

AFFDL-TR-67-144
Volume III

AD 667804

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

THE BOEING COMPANY

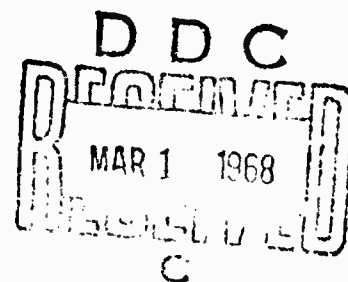
TECHNICAL REPORT AFFDL-TR-67-144, VOLUME III

December 1967

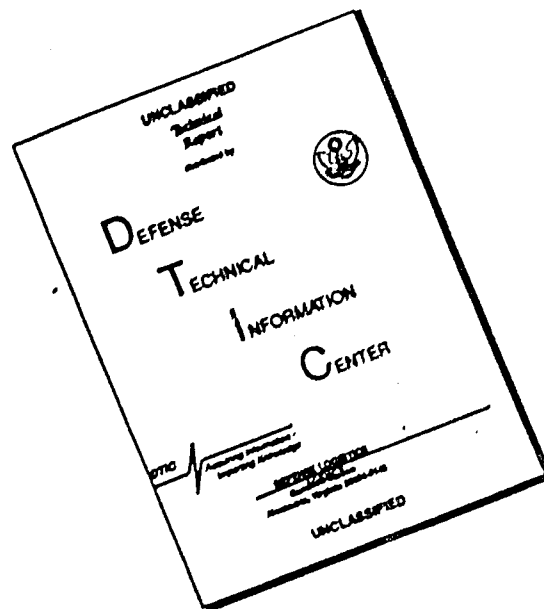
This document has been approved for public release and sale;
its distribution is unlimited.

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



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.

ACCESSION No.	
CPSTY	WHITE SECTION ☒
DOC	DIFF SECTION ☐
GRANDUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DIST.	AVAIL. and/or SPECIAL
/	

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. Deriugin, 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 ~~over~~ 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.

Philip P. Antonatos

PHILIP P. ANTONATOS

Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

ABSTRACT

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

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

TABLE OF CONTENTS (Continued)

	Page
30) TABI18	67
31) TABI19	67
32) TABL20	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. $\rho_r \mu_r$ Program Listing	110
b. $\rho_r \mu_r$ Deck Setup	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 $x = x_0$	192
4) Calculation of $\delta \eta$	193
5) Calculation of y-derivatives and v-profile	193
6) Calculation of x derivatives	196
7) Calculation of δ^* , θ , Q_w , τ_w , L , and $\tau_{w,T}$ at x	196
8) Calculation of u and I-profiles at $x + \Delta x$	196
c. Stagnation Region Calculation	197

TABLE OF CONTENTS (Concluded)

	Page
3. INPUT-OUTPUT DESCRIPTION	198
a. Data Input Preparation	198
1) Data card formats	198
2) Case dependence	203
b. Output Description	204
c. NSBL Input Form	205
4. PROGRAMMING INFORMATION	206
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 $\rho_r \mu_r$ 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.

SECTION II

THE LAMINAR-TURBULENT MOMENTUM INTEGRAL COMPUTER PROGRAM ($\rho_r \mu_r$)

1. DISCUSSION

The computer program described in this report is based on the Hanks $\rho_r \mu_r$ 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 (Ω). 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

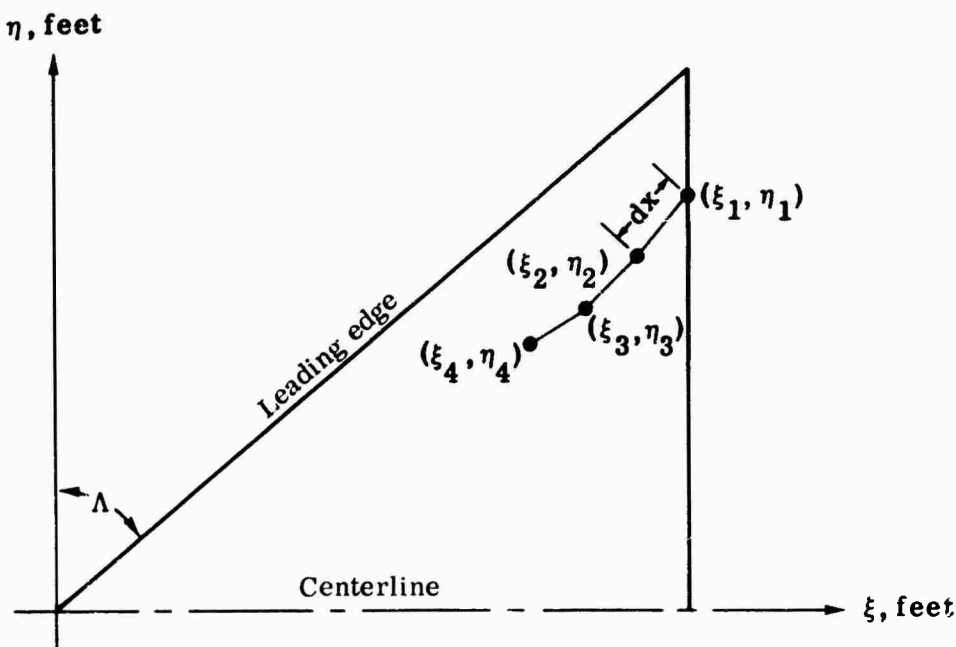
The $\rho_r \mu_r$ integral program is actually four programs, any of which can be operated separately. The first program, Program A, calculates heating rates for any body shape at any flight or ground-facility test condition given the external flow properties, gas properties at the wall, gas properties 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 (ξ, η) 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 axis-symmetric 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 $\delta > 0$

$$\frac{C_P}{\sin^2 \delta} = 1.05 + \left[1.1025 + \frac{4}{(M_\infty \sin \delta)^2} \right]^{1/2} - \frac{1.278 \sin^2 \delta}{M_\infty^{0.6}} \quad (1)$$

and for $\delta \leq 0$

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

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

$$C_P = K \sin^2 \delta \quad (3)$$

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

$$u_e = u_\infty \left[1 - \left(\frac{P}{P_\infty} - 1 \right) \frac{1}{\gamma M_\infty^2} \right]^{1/2} \quad \text{if } P_{\text{initial}} \leq .528 P_o \quad (4)$$

and

$$u_e = a^* \frac{x}{x_{\text{sonic point}}} \quad \text{if } P_{\text{initial}} > .528 P_o \quad (5)$$

$x < x_{\text{sonic point}}$

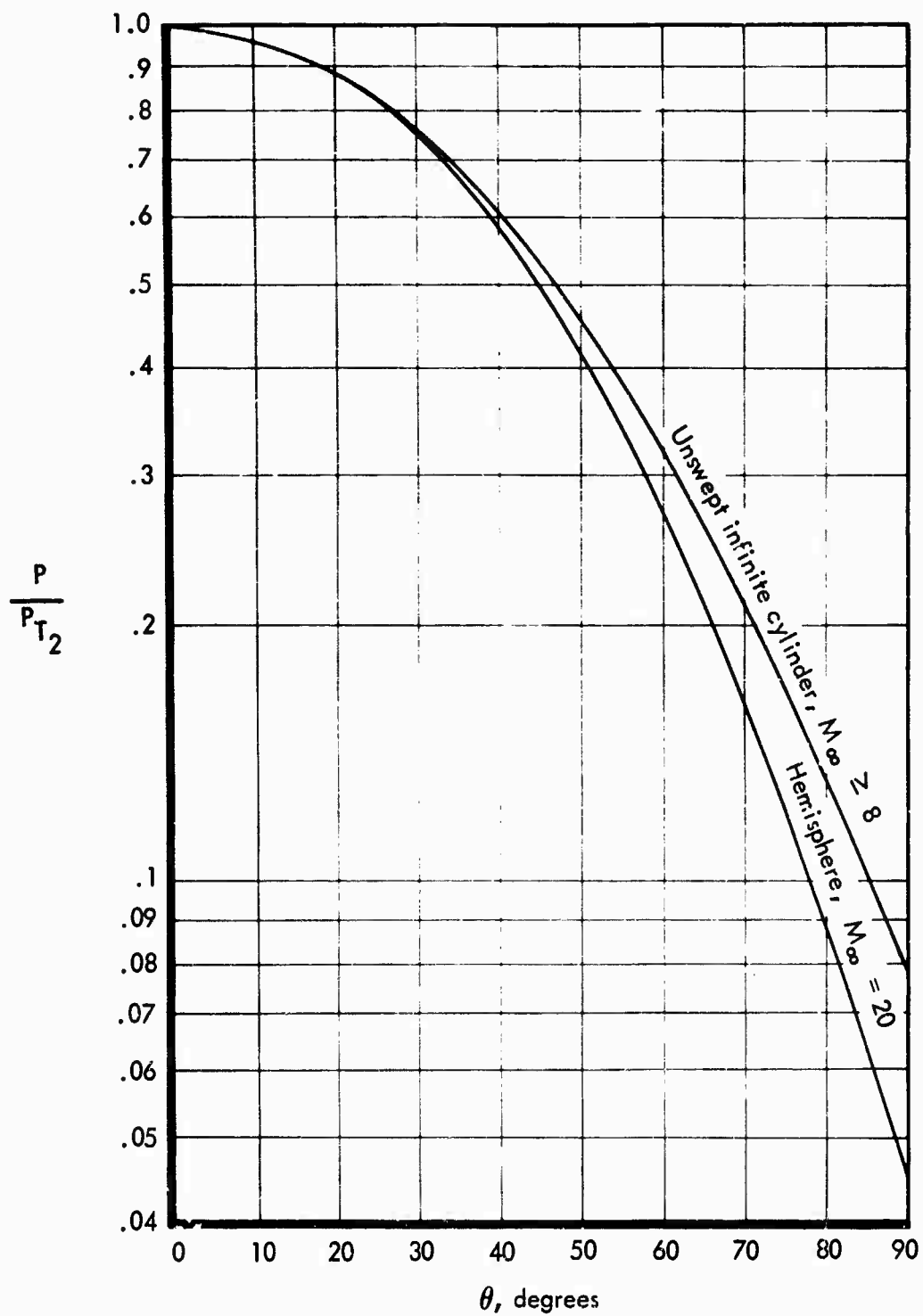


Figure 1: PRESSURE DISTRIBUTION ON A HEMISPHERE AND UNSWEPT INFINITE CYLINDER (REF. 3)

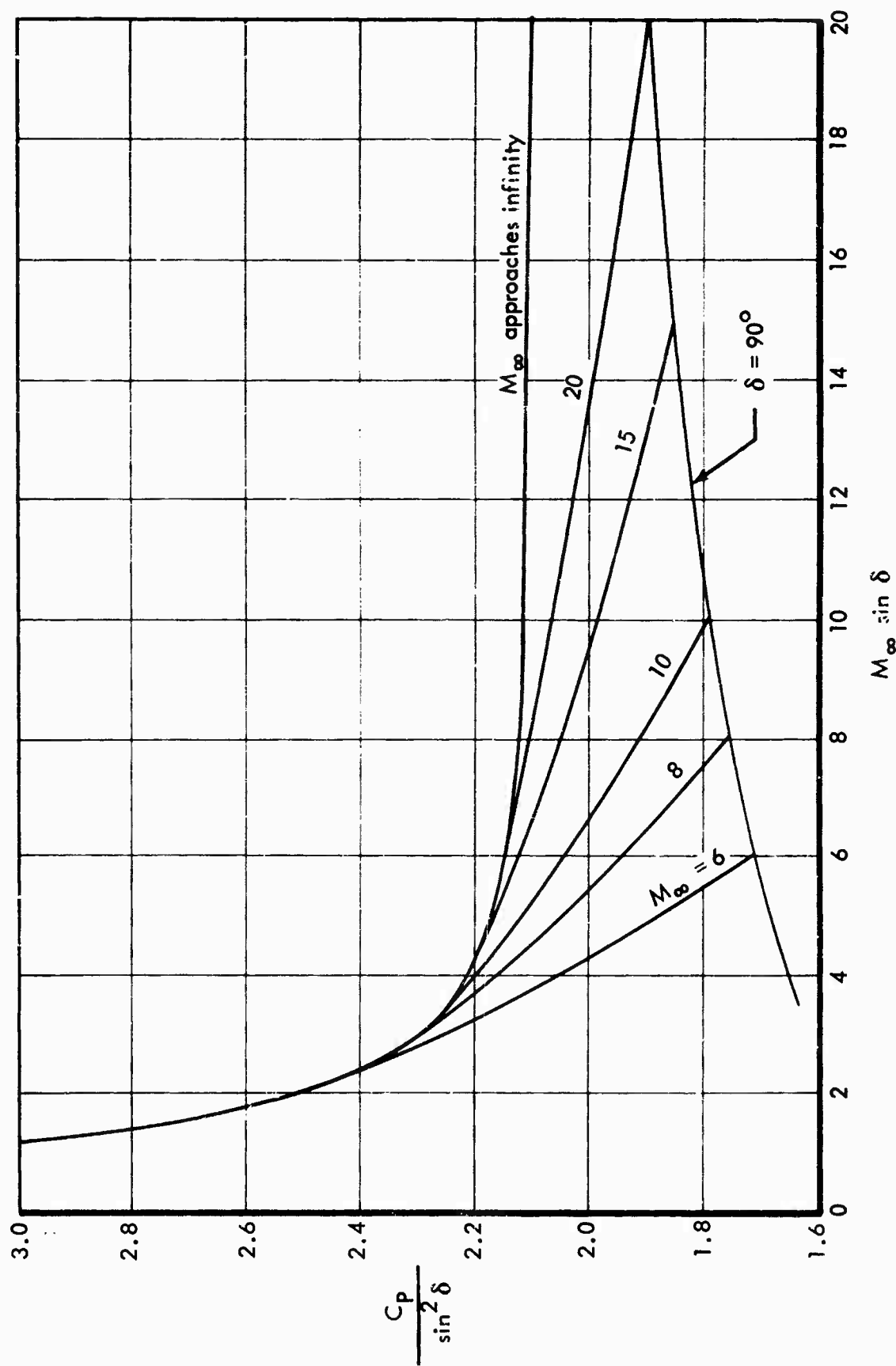


Figure 2: BOEING MODIFIED NEWTONIAN HYPERSONIC PRESSURE COEFFICIENTS

and

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

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

$$u_e = u_\infty \left[1 - \frac{\delta^2}{5600} \right] \quad (7)$$

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 streamline angles on the swept cylinder are obtained from

$$\tan \theta_e = \frac{v_e}{u_\infty \sin \Lambda} \quad (8)$$

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

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

where

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

$\left(\frac{d\theta}{d\epsilon} \right)_{CL}$ is the gradient of the streamline angle with respect to the ray angle ϵ

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.

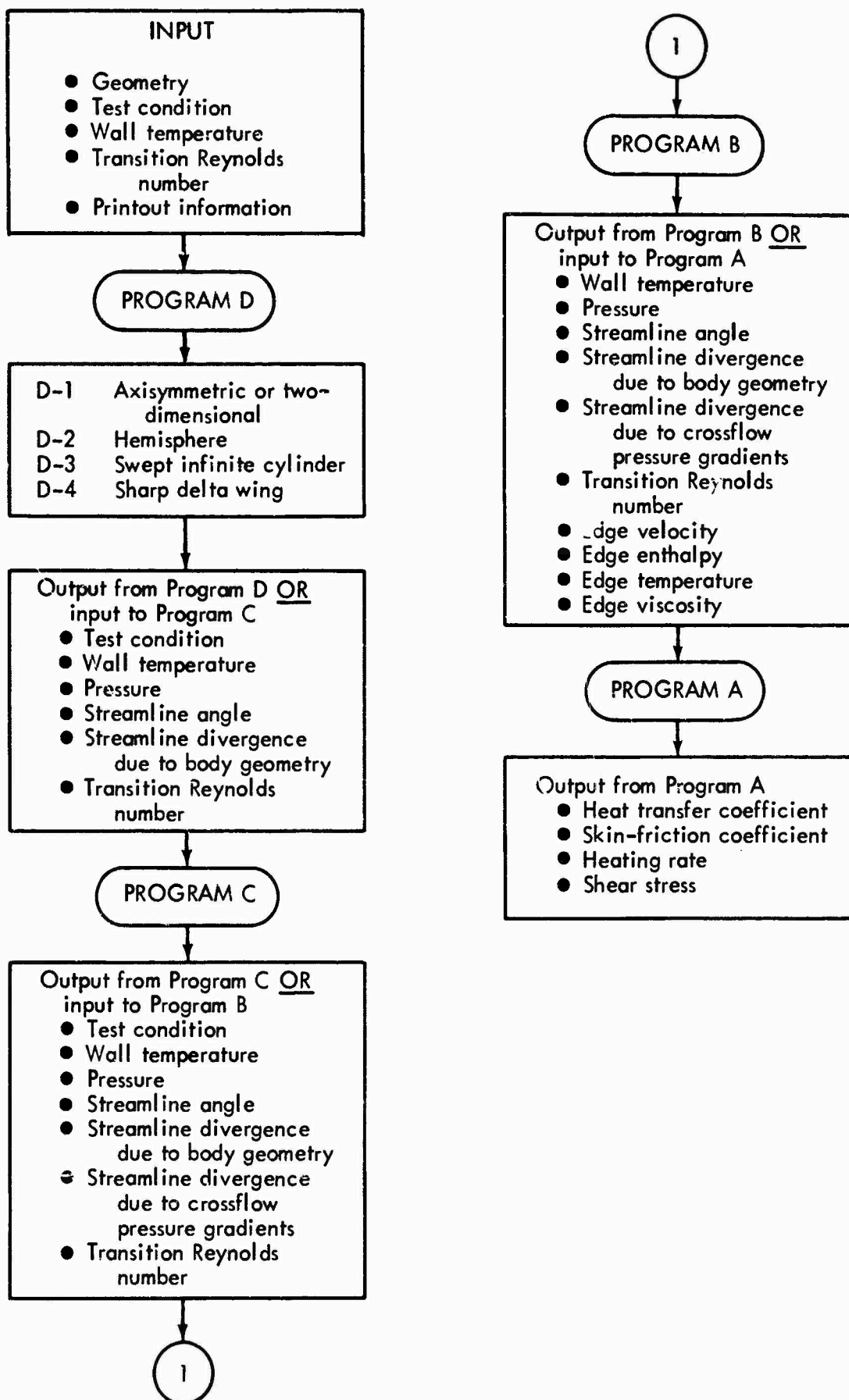
Program D is easily modified and extended as more information becomes available, or as certain specific shapes become important.

b. $\rho_r \mu_r$ 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

a^*	sonic velocity
C_p	pressure coefficient
f	streamline divergence due to crossflow pressure gradients
M_∞	free stream Mach number
P	local pressure
P_o	stagnation point pressure
P_{T_2}	stagnation point or stagnation line pressure
r	streamline divergence due to body geometry
u_e	boundary layer edge velocity
u_∞	free stream velocity
v_e	spanwise velocity at the boundary layer edge
x	distance along a streamline
δ	local flow deflection angle measured from the free stream velocity vector
Δ	total streamline divergence, includes body geometry and crossflow pressure gradients (rf)
ϵ	ray angle on a delta wing
θ	angular location



θ_e local streamline angle

Λ leading edge sweep angle

(ξ, η) coordinate system

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

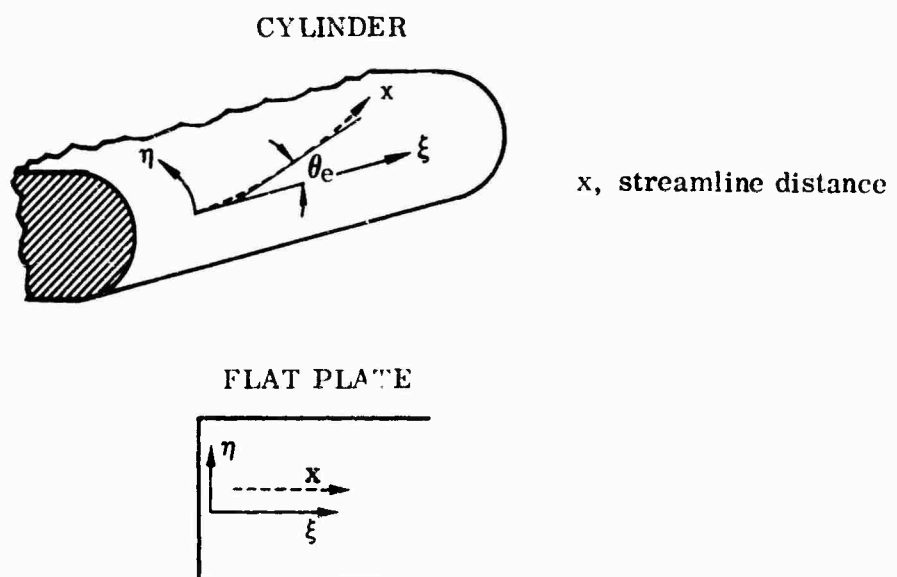
2. $\rho_r \mu_r$ PROGRAM DESCRIPTION

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.

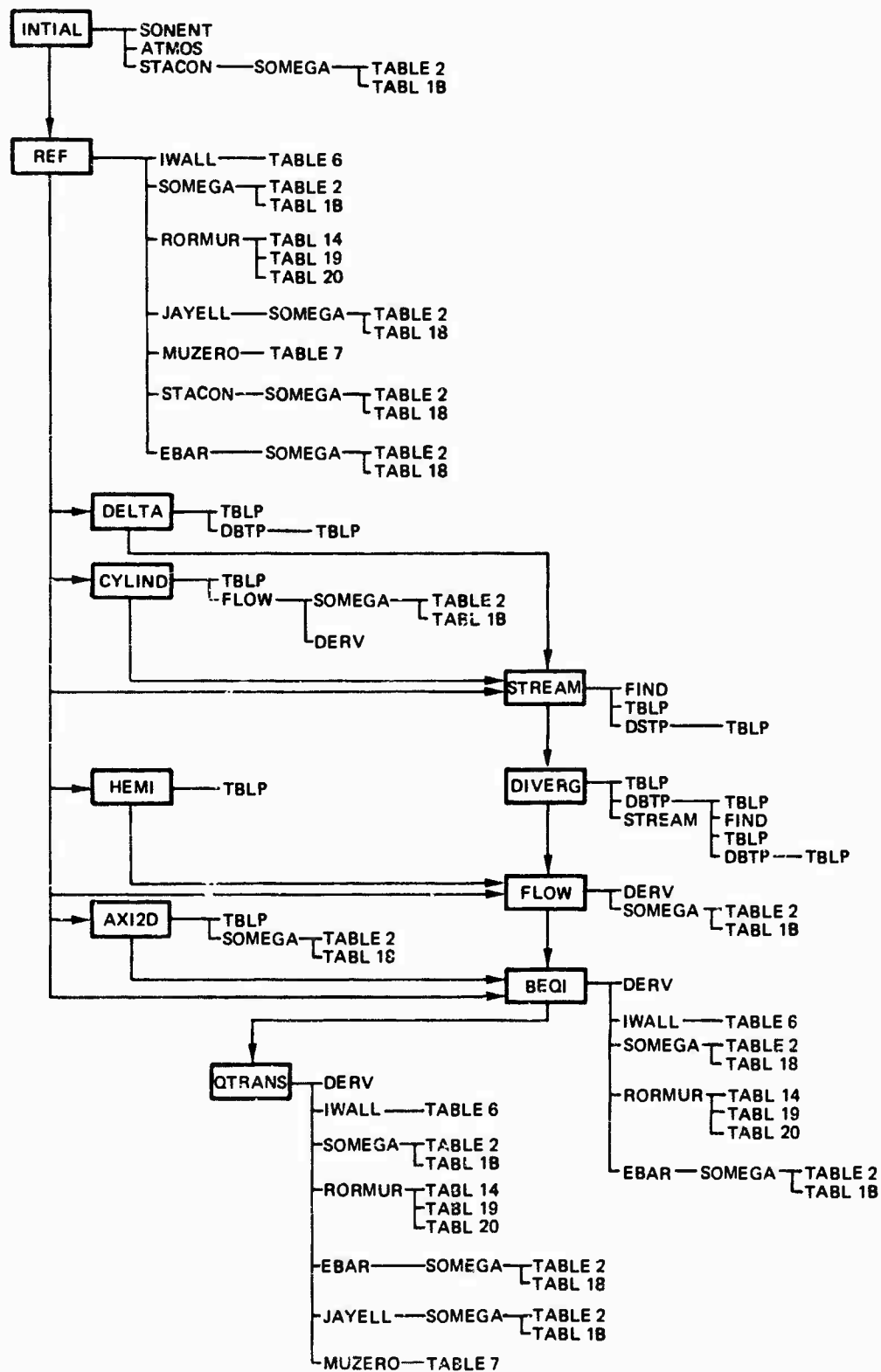
Before proceeding, a description of the program coordinate system is required. The program operates in a two-dimensional curvilinear coordinate system where the coordinate axes are designated as η and ξ . Restrictions on this coordinate system are:

1. η and ξ are orthogonal.
2. Both η and ξ must be parallel to the surface over which the boundary layer flow is being considered, and
3. ξ 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, η and ξ will lie in the same plane.



a. $\rho_r \mu_r$ Program Flow Chart



b. Subroutines

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

ALT altitude, ft

Output

SONIC a_∞ free-stream velocity of sound, ft/sec

PINF P_∞ free-stream pressure, lb/ft²

RHO ρ_∞ free-stream density, slug/ft³

TEMP T_∞ free-stream temperature, °R

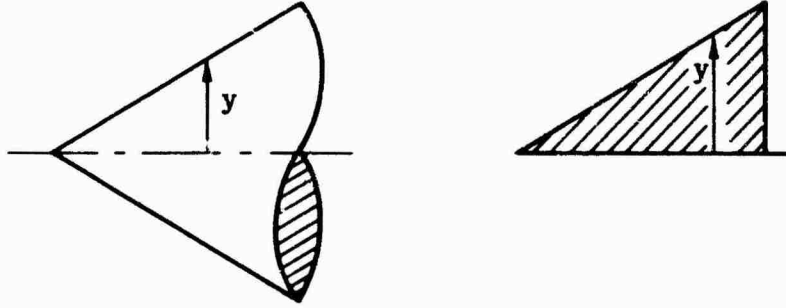
2) AXI2D (Option D-1)

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

- (a) With the geometry specified by the coordinates (ξ_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

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

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

where

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

$$\frac{C_P}{\sin^2 \delta} = 1.05 + \left[1.1025 + \frac{4}{M_{\infty}^2 \sin^2 \delta} \right]^{1/2} - \frac{1.278 \sin^2 \delta}{M_{\infty}^{0.6}}$$

for $\delta > 0$

The end points x_I 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_{∞} , 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 $x^* = x_I$ and calculate the flow conditions at each increment dx along the streamline

$$x_i^* = x_{i-1}^* + dx \quad i \geq 2$$

Linear interpolation is performed to find $P/P_{SL}(x^*)$ at each x^* using the tables of P and x calculated previously.

If $P(x_I) \leq .528 P_0$

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

If $P(x_I) > .528 P_0$ find the sonic location X^* where $P = .528 P_0$ from interpolation of the x and P tables. Then

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

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

$$a^* = (H/3)^{1/2} \quad \text{Sonic velocity}$$

The remaining flow conditions are calculated from

$$\left. \begin{aligned} \frac{\Delta}{\Delta_i}(x^*) &= 1.0 \\ \frac{r}{r_i}(x^*) &= 1.0 \end{aligned} \right\} \quad \text{Two-dimensional body}$$

$$\left. \begin{aligned} \frac{\Delta}{\Delta_i}(x^*) &= y(x^*) \\ \frac{r}{r_i}(x^*) &= y(x^*) \end{aligned} \right\} \quad \text{Axisymmetric body}$$

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

$$\left. \begin{aligned} \omega_e(x^*) &= f(P/P_{SL}(x^*), i_e(x^*)) \\ ZT_e(x^*) &= f(P/P_{SL}(x^*), i_e(x^*)) \end{aligned} \right\} \quad [\text{SOMEGA}]$$

- (d) Step (c) is repeated until the flow properties have been calculated for all points on the streamline.

Input

H	H	total enthalpy, ft^2/sec^2
XSI	ξ	coordinate in free-stream flow direction η , ft
Y	$y(\xi)$	geometry coordinate orthogonal to ξ , ft
XI	x_I	streamline coordinate at which calculations are to begin, ft
DX	dx	streamline coordinate increment, ft
POPSL	P_o/P_{SL}	stagnation pressure, atm
ACH	M_∞	free-stream Mach number
PINF	P_∞	free-stream pressure, lb/ft^2

Output

II		number of points calculated along streamline
XSTAR	x^*	streamline coordinate, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
UE	u_e	edge velocity, ft/sec
AIEE	$i_e(x^*)$	edge enthalpy, ft^2/sec^2
ZTEX	$(ZT)_e(x^*)$	edge compressibility-temperature product, $^\circ\text{R}$
OMEGAE	$\omega_e(x^*)$	edge viscosity parameter, $\text{slug}/\text{ft-sec-}^\circ\text{R}$
RORI	$(r/r_i)(x^*)$	divergence parameter due to body-shock layer geometry
DODI	$(\Delta/\Delta_i)(x^*)$	distance between streamlines, or total streamline divergence

3) BEQI

BEQI calculates the equivalent distance parameters at the initial streamline coordinate x_1 . 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]

$$\left. \begin{array}{l} ZT_S \\ \omega_S \end{array} \right\} = f(H, P/P_{SL}) \quad [\text{SOMEGA}]$$

$$\left. \begin{array}{l} ZT_w \\ \omega_w \end{array} \right\} = f(i_w, P/P_{SL}) \quad [\text{SOMEGA}]$$

(d) Calculate the reference terms

$$\omega_r, \rho_r \mu_r, ZT_r, i_r, \sigma_r \quad [\text{RORMUR}]$$

(e) Calculate

$$\frac{d(\Delta/\Delta_i)}{dx} \quad [\text{DERV}]$$

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

$$N = \frac{x^* d(\Delta/\Delta_i)}{(\Delta/\Delta_i) dx}$$

If $N \leq .05$; $.99 \leq N \leq 1.01$; $\text{EXPK} = 0$, $i_{e,SL} = i_e$

$$.05 < N < .99; \text{EXPK} = -.194 e^{-\frac{2}{3}N(N-1)}, \quad \bar{\theta}_{SL} = (N-1) \theta_e$$

$$N > 1.01; \text{EXPK} = .194 e^{-\frac{2}{3}(N-1)}, \quad \bar{\theta}_{SL} = \left(\frac{N-1}{N}\right) \theta_e$$

$$v_P = u_e \cos \bar{\theta}_{SL}; \quad i_{e, SL} = H - \frac{v_P^2}{2}$$

(g) Obtain $\bar{E}_L$ and $\bar{E}_T$ [EBAR]

(h) Calculate the equivalent distance parameter

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

where

$$G_L = (\rho_r \mu_r) u_e (r/r_i)^2 (f/f_i)^2 \bar{E}_L x_I$$

$$G_T = (\rho_r \mu_r) u_e (r/r_i)^{5/4} (f/f_i)^{5/4} \bar{E}_T$$

and

$$f/f_i = \frac{\Delta/\Delta_i}{r/r_i}$$

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

$$dx_i = \frac{dx_i}{10^i} \quad \text{and} \quad \left[\left(\frac{b_{eq}}{x} \right)_{x_I} \right]_i = \left[\left(\frac{b_{eq}}{x} \right)_{x_{II}} \right]_{i-1}$$

The final $\left[(b_{eq}/x)_{x_{II}} \right]_{i=5}$ is then used as the $(b_{eq}/x)_{x_I}$ for the general calculation (QTRANS).

Input

XSTAR	x^*	streamline coordinate at dx intervals, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
TW	$T_w(x^*)$	wall temperature, °R
UE	$u_e(x^*)$	edge velocity, ft/sec
THETAS	$\theta_e(x^*)$	streamline (edge) angle, degrees
OMEGAE	$\omega_e(x^*)$	edge viscosity parameter, slug/ft-sec-°R
DODI	$\Delta/\Delta_i(x^*)$	dimensionless distance between streamlines
RORI	$r/r_i(x^*)$	divergence parameter due to body-shock layer geometry
DX	dx	streamline coordinate increment, ft

Output

BEQXLI	$(b_{eq, L/x})_{x_I}$	laminar equivalent distance parameter at x_I
BEQXTI	$(b_{eq, T/x})_{x_I}$	turbulent equivalent distance parameter at x_I

4) CYLIND (Option D-3)

CYLIND calculates the coordinates ξ and η 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 η at $\xi = 0$ and are assumed to be constant in the ξ direction.

$$\text{Pressure: } \frac{P(\eta)}{P_o} = \left[\frac{P(\theta)}{P_{T_2}} \right] \times \left[\frac{P_{T_2}}{P_o} \right]$$

where

$$\theta = 57.3 \frac{\eta}{R_{CYL}}$$

$P/P_{T_2}(\theta)$ is found by linear interpolation on a built-in table (Figure 1).

$$P_{T_2} = P_{\infty} (1.2 M_N)^{3.5} \left(\frac{6}{7M_N^2 - 1} \right)^{2.5} \quad \begin{array}{l} \text{Stagnation} \\ \text{Line Pressure} \end{array}$$

$$M_N = M_{\infty} \cos \Lambda$$

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

Streamline angle:

$$\theta_e(\eta) = \tan^{-1} \frac{v(\eta)}{u_{\infty} \sin \Lambda}$$

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

Then

$$P/P_0(\xi, \eta) = P/P_0(\eta) \quad \text{for all } \xi$$

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

$$\theta_e(\xi, \eta) = \theta_e(\eta) \quad \text{for all } \xi$$

$$(u_e)_{x_I} = u_{\infty} \sin \Lambda$$

Input

RADIUS	R_{CYL}	cylinder radius, ft
SWEEP	Λ	sweep angle, degrees
ACH	M_{∞}	free-stream Mach number
VEL	u_{∞}	free-stream velocity, ft/sec
ETAF	η_f	final η coordinate for which streamline is to be calculated, ft
XSIF	ξ_f	final ξ coordinate for which streamline is to be calculated, ft
TW	T_w	wall temperature, °R

PINF	P_{∞}	free stream pressure, lb/ft ²
POPSL	P_0/P_{SL}	stagnation pressure, atm
<u>Output</u>		
XSI	ξ	coordinate along axis of cylinder, ft
ETA	η	coordinate tangential to cylinder, ft
M		number of ξ values
N		number of η values
ETAMAX	$\eta_{\max}(\xi)$	maximum η value, ft
THSMX	$\theta_{\max}(\xi)$	streamline angle at $\eta_{\max}$, degrees
PRESSR	$P/P_0(\xi, \eta)$	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 $R \leq 0$ a sharp delta wing case is indicated and $\delta = \alpha$ (degrees) for all ξ .

