THE UNIVERSITY OF TEXAS AT AUSTIN

NASA CR-
151658

# THE ANALYSIS OF A NONSIMILAR LAMINAR BOUNDARY LAYER

Dennis D. Stalmach and
John J. Bertin

(NASA-CR-151658) THE ANALYSIS OF A NONSIMILAR LAMINAR BOUNDARY LAYER (Texas Univ.) 121 p HC A06/MF A01 CSCL 01A

N78-20066

Unclas
G3/02 11814



Aerospace Engineering Report 78001

This work was supported by
the Johnson Space Center
through NASA Contract NAS 9-13680

February 1978

Department of Aerospace Engineering and Engineering Mechanics

# THE ANALYSIS OF A NONSIMILAR LAMINAR BOUNDARY LAYER*

by

Dennis D. Stalmach
and
John J. Bertin

AEROSPACE ENGINEERING REPORT 78001

*This work was supported by the
Johnson Space Flight Center
through Contract NAS 9-13680

Department of Aerospace Engineering
and Engineering Mechanics

The University of Texas at Austin

February 1978

## ACKNOWLEDGEMENTS

This research was supported by the Johnson Space Flight Center (NASA) through contract NAS 9-13680. The authors acknowledge the considerable efforts of Dr. W. D. Goodrich in providing technical assistance and guidance to the effort. The authors would also like to thank Mr. Paul Wing for skillfully typing the manuscript.

# ABSTRACT

A computer code is described which yields accurate solutions for a broad range of laminar, nonsimilar boundary layers, providing the inviscid flow field is known. The boundary layer may be subject to mass injection for perfect-gas, nonreacting flows. If no mass injection is present, the code can be used with either perfect-gas or real-gas thermodynamic models. Solutions, ranging from two-dimensional similarity solutions to solutions for the boundary layer on the Space Shuttle Orbiter during reentry conditions, have been obtained with the code. Comparisons of these solutions, and others, with solutions presented in the literature; and with solutions obtained from other codes, demonstrate the accuracy of the present code.

TABLE OF CONTENTS

# INTRODUCTION

Calculation of the convective heat-transfer-rate distribution for a given configuration requires information about the character of the viscous boundary layer. The present work discusses a numerical procedure which provides solutions of a thin, laminar boundary layer bounded by a specified inviscid flow field. The inviscid flow field is required as input to the present code. This code can be used as a tool for analysis of data from wind tunnels and for the extrapolation of the correlations to flight conditions. Furthermore, it will serve as an instructional aid for courses in boundary layer theory and convective heat transfer at the University of Texas at Austin.

For compressible flow problems, where the wall temperature is constant and the static pressure at the edge of the boundary layer is constant, the convective heat-transfer rate can be calculated using the Eckert reference temperature relations [1]. The heating rates calculated using this method, which does not require the complete solution of the viscous equations, compare favorably with experimental measurements even for flows with moderate, favorable pressure gradients [2]. However, this technique cannot be applied for numerous problems of interest, e.g., those involving gas injection at the surface, nonuniform surface temperatures, real gas effects, etc.

A more rigorous calculation of the heat transfer rates could be obtained from a solution of the viscous equations, assuming the similarity conditions are satisfied [3]. As noted by Hayes and Probstein [4], these conditions are satisfied in the following cases:

(a) for flow with a uniform free stream (i.e., constant pressure solutions), and a constant surface temperature,

(b) near a stagnation point, if the fluid properties in the stagnation region are assumed approximately constant, and

(c) for the limiting case of locally hypersonic flow, where $\frac{u_e^2}{H_e} \to 2$.

For other flow problems, approximate solutions can be obtained, if the flow is assumed to be locally similar. Special cases involving mass injection can also be treated assuming similarity. One such example would be:

$$(\rho v)_{inj} \simeq \frac{\text{const}}{\sqrt{x}}$$

where the injectant is the same gas as the free stream.

The procedures described above are still too limited to be of use in analyzing heat transfer data from many experimental programs. With these requirements in mind, Bertin and Byrd [5] developed a finite difference scheme (NONSIMBL) to obtain solutions for a laminar boundary layer for a flow which is:

(a) either compressible or incompressible,

(b) possibly subject to a pressure gradient in the external flow,

(c) either two-dimensional or axisymmetric,

(d) of arbitrary free-stream gas,

(e) possibly subjected to an arbitrary injectant, injection rate, and injection-rate distribution, and

(f) bounded by an arbitrary wall temperature distribution.

The original code was limited to a mixture of perfect-gas, nonreacting flows having a constant stagnation temperature. Their solution technique closely follows that developed by Marvin and Sheaffer [6].

The NONSIMBL code was later modified to handle real-gas flows as well as perfect-gas flows for either variable entropy or isentropic flow conditions at the edge of the boundary layer. It has been used extensively with good success but the computed solutions exhibited problems in certain applications, specifically accelerating flow past a highly cooled body [7]. The problems which occurred in these cases were due to the relation used to evaluate the transformed y-coordinate,

$$n = 1 - e^{-\alpha\eta}$$

where $\eta$ is the transformed y-coordinate using the standard Lees-Dorodnitsyn transformation and $\alpha$ is a scale factor which was assumed constant. This transformation allowed numerical integrations to be carried out over a fixed interval (zero to one) rather than the usual interval in the $\eta$-coordinate system (zero to infinity). Therefore, the need for an iteration to define the boundary layer edge was elminated. However, the coding procedure also eliminated the iterative procedure which required that zero velocity and temperature gradients exist at the edge of the boundary layer. Note that if the scale factor $\alpha$ is assumed a constant for every station, as it was in [5], there is a unique relation between $\eta$ and n. As a result, $\eta_{edge}$ is forced to be a constant. However, the Faulkner-Skan solution shows that, for a boundary layer in the presence of a given pressure gradient, $\eta_{edge}$ is not a constant but depends on the value of the pressure gradient parameter $\beta$ [8]. For small streamwise variations of $\beta$, a constant value of the scale factor is not a bad assumption, but for accelerating flows past highly cooled walls, such as in [7], it is unrealistic.

For flow conditions such as those discussed in [7], the solutions generated by the NONSIMBL code had relatively large velocity and temperature gradients at the boundary layer edge due to the constant $\alpha$ assumption. Although these gradients had only a small effect on the heat transfer and skin friction (which were of primary importance at that time) they had a significant effect on the temperature and the velocity distributions. Subsequent applications have required that the code be capable of providing accurate calculations of displacement and momentum thicknesses which are very sensitive to the velocity and temperature profiles. The present code includes an iterative procedure in which $\alpha$ is varied so that the velocity gradient at the edge of the boundary layer is essentially zero. There is still no restriction on the temperature gradient at the edge of the boundary layer, but for most cases computed to date it is small.

Another refinement added to the code removes the need to input velocity and temperature profiles at the initial station. The code now calculates the initial profile using a Faulkner-Skan solution for a perfect gas with $Pr = 0.7$. The initial guesses for $f''(0)$ and $g'(0)$ needed to obtain the Faulkner-Skan solution are calculated using a correlation of these parameters as functions of $g(0)$ and $\beta$. Thus, a large range of flow conditions can be solved with a minimum of external calculations.

The modified NONSIMBL code which is called NSBL, i coded in FORTRAN IV for the CDC 6600 at the University of Texas at Austin and is described herein along with a discussion of its uses and limitations.

## NOMENCLATURE

| | |
|---|---|
| $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ | matrices, equation 31 |
| C | Chapman-Rubesin factor $\frac{\rho\mu}{\rho_e\mu_e}$ |
| $C_f$ | local skin friction coefficient |
| $C_i$ | mass fraction for species i |
| $C_{pi}$ | specific heat of species i |
| $\bar{C}_p$ | specific heat, $\sum_i C_i C_{pi}$ |
| D* | displacement thickness including mass injection, equation 42 |
| $D_i$ | diffusion coefficient for species i |
| F | dimensionless streamwise component of the local velocity $\frac{u}{u_e}$ |
| f | stream function, equation 7 |
| $\tilde{G}$ | matrix, equation 35 |
| $\tilde{H}$ | matrix, equation 34 |
| H | total enthalpy |
| i | mesh point coordinate normal to the wall |
| k | thermal conductivity |
| L | length of the Space Shuttle Orbiter |

| | |
|---|---|
| M | Mach number |
| $M_i$ | molecular weight of species i |
| m | mesh point coordinate in streamwise direction |
| N | number of mesh points normal to the wall |
| n | transformed y-coordinate, equation 11 |
| p | pressure |
| Pr | Prandtl number, $\frac{\mu \bar{C}_p}{k}$ |
| $\dot{q}$ | local convective heat-transfer rate |
| R | universal gas constant |
| r | distance from surface of body to axis of symmetry, measured normal to the axis of symmetry |
| s | transformed x-coordinate, equation 6 |
| Sc | Schmidt number, $\frac{\mu}{\rho D_{12}}$ |
| St | Stanton number, $\frac{\dot{q}}{\rho_e u_e C_p (T_r - T_w)}$ |
| T | temperature |
| u | velocity in streamwise direction |
| v | velocity normal to the wall |
| x | physical streamwise wetted distance from the stagnation point |
| $x_a$ | axial distance from the nose of the Space Shuttle Orbiter |
| y | physical distance normal to the wall |

| | |
|---|---|
| $\alpha$ | scale factor, equation 11 |
| $\beta$ | pressure gradient parameter, $\frac{2s}{u_e}\frac{du_e}{ds}$ |
| $\delta^*$ | displacement thickness |
| $\eta$ | transformed y-coordinate, equation 6 |
| $\eta_{edge}$ | that point in the boundary layer where $u = 0.99u_e$ |
| $\theta$ | non-dimensional temperature, $\frac{T}{T_{te}}$ |
| $\theta$ | momentum thickness, equation 48 |
| $\mu$ | viscosity |
| $\rho$ | density |
| $\underset{\sim}{\omega}$ | column matrix representing the unknowns, equation 31 |
| $\tau$ | local skin friction |
| $\psi$ | stream function, equation 7 |

Subscripts

| | |
|---|---|
| a,b,c,d,e,f | mesh points, sketch 1 |
| e | edge value |
| i | species i |
| inj | properties of the injected species |
| N | evaluated at boundary layer edge when used with matrices only |
| r | denotes recovery temperature |
| t | stagnation value |
| w | wall value |

| | |
|---|---|
| 1 | properties of stream species |
| 1 | evaluated at the wall (which is the first node) when used with matrices only |
| 2 | injectant species |

Superscripts

| | |
|---|---|
| k | body geometry factor, k=0, for two-dimensional flow; and k=1 for axisymmetric flow |

# THEORETICAL ANALYSIS

## Governing Equations

Solutions are sought for the laminar boundary layer for an axisymmetric or a two-dimensional configuration with possible ablation or transpiration cooling. To model this flow, the present analysis is not restricted to the requirements for a similar, laminar boundary layer with mass injection. The injected gas may differ from the stream species and a general injection-rate distribution may be specified. The body may be either axisymmetric or two-dimensional providing the radius of curvature is large in comparison to the boundary layer thickness, i.e., centrifugal forces are neglected. Approximate solutions for a three-dimensional boundary layer with small cross flow can be obtained using the axisymmetric analog [9] in which an effective radius of curvature is used to describe the streamline divergence. For flow with no mass injection at the wall, the thermodynamic properties of the free-stream gas may be modeled with the ideal gas relations [10] or with real gas properties using the thermodynamic subroutine, "MOLIER." For flows with mass injection, the thermodynamic properties of the mixture of injectant and stream gases are approximated with the ideal gas relations. Chemical reactions between the species are not considered. The governing equations applicable to the flow model are as follows:

$$\text{continuity:} \quad \frac{\partial(\rho u r^k)}{\partial x} + \frac{\partial(\rho v r^k)}{\partial y} = 0 \tag{1}$$

where $k=1$ for axisymmetric flows and $k=0$ for two-dimensional flow.

$$\text{species:} \quad \rho u \frac{\partial c_i}{\partial x} + \rho v \frac{\partial c_i}{\partial y} = \frac{\partial}{\partial y}\left(\rho\, D_i \frac{\partial c_i}{\partial y}\right) \tag{2}$$

x-component of momentum:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = - \frac{\partial p_e}{\partial x} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \tag{3}$$

y-component of momentum:

$$\frac{\partial p}{\partial y} \simeq 0 \tag{4}$$

which represents the standard boundary-layer assumption regarding the pressure gradients normal to the wall.

energy:

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = u \frac{dp_e}{dx} + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y} + \rho\, D_i \sum_i h_i \frac{\partial c_i}{\partial y}\right) + \mu \left(\frac{\partial u}{\partial y}\right)^2 \tag{5}$$

The governing equations which describe the nonsimilar, possibly compressible, flow in physical coordinates are nonlinear, partial differential equations. Therefore, a transformation is sought to simplify the solution procedures. Using the standard Lee-Dorodnitsyn coordinate transformation [11]:

$$s = \int_o^x \rho_e \mu_e u_e r^{2k} dx \qquad (6)$$

$$\eta = \frac{\rho_e u_e r^k}{\sqrt{2s}} \int_o^y \frac{\rho}{\rho_e} dy$$

where the stream function is defined as:

$$\psi = (2s)^{0.5} f \qquad (7)$$

one obtains the following equations:

species:

$$\left(\frac{C}{Sc} C_1'\right)' + f C_1' = 2s \left(C_1^s f' - f^s C_1'\right) \qquad (8)$$

x- ɔmentum:

$$ff'' + (Cf'')' + \beta\left(\frac{\rho_e}{\rho} - (f')^2\right) = 2s \left(f'(f')^s - f^s f''\right) \qquad (9)$$

energy:

$$f\theta' + \frac{C_{1e}}{\bar{\rho}} \frac{C}{Sc} \left(C_{p_1} - C_{p_2}\right)\theta' C_1' + \frac{Cu_e^2}{\bar{C}_p T_{te}} (f'')^2 + \frac{1}{\bar{C}_p} \left(\frac{C}{Pr} \bar{C}_p \theta'\right)'$$

$$- \frac{\rho_e u_e^2 \beta f'}{\rho \bar{C}_p T_{te}} = 2s\,(f'\theta^s - \theta' f^s) \qquad (10)$$

where superscript prime denotes differentiation with respect to $\eta$ and superscript $s$, with respect to $s$. Having introduced the stream function, which automatically satisfies the overall continuity equation, the new set of governing equations contains one less dependent variable. Except for a limited number of injection-rate distributions, it is not possible to reduce the governing equations to a form analogous for similar solutions, as discussed in the Introduction. An additional coordinate transformation can be made, as suggested in [12] and as mentioned in the Introduction:

$$n = 1 - e^{-\alpha\eta} \tag{11}$$

This transformation is for numerical purposes. Numerical integrations can now be carried out over a fixed interval (zero to one) rather than the usual interval in the $\eta$-coordinate system (zero to infinity). This coordinate system eliminates the need for an iteration to define the boundary layer edge. Note that in the present approach it is assumed that the edge of the viscous boundary layer, the edge of the thermal boundary layer, and the edge of the species concentration layer, all occur at the same $\eta$.

Also, the transformation affects nodal point spacing in the physical-coordinate plane. Points which are evenly spaced with respect to the n-coordinate are not evenly spaced in physical space. Spacing of the y-coordinates of the nodal locations varies with position, such that $\Delta y$ increases with distance from the wall.

This results in more nodal points in the region near the wall, where gradients are large, and fewer points in the region away from the wall, where gradients are smaller. The scale factor $\alpha$ will be treated as a constant for any two adjacent streamwise stations. The resultant governing equations in the transformed system with $F = \frac{u}{u_e}$ replacing $f'$, are as follows.

species:

$$\alpha^2(1-n)^2 \left(\frac{C}{Sc}\right)_n C_{1n} + \alpha^2(1-n) \left((1-n)\, C_{1nn} - C_{1n}\right) \frac{C}{Sc} + \alpha(1-n)\, f\, C_{1n}$$

$$= 2s \left(C_{1s}F - \alpha(1-n)\, F_s\, C_{1n}\right) \tag{12}$$

momentum:

$$fF_n\, \alpha(1-n) + \alpha^2(1-n)^2\, C_nF_n + C\alpha^2(1-n) \left((1-n)\, F_{nn} - F_n\right)$$

$$+ \beta\, \frac{\rho_e}{\rho} - F^2 = 2s \left(FF_s - f_sF_n\, \alpha(1-n)\right) \tag{13}$$

energy:

$$\frac{1}{\bar{C}_p}\, \alpha^2(1-n)^2 \left(\frac{C\bar{C}_p}{Pr}\right)_n \theta_n + \left(\frac{C}{Pr}\right)\alpha^2(1-n) \left((1-n)\theta_{nn} - \theta_n\right) + \alpha(1-n)\, f\, \theta_n$$

$$+ \frac{C_{1e}}{\bar{C}_p}\, \frac{C}{Sc} \left(C_{p_1} - C_{p_2}\right)\alpha^2(1-n)^2\, \theta_nC_{1n} + \frac{Cu_e^{\,2}}{\bar{C}_pT_{te}} F_n^2\, \alpha^2(1-n)^2$$

$$- \frac{u_e^{\,2}}{\bar{C}_pT_{te}}\, \beta\, \frac{\rho_e}{\rho}\, F = 2s \left(F\theta_s - \alpha(1-n)\, \theta_nf_s\right) \tag{14}$$

where the subscripts $n$ and $s$ denote differentiation with respect to $n$ and $s$, respectively.

## Boundary Conditions

In the previous section, the governing equations were written in terms of three dimensionless, dependent variables, $F$, $C_1$, and $\theta$. Since the surface temperature and the inviscid flow field are known a priori, values for $F$ and $\theta$ are immediately determined at both boundaries.

At the wall, i.e., $n = 0$

$$F = 0, \ \theta = T_w/T_{te} \tag{15}$$

At the boundary layer edge, i.e., $n = 1$

$$F = 1, \ \theta = T_e/T_{te} \tag{16}$$

Also, it is possible to readily define the species concentrations at $n = 1$

$$C_1 = 1, \ C_2 = 0 \tag{17}$$

The values of the concentration fractions at the wall, however, are not explicitly known. The appropriate wall boundary condition is obtained from the conservation of species equation:

$$(\rho v)_w \ C_{1w} = \rho_w \ D_{12} \left(\frac{\partial C_1}{\partial y}\right)_w \tag{18}$$

Physically, this boundary condition specifies the balance between the convective flux away from the wall and the diffusive flux toward the surface for the free-stream species. Written for the n,s coordinate system in terms of nondimensional parameters, this boundary condition becomes

$$C_{1nw} = \frac{(\rho v)_w \, C_{1w} \, Sc_w \sqrt{2s}}{\alpha \, \rho_w \, \mu_w \, u_e \, r^k} \tag{19}$$

In addition to the dependent variables $C_1$, $\theta$, and $F$, the governing equations also contain the stream function, f. By definition:

$$f' \equiv F$$

$$f = \int_0^n \frac{F}{\alpha(1-n)} \, dn + f(0) \tag{20}$$

From the continuity equation it follows that

$$f(0) = -\frac{1}{\sqrt{2s}} \int_0^x (\rho v)_w \, r^k \, dx \tag{21}$$

In summary, the boundary conditions for $C_1$ and $f(0)$ at $n = 0$ are determined by substituting the specified injection-rate distribution into the appropriate equations. The other boundary conditions are determined from the usual physical constraints. A complete listing of the boundary conditions is as follows:

At the wall, n = 0:

$$F = 0$$

$$\theta = \frac{T_w}{T_{te}}$$

$$C_{1n} = \frac{(\rho v)_w \, C_{1w} \, Sc_w \sqrt{2s}}{\alpha \, \rho_w \, \mu_w \, u_e \, r^k}$$

$$f(0) = -\frac{1}{\sqrt{2s}} \int_0^x (\rho v)_w \, r^k \, dx$$

At the boundary layer edge, n = 1:

$$F = 1$$

$$\theta = \frac{T_e}{T_{te}}$$

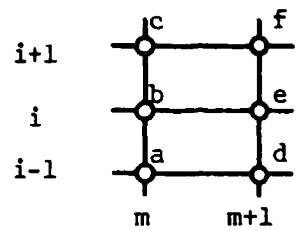
$$C_1 = 1$$

$$C_2 = 0$$

## Formulation of the Solution Algorithm

The governing partial differential equations, and the appropriate boundary conditions have been developed. However, to obtain a solution it will be necessary to use the finite difference form of the equations. These finite difference approximations are computed using the nodal scheme shown in Sketch 1.

Sketch 1. -
Nodal Pattern




The finite difference approximations for the partial derivatives of the dependent variables are functions of their values at the streamwise coordinate, $m + 1$, where the solutions are to be calculated; and at the station, $m$, where the flow field is already known.

G represents a particular dependent variable

$$G = G_e$$

$$G_s = \frac{1}{\Delta s}(G_e - G_b) \qquad (22)$$

$$G_n = \frac{1}{4\Delta n}(G_c - G_a + G_f - G_d)$$

$$G_{nn} = \frac{1}{2(\Delta n)^2}(G_c - 2G_b + G_a + G_f - 2G_e + G_d)$$

The approximations for the thermodynamic properties are evaluated only in terms of their values at streamwise coordinate, $m$:

R represents a thermodynamic property

$$R_n = \frac{1}{2\Delta n}(R_c - R_a) \qquad (23)$$

$$R_{nn} = \frac{1}{(\Delta n)^2}(R_c - 2R_b + R_a)$$

Difference approximations for partial derivatives involving products are defined so that the dependent variables at the $m + 1$ streamwise coordinate appear linearly in the resulting equation.

$$G\, M_s = \frac{1}{\Delta s} G_b \left(M_e - M_b\right)$$

$$G_n M_n = \frac{1}{8(\Delta n)^2} \left[(G_c - G_a)(M_f - M_d) + (G_f - G_d)(M_c - M_a)\right]$$

$$G^2 = G_b\, G_e \tag{24}$$

$$G\, M_n = G_b \frac{1}{4\Delta n} \left(M_c - M_a + M_f - M_d\right)$$

$$G_n^2 = \frac{1}{4(\Delta n)^2} (G_c - G_a)(G_f - G_d)$$

Substitution of the appropriate approximations into the momentum equation, using $\tilde{\alpha} = \alpha(1-n)$, results in the following:

$$\begin{aligned}
&\tilde{\alpha}^2\, C_n \cdot \frac{1}{4\Delta n} (F_c - F_a + F_f - F_d) + \tilde{\alpha}\, f \frac{1}{4\Delta n} (F_c - F_a + F_f - F_d) \\
&\quad + C\, \tilde{\alpha}^2 \frac{1}{2(\Delta n)^2} (F_c - 2F_b + F_a + F_f - 2F_e + F_d) \\
&\quad - C\, \alpha\, \tilde{\alpha}\, \frac{1}{4\Delta n} (F_c - F_a + F_f - F_d) \qquad (25) \\
&\quad + \beta \left(\frac{\rho_e}{\rho} - F_b F_e\right) = 2s\, F_b \frac{F_e - F_b}{\Delta s} \\
&\quad - \tilde{\alpha}\, (2s)\, f_s \frac{1}{4\Delta n} (F_c - F_a + F_f - F_d)
\end{aligned}$$

In the above equation, $f_s$ is evaluated using the relations for the thermodynamic properties. Although the derivatives of the properties are not represented in finite difference forms, it is understood that the values are to be obtained in this manner. The finite difference form of the momentum equation takes on the form

$$F_f \left( \frac{\tilde{\alpha}\, f}{4\Delta n} + \frac{\tilde{\alpha}^2\, C_n}{4\Delta n} + \frac{\tilde{\alpha}^2\, C}{2(\Delta n)^2} - \frac{\tilde{\alpha}\alpha\, C}{4\Delta n} + \frac{2s\, \tilde{\alpha}\, f_s}{4\Delta n} \right)$$

$$+ F_e \left( - \frac{2\tilde{\alpha}^2\, C}{2(\Delta n)^2} - \beta\, F_b - \frac{2s\, F_b}{\Delta s} \right)$$

$$+ F_d \left( - \frac{\tilde{\alpha}\, f}{4\Delta n} - \frac{\tilde{\alpha}^2\, C_n}{4\Delta n} + \frac{\tilde{\alpha}^2\, C}{2(\Delta n)^2} + \frac{\tilde{\alpha}\alpha\, C}{4\Delta n} - \frac{2s\, \tilde{\alpha}\, f_s}{4\Delta n} \right) \qquad (26)$$

$$= - \frac{\tilde{\alpha}\, f\, (F_c - F_a)}{4\Delta n} - \frac{\tilde{\alpha}^2\, C_n (F_c - F_a)}{4\Delta n} - \frac{\tilde{\alpha}^2\, C(F_c - 2F_b + F_a)}{2(\Delta n)^2}$$

$$+ \frac{\tilde{\alpha}\alpha\, C(F_c - F_a)}{4\Delta n} - \beta\, \frac{\rho_e}{\rho} - \frac{2s\, F_b^2}{\Delta s} - \frac{2s\, \tilde{\alpha}\, f_s\, (F_c - F_a)}{4\Delta n}$$

The quantities enclosed in brackets depend only upon properties and dependent variables evaluated at the streamwise coordinate m, where the solution is already known. The equation can be rewritten:

Conservation of momentum:

$$A_{11} F_d + B_{11} F_e + C_{11} F_f = D_1 \qquad (27)$$

Conservation of energy:

$$A_{21} F_d + B_{21} F_e + C_{21} F_f + A_{22} \theta_d + B_{22} \theta_e + C_{22} \theta_f + A_{23} C_{1d} + C_{23} C_{1f} = D_2 \qquad (28)$$

Conservation of species:

$$A_{33} C_{1d} + B_{33} C_{1e} + C_{33} C_{1f} = D_3 \qquad (29)$$

The following matrix equation can be used to represent these three equations, where the A, B, and C matrix elements are defined in Appendix A.

$$\begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix} \begin{bmatrix} F_d \\ \theta_d \\ C_{1d} \end{bmatrix} + \begin{bmatrix} B_{11} & 0 & 0 \\ B_{21} & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix} \begin{bmatrix} F_e \\ \theta_e \\ C_{1e} \end{bmatrix} + \begin{bmatrix} C_{11} & 0 & 0 \\ C_{21} & C_{22} & C_{23} \\ 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} F_f \\ \theta_f \\ C_{1f} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \qquad (30)$$

