

IF(LV1(8)-1) 994,88,994

1781T208

1781T209

1781T210

1781T211

1781T212

1781T213

1781T214

1781T215

1781T216

1781T217

1781T218

1781T219

1781T220

1781T221

1781T222

1781T223

1781T224

1781T225

1781T226

1781T227

1781T228

1781T229

1781T230

1781T231

1781T232

1781T233

1781T234

1781T235

1781T236

1781T237

1781T238

1781T239

1781T240

1781T241

1781T242

1781T243

1781T244

1781T245

1781T246

1781T247

1781T248

1781T249
88 DO 95 I=1,200
   Y1=I-1
   Y(I)=Y1*DY
   IF(I.NE.LIST) 891,891,890
890 U(I)=U(I-1)+DUDYL*DY
   H(I)=H(LIST)
891 T(I)=(H(I)-U(I)**2/2.0)/CP
   XMU(I)=2.272E-08*T(I)**1.5/(T(I)+198.6)
   RHO(I)=PRESS/(1716*T(I))
   CONTINUE
C
   COMPUTE INITIAL VALUE OF S
   S=S1+RHO(J)*U(J)*XMU(J)*R**2*X1
C
   LOOP TO SAVE INITIAL PROFILES
C
   953 PRESVE=PRESS
   DPDXSV=DPDX
   RSVE=R
   DRDXSV=DRDX
   DUDYSV=DUDYL
   SSVE=S
   VISVE=V(I)
   DO 954 I=1,200
     USVE(I)=U(I)
     HSVE(I)=H(I)
     TSVE(I)=T(I)
     XMUSEVE(I)=XMU(I)
     RHOSEVE(I)=RHO(I)
   954 CONTINUE
C
   CALCULATE Y DERIVATIVES, V PROFILE, DVDY
C
   NLOOP=0
   NDO=ETAMAX
   NDO1=NDO
   29 DELETA=U(J)*R*DY*XMU(J)**.6/S**.8
      NI=0
      NJ=0
      DVDY(I)=V(I)*((T(2)/T(I)-1.0))/DY
LOOP=1
TAULY(1)\*DPDX
TAUTY(1)\=0
TAULP(1)\=0
TAUTP(1)\=0
DG(1)\=0
QP(1)\=0
IF(NLOOP)320\*320\*310
310 CONTINUE
NDO=199
320 CONTINUE
DO 40 I=2,NDO
TURBS(I)=.054*(RHO(I)\*U(1)\*Y(I)/XMU(I))**.833*(U(I)/U(J))**12
1*(T(I)/T(J))**4
DMUDY(I)=(XMU(J)+1-XMU(J-1))/(2\*DY)
IF(I-2)5000\*5000\*5001
5000 CONTINUE
DUDY(2)=(9\*U(3)-2\*U(4)-6\*U(2))\/6\(/DY
DHDY(2)=(9\*H(3)-2\*H(4)-6\*H(2)-H(1))\//6\(/DY
D2UDY(2)=(U(4)+3\*U(3)-9\*U(2))\//6\(/DY**2
D2HDY(2)=(H(4)+3\*H(3)-9\*H(2)-5\*H(1))\//6\(/DY**2
GO TO 5002
5001 CONTINUE
DUDY(I)=U(I+1)-U(I-1)/(2\*DY)
DHDY(I)=H(I+1)-H(I-1)/(2\*DY)
D2UDY(I)=(U(I+1)+U(I-1)-2\*U(I))\//DY**2