(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

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

(c) The initial velocity is calculated from

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

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

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

$$M_N = M_\infty \sin \mathcal{L}$$

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

$$\gamma = \frac{5(\bar{\Phi}^{**})^4}{5(\bar{\Phi}^{**})^4 + 1}$$

$$\theta^* = (90 - \Lambda)(1 + \bar{\Phi}^{**})\gamma$$

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

$$\text{If } \bar{\Phi}^{**} \leq 1.2 \quad \bar{N}_{CL} = f(\bar{\Phi}^{**})$$

$$\text{If } \bar{\Phi}^{**} > 1.2 \quad \bar{N}_{CL} = f(\Lambda, \bar{\Phi}^{**})$$

$$N_{CL} = \frac{1 + \bar{\Phi}^{**}}{\frac{M_N^2 + 1}{M_N^2}} \bar{N}_{CL}$$

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

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

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

$$\frac{r}{r_i} = 1.0$$

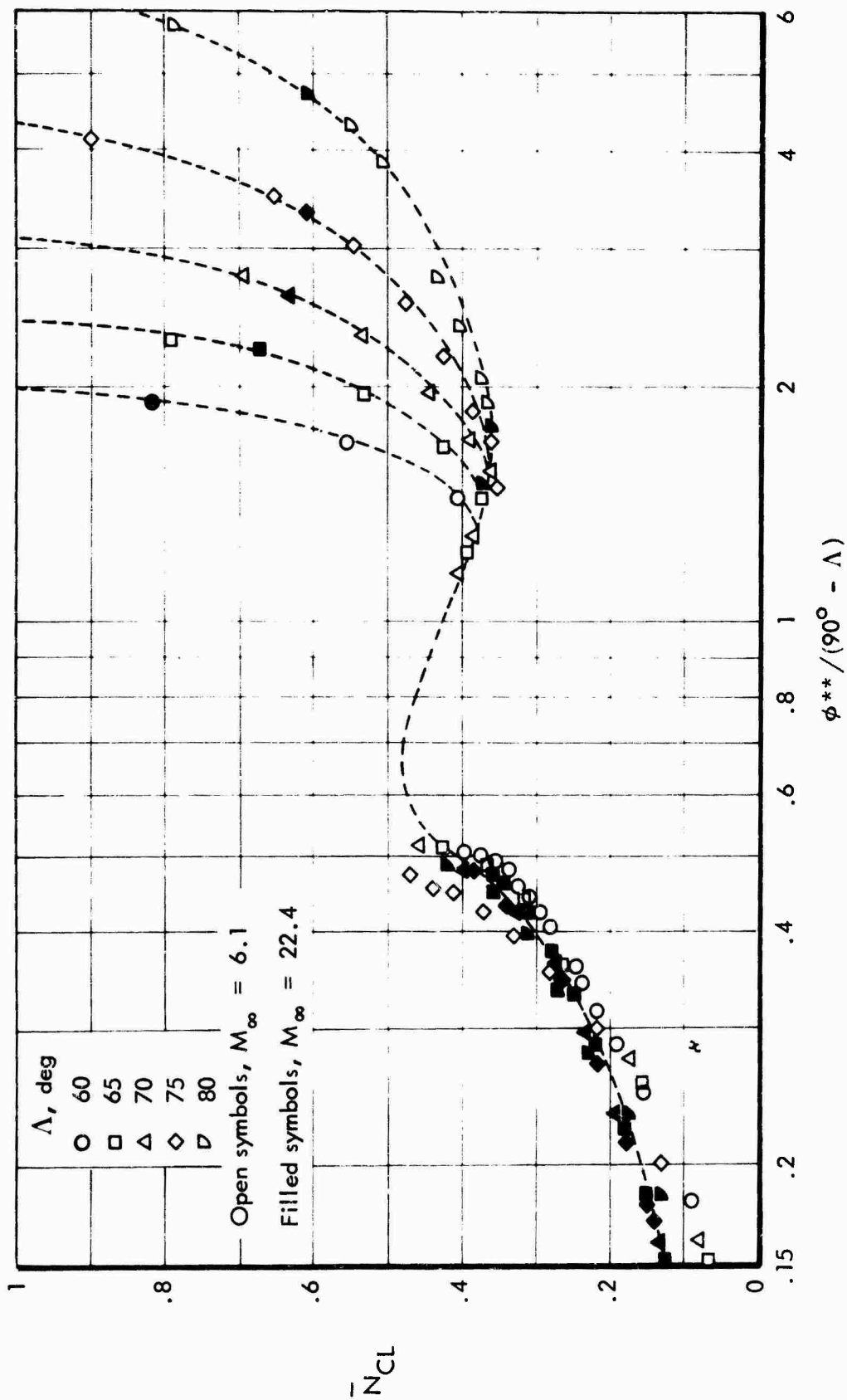


Figure 3: STREAMLINE CORRELATION FOR SHARP DELTA WINGS

Input

ACH	M_∞	free stream Mach number
PINF	P_∞	free stream pressure, lb/ft ²
POPSL	P_0/P_{SL}	stagnation pressure, atm
VEL	u_∞	free stream velocity, ft/sec
ALPHA	α	angle of attack, degrees
SWEEP	Λ	sweep angle, degrees
RADIUS	R	leading edge radius, ft
XSII	ξ_I	initial streamline ξ coordinate, ft
ETAI	η_I	initial streamline η coordinate, ft
XSIF	ξ_f	final streamline ξ coordinate, ft
ETAF	η_f	final streamline η coordinate, ft
DELXSI	$\Delta\xi$	increment used to calculate ξ coordinate, ft
DELETA	$\Delta\eta$	increment used to calculate η coordinate, ft

Output

M		number of ξ values
N		number of η values
XSI	ξ	geometry coordinate, ft
ETA	η	geometry coordinate, ft
ETAMAX	$\eta_{\max}$	leading edge coordinate, ft
TH8MX	$\theta_{\max}$	streamline angle at leading edge, degrees
THETA E	θ_e	streamline angle, degrees
PRESSR	P/P_0	pressure ratio
URATIO	u/u_∞	velocity ratio

7) DERV (V, W, X, Y, Z, DBYDX, DX, ITYPE)

DERV calculates the derivative of a tabulated function f by Lagrangian 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:

$$f'_{-2} = \frac{1}{h} (-f_{-2} + f_{-1}) ; f'_{-1} = \frac{1}{2h} (-f_{-2} + f_0)$$

$$f'_0 = \frac{1}{2h} (-f_{-1} + f_1) ; f'_1 = \frac{1}{2h} (-f_0 + f_2)$$

$$f'_2 = \frac{1}{h} (-f_1 + f_2)$$

Input

V	f_{-2}	}	functional values
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 , θ_e , r/r_i and Δ/Δ_i along a given streamline by the following method:

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

$$\eta_{F_2} = \eta_{F_1} \pm \Delta\eta_{\max}$$

(b) Calculate the following at each dx location on the streamline of interest

$\theta_e(x^*)$ linear interpolation on $\theta_e(x)$

$T_w(x^*)$ double linear interpolation on $T_w(\xi, \eta)$

$P/P_o(x^*)$ double linear interpolation on $P/P_o(\xi, \eta)$

$$\frac{r}{r_i}(x^*) = 1.0$$

$$\frac{\Delta}{\Delta_i}(x^*) = \frac{\eta_1(x^*) - \eta_2(x^*)}{\eta_{i1}(x^*) - \eta_{i2}(x^*)} \frac{\cos \theta_{e_1}(x^*)}{\cos \theta_{e_{i1}}(x^*)}$$

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_I	initial streamline distance, ft
DX	dx	streamline coordinate increment, ft
ETAF	η_f	final streamline coordinate, ft
XSIF	ξ_f	final streamline coordinate, ft
M		number of ξ input values
N		number of η input values
XSI	ξ	streamline coordinate array, ft
ETA	η	streamline coordinate array, ft
PRESSR	$P/P_o(\xi, \eta)$	pressure ratio array
TWALL	$T_w(\xi, \eta)$	wall temperature array, °R
XSIX	$\xi(x)$	streamline coordinate, ft
ETAX	$\eta(x)$	streamline coordinate, ft

X	x	streamline coordinate, ft
ETAMAX	$\eta_{\max}$	η coordinate for geometry leading edge, ft
THETAX	$\theta_e(x)$	streamline angle, degrees
<u>Output</u>		
XSTAR	x^*	streamline coordinate, ft
PRESRX	$P/P_0(x^*)$	pressure ratio along streamline
THETAS	$\theta_e(x^*)$	streamline angle, degrees
DODI	$(\Delta/\Delta_i)(x^*)$	dimensionless distance between streamlines
RORI	$(r/r_i)(x^*)$	divergence parameter
TW	T_w	wall temperature, °R
XF	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

X	}	independent variables
Y		
XA		table of X values
YA		table of Y values
ZA		table of Z values
NX		number of values in X table
NY		number of values in Y table
NZ		number of values in Z table

IX order of interpolation in X direction

IV order of interpolation in Y direction

Output

Z dependent variable

10) EBAR (AYEW, AYESL, II, SIGR, EXPK, POPSLX, OMEGE, OMEGS, GAMC, GAMO, EBARL, EBART)

EBAR calculates the crossflow pressure gradient parameters $\bar{E}_L$ and $\bar{E}_T$ and the pressure gradient correlation parameters Γ_c and Γ_o using the following method:

(a) Interpolate to find

$$ZT_{e,SL} = f(P/P_{SL}, i_{e,SL}) \quad [\text{SOMEGA}]$$

(b) Calculate the local pressure in lb/ft² and the $\rho\mu$ terms

$$P_r = P_{SL} \left(\frac{P}{P_{SL}} \right)$$

$$\rho_T \mu_T = \frac{P_r}{R} \omega_s$$

$$\rho_c \mu_e = \frac{P_r}{R} \omega_e$$

(c) Calculate the mean enthalpies

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

$$i_{m,c} = i_{avg} + .2 (H - i_{e,SL}) \sigma_r \left(\frac{\rho_T \mu_T}{\rho_e \mu_e} \right)^{1/2}$$

(d) Interpolate to find

$$ZT_{m,c} = f(P/P_{SL}, i_{m,c}) \quad [\text{SOMEGA}]$$

$$ZT_{m,o} = f(P/P_{SL}, i_{avg}) \quad [\text{SOMEGA}]$$

(e) Compute:

$$F_{\Sigma_c} = \frac{(\Sigma_c - .294) \sigma_r^{.355}}{.4018}$$

$$F_{\Sigma_o} = \frac{(\Sigma_o - .294) \sigma_r^{.355}}{.4018}$$

and

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

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

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:

$$\Gamma_c = .71764 F_{K_c} (\sqrt{1 + F_{\Sigma_c}} - 1)$$

$$\Gamma_o = .71764 F_{K_o} (\sqrt{1 + F_{\Sigma_o}} - 1)$$

(g) Compute the crossflow pressure gradient parameters:

$$\bar{E}_L = 1 + \Gamma_c$$

$$\bar{E}_T = (1 + .76519 \Gamma_o) \left(\frac{1 + \Gamma_c}{1 + \Gamma_o} \right)^4$$

Input

AYEW	i_w	wall enthalpy, ft^2/sec^2
AYESL	$i_{e,SL}$	edge enthalpy at flow line of symmetry, ft^2/sec^2
H	H	total enthalpy, ft^2/sec^2
SIGR	σ_r	Prandtl number
POPSLX	P/P_{SL}	local pressure along streamline, atm
OMEGE	ω_e	edge viscosity parameter, slug/ft-sec-°R
OMEGS	ω_s	stagnation viscosity parameter, slug/ft-sec-°R

Output

GAMC	Γ_c	pressure gradient correlation parameter
GAMO	Γ_o	pressure gradient correlation parameter
EBARL	$\bar{E}_L$	laminar crossflow pressure gradient parameter
EBART	$\bar{E}_T$	turbulent crossflow pressure gradient parameter

11) FIND (ZA, Z, NZ, NZS)

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

Input

ZA	array to be searched
Z	variable for which location is desired

NZ number of values in ZA
 NZS location of Z in ZA from previous call to FIND

Output

NZS location of Z in ZA

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:

$$(i_e)_{x_I} = H - \frac{(u_e)_{x_I}^2}{2}$$

For the special case of the swept cylinder tangential velocity calculation

$$(i_e)_{x_I} = H - \frac{(v^2)_{x_I}}{2} - \frac{(u_\infty \sin \Lambda)^2}{2}$$

Then

$$ZT_e = f(i_e, P/P_{SL}) \quad [\text{SOMEGA}]$$

$$\omega_e = f(i_e, P/P_{SL}) \quad [\text{SOMEGA}]$$

Let

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

and define

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

as the Kth iteration for u at the point x*.

Similarly define $\left\{ (ZT_e)_{x^*} \right\}_K$ as the Kth iteration for ZT_e at x^* .

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

$$\left\{ u_{x^*} \right\}_K = u_{x^*-dx} - \frac{dx R}{u_\infty^2} \left\{ (ZT_e)_{x^*} \right\}_K \frac{\left[\frac{d}{dx} \left(\frac{P}{P_o} \right) \right]_{\text{avg}}}{\left(\frac{P}{P_o} \right)_{\text{avg}}}$$

where

$$\frac{d(P/P_o)}{dx} = \frac{1}{2} \left\{ \frac{d}{dx} (P/P_o) \right\}_{x^*} + \frac{d}{dx} (P/P_o) \Big|_{x^*-dx}$$

and

$$\left(\frac{P}{P_o} \right)_{\text{avg}} = \frac{1}{2} \left\{ \frac{P}{P_o} (x^*) + \frac{P}{P_o} (x^* - dx) \right\}$$

The derivatives are calculated numerically using subroutine DERV.

For each iteration:

$$\left\{ (i_e)_{x^*} \right\}_K = H - u_\infty^2 \left\{ u_{x^*} \right\}_K$$

For the swept cylinder tangential velocity

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

Then

$$\left\{ \begin{array}{l} (ZT_e)_{x^*} \\ (\omega_e)_{x^*} \end{array} \right\}_K = \left\{ \begin{array}{l} = \\ = \end{array} \right\} f \left[\left\{ (i_e)_{x^*} \right\}_K, \frac{P}{P_{SL}} (x^*) \right] \quad [\text{SOMEGA}]$$

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

$$\left| \frac{|u_{x^*}|_K - |u_{x^*}|_{K-1}}{|u_{x^*}|_K} \right| < \epsilon$$

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

$$(u_e)_{x^*} = u_{\infty} (2u_{x^*})^{1/2}$$

$$(i_e)_{x^*} = \left| (i_e)_{x^*} \right|_I$$

$$(ZT_e)_{x^*} = \left| (ZT_e)_{x^*} \right|_I$$

$$(\omega_e)_{x^*} = \left| (\omega_e)_{x^*} \right|_I$$

A special case results when $(u_e)_{x_I} = 0$. This corresponds to a stagnation point where $x_I = 0$ and $\left| d/dx(P/P_0) \right|_{x_I} = 0$. 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_I is calculated from:

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

where the pressure derivative is obtained from the numerical derivative equation

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

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

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

$$(i_e)_{x_I+dx} = H - \frac{1}{2} (u_e^2)_{x_I+dx}$$

$$\left. \begin{array}{l} (ZT_e)_{x_I+dx} \\ (\omega_e)_{x_I+dx} \end{array} \right\} = f \left[(i_e)_{x_I+dx}, \left(\frac{P}{P_{SL}} \right)_{x_I+dx} \right] \quad [\text{SOMEGA}]$$

The calculations continue, using the previously described procedure where the starting point is now $x_I + dx$.

Input

PRESRX	$P/P_O(x^*)$	ratio of local to stagnation pressure along streamline
II		number of values in pressure array
URATIO	$(u_e/u_\infty)_{x_I}$	velocity ratio at x_I
II	II	total enthalpy, ft^2/sec^2
VEL	u_∞	free stream velocity, ft/sec
DX	dx	streamline coordinate increment, ft
POPSL	P_O/P_{SL}	stagnation pressure, atm

Output

The following outputs are a function of streamline distance:

UE	u_e	edge velocity, ft/sec
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm

AIEE	i_e	edge enthalpy, ft^2/sec^2
ZTEX	ZT_e	edge compressibility-temperature product, $^\circ\text{R}$
OMEGAE	ω_e	edge viscosity temperature ratio, $\text{slug}/\text{ft-sec-}^\circ\text{R}$
XSTAR	x^*	streamline coordinate, ft

13) HEMI (Option D-2)

HEMI calculates the streamline and the pressure and divergence parameters along the streamline for a hemisphere geometry.

- Set $x_1^* = x_I$
- Calculate x^* , $P/P_0(x^*)$, $r/r_i(x^*)$, and $\Delta/\Delta_i(x^*)$ for all dx increments along the streamline until $\theta = 90^\circ$

$$\theta = \frac{57.3 x^*}{R_{\text{HEMI}}}$$

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

$$r/r_i(x^*) = \sin\left(\frac{x^*}{R_{\text{HEMI}}}\right)$$

$$\Delta/\Delta_i(x^*) = \sin\left(\frac{x^*}{R_{\text{HEMI}}}\right)$$

- URATIO = $u_{e_{x_I}}/u_\infty$

Input

RHEM	R_{HEMI}	hemisphere radius, ft
XI	x_I	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 (L1, NTYPE)

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

(a) Wind Tunnel Class I

$$u_{\infty} = 49 M_{\infty} \sqrt{T_{\infty}}$$

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

$$q_{\infty} = 0.7 P_{\infty} M_{\infty}^2$$

$$\rho_{\infty} = P_{\infty} / (1716 T_{\infty})$$

(b) Wind Tunnel Class II

$$i_{\infty} = H - \frac{u_{\infty}^2}{2}$$

$$a_{\infty} = f(i_{\infty}) \quad \text{when } 5.93 \times 10^6 \leq i_{\infty} \leq 20.3 \times 10^6 \text{ ft}^2/\text{sec}^2 \quad [\text{SONENT}]$$

$$a_{\infty} = .6325 \sqrt{i_{\infty}} \quad \text{when } i_{\infty} < 5.93 \times 10^6 \text{ ft}^2/\text{sec}^2$$

$$M_{\infty} = \frac{u_{\infty}}{a_{\infty}}$$

$$T_{\infty} = \left(\frac{u_{\infty}}{49 M_{\infty}} \right)^2$$

$$\rho_{\infty} = P_{\infty} / (1716 T_{\infty})$$

P_0/P_∞ 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

$$\left. \begin{aligned} a_\infty &= f(\text{altitude}) \\ P_\infty &= f(\text{altitude}) \\ T_\infty &= f(\text{altitude}) \end{aligned} \right\} \quad [\text{ATMOS}]$$

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

$$M_\infty = \frac{u_\infty}{a_\infty}$$

$$q_\infty = 0.7 P_\infty M_\infty^2$$

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 M_∞ free stream Mach number

TEMP T_∞ free stream temperature, °R

PINF P_∞ free stream pressure, lb/ft²

Wind Tunnel Class II, NTYPE = WT2

VEL	u_{∞}	free stream velocity, ft/sec
H	H	total enthalpy, ft^2/sec^2
POPSL	P_o/P_{SL}	stagnation pressure, atm
PINF	P_{∞}	free stream pressure, lb/ft^2

Flight, NTYPE = FLIGHT

ALT		flight altitude, ft
VEL	u_{∞}	free stream velocity, ft/sec

Output

ACH	M_{∞}	free stream Mach number
H	H	total enthalpy, ft^2/sec^2
PINF	P_{∞}	free stream pressure, lb/ft^2
POPINF	P_o/P_{∞}	stagnation to free stream pressure ratio
POPSL	P_o/P_{SL}	stagnation pressure, atm
QINF	q_{∞}	free stream dynamic pressure, lb/ft^2
SONIC	a_{∞}	speed of sound, ft/sec
VEL	u_{∞}	free stream velocity, ft/sec

15) IWALL (TW, POPSLX, AYEWE)

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

(a) Set $T_{w_{\Delta}} = T_w/1.8$ (conversion to °K)

(b) Calculate wall enthalpy

$$i_w = .432 T_{w_K} \quad \text{when } T_{w_K} < 300^\circ\text{K}$$

$$i_w = .432 T_{w_K} + 3.82 \times 10^{-5} (T_{w_K} - 300)^2 \quad \text{when } 300 < T_{w_K} < 1800^\circ\text{K}$$

$$i_w = f(\log_{10} P/P_{SL}, T_{w_K}); \text{ [TABLE 6]} \quad \text{when } T_{w_K} > 1800^\circ\text{K}$$

(c) Convert i_w to ft^2/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	$P/P_{SL}(x^*)$	local pressure along streamline, atm

Output

AYEW	i_w	wall enthalpy, ft^2/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:

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

(c) Calculate

$$F_{\Sigma_s} = \frac{(\Sigma_s - .294)}{.4018} \left(\frac{i_{aw}}{H} \right) \sigma_r^{.355}$$

where

$$\Sigma_s = \frac{ZT_{m,s}}{ZT_s}$$

$$i_{aw} = H \left[\frac{i_e}{H} + \sigma_r^{1/2} \left(1 - \frac{i_e}{H} \right) \right]$$

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

If $\beta_s \geq 0$ set $j = 1$

See subroutine QTRANS for β_s calculation.

(e) Calculate the streamwise pressure gradient function:

$$F_{\beta_s} = \frac{(1 + 2 \text{ CPES}) |\beta_s|}{2 \text{ CPES} + \left(\frac{j+1}{2}\right) \beta_s}$$

where

$$\text{CPES} = \frac{ZT_e/ZT_s}{i_e/H}$$

(f) Calculate J_L

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

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, $^{\circ}\text{R}$
ZTS	ZT_s	stagnation compressibility-temperature product, $^{\circ}\text{R}$

SIGR	σ_r	Prandtl number
BETAS	β_s	streamwise pressure gradient parameter
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
<u>Output</u>		
XJL	J_L	laminar pressure gradient effect correlation parameter

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

MUZERO computes the reference stagnation viscosity μ_o and the diffusion effect parameter $\mathcal{L}$.

(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 μ_o and $\mathcal{L}$

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

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

<u>Input</u>		
OMEGR	ω_r	reference viscosity-temperature ratio, slug/ft-sec-°R
ZTR	ZT_r	reference compressibility-temperature product, °R
AYER	i_r	reference enthalpy, ft ² /sec ²

AYEE	i_e	edge enthalpy, ft^2/sec^2
H	H	total enthalpy, ft^2/sec^2
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
<u>Output</u>		
XMVO	μ_o	reference stagnation viscosity, slug/ft-sec
XL	$\mathcal{L}$	parameter indicating diffusion effect on heat transfer

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 $\beta_s = 0.5$
- (c) Interpolate for

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

- (d) Interpolate to find

$$\left. \begin{array}{l} ZT_s \\ \omega_s \end{array} \right\} = f(H, P/P_{SL}) \quad [\text{SOMEGA}]$$

$$\left. \begin{array}{l} ZT_w \\ \omega_w \end{array} \right\} = f(i_w, P/P_{SL}) \quad [\text{SOMEGA}]$$

- (e) Calculate

$$\omega_r, \rho_r \mu_r, ZT_r, i_r, \sigma_r \quad [\text{RORMUR}]$$

(f) Calculate

$$\frac{du_e}{dx}, \frac{d(\Delta/\Delta_i)}{dx} \quad [\text{DERV}]$$

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

$$N = \frac{x^*}{\frac{\Delta}{\Delta_i}} \frac{d(\Delta/\Delta_i)}{dx}$$

If $N \leq .05$; $.99 \leq N \leq 1.01$; $\text{EXPK} = 0$, $i_{e,SL} = i_e$

$$.05 < N < .99; \text{EXPK} = -.194e^{-\frac{2}{3}N(N-1)}, \quad \bar{\theta}_{SL} = (N-1)\theta_e$$

$$N > 1.01; \text{EXPK} = .194e^{-\frac{2}{3}(N-1)}, \quad \bar{\theta}_{SL} = \left(\frac{N-1}{N}\right)\theta_e$$

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

(i) Calculate the equivalent distance parameters defined by:

$$\begin{aligned} \frac{b_{eq}}{x} &= \frac{1}{x^* G(x^*)} \left\{ x_i G(x_i) \left(\frac{b_{eq}}{x} \right)_{x_i} + \int_{x_i}^{x^*} G(x) dx \right\} \\ &= \frac{1}{x^* G(x^*)} \left\{ x_i G(x_i) \left(\frac{b_{eq}}{x} \right)_{x_i} + \int_{x_i}^{x^*-2dx} G(x) dx + \int_{x^*-2dx}^{x^*} G(x) dx \right\} \end{aligned}$$

where the integrand G is calculated from

$$G_L = (\rho_r \mu_r) u_e \left(\frac{r}{r_i} \right)^2 \left(\frac{f}{f_i} \right)^{2\bar{E}_L}$$

$$G_T = (\rho_r \mu_r) u_e \left(\frac{r}{r_i} \right)^{5/4} \left(\frac{f}{f_i} \right)^{(5/4)\bar{E}_T}$$

and

$$\frac{f}{f_i} = \frac{\Delta/\Delta_i}{r/r_i}$$

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.

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

where

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

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 β_s from the following:

$$\beta_s(x^*) = \beta_s(x^* - 2dx) + F_{dx} \left(d\beta_s - \beta_s \right)_{x^*-2dx}$$

where

$$F_x = 71/(70 + 1/c_x)$$

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

and

$$\begin{aligned} \left(d\beta_s \right)_{x^*-2dx} &= \left[2 \left(\frac{H}{i_e} \right) \left(\frac{b_{eq}}{x} \right)_L \right]_{x^*-2dx} \quad u_e = 0 \\ &= \left[2 \left(\frac{H}{i_e} \right) \left(\frac{b_{eq}}{x} \right)_L \frac{x^*}{u_e} \frac{du_e}{dx} \right]_{x^*-2dx} \quad u_e \neq 0 \end{aligned}$$

- (k) Calculate the reference Reynolds number $R_{r,Q}$; calculate J_L [JAYELL] and μ_o , $\mathcal{L}$ [MUZERO]

then

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

$$\left(\frac{S_{eq}}{x} \right)_T = \frac{\left(\frac{b_{eq}}{x} \right)_L}{J_L^{.9/16}} ; \quad R_{r,Q} = \frac{\rho_r \mu_r^u \left(\frac{x_{eq}}{x} \right)_L x^*}{\mu_o^2 F_{x,Q}^2}$$

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

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

$$R_{L,S} = \frac{\rho_r \mu_r^u S_{eq,L}}{\mu_o^2}$$

$$R_{T,S} = \frac{\rho_r \mu_r^u S_{eq,T}}{\mu_o^2}$$

$$R_{\theta, L} = \frac{\mu_o}{\mu_e} \left[.664 (R_{L, S})^{1/2} \right] = \frac{\rho_e u_e \theta_L}{\mu_e}$$

$$R_{\theta, T} = \frac{\mu_o (.2135 R_{T, S})}{\mu_e (\log_{10} R_{T, S} - .407)^{2.64}} = \frac{\rho_e u_e \theta_T}{\mu_e}$$

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

$$\left(\frac{b_{\text{eq}}}{x} \right)_T = \left(\frac{b_{\text{eq}}}{x} \right)_T - \frac{C_{TR}}{G_T x^*}$$

If $R_{r, Q} < R_{\text{transition}}$ (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 $\mu_o, \mathcal{L}$ [JAYELL] and [MUZERO]

Then

$$\left(\frac{S_{\text{eq}}}{x} \right)_L = \frac{\left(\frac{b_{\text{eq}}}{x} \right)_L}{J_L^2}$$

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

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

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

Reference Reynolds numbers:

$$R_{r,Q} = \frac{\rho_r \mu_r u_e (x_{eq}/x)_L x^*}{\mu_o^2 F_{x,Q}^2}$$

$$R_{r,S} = \frac{\rho_r \mu_r u_e (S_{eq}/x)_L x^*}{\mu_o^2 F_{x,S}^2}$$

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

$$\frac{\tau_L}{u_e} = .332 \frac{J_L^{3/2} \mu_o F_{x,S} (R_{r,S})^{1/2}}{(S_{eq}/x)_L x^*}$$

Skin friction coefficients:

$$C_{f,e,L} = 2 \frac{\tau_L}{u_e} \frac{1}{\rho_e u_e}$$

$$C_{f,e,T} = 2 \frac{\tau_T}{u_e} \frac{1}{\rho_e u_e}$$

where

$$\rho_e = 1.232 \frac{P/P_{SL}}{Z T_e}$$

Heat transfer coefficients based on enthalpy:

$$H_L = \frac{.332 g \mathcal{L} J_L^{3/10}}{\sigma_r^{.645}} \frac{\mu_o F_{x,Q} (R_{r,Q})^{1/2}}{(x_{eq}/x)_L^{x^*}}$$

$$H_T = \frac{g \mathcal{L} J_L^{3/10} \mu_o F_{x,Q}}{\sigma_r^{.645} (x_{eq}/x)_L^{x^*}} \left\{ \frac{.185 R_{r,Q}}{\left[\log_{10} (R_{r,Q} + 3000) \right]^{2.584}} \right\}$$

$$H_L = \frac{H_L}{H_o}$$

$$H_T = \frac{H_T}{H_{ref, T}}$$

Nonisothermal wall parameters:

$$\Phi_{local} = \left(i_{w_{local}} - i_{w_o} \right) + \sum_{i=1}^n \left(i_{w_i} - i_{w_{i-1}} \right) F_{\bar{S}_i}$$

where

$$F_{\bar{S}_i} = \left[1 - (\bar{S}_L^*)_i^{3/4} \right]^{-1/3} \quad \text{laminar}$$

$$F_{\bar{S}_i} = \left[1 - (\bar{S}_T^*)_i^{9/10} \right]^{-1/9} \quad \text{turbulent}$$

and

$$\bar{S}_{L_i}^* = \frac{S_{L_i} - \frac{1}{3} (S_{L_i} - S_{L_{i-1}})}{(S_L)_{local}}; \quad S_{L_i} = \int_0^{x_i} G_L dx$$

$$\bar{S}_{T_i}^* = \frac{S_{T_i} - \frac{1}{3} (S_{T_i} - S_{T_{i-1}})}{(S_T)_{local}}; \quad S_{T_i} = \int_0^{x_i} G_T dx$$

Adiabatic wall enthalpy:

$$i_{aw,L} = i_e - \sigma_r^{1/2} (H_o - i_e)$$

$$i_{aw,T} = i_e + \sigma_r^{1/2} (H_o - i_e)$$

Nonisothermal heat transfer rates:

$$\dot{q}_L = H_L (i_{aw,L} - i_w + \Phi_{L,local})$$

$$\dot{q}_T = H_T (i_{aw,T} - i_w + \Phi_{T,local})$$

Isothermal heat transfer rates:

$$\dot{q}_{L,iso} = H_L (i_{aw,L} - i_w)$$

$$\dot{q}_{T,iso} = H_T (i_{aw,T} - i_w)$$

(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

XSTAR	x^*	streamline coordinate at dx intervals, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
TW	$T_w(x^*)$	wall temperature, °R
AIEE	$i_e(x^*)$	edge enthalpy, ft^2/sec^2

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	σ_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_{w0}	reference wall enthalpy, ft ² /sec ² (needed <u>only</u> if $T_w \neq \text{constant}$)
BEQXLI	$\left(\frac{b_{eq}}{x} \right)_{x_I}$	initial value of b_{eq}/x (laminar and turbulent)
BEQXTI		

Output

XSTAR	x^*	streamline coordinate, ft
POPSLX	$P/P_{SL}(x^*)$	local pressure along streamline, atm
UE	$u_e(x^*)$	edge velocity, ft/sec
Δ/Δ_i	$\Delta/\Delta_i(x^*)$	distance between streamlines

RORI	$r/r_i(x^*)$	divergence parameter due to body-shock layer geometry
TW	$T_w(x^*)$	wall temperature, °R
ZTEX	$ZT_e(x^*)$	edge compressibility-temperature product, °R
OMEGAE	$\omega_e(x^*)$	edge viscosity temperature ratio, slug/ft-sec-°R
AIEE	$i_e(x^*)$	edge enthalpy, ft ² /sec ²
AYEW	$i_w(x^*)$	wall enthalpy, ft ² /sec ²
BEQXL	$b_{eq, L}/x(x^*)$	ratio of laminar equivalent distance parameter to distance x along streamline
BEQXT	$b_{eq, T}/x(x^*)$	ratio of turbulent equivalent distance parameter to distance x along streamline
AYEAWL	$i_{aw, L}(x^*)$	laminar adiabatic wall enthalpy, ft ² /sec ²
AYEAWT	$i_{aw, T}(x^*)$	turbulent adiabatic wall enthalpy, ft ² /sec ²
HL	$H_L(x^*)/H_0$	laminar heat transfer coefficient ratio, lb _m /ft ² /sec
HT	$H_T(x^*)/H_{ref, T}$	turbulent heat transfer coefficient ratio, lb _m /ft ² /sec
QLISO	$\dot{q}_{iso, L}(x^*)$	laminar isothermal heat transfer rate, Btu/ft ² -sec
QTISO	$\dot{q}_{iso, T}(x^*)$	turbulent isothermal heat transfer rate, Btu/ft ² -sec
QDOTL	$\dot{q}_L(x^*)$	laminar heat transfer rate, Btu/ft ² -sec
QDOTT	$\dot{q}_T(x^*)$	turbulent heat transfer rate, Btu/ft ² -sec
CFEL	$C_{f, e, L}(x^*)$	laminar skin friction coefficient
CFET	$C_{f, e, T}(x^*)$	turbulent skin friction coefficient
RRQ	$R_{r, Q}$	heat transfer reference Reynolds number
RRS	$R_{r, S}$	skin friction reference Reynolds number

19) REF

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:

$$i_{w_0} \quad [\text{IWALL}]$$

$$ZT_w, ZT_e, ZT_s, \omega_w, \omega_e, \omega_s \quad [\text{SOMEGA}]$$

$$\omega_r, \rho_r \mu_r, ZT_r, i_r, \sigma_r \quad [\text{RORMUR}]$$

$$J_L \quad [\text{JAYELL}]$$

$$\mu_0, \mathcal{L} \quad [\text{MUZERO}]$$

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

$$H_o = \frac{.664 g \mathcal{L}}{\sigma_r^{.645}} \left[\frac{\rho_r \mu_r u_\infty J_L u'_{MN}}{R_{\text{HEMI, ref}}} \right]^{1/2}$$

$$\dot{q}_o = H_o (H - i_{w_0})$$

The reference conditions are calculated from the following:

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

(e) Obtain the following from the indicated subroutine:

$$P_{ST}/P_\infty, i_{ST}, u'_{MN} \quad [\text{STACON}]$$

$$i_{w, \text{ref}} \quad [\text{IWALL}]$$

$$ZT_w, ZT_e, ZT_s, \omega_w, \omega_e, \omega_s \quad [\text{SOMEGA}]$$

$$\omega_r, \rho_r \mu_r, ZT_r, i_r, \sigma_r \quad \text{[RORMUR]}$$

$$\Gamma_c, \Gamma_o \quad \text{[EBAR]}$$

$$\mu_o, \mathcal{L} \quad \text{[MUZERO]}$$

(f) Calculate the following:

$$x_{eq, L} = \frac{.866 R_{CYL, ref}}{(1 + \Gamma_c) u'_{MN}}$$

$$F_{x, Q} = \left[1.6 \left(\frac{1 + \Gamma_o}{1 + .7625 \Gamma_o} \right) \right]^{1/3} \frac{1 + \Gamma_o}{1 + \Gamma_c}$$

$$R_{r, Q, ref} = \frac{.866 \rho_r \mu_r u_{\infty} x_{eq, L}}{\mu_o^2 F_{x, Q}^2}$$

$$H_{ref, L} = \frac{.332 g \mathcal{L}}{\sigma_r^{.645}} \frac{\mu_o F_{x, Q}}{x_{eq, L}} (R_{r, Q, ref})^{1/2}$$

$$H_{ref, T} = \frac{.185 g \mathcal{L}}{\sigma_r^{.645}} \frac{\mu_o F_{x, Q}}{x_{eq, L}} \frac{R_{r, Q, ref}}{\left[\log_{10} (R_{r, Q, ref} + 3000) \right]^{2.584}}$$

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

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

$$\dot{q}_{ref, L} = H_{ref, L} (i_{aw, L} - i_{w, ref})$$

$$\dot{q}_{ref, T} = H_{ref, T} (i_{aw, T} - i_{w, ref})$$

$$\tau_{\text{ref, L}} = \frac{.866 u_{\infty} \sigma_r^{.645}}{g \mathcal{L}} H_{\text{ref, T}}$$

$$\tau_{\text{ref, T}} = \frac{.866 u_{\infty} \sigma_r^{.645}}{g \mathcal{L}} H_{\text{ref, T}}$$

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

Input

PINF	P_{∞}	free stream pressure, lb/ft ²
ACH	M_{∞}	free stream Mach number
VEL	u_{∞}	free stream velocity, ft/sec
VELP	u'_{MN}	modified Newtonian velocity gradient, 1/sec
H	H	total enthalpy, ft ² /sec ²
POPSL	P_0/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_0}	stagnation wall enthalpy, ft ² /sec ²
HO	H_0	stagnation heat transfer coefficient, lb _m /ft ² -sec
QDOTO	$\dot{q}_0$	stagnation heat transfer rate, Btu/ft ² -sec
HREFL	$H_{\text{ref, L}}$	laminar reference heat transfer coefficient, lb _m /ft ² -sec
HREFT	$H_{\text{ref, T}}$	turbulent reference heat transfer coefficient, lb _m /ft ² -sec

AYAWRL	$i_{aw, ref, L}$	laminar reference adiabatic wall enthalpy, ft^2/sec^2
AYAWRT	$i_{aw, ref, T}$	turbulent reference adiabatic wall enthalpy, ft^2/sec^2
TAURL	$\tau_{ref, L}$	laminar reference shear stress, lb/ft^2
TAURT	$\tau_{ref, T}$	turbulent reference shear stress, lb/ft^2
QREFL	$\dot{q}_{ref, L}$	laminar reference heat transfer rate, $lb_m/ft^2\text{-sec}$
QREFT	$\dot{q}_{ref, T}$	turbulent reference heat transfer rate, $lb_m/ft^2\text{-sec}$
CFREFL	$C_{f, e, ref, L}$	laminar reference skin friction coefficient
CFREFT	$C_{f, e, ref, T}$	turbulent reference skin friction coefficient
AYEREF	$i_{w, ref}$	reference wall enthalpy, ft^2/sec^2

20) RORMUR

RORMUR calculates the following reference properties:

(a) Viscosity parameter

$$K^* = \frac{\omega_s}{\omega_w} \left[\frac{1,005}{.005 + \left(\frac{\omega_s}{\omega_e} \right)^7} \right]^{1/14}$$

$$K = (K^*) \left[\frac{13}{16} + \frac{3}{16} e^{-\left(K^* \frac{\omega_e}{\omega_w} \right)} \right]$$

$$F_1 = K \frac{\omega_e}{\omega_w}$$

$$\omega_r = \omega_w F_1^{7/8} \left(\frac{1.2}{.2 + F_1^5} \right)^{1/10}$$

(b) Density-viscosity product

$$P_r = \left(\frac{P}{P_{SL}} \right) P_{SL}$$

$$\rho_r \mu_r = P_r \frac{\omega_r}{R}$$

(c) Compressibility-temperature product ZT_r

Change units of ω_r to slug/ft-sec-°K

$$\omega_{r, \text{°K}} = 1.8 \omega_{r, \text{°R}}$$

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

$$ZT_{r, \text{°K}} = \frac{23213.13 \times 10^{-20}}{\omega_{r, \text{°K}}^2} \left[1 + \left(1 - .478656 \times 10^{18} \omega_{r, \text{°K}}^2 \right)^{1/2} \right]^2$$

Otherwise $ZT_{r, \text{°K}}$ is obtained from interpolation

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

$ZT_{r, \text{°K}}$ is converted to °R by $ZT_{r, \text{°R}} = 1.8 ZT_{r, \text{°K}}$

(d) Enthalpy i_r

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

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

(e) Prandtl number σ_r

$$\sigma_r = f(ZT_r) \quad [\text{TABL 14}]$$

Input

POPSLX	P/P_{SL}	local pressure along streamline, atm
OMEGS	ω_s	stagnation viscosity-temperature ratio, slug-ft-sec-°R
OMEGE	ω_e	edge viscosity-temperature ratio, slug/ft-sec-°R
OMEGW	ω_w	wall viscosity-temperature ratio, slug/ft-sec-°R

Output

OMEGR	ω_r	reference viscosity-temperature ratio, slug/ft-sec-°R
ROMUR	$\rho_r \mu_r$	reference density-viscosity product, slug ² /ft ⁴ -sec
ZTR	ZT_r	reference compressibility-temperature product, °R
AYER	i_r	reference enthalpy, ft ² /sec ²
SIGR	σ_r	reference Prandtl number

21) SIRCH (Y, X, YA, XA, N, IO)

SIRCH uses TAB to perform single interpolation on a table of $y = f(x)$.