Replace

$$\underset{\sim}{\omega}_d \equiv \begin{bmatrix} F_d \\ \theta_d \\ C_{1d} \end{bmatrix} \quad \underset{\sim}{A} \equiv \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix}$$

Noting that d corresponds to i-1, e to i, and f to i+1, the matrix equation can be rewritten in more compact notation:

$$\underset{\sim}{A}_{i,m}\ \underset{\sim}{\omega}_{i-1,m+1} + \underset{\sim}{B}_{i,m}\ \underset{\sim}{\omega}_{i,m+1} + \underset{\sim}{C}_{i,m}\ \underset{\sim}{\omega}_{i+1,m+1} = \underset{\sim}{D}_{i,m} \qquad (31)$$

The subscripts i,m refer to the nodal point at which the matrix is evaluated. The terms in the A, B, C and D matrices depend only upon quantities at station m. Since these quantities are known, the A, B, C and D matrices can be evaluated. As the value of i varies from 1 at the wall to N at the boundary-layer edge, N-2 matrix equations of the form shown above are produced. The equation is not valid at i=1 or i=N because the values of the elements in the A, B, C and D matrices for any point n,m also depend upon properties at points i+1,m and i-1,m. At the wall and at the boundary layer edge, the boundary conditions will be used to evaluate the flow variables. In general, the set of matrix equations for all N points can be represented using still another matrix representation:

$$\begin{pmatrix} \underset{\sim}{B}_1 & \underset{\sim}{C}_1 & 0 & 0 & - & - & - & 0 \\ \underset{\sim}{A}_2 & \underset{\sim}{B}_2 & \underset{\sim}{C}_2 & 0 & - & - & - & 0 \\ - & - & - & - & - & - & - & - \\ 0 & - & - & \underset{\sim}{A}_i & \underset{\sim}{B}_i & \underset{\sim}{C}_i & - & 0 \\ - & - & - & - & - & - & - & - \\ 0 & - & - & - & - & \underset{\sim}{A}_{N-1} & \underset{\sim}{B}_{N-1} & \underset{\sim}{C}_{N-1} \\ 0 & - & - & - & - & - & \underset{\sim}{A}_N & \underset{\sim}{B}_N \end{pmatrix} \begin{pmatrix} \underset{\sim}{\omega}_1 \\ \underset{\sim}{\omega}_2 \\ - \\ \underset{\sim}{\omega}_i \\ - \\ \underset{\sim}{\omega}_{N-1} \\ \underset{\sim}{\omega}_N \end{pmatrix} = \begin{pmatrix} \underset{\sim}{D}_1 \\ \underset{\sim}{D}_2 \\ - \\ \underset{\sim}{D}_i \\ - \\ \underset{\sim}{D}_{N-1} \\ \underset{\sim}{D}_N \end{pmatrix} \quad (32)$$

To solve this set of linear matrix equations, the Gauss-Jordan method of elimination is used as described in [13]. Briefly,

$$\underset{\sim}{\omega}_i = -\underset{\sim}{H}_i \, \underset{\sim}{\omega}_{i+1} + \underset{\sim}{G}_i \qquad 1 \leq i \leq N-1 \qquad (33)$$

where

$$\underset{\sim}{H}_i = (\underset{\sim}{B}_i - \underset{\sim}{A}_i \, \underset{\sim}{H}_{i-1})^{-1} \underset{\sim}{C}_i \qquad (34)$$

and

$$\underset{\sim}{G}_i = (\underset{\sim}{B}_i - \underset{\sim}{A}_i \, \underset{\sim}{H}_{i-1})^{-1} \, (\underset{\sim}{D}_i - \underset{\sim}{A}_i \, \underset{\sim}{G}_{i-1}) \quad 2 \leq i \leq N-1 \qquad (35)$$

Equations (34) and (35) do not provide the values of $\underset{\sim}{G}_1$ and $\underset{\sim}{H}_1$ which are necessary to compute $\underset{\sim}{\omega}_1$ using equation (33). These values are obtained using the boundary conditions at the wall. Recall that, at the wall:

$$F = 0$$

$$\theta = \theta_w$$

and

ORIGINAL PAGE IS
OF POOR QUALITY

$$C_{1n} = \frac{\sqrt{2s}\; C_{1w}\; (\rho v)_w\; Sc_w}{\alpha\; \rho_w\; \mu_w\; u_e\; r^k}$$

Following the method outlined in [6], one can write

$$C_1(i=2) = C_1(i=1) + \Delta n[C_1(i=1)]_n + (\Delta n)^2[C_1(i-1)]_{nn}$$

where $C_1(i=2)$ is the value of $C_1$ at the $i=2$ node in the direction normal to the surface. The value of $[C_1(i+1)]_n$ can be determined using equation (19). The value of $[C_1(i=1)]_{nn}$ can be obtained, after some manipulation, from the species equation for $i=1$. The above equation results in the relation:

$$C_1(i=1) = \frac{C_1(i=2)}{1 + ENS} \tag{36}$$

where

$$ENS = \frac{\sqrt{2s}\; Sc_w\; (\rho v)_w}{\alpha(\rho\mu)_w\; u_e\; r^k} \left\{ \Delta n + \frac{(\Delta n)^2}{2} \left[ 1 - \frac{Sc_w}{C_w} \left( \frac{C_w}{Sc_w} \right)_n + \frac{\sqrt{2s}\; Sc_w\; (\rho v)_w}{\alpha(\rho\mu)_w\; u_e\; r^k} \right] \right\}$$

Using this expression for the boundary condition, the appropriate values for $\underset{\sim}{H}_1$ and $\underset{\sim}{G}_1$ are:

$$\underset{\sim}{H}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{1 + ENS} \end{bmatrix} \qquad (37)$$

$$\underset{\sim}{G}_1 = \begin{bmatrix} 0 \\ \theta_w \\ 0 \end{bmatrix} \qquad (38)$$

The solution for the $\underset{\sim}{\omega}$-vector at each of the N-stations of the boundary layer for a given streamwise coordinate, $m+1$, begins at the outer edge of the boundary layer. Using the known boundary conditions:

$$F = 1,\ \theta = \theta_e,\ \text{and}\ C_1 = 1$$

one finds that:

$$\underset{\sim}{\omega}_N = \begin{bmatrix} 1 \\ \theta_e \\ 1 \end{bmatrix} \qquad (39)$$

The solution proceeds inward, i.e., toward the wall, applying equation (33) sequentially to determine the velocity, the temperature, and the species profiles. Having obtained numerical values for these parameters, a convergence check is made. The convergence criteria is satisfied if consecutive solutions of the boundary layer have the slope of the velocity function at the wall within

0.05% of each other. For reference, this term is titled "CONVG" in the Fortran listing of the program. Should the convergence criteria not be satisfied, the elements of the A, B, C and D matrices are reevaluated using the average of the newly calculated value of F (or $\theta$ or $C_1$) and its value at the previous streamwise station. The iteration process continues until consecutive values of the velocity gradient agree. Once the convergence criteria is satisfied, the y-gradient of velocity at the edge of the boundary layer is checked to see if it is within a specified tolerance. Only if the value of the edge shear, which is defined as

$$\frac{F_N - F_{N-1}}{\Delta n} ,$$

is greater than zero and less than 0.2, is the solution considered to be the desired one. Otherwise, $\alpha$ is changed by 5% and a linear extrapolation method is used to recalculate the boundary layer until the edge shear criteria is met. In addition to the viscous profiles, numerical values are obtained for the heat-transfer rate, the skin-friction, the displacement thickness, etc. The procedure is then repeated to obtain the boundary layer profiles at the next streamwise station, until solutions are obtained at all stations for which the inviscid flow has been specified.

The solution of the initial station is obtained using a Faulkner-Skan formulation for a compressible, laminary boundary layer for a perfect gas with $Pr = 0.7$. This initial profile is calculated in the NSBL code by the subroutine entitled PIGYBAK. A linear double interpolation routine is used to calculate reasonable first guesses for $f''(0)$ and $g'(0)$ based on the values of $g(0)$ and $\beta$, as determined from the boundary conditions.

The numerical routine described herein computes the viscous-layer profile of the velocity function $F$, the temperature function $\theta$, and the stream species concentration $C_1$. Values are also obtained for the following parameters:

CF: the local skin-friction coefficient

$$C_f = \frac{\tau}{0.5\rho_e\, u_e^2} \tag{40}$$

DELST: the displacement thickness, feet

$$\delta^* = \int_0^y \left(1 - \frac{\rho u}{\rho_e\, u_e}\right) dy = \frac{\sqrt{2s}}{\rho_e\, u_e\, r^k} \int_0^n \left(\frac{\rho_e}{\rho} - \frac{u}{u_e}\right) \frac{dn}{\alpha(1-n)} \tag{41}$$

DSTAR: a thickness which represents the free-stream mass flow entrained in the boundary layer with injection

$$D^* = \delta^* + \frac{1}{r^k} \int_0^x \frac{(\rho v)_w\, r^k}{\rho_e\, u_e}\, dx \tag{42}$$

This equation corresponds to the definition of Hayasi [14].

HCOEF: the local heat-transfer coefficient, Btu/ft² sec °R

$$h = \frac{\dot{q}}{T_{te} - T_w} \tag{43}$$

HRATIO: the ratio of the local heat-transfer coefficient to the reference heat-transfer coefficient

$$\text{HRATIO} = \frac{h}{h_{ref}} \tag{44}$$

QDOT: the local convective heat-transfer rate, Btu/ft² sec

$$\dot{q} = \left(k \frac{\partial T}{\partial y}\right)_w = \frac{k_w \rho_w u_e T_{te} r^k \alpha(1-n)}{\sqrt{2s}} \left(\frac{\partial \theta}{\partial n}\right)_w \tag{45}$$

STNO: the Stanton number

$$St = \frac{\dot{q}}{\rho_e u_e C_p (T_r - T_w)} \tag{46}$$

TAU: the local skin friction, lbs/ft²

$$\tau = \left(\mu \frac{\partial u}{\partial y}\right)_w = \frac{\rho_w \mu_w u_e^2 r^k \alpha(1-n)}{\sqrt{2s}} \left(\frac{\partial F}{\partial n}\right)_w \tag{47}$$

THMOM: the momentum thickness, feet

$$\theta = \int_0^y \frac{\rho u}{\rho_e u_e}\left(1 - \frac{u}{u_e}\right) dy = \frac{\sqrt{2s}}{\rho_e u_e r^k} \int_0^n \frac{F(1-F)\, dn}{\alpha(1-n)} \tag{48}$$

TREC: the recovery temperature, °R

$$T_r = r(T_{te} - T_e) + T_e \tag{49}$$

ORIGINAL PAGE IS
OF POOR QUALITY

## DISCUSSION OF RESULTS

The NSBL code has been used to generate numerical solutions for a variety of problems. Presented herein are the solutions for some representative flows. Specifically, there are two flows where the similarity conditions are satisfied and two flows where they are not. The two flows where the similarity conditions are satisfied are: (1) incompressible flow over a flat plate (the Blasius problem); and (2) supersonic flow past a sharp cone. The two flows which do not satisfy the similarity conditions are reentry flight conditions of the Space Shuttle Orbiter at (1) a relatively low Mach number ($M = 9.49$); and (2) a very high Mach number ($M = 29.86$).

### Incompressible, Laminar Boundary Layer on a Flat Plate

A solution of the incompressible, laminar boundary layer was obtained with the NSBL code for flow past a flat plate at zero angle of attack. The free-stream velocity was 167 ft/sec; the stagnation pressure was 1.0 atmosphere. Both the stagnation temperature and the wall temperature were 540°R. The skin-friction coefficient, the displacement thickness, and the momentum thickness as calculated using the NSBL code are compared in Table 1 with the results reported by Schlichting [15], and the values calculated using the BLAST code [3]. The values obtained using the NSBL code are in close agreement with the values from the other solutions. The velocity profile computed using the NSBL code is presented in Figure 1. Included for comparison are the velocity profile presented in [15] and that obtained using the BLAST code. The NSBL profile is that calculated for $x = 1.0$ ft, i.e., for a distance of 1.0 ft from the leading edge. It should be noted that the y-transformation used in [15] differs by a factor of $\sqrt{2}$ from that used in the two codes developed at the University of Texas and in the earlier reference of Blasius [16]. However, the solution given in [15] has been converted to the present definition for $\eta$. The velocity profiles calculated by the three methods are in excellent agreement.

## Laminar boundary Layer for Supersonic Flow Past a Sharp Cone

The NSBL code was also used to obtain a solution for the laminar boundary layer on a sharp cone which had a half angle of 12°. The cone was at zero angle of attack in a supersonic stream. The free-stream Mach number was such that the flow downstream of the shock was supersonic and the flow properties were constant along the surface of the cone. At the edge of the boundary layer the static pressure was 0.05 atmosphere, the static temperature was 1000°R, and the Mach number was 4.84. The wall temperature of the cone was 540°R. The velocity profile at a distance of 1.0 ft from the apex of the cone as calculated with the NSBL code is compared in Figure 2 with that computed using the BLAST code. The temperature profiles obtained from the two codes are compared in Figure 3. Note that the maximum temperature in the boundary layer is greater than either the wall temperature or the static temperature at the edge of the boundary layer. This region of elevated temperature within the boundary layer is due to the effect of viscous dissipation. Once again, the agreement between the different calculation procedures is excellent. The convective heat-transfer-rate distributions as calculated by the two codes are presented in Figure 4. Included for comparison is the distribution calculated using Eckert's reference temperature technique (as discussed briefly in the Introduction). The heat-transfer rates were nondimensionalized with the theoretical heat-transfer rate to the stagnation point of a 1.0 ft radius sphere under the same flow conditions as calculated using the theory of Detra, Kemp, and Riddell [17]. The agreement between these three solutions is very good.

## Compressible, Laminar Boundary Layers for Reentry Flow Conditions

As was mentioned in the Introduction, it was desired to write a code that could accurately solve for the boundary layer on a highly cooled wall in an accelerating, high-speed flow. The NSBL code was used to solve for the boundary layer on the Space

Shuttle Orbiter for two flight conditions from the reentry trajectory. These conditions included a relatively low Mach number (M = 9.49) and a very high Mach number (M = 29.86). The inviscid flow fields for these two flight conditions, which were provided by Dr. W. D. Goodrich of NASA Johnson Space Flight Center [18], are presented in Table 2. Additional points were added to those in [18] to improve the accuracy of the NSBL solution. Velocity and temperature profiles, displacement-thickness distributions, heat-transfer-rate distributions, and skin-friction coefficient distributions for these two flight cases, as calculated by the NSBL code, are presented in Figures 5 through 14. Also shown for comparison purposes are the solutions obtained using the BLIMP code [19], as provided by JSC.

For the first flight condition, the free-stream Mach number was 9.49 and the angle of attack was 30.8 degrees. The conditions at the stagnation point and the streamwise distributions for the static pressure, the wall temperature, the entropy at the edge of the boundary layer, and the radius of the equivalent body of revolution are presented in Table 2(a). These conditions were used as the input boundary conditions for the NSBL code and for the BLIMP code. Solutions using the NSBL code were obtained for two thermodynamic flow models: (1) for air that behaves as a thermally perfect ($p = \rho RT$) and a calorically perfect ($C_p$ = constant) gas; and (2) for air that behaves as a real gas. For the perfect-gas model the viscosity was computed using Sutherland's equation,

$$\mu = 2.270 \frac{T^{1.5}}{T + 198.6} \times 10^{-8} \frac{\text{lbf sec}}{\text{ft}^2}$$

Furthermore, the Prandtl number was assumed a constant, Pr = 0.70. The thermal conductivity was calculated using the definition of the Prandtl number, $k = \frac{\mu C_p}{Pr}$. For the real-gas model, thermodynamic properties were calculated using the tabulated values of [20]. The viscosity, thermal conductivity, and Prandtl number were calculated using linear, double-interpolation of the properties tabulated by Hansen [21]. The specific heat was calculated using the definition of the Prandtl number. In the BLIMP solutions the real-gas model for air was used both for the thermodynamic properties and for the transport properties.

The velocity profiles as calculated by the NSBL code are compared to the BLIMP calculations in Figure 5. Figure 5(a) presents the velocity profiles at $x_a = 0.10L$ and Figure 5(b) presents the profiles at $x_a = 0.498L$. At both streamwise locations, the agreement between the NSBL real-gas calculations and the BLIMP calculations is very good. However, the solutions obtained for the perfect-gas model differ substantially from those obtained for the real-gas model. The temperature profiles at these same two streamwise locations are compared in Figure 6. Again, the agreement between the NSBL real-gas and the BLIMP calculations is very good.

The NSBL perfect-gas calculations differ significantly, especially at the downstream location. Note that, at $x_a = 0.498L$, $T = 0.385T_e$ at the wall for the real-gas solution, whereas the corresponding temperature is $0.685T_e$ for the perfect-gas solution. Since the wall temperature is specified as 1500°F for both gas models (see Table 2a), the difference in the two dimensionless values of $T_w/T_e$ indicates a difference between the values of $T_e$. The same inviscid flow field, as defined by the pressure ratio $(p_e/p_{t2})$, the local entropy $(s_e/R)$, and the streamline divergence metric (RDS), was used for both gas property models. Therefore, it is clear that the thermodynamic model used in the calculations can have a significant effect on the resultant solutions.

The displacement-thickness distributions are compared in Figure 7. The BLIMP and the NSBL real-gas distributions are in good agreement. The displacement-thickness distribution calculated using the NSBL perfect-gas model compares reasonably well with the other two.

The heat-transfer-rate distributions are compared in Figure 8. The local heat-transfer rates were divided by the theoretical heat-transfer rate to the stagnation point of a 1.0 ft radius sphere as calculated using the theory in [17]. The NSBL real-gas distribution compares very well with the BLIMP results. However, the values for the NSBL perfect-gas solution are substantially lower than those for the other two models. The skin-friction-coefficient distributions are compared in Figure 9. The agreement between the NSBL real-gas solution and the BLIMP solution is very good. Although the values from the NSBL perfect-gas solution are below those of the other two models, the differences are not as great as they are for the heat-transfer-rate distributions. In summary, the NSBL real-gas calculations and the BLIMP calculations compare very favorably for this flight case. The NSBL perfect-gas calculations, however, differ significantly from those of the other two models and demonstrate the error that could be introduced by using perfect-gas assumptions for a real-gas situation.

For the second flight condition, the free-stream Mach number was 29.86 and the angle of attack was 41.4°. The input boundary conditions for the NSBL code and the BLIMP code are presented in Table 2(b). The NSBL code was used to obtain a solution for the real-gas model only. The velocity profiles at $x_a = 0.107L$ and at $x_a = 0.444L$ as calculated by the NSBL code are compared to the BLIMP solutions in Figures 10(a) and 10(b), respectively. The agreement between the profiles calculated by the two codes is excellent at both streamwise locations. The temperature profiles at the same two streamwise locations are compared in Figures 11(a) and 11(b). Again, the agreement between the solutions provided by the two codes is excellent. It is interesting to note that both codes indicate an inflection point in the temperature profiles at the two streamwise locations. In both cases the physical temperature in the boundary layer in the region of the inflection point was on the order of 4500 to 5000°R. In this temperature range, the oxygen molecules begin to dissociate at the static pressure of this flow (which is approximately 0.01 atm). In this temperature range, the thermal conductivity changes rapidly with temperature. Furthermore, the value of the thermal conductivity at a specific temperature also depends on the static pressure.

The displacement-thickness, the heat-transfer-rate (non-dimensionalized as before), and the skin-friction-coefficient distributions as calculated by the two codes are compared in Figures 12, 13 and 14, respectively. Once again, the real-gas calculations of the NSBL code compare very well with those of the BLIMP code.

## CONCLUDING REMARKS

The agreement between the sample calculations made using the NSBL code and the results available in the literature, along with the calculations of other codes, demonstrates the validity of the solution algorithm of the NSBL code. It should be noted, however, that the equations employed in the code incorporate the assumptions typically applied to thin boundary layers. Furthermore, it is assumed that the viscous boundary layer, the thermal boundary layer, and the species concentration all have the same thickness. Thus, because the iterative solution procedure requires that only the shear be negliglbe at the edge of the boundary layer, the code should be applied with caution to flows where the Prandtl number and the Schmidt number are significantly different than unity. It should also be noted that the NSBL code is applicable only to laminar boundary layers. Therefore, comparisons between experimental results and the solutions obtained with the code are valid only when the physical boundary layer actually is laminar.

Table 1. - The laminar boundary layer for incompressible flow over a flat plate.

| | NSBL | BLAST (Ref.3) | Schlichting (Ref. 15) |
|---|---|---|---|
| $C_f$ | $\frac{0.6816}{\sqrt{Re_x}}$ | $\frac{0.6644}{\sqrt{Re_x}}$ | $\frac{0.6641}{\sqrt{Re_x}}$ |
| $\frac{\delta^*}{x}$ | $\frac{1.7206}{\sqrt{Re_x}}$ | $\frac{1.7963}{\sqrt{Re_x}}$ | $\frac{1.7208}{\sqrt{Re_x}}$ |
| $\frac{\theta}{x}$ | $\frac{0.6541}{\sqrt{Re_x}}$ | $\frac{0.6633}{\sqrt{Re_x}}$ | $\frac{0.664}{\sqrt{Re_x}}$ |

Table 2. - The input boundary conditions for Space Shuttle Orbiter Flight Design Trajectory.

(a) $M_\infty$ = 9.49; altitude = 162,000 ft; angle of attack = 30.83°; $P_{t2}$ = 215.8 lbf/ft$^2$; $T_{t2}$ = 5508°R.

| M | x(ft) | $p_e/p_{t2}$ | $T_w$(°R) | $S_e/R$ | RDS(ft) |
|---|---|---|---|---|---|
| 1 | 0.8325 | 0.9164 | 2265. | 37.24 | 0.8225 |
| 2 | 1.0113 | 0.8900 | 2251. | 37.23 | 0.9900 |
| 3 | 1.1900 | 0.8595 | 2238. | 37.22 | 1.1500 |
| 4 | 1.5188 | 0.8120 | 2211. | 37.19 | 1.4600 |
| 5 | 1.8475 | 0.7687 | 2182. | 37.15 | 1.7500 |
| 6 | 2.1450 | 0.7310 | 2154. | 37.10 | 2.0100 |
| 7 | 2.4425 | 0.6992 | 2124. | 37.07 | 2.2500 |
| 8 | 2.7088 | 0.6710 | 2097. | 37.06 | 2.5000 |
| 9 | 2.9750 | 0.6475 | 2071. | 37.05 | 2.7100 |
| 10 | 3.2875 | 0.6190 | 2041. | 37.04 | 2.9600 |
| 11 | 3.6000 | 0.5930 | 2012. | 37.03 | 3.1800 |
| 12 | 4.1625 | 0.5540 | 1964. | 36.99 | 3.6100 |
| 13 | 4.7250 | 0.5227 | 1923. | 36.95 | 4.0230 |
| 14 | 5.2875 | 0.4990 | 1887. | 36.91 | 4.4250 |
| 15 | 5.8500 | 0.4780 | 1859. | 36.88 | 4.8000 |
| 16 | 6.3875 | 0.4600 | 1834. | 36.85 | 5.1800 |
| 17 | 6.9250 | 0.4454 | 1815. | 36.82 | 5.5480 |
| 18 | 7.4750 | 0.4310 | 1799. | 36.79 | 5.9000 |
| 19 | 8.0250 | 0.4182 | 1784. | 36.77 | 6.2600 |
| 20 | 8.6000 | 0.4060 | 1770. | 36.74 | 6.6300 |
| 21 | 9.1000 | 0.3968 | 1760. | 36.72 | 6.9450 |
| 22 | 9.6500 | 0.3870 | 1750. | 36.69 | 7.2800 |
| 23 | 10.2000 | 0.3770 | 1740. | 36.68 | 7.6200 |
| 24 | 10.7380 | 0.3690 | 1733. | 36.65 | 7.9450 |
| 25 | 11.275 | 0.3610 | 1727. | 36.63 | 8.2700 |

ORIGINAL PAGE IS
OF POOR QUALITY

Table 2. - (a) Continued.

| M | x(ft) | $p_e/p_{t2}$ | $T_w$(°R) | $S_e/R$ | RDS(ft) |
|---|---|---|---|---|---|
| 26 | 11.813 | 0.3540 | 1723. | 36.60 | 8.5890 |
| 27 | 12.350 | 0.3477 | 1720. | 36.58 | 8.9075 |
| 28 | 12.888 | 0.3415 | 1717. | 36.56 | 9.2160 |
| 29 | 13.425 | 0.3360 | 1715. | 36.53 | 9.5250 |
| 30 | 13.963 | 0.3300 | 1712. | 36.51 | 9.8340 |
| 31 | 14.500 | 0.3255 | 1710. | 36.49 | 10.143 |
| 32 | 15.575 | 0.3170 | 1706. | 36.44 | 10.738 |
| 33 | 16.650 | 0.3086 | 1701. | 36.40 | 11.333 |
| 34 | 17.713 | 0.3020 | 1691. | 36.36 | 11.925 |
| 35 | 18.775 | 0.2955 | 1675. | 36.31 | 12.518 |
| 36 | 20.388 | 0.2880 | 1612. | 36.25 | 13.363 |
| 37 | 22.000 | 0.2819 | 1557. | 36.19 | 14.208 |
| 38 | 24.680 | 0.2770 | 1537. | 36.07 | 15.599 |
| 39 | 27.360 | 0.2751 | 1530. | 36.96 | 16.990 |
| 40 | 30.030 | 2.2760 | 1526. | 35.85 | 18.399 |
| 41 | 32.700 | 0.2772 | 1522. | 35.76 | 19.808 |
| 42 | 35.385 | 0.2780 | 1518. | 35.68 | 21.174 |
| 43 | 38.070 | 0.2795 | 1515. | 35.62 | 22.540 |
| 44 | 40.735 | 0.2800 | 1512. | 35.56 | 23.920 |
| 45 | 43.400 | 0.2810 | 1510. | 35.50 | 25.300 |
| 46 | 46.085 | 0.2820 | 1507. | 35.44 | 26.695 |
| 47 | 48.770 | 0.2835 | 1505. | 35.39 | 28.090 |
| 48 | 51.435 | 0.2843 | 1502. | 35.34 | 29.480 |
| 49 | 54.100 | 0.2850 | 1500. | 35.29 | 30.870 |
| 50 | 56.785 | 0.2860 | 1493. | 35.24 | 32.250 |