D2HDY(I)=(H(I+1)+H(I-1)-2\*H(I))\//DY**2
5002 CONTINUE
TAULY(I)=XMU(I)*D2UDY(I)+DMUDY(I)*DUDY(I)
TAUTY(I)=TURBS(I)*TAULY(I)
TAULP(I)=TAULP(I-1)+(TAULY(I-1)+TAUTY(I-1))*DY/2
TAUTP(I)=TAUTP(I-1)+(TAUTY(I-1)+TAUTY(I))*DY/2
31 ETA(I)=ETA(I-1)+(RHO(I)+RHO(I-1))*DELETA
C C ERROR CRITERIA
C
35 IF(U(I)+U(J)) 63,37,37
37 IF(H(I)) 64,64,41
41 IF(T(I)) 65,65,43
43 IF(X) 430,430,44
430 IF(I-LIST4) 40,435,435
EPSY(I)=(TURBS(I)+TAULY(I)-EPS(I)*D2UDY(I))/DUDY(I)
DQ(I)=1.+PR*(XMU(I)+EPS(I)*D2HDY(I)+DUMU(I)+EPSY(I))
1DHDY(I)=(U(I+1)+U(I-1)-2*U(I))/DY
QP(I)=QP(I-1)+(DQ(I-1)+DQ(I))*DY/2
PART1=U(I)**2*(DRDX/R+DPDX/Press)
PART2=1./RHO(I)/U(I)*(TAULY(I)+TAUTY(I)-DPDX)
PART3=1./RHO(I)/U(I)*(DQ(I)+U(I+1)-2*U(I)-U(I-1))
1/2*DY
PART6=U(I)*(2*V(I-1)/DY+DVDY(I-1))
V(I)+1./DUDY(I)-2*U(I)/DY)*(PART1+U(I)*PART2-U(I)**2/
1(H(I)-U(I)**2/2.)*(PART3-U(I)*PART2)-PART6)
DVDY(I)=2./DY*(V(I)-V(I-1))-DHDY(I-1)
DHUDX(I)=PART2-V(I)/U(I)*DUDY(I)
DHUDX(I)=PART3-V(I)/U(I)*DHDY(I)
3049 CONTINUE
IF(DS)=50*50+3
3 IF(X-STARTX>300*300*50
300 I=1
4 IF (ABS(DUDY(I)-U(I)/X)>1*U(LIST)) 24, 24, 25
24 I=I+1
4 IF (I-LIST) 4, 4* 50
25 DO 21 I=1+LIST
21 H(I)=H(I)+DHUDX(I)*D
UI(I)=(U(I)+DUDX(I)*D)*(STARTX/(STARTX+DS))
IF (H(I)) 70*70*71
70 DSD=DS/2.
LOUP=LOUP+1
IF(LOUP-2)=75*75*1007
C ********** STATEMENT BELOW IS IN ERROR **********
75 WRITE (6)*1015) DS
1015 FORMAT(88H DURING THE STAGNATION SEARCH A NEGATIVE H WAS ENCOUNTERED)
1ED, WE WILL RUN AGAIN USING DS= F8*6)
1ED, WE WILL RUN AGAIN USING DS= F8*6)
GO TO 1003
71 IF(U(I))72*21, 21
72 DS=DS/2.
LOUP=LOUP+1
IF(LOUP-2)=76*76*1007
C ********** STATEMENT BELOW IS IN ERROR **********
76 WRITE (6)*1016) DS
1016 FORMAT(88H DURING THE STAGNATION SEARCH A NEGATIVE U WAS ENCOUNTERED)
1ED, WE WILL RUN AGAIN USING DS= F8*6)
GO TO 1003
THETA = THETA * DY / 2.0