Input

X	x	independent variable
YA		dependent array
XA		independent array
N		number of values in XA and YA arrays
IO		order of interpolation

Output

Y	y	dependent variable
---	---	--------------------

22) SOMEGA (ENTH, POPSIX, ZT, OMEGA)

SOMEGA calculates the viscosity-temperature ratio ω and the compressibility-temperature product ZT as functions of enthalpy and pressure.

(a) Convert enthalpy to Btu/lb_m

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

(b) Calculate ZT

For $i \leq 180$

$$ZT = i/.432$$

For $i > 180$

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

Calculate ω

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

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

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

(c) Convert ZT and ω to °R

$$ZT_{\circ R} = 1.8 ZT_{\circ K}$$

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

Input

ENTH	i	enthalpy
POPSIX	P/P _{SL}	local pressure along streamline, atm

Output

ZT	ZT	compressibility-temperature product, °R
OMEGA	ω	viscosity-temperature ratio, slug/ft-sec-°R

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:

$$a_{\infty} = (\gamma R T_{\infty})^{1/2}$$

where

$$T_{\infty} = f(i_{\infty}/RT_0)$$

$$\gamma_{\infty} = f(T_{\infty})$$

Input

ENTH	i_{∞}	free stream enthalpy, ft ² /sec ²
------	--------------	---

Output

SONIC	a_{∞}	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_{∞}

$$\left(\frac{P_{ST}}{P_0} \right) = \frac{P_{\infty}}{P_{SL}} \left(1.2 M_N^2 \right)^{3.5} \left(\frac{6}{7M_N^2 - 1} \right)^{2.5} \quad M_N \geq 1$$

$$(P_{ST})_0 = \frac{P_\infty}{P_{SL}} \left(1 + \frac{M_N^2}{5} \right)^{3.5} \quad M_N < 1$$

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

Interpolate for

$$(ZT_{ST})_0 = f(P_{ST_0}, i_{ST}) \quad [\text{SOMEGA}]$$

$$(P_{ST})_1 = \frac{P_\infty}{P_{SL}} + \left(P_{ST_0} - \frac{P_\infty}{P_{SL}} \right) \left[\frac{2 \left(\frac{P_2}{P_1} \right) - \frac{6006 (ZT_{ST})_0}{i_{ST}}}{2(\rho_2/\rho_1) - 1} \right]$$

Interpolate for

$$(ZT_{ST})_1 = f(P_{ST_1}, i_{ST}) \quad [\text{SOMEGA}]$$

$$(P_{ST})_2 = \frac{P_\infty}{P_{SL}} + \left(P_{ST_0} - \frac{P_\infty}{P_{SL}} \right) \left[\frac{2(\rho_2/\rho_1) - \frac{6006 ZT_{ST_1}}{i_{ST}}}{2(\rho_2/\rho_1) - 1} \right]$$

$$\frac{P_0}{P_\infty} = \frac{(P_{ST})_0}{(P_\infty/P_{SL})}$$

(c) Modified Newtonian pressure gradient

$$u'_{MN_0} = \left\{ \frac{2}{7} \left(1 + \frac{5}{M_N^2} \right) \left(1 - \frac{P_\infty}{P_0} \right) \left[\frac{6006 (ZT_{ST})_1}{i_{ST}} \right] \right\}^{1/2}$$

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_∞ (Hemisphere).

$$u'_{MN} = (u'_{MN}) \cdot (\text{CORRECTION FACTOR})^5$$

For the correction factor see Figure 4.

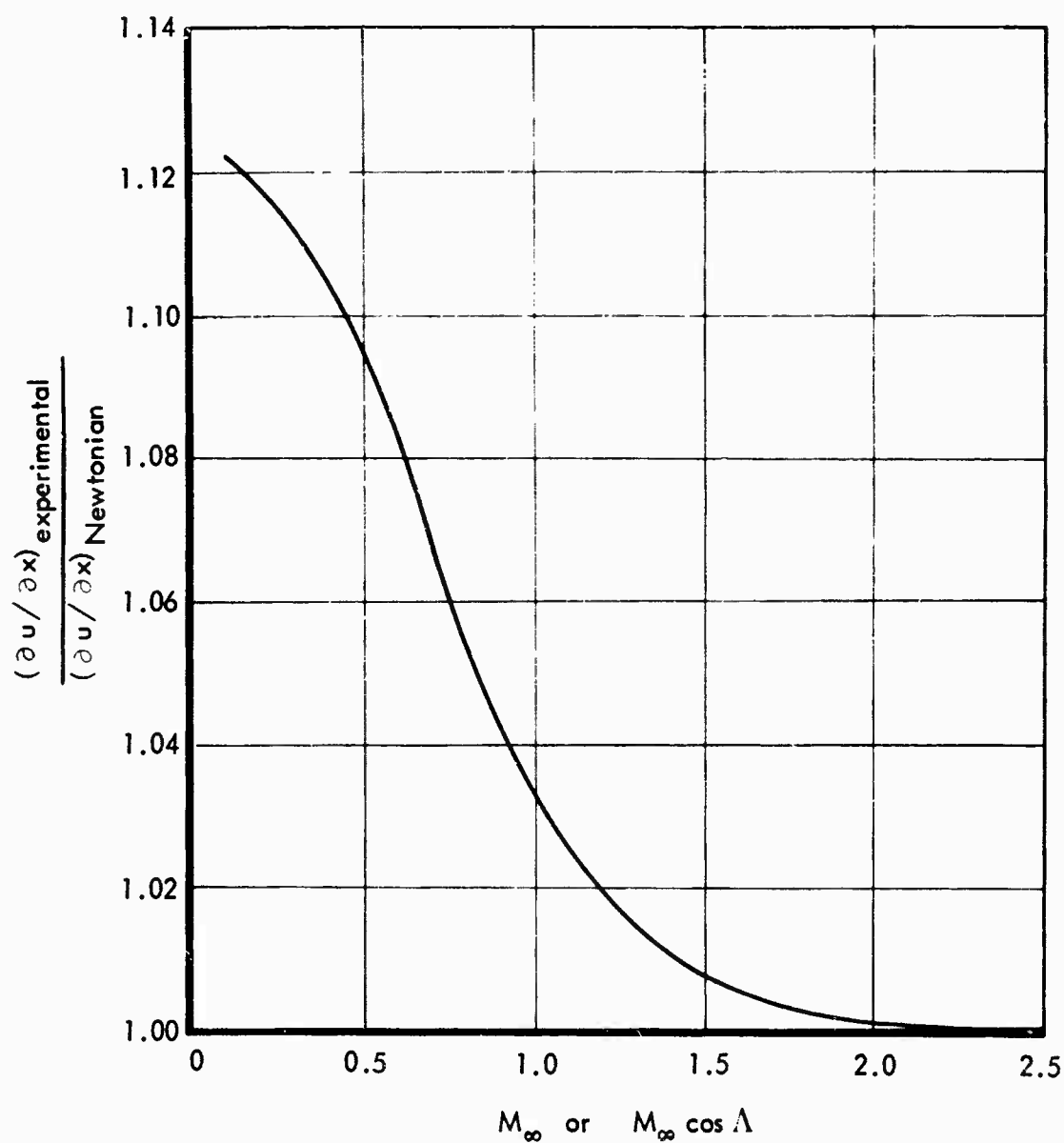


Figure 4: CORRECTION FACTOR FOR THE VELOCITY GRADIENT AT A
HEMISPHERE STAGNATION POINT OR CYLINDER STAGNATION
LINE

Input

ACH	M_∞	free stream Mach number
ACHN	M_N	normal Mach number
H	H	total enthalpy
PINF	P_∞	free stream pressure, lb/ft ²
VEL	u_∞	free stream velocity, ft/sec

Output

PSTPNF	P_0/P_∞	stagnation to free stream pressure ratio
AYEST	i_{ST}	stagnation enthalpy, ft ² /sec ²
VELP	u'_{MN}	modified Newtonian velocity gradient, 1/sec

25) STREAM (Program C)

STREAM calculates the streamline coordinate x as a function of the geometry coordinates ξ and η . A complete description of the geometry and streamline pattern are provided by the coordinates ξ and η , $\theta_e(\xi, \eta)$, $\eta_{\max}(\xi)$ and $\eta_{\max}(\xi)$.

$\theta_e(\xi, \eta)$ must be provided for the grid of ξ and η that lies within the geometry boundaries. With $\theta_{\max}(\xi)$ known along the upper η geometry boundary the complete streamline pattern on the body is defined. $\eta_{\max}(\xi)$ specifies the upper η boundary thus defining the geometry.

The coordinates (ξ_f, η_f) determine a point on a particular streamline. The streamline calculation begins at (ξ_f, η_f) and proceeds upstream until the geometry boundary is reached. The calculation of the streamline is done by the following procedure:

Let $(\xi_x)_1 = \xi_f$, $(\eta_x)_1 = \eta_f$, and $(x)_1 = 0$. Linear interpolation is performed on $\theta_e(\xi, \eta)$ and $\theta_{\max}(\xi)$ if necessary, to determine $(\theta_x)_1$. Then

$$(d\eta_x)_1 = dx \sin (\theta_x)_1$$

$$(d\xi_x)_1 = dx \cos (\theta_x)_1$$

Then the coordinates of the new point are:

$$(\xi_x)_2 = (\xi_x)_1 - (d\xi_x)_1$$

$$(\eta_x)_2 = (\eta_x)_1 - (d\eta_x)_1$$

$$(x_2)_2 = dx$$

Continuing this procedure, the general equations are.

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

$$(d\xi_x)_{K-1} = dx \sin (\theta_x)_{K-1}$$

$$(d\eta_x)_{K-1} = dx \cos (\theta_x)_{K-1}$$

and

$$(\xi_x)_K = \xi_F - \sum_{i=1}^{K-1} (d\xi_x)_i$$

$$(\eta_x)_K = \eta_F - \sum_{i=1}^{K-1} (d\eta_x)_i$$

$$(x)_K = \sum_{i=1}^{K-1} d(x_x)_i$$

The streamline calculation is completed by rearranging the ξ_x , η_x and x arrays such that ξ_x , η_x and x are measured from the upstream end of the streamline.

The subroutine STREAM assumes zero streamline divergence due to body geometry

$$\frac{r}{r_i} = 1.0$$

Input

M		number of ξ coordinates
N		number of η coordinates
XSIF	ξ_f	final streamline ξ coordinate, ft
ETAF	η_f	final streamline of η coordinate, ft
DX	dx	incremental streamline distance, ft
XSI	ξ	geometry coordinate, ft
ETA	η	geometry coordinate, ft
ETAMAX	$\eta_{\max}$	body edge coordinate, ft
TH8MX	$\theta_{\max}$	streamline angle at body edge, degrees
THETA E	θ_e	streamline angle, degrees

Output

II		number of points calculated along streamline
XSIX	ξ_x	streamline coordinate, ft
ETAX	η_x	streamline coordinate, ft
X	x	streamline coordinate, ft
THETAX	θ_x	streamline angle, degrees
XF	x_f	final x-coordinate on streamline, ft

26) TABLE2 (AYES, PEDGEL, TS, CHECK)

TABLE2 obtains by table interpolation the compressibility-temperature product as a function of pressure and enthalpy.

Input

AYES	i	enthalpy, Btu/lb _m
PPLOG	$\log_{10} P/P_{SL}$	log of ratio of pressure to sea level pressure, pressure in atm

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, AYEWE)

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

Input

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

Output

AYEW	i_w	wall enthalpy, Btu/lb _m
------	-------	------------------------------------

28) TABLE7 (AYEE, PPLOG, AYEAYE)

TABLE7 interpolates on a table of pressure and edge enthalpy to find the dissociation on reaction enthalpy ratio i_D/i_e .

Input

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

Output

AYEAYE	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

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

Output

OMEGA	ω	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	ω	viscosity-temperature ratio, slug/ft-sec-°K
PPLOG	$\log_{10} P/P_{SL}$	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 i/RT_0 as a function of pressure and compressibility-temperature product.

Input

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

Output

AYERTO	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)

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

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

2	1-10	F10	Flight altitude, ft.
2	11-20	F10	Flight velocity, ft/sec.

WT1

2	1-10	F10	Free-stream Mach number.
2	11-20	F10	Free-stream static temperature, R.
2	21-30	F10	Free-stream static pressure, lb/ft ² .

WT2

2	1-10	F10	Free-stream velocity, ft/sec.
2	11-20	F10	Total enthalpy, ft ² /sec ² .

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
2	21-30	F10	Ratio of stagnation pressure to sea level pressure.
2	41-50	F10	Free-stream dynamic pressure, lb/ft^2 .
3	1-10	F10	Radius of 60° swept infinite cylinder for the turbulent reference heating rate, ft.
3	11-20	F10	Radius of hemisphere for the laminar reference heating rate, ft.
3	21-30	F10	Reference wall temperature, $^\circ\text{R}$.
4	1-14	A6	Contains the word AXISYMMETRIC or TWODIMENSIONAL.
5	1-5	I5	M is the number of values in the geometry tables $y(\xi)$ and ξ . $3 \leq M \leq 50$.
6	----	8F10	This table contains M values of ξ , ft.
7	----	8F10	This table contains M values of $y(\xi)$ corresponding to ξ in the preceding table, ft.

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.

8	1-10	F10	Initial x location, point at which calculation begins, ft.
8	11-20	F10	Increment in x, ft. $dx > (x_f - x_o)/750$.
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. $2 \leq \text{ITC} \leq 50$. If $\text{ITC} \leq 3$ then the wall temperature must be nonisothermal.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
10	----	8F10	This table contains ITC values of surface distance, ft.
11	----	8F10	This table contains ITC values of wall or surface temperature corresponding to the preceding distance table, °R. If $T_w = \text{constant}$ then enter two values.
12	1-5	I5	K_{PO} is the number of printout locations desired. $K_{PO} \leq 500$.
12	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
13	----	8F10	This table contains K_{PO} locations at which printout is desired.

1	TYPE 6		TITLE	
2	MACH NO.		STATIC TEMP.	
3	VELOCITY		TOTAL ENTHALPY	
4	ALTITUDE		VELOCITY	
5	R _{CYL}		R _{HEMI}	
6	MACH NO.		STATIC TEMP.	
7	VELOCITY		TOTAL ENTHALPY	
8	ALTITUDE		VELOCITY	
9	R _{CYL}		R _{HEMI}	
10	MACH NO.		STATIC TEMP.	
11	VELOCITY		TOTAL ENTHALPY	
12	ALTITUDE		VELOCITY	
13	R _{CYL}		R _{HEMI}	
14	MACH NO.		STATIC TEMP.	
15	VELOCITY		TOTAL ENTHALPY	
16	ALTITUDE		VELOCITY	
17	R _{CYL}		R _{HEMI}	
18	MACH NO.		STATIC TEMP.	
19	VELOCITY		TOTAL ENTHALPY	
20	ALTITUDE		VELOCITY	
21	R _{CYL}		R _{HEMI}	
22	MACH NO.		STATIC TEMP.	
23	VELOCITY		TOTAL ENTHALPY	
24	ALTITUDE		VELOCITY	
25	R _{CYL}		R _{HEMI}	
26	MACH NO.		STATIC TEMP.	
27	VELOCITY		TOTAL ENTHALPY	
28	ALTITUDE		VELOCITY	
29	R _{CYL}		R _{HEMI}	
30	MACH NO.		STATIC TEMP.	
31	VELOCITY		TOTAL ENTHALPY	
32	ALTITUDE		VELOCITY	
33	R _{CYL}		R _{HEMI}	
34	MACH NO.		STATIC TEMP.	
35	VELOCITY		TOTAL ENTHALPY	
36	ALTITUDE		VELOCITY	
37	R _{CYL}		R _{HEMI}	
38	MACH NO.		STATIC TEMP.	
39	VELOCITY		TOTAL ENTHALPY	
40	ALTITUDE		VELOCITY	
41	R _{CYL}		R _{HEMI}	
42	MACH NO.		STATIC TEMP.	
43	VELOCITY		TOTAL ENTHALPY	
44	ALTITUDE		VELOCITY	
45	R _{CYL}		R _{HEMI}	
46	MACH NO.		STATIC TEMP.	
47	VELOCITY		TOTAL ENTHALPY	
48	ALTITUDE		VELOCITY	
49	R _{CYL}		R _{HEMI}	
50	MACH NO.		STATIC TEMP.	
51	VELOCITY		TOTAL ENTHALPY	
52	ALTITUDE		VELOCITY	
53	R _{CYL}		R _{HEMI}	
54	MACH NO.		STATIC TEMP.	
55	VELOCITY		TOTAL ENTHALPY	
56	ALTITUDE		VELOCITY	
57	R _{CYL}		R _{HEMI}	
58	MACH NO.		STATIC TEMP.	
59	VELOCITY		TOTAL ENTHALPY	
60	ALTITUDE		VELOCITY	
61	R _{CYL}		R _{HEMI}	
62	MACH NO.		STATIC TEMP.	
63	VELOCITY		TOTAL ENTHALPY	
64	ALTITUDE		VELOCITY	
65	R _{CYL}		R _{HEMI}	
66	MACH NO.		STATIC TEMP.	
67	VELOCITY		TOTAL ENTHALPY	
68	ALTITUDE		VELOCITY	
69	R _{CYL}		R _{HEMI}	
70	MACH NO.		STATIC TEMP.	
71	VELOCITY		TOTAL ENTHALPY	
72	ALTITUDE		VELOCITY	
73	R _{CYL}		R _{HEMI}	
74	MACH NO.		STATIC TEMP.	
75	VELOCITY		TOTAL ENTHALPY	
76	ALTITUDE		VELOCITY	
77	R _{CYL}		R _{HEMI}	
78	MACH NO.		STATIC TEMP.	
79	VELOCITY		TOTAL ENTHALPY	
80	ALTITUDE		VELOCITY	
81	R _{CYL}		R _{HEMI}	
82	MACH NO.		STATIC TEMP.	
83	VELOCITY		TOTAL ENTHALPY	
84	ALTITUDE		VELOCITY	
85	R _{CYL}		R _{HEMI}	
86	MACH NO.		STATIC TEMP.	
87	VELOCITY		TOTAL ENTHALPY	
88	ALTITUDE		VELOCITY	
89	R _{CYL}		R _{HEMI}	
90	MACH NO.		STATIC TEMP.	
91	VELOCITY		TOTAL ENTHALPY	
92	ALTITUDE		VELOCITY	
93	R _{CYL}		R _{HEMI}	
94	MACH NO.		STATIC TEMP.	
95	VELOCITY		TOTAL ENTHALPY	
96	ALTITUDE		VELOCITY	
97	R _{CYL}		R _{HEMI}	
98	MACH NO.		STATIC TEMP.	
99	VELOCITY		TOTAL ENTHALPY	
100	ALTITUDE		VELOCITY	
101				

11	21	31	41	51	61	71	80
----	----	----	----	----	----	----	----

8

x_I

dx

u_{e,x_I}

9

5

ITC

x TABLE (ITC VALUES)

10									

T_w TABLE (ITC VALUES)

11									

12

K_{PO}

5_I

$R_{T, transition}$

x TABLE (K_{PO} VALUES)

13									

2) HEMISPHERE CALCULATION (Option D-2)

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	Same as Option D-1		
2			
3			
4	1-10	A6	Contains title of program - HEMISPHERE.
5	1-10	F10	Hemisphere radius, ft.
6	1-5	I5	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.
7	----	8F10	This is a table of ITC values of x where T_w are input, ft.
8	----	8F10	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 .
9	1-5	I5	K_{PO} is the number of printout locations desired. $K_{PO} < 500$.
9	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
10	----	8F10	This table contains K_{PO} surface distance locations around the hemisphere at which printout is desired.

3) SWEPT INFINITE CYLINDER (Option D-3)

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	Same as Option D-1		
2			
3			
4	1-8	A6	This card contains the word CYLINDER to allow the program to select the D-3 options.
5	1-10	F10	Cylinder radius. ft.
5	11-20	F10	Cylinder sweep angle. measured from a line normal to the flow. $0 < \Lambda < 90^\circ$.
6	1-10	F10	ξ_f is the abscissa at which the streamline calculation begins. ft. ξ_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 $\Lambda = 40^\circ$ and 60° .
6	11-20	F10	η_f is the ordinate at which the streamline calculation begins. ft.
6	21-30	F10	dx is the increment in distance along the streamline used in the boundary layer calculations (Program A). ft. $dx \geq (\xi_f^2 + \eta_f^2)^{1/2} / 600$
7	1-5	I5	ITC is the number of values in T_w and η tables. $2 \leq ITC \leq 50$. If $ITC \geq 3$ the wall temperature is nonisothermal.
8	----	8F10	This table contains ITC values of η , where η is the distance around the cylinder, ft.
9	----	8F10	This table contains ITC values of T_w corresponding to the η table, $^\circ R$.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
10	1-5	I5	K_{PO} is the number of x printouts desired along the streamline. $K_{PO} < 500$.
10	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
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_{PO} is a table containing locations along the streamline at which printout is desired.

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

TYPE	6	TITLE						
1								
2	M_∞	T_∞	P_∞	WT1				
	u_∞	H_o	P_o/P_{SL}	P_∞	q_∞	WT2		
	ALTITUDE	u_∞	FLIGHT					
3	R_{CYL}	R_{HEMI}	T_w					
4	NPROG 81							
	CYLINDER							
	CYLINDER							
	RADIUS							
5	Λ							
6	ξ_F	η_F	$d\alpha$					
7	5	ITC						

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

η TABLE (ITC VALUES)

8								

T_w TABLE (ITC VALUES)

9								

10	K_{PO}	51	R_r , transition	$\Delta\eta_f$

x TABLE (K_{PO} TABLE)

11								

4) SHARP DELTA WING (Option D-4)

<u>Cards</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	Same as Option D-1		
2			
3			
4	1-6	A-6	Contains the word DELTA.
5	1-5	I5	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.
6	1-10	F10	Angle of attack, degrees.
6	11-20	F10	Wing sweep angle measured from a line normal to the flow, degrees.
6	21-30	F10	ξ_f is the final or end coordinate along the streamline at which heating data is desired, ft.
6	31-40	F10	η_f is the final or end coordinate along the streamline at which heating data is desired, ft.
6	41-50	F10	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.
7	1-10	F10	$\Delta\xi$ is the increment in ξ which the program uses to setup an (ξ, η) array of streamline angles, velocity, and pressures, ft.

$$\Delta\xi \geq \frac{\xi_f}{50}$$

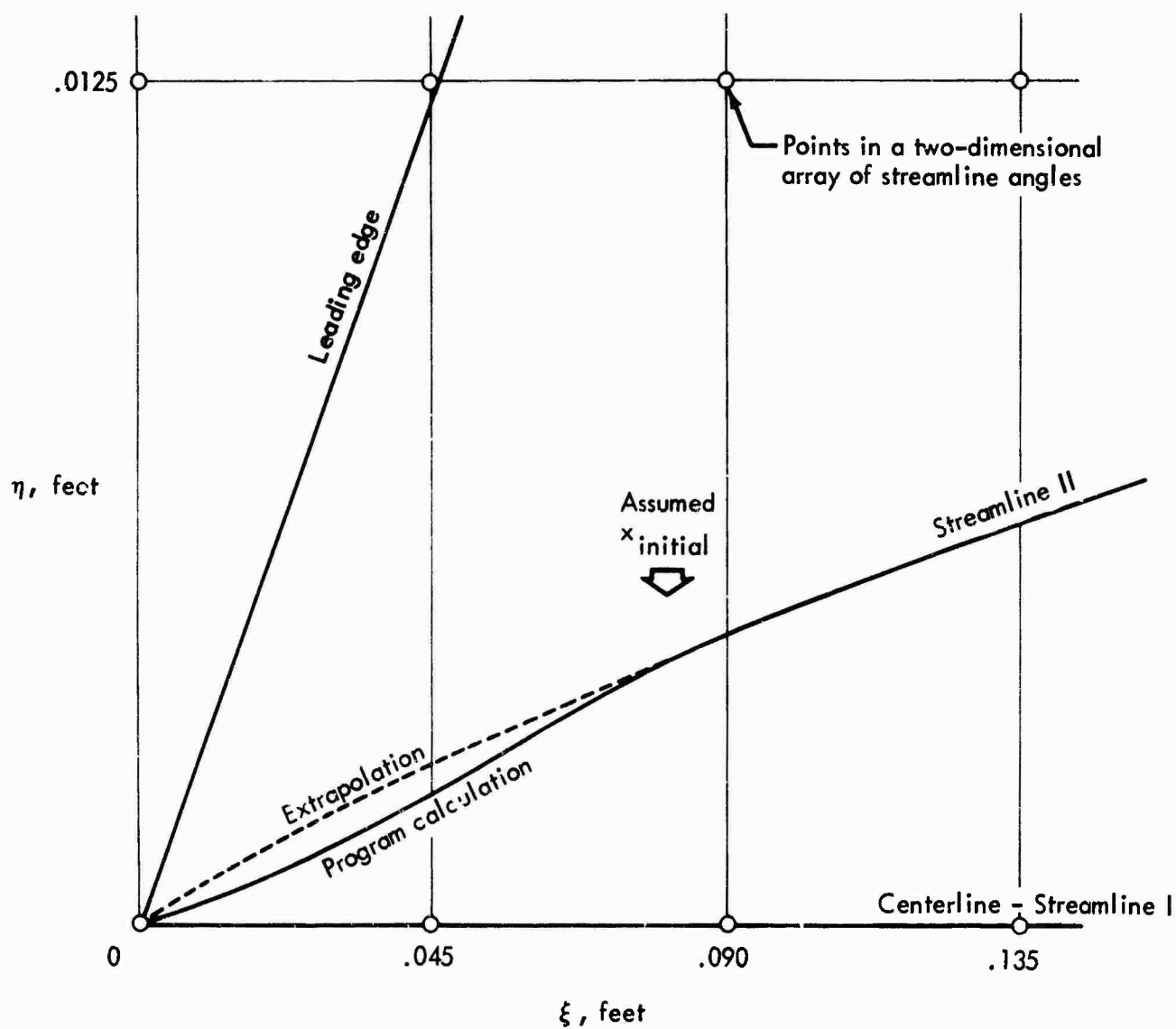
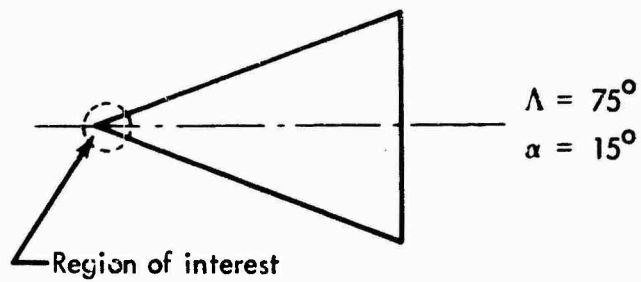


Figure 5: STREAMLINES NEAR THE APEX OF A SHARP DELTA WING

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
7	11-20	F10	$\Delta\eta$ is the increment in η the program uses to setup an (ξ, η) 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. $dx > (x_f - x_I)/750$.
Note: If T_w is a constant value, enter one value into card 11, and delete cards 8, 9, and 10.			
8	1-5	I5	M is the number of points in the ξ array. $2 \leq M \leq 50$.
8	6-10	I5	N is the number of points in the η array. $2 \leq N \leq 50$.
9	----	8F10	This is a table of M values of ξ where wall temperature will be input, if $ITC > 1$.
10	----	8F10	This is a table of N values of η where wall temperature will be input if $ITC > 1$.
11	----	8F10	This is a table containing $(N) \cdot (M)$ values of wall temperature, input so ξ varies with each η .
12	1-5	I5	K_{pQ} is the number of x printouts desired along the streamline. $K_{pQ} < 500$.
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_{pQ} is a table of K_{pQ} locations along the streamline where printout is desired.

1	11	21	31	41	51	61	71	80
TYPE 6	TITLE							
1	[Redacted]							
2	M_{∞}	T_{∞}	P_{∞}	WT1				
	u_{∞}	H_0	P_0/P_{SL}	P_{∞}	q_{∞}	WT2		
	ALTITUDE	u_{∞}	FLIGHT					
3	RCYL	RHEMI	T_w					
4	NPROG 6	DELTA						
5	ITC 5							
6	α	Λ	ξ_f	η_f	x_l			

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

7

$\Delta \xi$	$\Delta \eta$	dx
--------------	---------------	------

8

M	N	10
51		

ξ TABLE (M VALUES)

9

η TABLE (N VALUES)

10

T_w TABLE (M x N VALUES)

11

[illegible]

5) STREAMLINE CALCULATIONS (PROGRAM C)

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	} Same as Option D-1		
2			
3			
4	1-6	A6	Must contain the word STREAM.
5	1-5	I5	M is the number of points in the array ξ . $2 \leq M \leq 50$.
5	6-10	I5	N is the number of points in the array η $2 \leq N \leq 50$.
5	11-15	I5	ITC is an indicator used to test whether the wall temperature is a constant or a variable. If T_w is a constant ITC = 1 and enter one value of T_w into card 16 and delete cards 13, 14, and 15.
6	1-10	F10	ξ_f is ξ at the final or end point, ft.
6	11-20	F10	η_f is η at the final or end point, ft.
6	21-30	F10	dx is the increment in distance along the streamline, ft. $dx \geq \frac{(\xi_f^2 + \eta_f^2)^{1/2}}{750}$
6	31-40	F10	x_I is the point along the streamline at which the calculations will begin, ft.
6	41-50	F10	u_{e,x_I} is the edge velocity at the initial or start of calculations, ft/sec.
7	----	8F10	This is a table of M values of ξ $2 \leq M \leq 50$. θ_e , P/P_0 and T_w will be input as functions of ξ and η , ft.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
8	----	8F10	This is a table of N values of η $2 \leq M \leq 50$. θ_e , P/P_0 and T_w will be input as functions of ξ and η , ft.
9	----	8F10	This is a table of M values of $\eta_{\max}$ as a function of ξ . $\eta_{\max}$ defines the upper or outer boundary of the body, ft.
10	----	8F10	This is a table of M values of the stream- line angle $\theta_{\max}$ along the boundary as a function of ξ , deg.
11	----	8F10	This is a table of (M) · (N) values of the local streamline angle θ_e and should be input along lines of constant η and ξ , deg.
12	----	8F10	This is a table of (M) · (N) values the local pressure divided by the stagnation pressure and should be input along lines of constant η and varying ξ .
13	1-5	I5	MT is the number of ξ values entered into the table $T_w = f(\xi, \eta)$. Also see ITC above.
13	6-10	I5	NT is the number of η values entered into the table $T_w = f(\xi, \eta)$. Also see ITC above.
14	----	8F10	This is a table of MT values of ξ at which T_w will be input. Also see ITC above.
15	----	8F10	This is a table of NT values of η at which T_w will be input. Also see ITC above.
16	----	8F10	This is a table of (MT) · (NT) values of T_w as a function ξ and η , °R. T_w should be input as a function of ξ at constant η .
17	1-5	I5	K_{p0} is the number of x locations along the streamline at which printout is desired. $K_{p0} < 500$.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
17	11-20	F10	Transition Reynolds number. see definition of $R_{T,Q}$ in Program A. labeled QTRANS in section 2. b. 18.
17	21-30	F10	See card 19. Swept infinite cylinder. option D-3.
18	1-10	F10	$x_p()$ is a table of $K_p()$ locations along the streamline at which printout is desired.

	11	21	31	41	51	61	71	80
1	TYPE 61 TITLE							
2	M_∞		T_∞	P_∞	WT1		WT2	
	u_∞		H_0	P_0/PSL	P_∞	q_∞		
	ALTITUDE		u_∞	FLIGHT				
3	RCYL		R_{HEMI}	T_w				
4	NPROG 6							
	STREAM							
5	M 5	N 10	ITC 15					
6	ξ_f	η_f	dx	x_f	$u_\infty x_f$			
7	ξ TABLE (M VALUES)							
8	η TABLE (N VALUES)							

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

ξ TABLE (MT VALUES)

14								

η TABLE (NT VALUES)

15								

T_w TABLE (MT x NT VALUES)

16								

17	K _{PO}	51	R _{f,transition}		Δη _f			

x TABLE (K_{PO} VALUES)

18								

6) REFERENCE CALCULATIONS (PROGRAM REF)

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Descrip'on</u>
1	Same as Option D-1		
2			
3			
4	1-10	A6	Must contain the word REFERENCE.

1	11	21	31	41	51	61	71	80
TYPE 6 TITLE								
1	M_{∞} T_{∞} P_{∞} WT1 u_{∞} H P_o/PSL P_{∞} q_{∞} WT2 ALTITUDE u_{∞} FLIGHT							
2								
3	RCYL		RHEMI		T _w			
4	NPROG							
REFERENCE ✓								

7) BOUNDARY LAYER CALCULATIONS (PROGRAM B)

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	Same as Option D-1		
2			
3			
4	1-6	A6	Must contain the word FLOW.
5	1-5	I5	IS is the number of values in the P/P_0 , θ_e , Δ/Δ_i and r/r_i tables as a function of streamline distance. $2 \leq IS \leq 50$.
6	----	8F10	This is a table containing IS number of x locations, ft, along the streamline at which P/P_0 , θ_e , Δ/Δ_i and r/r_i will be input.
7	----	8F10	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.
8	----	8F10	This table contains IS values of the local streamline angle relative to the axis of symmetry, degrees. The values of θ_c correspond to the above x table, degrees.
9	----	8F10	This table contains IS values of the streamline divergence parameter Δ/Δ_i . The values of Δ/Δ_i correspond to the above x table.
10	---	8F10	This table contains IS values of the body shock geometry parameter r/r_i . The values of r/r_i correspond to the above x table.
11	1-10	F10	x_1 is point at which the calculation will begin, ft.

<u>Card</u>	<u>Column</u>	<u>Format</u>	<u>Description</u>
11	11-20	F10	dx is the increment in distance along the streamline, ft $dx > \frac{x_{\text{FINAL}} - x_1}{750}$
11	21-30	F10	u_{e,x_1} is the edge velocity at ft/sec.
12	1-5	I5	ITC is the number of values in the T_w versus x table. $2 \leq \text{ITC} \leq 50$. If $T_w = \text{constant}$ then $\text{ITC} = 2$.
13	----	8F10	This is a table of x locations (ITC) along the streamline where T_w will be input.
14	----	8F10	This is a table of ITC values of T_w along the streamline corresponding to the x table. If $\text{ITC} = 2$ enter only two values of T_w , °R.
15	1-5	I5	K_{PO} is the number of x locations along the streamline at which printout is desired. $K_{PO} < 500$.
15	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
16	----	8F10	This is a table of K_{PO} locations along the streamline where printout is desired.