Table 2. - Continued

(b) $M_\infty$ = 29.86; altitude = 246,000 ft; angle of attack = 41.4°; $p_{t2}$ = 46.26 lbf/ft$^2$; $T_{t2}$ = 10,798°R.

| M | x(ft) | $p_e/p_{t2}$ | $T_w$(°R) | $S_e/R$ | RDS(ft) |
|---|---|---|---|---|---|
| 1 | 1.9500 | 0.8950 | 3095. | 54.13 | 1.8900 |
| 2 | 2.0470 | 0.8890 | 3076. | 54.09 | 1.9800 |
| 3 | 2.1430 | 0.8840 | 3061. | 54.03 | 2.0600 |
| 4 | 2.2400 | 0.8780 | 3048. | 53.97 | 2.1500 |
| 5 | 2.3370 | 0.8740 | 3037. | 53.92 | 2.2300 |
| 6 | 2.5300 | 0.8640 | 3016. | 53.81 | 2.4100 |
| 7 | 2.7230 | 0.8540 | 2996. | 53.70 | 2.5900 |
| 8 | 2.9200 | 0.8440 | 2977. | 53.60 | 2.7700 |
| 9 | 3.1100 | 0.8355 | 2960. | 53.50 | 2.9300 |
| 10 | 3.3000 | 0.8260 | 2942. | 53.39 | 3.1000 |
| 11 | 3.4900 | 0.8170 | 2927. | 53.29 | 3.2700 |
| 12 | 3.6800 | 0.8090 | 2912. | 53.20 | 3.4300 |
| 13 | 3.8700 | 0.8000 | 2897. | 53.10 | 3.6000 |
| 14 | 4.0600 | 0.7920 | 2883. | 53.01 | 3.7500 |
| 15 | 4.2500 | 0.7837 | 2869. | 52.92 | 3.9100 |
| 16 | 4.6170 | 0.7680 | 2839. | 52.76 | 4.2300 |
| 17 | 4.9830 | 0.7530 | 2812. | 52.61 | 4.5300 |
| 18 | 5.3500 | 0.7390 | 2784. | 52.48 | 4.8400 |
| 19 | 5.7170 | 0.7250 | 2758. | 52.36 | 5.1500 |
| 20 | 6.0830 | 0.7125 | 2732. | 52.26 | 5.4300 |
| 21 | 6.4500 | 0.7006 | 2707. | 52.19 | 5.7300 |
| 22 | 6.8100 | 0.6900 | 2684. | 52.11 | 6.0200 |
| 23 | 7.1700 | 0.6790 | 2660. | 52.05 | 6.3200 |
| 24 | 7.5300 | 0.6677 | 2637. | 51.99 | 6.5900 |
| 25 | 7.8970 | 0.6590 | 2614. | 51.93 | 6.8800 |

Table 2. - (b) Conclusion.

| M | x(ft) | $p_e/p_{t2}$ | $T_w$(°R) | $S_e/R$ | RDS(ft) |
|---|---|---|---|---|---|
| 26 | 8.2630 | 0.6500 | 2595. | 51.87 | 7.1700 |
| 27 | 8.6300 | 0.6411 | 2577. | 51.82 | 7.4300 |
| 28 | 9.1700 | 0.6290 | 2552. | 51.75 | 7.8300 |
| 29 | 9.7100 | 0.6175 | 2535. | 51.68 | 8.2500 |
| 30 | 10.255 | 0.6080 | 2522. | 51.61 | 8.6500 |
| 31 | 10.800 | 0.5990 | 2511. | 51.54 | 9.0300 |
| 32 | 11.350 | 0.5900 | 2499. | 51.46 | 9.4600 |
| 33 | 11.900 | 0.5815 | 2488. | 51.37 | 9.8400 |
| 34 | 12.400 | 0.5740 | 2479. | 51.30 | 10.230 |
| 35 | 12.900 | 0.5667 | 2470. | 51.24 | 10.600 |
| 36 | 13.450 | 0.5603 | 2460. | 51.18 | 11.100 |
| 37 | 14.000 | 0.5540 | 2450. | 51.13 | 11.600 |
| 38 | 15.050 | 0.5425 | 2433. | 51.05 | 12.250 |
| 39 | 16.100 | 0.5313 | 2419. | 51.02 | 12.900 |
| 40 | 17.200 | 0.5210 | 2403. | 50.95 | 13.610 |
| 41 | 18.300 | 0.5128 | 2388. | 50.88 | 14.400 |
| 42 | 19.900 | 0.4990 | 2365. | 50.77 | 15.600 |
| 43 | 21.500 | 0.4912 | 2345. | 50.68 | 16.700 |
| 44 | 24.200 | 0.4880 | 2311. | 50.55 | 18.400 |
| 45 | 26.900 | 0.4874 | 2281. | 50.45 | 20.300 |
| 46 | 29.550 | 0.4874 | 2259. | 50.36 | 22.100 |
| 47 | 32.200 | 0.4874 | 2237. | 50.25 | 23.900 |
| 48 | 34.900 | 0.4874 | 2218. | 50.10 | 25.700 |
| 49 | 37.600 | 0.4874 | 2200. | 49.95 | 27.500 |
| 50 | 40.250 | 0.4874 | 2182. | 49.80 | 29.300 |
| 51 | 42.900 | 0.4874 | 216[illegible]. | 49.65 | 31.200 |
| 52 | 45.600 | 0.4874 | 2151. | 49.52 | 32.900 |
| 53 | 48.300 | 0.4874 | 2137. | 49.39 | 34.800 |
| 54 | 50.950 | 0.4874 | 2122. | 49.25 | 36.600 |

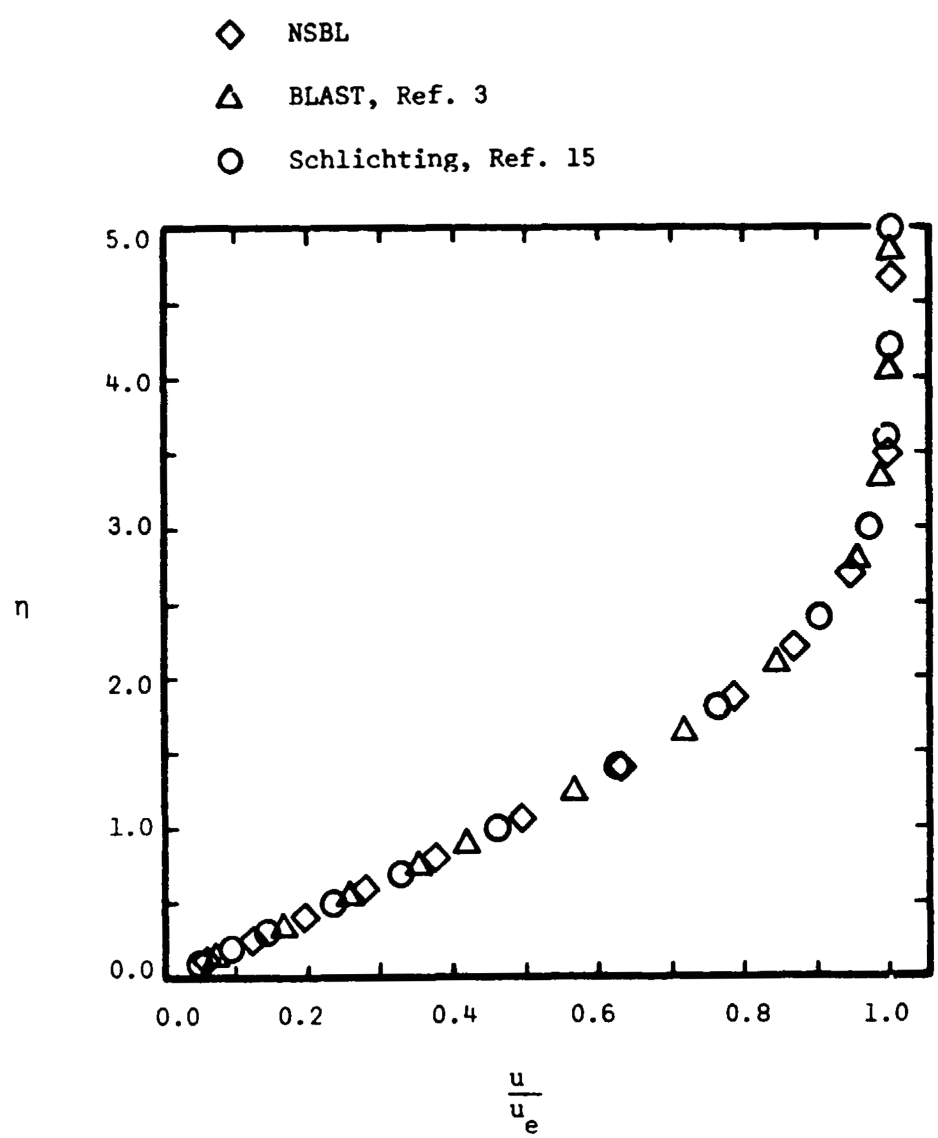



Figure 1. - Velocity profile for incompressible, laminar flow past a flat plate; angle of attack = 0°.

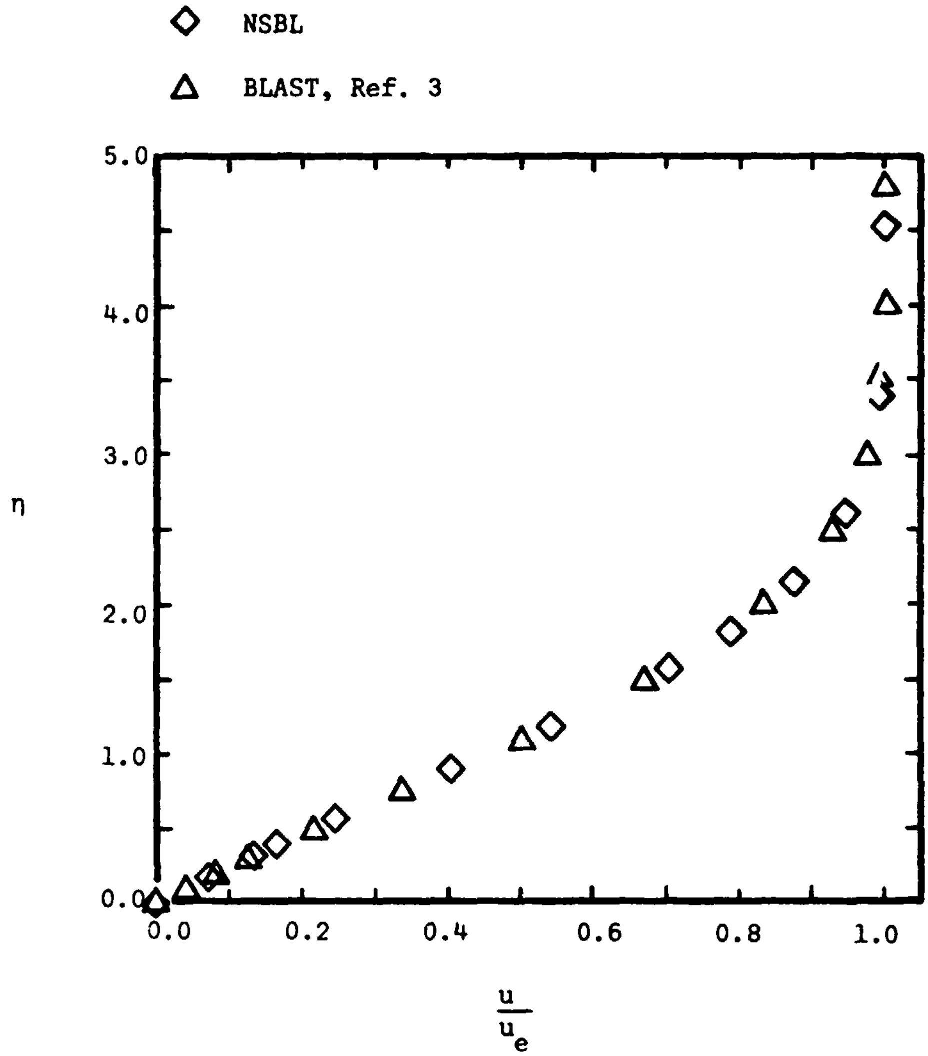



Figure 2. - Velocity profile for supersonic, laminar flow past a 12° cone; $M_e$ = 4.84; angle of attack = 0°; x = 1.0 ft.

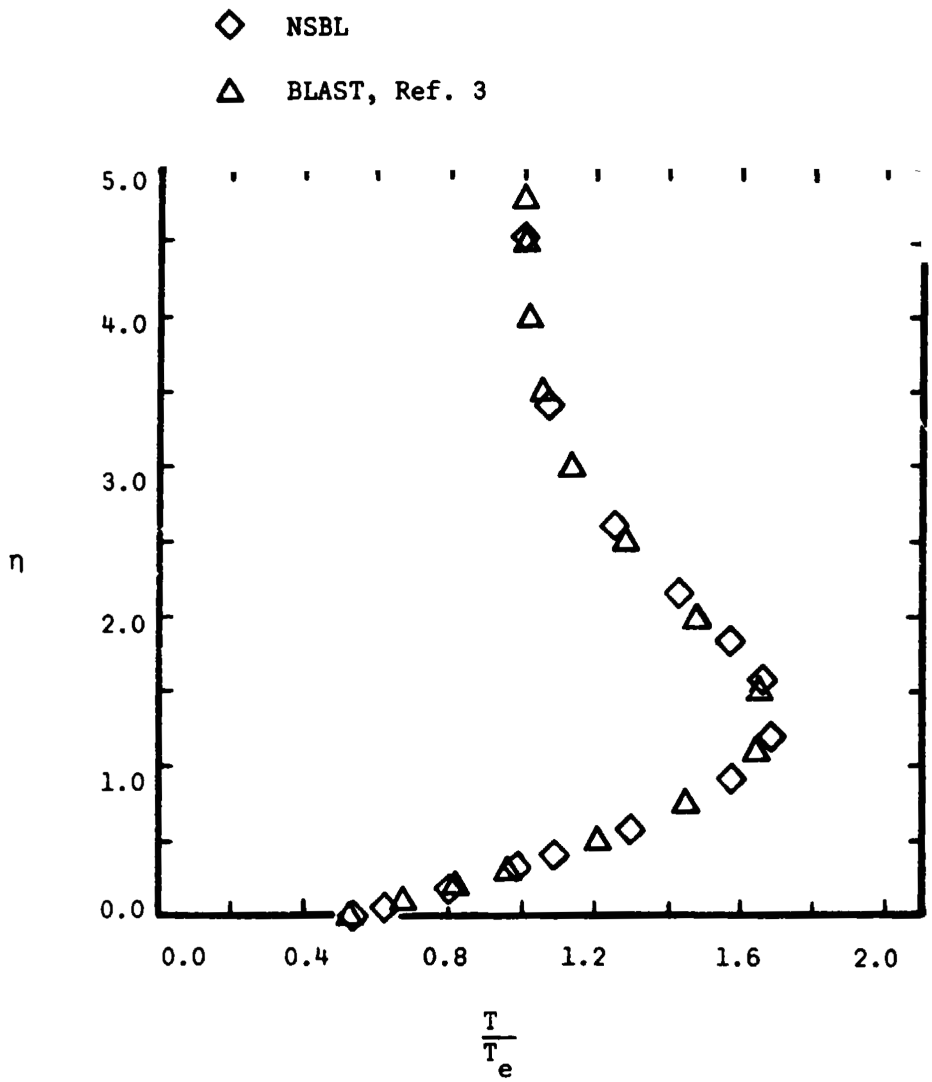



Figure 3. - Temperature profile for supersonic, laminar flow past a 12° cone, $M_e$ = 4.84; angle of attack = 0°. x = 1.0 ft.

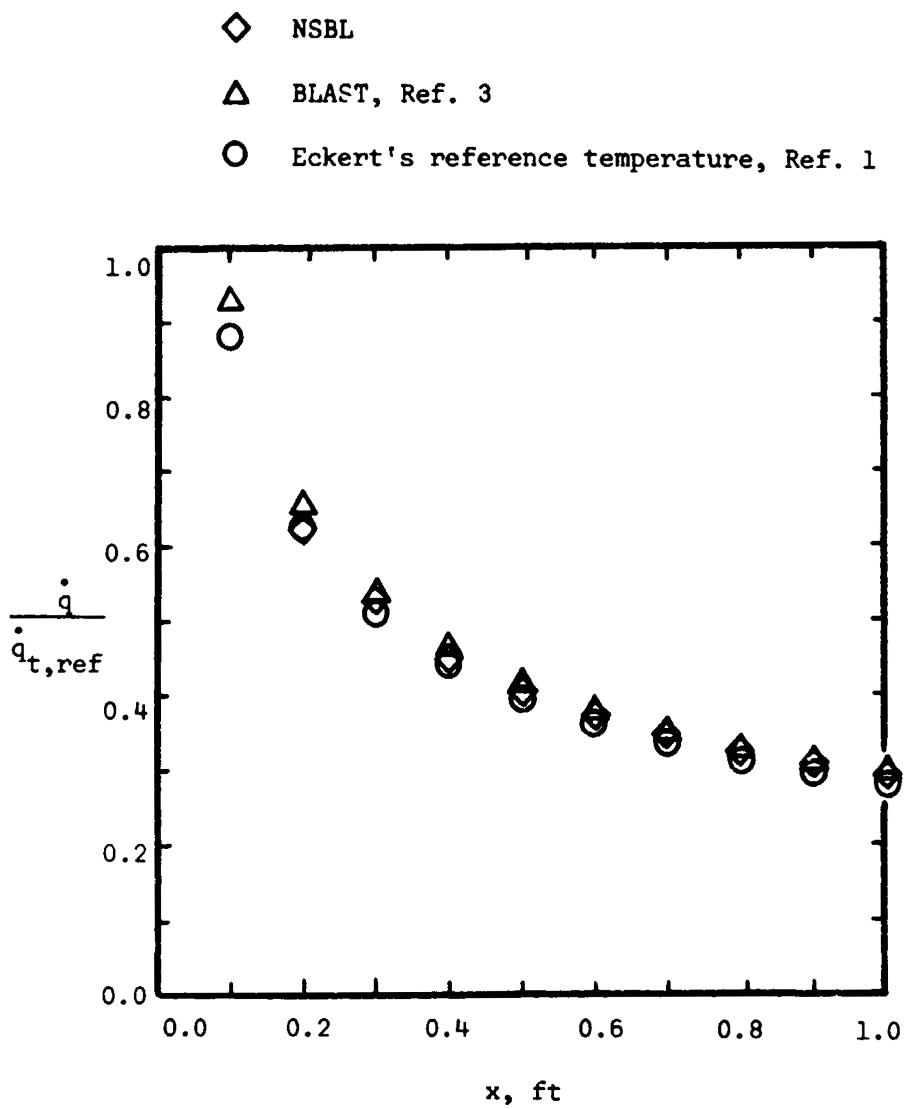
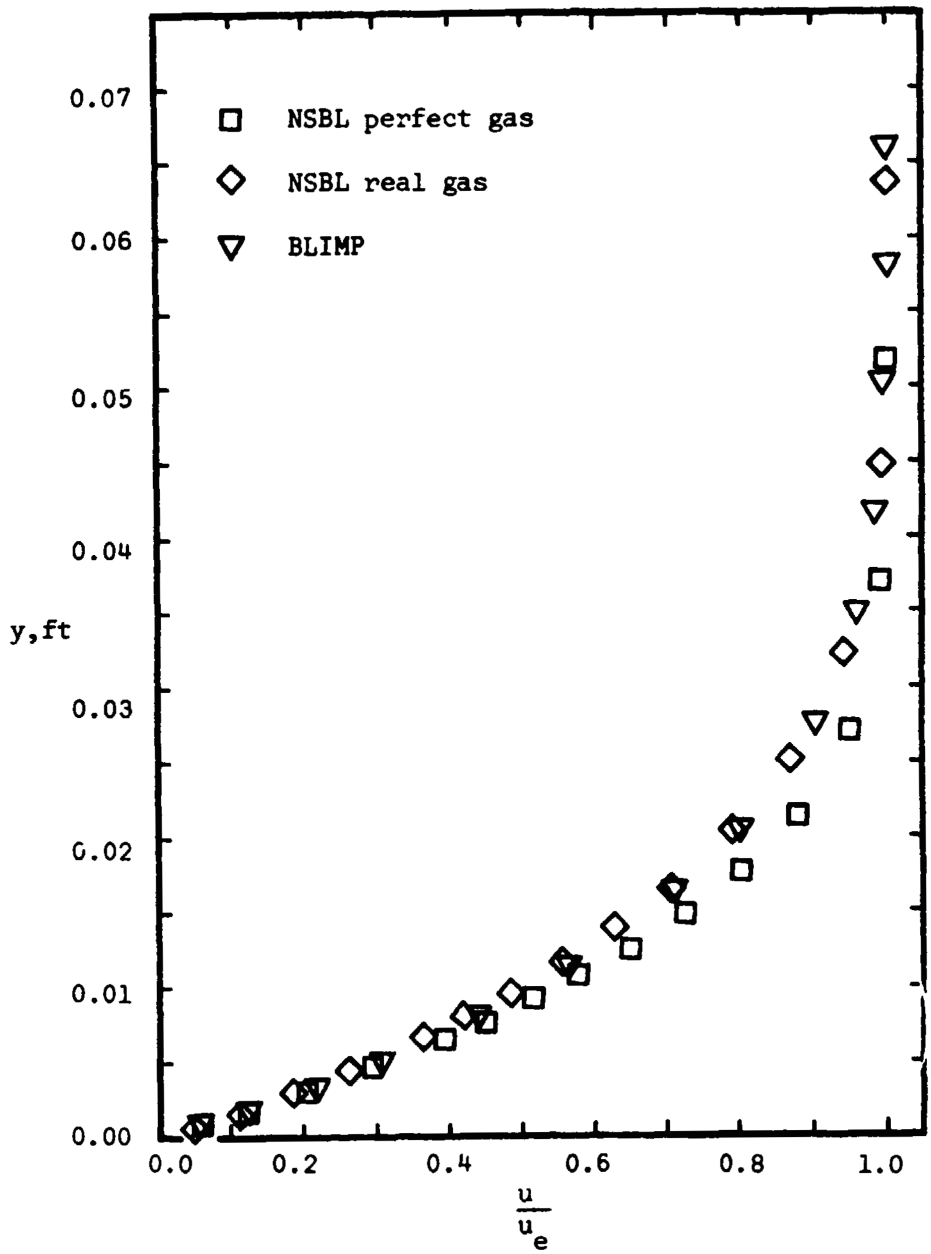
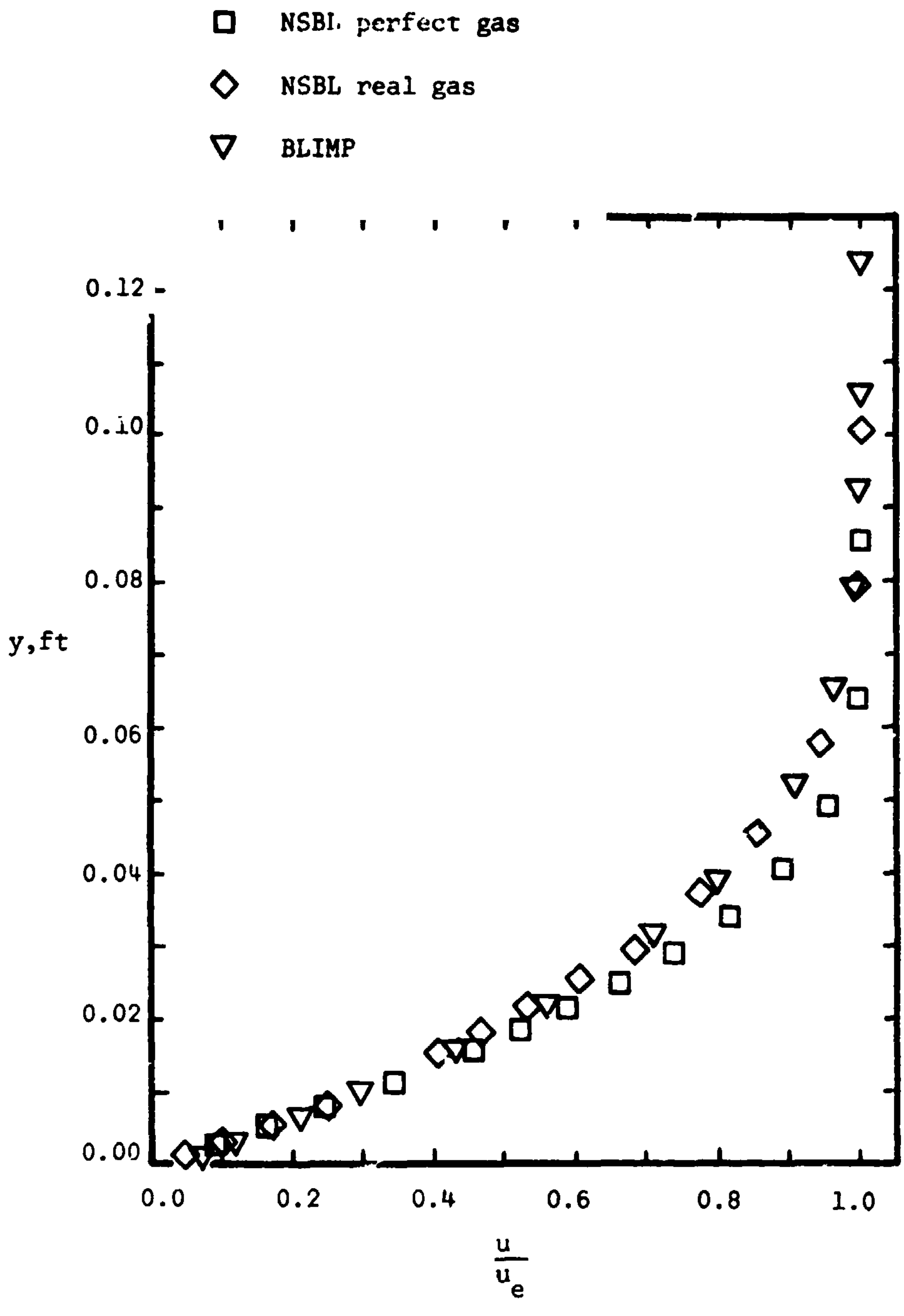



Figure 4. - Heat-transfer-rate distribution for supersonic flow past a 12° cone, $M_e$ = 4.84; angle of attack = 0°.