C PRINT VARIABLES
C
55 WRITE (6, 10) (TITLE(I), I = 1, 8), X
WRITE (6, 11)
WRITE (6, 12) (D(I), U(I), Q(I), D(V)(I), D(V)(I), T(I), TAU(I), H(I)), THETA, ITER
WRITE (6, 206) QW, TAUW, TAUt, DELSTR, THETA, ITER
WRITE (6, 2060) RO, LIST, DPDX
2060 FORMAT (25H QW = E14.8, 5X, 7HDPDX = E14.8)
WRITE (6, 2060) QW
2060 FORMAT (25H QW FOR THIS PROFILE IS E12.5)

C STEP UP X AND READ NEW X-DEPENDENT VARIABLES
C
22 TIMES = TIMES + 1,
X = STARTX + TIMES * DX
LP(8) = 0
PRESS = TAB(X, LP(1))
IF(LP(8) - 1) 990, 104, 990
104 LDP(8) = 0
DPDX = TAB(X, LDP(1))
IF(LDP(8) - 1) 993, 110, 993
110 LR(8) = 0
R = TAB(X, LR(1))
IF(LR(8) - 1) 996, 123, 996
123 LDR(8) = 0
DRDX = TAB(X, LDR(1))
IF(LDR(8) - 1) 997, 90, 997
90 LH(8) = 0
H(1) = TAB(X, LH(1))
IF(LH(8) - 1) 995, 109, 995
109 LDU(8) = 0
1781T457
1781T458
1781T459
1781T460
1781T461
1781T462
1781T463
1781T464
1781T465
1781T466
1781T467
1781T468
1781T469
1781T470
1781T471
1781T472
1781T473
1781T474
1781T475
1781T476
1781T477
1781T478
1781T479
1781T480
1781T481
1781T482
1781T483
1781T484
1781T485
1781T486
1781T487
1781T488
1781T489
1781T490
1781T491
1781T492
1781T493
1781T494
1781T495
1781T496
1781T497
1781T498
DUDYL=TAB(X*LDU(1))
IF(LDU(8)-1) 998*1079,998
1079 LV1(8)=0
V(1)=TAB(X*LV1(1))
IF(LV1(8)-1)994*1080,994
C
C INTEGRATION LOOP
C
C ----MOVE X-DEPENDENT VARIABLES TO NEW PROFILE
1080 S=S+RHO(J)*XMU(J)*U(J)*R**2*DX
DO 124 I=1,200
IF (I-I) 1084, 1084, 1081
1001 IF (I-K) 1082, 1082, 1083
1082 H(I)=H(I)+DMDX(I)*DX
U(I)=U(I)+DUDX(I)*DX
GO TO 1084
1083 H(I) = H(K)
U(I) = U(I-1) + DUDY*DY
1084 T(I)=(H(I)-U(I)**2/2)/CP
IF(T(I)>65.65*1085
1085 XMU(I)=2*72E-08*T(I)**1.5/(T(I)+198.6)
RHO(I)=PRESS/(1716*T(I))
124 CONTINUE
GO TO 29
C
C ERROR INDICATORS
C
63 WRITE (6*14)
GO TO 1000
64 WRITE (6*15)
GO TO 1000
65 WRITE (6*13)
GO TO 1000
990 WRITE (6*2)
GO TO 999
993 WRITE (6*5)
GO TO 999
994 WRITE (6*216)
GO TO 999
995 WRITE (6*6)
GO TO 999
17817499
17817500
17817501
17817502
17817503
17817504
17817505
17817506
17817507
17817508
17817509
17817510
17817511
17817512
17817513
17817514
17817515
17817516
17817517
17817518
17817519
17817520
17817521
17817522
17817523
17817524
17817525
17817526
17817527
17817528
17817529
17817530
17817531
17817532
17817533
17817534
17817535
17817536
17817537
17817538
17817539
996 WRITE (6,19)
997 GO TO 999
998 WRITE (6,210)
999 WRITE (6,10) (TITLE(I),I=1,8)*X
   WRITE (6,11)
   WRITE (6,12) (Y(I),U(I),G(I),DVY(I),V(I),DUDX(I), T(I),TAU(I),H(I))
   1*ETA(I),DHDX(I),EPS(I),I=1,KK)
1000 WRITE (6,10) (TITLE(I),I=1,8)*X
   WRITE (6,11)
   WRITE (6,12) (Y(I),U(I),G(I),DVY(I),V(I),DUDX(I), T(I),TAU(I),H(I))
   1*ETA(I),DHDX(I),EPS(I),I=1, KK)
C
DECREASE DX AND RESTORE INITIAL PROFILES
C
IF (NTRY=3) 1001, 1007 * 1007
1001 NTRY=NTRY+1
   STOP
   X=STARTX
   XJOT=STARTX
   LX=1
   TIMES=0.
   DX=DX/2.
   DS=0.
   WRITE (6,16) NTRY,DX
   IF (LIST) 1003 * 1003 * 1002
C
LOOP TO RESTORE INITIAL PROFILES
C
1002 J=LIST
1003 PRESS=PRESVE
   DPDX=DPDXSV
   RD=RSVE
   DRD=DRDSE
   DUDY=DUDYSV
   S=SVSE
   V(I)=VISVE
   DO 1004 I=1,200
   U(I)=USVE(I)
H(I)=HSVE(I)
T(I)=TSVE(I)
XMU(I)=XMUSVE(I)
RHO(I)=RHOSVE(I)

1004 CONTINUE
GO TO 29
END

1781T582
1781T583
1781T584
1781T585
1781T586
1781T587
1781T588

000388
REFERENCES


5. Hilsenrath, J.; and Beckett, C. W.: Thermodynamic Properties of Argon-Free Air (0.78847 N₂, 0.21153 O₂) to 15,000 °K. NBS Report No. 3991, July 1955.


This report presents a combined analytical and experimental investigation of turbulent heat transfer on basic and composite configurations at hypersonic speeds. The analytical results are presented in Volume I, the experimental results, including data-theory comparisons, are presented in Volume II, and computer programs incorporating the analytical methods described herein are presented in Volume III of this report.

The two heat-transfer prediction methods programmed are the laminar-turbulent $\rho_r \mu_r$ momentum integral method and the turbulent nonsimilar method. This volume of the report describes the numerical method and presents flow charts, program listings, input forms, and a description of the output for each program. The programs are written in Fortran IV language for operation on the IBM 7094.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonsimilar Boundary Layer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Momentum Integral</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerodynamic Heating</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laminar Flow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turbulent Flow</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>