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

TYPE 6	TITLE							
--------	-------	--	--	--	--	--	--	--

2	M_∞	T_∞	P_∞	WT1				
	u_∞	H	P_o/P_{SL}	P_∞	q_∞	WT2		
	ALTITUDE	u_∞	FLIGHT					
	R_{CYL}	R_{HEMI}	T_w					

3	NPROG 6
4	FLOW ✓✓
5	IS 5

x TABLE (IS VAL JES)

6								

P/P_o TABLE (IS VALUES)

7								

θ_e TABLE (IS VALUES)

8								

[illegible][illegible]

x_1	dx	$u_e x_1$
-------	------	-----------

ITC 5

[illegible][illegible]

15	K _{PO}	5	R _{r, transition}
----	-----------------	---	----------------------------

[illegible]

8) HEAT TRANSFER CALCULATIONS (PROGRAM A)

Card	Columns	Format	Description
1	Same as Option D-1		
2			
3			
4	1-6	A6	Must contain the word QTRANS.
5	1-5	I5	IS is the number of values in each of the following tables: x , P/P_{SL} , u_e , θ_e , Δ/Δ_i and r/r_i . $2 \leq IS \leq 50$.
6	1-10	F10	x_I is the initial x location along the streamline at which the calculations will start, ft.
6	11-20	F10	dx is the increment in distance along the streamline. (ft.) $dx > \frac{x_{FINAL} - x_I}{750}$
6	21-30	F10	u_{e,x_I} is the boundary layer edge velocity at x_I , ft/sec.
7	----	8F10	This is a table containing x locations (IS) along the streamline at which P/P_{SL} , u_e , θ_e , Δ/Δ_i and r/r_i will be specified, ft.
8	----	8F10	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.
9	----	8F10	This is a table containing IS values of the boundary layer edge velocity corresponding to locations specified in the x table. ft/sec.
10	----	8F10	This is a table containing IS values of the local streamline angle θ_e relative to the axis of symmetry. The angles correspond to x locations specified in the x table. deg.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
11	----	8F10	This is a table containing IS values of the streamline divergence parameter Δ/Δ_i and correspond to x locations specified in the x table.
12	----	8F10	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.
13	1-5	I5	ITC is the number of values in the T_w versus x table. $2 \leq ITC \leq 50$ If T_w is a constant, $ITC = 2$. Input two values of T_w into card 15.
14	----	8F10	This is a table containing ITC values of x along the streamline, ft.
15	----	8F10	This is a table of ITC values of T_w along the streamline and corresponding to the x locations in the preceding table. ($^{\circ}R$)
16	1-5	I5	K_{PO} is the number of x printout locations desired along the streamline. $K_{PO} < 500$.
16	11-20	F10	Transition Reynolds number, see definition of $R_{T,Q}$ in Program A, labeled QTRANS in section 2. b. 18.
16	21-30	F10	BEQL is the ratio of the laminar equivalent distance parameter to the distance along the streamline. If $x_I > 0$ then a value of BEQL is required. If $x_I = 0$ input BEQL=0.
16	31-40	F10	BEQT is the turbulent equivalent distance parameter. See BEQL above.
17	----	8F10	This is a table of K_{PO} locations along the streamline where printout is desired.

11	21	31	41	51	61	71	80
----	----	----	----	----	----	----	----

TYPE	61	TITLE					
1							

2	M_∞	T_∞	P_∞	WT1		WT2	
	u_∞	H	P_o/P_{SL}	P_∞	q_∞		
	ALTITUDE		u_∞	FLIGHT			

3	R_{CYL}	R_{HEMI}	T_w
---	-----------	------------	-------

4	NPROG	61
	QTRANS	

5	IS	51
	x_I	

6	dx	u_e, x_I
---	------	------------

7	x TABLE (IS VALUES)							

8	P/P _{SL} TABLE (IS VALUES)							

9 u_θ TABLE (IS VALUES)

10 θ_0 TABLE (IS VALUES)

11 Δ/Δ_i TABLE (IS VALUES)

12 r/r_i TABLE (IS VALUES)

1	11	21	31	41	51	61	71	80
---	----	----	----	----	----	----	----	----

13

5	ITC
---	-----

14

x TABLE (ITC VALUES)							

15

w TABLE (ITC VALUES)							

16

KPO	5	$R_{r,transition}$	$(b_{eq}/x)_L$	$(b_{eq}/x)_T$
-----	---	--------------------	----------------	----------------

17

x TABLE (KPO VALUES)							

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	$\dot{q}_o/(i_{aw}-i_w) = h_o/C_p$ where h_o is the hemisphere stagnation point heat transfer coefficient, $\text{lb}_m/\text{ft}^2\text{-sec}$
.24HO	.24 h_o/C_p , $\text{Btu}/\text{ft}^2\text{-sec-}^\circ\text{F}$.
QDOTO	hemisphere stagnation point heating rate, $\text{Btu}/\text{ft}^2\text{-sec}$
HREF, L	$\dot{q}_{REF, L}/(i_{aw, L} - i_{w, REF}) = h_{REF, L}/C_p$ where $h_{REF, L}$ is the laminar heat transfer coefficient based on enthalpy at the stagnation line of a 60° swept infinite cylinder, $\text{lb}_m/\text{ft}^2\text{-sec}$
RHEM, REF	hemisphere radius, ft
RCYL, REF	cylinder radius, ft
SWEEP	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, $\text{Btu}/\text{ft}^2\text{-sec}$

HREF, T	$\dot{q}_{REF, T} / (i_{aw, T} - i_{w, REF}) = h_{REF, T} / C_P$ where $h_{REF, T}$ is the turbulent heat transfer coefficient at the stagnation line of a 60° swept infinite cylinder based on enthalpy, $lb_m/ft^2\text{-sec}$
.24HREF, T	.24 $h_{REF, T} / C_P$
QDOT, REF, T	turbulent heating rate at the stagnation line of a 60° swept infinite cylinder, $Btu/ft^2\text{-sec-}^\circ R$
IAW, REF, L	laminar adiabatic wall enthalpy on a 60° swept infinite cylinder, ft^2/sec^2
IAW, REF, T	turbulent adiabatic wall enthalpy on a 60° swept infinite cylinder, ft^2/sec^2
IW, REF	reference wall enthalpy, ft^2/sec^2
TAUREF, L	laminar shear stress on a 60° swept infinite cylinder, lb_f/ft^2
CFINF, REF, L	$2\tau_{ref, L} / (\rho_\infty u_\infty^2)$, laminar skin friction coefficient based on free stream conditions
IW, O	stagnation wall enthalpy, ft^2/sec^2
TAUREF, T	turbulent shear stress on a 60° swept infinite cylinder, lb_f/ft^2
CFINF, REF, T	$2\tau_{ref, T} / (\rho_\infty u_\infty^2)$, turbulent skin friction coefficient based on free stream conditions
TW, REF	wall temperature for 60° swept infinite cylinder heating calculations, $^\circ R$

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
THETA E	streamline angle, degrees

BOUNDARY LAYER CALCULATIONS

X	streamline distance, ft
ETA	ordinate of coordinate system, ft
XSI	abscissa of coordinate system, ft
BEQXL	$(b_{eq}/x)_L$, laminar equivalent distance parameter
BEQXT	$(b_{eq}/x)_T$, 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	Δ/Δ_i , total streamline divergence
N	$x/(\Delta/\Delta_i) [d(\Delta/\Delta_i)/dx]$, streamline divergence parameter
QLISO	laminar isothermal heat transfer rate, $\text{Btu}/\text{ft}^2\text{-sec-}^\circ\text{R}$
QTISO	turbulent isothermal heat transfer rate, $\text{Btu}/\text{ft}^2\text{-sec-}^\circ\text{R}$
RORI	r/r_i , streamline divergence due to body-shock geometry

QDOTL	laminar heating rate, $\text{Btu/ft}^2\text{-sec-}^\circ\text{R}$
QDOTT	turbulent heating rate, $\text{Btu/ft}^2\text{-sec-}^\circ\text{R}$
TW	wall temperature, $^\circ\text{R}$
CFEL	laminar skin friction coefficient
CFET	turbulent skin friction coefficient
	} based on dynamic pressure at boundary layer edge
ZTEX	ZT_e , compressibility factor times the edge temperature, $^\circ\text{R}$
RRQ	heat transfer reference Reynolds number
RRS	skin friction reference Reynolds number
OMEGAE	edge viscosity divided by ZT_e , $\text{lb}_f\text{-sec/ft}^2\text{-}^\circ\text{R}$
AIEE	i_e , edge enthalpy, ft^2/sec^2
AYEW	i_w , wall enthalpy, ft^2/sec^2
THETA	θ_e , local streamline angle, degrees
MUO	μ_o , absolute viscosity at the stagnation reference condition, $\text{lb}_f\text{-sec/ft}^2$
ZTR	compressibility factor-temperature product evaluated at the reference enthalpy, $^\circ\text{R}$
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, $\text{lb}_f\text{-sec/ft}^2\text{-}^\circ\text{R}$

THETAL laminar momentum thickness, ft

THETAT turbulent momentum thickness, ft

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 ξ 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 TABL18 LOOK-UP

An error occurred during table interpolation in subroutine TABLE2 or TABL18.

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 $\theta_e(\xi, \eta)$ grid in subroutine STREAM.

4. PROGRAMMING INFORMATION

a. $\rho_r \mu_r$ Program Listing

```

SIDFTC AS2419 DECK
C
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),AYEWO ) , (A( 5),BEQXL ) , (A( 6),BEQXT ) ,
3(A( 7),DELETA ) , (A( 8),DELXSI ) , (A( 9),DETA ) ,
4(A( 10),DETA ) , (A( 11),DX ) , (A( 12),DXP ) ,
5(A( 13),ETAF ) , (A( 14),ETAI ) , (A( 15),H ) ,
6(A( 16),HO ) , (A( 17),HREF ) , (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 ) ,
1(A( 28),TEMP ) , (A( 29),TWREF ) , (A( 30),URATIO ) ,
2(A( 31),VEL ) , (A( 32),VELP ) , (A( 33),XF ) ,
3(A( 34),XI ) , (A( 35),XSIF ) , (A( 36),XSII ) ,
4(A( 37),XSI ) , (A( 38),ETA ) , (A( 39),Y ) ,
5(A( 137),ETAMAX ) , (A( 187),TH8MX ) , (A( 237),THETA ) ,
6(A( 237),POPSLX ) , (A( 987),DODI ) , (A( 1737),RORI ) ,
7(A( 2737),PRESSR ) , (A( 2737),AIEE ) , (A( 3487),OMEGAE ) ,
8(A( 4237),ZTEX ) , (A( 5237),TWALL ) , (A( 5237),X ) ,
9(A( 5987),XSIX ) , (A( 6737),ETAX ) , (A( 7737),XSTAR ) ,
EQUIVALENCE
1(A( 8487),PRESRX ) , (A( 9237),THETAS ) , (A( 9987),UE ) ,
2(A( 10737),TW ) , (A( 11487),XPO ) , (A( 11587),P ) ,
EQUIVALENCE
1(IA( 1),M ) , (IA( 2),N ) , (IA( 3),KF ) ,
3(IA( 4),I ) , (IA( 8),NPR ) , (IA( 9),ITC ) ,
DIMENSION XSI(50),ETA(50),Y(50),ETAMAX(50),TH8MX(50),
1THETA(50,50),PRESSR(50,50),XSTAR(750),PRESRX(750),POPSLX(750),
2UE(700),AIEE(750),ZTEX(750),OMEGAE(750),THETAS(750),DODI(750),
3 RORI(750),TW(750),XPO(100),TWALL(50,50)
DIMENSION P(5),ETAX(750),XSIX(750),X(750)
DIMENSION DODS(50),IPROG(9),ITYPE(3),PS(50),RORS(50),TWS(50),
1UES(50),XS(50),THETS(50)
DIMENSION XSIT(50),ETAT(50)
DIMENSION TITLE(10)
DATA (ITYPE(1),I=1,3)/6HWT1 ,6HWT2 ,6HFLIGHT/
DATA(IPROG(1),I=1,9)/6HREFERE,6HAXISYM,6HTWODIM,6HHEMISP,
16HDELTA ,6HCYLIND,6HSTREAM,6HFLOW ,6HTRANS/
5000 FORMAT(15I5)

```

```

5010 FORMAT(8F10.0)
5020 FORMAT(A6,4X,10A6)
5050 FORMAT(I5,5X,5F10.0)
5060 FORMAT(2F10.0,10I5)
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
        L1=I
        IF(NTYPE-I*TYPE(I))1,3,1
          1 CONTINUE
          WRITE(6,6010)NTYPE
6010 FORMAT(1H0,29HERROR IN CONTROL CARD (NTYPE=,A6,2H ))
          STOP
          3 CONTINUE
C
      GO TO (10,20,30),L1
      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,RHMRtF,TWREF
      READ(5,5020)NPROG
      WRITE(6,6015)NPROG
6015 FORMAT(1H0,A6)
      CALL INTIAL(L1,NTYPE)
C
      CALL REF
      DO 60 I=1,9
        IF(NPROG-I*PROG(I))60,50,60
          50 NPR=I
          GO TO 70
          60 CONTINUE
          WRITE(6,6020)NPROG

```

```

24190042
24190043
24190044
24190045
24190046
24190047
24190048
24190049
24190050
24190051
24190052
24190053
24190054
24190055
24190056
24190057
24190058
24190059
24190060
24190061
24190062
24190063
24190064
24190065
24190066
24190067
24190068
24190069
24190070
24190071
24190072
24190073
24190074
24190075
24190076
24190077
24190078
24190079
24190080
24190081
24190082
24190083

```

```

6020 FORMAT(30H0ERROR IN CONTROL CARD (NPROG=,A6,1H))
      STOP
C
      70 CONTINUE
      GO TO(100,200,300,400,500,600,700,800),NPR
C
      100 CONTINUE
      GO TO 1000
C
      200 CONTINUE
      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 BEQ1
      CALL QTRAN
      GO TO 1000
C
      300 CONTINUE
      C SWEPT CYLINDER
      READ(5,5010)RADIUS,SWEEP
      READ(5,5010)XSIF,ETAF,DX
      WRITE(6,6300) RADIUS,SWEEP,XSIF,ETAF,DX
      6300 FORMAT(1H0,14H SWEPT CYLINDER //3X,6HRADIUS,6X,5HSWEEP,7X,4HXSIF,8X,24190115
      1,4HETAF,8X,2HDX//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,DETAS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL CYLIND
      CALL STREAM

```

24190084
 24190085
 24190086
 24190087
 24190088
 24190089
 24190090
 24190091
 24190092
 24190093
 24190094
 24190095
 24190096
 24190097
 24190098
 24190099
 24190100
 24190101
 24190102
 24190103
 24190104
 24190105
 24190106
 24190107
 24190108
 24190109
 24190110
 24190111
 24190112
 24190113
 24190114
 24190115
 24190116
 24190117
 24190118
 24190119
 24190120
 24190121
 24190122
 24190123
 24190124

```

CALL DIVERG
CALL FLOW
NXT=1
DO 350 I=1,II
350 TW(I)=TBLP(ETAT,TWS,ETAX(I),ITC,NXT)
CALL BEQI
CALL QTRAN
GO TO 1000

C
400 CONTINUE
      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)KF,RTRANS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL WALLT1(XS,TWS)
      CALL BEQI
      CALL QTRAN
      GO TO 1000

C
500 CONTINUE
      DELTA WING
      READ(5,5000)ITC
      XSII=0.
      ETAI=0.
      READ(5,5010)ALPHA,SWEEP,XSIF,ETAF,XI
      READ(5,5010)DELXSI,DELETA,UX
      WRITE(6,6500)ALPHA,SWEEP,XSII,ETAI,XSIF,ETAF,DELXSI,
1DELETA,DX
6500 FORMAT(11H0DELTA WING //3X,5SHALPHA,7X,5HSWEEP,7X,4HXSII,8X,4HETAI,
18X,4HXSIF,8X,4HETAF,8X,6HDELXSI,6X,6HDELETA,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)(ETAT(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
        TW(I)=TWC
515  CONTINUE
520  CONTINUE
      READ(5,5050)KF,RTRANS,DETA5
      READ(5,5010)(XPO(I),I=1,KF)
      CALL DIVERG
      IF(ITC .EQ. 1) GO TO 505
      CALL WALLT2(MT,NT,XSIT,ETAT)
505  CONTINUE
      CALL FLOW
      CALL BEQI
      CALL QTRAN
      GO TO 1000
C
600  CONTINUE
      STREAMLINE
      READ(5,5000)M,N,ITC
      READ(5,5010)XSIF,ETAF,DX,XI,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)(TH8MX(J),J=1,M)
      READ(5,5010)((THETA(I,J),I=1,M),J=1,N)
      READ(5,5010)((PRESSR(I,J),I=1,M),J=1,N)
      URATIO=URATIO/VEL
      IF(ITC .EQ. 1)GO TO 610
      READ(5,5000)MT,NT
      READ(5,5010)(XSIT(I),I=1,MT)
      READ(5,5010)(ETAT(I),I=1,NT)
      READ(5,5010)((TWALL(I,J),I=1,MT),J=1,NT)
      GO TO 620
610  CONTINUE
      READ(5,5010)TWC
620  CONTINUE

```

```

24190167
24190168
24190169
24190170
24190171
24190172
24190173
24190174
24190175
24190176
24190177
24190178
24190179
24190180
24190181
24190182
24190183
24190184
24190185
24190186
24190187
24190188
24190189
24190190
24190191
24190192
24190193
24190194
24190195
24190196
24190197
24190198
24190199
24190200
24190201
24190202
24190203
24190204
24190205
24190206
24190207

```

```

READ(5,5050)KF,RTRANS,DETAS
READ(5,5010)(XPO(I),I=1,KF)
CALL STREAM
CALL DIVERG
IF(ITC.EQ.1) GO TO 625
CALL WALLT2(MT,NT,XSIT,ETAT)
GO TO 630
625 DO 615 I=1,II
615 TW(I)=TWC
630 CONTINUE
CALL FLOW
CALL BEQI
CALL QTRAN
GO TO 1000

C 700 CONTINUE
C FLOW CONDITIONS
READ(5,5000)IS
READ(5,5010)(XS(I),I=1,IS)
READ(5,5010)(PS(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)
READ(5,5010)XI,DX,URATIO
II=(XS(IS)-XI)/DX
II=II+1
XSTAR(1)=XI
DO 720 I=1,II
XSTAR(I+1)=XSTAR(I)+DX
X(I)=XSTAR(I)
XSIX(I)=0.
ETAX(I)=0.
PRESRX(I)=TBLP(XS,PS,XSTAR(I),IS,NXSV)
THETAS(I)=TBLP(XS,THETS,XSTAR(I),IS,NXSV)
DODI(I)=TBLP(XS,DODS,XSTAR(I),IS,NXSV)
RORI(I)=TBLP(XS,RORS,XSTAR(I),IS,NXSV)
720 CONTINUE
P(1)=PS(1)
P(2)=PS(2)
P(3)=PS(3)
DXP=.5*(XS(3)-XS(1))
URATIO=URATIO/VEL

```

```

24190208
24190209
24190210
24190211
24190212
24190213
24190214
24190215
24190216
24190217
24190218
24190219
24190220
24190221
24190222
24190223
24190224
24190225
24190226
24190227
24190228
24190229
24190230
24190231
24190232
24190233
24190234
24190235
24190236
24190237
24190238
24190239
24190240
24190241
24190242
24190243
24190244
24190245
24190246
24190247
24190248
24190249

```

```

      READ(5,5000)ITC
      READ(5,5010)(XS(I),I=1,ITC)
      READ(5,5010)(TWS(I),I=1,ITC)
      CALL WALLT1(XS,TWS)
      READ(5,5050)KF,RTRANS
      READ(5,5010)(XPO(I),I=1,KF)
      CALL FLOW
      CALL BEQI
      CALL QTRAN
      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(1)=XS(1)
      X(1)=XSTAR(1)
      XSIX(1)=0.
      ETAX(1)=0.
      POPSLX(1)=PS(1)
      UE(1)=UES(1)
      AIEE(1)=H-UE(1)**2/2.
      CALL SOMEGA(AIEE(1),POPSLX(1),ZTEX(1),OMEGAE(1))
      THETAS(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.
      POPSLX(I)=TBLP(XS,PS,XSTAR(I),IS,NXSV)
      UE(I)=TBLP(XS,UES,XSTAR(I),IS,NXSV)

```

```

24190250
24190251
24190252
24190253
24190254
24190255
24190256
24190257
24190258
24190259
24190260
24190261
24190262
24190263
24190264
24190265
24190266
24190267
24190268
24190269
24190270
24190271
24190272
24190273
24190274
24190275
24190276
24190277
24190278
24190279
24190280
24190281
24190282
24190283
24190284
24190285
24190286
24190287
24190288
24190289
24190290

```

```

      AIEE(I)=H-UE(I)**2/2.
      CALL SOMEGA(AIEE(I),POPSLX(I),ZTEX(I),OMEGAE(I))
      THETAS(I)=TBLP(XS,THETS,XSTAR(I),IS,NXSV)
      DODI(I)=TBLP(XS,DODS,XSTAR(I),IS,NXSV)
      RORI(I)=TBLP(XS,RORS,XSTAR(I),IS,NXSV)
820  CONTINUE
      URATIO=URATIO/VEL
      READ(5,5000)ITC
      READ(5,5010)(XS(I),I=1,ITC)
      READ(5,5010)(TWS(I),I=1,ITC)
      CALL WALLT1(XS,TWS)
      READ(5,5050)KF,RTRANS,BEQXL,BEQXT
      READ(5,5010)(XPO(I),I=1,KF)
      IF(BEQXL)825,825,830
825  CONTINUE
      CALL BEQ1
830  CONTINUE
      CALL QTRAN
      GO TO 1000
      END
$18FTC DBTP      DECK
C      FUNCTION DBTP USES TBLP TO PERFORM DOUBLE LINEAR INTERPOLATION
      FUNCTION DBTP(X,YY,ZZ,X,Y,NX,NY,NXS,NYS)
      DIMENSION XX(NX),YY(NY),ZZ(50,50)
      ML=0
      IF(X-XX(1))10,20,30
10  ML=1
20  NX5=1
      X1=XX(1)
      X2=0.
      GO TO 170
30  CONTINUE
      IF(X-XX(NX))60,50,40
40  ML=1
50  CONTINUE
      NX5=NX
      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
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)172,175,172
172 WRITE(6,6010)N,X,XX(1),NX,XX(NX)
6010 FORMAT(1H0,10HDBTP 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
SIBFCTIBLP    DECK

```

```

C
C
FUNCTION TBLP(XT,YT,X,NTAB,N)
DIMENSION XT(NTAB),YT(NTAB)
  THIS IS A MODIFICATION OF THE UNIVAC 1107 ROUTINE

  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
  IF(X-XT(I))7,12,11
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
  DIVERG CALCULATES D/DI,P,T,THETA AT DX INCREMENTS
  ALONG STREAMLINE
  COMMON/Q/A(11600),IA(10)
  EQUIVALENCE
1(A( 10),DETS) , (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),XSI) ,
4(A( 87),ETA) , (A( 137),ETAMAX) , (A( 187),TH8MX) ,
5(A( 987),UODI) , (A( 1737),RORI) , (A( 2737),PRESSR) ,
6(A( 5237),X) , (A( 5987),XSIX) , (A( 6737),ETAX) ,
7(A( 7737),XSTAR) , (A( 8487),PRESRX) , (A( 9237),THETAX) ,
7(IA( 1),M) ,(IA( 2),N) ,(IA( 4),I) ,
7(IA( 8),NPR) ,

```

24190374
 24190375
 24190376
 24190377
 24190378
 24190379
 24190380
 24190381
 24190382
 24190383
 24190384
 24190385
 24190386
 24190387
 24190388
 24190389
 24190390
 24190391
 24190392
 24190393
 24190394
 24190395
 24190396
 24190397
 24190398
 24190399
 24190400
 24190401
 24190402
 24190403
 24190404
 24190405
 24190406
 24190407
 24190408
 24190409
 24190410
 24190411
 24190412
 24190413
 24190414
 24190415

```

      6(A(11587),P      ), (A( 2737),THETAD)
      EQUIVALENCE (THETAX,THETAS)

      DIMENSION XSID(750,2),ETAD(750,2),XD(750,2),THETAD(750,2),XSIX(750,2),XSI(50),
      1)ETAX(750),X(750),THETAX(750),ETA(50),XSI(50),
      2PRESSRX(750),
      DIMENSION DODI(750),RORI(750),THETAS(750),P(5)
      DIMENSION ETAMAX(50)

      REWIND 3
      WRITE(3)((PRESSR(I,J),I=1,M),J=1,N)
      NX=0
      SAVE ORIGINAL STREAMLINE DATA
      I2=I1
      DO 10 I=1,I2
      XSID(I,2)=XSIX(I)
      ETAD(I,2)=ETAX(I)
      XD(I,2)=X(I)
      THETAD(I,2)=THETAX(I)
      XSIS2=XF
10  CONTINUE
      CALCULATE STREAMLINE 1
      ETAF=ETAF
      NXSV=1
      ETAMX=TLPL(XSI,ETAMAX,XSIF,M,NXSV)
      ETAF=ETAF-DETA
      IF(ETAF.GT.ETA(1)) GO TO 15
      ETAF=ETAF+DETA
      NX=1
15  CONTINUE
      CALL STREAM
      I1=I1
      DO 20 I=1,I1
      XSID(I,1)=XSIX(I)
      ETAD(I,1)=ETAX(I)
      XD(I,1)=X(I)
      THETAD(I,1)=THETAX(I)
      XSIS1=XF
20  CONTINUE
      IF(XSIS1.GT.XSIS2) GO TO 22
      GO TO 45

```

```

22 CONTINUE
  I1=I1+1
  I=I1
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(1,1)=XSID(1,2)
  ETAD(1,1)=ETAD(2,1)
45 CONTINUE
  WRITE(6,6100)
6100 FORMAT(1H0,37HSTREAMLINE COORDINATES AND FLOW ANGLE //19H PRIMARY
1STREAMLINE ,36X,20HAUXILIARY STREAMLINE//10X,3HXS1,9X,3HETA,9X,1HX24190472
2,11X,5HTHETA,27X ,3HXS1,9X,3HETA,9X,1HX,11X,5HTHETA/)
  DO 12 I=1,I2
    WRITE(6,6200)I,XSID(I,2),ETAD(I,2),XD(I,2),THETAD(I,2),I,XSID(I,1)
1,ETAD(I,1),XD(I,1),THETAD(I,1)
6200 FORMAT(1H ,15,2X,4E12.5,10X,15,2X,4E12.5)
12 CONTINUE
  C      CALCULATE P,T,THETA,D/DI AT DX INCREMENTS
  NXS=1
  NYS=1
  NS2=1
  NS1=1
  XSTAR(1)=XI
  XF=XD(I2,2)-3.*DX
44 CONTINUE
  DO 51 I=i,I2
    THETAS(I)=TBLP(XD(1,2),THETAD(1,2),XSTAR(I),I2,NS2)
51 XSTAR(I+1)=XSTAR(I)+DX
    NS2=1
    NI=0
  DO 50 I=1,749
    XSI=TBLP(XD(1,2),XSID(1,2),XSTAR(I),I2,NS2)
    ETAS= TBLP(XD(1,2),ETAD(1,2),XSTAR(I),I2,NS2)
    ETA1=TBLP(XSID(1,1),ETAD(1,1),XSI,I1,NS1)
    ETA2=TBLP(XSID(1,2),ETAD(1,2),XSI,I2,NS2)
    IF(NX .GT. 0) GO TO 41
    IF(ETA1 .LT. ETA2) GO TO 42

```

```

99 CONTINUE
WRITE(6,6099)XSIS
6099 FORMAT(1H0,26HSTREAMLINES 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
PRESRX(1)=DBTP(ETA,XSI,PRESSR,ETAS,XSIS,N,M,NYS,NXS)
IF(XSTAR(1) .GT. XF)GO TO 60
XSTAR(1+1)=XSTAR(1)+DX
II=1
50 CONTINUE
60 CONTINUE
C      RESTORE ORIGINAL STREAMLINE
DO 100 I=1,I2
XSIX(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.)GO TO 1890

```

```

24190499
24190500
24190501
24190502
24190503
24190504
24190505
24190506
24190507
24190508
24190509
24190510
24190511
24190512
24190513
24190514
24190515
24190516
24190517
24190518
24190519
24190520
24190521
24190522
24190523
24190524
24190525
24190526
24190527
24190528
24190529
24190530
24190531
24190532
24190533
24190534
24190535
24190536
24190537
24190538
24190539

```

```

P(1)=PRESRX(1)
P(2)=PRESRX(2)
P(3)=PRESRX(3)
DXP=DX
1890 CONTINUE
RETURN
END
$IBFTC 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
GO TO 170
30 IF(Z-ZA(NZ))60,50,40
40 ML=1
50 NZS=NZ
GO TO 170
60 IF(Z-ZA(NZS))80,170,120
80 N=NZS-1
90 IF(Z-ZA(N))100,160,160
100 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)
6010 FORMAT(1H0,32H VARIABLE NOT IN RANGE OF TABLE, /5H Z = ,E12.5,9H ZA,24190576
1(1) = ,E12.5,4H ZA(,13,4H) = ,E12.5)
180 CONTINUE
RETURN
END
$IBFTC STREAM DECK

```

```

24190540
24190541
24190542
24190543
24190544
24190545
24190546
24190547
24190548
24190549
24190550
24190551
24190552
24190553
24190554
24190555
24190556
24190557
24190558
24190559
24190560
24190561
24190562
24190563
24190564
24190565
24190566
24190567
24190568
24190569
24190570
24190571
24190572
24190573
24190574
24190575
24190576
24190577
24190578
24190579
24190580
24190581

```

```

C
C
SUBROUTINE STREAM
CALCULATES STREAMLINE COORDINATES, PRESSURE AND DIVERGENCE
PARAMETERS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A( 11),DX ) , (A( 13),ETAF ) , (A( 33),XF ) ,
2(A( 34),XI ) , (A( 35),XSIF ) , (A( 37),XSI ) ,
3(A( 87),ETA ) , (A( 137),ETAMAX ) , (A( 187),TH8MX ) ,
4(A( 237),THETA ) , (A( 5237),X ) , (A( 5987),XSIX ) ,
5(A( 6737),ETAX ) , (A( 9237),THETAS ) , (A( 8),DELXSI) ,
6(A( 8487),ETAS ) , (A(10737),XSIS ) , (A( 9987),XS ) ,
7(A( 7737),THETS ) ,
6(IA( 1),M ) , (IA( 2),N ) , (IA( 4),II )
EQUIVALENCE (THETA,X,THETAS)
DIMENSION XSI(50),ETA(50),ETAMAX(50),TH8MX(50),THETA(50,50)
DIMENSION ETAS(750),XSIS(750),XS(750),ETAX(750),XSIX(750),X(750),
1THETAS(750),THETS(750),THETAX(750)
DIMENSION D(18)
DATA (D(I),I=1,18)/18*6H*****/
RADIAN=57.2958

C
IF(17.EQ. 1) GO TO 90
WRITE(6,6000) (XSI(I),I=1,9),D
6000 FORMAT(19H0STREAMLINE ANGLES //16H THETA(XSI,ETA)//19X,3HXSI/16X,
1 9E12,5/16X,18A6)
DO 10 J=1,N
WRITE (6,6010) ETA(J),(THETA(I,J),I=1,9)
6010 FORMAT(4H ETA,E11.4,2H *,9E12,5)
10 CONTINUE
M1=1
M2=9
15 CONTINUE
IF'M .LE. M2) GO TO 80
M1=M1+9
M2=M2+9
WRITE(6,6020) (XSI(I),I=M1,M2),D
6020 FORMAT(19X,3HXSI/16X,9E12,5/16X,18A6)
DO 20 J=1,N
WRITE(6,6010) ETA(J),(THETA(I,J),I=M1,M2)
20 CONTINUE
GO TO 15
24190582
24190583
24190584
24190585
24190586
24190587
24190588
24190589
24190590
24190591
24190592
24190593
24190594
24190595
24190596
24190597
24190598
24190599
24190600
24190601
24190602
24190603
24190604
24190605
24190606
24190607
24190608
24190609
24190610
24190611
24190612
24190613
24190614
24190615
24190616
24190617
24190618
24190619
24190620
24190621
24190622

```

24190623
24190624
24190625
24190626
24190627
24190628
24190629
24190630
24190631
24190632
24190633
24190634
24190635
24190636
24190637
24190638
24190639
24190640
24190641
24190642
24190643
24190644
24190645
24190646
24190647
24190648
24190649
24190650
24190651
24190652
24190653
24190654
24190655
24190656
24190657
24190658
24190659
24190660
24190661
24190662
24190663
24190664

```

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.
  XSIS1=XSIF
  ETAS1=ETAF

  CALL FIND(XSI,XSIS1,M,NX)
  CALL FIND(ETA,ETAS1,N,NY)
  THET1=THETA(NX,NY+1)
  THET2=THETA(NX+1,NY+1)
  THET3=THETA(NX+1,NY)
  THET4=THETA(NX,NY)
  TH8S1=DBTP(ETA,XSI,THETA,ETAS1,XSIS1,N,M,NYS,NXS)
  THETS(1)=TH8S1
  XSIS(1)=XSIF
  ETAS(1)=ETAF
  XS(1)=0.

105 CONTINUE
  DXSI=DX*COS(TH8S1/RADIAN)
  DETA=DX*SIN(TH8S1/RADIAN)
  IF(TH8S1.LE.0.) DXSI=DX
  IF(TH8S1.LE.0.) DETA=0.
  ETAS2=ETAS1-DETA
  IF(TH8S1.LE.0.) ETAS2=ETAS1
  XSIS2=XSIS1-DXSI
  XS2=XS1+DX
  ETAD=XSIS2

```

```

114 IF(ETAS2,GE, ETA(1)) GO TO 130
115 ETAS2=ETA(1)
116 XSI2=XSI1-(ETAS1-ETA(1))/TAN(TH8S1/RADIAN)
117 XS2=XSI1+(ETAS1-ETA(1))/SIN(TH8S1/RADIAN)
118 IX=1
119 GO TO 135
120 130 CONTINUE
121 IX=0
122 135 CONTINUE
123 IF(XSI2,GE, XSI(1)) GO TO 145
124 XSI2=XSI(1)
125 ETAS2=ETAS1-(XSI1-XSI(1))*TAN(TH8S1/RADIAN)
126 XS2=XSI1+(XSI1-XSI(1))/COS(TH8S1/RADIAN)
127 IX=1
128 145 CONTINUE
129 IF(IX.GT.0)GO TO 2000
130 CALL FIND(XSI,XSIS2,M,NX)
131 XSI1=XSI(NX)
132 XSI2=XSI(NX+1)
133 XSI3=XSI(NX+1)
134 XSI4=XSI(NX)
135 DC=XSI2-XSI1
136 CALL FIND(ETA,ETAS2,N,NY)
137 ETA1=ETA(NY+1)
138 ETA2=ETA(NY+1)
139 ETA3=ETA(NY)
140 ETA4=ETA(NY)
141 DE=ETA1-ETA4
142 THET1=THETAE(NX,NY+1)
143 THET2=THETAE(NX+1,NY+1)
144 THET3=THETAE(NX+1,NY)
145 THET4=THETAE(NX,NY)
146 ETAMX1=TBLP(XSI,ETAMAX,XSI1,M,NXSV)
147 ETAMX2=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
148 ETAMXS=TBLP(XSI,ETAMAX,XSI2,M,NXSV)
149 SLOPE=ETAMX2-ETAMX1
150 IF(ETAS2-ETAMXS)150,150,1900
151 150 CONTINUE
152 IF(SLOPE)152,152,200

```

C

C

```

152 CONTINUE
   IF(ETA2-ETAMX2)1100,1100,154
154 CONTINUE
   IF(ETA1-ETAMX1)156,156,158
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)220,220,230
220 CONTINUE
   IF(ETA4-ETAMX1)1200,1200,1300
230 CONTINUE
   IF(ETA3-ETAMX2)240,240,999
240 CONTINUE
   IF(ETA4-ETAMX1)1400,1400,1500
C
C      ROUTINES TO CALCULATE THETA AT XSIS2, ETAS2
C
1100 CONTINUE
      ALL POINTS ON BODY
      TH8S2=DBTP(ETA,XSI,THETA,ETAS2,XSIS2,N,M,NY,NXS)
      GO TO 2000
C
1200 CONTINUE
      POINTS 2-3-4 ON BODY - POSITIVE SLOPE
      TH83=THET4+(XSIS2-XSI4)/DC*(THET3-THET4)
      IF(ETAMXS.LE.ETA1) GO TO 1210
      THM1=TBPLP(ETAMAX,TH8MX,ETA2,M,NXSV)
      XSIM=TBPLP(ETAMAX,XSI,ETA2,M,NXSV)
      TH81=THM1+(XSIS2-XSIM)/(XSI2-XSIM)*(THET2-THM1)
      TH8S2=TH83+(ETAS2-ETA3)/DE*(TH81-TH83)
      GO TO 1220
1210 CONTINUE
      TH81=TBPLP(XSI,TH8MX,XSIS2,M,NXSV)
      ETAM=TBPLP(XSI,ETAMAX,XSIS2,M,NXSV)
      TH8S2=TH83+(ETAS2-ETA3)/(ETAM-ETA3)*(TH81-TH83)