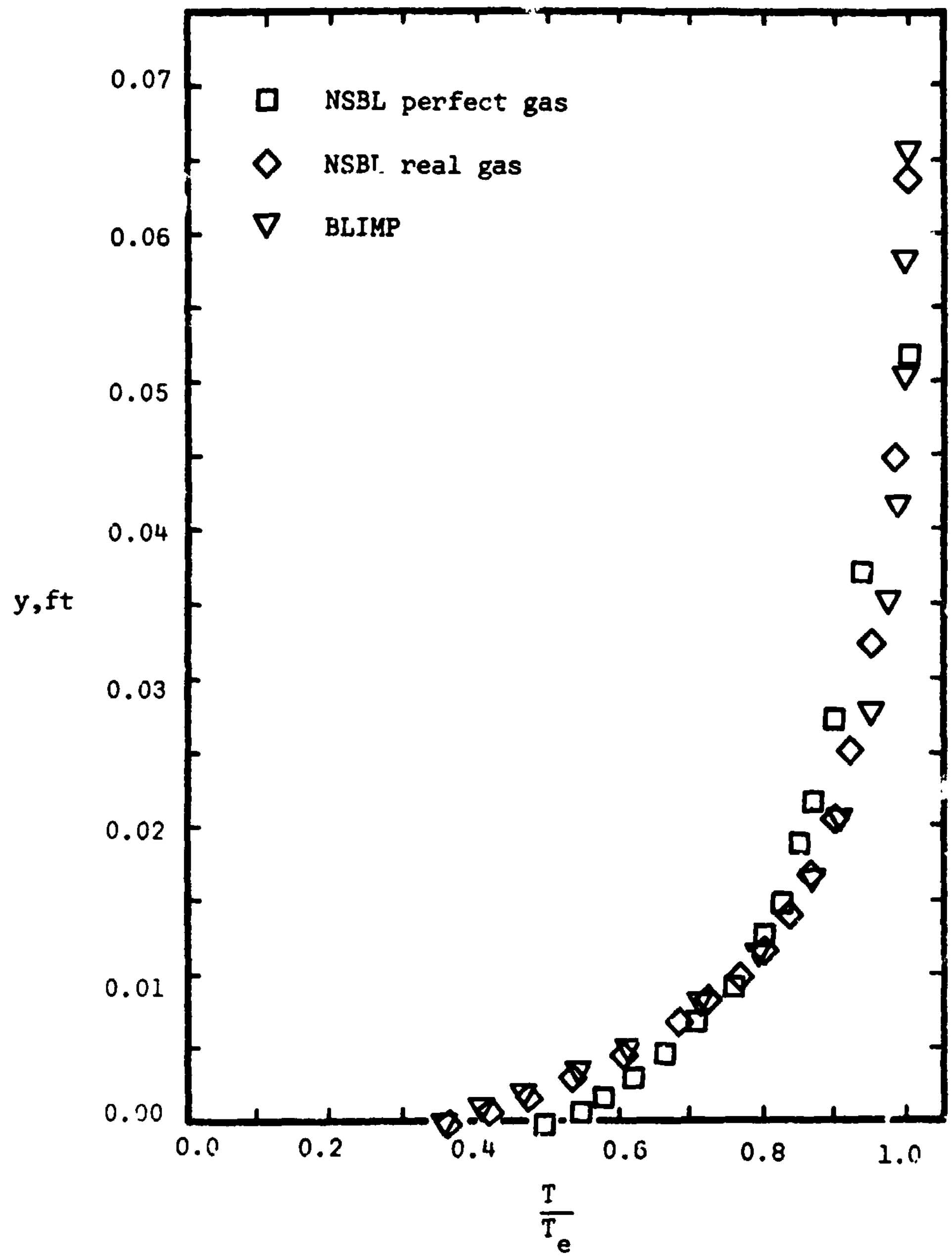

(a) $x_a/L = 0.10$

Figure 5. - Velocity distribution across a laminar boundary layer, Orbiter Flight Design Trajectory; M = 9.49; angle of attack = 30.83°.



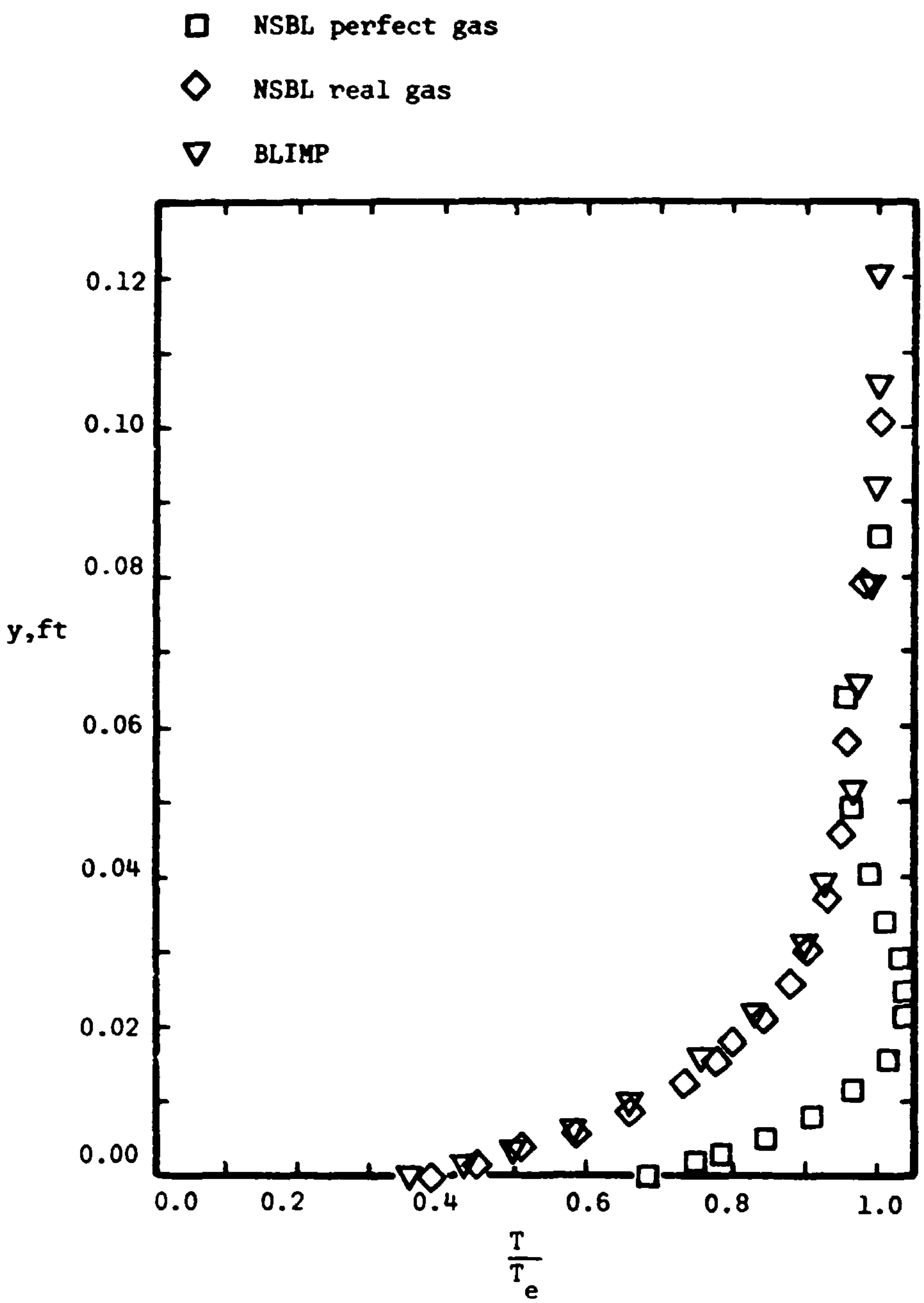

(b) $x_a/L = 0.498$

Figure 5. - Concluded.




(a) $x_a/L = 0.10$

Figure 6. - Temperature distribution across a laminar boundary layer, Orbiter Flight Design Trajectory; M = 9.49; angle of attack = 30.83°.




(b) $x_a/L = 0.498$

Figure 6. - Concluded.

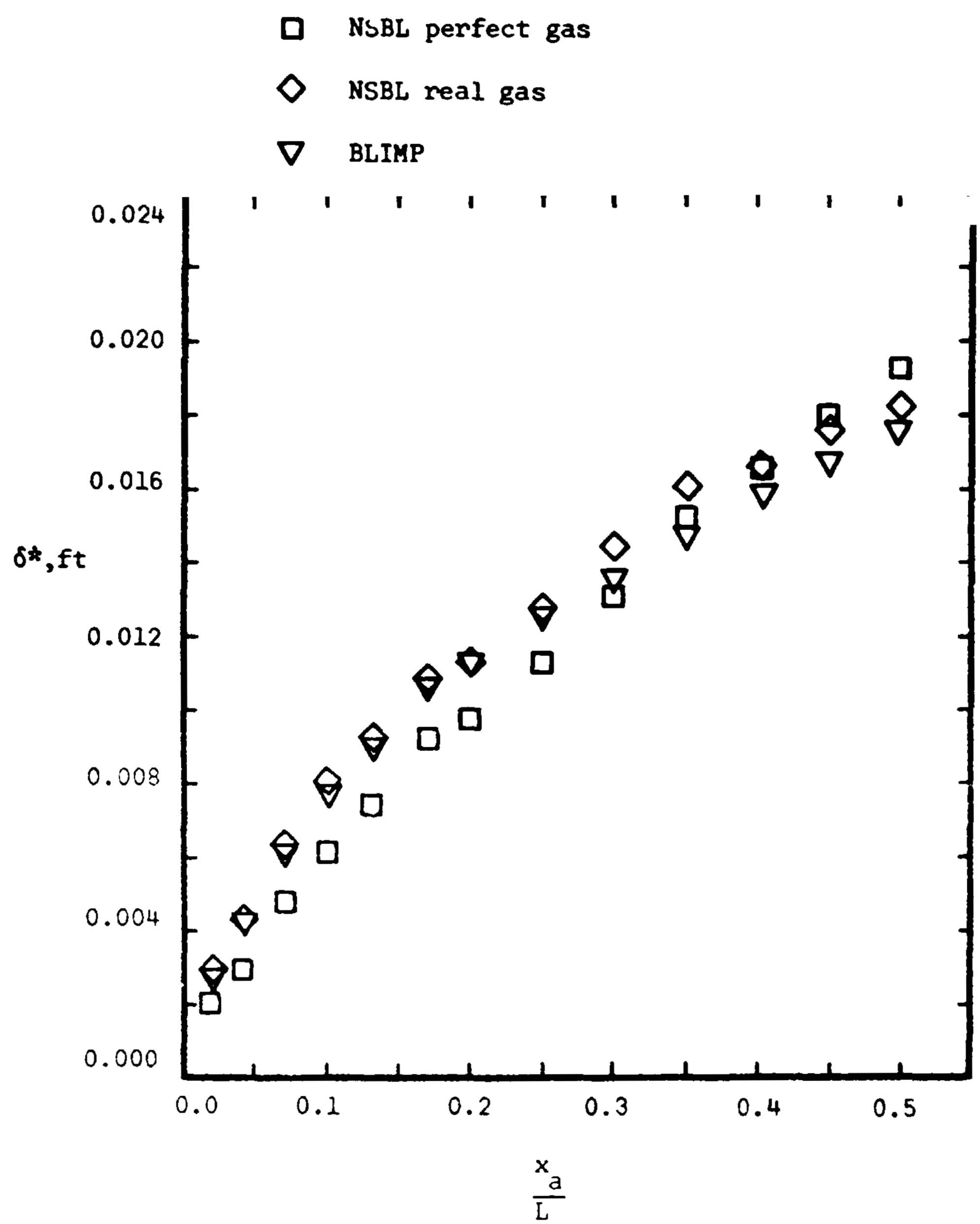



Figure 7. - Displacement-thickness distribution, Orbiter Flight Design Trajectory; M = 9.49; angle of attack = 30.83°.

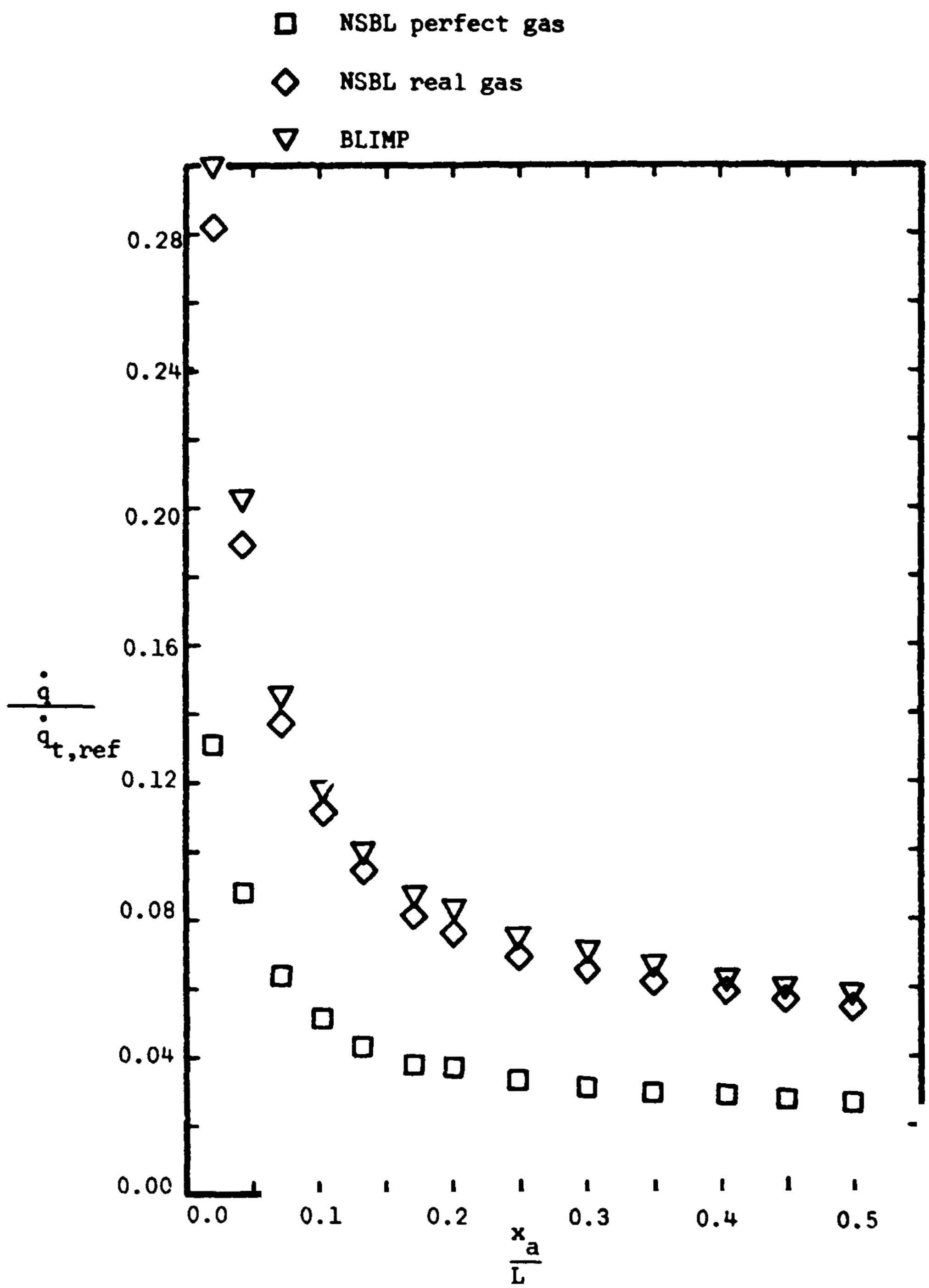



Figure 8. - Heat-transfer-rate distribution, Orbiter Flight Design Trajectory; M = 9.49; angle of attack = 30.83°.

ORIGINAL PAGE IS OF POOR QUALITY

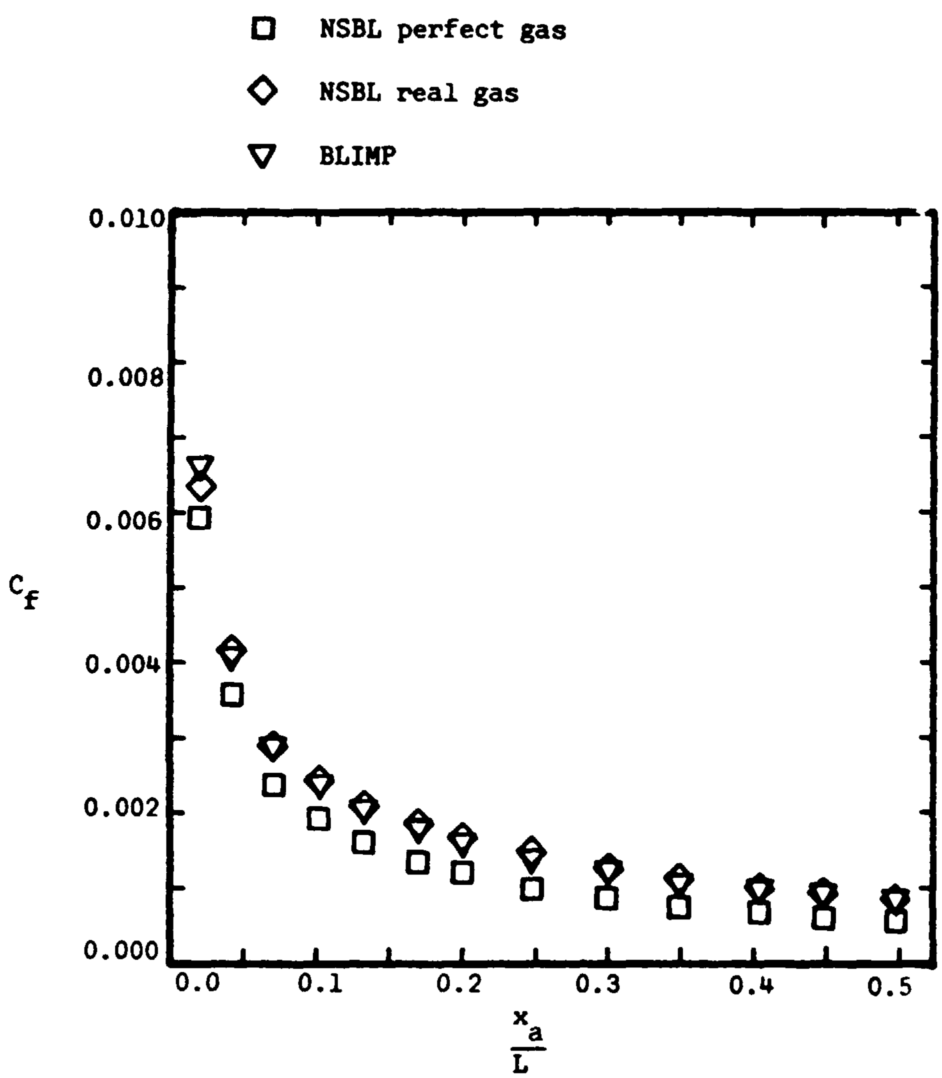
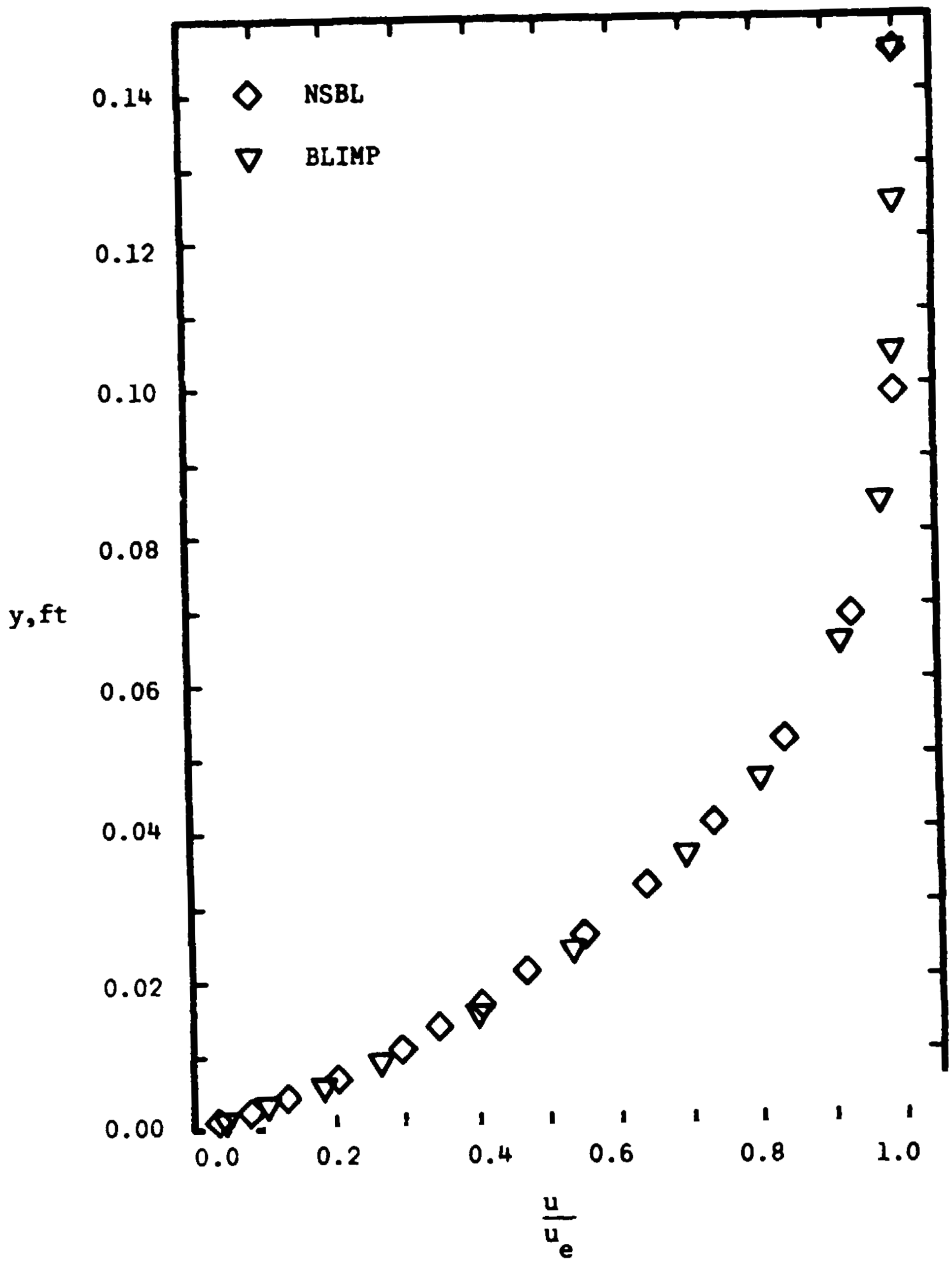
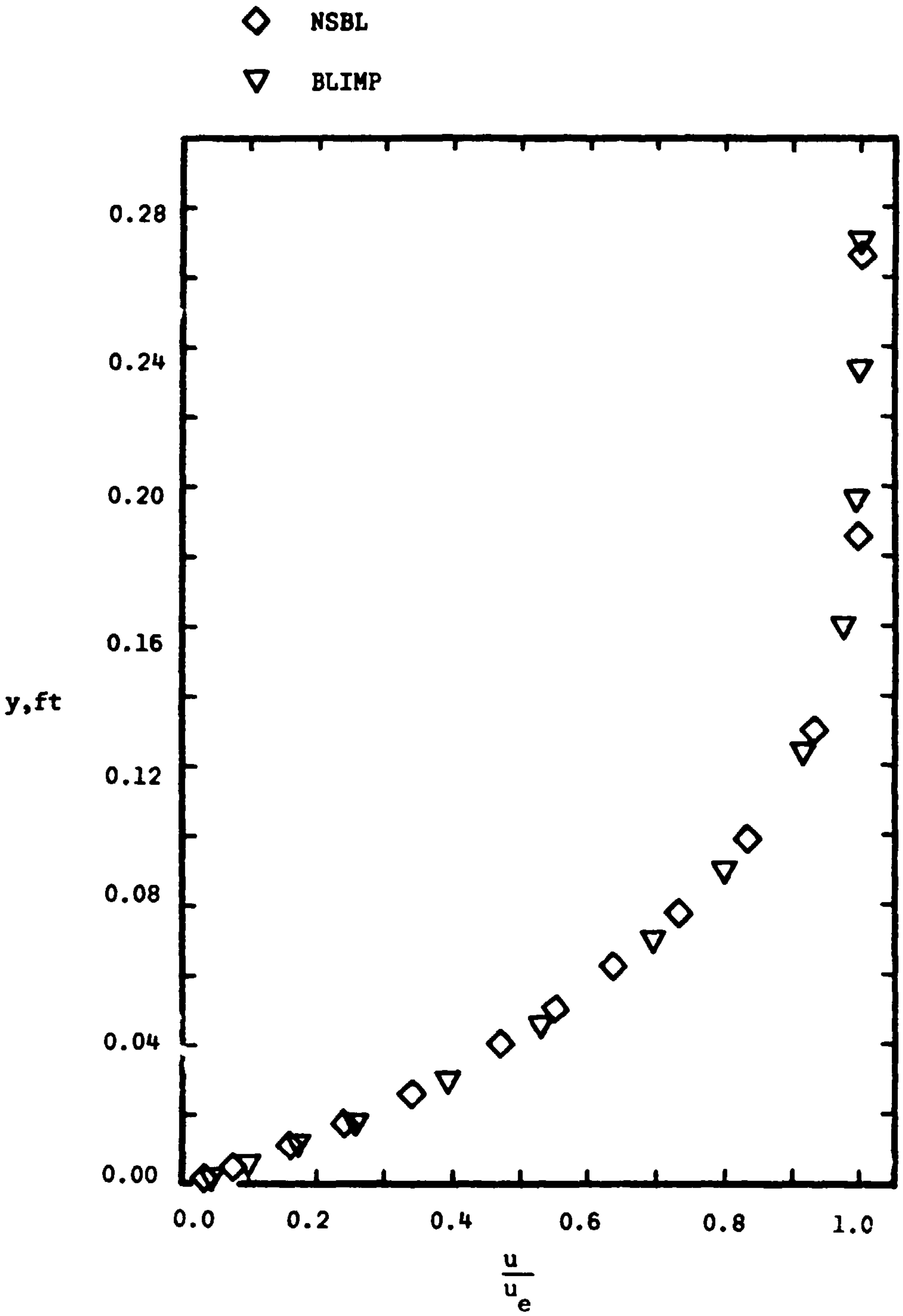



Figure 9. - Skin-friction-coefficient distribution, Orbiter Flight Design Trajectory; M = 9.49; angle of attack = 30.83°.