```

```

24190706
24190707
24190708
24190709
24190710
24190711
24190712
24190713
24190714
24190715
24190716
24190717
24190718
24190719
24190720
24190721
24190722
24190723
24190724
24190725
24190726
24190727
24190728
24190729
24190730
24190731
24190732
24190733
24190734
24190735
24190736
24190737
24190738
24190739
24190740
24190741
24190742
24190743
24190744
24190745
24190746
24190747

```

```

1220 CONTINUE
GO TO 2000
C
1300 CONTINUE
POINTS 2-3 ON BODY - POSITIVE SLOPE
THM=TLPL(ETAMAX,TH8MX,ETA3,M,NXSV)
XSIM=TLPL(ETAMAX,XSI,ETA1,M,NXSV)
TH83=THM+(XSI2-XSIM)/(XSI3-XSIM)*(THET3-THM)
IF(ETAMXS.LE.ETA1) GO TO 1310
THM=TLPL(ETAMAX,TH8MX,ETA1,M,NXSV)
XSIM=TLPL(ETAMAX,XSI,ETA1,M,NXSV)
TH81=THM+(XSI2-XSIM)/(XSI2-XSIM)*(THET2-THM)
TH82=TH83+(ETAS2-ETA3)/DE*(TH81-TH83)
GO TO 1320
1310 CONTINUE
TH81=TLPL(XSI,TH8MX,XSI2,M,NXSV)
ETAM=TLPL(XSI,ETAMAX,XSI2,M,NXSV)
TH82=TH83+(ETAS2-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=TLPL(XSI,TH8MX,XSI2,M,NXSV)
ETAM=TLPL(XSI,ETAMAX,XSI2,M,NXSV)
TH82=TH83+(ETAS2-ETA3)/(ETAM-ETA3)*(TH81-TH83)
GO TO 2000
1500 CONTINUE
ONE POINT ON BODY (POINT 3)
POSITIVE SLOPE
TH81=TLPL(ETAMAX,TH8MX,ETA2,M,NXSV)
TH82=TLPL(XSI,TH8MX,XSI2,M,NXSV)
ALPHA=ATAN((ETAS2-ETA3)/(XSI3-XSI2))
ETAMX=TLPL(XSI,ETAMAX,XSI2,M,NXSV)
XSIMX=TLPL(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=SQRT(CD**2+BD**2)
TH83=TH81+CE/BC*(TH82-TH81)
ED=SIN(GAM)/SIN(DEL)*CD
AD=(XSI3-XSIS2)/COS(ALPHA)
EA=ED+AD
TH84=THETA(NX+1,NY)
TH852=TH83+ED/EA*(TH84-TH83)
GO TO 2000

C 1600 CONTINUE
TH81=THET4+(XSIS2-XSI4)/DC*(THET3-THET4)
TH82=TBLP(XSI,TH8MX,XSIS2,M,NXSV)
TH852=TH81+(ETAS2-ETA3)/(ETAMXS-ETA3)*(TH82-TH81)
GO TO 2000

C 1700 CONTINUE
NEGATIVE SLOPE
TWO POINTS ON BODY (POINTS 1 AND 4)
TH81=THET4+(ETAS2-ETA4)/DE*(THET1-THET4)
TH82=TBLP(ETAMX,TH8MX,ETAMXS,M,NXSV)
XSIMX=TBLP(ETAMX,XSI,ETAS2,M,NXSV)
TH852=TH81+(XSIS2-XSI4)/(XSIMX-XSI4)*(TH82-TH81)
GO TO 2000

C 1800 CONTINUE
ONE POINT ON BODY (POINT 4)
TH81=TBLP(ETAMX,TH8MX,ETAS2,M,NXSV)
TH82=TBLP(XSI,TH8MX,XSIS2,M,NXSV)
ALPHA=ATAN(ETAS2-ETA4)/(XSIS2-XSI4)
ETAMX=TBLP(XSI,ETAMX,XSIS2,M,NXSV)
XSIMX=TBLP(ETAMX,XSI,ETAS2,M,NXSV)
CD=XSIMX-XSIS2
BD=ETAMX-ETAS2
GAM=ATAN(BD/CD)
DEL=3.14159-ALPHA-GAM
BC=SQRT(BD**2+CD**2)
CE=SIN(ALPHA)/SIN(DEL)*CD
TH83=TH81+CE/BC*(TH82-TH81)
ED=SIN(GAM)/SIN(DEL)*CD
AD=(XSIS2-XSI4)/COS(ALPHA)
EA=ED+AD
TH84=THETA(NX,NY)

```

24190789
 24190790
 24190791
 24190792
 24190793
 24190794
 24190795
 24190796
 24190797
 24190798
 24190799
 24190800
 24190801
 24190802
 24190803
 24190804
 24190805
 24190806
 24190807
 24190808
 24190809
 24190810
 24190811
 24190812
 24190813
 24190814
 24190815
 24190816
 24190817
 24190818
 24190819
 24190820
 24190821
 24190822
 24190823
 24190824
 24190825
 24190826
 24190827
 24190828
 24190829
 24190830

```

C      TH8S2=TH83+ED/EA*(TH84-TH83)
      GO TO 2000
C
      1900 CONTINUE
      C      * POINT IS OFF OF BODY
      1710 CONTINUE
      XSIMX=TBPLP(ETAMAX,XS1,ETAS2,M,NXSV)
      ETAMX=ETAS2+(XSIMX-XSIS2)*TAN(TH8S1/RADIAN)
      IF(ABS(ETAMX-ETAS2)-1.E-06)1750,1750,1720
      1720 CONTINUE
      XSIS2=XSIMX
      ETAS2=ETAMX
      GO TO 1710
      1750 CONTINUE
      XSIS2=XSIMX
      ETAS2=ETAMX
      TH8S2=TBPLP(ETAMAX,TH8MX,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(1)+.00001)XSIS(I)=0.
      IF(ETAD .LE. ETA(1)+.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

```

```

24190831
24190832
24190833
24190834
24190835
24190836
24190837
24190838
24190839
24190840
24190841
24190842
24190843
24190844
24190845
24190846
24190847
24190848
24190849
24190850
24190851
24190852
24190853
24190854
24190855
24190856
24190857
24190858
24190859
24190860
24190861
24190862
24190863
24190864
24190865
24190866
24190867
24190868
24190869
24190870
24190871

```

```

C      2100 CONTINUE
C      REORDER STREAMLINE COORDINATES
      XSIX(1)=XSI(1)
      ETAX(1)=ETAS(1)
      X(1)=0.
      THETAX(1)=THETS(1)
      J=1
      DO 2200 J=2,I
      K=I-J+1
      XSIX(J)=XSI(K)
      ETAX(J)=ETAS(K)
      X(J)=XS(I)-XS(K)
      THETAX(J)=THETS(K)
2200 CONTINUE
3000 CONTINUE
      XF=XSI(1)
      II=I
C
      GO TO 5000
999 CONTINUE
      WRITE(6,6999)ETAMX1,ETAMX2,ETA3,ETA4
6999 FORMAT(1H0,31HERROR IN STREAMLINE CALCULATION/11H ETAMAX1 = ,E12.5,
11H ETAMAX2 = ,E12.5,8H ETA3 = ,E12.5,8H ETA4 = ,E12.5)
      STOP
5000 CONTINUE
      RETURN
      END
SIBFTC WALLT1 DECK
      SUBROUTINE WALLT1(XS,TWS)
      CALCULATES TW(X)
      COMMON/Q/A(11600),IA(10)
      EQUIVALENCE
      1(A( 7737),XSTAR ), (A(10737),TW ), (IA( 4),II ),
      2(IA( 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

```

```

24190872
24190873
24190874
24190875
24190876
24190877
24190878
24190879
24190880
24190881
24190882
24190883
24190884
24190885
24190886
24190887
24190888
24190889
24190890
24190891
24190892
24190893
24190894
24190895
24190896
24190897
24190898
24190899
24190900
24190901
24190902
24190903
24190904
24190905
24190906
24190907
24190908
24190909
24190910
24190911
24190912
24190913

```

```

$IBFTCTWALLT2 DECK
SUBROUTINE WALLT2(MT,NT,XSIT,ETAT)
C
  CALCULATES TW(XSI,ETA)
  COMMON/Q/A(11600),IA(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)
  NX5=1
  NYS=1
  IF(ITC .LT. 2) GO TO 20
  DO 10 I=1,II
    TW(I)=DBTP(ETAT,XSIT,TWALL,ETAX(I),XSIX(I),NT,MT,NYS,NXS)
  10 CONTINUE
  GO TO 30
  20 CONTINUE
  WRITE(6,6010)
  6010 FORMAT(23H0ERROR IN WALLT2. ITC =12)
  30 CONTINUE
  RETURN
END

$IBFTC BLKDTA DECK
BLOCK DATA
COMMON/BLKDAT/PRESST(7),ZIT(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(1),I=1,7)/
1-4.,-3.,-2.,-1.,0.,1.,2./
DATA(ZIT(I,J),I=1,7),J=1,26)/
17*111., 7*223., 7*333., 7*444., 7*555., 7*833.,
27*1110., 7*1387., 7*1665., 7*1944., 7*2222., 7*2500.,
37*2778., 7*3056., 7*3333., 7*3889., 7*4444., 7*5000.,
47*5556., 7*6667., 7*7778., 7*8889., 7*10000., 7*11111.,
57*13889., 7*16667./
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)/
1 30.04, 28.73, 28.02, 28.02, 28.02, 28.02, 28.02, 28.02,

```

```

24190914
24190915
24190916
24190917
24190918
24190919
24190920
24190921
24190922
24190923
24190924
24190925
24190926
24190927
24190928
24190929
24190930
24190931
24190932
24190933
24190934
24190935
24190936
24190937
24190938
24190939
24190940
24190941
24190942
24190943
24190944
24190945
24190946
24190947
24190948
24190949
24190950
24190951
24190952
24190953
24190954

```

2	40.10,	36.57,	34.26,	33.05,	32.84,	32.83,	32.83,	24190955
3	53.61,	46.59,	42.07,	40.05,	38.55,	37.54,	37.54,	24190956
4	69.12,	59.63,	52.12,	47.60,	45.08,	43.27,	43.25,	24190957
5	80.09,	72.11,	63.13,	56.60,	52.09,	49.57,	47.48,	24190958
6	88.06,	82.58,	75.10,	66.11,	60.09,	56.37,	54.47,	24190959
7	99.12,	96.60,	93.09,	86.11,	77.12,	70.12,	67.60,	24190960
8	116.16,	111.13,	105.09,	101.09,	93.60,	86.12,	81.10,	24190961
9	138.27,	129.19,	118.17,	114.13,	108.12,	102.12,	94.61,	24190962
DATA ((AYRTOT(I,J),I=1,7),J=19,26)/								24190963
1	168.43,	150.34,	137.25,	128.19,	121.17,	115.13,	107.14,	24190964
2	248.47,	212.44,	184.40,	164.31,	152.24,	139.20,	132.16,	24190965
3	323.51,	282.49,	247.45,	213.41,	190.31,	170.27,	157.22,	24190966
4	394.45,	350.49,	310.45,	270.45,	233.40,	207.31,	187.27,	24190967
5	450.27,	410.45,	366.49,	325.48,	282.43,	245.38,	220.32,	24190968
6	480.26,	460.28,	420.36,	378.44,	330.44,	287.42,	255.36,	24190969
7	545.63,	530.40,	510.33,	490.33,	440.45,	393.49,	347.45,	24190970
8	670.77,	610.49,	575.40,	555.40,	530.55,	490.59,	437.55,	24190971
DATA((OMEGAT(I,J),I=1,7),J=1,10)/								24190972
1	714.47,	713.68,	712.46,	711.54,	710.80,	709.34,	708.316,	24190973
2	7560,	7560,	7560,	7560,	7560,	7560,	7560,	24190974
DATA((OMEGAT(I,J),I=1,7),J=11,20)/								24190975
1	6030,	6102,	6138,	6138,	6138,	6138,	5580,	24190976
2	5760,	5832,	5832,	5832,	5832,	5202,	5400,	24190977
3	544,	5544,	5544,	4896,	4986,	5094,	5256,	24190978
4	310,	4644,	4734,	4824,	4914,	5004,	5058,	24190979
5	320,	4392,	4482,	4590,	4680,	4680,	4014,	24190980
6	158,	4230,	4338,	4410,	3726,	3780,	3906,	24190981
7	068,	4140,	3546,	3600,	3654,	3708,	3906,	24190982
8	186,	3276,	3330,	3384,	3438,	3510,	3906,	24190983
DATA((OMEGAT(I,J),I=1,7),J=21,26)/								24190984
1	934,	3024,	3078,	3132,	3204,	3276,	2736,	24190985
2	880,	2952,	3006,	3060,	3132,	2574,	2718,	24190986
3	844,	2916,	2970,	2448,	2520,	2592,	2718,	24190987
4	844,	2214,	2286,	2358,	2430,	2484,	2592,	24190988
5	142,	2214,	2268,	2304,	2376,	2430,	2070,	24190989
END								24190990
\$IBFTC DERV DECK								24190991
SUBROUTINE DERV(V,W,X,Y,Z,DBYDX,DX,ITYPE)								24190992
GO TO (10,20,30,40,50),ITYPE								24190993
10 DBYDX=(-V+W)/DX								24190994
GO TO 60								24190995
								24190996

```

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,PSL,ACCG,RADIAN,EPS
CALL SOMEGA(AYESL,POPSL,ZTESL,OMESL)
AYEAV=.5*(AYEW+AYESL)
PR=PSL*POPSL
ROMUT=PR*OMEGS/R
ROMUE=PR*OMEGE/R
AYEMC=AYEAV+.2*(H-AYESL)*SIGR**SQRT(ROMUT/ROMUE)
CALL SOMEGA(AYEMC,POPSL,ZTMC,OMMC)
SIGC=ZTMC/ZTESL
C
CALL SOMEGA(AYEAV,POPSL,ZTMO,OMMO)
SIGO=ZTMO/ZTESL
C
FSC=(SIGC-.294)*SIGR**355/.4018
FSO=(SIGO-.294)*SIGR**355/.4018
FKC=(2.*SIGC)**EXPK
FKO=(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.+GAMC)/(1.+GAMO))**.4
RETURN
END
$IBFTC IWALL DECK
SUBROUTINE IWALL(TW,POPSL,AYEW)
1105 CONTINUE
PPLOG=ALOG10(POPSL)
TWK=TW/1.8

```

```

24190997
24190998
24190999
24191000
24191001
24191002
24191003
24191004
24191005
24191006
24191007
24191008
24191009
24191010
24191011
24191012
24191013
24191014
24191015
24191016
24191017
24191018
24191019
24191020
24191021
24191022
24191023
24191024
24191025
24191026
24191027
24191028
24191029
24191030
24191031
24191032
24191033
24191034
24191035
24191036
24191037

```

```

1110 IF(TWK-300.)1110,1110,1115
      CONTINUE
      AYEW=.432*TWK
      GO TO 1130
1115 CONTINUE
1120 IF(TWK-1800.)1120,1120,1125
      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
      AYEWHL=AYEE/H+SQRT(SIGR)*(1.-AYEE/H)
      FSS=(SIGS-.294)*SIGR*.355*AYEWHL/.4018
      CPES=(ZTE/ZTS)/(AYEE/H)
      IF(BETAS)10,20,20
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.)/((H/AYER)*ZTR+200.)
      AYEEC=AYEE/25031.6/33.86
      PPLOG=ALOG10(PSTPSL)

```

```

24191038
24191039
24191040
24191041
24191042
24191043
24191044
24191045
24191046
24191047
24191048
24191049
24191050
24191051
24191052
24191053
24191054
24191055
24191056
24191057
24191058
24191059
24191060
24191061
24191062
24191063
24191064
24191065
24191066
24191067
24191068
24191069
24191070
24191071
24191072
24191073
24191074
24191075
24191076
24191077
24191078
24191079

```

```

CALL TABLE7(AYEEC,PPLOG,AYEDIE)
IF(AYEDIE)10,20,20
10 CONTINUE
AYEDIE=0.
20 CONTINUE
XL=1.+0.19*AYEDIE*AYEE/H
RETURN
END
$IBFTC RORMUR DECK
SUBROUTINE RORMUR(P,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,AYER,SIGR)
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
PPLOG=ALOG10(P)
OMSE=OMEGS/OMEGE
XKS=OMSE*(1.005/(.005+OMSE**7))**(.1./14.)
XKEXP=.8125+.1875*EXP(-XKS*OMEGE/OMEGW)
XK=XKS**XKEXP
F1=XK*OMEGE/OMEGW
OMRW=F1**0.875*(1.2/(.2+F1**5))**0.1
PR=P*PSL
OMGR=OMRW*OMEGW
ROMUR=OMGR/R*PR
OMRK=1.8*OMGR
IF(OMRK-3.585E-10)30,10,10
10 CONTINUE
IF(OMRK-8.03E-10)20,20,30
20 CONTINUE
ZTRK=23213.13E-20/OMRK**2*(1.+SQRT(1.-.478656E18*OMRK**2))**2
GO TO 40
30 CONTINUE
CALL TABL19(OMRK,PPLOG,ZTRK)
40 CONTINUE
CALL TABL20(ZTRK,PPLOG,AYERTO)
AYER=847559.8*AYERTO
ZTR=ZTRK*1.8
CALL TABL14(ZTR,SIGR)
RETURN
END
$IBFTC SOMEGA DECK
C ROUTINE TO OBTAIN OMEGA AND ZT
C ENTHALPY IN FT**2/SEC**2 AND PRESSURE IN ATMOSPHERES
SUBROUTINE SOMEGA(ENTH,PRESS,ZT,OMEGA)

```

```

24191080
24191081
24191082
24191083
24191084
24191085
24191086
24191087
24191088
24191089
24191090
24191091
24191092
24191093
24191094
24191095
24191096
24191097
24191098
24191099
24191100
24191101
24191102
24191103
24191104
24191105
24191106
24191107
24191108
24191109
24191110
24191111
24191112
24191113
24191114
24191115
24191116
24191117
24191118
24191119
24191120

```

```

ENTHC=ENTH/25031.3
PPLOG=.434294*ALOG(PRESS)
20 IF(ENTHC-180.)21,21,22
21 ZT=ENTHC/.432
GO TO 40
22 ENTH=ENTHC/33.86
CALL TABLE2(ENTHC,PPLOG,ZT,CHECK)
IF(CHECK-1.)30,40,30
30 CONTINUE
WRITE(6,6000) PPLOG,ENTHC,ZT
6000 FORMAT(1H0,23HERROR IN TABLE2 LOOK-UP,5X,3E12.5)
40 IF(ZT-111.11)41,41,42
41 OMEGA=14.454E-10
GO TO 60
42 IF(ZT-2000.)43,43,44
43 OMEGA=SQRT(1.8*ZT/200.)*(400./(1.8*ZT+200.))*14.454E-10
GO TO 60
44 CALL TABL18(ZT,PPLOG,OMEGA,CHECK)
IF(CHECK-1.)50,60,50
50 WRITE(6,6100) PPLOG,ZT,OMEGA
6100 FORMAT(1H0,23HERROR IN TABL18 LOOK-UP,5X,3E12.5)
60 ZT=ZT*1.8
OMEGA=OMEGA/1.8
70 RETURN
END
$IBFTC TABLE2 DECK
C ZT=F(1/RT;.7)
SUBROUTINE TABLE2(AYES,PEDGEL,TS,CHECK)
COMMON/BLKDAT/PRESST(7),ZIT(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))

```

```

24191121
24191122
24191123
24191124
24191125
24191126
24191127
24191128
24191129
24191130
24191131
24191132
24191133
24191134
24191135
24191136
24191137
24191138
24191139
24191140
24191141
24191142
24191143
24191144
24191145
24191146
24191147
24191148
24191149
24191150
24191151
24191152
24191153
24191154
24191155
24191156
24191157
24191158
24191159
24191160
24191161
24191162

```

```

L(10)=LOC(ZTT(1,1))
L(11)=7
L(12)=7
L(13)=26
KS2=2
10 L(8)=0
TS=DTAB(AYES,PEDGEL,L(1))
CHECK=L(8)
RETURN
END
$IBFTC TABLE6 DECK
C
TABLE6 I=F(T,P)
SUBROUTINE TABLE6(TW,PEDGEL,AYEW)
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=1,19),I=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.,3680.,3805.,
10.,240.,450.,700.,870.,990.,1105.,1225.,1355.,1494.,
21647.,1822.,2023.,2250.,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)1,1,10
1 CONTINUE
L(1)=LOC(L(1))

```

```

24191163
24191164
24191165
24191166
24191167
24191168
24191169
24191170
24191171
24191172
24191173
24191174
24191175
24191176
24191177
24191178
24191179
24191180
24191181
24191182
24191183
24191184
24191185
24191186
24191187
24191188
24191189
24191190
24191191
24191192
24191193
24191194
24191195
24191196
24191197
24191198
24191199
24191200
24191201
24191202
24191203

```

[illegible]

```
DATA(AYDAYE(I,26),I=1,7)/
1.748, .723, .704, .686, .664, .646, .618, .559/
IF(KS7-1)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(AYDAYE(1,1))
L(11)=7
L(12)=7
L(13)=26
KS7=2
10 L(8)=0
AYEAYE AB(AYEE,PEDGEL,L(1))
RETURN
END
$IBFIC TABL14 DECK
C TABL14 SIG=F(ZT)
SUBROUTINE TABL14(ZTW,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, 24191278
2 .740, .758, .770, .776, .777, .774, .770, .761, .750, .742, .733, 24191279
3 .724, .715, .708, .701, .695, .690, .687, .685, .684, .685, .689, 24191280
4 .694, .704 /
IF(KS14-1)1,1,10
1 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(ZTT(1))
```

24191246
24191247
24191248
24191249
24191250
24191251
24191252
24191253
24191254
24191255
24191256
24191257
24191258
24191259
24191260
24191261
24191262
24191263
24191264
24191265
24191266
24191267
24191268
24191269
24191270
24191271
24191272
24191273
24191274
24191275
24191276
24191277
24191278
24191279
24191280
24191281
24191282
24191283
24191284
24191285
24191286

24191287
24191288
24191289
24191290
24191291
24191292
24191293
24191294
24191295
24191296
24191297
24191298
24191299
24191300
24191301
24191302
24191303
24191304
24191305
24191306
24191307
24191308
24191309
24191310
24191311
24191312
24191313
24191314
24191315
24191316
24191317
24191318
24191319
24191320
24191321
24191322
24191323
24191324
24191325
24191326
24191327
24191328

```

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/PRESST(7),ZTT(7,26),AYRTOT(7,26),OMEGAT(7,26)
   COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
   DIMENSION L(13)
   DATA KS18/1/
   IF(KS18-1)1,1,10
1 CONTINUE
   L(1)=LOC(L(1))
   L(2)=LOC(PRESST(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(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/PRESST(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(PRESST(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
10 L(8)=0
OMEGA = OMEGA/1.E-10
ZT=DTAB(OMEGA,PRESSL,L(1))
RETURN
END
$IBFCTITABL20 DECK
C
TABL20 1/RT=F(ZT,P)
SUBROUTINE TABL20(ZT,PRESSL,AYERTO)
COMMON/BLKDAT/PRESST(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(PRESST(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

```

```

24191329
24191330
24191331
24191332
24191333
24191334
24191335
24191336
24191337
24191338
24191339
24191340
24191341
24191342
24191343
24191344
24191345
24191346
24191347
24191348
24191349
24191350
24191351
24191352
24191353
24191354
24191355
24191356
24191357
24191358
24191359
24191360
24191361
24191362
24191363
24191364
24191365
24191366
24191367
24191368
24191369

```

```

L(13)=26
KS20=2
10 L(8)=0
  AYERTO=DTAB(ZT,PRESSL,L(1))
  RETURN
END
$IBFTC DELTA DECK
SUBROUTINE DELTA
  CALCULATES INPUTS TO STREAM FOR DELTA WING
  RESTRICTIONS
    60 GE SWEEP LE 80
    6 GE MACH LE 22
    ANGLE OF ATTACK GT 0.
  COMMON/Q/A(11600),IA(10)
  EQUIVALENCE
    1(A( 1),ACH ), (A( 2),ALPHA ), (A( 7),DELETE),
    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),SWEEP ),
    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),TH8MX ), (A( 237),THETA ),
    8(A( 2737),PRESSR), (IA( 1),M ), (IA( 2),N )
  COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
  DIMENSION XSI(50),ETAMAX(50),TH8MX(50),ETA(50),THETA(50,50),
  IPRESSR(50,50),PHIT1(25),XN1(25),SWEEP(50),XNT2(50,50),PHIT2(50)
  DATA NT1,(PHIT1(I),XN1(I),I=1,21)/ 21,
  1.10, .095, .15, .122, .20, .143, .25, .183, .30, .222,
  2.35, .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,
  51.2, .398 /
  DATA NT1,NTJ,(PHIT2(J),J=1,50)/ 5,50,
  11.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2,
  22.3, 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, 4.8, 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 SWEPT(1),(XNT2(1,J),J=1,50)/ 60,,
  1.398, .390, .395, .421, .480, .550, .645, .760, 1.00, 41*1.00 /
  DATA SWEPT(2),(XNT2(2,J),J=1,50)/ 65,,
  1.398, .387, .377, .380, .400, .426, .460, .498, .542, .597, .663,

```

```

24191370
24191371
24191372
24191373
24191374
24191375
24191376
24191377
24191378
24191379
24191380
24191381
24191382
24191383
24191384
24191385
24191386
24191387
24191388
24191389
24191390
24191391
24191392
24191393
24191394
24191395
24191396
24191397
24191398
24191399
24191400
24191401
24191402
24191403
24191404
24191405
24191406
24191407
24191408
24191409
24191410
24191411

```

```

2.755, .900, 1.00, 36*1.00 /
DATA SWEEP(3), (XNT2(3,J), J=1,501/ 70.,
1.398, .387, .377, .370, .372, .390, .398, .421, .445, .470, .498,
2.525, .577, .587, .620, .664, .715, .780, .860, 1.00, 30*1.00 /
DATA SWEEP(4), (XNT2(4,J), J=1,501/ 75.,
1.398, .387, .377, .370, .366, .372, .378, .385, .394, .405, .418,
2.432, .443, .460, .478, .495, .510, .528, .545, .565, .585, .610,
3.635, .660, .690, .720, .755, .790, .835, .880, .930, 19*1.00 /
DATA SWEEP(5), (XNT2(5,J), J=1,501/ 80.,
1.398, .387, .377, .370, .365, .362, .364, .366, .372, .375, .380,
2.384, .392, .398, .403, .412, .421, .430, .438, .443, .453, .461,
3.470, .479, .487, .497, .505, .517, .524, .536, .546, .560, .570,
4.583, .598, .612, .627, .642, .660, .677, .694, .710, .730, .750,
5.768, .805, .850, .898, .945, 1.00/
      SHARP DELTA WING

DEL=ALPHA/57.2958
CPSD=0.
IF(DEL.GT. 0.) CPSD=1.05+SORT(1.1025+.4./ACH**2/SIN(DEL)**2)-
11.278*SIN(DEL)**2/ACH**0.6
P2=PI*10.7*ACH**2*CPSD*SIN(DEL)**2+1.)
U=1.-.5863*DEL**2
URATIO=U
P=P2/PSL
AIE=M-(U*VEL)**2/2.
CALL SOMEGA(AIE,P,ZTE,OMEGA)
RHO2=P2/1716./ZTE
RR2=RHO/RHO2
ZETA=SQRT((DEL+1.22*RR2*(TAN(DEL)/TAN(SWEEP/RADIAN)))
1**566)**2+ASIN(1./ACH)**2)
ACHN2=(ACH*SIN(ZETA))**2
PHI=(1./190.-SWEEP))*ATAN(SQRT(((1.+5./ACH)**2*(ACHN2-1.)/ACHN2)/
1U**2-1.)/6.))*RADIAN
GAM=PHI**4/(PHI**4+.2)
THET=(90.-SWEEP)*(1.+PHI)*GAM
IF(PHI-1.02)10,10,20
10 CONTINUE
NN=1
XNCL=TB*LP(PHI*11,XN1,PHI*NT1,NN)
GO TO 30
20 CONTINUE

```

C C

```

NX=1
NY=1
XNCL=DBTP(PHIT2,SWEPT,XNT2,PHI,SWEEP,NTJ,NTI,NX,NY)
30 CONTINUE
XNCL=(1.+PHI)*XNCL/((ACHN2+1.)/ACHN2)
DELN=(90.-SWEEP)*XNCL
THDN=THET-DELN
TAND=TAN((90.-SWEEP)/RADIAN)
I=1
XSI(I)=XSI1
40 CONTINUE
J=1
ETAMAX(I)=XSI(I)*TAND
TH8MX(I)=DELN+THDN
ETA(J)=ETA1
50 CONTINUE
AOAM=TAN(ETA(J)/XSI(I))/TAND
THETA(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.ETAMAX(I)+DELETA) GO TO 50
I=I+1
XSI(I)=XSI(I-1)+DELXSI
60 CONTINUE
IF(XSI(I).GT.XSIF+DELXSI) GO TO 70
GO TO 40
70 CONTINUE
M=I-1
N=J-1
100 CONTINUE
RETURN
END
$1BFTC CYLIND DECK
SUBROUTINE CYLIND
C
CALCULATES INPUTS TO STREAM FOR SWEEP CYLINDER
COMMON/Q/A(11600),IA(10)
COMMON/BLKCON/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),SWEEP ) , (A( 30),URATIO) , (A( 31),VEL ) ,

```

24191453
24191454
24191455
24191456
24191457
24191458
24191459
24191460
24191461
24191462
24191463
24191464
24191465
24191466
24191467
24191468
24191469
24191470
24191471
24191472
24191473
24191474
24191475
24191476
24191477
24191478
24191479
24191480
24191481
24191482
24191483
24191484
24191485
24191486
24191487
24191488
24191489
24191490
24191491
24191492
24191493
24191494

```

4(A( 37),XSI ) , (A( 87),ETA ) , (A( 137),ETAMAX),
5(A( 187),TH8MX ) , (A( 237),THETA ) , (A( 5237),TWALL ) ,
6(A( 8487),PRESRX ) , (A( 9987),UE ) , (A(10737),TW ) ,
7(A(11587),P ) , (A( 2737),PRESSR ) , (UE ,V )
8,(IA( 1),M ) , (IA( 2),N ) , (IA( 4),I )
C
DIMENSION XSI(50),ETA(50),ETAMAX(50),TH8MX(50),THETA(50,50),
1PRESSR(50,50),UE(750),V(750),PRESRX(750),TW(750),TWALL(50,50)
DIMENSION TH8T(50),PPT2T(50)
DIMENSION P(5)
DIMENSION XSTAR(750),TWX(750)
C
C
DATA NT,(TH8T(1),PPT2T(1),I=1,44)/ 44,
1 0.,1.00000, .2, .99998782, .4, .99995126, .6, .99989034,
1 .8, .99980, 1.0, .99970, 1.2, .99956, 1.4, .99940,
1 1.6, .99922, 1.8, .99902, 2.0, .99878, 3.0, .99726,
2 4., .99512, 5., .99239, 6., .98907, 7., .98516,
3 8., .98063, 9., .97553, 10., .96985, 11., .963597,
4 12., .95677, 13., .949397,14., .94148, 15., .93302,
5 16., .92402, 17., .91451,18., .90452, 19., .89401,
1 20.,.887, 24.,.840, 28.,.795, 32.,.732, 36.,
2.670, 40.,.608, 44.,.546, 48.,.485, 52.,.426, 56.,.370, 60.,.317,
365.,.258, 70.,.207, 80.,.124, 90.,.0775, 100.,.05/
C
NXS=1
NXSV=1
XSV=DX
USIN=VEL*SIN(SWEEP/RADIAN)
ACHN=ACH*COS(SWEEP/RADIAN)
DETA=.01*RADIUS
PT2=(1.2*ACHN**2)**3.5*(6./((7.*ACHN**2-1.))**2.5*PINF
XSTAR(1)=0.
DO 20 I=1,750
THETR=RADIAN*XSTAR(I)/RADIUS
IF(THETR-100.)10,30,30
10 CONTINUE
POPT2=T6LP(TH8T,PPT2I,THETR,NT,NXS)
PRESRX(I)=POPT2*PT2/(POPSL*PSL)
TWX(I)=T6LP(ETA,TW,XSTAR(I),N,NXSV)
XSTAR(I+1)=XSTAR(I)+DETA
24191495
24191496
24191497
24191498
24191499
24191500
24191501
24191502
24191503
24191504
24191505
24191506
24191507
24191508
24191509
24191510
24191511
24191512
24191513
24191514
24191515
24191516
24191517
24191518
24191519
24191520
24191521
24191522
24191523
24191524
24191525
24191526
24191527
24191528
24191529
24191530
24191531
24191532
24191533
24191534
24191535

```

```

II=I
20 CONTINUE
WRITE(6,6100)THETR,ETA(II)
6100 FORMAT(1H0,67HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU2
1BROUTINE CYLIND /15H THETAR-ETA(II) ,5X,2E12.5//26H RUN TERMINATED
2 BY PROGRAM )
STOP
30 CONTINUE
II=II+1
XSTAR(II)=XSTAR(II-1)+DETA
PRESRX(II)=PRESRX(II-1)
TWX(II)=TWX(II-1)
DX=DETA
URATIO=0.
P(1)=1.0
P(2)=.99998782
P(3)=.99995126
DXP=.00349
CALL FLOW
M=2
N=II/4
J=1
DO 35 I=1,N
PRESRX(I)=PRESRX(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
PRESRX(I,J)=PRESRX(J)
50 CONTINUE
60 CONTINUE
XSI(1)=0.
XSI(2)=1.E+6

```

```

ETAMAX(1)=ETA(N)
ETAMAX(2)=ETA(N)
TH8MX(1)=THETA(1,N)
TH8MX(2)=THETA(1,N)
DX=XSV
URATIO= SIN(SWEEP/57.2958)
RETURN
END
$IBFTC FLOW DECK
SUBROUTINE FLOW
FLOW CALCULATES THE EXTERNAL FLOW AND STAGNATION
PROPERTIES
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),POPSL ) , (A( 30),URATIO) , (A( 31),VEL ) ,
3(A( 27),SWEEP ) , (A( 237),POPSLX) , (A( 2737),AIEE ) ,
4(A( 3487),OMEGAE) , (A( 4237),ZTEX ) , (A( 7737),XSTAR ) ,
5(A( 8487),PRESRX) , (A( 9987),UE ) , (A(11587),P ) ,
5(IA( 4),II ) , (IA( 8),NPR )
DIMENSION PRESRX(750),POPSLX(750),UE(750),AIEE(750),
1ZTEX(750),OMEGAE(750),XSTAR(750)
DIMENSION P(5)
C
C
C
C
DO 780 J=1,II
POPSLX(J)=POPSL*PRESRX(J)
780 CONTINUE
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/57.2958)
AIEE(1)= H-.5*U**2
CALL SOMEGA(AIEE(1),POPSLX(1),ZTEX(1),OMEGAE(1))
24191578
24191579
24191580
24191581
24191582
24191583
24191584
24191585
24191586
24191587
24191588
24191589
24191590
24191591
24191592
24191593
24191594
24191595
24191596
24191597
24191598
24191599
24191600
24191601
24191602
24191603
24191604
24191605
24191606
24191607
24191608
24191609
24191610
24191611
24191612
24191613
24191614
24191615
24191616
24191617
24191618

```

```

      ZTE=ZTEX(1)
      IF(URATIO)805,805,810
805  CONTINUE
      SECDRV=(P(1)-2.*P(2)+P(3))/DXP**2
      UEOUI=DX/VEL*SQRT(-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 SOMEGA(AIEE(2),POPSLX(2),ZTEX(2),OMEGAE(2))
      ZTE=ZTEX(2)
      KK=KK+1
810  CONTINUE
      DO 865 I=KK,II
      IF(I-3)815,820,825
815  CONTINUE
      CALL DERV(PRESRX(I-1),PRESRX(I),PRESRX(I+1),PRESRX(I+2),
1PRESRX(I+3),DBYDX1,DX,1)
      CALL DERV(PRESRX(I-1),PRESRX(I),PRESRX(I+1),PRESRX(I+2),
1PRESRX(I+3),DBYDX2,DX,2)
      GO TO 845
820  CONTINUE
      CALL DERV(PRESRX(I-2),PRESRX(I-1),PRESRX(I),PRESRX(I+1),
1PRESRX(I+2),DBYDX1,DX,2)
      CALL DERV(PRESRX(I-2),PRESRX(I-1),PRESRX(I+1),PRESRX(I+2),
1DBYDX2,DX,3)
      GO TO 845
825  CONTINUE
      IF(I-11+1)830,835,840
830  CONTINUE
      CALL DERV(PRESRX(I-3),PRESRX(I-2),PRESRX(I),PRESRX(I+1),
1DBYDX1,DX,3)
      CALL DERV(PRESRX(I-2),PRESRX(I-1),PRESRX(I+1),PRESRX(I+2),
1DBYDX2,DX,3)
      GO TO 845
835  CONTINUE
      CALL DERV(PRESRX(I-3),PRESRX(I-2),PRESRX(I),PRESRX(I+1),
1DBYDX1,DX,3)
      CALL DERV(PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),PRESRX(I),
1PRESRX(I+1),DBYDX2,DX,4)
      GO TO 845

```

```

24191619
24191620
24191621
24191622
24191623
24191624
24191625
24191626
24191627
24191628
24191629
24191630
24191631
24191632
24191633
24191634
24191635
24191636
24191637
24191638
24191639
24191640
24191641
24191642
24191643
24191644
24191645
24191646
24191647
24191648
24191649
24191650
24191651
24191652
24191653
24191654
24191655
24191656
24191657
24191658
24191659
24191660

```

```

840 CONTINUE
CALL DERV(PRESRX(I-4),PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),
1PRESRX(I),DBYDX1,DX,4)
CALL DERV(PRESRX(I-4),PRESRX(I-3),PRESRX(I-2),PRESRX(I-1),
1PRESRX(I),DBYDX2,DX,5)
845 CONTINUE
DBYDX=(DBYDX1+DBYDX2)/2.
JJ = 1
DO 855 NO=1,10
UEINF2=UEINF1
UEINF1 = UEINF-R*ZTE*DBYDX*DX/(VEL**2*((PRESRX(I-1)+PRESRX(I))/2.))
1)
AIEE(I)=H-UEINF1*VEL**2
IF(NPR.EQ.6)AIEE(I)=AIEE(I)-.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(1H0,62HCONVERGENCE NOT ESTABLISHED IN VELOCITY CALCULATION.
1 UEINF1 = ,E12.5,10H UEINF2 = ,E12.5)
860 CONTINUE
UEINF=UEINF1
UE(I)=VEL*SQRT(2.*UEINF)
ZTEX(I)=ZTE
OMEGAE(I)=OMEGA
CONTINUE
IF(NPR.EQ.6) NPR=100
C
RETURN
END
$1BFTC AXI2D DECK
SUBROUTINE AXI2D
C CALCULATES INPUTS TO QTRAN FOR TWO-DIMENSIONAL AND
C AXISYMMETRIC BODIES
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
24191661
24191662
24191663
24191664
24191665
24191666
24191667
24191668
24191669
24191670
24191671
24191672
24191673
24191674
24191675
24191676
24191677
24191678
24191679
24191680
24191681
24191682
24191683
24191684
24191685
24191686
24191687
24191688
24191689
24191690
24191691
24191692
24191693
24191694
24191695
24191696
24191697
24191698
24191699
24191700
24191701

```

```

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( 2737),AIEE ) , (A( 3487),OMEGAE) , (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),N ) , (IA( 4),I ) ,
1(IA( 8),NPR )
DIMENSION XSI(50),Y(50),XSTAR(750),POPSLX(750),UE(750),AIEE(750),
1ZTEX(750),OMEGAE(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)

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(I)=0.5*(XSI(1)+XSI(I-1))
YS(I)=0.5*(Y(1)+Y(I-1))
XS(I)=XS(I-1)+SQRT((YS(I)-YS(I-1))**2+(XSI(I)-XSI(I-1))**2)
DELS(I)=ATAN((Y(I)-Y(I-1))/(XSI(I)-XSI(I-1)))
IF(XSI(I) .EQ. XSI(I-1)) DELS(I)=1.5707963
DEL=DELS(I)
IF(DELS(I))10,10,20
10 CONTINUE
CPSD=0.