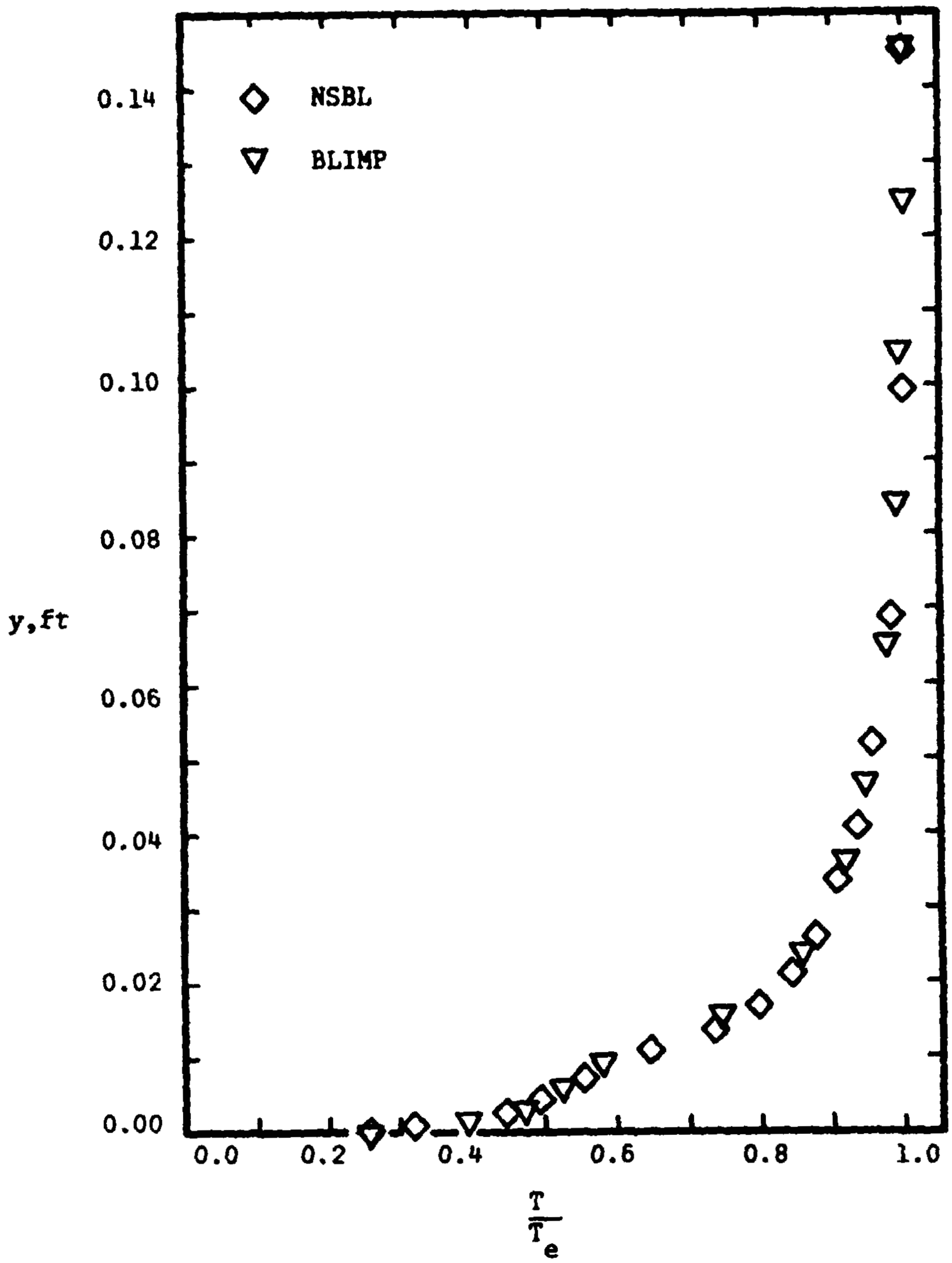

(a) $x_a/L = 0.107$

Figure 10. - Velocity distribution across a laminar boundary layer, Orbiter Flight Design Trajectory; M = 29.86; angle of attack = 41.4°.



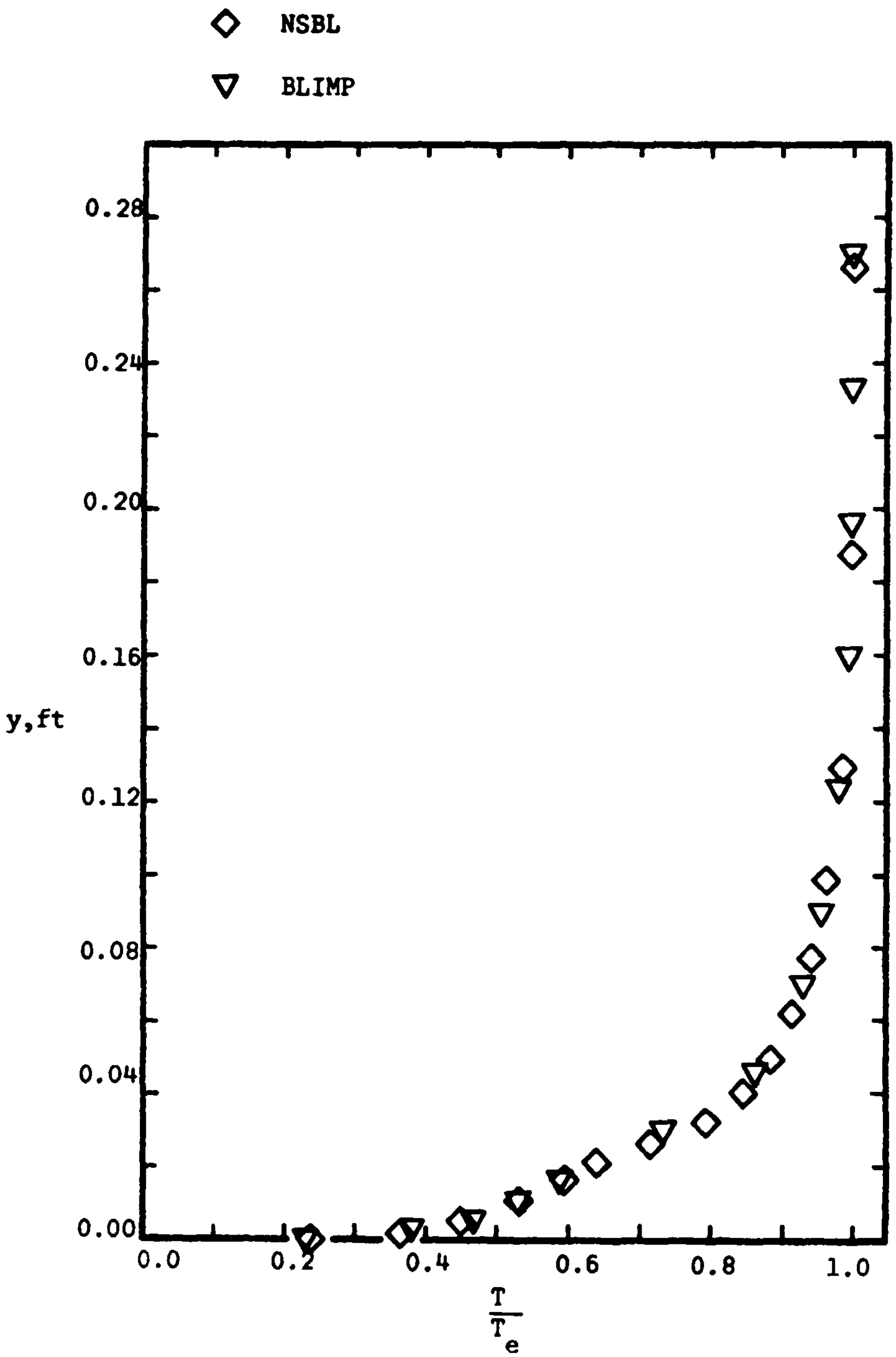

(b) $x_a/L = 0.444$

Figure 10. - Concluded.




(a) $x_a/L = 0.107$

Figure 11. - Temperature distribution across a laminar boundary layer, Orbiter Flight Design Trajectory; M = 29.86; angle of attack = 41.4°.




(b) $x_a/L = 0.444$

Figure 11. - Concluded.

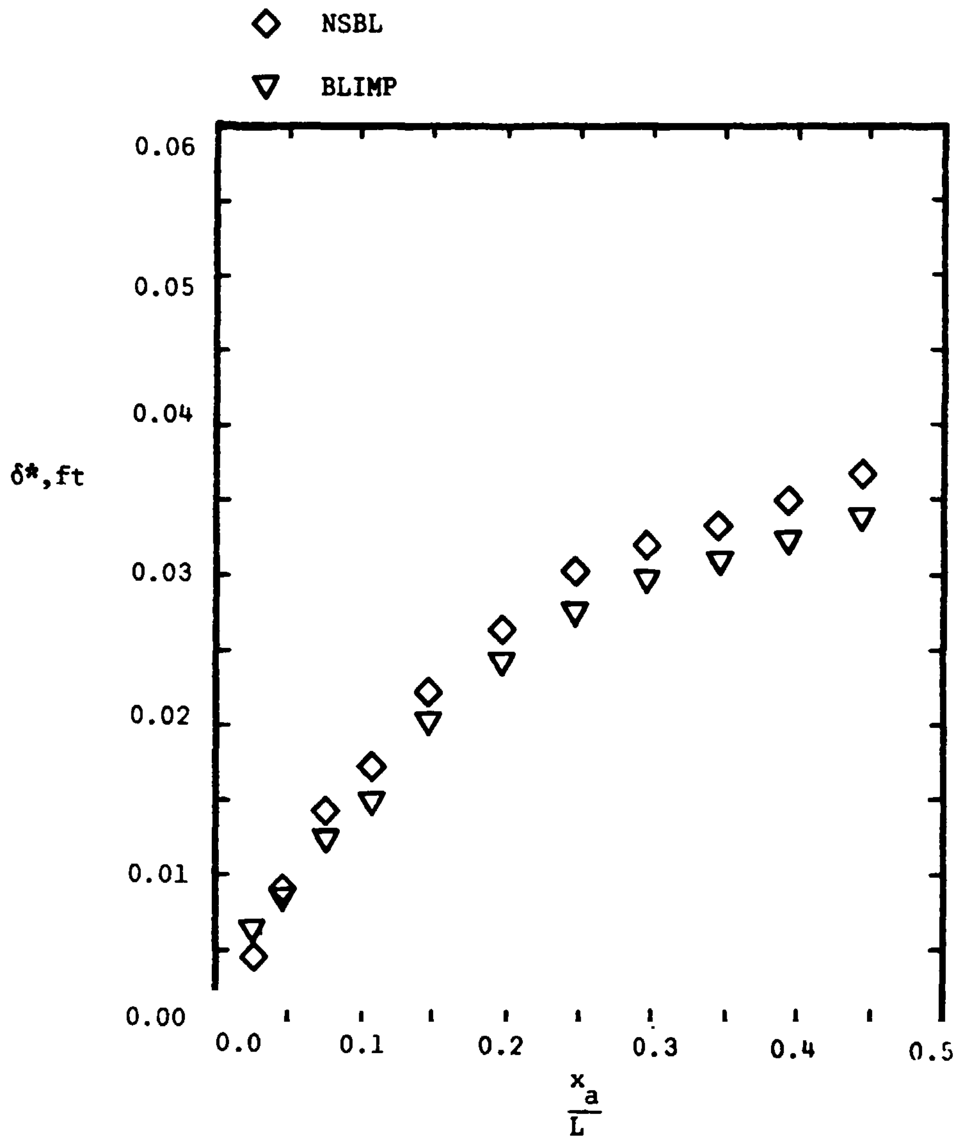



Figure 12. - Displacement-thickness distribution, Orbiter Flight Design Trajectory; M = 29.86; angle of attack = 41.4°.

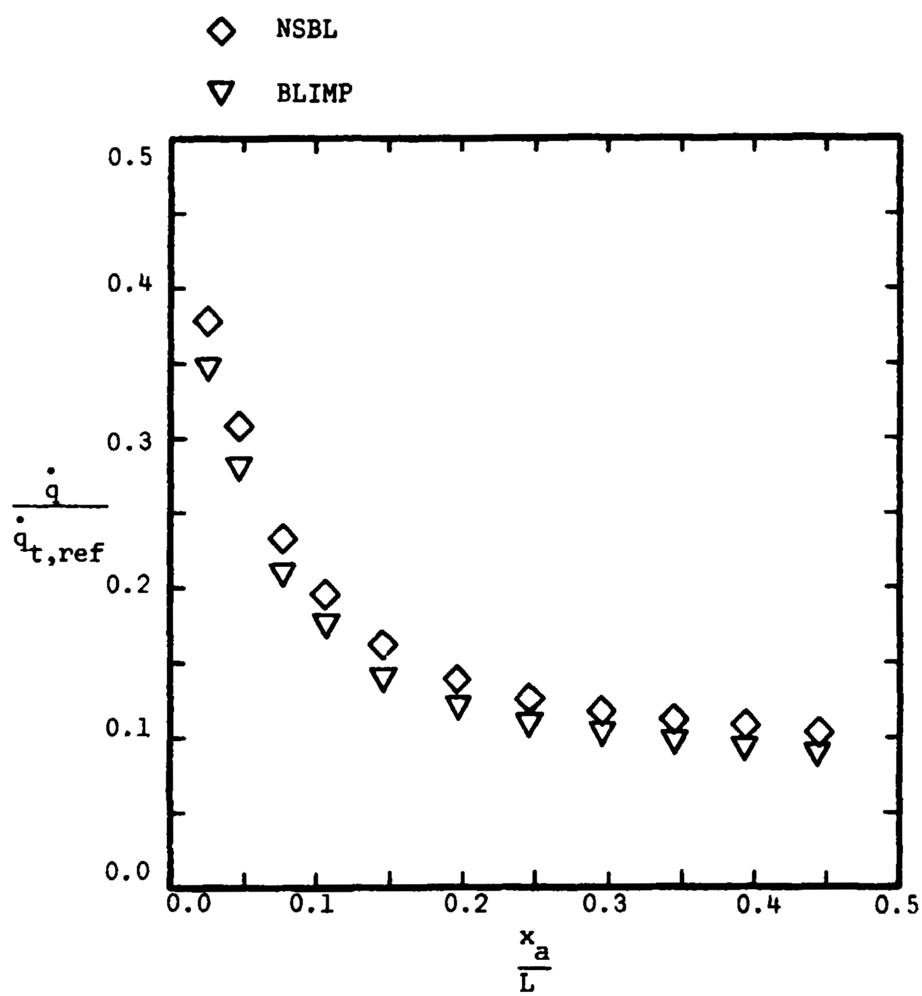



Figure 13. - Heat-transfer-rate distribution, Orbiter Flight Design Trajectory; M = 29.86; angle of attack = 41.4°.

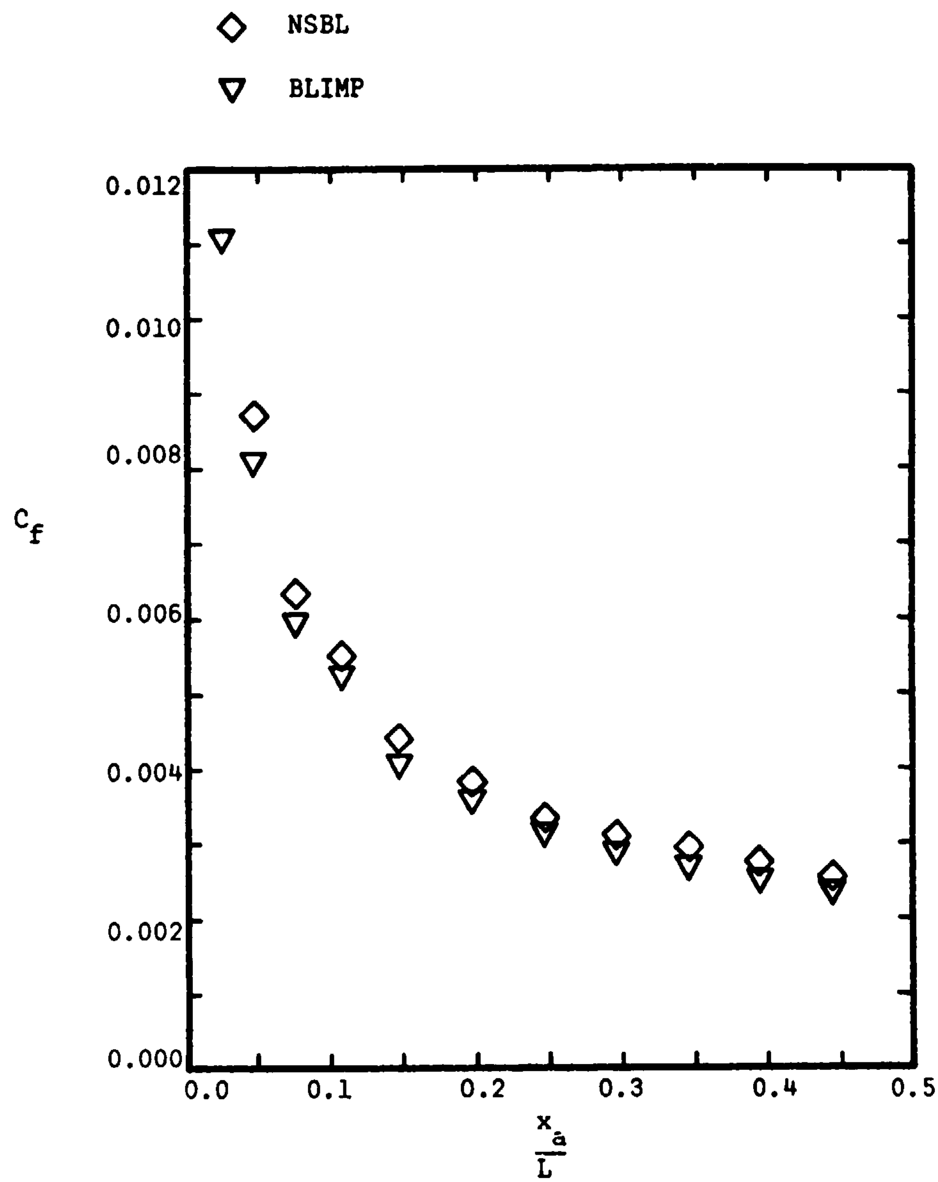



Figure 14. - Skin-friction-coefficient distribution; Orbiter Flight Design Trajectory; M = 29.86; angle of attack = 41.4°.

## APPENDIX A

## DEFINITION OF MATRIX ELEMENTS

The individual elements of the coefficient matrices which appear in equations (28), (29) and (30) are as follows:

$$A_{11} = \frac{\tilde{\alpha}\, f}{4\Delta n} + \frac{\tilde{\alpha}^2\, C_n}{4\Delta n} - \frac{\tilde{\alpha}^2\, C}{2(\Delta n)^2} - \frac{\tilde{\alpha}\alpha\, C}{4\Delta n} + \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$A_{21} = \frac{Cu_e^2\,\tilde{\alpha}^2}{2\Delta n \bar{C}_p T_{te}} \left(\frac{F_c - F_a}{2\Delta n}\right)$$

$$A_{22} = \frac{\tilde{\alpha}^2}{4\Delta n \bar{C}_p} \left(\frac{C\bar{C}_p}{Pr}\right)_n - \frac{\tilde{\alpha}^2}{2(\Delta n)^2} \left(\frac{C}{Pr}\right) - \frac{\tilde{\alpha}\,\alpha}{4\Delta n}\frac{C}{Pr} + \frac{\tilde{\alpha}\, f}{4\Delta n}$$

$$+ \frac{C_{1e}}{\bar{C}_p}\frac{C}{Sc}\frac{(C_{p1} - C_{p2})\,\tilde{\alpha}^2}{4\Delta n}\left(\frac{C_{1c} - C_{1a}}{2\Delta n}\right) + \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$A_{23} = \frac{C_{1e}}{\bar{C}_p}\frac{C}{Sc}\frac{(C_{p1} - C_{p2})\,\tilde{\alpha}^2}{4\Delta n}\left(\frac{\theta_c - \theta_a}{2\Delta n}\right)$$

$$A_{33} = \frac{\tilde{\alpha}^2}{4\Delta n}\left(\frac{C}{Sc}\right)_n - \frac{\tilde{\alpha}^2}{2(\Delta n)^2}\frac{C}{Sc} - \frac{\alpha\tilde{\alpha}}{4\Delta n}\frac{C}{Sc} + \frac{\tilde{\alpha}\, f}{4\Delta n} + \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$B_{11} = \frac{\tilde{\alpha}^2\, C}{(\Delta n)^2} + \beta F_b + \frac{2sF_b}{\Delta s}$$

$$B_{21} = \frac{\beta u_e^2}{\bar{C}_p T_{te}}\frac{\rho_e}{\rho}$$

$$B_{22} = \frac{\tilde{\alpha}^2}{(\Delta n)^2} \frac{C}{Pr} + \frac{2sF_b}{\Delta s}$$

$$B_{23} = 0$$

$$B_{33} = \frac{\tilde{\alpha}^2}{(\Delta n)^2} \frac{C}{Sc} + \frac{2sF_b}{\Delta s}$$

$$C_{11} = \frac{\tilde{\alpha}\alpha C}{4\Delta n} - \frac{\tilde{\alpha} f}{4\Delta n} - \frac{\tilde{\alpha}^2 C_n}{4\Delta n} - \frac{\tilde{\alpha}^2 C}{2(\Delta n)^2} - \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$C_{21} = -\frac{Cu_e^2\,\tilde{\alpha}^2}{2\Delta n \bar{C}_p T_{te}} \left(\frac{F_c - F_a}{2\Delta n}\right)$$

$$C_{22} = -\frac{\tilde{\alpha}^2}{4\Delta n \bar{C}_p} \left(\frac{C\bar{C}_p}{Pr}\right)_n - \frac{\tilde{\alpha}^2}{2(\Delta n)^2} \frac{C}{Pr} + \frac{\tilde{\alpha}\,\alpha}{4\Delta n} \frac{C}{Pr} - \frac{\tilde{\alpha} f}{4\Delta n}$$

$$- \frac{C_{1e}}{\bar{C}_p} \frac{C}{Sc} \frac{(C_{p1} - C_{p2})\tilde{\alpha}^2}{4\Delta n} \left(\frac{C_{1c} - C_{1a}}{2\Delta n}\right) - \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$C_{23} = -\frac{C_{1e}}{\bar{C}_p} \frac{C}{Sc} - \frac{(C_{p1} - C_{p2})\,\tilde{\alpha}^2}{4\Delta n} \left(\frac{\theta_c - \theta_a}{2\Delta n}\right)$$

$$C_{33} = \frac{\tilde{\alpha}\,\alpha}{4\Delta n} \frac{C}{Sc} - \frac{\tilde{\alpha}^2}{4\Delta n} \left(\frac{C}{Sc}\right)_n - \frac{\tilde{\alpha}^2}{2(\Delta n)^2} \frac{C}{Sc} - \frac{\tilde{\alpha} f}{4\Delta n} - \frac{2s\,\tilde{\alpha}\, f_s}{4\Delta n}$$

$$D_1 = \frac{\tilde{\alpha} f(F_c - F_a)}{4\Delta n} + \frac{\tilde{\alpha}^2 C_n (F_c - F_a)}{4\Delta n} + \frac{\tilde{\alpha}^2 C(F_c - 2F_b + F_a)}{2(\Delta n)^2} - \frac{\tilde{\alpha}\alpha C(F_c - F_a)}{4\Delta n}$$

$$+ \beta \frac{\rho_e}{\rho} + \frac{2sF_b^2}{\Delta s} + \frac{2s\ \tilde{\alpha}\ f_s(F_c - F_a)}{4\Delta n}$$

$$D_2 = \frac{\tilde{\alpha}^2}{\overline{C}_p} \left(\frac{C\overline{C}_p}{Pr}\right)_n \left(\frac{\theta_c - \theta_a}{4\Delta n}\right) + \frac{\tilde{\alpha}^2 (\theta_c - 2\theta_b + \theta_a)}{2(\Delta n)^2} \frac{C}{Pr}$$

$$- \frac{\alpha\tilde{\alpha} (\theta_c - \theta_a)}{4\Delta n} \frac{C}{Pr} + \frac{\tilde{\alpha} f}{4\Delta n} (\theta_c - \theta_a) + \frac{2sF_b\theta_b}{\Delta s}$$

$$+ \frac{2s\ \tilde{\alpha}\ f_s(\theta_c - \theta_a)}{4\Delta n}$$

$$D_3 = \frac{\tilde{\alpha}^2 (C_{1c} - C_{1a})}{4\Delta n} \left(\frac{C}{Sc}\right)_n + \frac{\tilde{\alpha}^2 (C_{1c} - 2C_{1b} + C_{1a})}{2(\Delta n)^2} \frac{C}{Sc}$$

$$- \frac{\alpha\tilde{\alpha}(C_{1c} - C_{1a})}{4\Delta n} \frac{C}{Sc} + \frac{\tilde{\alpha} f(C_{1c} - C_{1a})}{4\Delta n} + \frac{2sF_b C_{1b}}{\Delta s}$$

$$+ \frac{2s\ \tilde{\alpha}\ f_s(C_{1c} - C_{1a})}{4\Delta n}$$

APPENDIX B

NSBL USERS GUIDE

Program Contents

The NSBL code is made up of the following blocks:

Program NSBL: calculates conditions at the edge of the boundary layer as well as $Re_x$, s, $\beta$, $f_{inj}$, $f_w$.

Subroutine PIGYBAK: called by NSBL - calculates the initial boundary layer profile using a Faulkner-Skan solution technique.

Subroutine NUMIN: called by PIGYBAK - solves ordinary differential equations using a multi-step method with a Runge-Kutta starter.

Subroutine DERIV: called by NUMIN - calculates derivatives

Subroutine FGES: called by PIGYBAK - calculates first guesses for $f''(0)$ and $g'(0)$.