```

24191702
24191703
24191704
24191705
24191706
24191707
24191708
24191709
24191710
24191711
24191712
24191713
24191714
24191715
24191716
24191717
24191718
24191719
24191720
24191721
24191722
24191723
24191724
24191725
24191726
24191727
24191728
24191729
24191730
24191731
24191732
24191733
24191734
24191735
24191736
24191737
24191738
24191739
24191740
24191741
24191742
24191743

C

```

GO TO 30
20 CONTINUE
  CPD=1.05+SQRT(1.1025+4./ACH**2/SIN(DEL)**2)-1.278*SIN(DEL)**2/
    1ACH**0.6
30 CONTINUE
  P2=PINF*(0.7*ACH**2*CPD*SIN(DEL)**2+1.)
  PRESS(I)=(P2/PSL)
100 CONTINUE
  PRESS(1)=PRESS(3)-(PRESS(3)-PRESS(2))*(XSI(3)-XSI(1))/(XSI(3)-
    1XSI(2))
  XSI(M+1)=XSI(M)
  YS(M+1)=Y(M)
  XS(M+1)=XS(M)+SQRT((YS(M+1)-YS(M))**2+(XSI(M+1)-XSI(M))**2)
  PRESS(M+1)=PRESS(M)+(PRESS(M)-PRESS(M-1))*(XSI(M+1)-XSI(M))/
    1(XSI(M)-XSI(M-1))
  M=M+1
  IF(PRESS(M) .LT. PINF/PSL) PRESS(M)=PINF/PSL
  IF(PRESS(M) .GT. POPSL) PRESS(M)=POPSL
  IF(PRESS(1) .LT. PINF/PSL) PRESS(1)=PINF/PSL
  DO 200 I=1,749
    XSTAR(I+1)=XSTAR(I)+DX
    X(I)=XSTAR(I)
    XSIX(I)=TBLP(XS,XSI,XSTAR(I),M,NXSV)
    ETAX(I)=TBLP(XS,YS,XSTAR(I),M,NXSV)
    THETAX(I)=TBLP(XS,DELS,XSTAR(I),M,NXSV)
    THETAX(I)=57.2958*THETAX(I)
    IF(XSTAR(I+1)-XSI(M))170,210,210
170 CONTINUE
    POPSLX(I)=TBLP(XS,PRESS,XSTAR(I),M,NXSV)
    DEL=TBLP(XS,DELS,XSTAR(I),M,NXSV)
    IF(PRESS(1)-.528*POPSL)40,40,50
40 CONTINUE
    UE(I)=VEL*SQRT(1.-(POPSLX(I)/PINF*PSL-1.)/1.4/ACH**2)
    GO TO 60
50 CONTINUE
    IF(I .GT. 1) GO TO 58
    P=.528*POPSL
    DO 55 K=1,M
      J=M-K+1
      PREST(K)=PRESS(J)
      XST(K)=XS(J)

```

24191744
 24191745
 24191746
 24191747
 24191748
 24191749
 24191750
 24191751
 24191752
 24191753
 24191754
 24191755
 24191756
 24191757
 24191758
 24191759
 24191760
 24191761
 24191762
 24191763
 24191764
 24191765
 24191766
 24191767
 24191768
 24191769
 24191770
 24191771
 24191772
 24191773
 24191774
 24191775
 24191776
 24191777
 24191778
 24191779
 24191780
 24191781
 24191782
 24191783
 24191784

```

55 CONTINUE
   XSONIC=RBPLP(PREST,XST,P,M,1)
   SONIC=SQRT(H /3.)
58 CONTINUE
   IF(XSTAR(I)-XSONIC)70,70,80
70 CONTINUE
   UE(I)=SONIC/XSONIC*XSTAR(I)
   GO TO 90
80 CONTINUE
   UE(I)=SONIC*SQRT(6.*(1*-(POPSLX(I)/POPSL)**.285714))
90 CONTINUE
60 CONTINUE
   ATEE(I)=H-UE(I)**2/2.
   CALL SOMEGA(AIEE(I),POPSLX(I),ZTEX(I),OMEGAE(I))
   IF(MPR-2)150,150,140
140 CONTINUE
   DODI(I)=1.
   RORI(I)=1.
   GO TO 160
150 CONTINUE
   DODI(I)=TBLP(XS,YS,XSTAR(I),M,NXSV)
   RORI(I)=DODI(I)
160 CONTINUE
   II=1
200 CONTINUE
   THETAX(I)=THETAX(2)
   WRITE(6,6100)XSTAR(I),M,XS(M),DX
6100 FORMAT(1H0,66HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU24191812
18ROUTINE AXI2D /20H XSTAR(I)-M-XS(M)-DX ,5X,E12.5,13,2X,2E12.5)
   WRITE(6,6120)
6120 FORMAT(1H0,25HRUN TERMINATED BY PROGRAM)
   STOP
210 CONTINUE
   URATIO=UE(I)/VEL
C
   IF(MPR-2)224,224,221
224 CONTINUE
   WRITE(6,6150)
6150 FORMAT(1H0,19HAXISYMMETRIC 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,5HXI = ,E12.5,6H DX = ,E12.5//12X,3HXS1,9X,1HY/)
M=M-1
DO 222 I=1,M
WRITE(6,6300)XSI(I),Y(I)
6300 FORMAT(1H ,10X,12E11.3)
222 CONTINUE
M=M+1
RETURN
END
$18FTC HEMI DECK
SUBROUTINE HEMI
C
C CALCULATES INPUTS TO FLOW FOR HEMISPHERE
C
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),POPSL ) , (A( 22),RADIUS) ,
3(A( 34),XI ) , (A( 987),DODI ) , (A( 1737),RORI ) ,
4(A( 5237),X ) , (A( 5987),XSIX ) , (A( 6737),ETAX ) ,
5(A( 7737),XSTAR ) , (A( 9237),THETAX) , (A(11587),P ) ,
6(A( 30),URATIO) , (A( 8487),PRESRX) , (IA( 4),Ii )
DIMENSION XSTAR(750),PRESRX(750),DODI(750),RORI(750)
DIMENSION TH8T(50),PPT2T(50)
DIMENSION P(5)
DIMENSIONX(750),XSIX(750),ETAX(750),THETAX(750)
C
DATA NT,(TH8T(I),PPT2T(I),I=1,44)/ 44,
1 0.,1.00000, .2, .99998782, .4, .99995126, .6, .99989034,
1 .8, .99980, 1.0, .99970, 1.2, .99956, 1.4, .99940,
1 1.6, .99922, 1.8, .99902, 2.0, .99878, 3.0, .99726,
2 4., .99512, 5., .99239, 6., .98907, 7., .98516,
3 8., .98063, 9., .97553, 10., .96985, 11., .963597,
4 12., .95677, 13., .949397, 14., .94148, 15., .93302,
5 16., .92402, 17., .91451, 18., .90452, 19., .89401,
6 20., .88302, 24., .838, 28., .785, 32., .720,
7 36., .653, 40., .585, 44., .517, 48., .450,
8 52., .384, 56., .320, 60., .262, 65., .200,

```

```

C          9 70., .147,      80., .076,      90., .045,      100., .4/
          NXSV=1
          XI=0.
          P(1)=1.0
          P(2)=.99998782
          P(3)=.99995126
          DX=.00349
          DX=.00300*RADIUS
          XSTAR(1)=XI
          DO 100 I=1,750
            THET=RADIAN*XSTAR(I)/RADIUS
            X(I)=XSTAR(I)
            XSIX(I)=0.
            ETAX(I)=0.
            THETAX(I)=90.-THET
            IF(THETAX .LT. 0.) THETAX(I)=0.
            IF(THET-100.)50,50,110
          50 CONTINUE
            POPT2=TBLP(TH8T,PPT2T,THET,NT,NXSV)
            PRESRX(I)=POPT2
            RORI(I)=SIN(XSTAR(I)/RADIUS)
            DODI(I)=RORI(I)
            XSTAR(I+1)=XSTAR(I)+DX
            II=I
          100 CONTINUE
            WRITE(6,6100)THET,XSTAR(I),DX
          6100 FORMAT(1H0,65HSTREAMLINE COORDINATE DIMENSION (750) EXCEEDED IN SU
            1BROUTINE HEMI /15H THETA-XSTAR-DX ,5X,3E12.5)
            WRITE(6,6120)
          6120 FORMAT(1H0,25HRUN TERMINATED BY PROGRAM)
          STOP
          110 CONTINUE
            URATIO=0.
C
          RETURN
        END
$IBFTC QTRAN  DECK
SUBROUTINE QTRAN
C  QTRN CALCULATES HEAT TRANSFER DATA GIVEN THE
C  STREAMLINE FUNCTIONS
COMMON/Q/A(11600),IA(10)

```

```

COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
EQUIVALENCE
1(A( 4),AYEWO ) , (A( 5),BEQXL ) , (A( 6),BEQXT ) ,
2(A( 11),DX ) , (A( 15),H ) , (A( 16),HO ) ,
3(A( 17),HREFT ) , (A( 21),QINF ) , (A( 26),RTRANS) ,
4(A( 31),VEL ) , (A( 32),VELP ) , (A( 34),XI ) ,
5(A( 237),POPSLX) , (A( 987),DODI ) , (A( 1737),RORI ) ,
6(A( 2737),AIEE ) , (A( 3487),OMEGAE) , (A( 4237),ZTEX ) ,
7(A( 5237),X ) , (A( 5987),XSIX ) , (A( 6737),ETAX ) ,
8(A( 7737),XSTAR ) , (A( 9237),THETAS) , (A( 9987),UE ) ,
9(A(10737),TW ) , (A(11487),XPO ) , (A( 21),RADIUS) ,
1(IA( 3),KF ) , (IA( 4),II ) , (IA( 8),NPR ) ,
2(IA( 2730),AYEW ) , (A( 232),SL ) , (A( 980),ST ) ,
3(IA( 9),ITC ) ,
DIMENSION AYEW(750),SL(750),ST(750)
DIMENSION XSTAR(750),POPSLX(750),UE(750),AIEE(750),ZTEX(750),
OMEGAE(750),THETAS(750),DODI(750),RORI(750),TW(750),XPO(100)
DIMENSION X(750),XSIX(750),ETAX(750)
INTEGER GSAVE,AA

C
FINTRP(QOFL,Q1FL,Q2FL,Q3FL,Q4FL)=Q3FL+(Q4FL-Q3FL)/
1(Q1FL-QOFL)*(Q2FL-QOFL)

C
WRITE(6,6200)
6200 FORMAT(1H1,3X,1HX,11X,5HP/PSL,7X,2HUE,10X,4HDODI,8X,4HRORI,8X,2HTW,
1,10X,3HZTE,9X,6HOMEGAE,6X,4HAIEE,8X,4HAYEW/
24X,3HXSIX,9X,5HBEQXL,7X,5HXEQXL,7X,6HAYEAWL,6X,2HJL,10X,5HEBARL,7X,24191936
32HHL,10X,5HQLISO,7X,5HQDOTL,7X,4HCFEL/
44X,3HETA,9X,5HBEQXT,7X,5HXEQXT,7X,6HAYEAWT,6X,2HJN ,10X,5HEBART,7X,24191938
52HHT,10X,5HQTISO,7X,5HQDOTT,7X,4HCFET/
64X,5HTHETA,7X,3HZTR,9X,3HMUO,9X,3HFXQ,9X,3HFXS,9X,3HKKRQ,9X,3HKKRS,
79X,6HOMEGAR,6X,6HTHETAL,6X,6HTHETAT)
DO 100 K=1,KF
KK=K
IF(XPO(K) .GT. (XSTAR(II)-3.*DX)) GO TO 110
100 CONTINUE
GO TO 120
110 CONTINUE
KF=KF-1
120 CONTINUE
INITIALIZATION
C

```

24191910
24191911
24191912
24191913
24191914
24191915
24191916
24191917
24191918
24191919
24191920
24191921
24191922
24191923
24191924
24191925
24191926
24191927
24191928
24191929
24191930
24191931
24191932
24191933
24191934
24191935
24191936
24191937
24191938
24191939
24191940
24191941
24191942
24191943
24191944
24191945
24191946
24191947
24191948
24191949
24191950

```

900  CONTINUE
     NXSV=1
     MXSV=1
     I=0
     K=1
     ITRAN=0
     ITR=0
     IPRINT=0
     INTERP=0
     GSAVE=0
     COUNT=0
     PHIL=0
     PHIT=0
     BETAS=0.5
     CTR=0
     IP=XI/DX
     IP=IP+I
     XTRANS=0
C
C      POINT FOR REPEATED LOOPING AS X IS INCREMENTED BY DX
C
1001 CONTINUE
     I=I+1
1100 CONTINUE
     CALL IWALL(TW(I),POPSLX(I),AYEW(I))
     CALL SOMEGA(H,POPSLX(I),ZTS,OMEGAS)
     CALL SOMEGA(AYEW(I),POPSLX(I),ZTW,OMEGAW)
     CALL RORMUR(POPSLX(I),OMEGAS,OMEGA(I),OMEGAW,OMEGA,ROMUR,ZTR,
     IAYER,SIGR)
C
C      VELOCITY AND DELTA DERIVATIVE
1200 CONTINUE
     IF(I-2)1210,1220,1230
1210 CONTINUE
     CALL DERV(DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DODI(I+4),DDELDX,D
     IX,1)
     CALL DERV(UE(I),UE(I+1),UE(I+2),UE(I+3),UE(I+4),DUEDX,DX,1)
     GO TO 1270
1220 CONTINUE
     CALL DERV(DODI(I-1),DODI(I),DODI(I+1),DODI(I+2),DODI(I+3),DDELDX,D
     IX,2)
     CALL DERV(UE(I-1),UE(I),UE(I+1),UE(I+2),UE(I+3),DUEDX,DX,2)

```

```

24191951
24191952
24191953
24191954
24191955
24191956
24191957
24191958
24191959
24191960
24191961
24191962
24191963
24191964
24191965
24191966
24191967
24191968
24191969
24191970
24191971
24191972
24191973
24191974
24191975
24191976
24191977
24191978
24191979
24191980
24191981
24191982
24191983
24191984
24191985
24191986
24191987
24191988
24191989
24191990
24191991
24191992

```

```

1230 GO TO 1270
      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,3)
      CALL DERV(UE(I-2),UE(I-1),0.,UE(I+1),UE(I+2),DUEDX,DX,3)
      GO TO 1270
1250 CONTINUE
      CALL DERV(DODI(I-3),DODI(I-2),DODI(I-1),DODI(I),DODI(I+1),DDEL DX,4)
      CALL DERV(UE(I-3),UE(I-2),UE(I-1),UE(I),UE(I+1),DUEDX,DX,4)
      GO TO 1270
1260 CONTINUE
      CALL DERV(DODI(I-4),DODI(I-3),DODI(I-2),DODI(I-1),DODI(I),DDEL DX,5)
      CALL DERV(UE(I-4),UE(I-3),UE(I-2),UE(I-1),UE(I),DUEDX,DX,5)
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.
      AYESL=AIEE(I)
      GO TO 1370
1340 CONTINUE
      THETSL=(XN-1.)*THETAS(I)/RADIAN
      XK=2./3.*XN*(XN-1.)
      EXPK=-.194*EXP(-XK)
      GO TO 1360
1350 CONTINUE
      THETSL=(1.-1./XN)*THETAS(I)/RADIAN
      XK=2./3.*(XN-1.)
      EXPK=.194*EXP(-XK)
1360 CONTINUE
      VP=UE(I)*COS(THETSL)
      AYESL=H-.5*VP**2

```

24191993
24191994
24191995
24191996
24191997
24191998
24191999
24192000
24192001
24192002
24192003
24192004
24192005
24192006
24192007
24192008
24192009
24192010
24192011
24192012
24192013
24192014
24192015
24192016
24192017
24192018
24192019
24192020
24192021
24192022
24192023
24192024
24192025
24192026
24192027
24192028
24192029
24192030
24192031
24192032
24192033

```

1370 CONTINUE
C
      EQUIVALENT DISTANCE PARAMETERS
C
1500 CONTINUE
C
      CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,POPSLX(I),OMEGAE(I),OMEGAS,GAMC,
      1GAMO,EBARL,EBART)
      G=ROMUR*UE(I)
      FOFI=DODI(I)/RORI(I)
      GLAM=G*RORI(I)**2*FOFI**(2.*EBARL)
      GTURB=G*RORI(I)**1.25*FOFI**(1.25*EBART)
      IF(GSAVE-1)1510,1520,1530
1510 CONTINUE
      GSAVE=1
      SUML=GLAM*XI*BEQXL
      SUMT=GTURB*XI*BEQXT
      SL(I)=SUML
      ST(I)=SUMT
      SVEL=SUML
      GO TO 1540
1520 CONTINUE
      GSAVE=2
      GL2=GLAM
      GT2=GTURB
      GO TO 1001
1530 CONTINUE
      GSAVE=1
      SUML=SUML+DX/3.*(GL1+4.*GL2+GLAM)
      SUMT=SUMT+DX/3.*(GT1+4.*GT2+GTURB)
      SL(I)=SUML
      ST(I)=SUMT
      BEQXL=SUML/(GLAM*XSTAR(I))
      BEQXT=SUMT/(GTURB*XSTAR(I))
      CX=1.-SVEL/SUML
      FDX=71.*CX/(70.*CX+1.)
      BETAS=BETAS+FDX*(DBS-BETAS)
      SVEL=SUML
1540 CONTINUE
      IF(UE(I))1560,1550,1560
1550 CONTINUE
      DBS=2.*(H/AIEE(I))*BEQXL

```

```

1560 GO TO 1570
1560 CONTINUE
1570 DBS=2.*H/AIEE(I)*XSTAR(I)/UE(I)*DUEDX*BEQXL
1570 CONTINUE
1570 GLI=GLAM
1570 GTI=GTURB
1580 CONTINUE
C
C      BOUNDARY LAYER TRANSITION CHECK
IF(RTRANS)1600,1583,1583
1583 CONTINUE
IF(ITRAN)1585,1585,1600
1585 CONTINUE
CALL JAYELL(AYEW(I),H,AIEE(I),ZTEX(I),ZIS,SIGR,BETAS,
IPOPSLX(I),XJL)
CALL MUZERO(OMEGAR,ZTR,AYER,AIEE(I),H,POPSLX(I),XMUO,XL)
SEQXL=BEQXL/XJL**2
SEQXT=BEQXT/XJL**2
SEQL=SEQXL*XSTAR(I)
SEQT=SEQXT*XSTAR(I)
XEQXL=BEQXL/XJL**4
XEQXT=BEQXT/XJL**4
FXQ=CBRT(XJL**85*BEQXT/BEQXL)
RRQ=ROMUR*UE(I)*XEQXL/XMUO**2/FXQ**2
IF(RRQ-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)/XMUO**2
RLS=RUM*SEQL
RTS=RUM*SEQT
XMUE=OMEGAE(I)*ZTEX(I)
RTHL=XMUO/XMUE*.664*SQRT(RLS)
ADEM=ALOG10(RTS)-.407
IF(ADEM)1593,1593,1594
1593 RTHT=0.
CTR=0.
GO TO 1595
1594 RTHT=XMUO/XMUE*.2135*RTS/ADEM**2*.64
CTR=SUMT*(1.-(RTHL/RTHT)**1.25)

```

24192076
24192077
24192078
24192079
24192080
24192081
24192082
24192083
24192084
24192085
24192086
24192087
24192088
24192089
24192090
24192091
24192092
24192093
24192094
24192095
24192096
24192097
24192098
24192099
24192100
24192101
24192102
24192103
24192104
24192105
24192106
24192107
24192108
24192109
24192110
24192111
24192112
24192113
24192114
24192115
24192116

```

1595 BEQXT=BEQXT-CTR/GTURB/XSTAR(1)
XTRANS=XSTAR(1)
1600 ITR=1
      GO TO 1510
      CHECK FOR PRINTOUT VALUES
      CONTINUE
1610 IF(XSTAR(1)-XPO(K))1610,1620,1630
      CONTINUE
1620 IF(XSTAR(1)+2.*DX-XPO(K))1001,1001,1630
      CONTINUE
1630 IPRINT=1
      CONTINUE
      CALL JAYELL(AYEW(1),H,AIEE(1),ZTEX(1),ZTS,SIGR,BETAS,
1POPSLX(1),XJL)
      CALL MUZERO(OMEGAR,ZTR,AYER,AIEE(1),H,POPSLX(1),XMUO,XL)
      SEQXL=BEQXL/XJL**2
      XEQXL=REQXL/XJL**4
      SEQL=SEQXL*XSTAR(1)
      XEQL=XEQXL*XSTAR(1)
      SEQXT=BEQXT/XJL**5625
      SEQXT=SEQXT*XSTAR(1)
      BTBL=BEQXT/REQXL
      FXS=(XJL**4.25*BTBL)**(1./3.)
      FXQ=(XJL**4.85*BTBL)**(1./3.)
      RUM=ROMUR*UE(1)/XMUO**2
      RRS=RUM*SEQL/FXS**2
      RRO=RUM*XFQL/FXQ**2
      P=XJL**1.5
      Q=XJL**3
      FPM=FXS*P*XMUO
      TUEL=.932*FPM*SQRT(RRS)/SEQL
      TUET=FPM/SEQL*(.185*RRS/ALOG10(RRS+3000.))**2.584)
      PHOE=1.297*POPSLX(1)/ZTEX(1)
      XMUE=OMEGAE(1)*ZTEX(1)
      RLS=RUM*SEQL
      RTS=RUM*CFCT
      RTHL=XMUO/XMUE*.664*SQRT(RLS)
      IF(1.EQ.1) GO TO 1633
      ADFM=ALOG10(RTS)-.407
      IF(ADEM)1631,1631,1632
1631 RTHT=0.
      GO TO 1633

```

```

24192117
24192118
24192119
24192120
24192121
24192122
24192123
24192124
24192125
24192126
24192127
24192128
24192129
24192130
24192131
24192132
24192133
24192134
24192135
24192136
24192137
24192138
24192139
24192140
24192141
24192142
24192143
24192144
24192145
24192146
24192147
24192148
24192149
24192150
24192151
24192152
24192153
24192154
24192155
24192156
24192157
24192158

```

```

1632 RTHT=XMUO/XMUE*.2135*RTS/ADEM**2.64
1633 THETAL=XMUE/RHOE/UE(I)*RTHL
      THETAT=XMUE/RHOE/UE(I)*RTHT
      CFEL=2.*TUET/RHOE/UE(I)
      CFET=2.*TUET/RHOE/UE(I)
      HL=.332*Q/SIGR**.645*XMUO*FXQ*SQRT(HRQ)/XEQL
      HL=ACCG*XL*HL
      IF(ITRAN .GE. 1) HL=0.
      HT=XMUO*FXQ*Q/SIGR**.645/XEQL*(.185*RRQ/ALOG10(HRQ+
13000.))*2.584)
      HT=ACCG*XL*HT
      IF(ITRAN .LE. 0) HT=0.
      AYEAWL=H-(1.-SQRT(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+1))/2.
      ST(J)=(ST(J-1)+ST(J+1))/2.
1635 CONTINUE
      PHISL=0.
      PHIST=0.
      AA=0
      DO 1640 J=2,I
      SLBS=(SL(J)-(SL(J)-SL(J-1))/3.)/SL(I)
      STBS=(ST(J)-(ST(J)-ST(J-1))/3.)/ST(I)
      FBSL=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(1H0,5X,97H IN THE CALCULATION OF THE NON-ISOTHERMAL WALL PA24192190
      1RAMEETER STBS9 A NEGATIVE VALUE WAS ENCOUNTERED.//6X,23HAB5(STBS9)24192191
      2WAS ASSUMED. )
      AA=1
1639 FSBST= ABS(STBS9)**(-1./9.)
      PHISL=(AYEW(J)-AYEW(J-1))*FSBSL+PHISL
      PHIST=(AYEW(J)-AYEW(J-1))*FSBST+PHIST
1640 CONTINUE
      PHIL=AYEW(I)-AYEW(I)-PHISL
      PHIT=AYEW(I)-AYEW(I)-PHIST

```

```

1650 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/HREFT
IF(ITR)1680,1680,1670
1670 CONTINUE
ITR=0
GO TO 1600
1680 CONTINUE
C
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
C OBTAINING THE ABOVE QUANTITIES AT THE REQUESTED
C PRINT OUT POINTS
C
1700 CONTINUE
IF(INTERP)1701,1735,1701
INTERPOLATION NEEDED FOR PRINTOUTS
C
1701 CONTINUE
AYAWLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWLS,AYEAWL)
AYAWT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYAWTS,AYEAWT)
BQLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXLS,BEQXL)
BQTINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),BEQXTS,BEQXT)
CFELT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFELTS,CFEL)
CFETT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),CFETS,CFET)
DODINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),DODI(I-2),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(I-2),AIEE(I))
FXQINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXQSAV,FXQ)
FXSINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),FXSSAV,FXS)
HLINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HLSAVE,HL)
HTINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),HTSAVE,HT)
XJLNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XJLSV,XJL)
XNINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XNSV,XN)
OMEGNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),OMEGAE(I-2),OMEGAE(I))

```

```

OMEGRT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),OMEGRS,OMEGAR)
PPSL=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),PPSLX(I-2),PPSLX(I))
QDOTLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTLS,QDOTL)
QDOTTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QDOTTS,QDOTT)
QLISOT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QLISOS,QLISO)
QTISOT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),QTISOS,QTISO)
RORINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RORI(I-2),RORI(I))
RRQINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRQSAV,RRQ)
RRSINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),RRSSAV,RRS)
THETLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETLS,THETAL)
THETTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETTS,THETAT)
THBSNT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),THETAS(I-2),THETAS(I))
TWINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),TW(I-2),TW(I))
UEDGE=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),UE(I-2),UE(I))
WENTH=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),AYEW(I-2),AYEW(I))
XEQXLT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEQXLS,XEQXL)
XEQXTT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XEQXTS,XEQXT)
XMUONT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),XMUOSV,XMUO)
ZTEINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTEX(I-2),ZTEX(I))
ZTRINT=FINTRP(XSTAR(I-2),XSTAR(I),XPO(K),ZTRSAV,ZTR)
COUNT=COUNT+1.
IF(COUNT-9.)1720,1720,1702
1702 CONTINUE
WRITE(6,6200)
COUNT=1.
1720 CONTINUE
XSIPO=TLPL(X,XSIX,XPO(K),IP,MXSV)
ETAPO=TLPL(X,ETAX,XPO(K),IP,NXSV)
WRITE(6,6400) XPO(K),PPSL,UEDGE,DODINT,RORINT,TWINT,ZTEINT,OMEGNT,
1ENTH,WENTH,
2XSIPO,BQLINT,XEQXLT,AYAWLT,XJLNT,EBARLT,HLINT,QLISOT,QJOTLT,CFELT,
3ETAPO,BQTINT,XEQXTT,AYAWTT,XNINT,EBARTT,HTINT,QTISOT,QDOTTT,CFETT,
4THBSNT,ZTRINT,XMUONT,FXQINT,FXSINT,RRQINT,RRSINT
5,OMEGRT,THETLT,THETTT
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
1742  CONTINUE
      WRITE(6,6200)
      COUNT=1.
1750  CONTINUE
      WRITE(6,6400) XPO(K),POPSLX(I),UE(I),DODI(I),RORI(I),TW(I),ZTEX(I)
1,OMEGAE(I),AIEE(I),AYEW(I),
2XSIX(I),BEQXL,XEQXL,AYEAWL,XJL,EBARL,HL,QLISO,QDOTL,CFEL,
3ETAX(I),BEQXT,XEQXT,AYEAWT,XN,EBART,HT,QTISO,QDOTT,CFET,
4THETAS(I),ZTR,XMUO,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
      BEQXLS=BEQXL
      BEQXTS=BEQXT
      CFELTS=CFEL
      CFETTS=CFET
      EBARLS=EBARL
      EBARTS=EBART
      FXQSAV=FXQ
      FXSSAV=FXS
      HLSAVE = HL
      HTSAVE=HT
      XJLSV=XJL
      XNSV=XN
      OMEGRS=OMEGAR
      QDOTLS=QDOTL
      QDOTTS=QDOTT

```

```

24192283
24192284
24192285
24192286
24192287
24192288
24192289
24192290
24192291
24192292
24192293
24192294
24192295
24192296
24192297
24192298
24192299
24192300
24192301
24192302
24192303
24192304
24192305
24192306
24192307
24192308
24192309
24192310
24192311
24192312
24192313
24192314
24192315
24192316
24192317
24192318
24192319
24192320
24192321
24192322
24192323
24192324

```

```

QLISOS=QLISO
QTISOS=QTISO
RRQSAV=RRQ
RRSSAV=RRS
THETLS=THETAL
THETTS=THETAT
XEQXLS=XEQXL
XEQXTS=XEQXT
XMUQSV=XMUO
ZTRSAV=ZTR
INTERP=1
GO TO 1001
1000 CONTINUE
WRITE(6,6100)XTRANS
6100 FORMAT(1H0,26HTRANSITION OCCURRED AT X = ,E12.5)
RETURN
END
$IBFTC 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),XI ) , (A( 237),POPSLX) ,
3(A( 987),DODI ) , (A( 1737),RORI ) , (A( 2737),AIEE ) ,
4(A( 3487),OMEGAE) , (A( 7737),XSTAR ) , (A( 9237),THETAS) ,
5(A( 9987),UE ) , (A(10737),TW ) , (A( 8),NPR )
DIMENSION XSTAR(750),POPSLX(750),UE(750),AIEE(750),
1OMEGAE(750),THETAS(750),DODI(750),RORI(750),TW(750)
DIMENSION T(11),P(11),OMEG(11),DOD(11),ROR(11),X(11),
1TH8S(11),U(11),FOF(11),AE(11)

RADIAN=57.3
BEQXLI=1.0
BEQXTI=1.0
T(1)=TW(1)
P(1)=POPSLX(1)
OMEG(1)=OMEGAE(1)
DOD(1)=DODI(1)
ROR(1)=RORI(1)
X(1)=XSTAR(1)
TH8S(1)=THETAS(1)
24192325
24192326
24192327
24192328
24192329
24192330
24192331
24192332
24192333
24192334
24192335
24192336
24192337
24192338
24192339
24192340
24192341
24192342
24192343
24192344
24192345
24192346
24192347
24192348
24192349
24192350
24192351
24192352
24192353
24192354
24192355
24192356
24192357
24192358
24192359
24192360
24192361
24192362
24192363
24192364
24192365
```

C

```

U(1)=UE(1)
AE(1)=AIEE(1)
DO 300 ITER=1,5
DX=DX/10.
BX=1./10.*ITER
DO 100 I=2,11
B=I-1
B=BX*B
X(I)=X(1)+B*(XSTAR(2)-XSTAR(1))
P(I)=P(1)+B*(POPSLX(2)-POPSLX(1))
T(I)=T(1)+B*(TW(2)-TW(1))
U(I)=U(1)+B*(UE(2)-UE(1))
AE(I)=AE(1)+B*(AIEE(2)-AIEE(1))
OMEG(I)=OMEG(1)+B*(OMEGAE(2)-OMEGAE(1))
DOD(I)=DOD(1)+B*(DODI(2)-DODI(1))
ROR(I)=ROR(1)+B*(RORI(2)-RORI(1))
TH8S(I)=TH8S(1)+B*(THETAS(2)-THETAS(1))
100 CONTINUE
C
NG=0
DO 200 I=1,11
CALL IWALL(T(I),P(I),AYEW)
CALL SOMEGA(H,P(I),ZTS,OMEGAS)
CALL SOMEGA(AYEW,P(I),ZTW,OMEGAW)
CALL RORMUR(P(I),OMEGAS,OMEG(I),OMEGAW,ROMUR,ZIR,AYER,
1SIGR)
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),DDELDX,DX,1)
GO TO 1270
1220 CONTINUE
CALL DERV(DOD(I-1),DOD(I),DOD(I+1),DOD(I+2),DOD(I+3),DDELDX,DX,2)
GO TO 1270
1230 CONTINUE
IF(I-10)1240,1250,1260
1240 CONTINUE
CALL DERV(DOD(I-2),DOD(I-1),0.,DOD(I+1),DOD(I+2),DDELDX,DX,3)
GO TO 1270
1250 CONTINUE
CALL DERV(DOD(I-3),DOD(I-2),DOD(I-1),DOD(I),DOD(I+1),DDELDX,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