Function TWOVAR: called by FGES and EJOYCE - does linear double interpolation.

Subroutine EJOYCE: called by NSBL - calculates the downstream boundary layer profiles using finite difference methods.

Subroutine MOLIER: called by EJOYCE - calculates thermodynamic properties for air as a real gas.

Subroutine DINT2: called by MOLIER - does double interpolation.

Subroutine DINT1: called by DINT2 - does double interpolation.

## Description of Input

The definitions of the input parameters are given below. Figure B-1 presents a flowchart of the input sequence with the formats used.

MM number of x-points to be calculated

N number of y-points in the boundary layer

KK equals zero for two-dimensional flow; equals one for axisymmetric flow

ISEN equals zero for variable entropy flow; equals one for isentropic flow

IPERF equals zero for real-gas properties; equals one for perfect-gas properties

NTEMP number of elements in the transport property temperature vector

NPRES number of elements in the transport property pressure vector

DDSTR displacement thickness due to injection at the first station

| | |
|---|---|
| FLCW | value of the stream function at the wall at the first station |
| FINJ | value of the stream function of the injectant species at the first station |
| WMI | molecular weight of the injectant species |
| WMS | molecular weight of the stream species |
| TEMP(M) | Mth temperature of the transport property temperature vector, °R (temperatures must be in increasing order) |
| PRES(M) | Mth pressure of the transport property pressure vector, $\log_{10}$ (lbf/ft$^2$), (pressures must be in increasing order) |
| VISS(I,J) | viscosity of the stream species at TEMP(I) and PRES(J), lbf sec/ft$^2$ |
| TCS(I,J) | thermal conductivity of the stream species at TEMP(I) and PRES(J), Btu/ft sec °R |
| CPS(I,J) | specific heat of the stream species at TEMP(I) and PRES(J), Btu/slug °R |
| VISI(I,J) | viscosity of the injectant species at TEMP(I) and PRES(J), lbf sec/ft$^2$ |
| TCI(I,J) | thermal conductivity of the injectant species of TEMP(I) and PRES(J), Btu/ft sec °R |
| CPI(I,J) | specific heat of the injectant species at TEMP(I) and PRES(J), Btu/slug °R |
| E | convergence criterion on a boundary condition (typically 0.0005) |

| | |
|---|---|
| DELT | step size in the y-direction (typically .05) |
| EPS | convergence criterion when a variable step size is used |
| PTE(1) | total pressure behind the shock, $lbf/ft^2$ |
| TTE(1) | total temperature behind the shock, °R |
| HTREF | reference value of the heat transfer coefficient, $Btu/ft^2$ sec °R |
| RGAS | gas constant of the stream species, ft lbf/lbm °R |
| GK | ratio of the specific heats of the stream species |
| PRATIO | ratio of the local edge pressure to PTE(1) |
| TW | wall temperature, °R |
| ROVIN | density of the injectant species, $slug/ft^3$ |
| RDS | radius of the equivalent body of revolution, ft |
| X | streamwise wetted distance from the stagnation point, ft |
| SEL | local entropy at the edge of the boundary layer divided by the stream species gas constant, $S_e$/RGAS |
| SLE | total entropy behind the shock minus the local entropy at the edge of the boundary layer, Btu/lbm °R |
| SELT | total entropy behind the shock, Btu/lbm °R |
| IINIT | equals zero if the initial profile is to be calculated in the code; equals one if the initial profile is to be input |
| ALF1 | initial value of the scale factor, α, at the x-location of the input profile |

REXINT(1) integrated value of the Reynolds number at the x-location of the input profile

S(1) transformed x-coordinate at the x-location of the input profile

BETA(1) pressure gradient parameter at the x-location of the input profile

F(I) nondimensional velocity ($u/u_e$) at the $I^{th}$ point in the boundary layer

THETA(I) nondimensional temperature ($T/T_{te}$) at the $I^{th}$ point in the boundary layer

CS(I) mass fraction of the stream species at the $I^{th}$ point in the boundary layer

## Description of Output

The output for the NSBL code includes the input parameters, the free-stream parameters, the boundary layer profiles, and the boundary layer parameters. The output for the free-stream flow includes:

M station number in the x-direction

PE(M) static pressure at the edge of the boundary layer, $lbf/ft^2$

PTE(M) total pressure at the edge of the boundary layer, $lbf/ft^2$

THETAE(M) temperature ratio ($T_e/T_{te}$) at the edge of the boundary layer

TTE(M) total temperature at the edge of the boundary layer, °R

EMACH(M) Mach number at the edge of the boundary layer

UE(M) velocity at the edge of the boundary layer, ft/sec

URENOE(M) unit Reynolds number at the edge of the boundary layer, 1/ft

REXINT(M) Reynolds number at the edge of the boundary layer integrated over the wetted distance from the stagnation point

SEL(M) local entropy at the edge of the boundary layer divided by the stream species gas constant

S(M) transformed x-coordinate

ROVIN(M) density of the injectant species, slug/ft$^3$

BETA(M) pressure gradient parameter, β

FLCW(M) value of the stream function at the wall

The output for the boundary layer profiles and parameters for M-1 x-stations includes:

M station number in the x-direction

X(M) physical wetted distance from the stagnation point, ft

I station number in the y-direction

YN(I) transformed η-coordinate, η, at the I$^{th}$ point in the boundary layer

| | |
|---|---|
| Y(I) | physical y-coordinate at the $I^{th}$ point in the boundary layer, ft |
| ETA(I) | transformed y-coordinate, $\eta$, at the $I^{th}$ point in the boundary layer |
| F(I) | nondimensional velocity at the $I^{th}$ point in the boundary layer, $u(I)/u_e(M)$ |
| THETA(I) | nondimensional temperature at the $I^{th}$ point in the boundary layer, $T(I)/T_{te}(M)$ |
| CS(I) | mass fraction of the stream species at the $I^{th}$ point in the boundary layer |
| TAU | local skin friction, $lbf/ft^2$ |
| CF | local skin-friction coefficient |
| HCOEF | local heat-transfer coefficient |
| TREC | recovery temperature, °R |
| QDOT | local convective heat-transfer rate, $Btu/ft^2$ sec |
| STNO | local Stanton number |
| HRATIO | HCOEF/HTREF |
| DELST | local displacement thickness, ft |
| THMOM | local momentum thickness, ft |
| DSTAR | local displacement thickness with mass injection, ft |

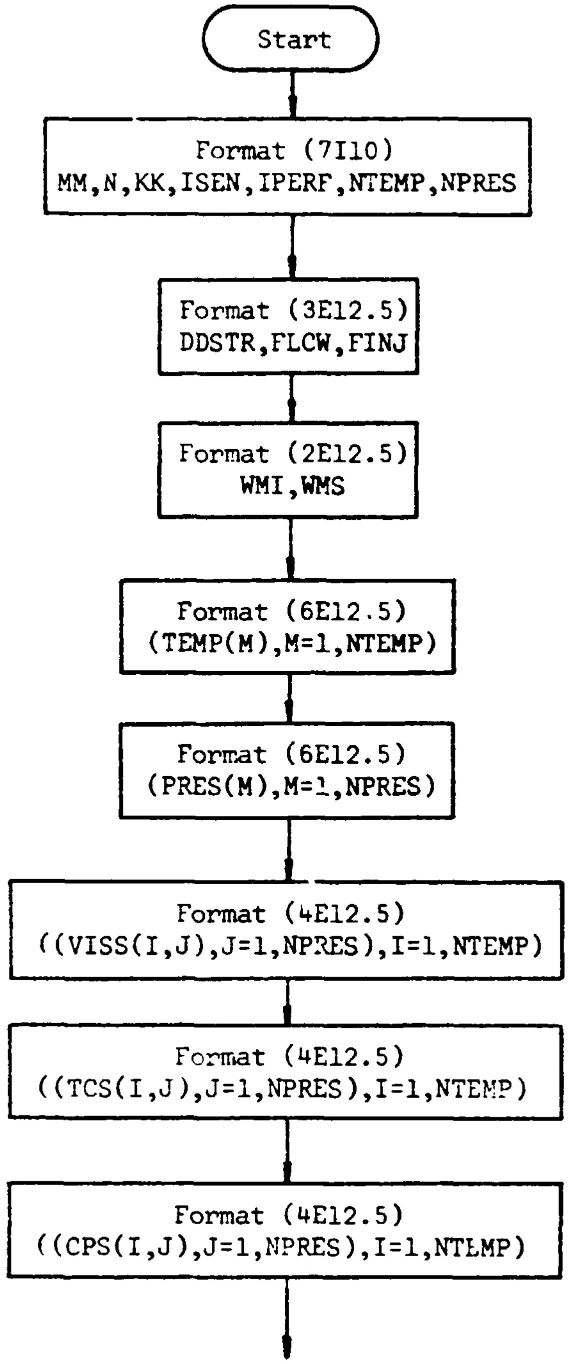



Figure B-1. - Flowchart of the input sequence and FORMATS for the NSBL code.

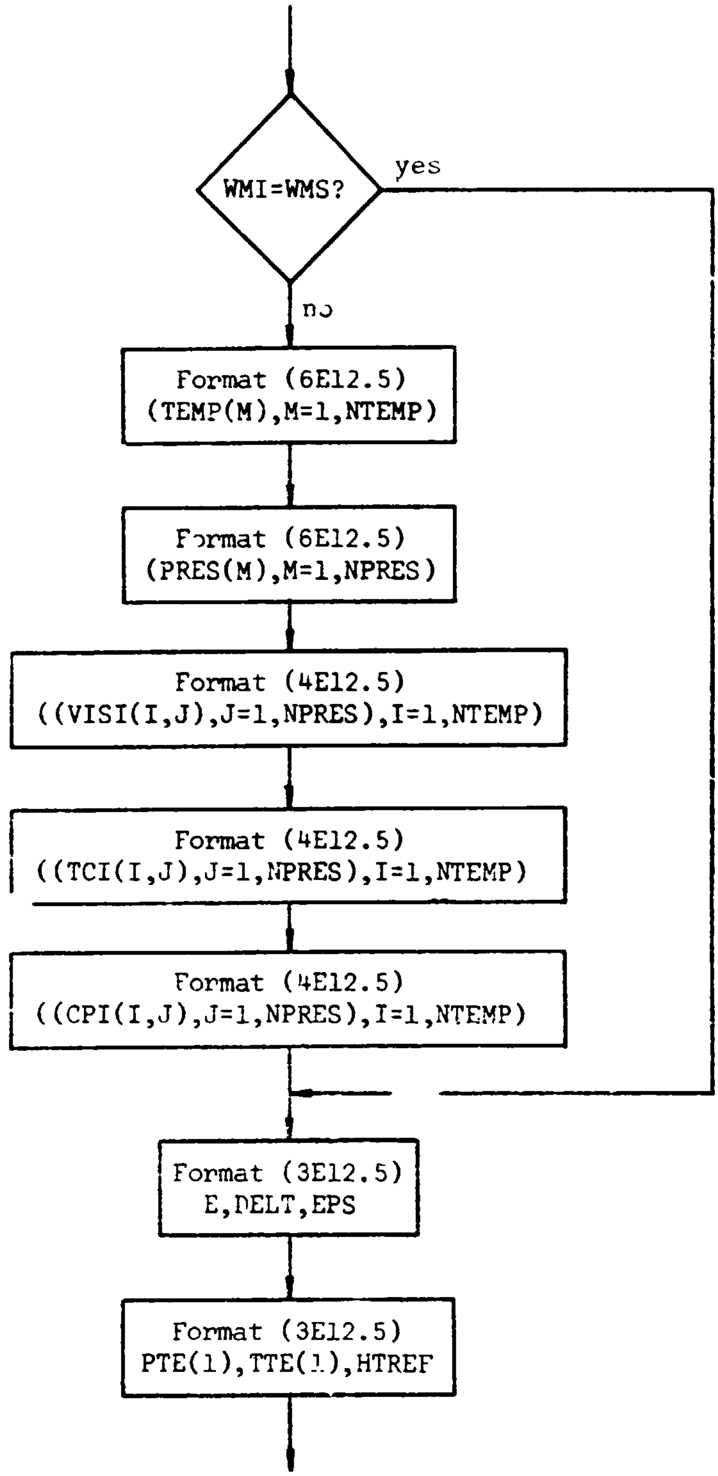



Figure B-1. - Continued

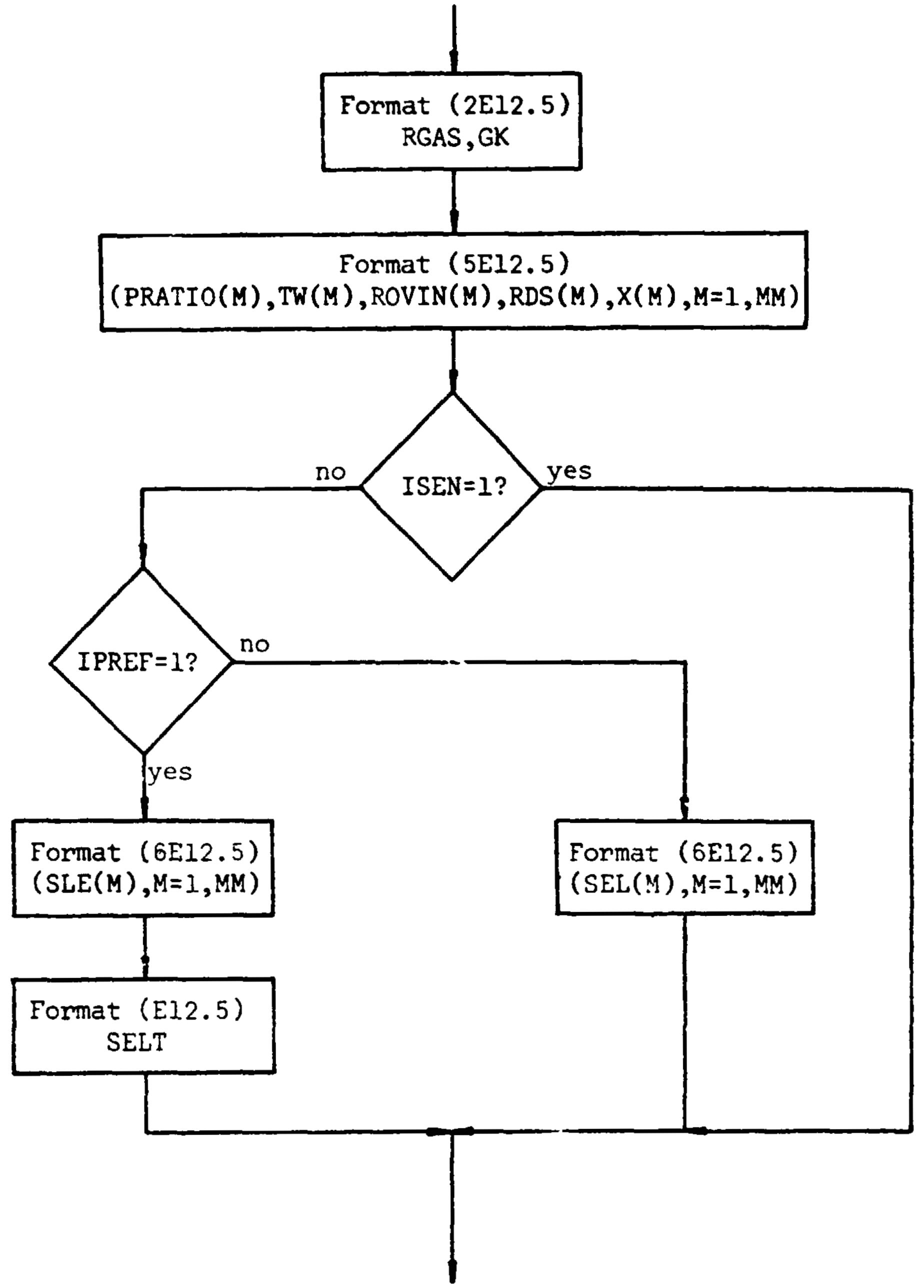
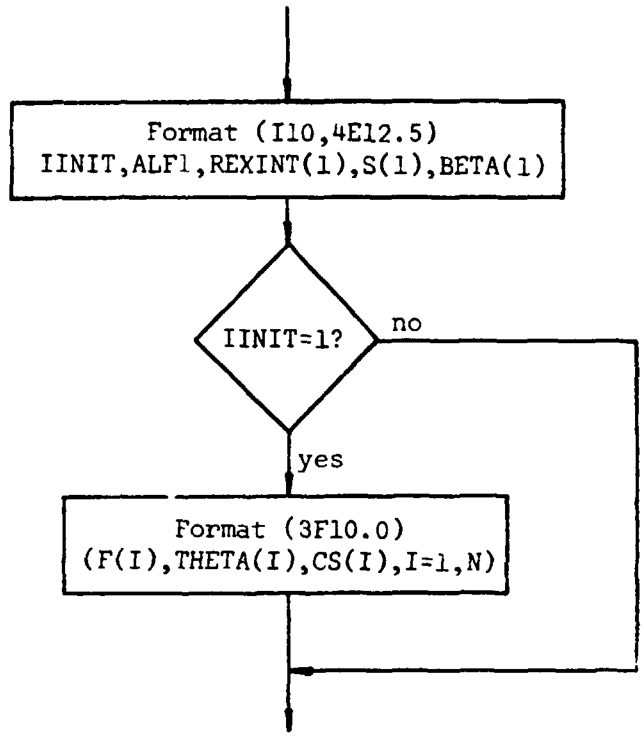



Figure B-1. - Continued

ORIGINAL PAGE IS
OF POOR QUALITY




End of the input sequence

Figure B-1. - Conclusion.

## APPENDIX C

## THERMODYNAMIC AND TRANSPORT PROPERTIES

### Calculation Procedures for a Perfect Gas

Before one can solve equations (12), (13), and (14) for the independent variables F, θ, and $C_1$, it is necessary to evaluate the thermodynamic properties which appear in the coefficients. The thermodynamic properties of the individual species are readily computed as a function of pressure and temperature. For cases of mass injection or when the free stream is assumed to be a perfect gas, the properties of the mixture are approximated with the ideal gas relations [10] once the relative concentrations of the stream species and of the injectant species are known.

DENSITY: The density of the mixture is calculated using the relation

$$\rho = \frac{P}{\bar{R}\ T} \quad \text{(C-1a)}$$

which can be written for a mixture of two gases as:

$$\rho = \frac{P}{R\ T} \frac{M_1 M_2}{M_1 + C_1(M_2 - M_1)} \quad \text{(C-1b)}$$

where, to have consistent units, the value of the universal gas constant, R, is

$$1545 \frac{\text{ft lbf}}{\text{lb mole } ^\circ\text{R}} .$$

The transport properties (specific heat, viscosity, and thermal conductivity) for a given component are computed by linearly interpolating the input tables as a function of temperature (the transport properties are independent of pressure for a perfect gas).

SPECIFIC HEAT: The specific heat of the mixture is computed using

$$\bar{C}_p = C_{p2} + C_1(C_{p1} - C_{p2}) \qquad \text{(C-2)}$$

VISCOSITY: The viscosity of the mixture is computed using the relation [11]:

$$\mu = \frac{\mu_1}{1 + G_{12}\,\dfrac{X_2}{X_1}} + \frac{\mu_2}{1 + G_{21}\,\dfrac{X_1}{X_1}} \qquad \text{(C-3)}$$

where

$$X_1 = \frac{C_1 M_2}{M_1 + C_1(M_2 - M_1)}$$

$$X_2 = 1 - X_1$$

$$G_{12} = \frac{\left[1 + \left(\dfrac{\mu_1}{\mu_2}\right)^{0.5}\left(\dfrac{M_2}{M_1}\right)^{0.25}\right]^2}{2.828\left(1 + \dfrac{M_1}{M_2}\right)^{0.5}}$$

and

$$G_{21} = \frac{\left[1 + \left(\dfrac{\mu_2}{\mu_1}\right)^{0.5}\left(\dfrac{M_1}{M_2}\right)^{0.25}\right]^2}{2.828\left(1 + \dfrac{M_2}{M_1}\right)^{0.5}}$$

THERMAL CONDUCTIVITY: The thermal conductivity of the mixture is computed using:

$$k = \frac{k_1}{1 + 1.065\, G_{12} \frac{X_2}{X_1}} + \frac{k_2}{1 + 1.065\, G_{21} \frac{X_1}{X_2}} \qquad \text{(C-4)}$$

Where $G_{12}$, $G_{21}$, $X_1$, and $X_2$ have been defined above.

DIFFUSION COEFFICIENT: The binary diffusion coefficient is computed using the procedure outlined in [22] for diffusion in a multicomponent mixture:

$$D_{ij} = \frac{\bar{D}}{F_i\, F_j} \qquad \text{(C-5a)}$$

Where $\bar{D}$ is a reference coefficient which can be approximated by:

$$\bar{D} = \frac{cT^{1.5}}{p} \qquad \text{(C-5b)}$$

Further,

$$F_i = \left(\frac{M_i}{26}\right)^{0.461} \qquad \text{(C-5c)}$$

and

$$F_j = \left(\frac{M_j}{26}\right)^{0.461} \qquad \text{(C-5d)}$$

The constant of equation (C-5b) was evaluated by fitting the equation to the tabulated values for binary diffusion coefficients. Specifically, the diffusion coefficients for mixtures of $N_2$:$CH_4$.

$H_2:CO_2$, $N_2:H_2$, Air:$CO_2$, $N_2:N_2$, for temperatures from 520°R to 12,000°R, as tabulated in [22]. Using a value of $c = 2.64 \times 10^{-8}$, the calculated values were within 20% of the tabulated values over the entire range of pressure and temperature considered. Thus, the equation used in the numerical code is:

$$D_{ij} = \frac{2.64 \times 10^{-8}\ T^{1.5}}{p(F_i)(F_j)} \qquad \text{(C-6)}$$

DIMENSIONLESS PARAMETERS: Once the dimensional thermodynamics of the mixture have been c[illegible]ed, it is a simple matter to compute the Chapman-Rubesin factor, $\frac{\rho\mu}{\rho_e\mu_e}$, the Prandtl number, $\frac{\mu c_p}{k}$, and the Schmidt number, $\frac{\rho D_{12}}{\mu}$.

Calculation Procedures for a Real Gas

For cases where there is no mass injection, the free-stream species may, at the user's option, be assumed to behave as a real gas. When this is the case, the density of the gas is calculated by the subroutine MOLI[illegible]R. The transport properties are computed by linearly interpolating the input tables as a function of temperature and pressure.

## REFERENCES


1. Eckert, E.R.G.: "Engineering Relations for Friction and Heat Transfer to Surfaces in High Velocity Flow," Journal of the Aeronautical Sciences, Aug 1955, Vol 22, No 8, pp 585-587.

2. Bertin, J.J.: "The Effects of Protuberances, Cavities, and Angle of Attack on the Wind-Tunnel Pressure and Heat-Transfer Distribution for the Apollo Command Module," NASA TMX-1243, Oct 1966.

3. Bertin, J.J.: "Boundary Layer Analysis with Similar Transpiration -- A Computer Code," Aerospace Engineering Report 70001, Feb 1970, University of Texas at Austin.

4. Hayes, W.D., and Probstein, R.F.: Hypersonic Flow Theory, Academic Press, New York, 1969.

5. Bertin, J.J., and Byrd, O.E.,Jr.: "The Anlaysis of a Nonsimilar Boundary Laycr - A Computer Code (NCNSIMBL)," Aerospace Engineering Report 70002, Aug 1970, University of Texas at Austin.

6. Marvin, J.G., and Sheaffer, Y.S.: "A Method for Solving the Nonsimilar Laminar Boundary-Layer Equations Including Foreign Gas Injection," NASA TND-5516, Nov 1969.

7. Bertin, J.J., and Goodrich, W.D.: "Effects of Surface Temperature and Reynolds Number on Leeward Shuttle Heating," Journal of Spacecraft and Rockets, Aug 1976, Vol 13, No 8, pp 473-480.

8. White, F.M.: Viscous Fluid Flow, McGraw-Hill Book Company, New York, 1974.

9. Cooke, J.C.: "An Axially Symmetric Analogue for General Three-Dimensional Boundary Layers," British Aeronautical Research Council, R&M 3200, 1961.

10. van Wylen, G.J.: Thermodynamics, John Wiley and Sons, Inc., New York, 1959.

11. Dorrance, W.H.: Viscous Hypersonic Flow, McGraw-Hill Book Company, New York, 1962.

12. Sills, J.A.: "Transformations for Infinite Regions and Their Applications to Flow Problems," AIAA Journal, Jan 1969, Vol 7, No 1, pp 117-123.

13. Becker, E.B., and Brisbane, J.J.: "Application of the Finite Element Method to Stress Analysis of Solid Propellant Rocket Grains," Vol 1, Rohm and Haas Company Report No S-76, Nov 1965.

14. Hayasi, N.: "Displacement Thickness of the Boundary Layer with Blowing," AIAA Journal, Dec 1965, Vol 3, No 12, pp 2348-2349.

15. Schlichting, H.: Boundary Layer Theory, McGraw-Hill Book Company, New York 1960.

16. Blasius, H.: "Boundary Layers in Fluids with Small Friction," English translation, NACA Technical Memorandum 1256, 1950.

17. Detra, R.W., Kemp, N.H., and Riddell, F.R.: "Addendum to Heat Transfer to Satellite Vehicles Re-entering the Atmosphere," Jet Propulsion, Dec 1957, Vol 27, No 12, pp 1256-1257.

18. Goodrich, W.D.: Private communication, Aug 1977.

19. Bartlett, E.P., and Kendall, R.M.: "An Analysis of the Coupled Chemically Reacting Boundary Layer and Charring Ablator, Part III, Nonsimilar Solution of the Multicomponent Laminar Boundary Layer by an Integral Matrix Method," NASA CR-1062, June 1968.

20. Moeckel, W.E., and Weston, K.C.: "Composition and Thermodynamic Properties of Air in Chemical Equilibrium," NACA Technical Note 4265, Apr 1958.

21. Hansen, C.F.: "Approximations for the Thermodynamic and Transport Properties of High-Temperature Air," NASA TR R-50, 1959.

22. Bartlett, E.P., Kendall, R.M., and Rindal, R.A.: "An Analysis of the Coupled Chemically Reacting Boundary Layer and Charring Ablator, Part IV - A Unified Approximation for Mixture of Transport Properties for Multi-component Boundary-Layer Applications," NASA CR-1063, June 1968.

```
      PROGRAM NSBL (INPUT,OUTPUT)
C
C     PROGRAM NSBL WAS WRITTEN FOR THE CDC 6600 AT THE
C     UNIVERSITY OF TEXAS AT AUSTIN.  THIS PROGRAM WAS
C     MODIFIED FROM AN EARLIER VERSION (NONSIMBOL) TO ITS
C     PRESENT FORM BY DENNIS STALMACH DURING THE FALL OF
C     1977.
C
C     THIS PROGRAM SOLVES FOR THE LAMINAR NONSIMILAR
C     BOUNDARY LAYER ON A 2-D SURFACE OR A BODY OF
C     REVOLUTION USING A FINITE DIFFERENCE TECHNIQUE.
C     IT CAN SOLVE FOR COMPRESSIBLE OR INCOMPRESSIBLE,
C     ISENTROPIC OR VARIABLE ENTROPY, REAL OR PERFECT
C     GAS FLOWS WITH ARBITRARY WALL TEMPERATURES AND AN
C     ARBITRARY FREE STREAM GAS.  THE PROGRAM CAN ALSO
C     TAKE INTO ACCOUNT MASS INJECTION WITH ARBITRARY
C     INJECTANT, INJECTION RATE, AND INJECTION RATE
C     DISTRIBUTION IF BOTH THE FREE STREAM AND INJECT-
C     ANT ARE PERFECT GASES.
C
C     THE INITIAL VELOCITY AND TEMPERATURE DISTRIBUTIONS
C     MAY BE INPUT, OTHERWISE SUBROUTINE PIGYBAK SOLVES
C     FOR THEM.  PIGYBAK USES PERFECT GAS RELATIONS AND
C     ASSUMES PR=.7.
C
C
      DIMENSION STE(60), GAMMA(60), Z(60), SLE(60), SEL(60), GAMTE(60),
     1RHOTE(60), ZT(60)
      DIMENSION HTE(60), PTE(60), HE(60), DELX(60), DDSTR(60), TW(60), P
     1RATIO(60)
      DIMENSION PE(60), ROVIN(60), THETAW(60), RDS(60), EMACH(60), TE(60
     1), VISE(60), RHOE(60), THETAE(60), UE(60), URENOE(60), REXINT(60),
     2 S(60), DS(60), BETA(60), X(60), FLCW(60), FINJ(60), EA(60)
      DIMENSION XB(2), XDATA(2,120)
      COMMON /PROP/ TEMP(40),PRES(4),NTEMP,NPRES
      COMMON /NEW/ F(50),THETA(50),ALF,N,KK,MM,CS(50),X,IPERF,NTEMP1
      COMMON /NEW2/ TTE(60),WMI,WMS,RDS,DDSTR
      COMMON /NEW3/ FLCW,THETAE,ROVIN,BETA,DS,S,UE,RHOE
      COMMON /NEW4/ PE,THETAW,TE,VISE,HTREF
      COMMON /NEW10/ KEY,VISS(160),VISI(160),TCS(160),TCI(160),CPI(160),
     1CPS(160)
      COMMON /CONST/ A1,A2,A3,A4,F,EPS,DELT,IERR
      COMMON /MAIN/ TWI,UEI,TTI
      COMMON /DDS/ BETA1
C
      KEY=0
      READ 270, MM,N,KK,ISEN,IPERF,NTEMP,NPRES
      READ 420, DDSTR(1),FLCW(1),FINJ(1)
      READ 420, WMI,WMS
      IF (ABS(WMI-WMS).LT.1.E-8) KEY=1
      READ 420, (TEMP(M),M=1,NTEMP)
      READ 420, (PRES(M),M=1,NPRES)
      NTP=NTEMP*NPRES
      READ 300, (VISS(I),I=1,NTP)
      READ 300, (TCS(I),I=1,NTP)
      READ 300, (CPS(I),I=1,NTP)
      IF (KEY.EQ.1) GO TO 10
      READ 300, (VISI(I),I=1,NTP)
      READ 300, (TCI(I),I=1,NTP)
      READ 300, (CPI(I),I=1,NTP)
   10 READ 420, E,DELT,EPS
```

```
C
      MMAX=MM-1
      PRINT 280
      PRINT 290
      NSPACE=(40-MM)/6
      IF (NSPACE.LT.0) NSPACE=0
      DO 20 IP=1,NSPACE
         PRINT 290
   20 CONTINUE
      PRINT 320
      PRINT 330, MM,K
      IF (KK.EQ.0) GO TO 30
      IF (KK.EQ.1) GO TO 40
      PRINT 340
      STOP
   30 PRINT 350
      GO TO 50
   40 PRINT 360
   50 PRINT 370, WMI,WMS
      IF (IPERF.EQ.1) GO TO 60
      PRINT 380
      GO TO 70
   60 PRINT 390
   70 PRINT 400, DDSTW(1),FLCW(1),FINJ(1)
      PRINT 410, E,DELT,EPS
C
C     CALCULATION OF CONDITIONS AT THE EDGE OF THE BOUNDARY LAYER
C
      READ 420, PTE(1),TTE(1),HTREF
      READ 420, RGAS,GK
      READ 310, (PRATIO(M),TW(M),ROVIN(M),RDS(M),X(M),M=1,MM)
C
      IF (ISEN.EQ.0) GO TO 110
      IF (IPERF.EQ.1) GO TO 90
      CALL MOLIER (HTE(1),PTE(1),2,TTE(1),Z,STE(1),RHO,GAMMA)
      DO 80 M=1,MM
         SLE(M)=0.0
         TTE(M)=TTE(1)
         HTE(M)=HTE(1)
         SEL(M)=STE(1)
         PTE(M)=PTE(1)
         PE(M)=PRATIO(M)*PTE(M)
         CALL MOLIER (HE(M),PE(M),1,TE(M),Z(M),SEL(M),RHOE(M),GAMMA(M))
         UE(M)=SQRT(2.*(HTE(M)-HE(M))*25031.0)
         EA(M)=SQRT(GAMMA(M)*32.174*RGAS*TE(M)*Z(M))
         EMACH(M)=UE(M)/EA(M)
         PLOG=ALOG10(PE(M))
         VISE(M)=TWOVAR(NPRES,NTEMP,PLOG,TE(M),PRES,TEMP,VISS)
         THETAW(M)=TW(M)/TTE(1)
         THETAE(M)=TE(M)/TTE(1)
         URENOE(M)=RHOE(M)*UE(M)/VISE(M)
   80 CONTINUE
      GO TO 160
   90 CONTINUE
      DO 100 M=1,MM
         SLE(M)=0.0
         SEL(M)=0.0
         HTE(M)=TTE(1)*CPS(1)/32.174
         PTE(M)=PTE(1)
         TTE(M)=TTE(1)
         PE(M)=PRATIO(M)*PTE(1)
         EMACH(M)=SQRT(((1./PRATIO(M))**((GK-1.)/GK)-1.)*(2./(GK-1.)))
```

```
          TE(M)=TTE(1)/(1.+(GK-1.)*EMACH(M)**2/2.)
          THETAE(M)=TE(M)/TTE(1)
          THETAW(M)=TW(M)/TTE(1)
          HE(M)=TE(M)*CPS(1)/32.174
          RHOE(M)=PE(M)/(TE(M)*32.174*RGAS)
          EA(M)=SQRT(GK*32.174*RGAS*TE(M))
          UE(M)=EA(M)*EMACH(M)
          PLOG=ALOG10(PE(M))
          VISE(M)=TWOVAR(NPRES,NTEMP,PLOG,TE(M),PRES,TEMP,VISS)
          URENOE(M)=RHOE(M)*UE(M)/VISE(M)
  100 CONTINUE
      GO TO 160
  110 CONTINUE
      PTE1=PTE(1)
      TTE1=TTE(1)
      IF (IPERF.EQ.1) GO TO 130
C
      READ 420, (SEL(M),M=1,MM)
C
C     SEL(M) IS READ IN IN TERMS OF S/R
C
      CALL MOLIER (HTE1,PTE1,2,TTE1,ZTE1,STE1,RHOTE1,GAMTE1)
      DO 120 M=1,MM
          HTE(M)=HTE1
          PE(M)=PRATIO(M)*PTE1
          CALL MOLIER (HE(M),PE(M),1,TE(M),Z(M),SEL(M),RHOE(M),GAMMA(M))
          EA(M)=SQRT(GAMMA(M)*32.174*RGAS*TE(M)*Z(M))
          UE(M)=SQRT((HTE1-HE(M))*50062.8)
          EMACH(M)=UE(M)/EA(M)
          PLOG=ALOG10(PE(M))
          VISE(M)=TWOVAR(NPRES,NTEMP,PLOG,TE(M),PRES,TEMP,VISS)
          URENOE(M)=RHOE(M)*UE(M)/VISE(M)
          CALL MOLIER (HTE(M),PTE(M),3,TTE(M),ZT(M),SEL(M),RHOTE(M),GAMTE
     1    (M))
          THETAW(M)=TW(M)/TTE(M)
          THETAE(M)=TE(M)/TTE(M)
  120 CONTINUE
      GO TO 160
  130 CONTINUE
C
C     SLE(M) IS READ IN UNITS OF BTU/LBM R
C
      READ 420, (SLE(M),M=1,MM)
      READ 420, SELT
C
      DO 140 M=1,MM
          SEL(M)=(SELT-SLE(M))*778.16/RGAS
          SLE(M)=SLE(M)*25031.4
  140 CONTINUE
C
C     SLE(M) UNITS ARE NOW IN FT LBF/SLUG R
C     SEL(M) IS THE LOCAL SE/R
C
      HTE1=CPS(1)*TTE1/32.174
      DO 150 M=1,MM
          TTE(M)=TTE1
          HTE(M)=HTE1
          PE(M)=PRATIO(M)*PTE1
          THETAE(M)=EXP(1./(CPS(1)*778.16)*(-SLE(M)+32.174*RGAS*ALOG(PRAT
     1    IO(M))))
          TE(M)=THETAE(M)*TTE1
          PLOG=ALOG10(PE(M))
```

```
      VISE(M)=TWOVAR(NPRES,NTEMP,PLOG,TE(M),PRES,TEMP,VISS)
      HE(M)=TE(M)*CPS(1)/32.174
      EA(M)=SQRT(GK*32.174*RGAS*TE(M))
      THETAW(M)=TW(M)/TTE1
      UE(M)=SQRT((HTE1-HE(M))*50062.0)
      EMACH(M)=UE(M)/EA(M)
      PTE(M)=PE(M)*((1.+(GK-1.)/2.*EMACH(M)*EMACH(M))**(GK/(GK-1.)))
      RHOE(M)=(PRATIO(M)*PTE1)/(TE(M)*RGAS*32.174)
      URENOE(M)=RHOE(M)*UE(M)/VISE(M)
  150 CONTINUE
  160 CONTINUE
      PRINT 430
      PRINT 440
      PRINT 450, (HTE(M),PRATIO(M),TW(M),RDS(M),X(M),ROVIN(M),M=1,MM)
      PRINT 280
      NSPACE=(57-MM)/6
      IF (NSPACE.LT.0) NSPACE=0
      DO 170 I=1,NSPACE
         PRINT 290
  170 CONTINUE
      PRINT 460
      PRINT 470
      PRINT 480, (M,X(M),PE(M),PTE(M),THETAE(M),TTE(M),EMACH(M),M=1,MM)
      PRINT 280
      NSPACE=(62-MM)/6
      IF (NSPACE.LT.0) NSPACE=0
      DO 180 I=1,NSPACE
         PRINT 290
  180 CONTINUE
      PRINT 490
C
      READ 530, TINIT,ALF1,REXINT(1),S(1),BETA(1)
C
      IF (TINIT.EQ.1) GO TO 190
      REXINT(1)=URENOE(1)*X(1)
      S(1)=RHOE(1)*VISE(1)*UE(1)*((RDS(1)**KK)**2)*X(1)/(3.0**KK)
  190 M=1
      PRINT 500, M,X(M),UE(M),URENOE(M),REXINT(M),SEL(M)
C
C     DOWNSTREAM STATIONS
C
      DO 200 M=2,MM
         DELX(M)=X(M)-X(M-1)
         REXINT(M)=REXINT(M-1)+(URENOE(M-1)+URENOE(M))*DELX(M)/2.0
         PRINT 500, M,X(M),UE(M),URENOE(M),REXINT(M),SEL(M)
         DDSTR(M)=(ROVIN(M)*(RDS(M)**KK)/(RHOE(M)*UE(M))+ROVIN(M-1)*(RDS
     1   (M-1)**KK)/(RHOE(M-1)*UE(M-1)))*DELX(M)/(2.0*(RDS(M)**KK))+DDST
     2   R(M-1)
C
C     EVALUATION OF S
C
         S(M)=S(M-1)+0.5*(RHOE(M-1)*UE(M-1)*VISE(M-1)*((RDS(M-1)**KK)**2
     1   )+RHOE(M)*UE(M)*VISE(M)*((RDS(M)**KK)**2))*DELX(M)
  200 CONTINUE
      DS(1)=S(2)-S(1)
      IF (TINIT.EQ.1) GO TO 210
      BETA(1)=(2.0*(S(2)+S(1))/(UE(2)+UE(1)))*((UE(2)-UE(1))/(S(2)-S(1))
     1)
  210 BETA1=BETA(1)
      DO 220 M=2,MMAX
         DS(M)=(S(M+1)-S(M-1))/2.0
         BETA(M)=(2.0*S(M)*(UE(M+1)-UE(M-1)))/(UE(M)*(S(M+1)-S(M-1)))
```

```
  220 CONTINUE
C
C     EVALUALTION OF FLC AT THE WALL
C
      DO 230 M=2,MM
         FINJ(M)=FINJ(M-1)+(ROVIN(M-1)*(RDS(M-1)**KK)+ROVIN(M)*(RDS(M)**
     1   KK))*DELX(M)/2.0
         FLCW(M)=-FINJ(M)/(SQRT(2.0*S(M)))
  230 CONTINUE
C
      PRINT 280
      NSPACE=(62-MM)/6
      IF (NSPACE.LT.0) NSPACE=0
      DO 240 I=1,NSPACE
         PRINT 290
  240 CONTINUE
      PRINT 510
      PRINT 520, (M,X(M),S(M),ROVIN(M),BETA(M),FLCW(M),M=1,MMAX)
C
C     CALCULATING THE FLOW PARAMETERS AT THE INITIAL STATION
C
      IF (IINIT.EQ.0) GO TO 250
      ALF=ALFI
C
      READ 540, (F(I),THETA(I),CS(I),I=1,N)
C
      GO TO 260
  250 M=1
      UEI=UE(1)
      HTEI=HTE(1)
      TWI=TW(1)
      TTI=TTE(1)
      CALL PIGYBAK (HTEI)
  260 CALL EJUYCE
C
  270 FORMAT (7I10)
  280 FORMAT (1H1)
  290 FORMAT (/)
  300 FORMAT (4E12.5)
  310 FORMAT (5E12.5)
  320 FORMAT (56X,24(1H*),/,56X,24H* THE INPUT PARAMETERS *,/,56X,24(1H*
     1),//)
  330 FORMAT (48X,9HTHERE ARE,I3,28H STATIONS IN THE X DIRECTION,/,48X,9
     1HTHERE ARE,I3,28H STATIONS IN THE Y DIRECTION,/)
  340 FORMAT (5X,10(1H*),116HYOU HAVE INPUT AN ERRONEOUS KK, KK MUST EQU
     1AL 0 FOR A TWO-DIMENSIONAL CONFIGURATION, OR 1 FOR A BODY OF -REVO
     2LUTION-)
  350 FORMAT (56X,43HTHE BODY IS A TWO-DIMENSIONAL CONFIGURATION)
  360 FORMAT (52X,32HTHE BODY IS A BODY OF REVOLUTION)
  370 FORMAT (/,37X,40HTHE MOLECULAR WEIGHT OF THE INJECTANT IS,F6.2,13H
     1 LBM/LBM-MOLE,/,38X,38HTHE MOLECULAR WEIGHT OF THE SPECIES IS,F6.2
     2,13H LBM/LBM-MOLE,/)
  380 FORMAT (57X,21HTHE GAS IS A REAL GAS,/)
  390 FORMAT (55X,24HTHE GAS IS A PERFECT GAS,/)
  400 FORMAT (42X,9HDDSTW(1)=,E12.5,10H, FLCW(1)=,E12.5,10H, FINJ(1)=,E1
     12.5,/)
  410 FORMAT (43X,2HE=,E12.5,7H, DELT=,E12.5,6H, EPS=,E12.5,/)
  420 FORMAT (6F12.5)
  430 FORMAT (1X,134(1H*))
  440 FORMAT (1H0,17X,6HHTE(M),14X,9HPRATIO(M),11X,5HTW(M),14X,6HRDS(M),
     113X,4HX(M),15X,8HROVIN(M))
  450 FORMAT (14X,E12.5,5(3X,1H*,3X,E12.5))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  460 FORMAT (58X,11HOUTPUT DATA,///)
  470 FORMAT (21X,1HM,8X,1HX,11X,5HPE(M),8X,6HPTE(M),7X,9HTHETAE(M),6X,6
     1HTTE(M),8X,8HEMAC(M))
  480 FORMAT (17X,I5,6E14.5)
  490 FORMAT (29X,1HM,7X,4HX(M),13X,5HUE(M),7X,9HURENOE(M),5X,9HREXINT(M
     1),8X,6HSEL(M))
  500 FORMAT (26X,I5,5E14.5)
  510 FORMAT (,29X,1HM,6X,4HX(M),10X,4HS(M),9X,8HROVIN(M),6X,7HBETA(M),7
     1X,7HFLCW(M),/)
  520 FORMAT (26X,I5,5E14.5)
  530 FORMAT (I10,4E12.5)
  540 FORMAT (3F10.0)
C
      E
```

```
      SUBROUTINE PIGYBAK (HTEI)
C
C     SUBROUTINE PIGYBAK CALCULATES THE INITIAL PROFILE
C     USING THE BLAST PROCEDURE.
C
      DIMENSION F(50), G(50), FN(50), GN(50), YN(50), ETAD(50), ETA(50),
     1 THTTA(50)
      DIMENSION Y(5), FPA(3), GA(3)
      DIMENSION DV(5), P(5), TE(5), YPR(5,4), DEP(5)
      DIMENSION ER(5), DPSAVE(5), FPR(200)
      EXTERNAL DERIV
      COMMON /NEW/ F,THETA(50),ALF,NY,KK,MM,CS(50),X(60),IPERF,NTEMP1
      COMMON /CONST/ A1,A2,A3,A4,E,EPS,DELT,
      COMMON /MAIN/ THI,UEI,TTI
C
C     CAUTION *PR=0.7* INITIAL PROFILE USES PERFECT GAS RELATIONS
C
      IERR=0
      KEY=0
      A1=UEI*UEI/(2.0*HTEI*32.174*778.)
      A2=1.0-A1
      PR=0.7
      A3=1.0/PR
      A4=UEI*UEI*(PR-1.0)/(HTEI*PR*32.1      .)
      PRINT 200, A1,A2,A3,A4
      AA=0.0
      BB=0.0
      DD=THI/TTI
      CALL FGES (DD,CC,EE)
      GO TO 20
   10 CC=.999*CC
      IERR=0
   20 PRINT 210, AA,BB,CC,DD,EE
      NCOUNT=0
      IN=NY-1
      Y(1)=AA
      Y(2)=BB
      Y(3)=CC
      Y(4)=DD
      Y(5)=EE
   30 GPSAVE=Y(5)
      FPPSAV=Y(3)
      FPP01=Y(3)
      GP01=Y(5)
      FPPOS=Y(3)
      GPOS=Y(5)
      II=1
      N=5
   40 CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,1)
      T=0.
      N=5
      I=1
      CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,2)
   50 CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,3)
      IF (IERR.EQ.1) GO TO 10
      IF (ABS(Y(3)).LT.E) GO TO 60
      I=I+1
      GO TO 50
   60 IF (ABS(1.0000-Y(2)).GT.E) GO TO 70
      IF (ABS(1.0000-Y(4)).LT.E) GO TO 100
   70 FPA(II)=Y(2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GA(II)=Y(4)
      II=II+1
      IF (II.EQ.2) GO TO 80
      IF (II.NE.3) GO TO 90
      Y(1)=AA
      Y(2)=BB
      Y(3)=FPPSAV
      Y(4)=DD
      Y(5)=1.005*GPSAVE
      FPP03=Y(3)
      GP03=Y(5)
      FPPOS=Y(3)
      GPOS=Y(5)
      GO TO 40
   80 Y(1)=AA
      Y(2)=BB
      Y(3)=1.005*FPPSAV
      Y(4)=DD
      Y(5)=GPSAVE
      FPP02=Y(3)
      GP02=Y(5)
      FPPOS=Y(3)
      GPOS=Y(5)
      GO TO 40
   90 DFPA=FPA(1)-FPA(2)
      DGA=GA(1)-GA(2)
      DFPA1=FPA(1)-FPA(3)
      DGA1=GA(1)-GA(3)
      DFPP0=FPP01-FPP02
      DGP0=GP01-GP03
      A6=DFPA/DFPP0
      B1=DFPA1/DGP0
      C1=1.0000-FPA(1)
      D1=DGA/DFPP0
      E1=DGA1/DGP0
      F1=1.0000-GA(1)
      DEFPP=(C1*E1-F1*B1)/(A6*E1-D1*B1)
      DEGP=(A6*F1-D1*C1)/(A6*E1-D1*B1)
      FPP=FPP01+DEFPP
      GP=GP01+DEGP
      Y(1)=AA
      Y(2)=BB
      Y(3)=FPP
      Y(4)=DD
      Y(5)=GP
      GO TO 30
  100 CONTINUE
      Y(1)=AA
      Y(2)=BB
      Y(3)=FPPOS
      Y(4)=DD
      Y(5)=GPOS
      CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,1)
      T=0.
      N=5
      DLST1=(Y(4)-A1*(Y(2)**2))/A2-Y(2)
      THTA1=(Y(2))*(1.0-Y(2))
      THTE1=(Y(2))*(1.0-(Y(2)**2))
      III=1
      FPR(III)=Y(2)
      G(III)=Y(4)
      ETAD(III)=T
```

```
      CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,2)
      DLST2=(Y(4)-A1*(Y(2)**2))/A2-Y(2)
      THTA2=(Y(2))*(1.0-Y(2))
      THTE2=(Y(2))*(1.0-(Y(2)**2))
      ADELST=(0.5*DLST1+DLST2)*DELT
      ATHETA=(0.5*THTA1+THTA2)*DELT
      ATHETE=(0.5*THTE1+THTE2)*DELT
  110 CALL NUMIN (N,DELT,T,DV,P,TE,ER,YPR,Y,EPS,DPSAVE,KEY,DERIV,3)
      DDELST=((Y(4)-A1*(Y(2)**2))/A2-Y(2))*DELT
      DTHETA=((Y(2))*(1.0-Y(2)))*DELT
      DTHETE=((Y(2))*(1.0-(Y(2)**2)))*DELT
      ADELST=ADELST+DDELST
      ATHETA=ATHETA+DTHETA
      ATHETE=ATHETE+DTHETE
      I11=I11+1
      ETAD(I11)=T
      FPR(I11)=Y(2)
      G(I11)=Y(4)
      IF (NCOUNT.NE.0) GO TO 130
      IF (FPR(I11)-0.99) 130,120,120
C
C     CALCULATION OF ALF
C
  120 ALF=-ALOG(1./FLOAT(IN))/ETAD(I11)
      NCOUNT=1
  130 CONTINUE
      IF (ABS(Y(3)).LT.E) GO TO 140
      GO TO 110
  140 CONTINUE
C
C     CROSS STREAM STATION TRANSFORM WITH -IN- NO. OF STATIONS
C
      DO 150 I=1,NY
         YN(I)=FLOAT(I-1)/FLOAT(IN)
  150 CONTINUE
      J=1
      ETA(J)=ALOG(1.0/(1.0-YN(J)))/ALF
      DO 190 I=1,I11
  160    IF (ETA(J)-ETAD(I)) 170,190,190
  170    II=I-1
         AF=(ETA(J)-ETAD(II))/(ETAD(I)-ETAD(II))
         F(J)=FPR(II)+(FPR(I)-FPR(II))*AF
         GN(J)=G(II)+(G(I)-G(II))*AF
         CS(J)=1.0
         THETA(J)=GN(J)-A1*F(J)*F(J)
         J=J+1
         IF (J.GT.IN) GO TO 180
         ETA(J)=ALOG(1.0/(1.0-YN(J)))/ALF
         GO TO 160
  180    ETA(J)=1.0E10
  190 CONTINUE
      F(IN+1)=1.0
      GN(IN+1)=G(I11)
      THETA(IN+1)=1.0-A1
      CS(IN+1)=1.0
      RETURN
C
  200 FORMAT (2X,3HA1=,F9.5,2X,3HA2=,F9.5,2X,3HA3=,F9.5,2X,3HA4=,F9.5)
  210 FORMAT (2X,3HAA=,E12.5,2X,3HBB=,E12.5,2X,3HCC=,E12.5,2X,3HDD=,E12.
     15,2X,3HEE=,E12.5)
C
      END
```

```
      SUBROUTINE FGES (DD,CC,EE)
      COMMON /DDS/ BETA1
      DIMENSION BETAS(5), G(4), FPP(5,4), GP(5,4)
      DATA G/.24,.3,.4,.45/
      DATA BETAS/.05,.1,.2,.3,.4/
      DATA FPP/.4179121,.4745672,.5753387,.6645224,.7451,.4355791,.49263
     147,.5941355,.6837839,.76:7083,.4586763,.5159473,.6178686,.7079041,
     2.7894819,.4681788,.5254897,.6274183,.7174269,.7989913/
      DATA GP/.2419834,.2474471,.2564553,.2626474,.268268,.2335142,.2385
     1798,.2466409,.2528482,.2581825,.2121744,.21649.5,.223541,.2287362,
     2.2331344,.1988107,.2028812,.209193,.2141326,.2181627/
      NH=5
      NG=4
      CC=TWOVAR(NH,NG,BETA1,DD,BETAS,G,FPP)
      EE=TWOVAR(NH,NG,BETA1,DD,BETAS,G,GP)
      RETURN
C
      END
```