```

1260 GO TO 1270
      CONTINUE
      CALL DERV(DOD(I-4),DOD(I-3),DOD(I-2),DOD(I-1),DDEL DX,DX,5)
1270 CONTINUE
      IF(NPR .LE. 4) GO TO 1320
      XN=X(I)/DOD(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.
      AYESL=AE(I)
      GO TO 1370
1340 CONTINUE
      THETSL=(XN-1.)*TH8S(I)
      XK=2./3.*XN*(XN-1.)
      EXPK=-.194*EXP(-XK)
      GO TO 1360
1350 CONTINUE
      THETSL=(1.-1./XN)*TH8S(I)
      XK=2./3.*(XN-1.)
      EXPK=.194*EXP(-XK)
1360 CONTINUE
      VP= U(I)*COS(THETSL/RADIAN)
      AYESL=H-.5*VP**2
1370 CONTINUE
      C
      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)151,152,153
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*BEQXLI
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
BEQXLI=BEQXL
BEQXTI=BEQXT
300 CONTINUE
BEQXL=BEQXLI
BEQXT=BEQXTI
C
310 CONTINUE
WRITE(5,6010)BEQXLI,BEQXTI
6010 FORMAT(1H0,18HINITIAL BEQ VALUES/10H BEQXLI = ,E12.5/10H BEQXTI =
1,E12.5//)
C
DX=1.E+5*DX
RETURN
END
$IBFTC INITIAL DECK
SUBROUTINE INITIAL(L1,NTYPE)
C
CALCULATES INITIAL CONDITIONS
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A( 1),ACH ) , (A( 3),ALT ) , (A( 15),H ) ,
2(A( 18),PINF ) , (A( 19),POPINF) , (A( 20),POPSL ) ,
3(A( 21),QINF ) , (A( 25),RHO ) , (A( 26),TEMP ) ,
4(A( 31),VEL ) , (A( 32),VELP )

```

```

C
C
C
      WIND TUNNEL I
      GO TO(10,20,30),L1
10  CONTINUE
      VEL=49.*ACH*SQR(T(TEMP))
      H=6000.*TEMP+.5*VEL**2
      RHO=PINF/(1716.*TEMP)
      GO TO 40

C
      WIND TUNNEL II
20  CONTINUE
      AYEI=H-.5*VEL**2
      IF(AYEI-5930000.)21,22,22
21  CONTINUE
      SONIC=.6325*SQR(T(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

C
      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

C
      ACHN=ACH
      CALL STACON(PSTPNF,AYEST,VELP,ACH,ACHN,H,PINF,VEL)
      GO TO(60,70,80),L1

C
60  CONTINUE
      POPINF=PSTPNF
      POPSL=POPINF*PINF/PSL

```

```

24192491
24192492
24192493
24192494
24192495
24192496
24192497
24192498
24192499
24192500
24192501
24192502
24192503
24192504
24192505
24192506
24192507
24192508
24192509
24192510
24192511
24192512
24192513
24192514
24192515
24192516
24192517
24192518
24192519
24192520
24192521
24192522
24192523
24192524
24192525
24192526
24192527
24192528
24192529
24192530
24192531

```

```

GO TO 80
70 CONTINUE
  POPINF=PSTPNF
  PINFP=POPSL*PSL/POPINF
  QINF=PINFP*VEL**2/2./R/TEMP
  PINF=PINFP
  QINF=QINF
80 CONTINUE
C
  WRITE(6,6020) NTYPE,VEL,ACH,H,TEMP,PINF,QINF,POPSL,POPINF,ALT
6020 FORMAT(1H0,23HINITIAL CONDITIONS FOR ,A6,5H CASE//4X,8HVELOCITY,4X24192542
1,4HMACH,8X,8HENTHALPY,4X,4HTINF,8X,4HPINF,8X,4HQINF,8X,6HPO/PSL,6X24192543
2,7HPO/PINF,4X,8HALTITUDE//1X,9E12.5///)
  RETURN
END
$IBFTC REF DECK
SUBROUTINE REF
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
COMMON/Q/A(11600),IA(10)
EQUIVALENCE
1(A( 1),ACH) , (A( 4),AYEWO) , (A( 15),H) ,
2(A( 16),HO) , (A( 17),HREF) , (A( 18),PINF) ,
3(A( 20),POPSL) , (A( 21),QINF) , (A( 23),RCLREF) ,
4(A( 24),RHMREF) , (A( 29),TWREF) , (A( 31),VEL) ,
5(A( 32),VELP)
C
  AYEE=H
  BETAS=0.5
C
  CALL JWall(TWREF,POPSL,AYEW)
  CALL SOMEGA(AYEW,POPSL,ZTW,OMEGW)
  CALL SOMEGA(AYEE,POPSL,ZTE,OMEGE)
  CALL SOMEGA(H,POPSL,ZTS,OMEGS)
  CALL RORMUR(POPSL,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,AYER,SIGR)
  CALL JAYELL(AYEW,H,AYEE,ZTE,ZTS,SIGR,BETAS,POPSL,XJL)
  CALL MUZERO(OMEGR,ZTR,AYER,AYEE,H,POPSL,XMUO,XL)
  HO=.664*ACCG*XL/SIGR**.645*SQRT(ROMUR*VEL*XJL*
1VELP/RHMREF)
  QDOTO=HO*(H-AYEW)/25031.3
  AYEWO=AYEW
  ACHN=0.5*ACH

```

24192532
24192533
24192534
24192535
24192536
24192537
24192538
24192539
24192540
24192541
24192542
24192543
24192544
24192545
24192546
24192547
24192548
24192549
24192550
24192551
24192552
24192553
24192554
24192555
24192556
24192557
24192558
24192559
24192560
24192561
24192562
24192563
24192564
24192565
24192566
24192567
24192568
24192569
24192570
24192571
24192572
24192573

```

EXPK=0.
CALL STACON(PSTPNF,AYEST,VELP,ACH,ACHN,H,PINF,VEL)
PSTPSL=PSTPNF*PINF/PSL
CALL IWALL(TWREF,PSTPSL,AYEW)
AYEE=AYEST
AYESL=AYEST
CALL SOMEGA(AYEW,PSTPSL,ZTW,OMEGW)
CALL SOMEGA(AYEE,PSTPSL,ZTE,OMEGE)
CALL SOMEGA(H,PSTPSL,ZTS,OMEGS)
CALL RORMUR(PSTPSL,OMEGS,OMEGE,OMEGW,OMEGR,ROMUR,ZTR,
IAYER,SIGR)
CALL EBAR(AYEW,AYESL,H,SIGR,EXPK,PSTPSL,OMEGE,GAMC,GAMO,
IEBARL,EBART)
CALL MUZERO(OMEGR,ZTR,AYER,AYEE,H,PSTPSL,XMUO,XL)
XEQ=866*RCLREF/(1.+GAMC)/VELP
FXQ=(1.6*(1.+GAMO))/(1.+7652*GAMO)**.333*(1.+GAMO)/(1.+GAMC)
RRQREF=.866*ROMUR*VEL*XEQ/(XMUO*FXQ)**2
HREFL=.332*ACCG*XL/SIGR**.645*XMUO*FXQ/XEQ*SQRT(RRQREF)
HREF=.185*ACCG*XL/SIGR**.645*XMUO*FXQ/XEQ*RRQREF/
1(ALOG10(RRQREF+3000.))**.584
AYAWRL=H-(1.-SQRT(SIGR))*(H-AYEE)
AYAWRT=H-(1.-SIGR**.333)*(H-AYEE)
QREFL=HREFL*(AYAWRL-AYEW)/25031.3
QREFT=HREFT*(AYAWRT-AYEW)/25031.3
TUREFL=SIGR**.645/ACCG/XL*HREFL
TUREFT=SIGR**.645/ACCG/XL*HREFT
TAURL=.866*VEL*TUREFL
TAURT=.866*VEL*TUREFT
CFREFL=TAURL/QINF
CFREFT=TAURT/QINF
AYEREF=AYEW
HO24=.24*HO
HRL24=.24*HREFL
HRT24=.24*HREFT
WRITE(6,6000) HO,HREFL,HREFT,AYAWRL,TAURL,TAURT,RHMREF,
1HO24,HRL24,HRT24,AYAWRT,CFREFL,CFREFT,RCLREF,
2QDOTO,QREFL,QREFT,AYEREF,AYEWO,TWREF
6000 FORMAT(1HO,2HHO,1OX,6HHREF,L,6X,6HHREF,T,6X,9HIAW,REF,L,3X,9HTAURE,
1F,L,4X,8HTAUREF,T,4X,7HREM,REF/
26H .24HO,7X,9H.24HREF,L,3X,9H.24HREF,T,3X,9HIAW,REF,T,3X,11HCFINF,24192613
3REF,L,1X,11HCFINE,REF,T,9H RCYL,REF/
24192614
24192574
24192575
24192576
24192577
24192578
24192579
24192580
24192581
24192582
24192583
24192584
24192585
24192586
24192587
24192588
24192589
24192590
24192591
24192592
24192593
24192594
24192595
24192596
24192597
24192598
24192599
24192600
24192601
24192602
24192603
24192604
24192605
24192606
24192607
24192608
24192609
24192610
24192611
24192612
24192613
24192614

```

```

46H QDOTO,7X,10HQDOT,REF,L,2X,10HQDOT,REF,I,2X,6HIW,REF,6X,4HIW,O,824192615
5X,6HTW,REF//
63(7E12.4/)/)
RETURN
END
$IBFTC SONENT DECK
SUBROUTINE SONENT(ENTH,SONIC,IEROR)
C ROUTINE TO OBTAIN FREE STREAM SPEED OF SOUND
C ENTHALPY IN FT**2/SEC**2 AND SPEED OF SOUND IN FT/SEC
C DIMENSION SONICT(10),AYE(10),LA(8)
INTEGER XLOC
DATA KSS/1/
DATA(AYE(1),I=1,9)/
15.93E6,8.475E6,10.17E6,11.86E6,13.56E6,15.26E6,16.95E6,
218.64E6,20.34E6/
DATA(SONICT(I),I=1,9)/
11540.,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=TAB(ENTH,LA(1))
GO TO 40
30 CONTINUE
WRITE(6,6100)ENTH
6100 FORMAT(1H0,63HFREE STREAM ENTHALPY EXCEEDS TABLE VALUE IN SONENT.
1 ENTHALPY = ,E12.5/25H MAXIMUM VALUE = 20.34E6. )
40 CONTINUE
RETURN
END
$IBFTC STACON DECK
SUBROUTINE STACON(PSTPNF,AYEST,VELP,A'CH,ACHN,H,PINF,VEL)
C DIMENSION L(8),ACOST(11),CORRT(11)
DATA (ACOST(I),I=1,11)/.1,.2,.3,.5,.7,.8,1.0,1.2,1.5,2.0,2.5/

```

```

DATA (CORRT(I),I=1,11)/1.122,1.118,1.1115,1.0955,1.068,1.055,
11.033,1.021,1.0076,1.002,1.000/
COMMON/BLKCON/R,PSL,ACCG,RADIAN,EPS
DATA (L(I),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.*ACHN**2-1.))*2.5
30 CONTINUE
R2R10=6.*ACHN**2/(ACHN**2+5.)
CALL SOMEGA(AYEST,PSTO,ZTSTO,OMEG)
PST1=PINF/PSL+(PSTO-PINF/PSL)*((2.*R2R10-6006.*ZTSTO/AYEST)/
1(2.*R2R10-1.))
CALL SOMEGA(AYEST,PST1,ZTST1,OMEG)
PST2=PINF/PSL+(PSTO-PINF/PSL)*((2.*R2R10-6006.*ZTST1/AYEST)/
1(2.*R2R10-1.))
PSTPNF=PST2*PSL/PINF
VELPO=SQRT(2./7.*(1.+5./ACHN**2)*(1.-1./PSTPNF))*6006.*
1ZTST1/AYEST))
IF(ACHN-2.5)40,40,50
C
40 CONTINUE
L(1)=LOC(L(1))
L(2)=LOC(ACOST(1))
L(3)=LOC(CORRT(1))
L(8)=0
CORR=TAB(ACHN,L(1))
GO TO 60
C
50 CONTINUE
CORR=1.0
60 CONTINUE
VELP=VELPO*CORR**5
RETURN
END
$1BFTC ATMOS DECK
SUBROUTINE ATMOS(ALT,CS,P,DENS,T)
C SAVE SENSE LIGHT

```

```

24192657
24192658
24192659
24192660
24192661
24192662
24192663
24192664
24192665
24192666
24192667
24192668
24192669
24192670
24192671
24192672
24192673
24192674
24192675
24192676
24192677
24192678
24192679
24192680
24192681
24192682
24192683
24192684
24192685
24192686
24192687
24192688
24192689
24192690
24192691
24192692
24192693
24192694
24192695
24192696
24192697

```

24192698	CALL SLITET(1,K000FX)
24192699	GO TO(4600,4700),K000FX
24192700	4600 SAVE = 1.0
24192701	GO TO 4800
24192702	4700 SAVE = 0.0
24192703	4800 CONTINUE
24192704	C SET ERROR INDICATOR
24192705	I = 0
24192706	C CONVERT TO METERS
24192707	C
24192708	Z = ALT*0.3048
24192709	C
24192710	COMPUTE GEOPOTENTIAL ALTITUDE
24192711	C
24192712	C
24192713	IF (Z-90000.0) 1000, 1000, 2000
24192714	C ALTITUDE LESS THAN NINETY THOUSAND METERS
24192715	1000 Z = 6356766.0*Z/(6356766.0+Z)
24192716	WEIMOL = 28.9644
24192717	IF (Z) 1, 12, 3
24192718	1 I = -1
24192719	Z = 0.0
24192720	2 GO TO 12
24192721	3 IF (Z-79000.0) 5, 4, 4
24192722	4 TMB = 180.65
24192723	GRAD = 0.0
24192724	ZB = 79000.0
24192725	PB = 1.0377E-2
24192726	GO TO 3000
24192727	5 IF (Z-47000.0) 6, 7, 7
24192728	6 IF (Z-20000.0) 8, 9, 9
24192729	7 IF (Z-61000.0) 10, 11, 11
24192730	8 IF (Z-11000.0) 12, 13, 13
24192731	9 IF (Z-32000.0) 14, 15, 15
24192732	10 IF (Z-52000.0) 16, 17, 17
24192733	11 TMB = 252.65
24192734	GRAD = -4.0
24192735	ZB = 61000.0
24192736	PB = 1.82099E-1
24192737	GO TO 3000
24192738	12 TMB = 288.15
24192739	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-400000.0) 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

```

28 GO TO 4000
 TMB = 210.65
 GRAD = 0.005
 ZB = 100000.0
 PB = .30074613E-01
 GO TO 4000
 29 IF (Z-120000.0) 35, 36, 36
 30 IF (Z-160000.0) 37, 38, 38
 31 IF (Z-190000.0) 39, 40, 40
 32 IF (Z-300000.0) 41, 42, 42
 33 TMB = 2160.65
 ZB = 400000.0
 PB = .40303675E-05
 GRAD = 0.0026
 GO TO 4000
 34 IF (Z-600000.0) 43, 44, 44
 35 TMB = 260.65
 ZB = 110000.0
 PB = .73544163E-02
 GRAD = 0.01
 GO TO 4000
 36 TMB = 360.65
 GRAD = 0.02
 ZB = 120000.0
 PB = .25216515E-02
 GO TO 4000
 37 TMB = 960.65
 GRAD = 0.015
 ZB = 150000.0
 PB = .50617014E-03
 GO TO 4000
 38 TMB = 1110.65
 GRAD = 0.01
 ZB = 160000.0
 PB = .36942892E-03
 GO TO 4000
 39 TMB = 1210.65
 GRAD = 0.007
 ZB = 170000.0
 PB = .27925979E-03
 GO TO 4000
 40 TMB = 1350.65

24192781
 24192782
 24192783
 24192784
 24192785
 24192786
 24192787
 24192788
 24192789
 24192790
 24192791
 24192792
 24192793
 24192794
 24192795
 24192796
 24192797
 24192798
 24192799
 24192800
 24192801
 24192802
 24192803
 24192804
 24192805
 24192806
 24192807
 24192808
 24192809
 24192810
 24192811
 24192812
 24192813
 24192814
 24192815
 24192816
 24192817
 24192818
 24192819
 24192820
 24192821
 24192822

```

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 90000.0 AND 170000.0, COMPUTE M
EM = (((.141865090E-27)*Z -.111458341E-21) *Z +.359201416E-16)*Z -24192855
1.606652130E-11) *Z +.565215183E-06)*Z -.275300356E-01) *Z +5.7695724192856
2191E2
WEIMOL = EM*28.9644/28.966
GO TO 4050
4500 CONTINUE
C Z GREATER THAN 170000.0, COMPUTE M
EM = (((.1805540E-33)*Z -3.4291620E-27)
1*Z +3.8159669E-21)*Z -2.0402054E-15)

```

24192823
24192824
24192825
24192826
24192827
24192828
24192829
24192830
24192831
24192832
24192833
24192834
24192835
24192836
24192837
24192838
24192839
24192840
24192841
24192842
24192843
24192844
24192845
24192846
24192847
24192848
24192849
24192850
24192851
24192852
24192853
24192854
24192855
24192856
24192857
24192858
24192859
24192860
24192861
24192862
24192863

```

2*Z +5.6422141E-10)*Z -1.0700489E-4)
3*Z +3.5629995E1
WEIMOL = EM*28.9644/28.966
4050 CONTINUE
C COMPUTE TEMPERATURE
T = TM*WEIMOL/28.9644
C SPEED OF SOUND IS CONSTANT
CS = 269.44
C COMPUTE PRESSURE
EMOR = 0.00348368
C = (-ZB+TMB/GRAD)*0.000001
D=6.3675708
CMD=C-D
CMD2=CMD*CMD
CMD3=CMD2*CMD
CMD4=CMD2*CMD2
CMD5=CMD2*CMD3
CMD6=CMD3*CMD3
CMD7=CMD4*CMD3
CMD8=CMD4*CMD4
CMD9=CMD5*CMD4
CMD10=CMD5*CMD5
C2=C*C
C3=C2*C
D2=D*D
PBLN=ALOG(PB)
TEM14 = EMOR/GRAD
CALL SLITE (1)
C COMPUTE USING ZB
Z6=ZB*.000001
4100 ZPD=Z6+D
ZPC=Z6+C
ZPD2=ZPD*ZPD
ZPD3=ZPD2*ZPD
ZPD4=ZPD2*ZPD2
ZPD5=ZPD2*ZPD3
ZPD6=ZPD3*ZPD3
ZPD7=ZPD3*ZPD4
ZPD8=ZPD4*ZPD4
ZPD9=ZPD5*ZPD4
ZPC2=ZPC*ZPC
ZPC3=ZPC2*ZPC

```

```

24192864
24192865
24192866
24192867
24192868
24192869
24192870
24192871
24192872
24192873
24192874
24192875
24192876
24192877
24192878
24192879
24192880
24192881
24192882
24192883
24192884
24192885
24192886
24192887
24192888
24192889
24192890
24192891
24192892
24192893
24192894
24192895
24192896
24192897
24192898
24192899
24192900
24192901
24192902
24192903
24192904
24192905

```



```

7000 P = PB*EXP(-9.80665*28.9644*0.001*(Z-Z0)/(8.31432*TMB))
8000 CONTINUE
      DENS = 28.9644*P/(8.31432*TM)*0.1
      CONVERT DENSITY(KG/M3) TO LB/FT3
      DENS = DENS*(0.3048)**3/0.45359237
      CONVERT DENSITY TO LB(MASS)/FT3
      DENS = DENS/32.1741
      CONVERT TO P(MM HG)
      P = P*.750061682E+00
      CONVERT TO IN HG
      P = P*.393700787E-01
      CONVERT TO LB/FT2
      P = P*70.7269
      CONVERT CS(M/SEC2) TO CS(FT/SEC2)
      CS = CS*3.2808399
      CONVERT T(K) TO T(R)
      T = T*1.8
      RESTORE SENSE LIGHT
8100 IF (SAVE) 8200, 8300, 8200
8200 CALL SLITE (1)
      GO TO 8400
8300 CALL SLITET(1,K000FX)
      GO TO(8400,8400),K000FX
8400 CONTINUE
      IF (I) 300, 9000, 200
200 WRITE (6,400)
      GO TO 9000
300 WRITE (6,500)
9000 CONTINUE
      RETURN
400 FORMAT(1H0,///5X,79HALTITUDE INPUT TO ATMOS EXCEEDED ROUTINE RANGE
1E - ASSIGNED VALUE OF 2,300,000 FT///)
500 FORMAT(1H0,///5X,68HALTITUDE INPUT TO ATMOS BELOW ROUTINE RANGE
1 ASSIGNED VALUE OF 0 FT///)
      END

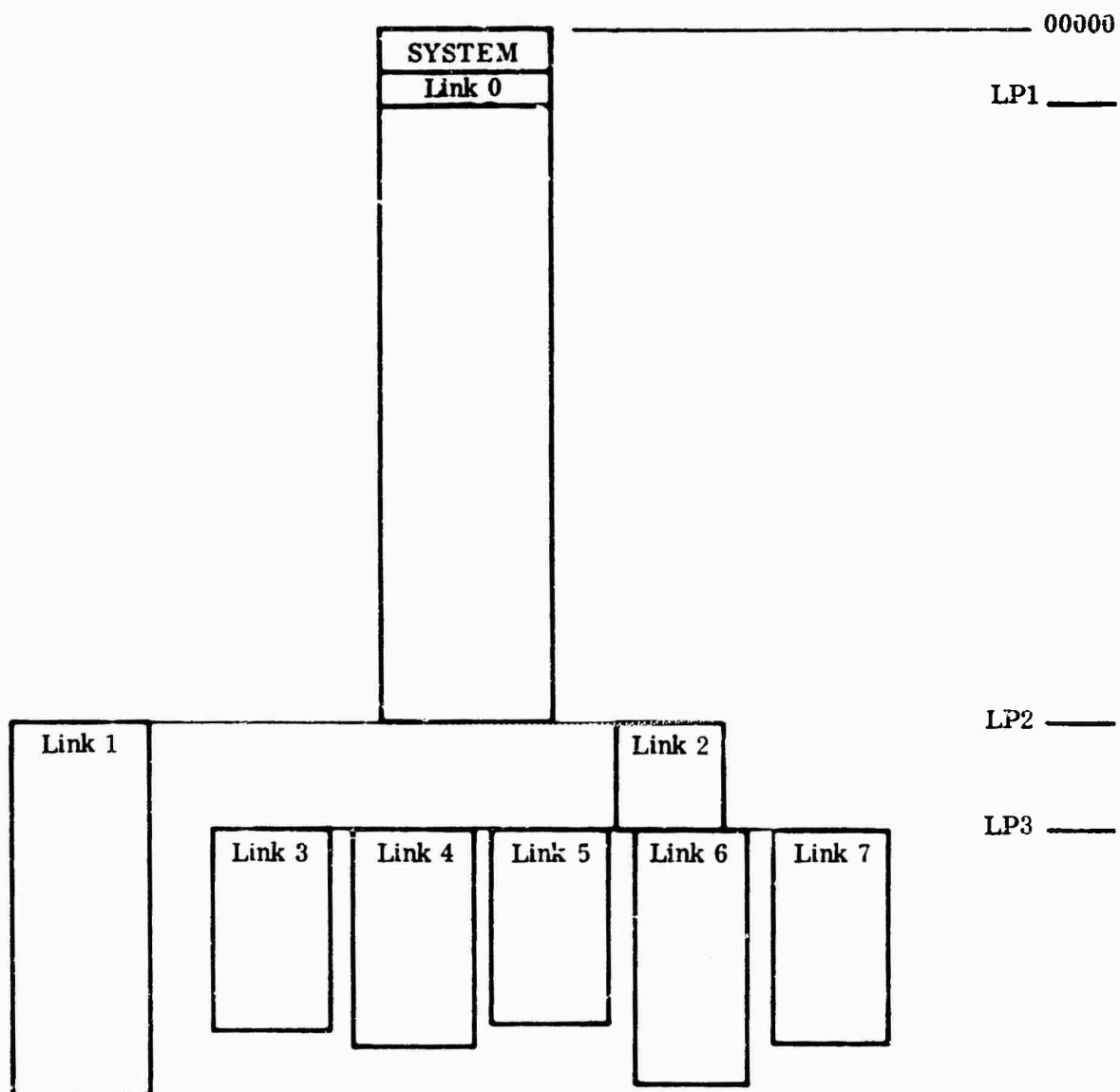
```

24192947
24192948
24192949
24192950
24192951
24192952
24192953
24192954
24192955
24192956
24192957
24192958
24192959
24192960
24192961
24192962
24192963
24192964
24192965
24192966
24192967
24192968
24192969
24192970
24192971
24192972
24192973
24192974
24192975
24192976
24192977
24192978
24192979
24192980
24192981
24192982

002982

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

[illegible]

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
SOMEGA
TABLE 2
TABLE 6
TABLE 7
TABLE 14
TABLE 18
TABLE 19
TABLE 20

Link 3

DELTA

Link 4

CYLIND
FLOW

Link 5

AXI2D
HEMI

Link 6

QTRAN
BEQI

Link 7

INTIAL
REF
SONENT
STACON
ATMOS

SECTION III

THE TURBULENT NONSIMILAR BOUNDARY LAYER PROGRAM (NSBL)

1. NOMENCLATURE

<u>Program Symbol</u>	<u>Math Symbol</u>	<u>Description</u>	<u>Units</u>
CP	C_P	specific heat of air at constant pressure, 6006	$\text{ft}^2/\text{sec}^2 - ^\circ\text{R}$
	δ	boundary layer thickness	ft
* DELETA	$\delta(\eta/\rho)$	increment in boundary layer parameter	ft^3/slug
DELSTR	δ^*	boundary layer displacement thickness	ft
* DHDX	$\partial I/\partial x$	derivative of total enthalpy	ft/sec^2
* DHDY	$\partial I/\partial y$	derivative of total enthalpy	ft/sec^2
* DMUDY	$\partial \mu/\partial y$	derivative of viscosity	$\text{lb-sec}/\text{ft}^3$
DQ	$\partial \dot{q}/\partial y$	derivative of the heating rate	$\text{Btu}/\text{ft}^3\text{-sec}$
DRDX	$\partial r/\partial x$	derivative of three-dimensional flow parameter	---
DS	ΔS	increment for stagnation region problems	ft
* DUDX	$\partial u/\partial x$	derivative of velocity	1/sec
* DUDY	$\partial u/\partial y$	derivative of velocity	1/sec
* DUDYL	$(\partial u/\partial y)_e$	velocity derivative at boundary layer edge	1/sec
* DVDY	$\partial v/\partial y$	derivative of velocity normal to surface	1/sec
DX	Δx	increment in x-direction	ft

* 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.

<u>Program symbol</u>	<u>Math symbol</u>	<u>Description</u>	<u>Units</u>
DY	Δy	increment in y-direction	ft
* D2HDY	$\partial^2 I / \partial y^2$	second derivative of total enthalpy	1/sec ²
* D2UDY	$\partial^2 u / \partial y^2$	second derivative of velocity	1/sec-ft
ENDX	x_f	final value of x	ft
* ENTH	i	static enthalpy	ft ² /sec ²
EPS	ϵ	eddy viscosity	lb-sec/ft ²
EPSY	$\partial \epsilon / \partial y$	derivative of eddy viscosity	lb-sec/ft ³
* ETA	η	similarity parameter	---
* H	I	total enthalpy	ft ² /sec ²
* PR	σ	Prandtl number	---
* PRESSURE	P	pressure	lb/ft ²
* QP	$\dot{q}$	heating rate	Btu/ft ² -sec
QW	$\dot{q}_w$	heating rate at the wall	Btu/ft ² -sec
QQW	QQW	heating rate at the wall	Btu/ft ² -sec
R	r	three-dimensional flow parameter	ft
	R	universal gas constant, 1716	ft ² /sec ² -°R
* RHO	ρ	density	lb-sec ² /ft ⁴
SI	S_I	initial value of S for the calculation of η	ft
STARTX	x_o	initial value of x	ft
* T	T	temperature	°R
* TAU	τ	shear force	lb/ft ²
TAULP	τ_L	local laminar shear force	lb/ft ²

<u>Program symbol</u>	<u>Math symbol</u>	<u>Description</u>	<u>Units</u>
TAULW	$\tau_{w,L}$	laminar shear force at the wall	lb/ft ²
TAULY	$\partial\tau_L/\partial y$	derivative of the laminar shear force	lb/ft ³
TAUTP	τ_T	local turbulent shear force	lb/ft ²
TAUTW	$\tau_{w,T}$	turbulent shear force at the wall	lb/ft ²
TAUTY	$\partial\tau_T/\partial y$	derivative of the turbulent shear force	lb/ft ³
TAUW	τ_w	shear force at the wall	lb/ft ²
THETA	θ	boundary layer momentum thickness	ft
TURBS	$\frac{\partial\tau_T/\partial y}{\partial\tau_L/\partial y}$	ratio of the turbulent to the laminar shear force derivative	---
* U	u	velocity in the x-direction	ft/sec
* V	v	velocity, $\tilde{v}$ in Volume I (Appendix C)	ft/sec
X	x	coordinate tangent to the surface	ft
XI	x_I	initial value of x for the calculation of η	ft
* XMU	μ	absolute viscosity of air	lb-sec/ft ²
* Y	y	coordinate normal to the surface	ft

Subscript notation:

Subscript

e	evaluated at the boundary layer edge
i	evaluated at the i th 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),

$$\rho = P/(RT)$$

Continuity (C-17),

$$v_i = \frac{1}{\left[\frac{\partial u_i}{\partial y} - \frac{2u_i}{\Delta y} \right]} \left\{ u_i^2 \left(\frac{1}{P} \frac{\partial P}{\partial x} + \frac{1}{r} \frac{\partial r}{\partial x} \right)_i + \frac{1}{\rho_i} \left[- \frac{\partial P}{\partial x} + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) + \frac{\partial \tau_T}{\partial y} \right]_i \right. \\ \left. - \frac{u_i}{i \rho_i} \left[\frac{\partial}{\partial y} \left(\mu \frac{\partial i}{\partial y} \right) + \frac{\partial \dot{q}_T}{\partial y} + \frac{\partial u}{\partial y} \left(\mu \frac{\partial u}{\partial y} + \tau_T \right) + u \frac{\partial P}{\partial x} \right]_i - \frac{2u_i v_{i-1}}{\Delta y} - \left(\frac{\partial v}{\partial y} \right)_{i-1} u_i \right\}$$

x-momentum (C-6),

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

and Energy (C-7):

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

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

$$\frac{\partial \tau_T}{\partial y} = \left(\frac{\rho}{\rho_e} \right)^4 \left[.054 \left(\frac{\rho u y}{\mu} \right)^{.833} \left(\frac{u}{u_e} \right)^{12} \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \right]$$

$$\epsilon_i = \frac{\tau_T}{\left(\frac{\partial u}{\partial y} \right)}$$

$$\dot{q}_T = \frac{\epsilon}{\sigma} \frac{\partial i}{\partial y}$$

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 $y = \Delta y$.

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

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

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 ($x_0^1 + \Delta x$) 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 η is calculated as indicated below in subsection b., item (4). A value of $\eta_{\max}$ 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 .

$$\dot{q}_w = -\frac{1}{778} \int_0^\delta \left[\rho u \frac{\partial I}{\partial y} + \rho v \frac{\partial I}{\partial y} - u \tau \right] dy$$

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

$$\tau_T = \int_0^\delta \frac{\partial \tau_T}{\partial y} dy$$

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 .

$$u = f(y) \text{ at } x_0 \qquad \frac{\partial r}{\partial x} = f(x)$$

$$I = f(y) \text{ at } x_0 \qquad \left. \frac{\partial u}{\partial y} \right|_e = f(x)$$

$$P = f(x) \qquad v_1 = f(x)$$

$$r = f(x) \qquad i_w = f(x)$$

In addition, the following quantities must be specified:

$$\Delta x \qquad \Delta y \qquad \eta_{\max} \qquad x_0 \qquad x_f \qquad s_I \qquad x_I$$

(1) Calculation of tabular values

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

$$i = 2, 3, \dots, (NP-1)$$

$$\left. \frac{\partial P}{\partial x} \right|_{i=1} = 0$$

(2) Values obtained from tables (linear interpolation)

$$P = f(x) \qquad \left(\frac{\partial u}{\partial y} \right)_e = f(x)$$

$$\frac{\partial P}{\partial x} = f(x) \qquad v_1 = f(x)$$

$$r = f(x) \qquad i_w = f(x)$$

$$\frac{\partial r}{\partial x} = f(x)$$

(3) Computation at initial $x = x_0$

$$y_i = (i - 1) \Delta y \qquad i = 1, 2, 3, \dots, 200$$

$$u_i = u_{i-1} + \left(\frac{\partial u}{\partial y} \right)_e \Delta y \qquad i = (LIST + 1), \dots, 200$$

$$I_i = I_{LIST}$$

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

$$T_i = (I - u^2/2)/C_P$$

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

$$\rho_i = P_i / (1716 T_i)$$

$$C_P = 6006 \text{ ft}^2/\text{sec}^2 \text{ } ^\circ\text{R}$$

$$i = 2, 3, \dots, 200.$$

(4) Calculation of $\delta\eta$

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

$$S = S_1 + \rho_e u_e \mu_e r^2 x_1 \text{ initially. At } x + \Delta x$$

$$S_{x+\Delta x} = \rho_e u_e \mu_e r^2 \Delta x$$

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

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

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

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

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

$$\left. \frac{\partial I}{\partial y} \right|_2 = \frac{1}{6 \Delta y} (9I_3 - 2I_4 - 6I_2 - I_1)$$

$$\left. \frac{\partial^2 I}{\partial y^2} \right|_2 = \frac{1}{6 \Delta y^2} (I_4 - 3I_3 - 9I_2 + 5I_1)$$

$i = 2$

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

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

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

$$\left. \frac{\partial^2 I}{\partial y^2} \right|_i = \frac{I_{i+1} + I_{i-1} - 2I_i}{\Delta y^2}$$

$$\left. \frac{\partial \tau_L}{\partial y} \right|_i = \mu_i \left. \frac{\partial^2 u}{\partial y^2} \right|_i + \left. \frac{\partial \mu}{\partial y} \right|_i \left. \frac{\partial u}{\partial y} \right|_i$$

$$\left. \frac{\partial \tau_T}{\partial y} \right|_i = .054 \left(\frac{\rho_i u_i y_i}{\mu_i} \right)^{.833} \left(\frac{u_i}{u_e} \right)^{12} \left(\frac{T_e}{T_i} \right)^4 \left(\frac{\partial \tau_L}{\partial y} \right)_i$$

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

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

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

$$i = 2, 3, \dots, \text{MIN}(j, 199)$$

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

$$\frac{\Delta y}{u} \left[\frac{\partial u}{\partial y} - \left. \frac{\partial u}{\partial y} \right|_e \right] \leq .00001 \quad \frac{\Delta y}{I} \frac{\partial I}{\partial y} \leq .00005$$

The following are calculated for the range $i \leq j$

$$\tau_{T_i} = \tau'_T|_i - \tau'_T|_j$$

$$\tau_{L_i} = \tau'_L|_i - \tau'_L|_j$$

$$\tau_i = \tau_{L_i} - \tau_{T_i}$$

$$\tau_1 = -\tau'_T|_j - \tau'_L|_j$$

$$\epsilon_i = \tau_{T_i} / (\partial u / \partial y)_i$$

$$\left. \frac{\partial \epsilon}{\partial y} \right|_i = \left\{ .054 \left(\frac{\rho_i u_i y_i}{\mu_i} \right)^{.833} \left(\frac{u_i}{u_e} \right)^{12} \left(\frac{T_e}{T_i} \right)^4 \left. \frac{\partial \tau_L}{\partial y} \right|_i - \epsilon_i \left. \frac{\partial^2 u}{\partial y^2} \right|_i \right\} \left\{ \frac{1}{(\partial u / \partial y)_i} \right\}$$

$$\left. \frac{d\dot{q}}{dy} \right|_i = \frac{1}{\sigma} \left\{ (\mu_i + \epsilon_i) \left. \frac{\partial^2 I}{\partial y^2} \right|_i + \left(\left. \frac{\partial \mu}{\partial y} \right|_i + \left. \frac{\partial \epsilon}{\partial y} \right|_i \right) \left. \frac{\partial I}{\partial y} \right|_i \right. \\ \left. - \frac{1}{2\Delta y} (u_{i+1} \tau_{i+1} - u_{i-1} \tau_{i-1}) \right\}$$

$$\dot{q}_i = \dot{q}_{i-1} + \left(\left. \frac{d\dot{q}}{dy} \right|_{i-1} + \left. \frac{\partial \dot{q}}{\partial y} \right|_i \right) \frac{\Delta y}{2}$$

$$\text{PART } 1_i = u_i^2 \left(\frac{1}{r} \frac{d^2}{dx^2} + \frac{1}{P} \frac{dP}{dx} \right)$$

$$\text{PART } 2_i = \frac{1}{\rho_i u_i} \left(\left. \frac{\partial \tau_L}{\partial y} \right|_i + \left. \frac{\partial \tau_T}{\partial y} \right|_i - \left. \frac{\partial P}{\partial x} \right|_i \right)$$

$$\text{PART } 3_i = \frac{1}{\rho_i u_i} \left[\left. \frac{d\dot{q}}{dy} \right|_i + (u_{i+1} \tau_{i+1} - u_{i-1} \tau_{i-1}) \right] \frac{1}{2\Delta y}$$

$$\text{PART } 6_i = u_i \left[\frac{2 v_{i-1}}{\Delta y} + \left. \frac{\partial v}{\partial y} \right|_{i-1} \right]$$

$$v_i = \frac{1}{\left. \frac{\partial u}{\partial y} \right|_i - \frac{2 u_i}{\Delta y}} \left[\text{PART } 1_i + u_i \text{PART } 2_i - \frac{u_i^2}{(1 - u_i^2/2)} (\text{PART } 3_i \right. \\ \left. - u_i \text{PART } 2_i) - \text{PART } 6_i \right]$$

$$\left. \frac{\partial v}{\partial y} \right|_i = \frac{2}{\Delta y} (v_i - v_{i-1}) - \left. \frac{\partial v}{\partial y} \right|_{i-1}$$

(6) Calculation of x derivatives

$$\left. \frac{\partial u}{\partial x} \right|_i = \text{PART } 2_i - \frac{v_i}{u_i} \left. \frac{\partial u}{\partial y} \right|_i$$

$$\left. \frac{\partial I}{\partial x} \right|_i = \text{PART } 3_i - \frac{v_i}{u_i} \left. \frac{\partial I}{\partial y} \right|_i$$

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

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

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

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

$$QQ_w = \frac{1}{778} \left\{ \frac{-\dot{q}_j + \frac{\tau_1^2}{\mu_i} \frac{\Delta y}{2}}{1 + \frac{\rho_1 v_1 \sigma}{\mu_1} \frac{\Delta y}{2}} \right\}$$

$$\tau_{w,L} = -\tau'_{Lj}$$

$$\tau_{w,T} = -\tau'_{Tj}$$

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

$$\left. \begin{aligned} I_{x+\Delta x} &= I_x + \left. \frac{\partial I}{\partial x} \right|_x \Delta x \\ u_{x+\Delta x} &= u_x + \left. \frac{\partial u}{\partial x} \right|_x \Delta x \end{aligned} \right\} \quad i = 1, 2, \dots, j$$

$$\left. \begin{aligned} u_i|_{x+\Delta x} &= u_j|_{x+\Delta x} \\ I_i|_{x+\Delta x} &= I_j|_{x+\Delta x} \end{aligned} \right\} \quad i = j + 1, j + 2, \dots, 200$$

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

$$\left| \left(\frac{\partial u}{\partial x} \right)_i - \frac{u_i}{x} \right| < .1 u_{LIST} \quad \text{for all } i$$

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

$$\left(\frac{\partial u}{\partial x} \right)_e = \frac{u_e}{u} \frac{\partial u}{\partial x}$$

$$\frac{\partial I_e}{\partial x} = 0$$

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

$$I_{i,S} = I_{i,S-\Delta S} + \left. \frac{\partial I}{\partial x} \right|_i \Delta S$$

$$u_{i,S} = \left[u_{i,S-\Delta S} + \left. \frac{\partial u}{\partial x} \right|_i \Delta S \right] \frac{x_o}{x_o + \Delta S}$$

If I_i or u_i become negative for some i , ΔS 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 ΔS 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.

ΔS is a pseudo length similar to Δx . The program also uses ΔS 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

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
1	1-48	8A6	Any information may be given on this card. It is used to identify the case.
2	1-10	7F10	Δx , 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	Δy , the increment in the y direction. Should be chosen to provide at least ten points in the boundary layer for accurate results.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
2	21-30	7F10	$\eta_{\max}$, the limiting value on the number of calculations in the y direction.
2	31-40	7F10	x_0 , the initial value of x. $x_0 > 0$.
2	41-50	7F10	x_f , the final value of x.
2	51-60	7F10	σ , the turbulent Prandtl number.
2	61-70	7F10	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.
3	1-10	4F10	ΔS . Calculations may be made for a blunt body. This involves calculating profiles near the stagnation point as well as along the body. This type of problem is indicated by $\Delta S > 0$ and MITR > 0 (card 4). For nonstagnation point calculations, $\Delta S = 0$.
3	11-20	4F10	Turbulent Prandtl number - must be equal to Prandtl number on card 2.
3	21-30	4F10	S_I , value of S used to initialize $S = S_I + \rho_e u_e^2 r^2 x_I$
3	41-50	4F10	x_I , value of x used to initialize $S = S_I + \rho_e u_e^2 r^2 x_I$

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.

4	1-5	1415	JOT1, the number of tabular values in the DJOT and XJOTT tables. $0 < JOT1 \leq 10$.
4	6-10	1415	LIST, the number of tabular values in the u and I profile tables. $2 \leq LIST \leq 50$.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
4	11-15	1415	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. $2 \leq NR \leq 50$.
4	16-20	1415	NP, the number of tabular values in the pressure table and the table of corresponding values of x. $2 \leq NP \leq 50$.
4	21-25	1415	NW, the number of tabular values in the wall enthalpy table and the table of corresponding values of x. $2 \leq NW \leq 50$.
4	26-30	1415	NDU, the number of tabular values in the $(\partial u / \partial y)_e$ table and the table of corresponding values of x. $2 \leq NDU \leq 50$.
4	31-35	1415	NV1, the number of tabular values of velocity normal to the surface and the table of corresponding values of x. $2 \leq NV1 \leq 50$.
4	36-40	1415	MITR, the maximum number of iterations allowed to calculate the profiles for the stagnation region.

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.

5 ²	-----	7F10	Table DJOT, assigns value to first, second, and successive printout intervals.
6	-----	7F10	Table XJOTT, assigns value of x up to which each interval of table DJOT is used.

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:

<u>Card</u>	<u>Column</u>	<u>Value</u>	
4	1-5	3	(right-adjusted)
5	1-10	.05	
	11-20	.01	
	21-30	.1	
6	1-10	.25	
	11-20	.3	
	21-30	1.0	

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
7	-----	7F10	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.
7	1-10	7F10	Velocity at wall.
7	11-20	7F10	Velocity at Δy from wall.
7	21-30	7F10	Velocity at $2\Delta y$ from wall.
8	-----	7F10	Table I-profile = $f(y)$. 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 Δy .
8	1-10	7F10	Total enthalpy at the wall.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
5	11-20	7F10	Total enthalpy at Δy from the wall.
6	21-30	7F10	Total enthalpy at $2\Delta y$ from the wall.
9	-----	7F10	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.
10	-----	7F10	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.
11	-----	7F10	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).
12	-----	7F10	Table pressure = $f(x)$. 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.
13	-----	7F10	Table x . This table contains NP tabular values of x corresponding to the entries made in the P table (card 12).
14	-----	7F10	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.
15	-----	7F10	Table x . This table contains NW tabular values of x corresponding to the entries made in the wall enthalpy table (card 14).
16	-----	7F10	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.

<u>Card</u>	<u>Columns</u>	<u>Format</u>	<u>Description</u>
17	-----	7F10	Table x. This table contains NDU tabular values of x corresponding to the entries made in the $(\partial u / \partial y)_e$ table (card 16).
18	-----	7F10	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.
19	-----	7F10	Table x. This table contains NV1 tabular values of x corresponding to the entries made in the V_w table (card 18).

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:

V_w table (card 14)	0.0	0.0	0.0
x table (card 15)	0.0	1.0	2.0

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.

If:			Then:
<u>LIST</u>	<u>CHECK</u>	<u>ERROR</u>	
0	0	---	Not dependent on the previous case. program continues.
0	1	0	Dependent on the previous case which worked. program continues, using final profile from previous case.
0	1	1	Dependent on the previous case which failed, so present case stops and program continues to next case.
0	---	0	Dependent on the previous case which worked, program continues, using final profile from previous case.
0	---	1	Dependent on the previous case which failed, so present case stops and program continues to next case.

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 :

y	u	$\dot{q}$	$\partial v / \partial y$	v	$\partial u / \partial x$
T	τ	I	η	$\partial I / \partial x$	ϵ

The following are printed as a function of the x at the end of each printout interval:

$$\dot{q}_w$$

$$\tau_{w,L}$$

$$\tau_{w,T}$$

$$\delta^*$$

$$\theta$$

$$\rho_1$$

$$\partial P / \partial x$$

$$QQW$$

$$S$$

c. NSBL Input Form

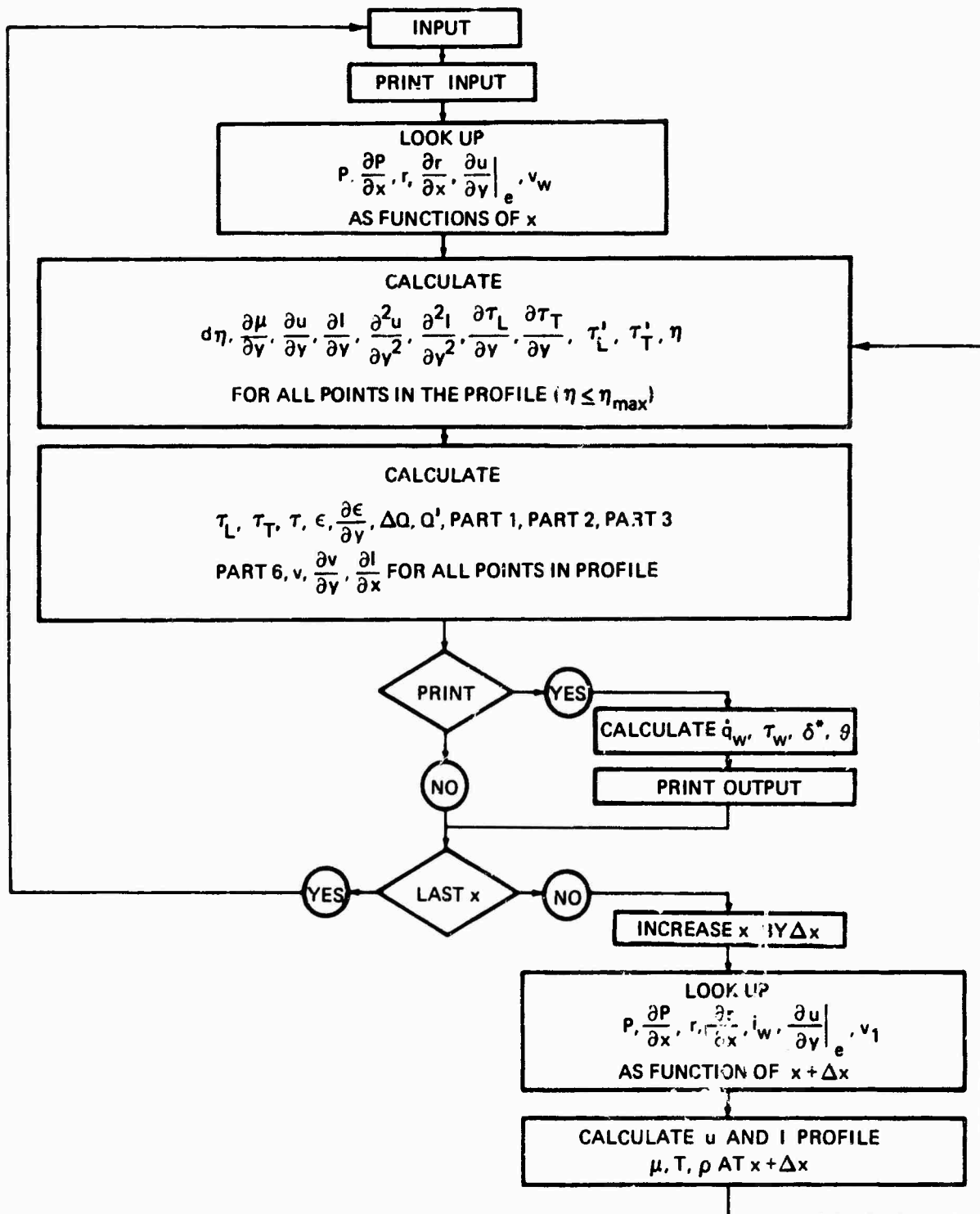
CARD

1	TITLE CARD FORMAT 8A6 ALPHANUMERIC ANY SPACE 1-48							
2	Δx		Δy		η_{max}		x_o	
3	ΔS		Prandtl no.		S_I		x_I	
4	JOT1	LIST	NR	NP	NW	NDU	NV1	MITR
5	DJOT	5	10	15	20	25	30	35
6	XJOTT							
7	u-profile							
8	l-profile							
9	r							
10	dr/dx							
11	x table							
12	Pressure							
13	x table							
14	Wall enthalpy							
15	x table							
16	$(du/dy)_e$							
17	x table							
18	v_w							
19	x table							

1 11 21 31 41 51 61 71

4. PROGRAMMING INFORMATION

a. Program Flow Chart



b. NSBL Program Listing

```

$IBFTC1A1781T DECK
C NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS
  DIMENSION Y(200),U(200),T(200),V(200),DUDY(200),D2UDY(200),DHDY(201),DHDY(201)
  10),D2HDY(200),DHDY(200),ETA(200),H(200),PT(50),XT(50),DPT(50),LDP(1781)T004
  28),LP(8),LV1(8),XV1T(50),V1T(50),RHO(200),XMU(200),DMUDY(200),
  3WALLH(50),XWH(50),LH(8),DJOT(10),XJOTT(10),DUDX(500),
  4DVDY(500),LR(8),RT(50),XRT(50),LDR(8),DRT(50),TITLE(8),LDU(8),DUT(1781)T007
  550),XDUT(50),USVE(200),HSVE(200),TSVE(200),XMUSVE(200),RHOSVE(200),
  6, TAU(200),TURBS(200)
  DIMENSION TAU(200),TAUTP(200),TAUTP(200),
  1TAUT(200),TAUL(200),EPS(200),EPSY(200),DQ(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(14I5)
    8 FORMAT(15HOPRESSURE TABLE/19X,1HX,12X,8HPRESSURE)
    9 FORMAT(20HOWALL-ENTHALPY TABLE/19X,1HX,7X,13HWALL-ENTHALPY)
   10 FORMAT(50H1NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS ,10X,
      18A6.7,3H0X=F8.5)
   11 FORMAT(1H0,8X,1HY,19X,1HU,19X,1HQ,18X,4HDVDY,17X,1HV,18X,4HDUDX/9X,
      1,1HT,18X,3HTAU,18X,1HH,18X,3HETA,16X,4HDHDX,17X,3HEPS//)
   12 FORMAT(2(6E20.6//))
   13 FORMAT(27H SOME T(I) IS LESS THAN 0.0)
   14 FORMAT(31H SOME U(I) IS LESS THAN (-U(J)))
   15 FORMAT(39H SOME H(I) IS LESS THAN OR EQUAL TO 0.0)
   16 FORMAT(1H0,11HTRY NUMBER 11,30H WILL BE MADE USING DX EQUALS F8.6/1781)T028
      1/)
   17 FORMAT(2F20.8)
   19 FORMAT(31H R FUNCTION TABLE LOOK-UP ERROR)
   200 FORMAT(25H DRDX TABLE LOOK-UP ERROR)
   201 FORMAT(50H1NON-SIMILAR BOUNDARY LAYER PROBLEM - IDEAL GAS ,10X,
      18A6)
   202 FORMAT( 34H0INPUT DATA FOR THE FOLLOWING CASE,/,77H (ANY QUANTITIES)
      15 NOT SHOWN HERE RETAIN THE VALUES USED IN THE PRECEDING CASE))
   203 FORMAT(15H0CASE CONSTANTS,/,5X,3HDX=E12.5,5X,3HDY=E12.5,6X,7HETAMA1781)T037
      1X=F7.3,/,1X,7HSTARTX=E12.5,3X,5HENDX=E12.5,10X,3HPR=F7.4)
   204 FORMAT(13H PRINT EVERY ,E12.5,9H UNTIL X=E12.5)
   205 FORMAT(8H0K TABLE/19X,1HX,19X,1HR,15X,5HDR/DX)
   206 FORMAT(28H0Q-WALL FOR THIS PROFILE IS E12.5/39H TAU-WALL (LAMINAR)1781)T041

```

```

1 FOR THIS PROFILE IS E12.5/41H TAU-WALL (TURBULENT) FOR THIS PROFILE 1781T042
2 LE IS E12.5/32H DELTA STAR FOR THIS PROFILE IS E12.5/ 1781T043
226H THE MOMENTUM THICKNESS IS E12.5/ 47H THE 1781T044
2 CONVERGENCE CRITERION WAS SATISFIED AFTER ,13.9H PROFILES) 1781T045
207 FORMAT(1H0,26X,65H*** THIS CASE WAS NOT RUN DUE TO DEPENDENCE ON P1781T046
1 PRECEDING CASE ***) 1781T047
208 FORMAT(3F20.5) 1781T048
209 FORMAT(1H0,11H DU/DY TABLE/19X,1HX,15X,5H DU/DY) 1781T049
210 FORMAT(1H ,24H DU/DY TABLE LOOK-UP ERROR) 1781T050
211 FORMAT(8A6) 1781T051
216 FORMAT(1H ,24HV(1) TABLE LOOK UP ERROR) 1781T052
217 FORMAT(1H0,10HV(1) TABLE/19X,1HX,16X,4HV(1)) 1781T053
1781T054
1781T055
1781T056
1781T057
1781T058
1781T059
1781T060
1781T061
1781T062
1781T063
1781T064
1781T065
1781T066
1781T067
1781T068
1781T069
1781T070
1781T071
1781T072
1781T073
1781T074
1781T075
1781T076
1781T077
1781T078
1781T079
1781T080
1781T081
1781T082
1781T083

C
C
C
PROGRAM CONSTANTS
DJO(1)=0.0
XJO(1)=0.0
CP = 6006.
DMUDY(1)=0.0
ETA(1)=0.

C
C
C
TABF CALLING SEQUENCES
***PRESSURE
LP(1)=XLLOC(LP(1))
LP(2)=XLLOC(XT(1))
LP(3)=XLLOC(PT(1))
LP(4)=1
LP(5)=1
LP(6)=1
LP(7)=50
C
C
C
***DPDX
LDP(1)=XLLOC(LDP(1))
LDP(2)=XLLOC(XT(1))
LDP(3)=XLLOC(DPT(1))
LDP(4)=1
LDP(5)=1
LDP(6)=1
LDP(7)=50

```

C	***R FUNCTION	1781T084
C		1781T085
C		1781T086
	LR(1)=XLOC(LR(1))	1781T087
	LR(2)=XLOC(XRT(1))	1781T088
	LR(3)=XLOC(RT(1))	1781T089
	LR(4)=1	1781T090
	LR(5)=1	1781T091
	LR(6)=1	1781T092
	LR(7)=50	1781T093
C	***DRDX	1781T094
C		1781T095
C		1781T096
	LDR(1)=XLOC(LDR(1))	1781T097
	LDR(2)=XLOC(XRT(1))	1781T098
	LDR(3)=XLOC(DRT(1))	1781T099
	LDR(4)=1	1781T100
	LDR(5)=1	1781T101
	LDR(6)=1	1781T102
	LDR(7)=50	1781T103
C	***WALL ENTHALPY	1781T104
C		1781T105
C		1781T106
	LH(1)=XLOC(LH(1))	1781T107
	LH(2)=XLOC(XWH(1))	1781T108
	LH(3)=XLOC(WALLH(1))	1781T109
	LH(4)=1	1781T110
	LH(5)=1	1781T111
	LH(6)=1	1781T112
	LH(7)=50	1781T113
C	***DUDX	1781T114
C		1781T115
C		1781T116
	LDU(1)=XLOC(LDU(1))	1781T117
	LDU(2)=XLOC(XDUT(1))	1781T118
	LDU(3)=XLOC(DUT(1))	1781T119
	LDU(4)=1	1781T120
	LDU(5)=1	1781T121
	LDU(6)=1	1781T122
	LDU(7)=50	1781T123
C		1781T124

C C

210

```

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

**V(1)
LV1(1)=XLOC(LV1(1))
LV1(2)=XLOC(XV1T(1))
LV1(3)=XLOC(V1T(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,SI,XI
X=STARTX
LENGTH=ENDX/DX+1.
LX=1
XJOT=STARTX
TIMES=0.0
NTRY=1
READ (5,7)JOT1,LIST,NR,NP,NW,NDU,NV1,MITR
KK=200
LIST4=4*LIST
IF(LIST)185,185,180
180 J=LIST4
185 JOT=JOT1+1
READ (5,1)(DJOT(I),I=2,JOT)
READ (5,1)(XJOTT(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)252,252,93
93 READ (5,1)(U(I),I=1,LIST)
READ (5,1)(H(I),I=1,LIST)
IF(NR)81,81,80
C
C
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
```



```

C      WRITE (6,17)(XVIT(I),VIT(I),I=1,NV1)
C      WRITE OUT VELOCITY AND ENTHALPY PROFILES
C
C      WRITE (6,218)(U(I),H(I),I=1,LIST)
218  FORMAT(31H0VELOCITY AND ENTHALPY PROFILES/19X,1HU,19X,1HH/(2F20.8)
1)
C
C      READ TABLES AND/OR SET UP INITIAL PROFILE
C      -----AS FAR AS READ IN
C
C      TEST CHECK AND ERROR***LIST IS POSITIVE
C      LOUP=0
C      ITTER=0
C      99 IF(CHECK)260,260,251
C      251 IF(ERROR)260,260,253
C
C      TEST ERROR***LIST IS ZERO
C
C      252 IF(ERROR)953,953,253
C      253 WRITE (6,207)
C      ERROR=1.
C      GO TO 18
C
C      260 ERROR=0.0
C      LP(8)=0
C      PRESS=TAB(X,LP(1))
C      IF(LP(8)-1)990,101,990
C
C      101 LDP(8)=0
C      DPDX=TAB(X,LDP(1))
C      IF(LDP(8)-1) 993,102,993
C
C      102 LR(8)=0
C      R=TAB(X,LR(1))
C      IF(LR(8)-1) 996,122,996
C
C      122 LDR(8)=0
C      DRDX=TAB(X,LDR(1))
C      IF(LDR(8)-1) 997,89,997
C
C      89 LDU(8)=0
C      DUDYL=TAB(X,LDU(1))
C      IF(LDU(8)-1) 998,87,998
C
C      87 LV1(8)=0
C      V(1)=TAB(X,LV1(1))
C      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
   YI=I-1
   Y(I)=YI*DY
   IF(I-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.)/CP
   XMU(I)=2.272E-08*T(I)**1.5/(T(I)+198.6)
   RHO(I)=PRESS/(1716.*T(I))
95 CONTINUE

C
C
C   COMPUTE INITIAL VALUE OF S
C
C   S=SI+RHO(J)*U(J)*XMU(J)*R**2*X I
C
C   LOOP TO SAVE INITIAL PROFILES
C
C
953 PRESVE=PRESS
   DPDXSV=DPDX
   RSVE=R
   DRDXSV=DRDX
   DUDYSV=DUDYL
   SSVE=S
   VLSVE=V(I)
DO 954 I=1,200
   USVE(I)=U(I)
   HSVE(I)=H(I)
   TSVE(I)=T(I)
   XMUSVE(I)=XMU(I)
   RHOSVE(I)=RHO(I)
954 CONTINUE

C
C
C   CALCULATE Y DERIVATIVES, V PROFILE, DVDY
C
C
C   NLOOP=0
C   NDO=ETAMAX
C   NDOI=NDO
29 DELETEA=U(J)*R*DY*XMU(J)**.6/S**.8
   NI=0
   NJ=0
   DVDY(I)=V(I)*(T(2)/T(I)-1.)/DY

```

```

1781T250
1781T251
1781T252
1781T253
1781T254
1781T255
1781T256
1781T257
1781T258
1781T259
1781T260
1781T261
1781T262
1781T263
1781T264
1781T265
1781T266
1781T267
1781T268
1781T269
1781T270
1781T271
1781T272
1781T273
1781T274
1781T275
1781T276
1781T277
1781T278
1781T279
1781T280
1781T281
1781T282
1781T283
1781T284
1781T285
1781T286
1781T287
1781T288
1781T289
1781T290

```

```

LOOP=1
TAULY(1)=DPDX
TAUTY(1)=0.
TAULP(1)=0.
TAUTP(1)=0.
DQ(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(I)*Y(I)/XMU(I))*833*(U(I)/U(J))*12
1*(T(J)/T(I))**4
DMUDY(I)=(XMU(I+1)-XMU(I-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)+TAULY(I))*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
C
ERROR CRITERIA
KK=I
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

```

```

1781T291
1781T292
1781T293
1781T294
1781T295
1781T296
1781T297
1781T298
1781T299
1781T300
1781T301
1781T302
1781T303
1781T304
1781T305
1781T306
1781T307
1781T308
1781T309
1781T310
1781T311
1781T312
1781T313
1781T314
1781T315
1781T316
1781T317
1781T318
1781T319
1781T320
1781T321
1781T322
1781T323
1781T324
1781T325
1781T326
1781T327
1781T328
1781T329
1781T330
1781T331
1781T332

```

```

435 K=LIST
J=LIST4
GO TO 49
C
C
C
    PROFILE SUCCESS CRITERIA
44 CONTINUE
  IF(NLOOP-1)40,640,45
640 IF(NI)641,641,5044
641 IF((1.-U(I-1))/U(I))-DUDYL*DY/U(I))-0.00001)647,647,5044
647 NI=1
5044 IF(NJ)642,642,5045
642 IF((1.-H(I-1))/H(I))-0.00005)643,643,5045
643 NJ=1
5045 NIJ=NI*NJ
  IF(NIJ)40,40,644
644 NDO1=I+5
  NLOOP=2
GO TO 40
45 CONTINUE
  IF(I-NDO1)40,646,646
646 CONTINUE
  K=I
  J=I
GO TO 49
40 CONTINUE
  J=NDO
  K=NDO
C
C
C
    CALCULATE X DERIVATIVES
49 CONTINUE
  NLOOP=1
DO 4049 I=2,J
  TAU(I)=TAUTP(I)-TAUTP(J)
  TAUL(I)=TAULP(I)-TAULP(J)
  TAU(I)=TAUL(I)+TAUT(I)
4049 CONTINUE
  TAU(I)=-TAUTP(J)-TAULP(J)
DO 3049 I=2,J
  EPS(I)=TAUT(I)/DUDY(I)

```

```

1781T333
1781T334
1781T335
1781T336
1781T337
1781T338
1781T339
1781T340
1781T341
1781T342
1781T343
1781T344
1781T345
1781T346
1781T347
1781T348
1781T349
1781T350
1781T351
1781T352
1781T353
1781T354
1781T355
1781T356
1781T357
1781T358
1781T359
1781T360
1781T361
1781T362
1781T363
1781T364
1781T365
1781T366
1781T367
1781T368
1781T369
1781T370
1781T371
1781T372
1781T373

```

```

EPSY(I)=(TURBS(I)*TAULY(I)-EPS(I)*D2UDY(I))/DUDY(I)
DQ(I)=1./PR*((XMU(I)+EPS(I))*D2HDY(I)+(DMUDY(I)+EPSY(I))*
1DH DY(I)-(U(I+1)*TAU(I+1)-U(I-1)*TAU(I-1))/2./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)*TAU(I+1)-U(I-1)*TAU(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/
1H(I)-U(I)**2/2.)*(PART3-U(I)*PART2)-PART6)
DVDY(I)=2./DY*(V(I)-V(I-1))-DVDY(I-1)
DUDX(I)=PART2-V(I)/U(I)*DUDY(I)
DHDX(I)=PART3-V(I)/U(I)*DH DY(I)
3049 CONTINUE
IF(DS)50,50,3
3 IF(X-STARTX)300,300,50
300 I=1
4 IF (ABS(DUDX(I)-U(I)/X)-.1*U(LIST)) 24, 24, 25
24 I=I+1
IF (I-LIST) 4, 4, 50
25 DO 21 I=1,LIST
H(I)=H(I)+DHDX(I)*DS
U(I)=(U(I)+DUDX(I)*DS)*(STARTX/(STARTX+DS))
IF (H(I)) 70,70,71
70 DS=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 ENCOUNTER
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 ENCOUNTER
1ED, WE WILL RUN AGAIN USING DS= F8.6)
GO TO 1003

```

```

21 CONTINUE
  ITTER=ITTER+1
  IF (ITTER-MITR) 88, 88, 23
  C ***** STATEMENT BELOW IS IN ERROR *****
  23 WRITE (6,1005)MITR
  1005 FORMAT(7H AFTER ,I3,94H PROFILES,THE QUANTITY (DUDX-U/X) IS GREATER
    1R IN ABSOLUTE MAGNITUDE THAN ONE PERCENT OF U(LIST))
    GO TO 999
  C
  C PROFILE COMPLETED
  C
  50 IF(X-XJOT+DX/2.) 108,108,51
  108 IF(X-ENDX+DX/2.) 22,22,51
  C
  C COMPUTE Q-WALL,TAU-WALL AND DELTA STAR
  C
  51 CONTINUE
  QQW = 0.
  DELSTR=0.
  THETA=0.
  Q(1)=(-QP(J)+U(2)*TAU(2)/2.)/(1.+RHO(1)*V(1))*PR
  1/XMU(1)*DY/2.)/778.
  DO 3150 I=2,J
  Q(I)=(QP(I)-QP(J))/778.
  3150 CONTINUE
  DO 150 I=2,K
  DQQW = (RHO(I)*U(I)*DHDX(I)+RHO(I-1)*U(I-1)*DHDX(I-1)+
  1RHO(I)*V(I)*DHDX(I)+RHO(I-1)*V(I-1)*DHDX(I-1))*0.5-
  2((U(I)*TAU(I)-U(I-1)*TAU(I-1))/DY)
  QQW = QQW-DQQW
  DELSTR=DELSTR+1.-((RHO(I)*U(I)+RHO(I-1)*U(I-1))/(2.*RHO(J)*U(J))
  THETA=THETA+(RHO(I)*U(I)
    *(1.-U(I)/U(J))
  1+RHO(I-1)*U(I-1)*(1.-U(I-1)/U(J)))/RHO(J)/U(J)
  150 CONTINUE
  QDOTW=Q(1)
  QW=QDOTW
  QQW = QQW*DY/778.
  TAUW=-TAUTP(K)-TAULP(K)
  TAUW=-TAULP(K)
  TAUW=-TAUTP(K)
  DELSTR=DELSTR*DY

```

```

1781T416
1781T417
1781T418
1781T419
1781T420
1781T421
1781T422
1781T423
1781T424
1781T425
1781T426
1781T427
1781T428
1781T429
1781T430
1781T431
1781T432
1781T433
1781T434
1781T435
1781T436
1781T437
1781T438
1781T439
1781T440
1781T441
1781T442
1781T443
1781T444
1781T445
1781T446
1781T447
1781T448
1781T449
1781T450
1781T451
1781T452
1781T453
1781T454
1781T455
1781T456

```

```

C      THETA=THETA*DY/2.
C      PRINT VARIABLES
C
55  WRITE (6,10)(TITLE(I),I=1,8),X
    WRITE (6,11)
    WRITE (6,12)(Y(I),U(I),Q(I),DV(DY(I),V(I),DUDX(I),T(I),TAU(I),H(I),
1    ,ETA(I),DMDX(I),EPS(I),I=1,K)
    WRITE (6,206)      QW,TAULW,TAUTW,DELSTR,THETA,ITIER
    WRITE (6,2006)RHO(LIST),DPDX
2006  FORMAT(7H RHO = E14.8,5X,7HDPDX = E14.8)
    WRITE (6,2060)QQW
2060  FORMAT(25H QQW FOR THIS PROFILE IS E12.5)
    WRITE(6,2061)S
2061  FORMAT(5H S = ,E12.5)
52  IF(X-XJOT(LX)+DX/2.) 52,52,53
52  XJOT=XJOT+DJOT(LX)
    GO TO 54
53  LX=LX+1
    XJOT=XJOT+DJOT(LX)
54  IF(X-ENDX+DX/2.) 22,22,18
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

```

```

17811457
17811458
17811459
17811460
17811461
17811462
17811463
17811464
17811465
17811466
17811467
17811468
17811469
17811470
17811471
17811472
17811473
17811474
17811475
17811476
17811477
17811478
17811479
17811480
17811481
17811482
17811483
17811484
17811485
17811486
17811487
17811488
17811489
17811490
17811491
17811492
17811493
17811494
17811495
17811496
17811497
17811498

```



```

996 WRITE (6,19)
    GO TO 999
997 WRITE (6,200)
    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),Q(I),DVDY(I),V(I),DUDX(I),T(I),TAU(I),H(I),
1,ETA(I),DHDX(I),EPS(I),I=1,KK)
1007 ERROR=1.0
    GO TO 18
1000 WRITE (6,10)(TITLE(I),I=1,8),X
    WRITE (6,11)
    WRITE (6,12)(Y(I),U(I),Q(I),DVDY(I),V(I),DUDX(I),T(I),TAU(I),H(I),
1,ETA(I),DHDX(I),EPS(I),I=1,KK)
C
C
C    DECREASE DX AND RESTORE INITIAL PROFILES
    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
C
C    LOOP TO RESTORE INITIAL PROFILES
1002 J=LIST4
1003 PRESS=PRESVE
    DPDX=DPDXSV
    R=RSVE
    DRDX=DRDXSV
    DUDYL=DUDYSV
    S=SSVE
    V(1)=V1SVE
    DO 1004 I=1,200
        U(I)=USVE(I)

```

```

1781T540
1781T541
1781T542
1781T543
1781T544
1781T545
1781T546
1781T547
1781T548
1781T549
1781T550
1781T551
1781T552
1781T553
1781T554
1781T555
1781T556
1781T557
1781T558
1781T559
1781T560
1781T561
1781T562
1781T563
1781T564
1781T565
1781T566
1781T567
1781T568
1781T569
1781T570
1781T571
1781T572
1781T573
1781T574
1781T575
1781T576
1781T577
1781T578
1781T579
1781T580
1781T581

```

```

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

```

```

000588

```

REFERENCES

1. Hanks, R. A. : and Savage, R. T. : Thermal Design Methods for Recoverable Launch Vehicles with Consideration of Arbitrary Wall Temperature and Surface Conditions, NASA CR-74714, August 1965.
2. Nagel, A. L. : Fitzsimmons, H. D. ; and Doyle, L. B. : Analysis of Hypersonic Pressure and Heat Transfer Tests on Delta Wings with Laminar and Turbulent Boundary Layers, NASA CR-535, August 1966.
3. Thomas, A. C. : Perlbachs, A. ; and Nagel, A. L. : Advanced Reentry Systems Heat Transfer Manual for Hypersonic Flight, AFFDL-TR-65-195, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, January 1966.
4. Hansen, C. F. : Approximations for the Thermodynamic and Transport Properties of High Temperature Air, NACA TN-4150, March 1958.
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.
6. Feldman, S. : Hypersonic Gas Dynamic Charts for Equilibrium Air, AVCO Research Report No. 40, January 1957.
7. Logan, J. ; and Treanor, C. : Tables of Thermodynamic Properties of Air from 3,000 to 10,000 °K, Cornell Aeronautical Laboratory, Inc., January 1957.
8. Ames Research Staff: Equations, Tables, and Charts for Compressible Flow, NACA Report 1135, 1953.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D		
(Security Classification of title, body of abstract and indexing annotation must be entered when the overall report is classified).		
1. ORIGINATING ACTIVITY (Corporate author) The Boeing Company Aerospace Group Space Division— Kent Facility		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE 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		
4. DESCRIPTIVE NOTES (Types of report and inclusive dates) Final Report, March 1965 to March 1967		
5. AUTHORS (First name, middle initial, last name) Savage, Richard T. Jaek, Carl L. Mitchell, James R.		
6. REPORT DATE December 1967	7. TOTAL NO. OF PAGES 222	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. AF33(615)-2372 b. Project No. 1366 c. Task No. 136607 d. BPSN 5(61136-62405334)	9a. ORIGINATOR'S REPORT NUMBERS D2-113531-3 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFFDL-TR-67-144, Volume III	
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	
13. ABSTRACT 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.		

UNCLASSIFIED

Security Classification

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Nonsimilar Boundary Layer Momentum Integral Aerodynamic Heating Laminar Flow Turbulent Flow						

UNCLASSIFIED

Security Classification