C.2

```
      SUBROUTINE NUMIN (N,DELT,T,DV,P,TE,ER,YPR,DEP,EPS,DPSAVE,KEY,DERIV
     1,INDX)
      COMMON /CONST/ A1,A2,A3,A4,E,EP,IERR
C
C     ------GENERAL NUMERICAL INTEGRATION WITH SINGLE-STEP ERROR ANALYSI
C
      DIMENSION A(4), B(5), DV(N), P(N), TE(N), ER(N)
      DIMENSION YPR(N,4), DEP(N), DPSAVE(N)
C
C     ------
C     ------
C
      EXTERNAL DERIV
C
C     -----------  DATA
C
      GO TO (10,20,50), INDX
   10 SIX=24.0
      ZERO=0.0
      A(1)=-9.0/SIX
      A(2)=37.0/SIX
      A(3)=-59.0/SIX
      A(4)=55.0/SIX
      B(1)=1.0/SIX
      B(2)=-5.0/SIX
      B(3)=19.0/SIX
      B(4)=-A(1)
      RATIO=19.0/270.0
      SIX=6.0
      TWO=2.0
      M1=4
      M2=1
      M3=2
      M4=3
      RETURN
C
C     --------   SET UP FOR INTEGRATION
C     --------          ENTRY RESET
C
   20 ASSIGN 60 TO IPLS
      KOP=0
      KOUNT=0
      DELBY6=DELT/SIX
      DELBY2=DELT/TWO
      DO 30 J=1,N
         DPSAVE(J)=DEP(J)
         DV(J)=DEP(J)
   30 CONTINUE
      CALL DERIV (T,DV,P,6,N)
      IF (IERR.EQ.1) RETURN
      DO 40 J=1,N
         YPR(J,M1)=P(J)
         ER(J)=ZERO
   40 CONTINUE
      RETURN
C
C     --------   ENTRY-POINT FOR ONE NUMERICAL INTEGRATION STEP
C
   50 V=T
      T=V+DELT
      KOP=KOP+1
```

```
      GO TO IPL5
C
C     ****** RUNGE-KUTTA PROCEDURE
C
   60 VV=V+DELBY2
      DO 70 J=1,N
         DV(J)=DEP(J)+YPR(J,M1)*DELBY2
   70 CONTINUE
      CALL DERIV (VV,DV,P,3,N)
      IF (IERR.EQ.1) RETURN
      DO 80 J=1,N
         DV(J)=DEP(J)+P(J)*DELBY2
   80 CONTINUE
      CALL DERIV (VV,DV,TE,4,N)
      IF (IERR.EQ.1) RETURN
      DO 90 J=1,N
         DV(J)=DEP(J)+TE(J)*DELT
         TE(J)=2.0*(TE(J)+P(J))
   90 CONTINUE
      CALL DERIV (T,DV,P,5,N)
      IF (IERR.EQ.1) RETURN
      DO 100 J=1,N
         DEP(J)=DEP(J)+DELBY6*(P(J)+TE(J)+YPR(J,M1))
         DV(J)=DEP(J)
  100 CONTINUE
      CALL DERIV (T,DV,P,5,N)
      IF (IERR.EQ.1) RETURN
      DO 110 J=1,N
         YPR(J,M4)=P(J)
  110 CONTINUE
      KOUNT=KOUNT+1
C
C     ****** CHECK THE NUMBER OF INTEGRATION STEPS MADE BY R-K
C            PROCEDURE
C
      IF (KOUNT.LT.3) GO TO 120
      ASSIGN 200 TO IPL5
  120 M0=M4
      M4=M3
      M3=M2
      M2=M1
      M1=M0
      IF (KEY.EQ.0) RETURN
      IF (KOP.LE.3) RETURN
      KINC=0
      DO 130 J=1,N
         IF (ER(J).GT.EPS) GO TO 150
         IF (ER(J)*10000.0.LT.EPS) KINC=KINC+1
  130 CONTINUE
      IF (KINC.EQ.N) GO TO 190
      DO 140 J=1,N
         DPSAVE(J)=DEP(J)
  140 CONTINUE
      RETURN
  150 CONTINUE
      IF (KOP.EQ.4) GO TO 180
      T=T-DELT
  160 DELT=0.5*DELT
      DO 170 J=1,N
         DEP(J)=DPSAVE(J)
  170 CONTINUE
      GO TO 20
```

```
  180 T=T-4.*DELT
      GO TO 160
  190 DELT=2.*DELT
      GO TO 20
C
C     ******** ADAMS PREDICTOR-CORRECTOR PROCEDURE
C
  200 DO 210 J=1,N
         TE(J)=B(3)*YPR(J,M1)+B(2)*YPR(J,M2)+B(1)*YPR(J,M3)
         DV(J)=DEP(J)+DELT*(A(4)*YPR(J,M1)+A(3)*YPR(J,M2))
         DV(J)=DV(J)+DELT*(A(2)*YPR(J,M3)+A(1)*YPR(J,M4))
C
C     ******** SAVE PREDICTED VALUE
C
         ER(J)=DV(J)
  210 CONTINUE
      CALL DERIV (T,DV,F,1,N)
      IF (IERR.EQ.1) RETURN
C
C     ********
C     ********  DOUBLE PRECISION ADAMS MOULTON PECEC
C
      DO 220 J=1,N
         DV(J)=DEP(J)+DELT*(B(4)*F(J)+TE(J))
  220 CONTINUE
      CALL DERIV (T,DV,F,2,N)
      DO 230 J=1,N
         YPR(J,M4)=F(J)
         DEP(J)=DEP(J)+DELT*(B(4)*F(J)+TE(J))
         DV(J)=DEP(J)
C
C     ********   SINGLE STEP ERROR ESTIMATE
C
         ER(J)=RATIO*ABS(ER(J)-DV(J))
  230 CONTINUE
      GO TO 120
C
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE DERIV (T,Y,DY,LOC,N)
      DIMENSION Y(5), DY(5)
      COMMON /DDS/ BETA1
      COMMON /CONST/ A1,A2,A3,A4,E,EPS,DELT,IERR
      X=((Y(4)-(A1*Y(2)**2))/A2)
      IF (X.LE.0.) GO TO 10
      X1=(X**(-0.25))
      X2=(-0.25)*(X**(-1.25))*((Y(5)/A2)-2.*(A1/A2)*Y(2)*Y(3))
      DY(1)=Y(2)
      DY(2)=Y(3)
      DY(3)=-(1./X1)*((Y(1)+X2)*Y(3)+BETA1*(X1-Y(2)*Y(2)))
      DY(4)=Y(5)
      DY(5)=-(1./(A3*X1))*((Y(1)+A3*X2)*Y(5)+((X1*Y(3)**2)-Y(1)*Y(2)*Y(3
     1))*A4)
      RETURN
   10 IERR=1
      RETURN
C
      END
```

```
      SUBROUTINE FJOYCE
C
C     SUBROUTINE FJOYCE USES FINITE DIFFERENCE METHODS
C     TO CALCULATE THE DOWNSTREAM PROFILES
C
      DIMENSION F1(5Ø), THETA1(5Ø), CS1(5Ø), F2(5Ø), DFLCDS(5Ø), DFLCS1(
     15Ø), T(5Ø), FF(5Ø), FLC(5Ø), FLC1(5Ø), THETA2(5Ø), CS2(5Ø), FLC2(5
     2Ø), YN(5Ø)
      DIMENSION CPI(5Ø), CPS(5Ø), VISI(5Ø), VISS(5Ø), TCONI(5Ø), TCONS(5
     1Ø), DIFFC(5Ø), CPM(5Ø), G12(5Ø), G21(5Ø), XS(5Ø), XI(5Ø), VISM(5Ø)
     2, TCONM(5Ø), RHOM(5Ø), CHF(5Ø), SCNC(5Ø), PRNO(5Ø)
      DIMENSION ETA(5Ø), DETA(5Ø), Y(5Ø), DELST(5Ø), THMOM(5Ø), X(6Ø)
      DIMENSION DSTAR(5Ø), DDSTR(5Ø)
      DIMENSION A(5Ø,3,3), B(5Ø,3,3), C(5Ø,3,3), AH(5Ø,3,3), H(5Ø,3,3),
     1HAH(5Ø,3,3), HAHINV(5Ø,3,3), ADJHAH(5Ø,3,3), AG(5Ø,3,1), G(5Ø,3,1)
     2, DAG(5Ø,3,1), D(5Ø,3,1), HW(5Ø,3,1), W(5Ø,3,1)
      DIMENSION TAW(5Ø), QDOTT(5Ø), STNDT(5Ø), SPECHT(5Ø), XTC(5Ø), QDOT
     1M(5Ø)
      DIMENSION TCNO(5Ø), DMDT(5Ø)
      DIMENSION RHOS(5Ø), XF(2)
      COMMON /PROP/ TEMP(4Ø),PRES(4),NTEMP,NPRES
      COMMON /NEW/ F(5Ø),THETA(5Ø),ALF,N,KK,MM,CS(5Ø),X,IPERF,NTEMP1
      COMMON /NEW2/ TTE(6Ø),WMT,WMS,HDS(6Ø),DDSTR
      COMMON /NEW3/ FLCW(6Ø),THETAE(6Ø),ROVIN(6Ø),BETA(6Ø),DS(6Ø),S(6Ø),
     1UE(6Ø),RHOE(6Ø)
      COMMON /NEW4/ PE(6Ø),THETAW(6Ø),TE(6Ø),VISE(6Ø),HTWEF
      COMMON /NEW1Ø/ KEY,VSC(16Ø),VIC(16Ø),TCSC(16Ø),TCIC(16Ø),CPIC(16Ø)
     1,CPSC(16Ø)
      DELT=1./FLOAT(N-1)
      NALF=Ø
      RUNIV=1545.
      FI=(WMT/26.Ø)**.461
      FS=(WMS/26.Ø)**.461
      M=1
      NN=N-1
      DO 1Ø I=1,N
         YN(I)=FLOAT(I-1)*DELT
         F1(I)=F(I)
         THETA1(I)=THETA(I)
         CS1(I)=CS(I)
   1Ø CONTINUE
      F2(1)=F1(1)
      F2(2)=F1(2)
      DO 2Ø I=1,N
         DFLCDS(I)=Ø.Ø
         DFLCS1(I)=DFLCDS(I)
         T(I)=TTE(M)*THETA(I)
   2Ø CONTINUE
      DO 3Ø I=1,NN
         FF(I)=F(I)/(ALF*(1.Ø-FLOAT(I-1)*DELT))
   3Ø CONTINUE
C
C     EVALUATION OF THE STREAM FUNCTION,FLC1
C
      FLC(1)=FLCW(M)
      FLC(2)=FLC(1)+(F(1)/ALF+F(2)/(ALF*(1.Ø-DELT)))*DELT/2.Ø
      DO 4Ø I=3,NN,2
         FLC(I)=FLC(I-2)+(FF(I-2)+4.Ø*FF(I-1)+FF(I))*DELT/3.Ø
         IF ((I+1).GT.NN) GO TO 4Ø
         FLC(I+1)=FLC(I-1)+(FF(I-1)+4.Ø*FF(I)+FF(I+1))*DELT/3.Ø
   4Ø CONTINUE
```

```
      DO 50 I=1,NN
         FLC1(I)=FLC(I)
   50 CONTINUE
      NSPACE=(54-N)/6
      IF (NSPACE.LT.0) NSPACE=0
      PRINT 690
      DO 60 I=1,NSPACE
         PRINT 700
   60 CONTINUE
      PRINT 710
      PRINT 720
      PRINT 730, (I,F(I),THETA(I),CS(I),I=1,N)
      I=1
      PLOG=ALOG10(PE(M))
      CPS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,CPSC)
      VISS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,VSC)
      TCONS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,TCSC)
      IF (KEY.EQ.1) GO TO 70
      CPI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,CPIC)
      TCONI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,TCIC)
      VISI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,VIC)
      GO TO 80
   70 CPI(I)=CPS(I)
      TCONI(I)=TCONS(I)
      VISI(I)=VISS(I)
   80 IF (IPERF.EQ.1) GO TO 90
      CALL MOLIER (H,PE(M),2,T(1),Z,ENT,RHOS(1),G)
      RHOM(1)=RHOS(1)
      GO TO 100
   90 RHOM(1)=PE(M)*WMS*WMI/(RUNIV*32.174*T(1)*(WMS+CS(1)*(WMI-WMS)))
  100 CONTINUE
      CPM(1)=CPI(1)+CS(1)*(CPS(1)-CPI(1))
      IF (CS(1)-1.0) 110,120,120
  110 XS(1)=(CS(1)*WMI)/(WMS+CS(1)*(WMI-WMS))
      XI(1)=1.0-XS(1)
      IF (XI(1).EQ.0.0) GO TO 120
      G12(1)=((1.0+SQRT(VISS(1)/VISI(1))*((WMI/WMS)**0.25))**2)/(2.8284*
     1SQRT(1.+WMS/WMI))
      G21(1)=((1.0+SQRT(VISI(1)/VISS(1))*((WMS/WMI)**0.25))**2)/(2.8284*
     1SQRT(1.+WMI/WMS))
      VISM(1)=VISS(1)/(1.0+G12(1)*XI(1)/XS(1))+VISI(1)/(1.0+G21(1)*XS(1)
     1/XI(1))
      TCONM(1)=TCONS(1)/(1.0+1.065*G12(1)*XI(1)/XS(1))+TCONI(1)/(1.0+1.0
     165*G21(1)*XS(1)/XI(1))
      GO TO 130
  120 VISM(1)=VISS(1)
      TCONM(1)=TCONS(1)
  130 CONTINUE
      TAU=RHOM(1)*VISM(1)*UE(M)*UE(M)*(ROS(M)**KK)*ALF*(F(2)-F(1))/(DELT
     1*((2.0*S(M))**0.5))
      CF=TAU/(0.5*RHOE(M)*UE(M)*UE(M))
      HEAT=-TCONM(1)*TTE(M)*UE(M)*RHOM(1)*(ROS(M)**KK)*ALF*(THETA(2)-THE
     1TA(1))/(DELT*((2.0*S(M))**0.5))
      QDOT=-HEAT
      I=N
      PLOG=ALOG10(PE(M))
      CPS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,CPSC)
      VISS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,VSC)
      TCONS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,TCSC)
      TCONM(I)=TCONS(I)
      VISM(I)=VISS(I)
      CPM(I)=CPS(I)
```

```
      PRNO(I)=VISM(I)*CPM(I)/TCONM(I)
      REC=SQRT(PRNO(N))
      TREC=REC*(TTE(M)-TE(M))+TE(M)
      STNO=QDOT/(RHOE(M)*UE(M)*CPM(N)*(TREC-THETAW(M)*TTE(M)))
      PRINT 740, TAU,CF
      PRINT 750, TREC,QDOT,STNO
      M=2
      KLOOP=0
  140 CONTINUE
  150 CONTINUE
      KLOOP=KLOOP+1
      IF (KLOOP.GT.150) GO TO 880
      PLOG=ALOG10(PE(M))
      DO 180 I=1,N
C
C     EVALUATION OF THE THERMO PROPERTIES OF THE COMPONENTS
C
         CPS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,CPSC)
         VISS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,VSC)
         TCONS(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,TCSC)
         IF (KEY.EQ.1) GO TO 160
         CPI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,CPIC)
         TCONI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,TCIC)
         VISI(I)=TWOVAR(NPRES,NTEMP,PLOG,T(I),PRES,TEMP,VIC)
         GO TO 170
  160    CPI(I)=CPS(I)
         TCONI(I)=TCONS(I)
         VISI(I)=VISS(I)
  170    DIFFC(I)=2.640E-8*(T(I)**1.5)*2116./(PE(M)*FI*FS)
  180 CONTINUE
      DO 240 I=1,NN
         IF (CS(I)-1.0) 200,190,190
  190    CONTINUE
         CPM(I)=CPS(I)
         VISM(I)=VISS(I)
         TCONM(I)=TCONS(I)
         GO TO 210
  200    CONTINUE
         CPM(I)=CPI(I)+CS(I)*(CPS(I)-CPI(I))
         G12(I)=((1.0+SQRT(VISS(I)/VISI(I))*((WMI/WMS)**0.25))**2)/(2.82
     1   84*SQRT(1.+WMS/WMI))
         G21(I)=((1.0+SQRT(VISI(I)/VISS(I))*((WMS/WMI)**0.25))**2)/(2.82
     1   84*SQRT(1.+WMI/WMS))
         XS(I)=(CS(I)*WMI)/(WMS+CS(I)*(WMI-WMS))
         XI(I)=1.0-XS(I)
         IF (XI(I).EQ.0.0) GO TO 190
         VISM(I)=VISS(I)/(1.0+G12(I)*XI(I)/XS(I))+VISI(I)/(1.0+G21(I)*(X
     1   S(I)/XI(I)))
         TCONM(I)=TCONS(I)/(1.0+1.065*G12(I)*(XI(I)/XS(I)))+TCONI(I)/(1.
     1   0+1.065*G21(I)*(XS(I)/XI(I)))
  210    CONTINUE
         IF (IPERF.EQ.1) GO TO 220
         CALL MOLIER (H,PE(M),2,T(I),Z,ENT,RHOS(I),G)
         RHOM(I)=RHOS(I)
         GO TO 230
  220    RHOM(I)=PE(M)*WMS*WMI/(RUNIV*32.174*T(I)*(WMS+CS(I)*(WMI-WMS)))
  230    CONTINUE
  240 CONTINUE
C
C     ******COMPUTATION OF FLOW PROPERTIES AT THE B. L . EDGE
C
      I=N
```

```
      TCONM(I)=TCONS(I)
      VISM(I)=VISS(I)
      CPM(I)=CPS(I)
      IF (IPERF.EQ.1) GO TO 250
      CALL MOLIEW (M,PE(M),2,T(I),Z,FNT,RHOS(I),G)
      RHOM(I)=RHOS(I)
      GO TO 260
  250 RHOM(I)=PE(M)*WMS/(RUNIV*32.174*T(I))
  260 CONTINUE
      ROMUE=RHOE(M)*VISE(M)
      DO 270 I=1,N
         CRF(I)=RHOM(I)*VISM(I)/ROMUE
         SCNO(I)=VISM(I)/(RHOM(I)*DIFFC(I))
         PRNO(I)=VISM(I)*CPM(I)/TCONM(I)
  270 CONTINUE
C
C     COMPUTATION OF THE MATRIX ELEMENTS
C
      DO 280 I=2,NN
         ALFT=ALF*(1.0-FLOAT(I-1)*DELT)
         A(I,1,1)=-(CRF(I)*(ALFT**2)/(2.0*(DELT**2))+(ALFT/(4.0*DELT))*(
     1   ALF*CRF(I)-FLC(I)-ALFT*(CRF(I+1)-CRF(I-1))/(2.0*DELT)-2.0*S(M)*
     2   DFLCDS(I)))
         A(I,1,2)=0.0
         A(I,1,3)=0.0
         A(I,2,1)=CRF(I)*(UE(M)**2)*(ALFT**2)*(F(I+1)-F(I-1))/(CPM(I)*TT
     1   E(M)*4.0*(DELT**2)*777.9)
         A(I,2,2)=-(CRF(I)*(ALFT**2)/(2.0*(DELT**2)*PRNO(I))+(ALFT/(4.0*
     1   DELT))*(ALF*CRF(I)/PRNO(I)-(ALFT/CPM(I))*((CRF(I+1)*CPM(I+1)/PR
     2   NO(I+1)-CRF I-1)*CPM(I-1)/PRNO(I-1))/(2.0*DELT))-FLC(I)-2.0*S(M
     3   )*DFLCDS(I)-CRF(I)*(CPS(I)-CPI(I))*ALFT*(CS(I+1)-CS(I-1))/(CPM(
     4   I)*SCNO(I)*2.0*DELT)))
         A(I,2,3)=(ALFT**2)*CRF(I)*(CPS(I)-CPI(I))*(THETA(I+1)-THETA(I-1
     1   ))/(CPM(I)*SCNO(I)*(8.0)*(DELT**2))
         A(I,3,1)=0.0
         A(I,3,2)=0.0
         A(I,3,3)=-((ALFT**2)*CRF(I)/(2.0*(DELT**2)*SCNO(I))+(ALFT/(4.0*
     1   DELT))*(ALF*CRF(I)/SCNO(I)-FLC(I)-2.0*S(M)*DFLCDS(I)-ALFT*(CRF(
     2   I+1)/SCNO(I+1)-CRF(I-1)/SCNO(I-1))/(2.0*DELT)))
         B(I,1,1)=(ALFT**2)*CRF(I)/(DELT**2)+BETA(M)*F(I)+2.0*S(M)*F(I)/
     1   DS(M)
         B(I,1,2)=0.0
         B(I,1,3)=0.0
         B(I,2,1)=BETA(M)*(UE(M)**2)*RHOE(M)/(CPM(I)*TTE(M)*RHOM(I)*777.
     1   9)
         B(I,2,2)=(ALFT**2)*CRF(I)/(PRNO(I)*(DELT**2))+2.0*S(M)*F(I)/DS(
     1   M)
         B(I,2,3)=0.0
         B(I,3,1)=0.0
         B(I,3,2)=0.0
         B(I,3,3)=(ALFT**2)*CRF(I)/(SCNO(I)*(DELT**2))+2.0*S(M)*F(I)/DS(
     1   M)
         C(I,1,1)=-A(I,1,1)-(ALFT**2)*CRF(I)/(DELT**2)
         C(I,1,2)=0.0
         C(I,1,3)=0.0
         C(I,2,1)=-A(I,2,1)
         C(I,2,2)=-A(I,2,2)-(ALFT**2)*CRF(I)/(PRNO(I)*(DELT**2))
         C(I,2,3)=-A(I,2,3)
         C(I,3,1)=0.0
         C(I,3,2)=0.0
         C(I,3,3)=-A(I,3,3)-(ALFT**2)*CRF(I)/(SCNO(I)*(DELT**2))
         D(I,1,1)=(ALFT**2)*CRF(I)*(F(I+1)-2.*F(I)+F(I-1))/(2.0*(DELT**2
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1))+(ALFT*(F(I+1)-F(I-1))/(4.0*DELT))*(FLC(I)+ALFT*(CRF(I+1)-CRF
     2(I-1))/(2.0*DELT)-ALF*CRF(I)+2.0*S(M)*DFLCDS(I))+BETA(M)*RHOE(M
     3)/RHOM(I)+2.0*S(M)*(F(I)**2)/DS(M)
      D(I,2,1)=(ALFT**2)*CRF(I)*(THETA(I+1)-2.0*THETA(I)+THETA(I-1))/
     1(2.0*(DELT**2)*PRNO(I))+(ALFT*(THETA(I+1)-THETA(I-1))/(4.0*DELT
     2))*(2.0*S(M)*DFLCDS(I)+FLC(I)-ALF*CRF(I)/PRNO(I)+ALFT*((CRF(I+1
     3)*CPM(I+1)/PRNO(I+1)-CRF(I-1)*CPM(I-1)/PRNO(I-1))/(2.0*DELT))/C
     4PM(I))+2.0*S(M)*F(I)*THETA(I)/DS(M)
      D(I,3,1)=(ALFT**2)*CRF(I)*(CS(I+1)-2.0*CS(I)+CS(I-1))/(2.0*(DEL
     1T**2)*SCNO(I))+(ALFT*(CS(I+1)-CS(I-1))/(4.0*DELT))*(FLC(I)+2.0*
     2S(M)*DFLCDS(I)-ALF*CRF(I)/SCNO(I)+ALFT*(CRF(I+1)/SCNO(I+1)-CRF(
     3I-1)/SCNO(I-1))/(2.0*DELT))+2.0*S(M)*F(I)*CS(I)/DS(M)
  280 CONTINUE
      ENS=(ROVIN(M)*SCNO(1)*SQRT(2.*S(M))/(ALF*RHOM(1)*VISM(1)*UE(M)*(RD
     1S(M)**KK)))*(DELT+.5*DELT*DELT*(1.-(SCNO(1)/CRF(1))*(CRF(2)/SCNO(2
     2)-CRF(1)/SCNO(1))/DELT+ROVIN(M)*SQRT(2.*S(M))*SCNO(1)/(ALF*RHOM(1)
     3*VISM(1)*UE(M)*(RDS(M)**KK))))
C
C     EVALUATION OF THE COEFFICIENTS
C
      H(1,1,1)=0.0
      H(1,1,2)=0.0
      H(1,1,3)=0.0
      H(1,2,1)=0.0
      H(1,2,2)=0.0
      H(1,2,3)=0.0
      H(1,3,1)=0.0
      H(1,3,2)=0.0
      H(1,3,3)=-1.0/(1.0+ENS)
      G(1,1,1)=0.0
      G(1,2,1)=THETAW(M)
      G(1,3,1)=0.0
      DO 360 I=2,NN
        DO 290 J=1,3
        DO 290 K=1,3
          AH(I,J,K)=0.
        DO 290 L=1,3
          AH(I,J,K)=AH(I,J,K)+A(I,J,L)*H(I-1,L,K)
  290   CONTINUE
        DO 300 J=1,3
        DO 300 K=1,3
          BAH(I,J,K)=B(I,J,K)-AH(I,J,K)
  300   CONTINUE
        ADJBAH(I,1,1)=BAH(I,2,2)*BAH(I,3,3)-BAH(I,3,2)*BAH(I,2,3)
        ADJBAH(I,2,1)=-BAH(I,2,1)*BAH(I,3,3)+BAH(I,2,3)*BAH(I,3,1)
        ADJBAH(I,3,1)=BAH(I,2,1)*BAH(I,3,2)-BAH(I,2,2)*BAH(I,3,1)
        ADJBAH(I,1,2)=-BAH(I,1,2)*BAH(I,3,3)+BAH(I,1,3)*BAH(I,3,2)
        ADJBAH(I,2,2)=BAH(I,1,1)*BAH(I,3,3)-BAH(I,1,3)*BAH(I,3,1)
        ADJBAH(I,3,2)=-BAH(I,1,1)*BAH(I,3,2)+BAH(I,1,2)*BAH(I,3,1)
        ADJBAH(I,1,3)=BAH(I,1,2)*BAH(I,2,3)-BAH(I,1,3)*BAH(I,2,2)
        ADJBAH(I,2,3)=-BAH(I,1,1)*BAH(I,2,3)+BAH(I,1,3)*BAH(I,2,1)
        ADJBAH(I,3,3)=BAH(I,1,1)*BAH(I,2,2)-BAH(I,2,1)*BAH(I,1,2)
        DETBAH=BAH(I,1,1)*BAH(I,2,2)*BAH(I,3,3)+BAH(I,1,2)*BAH(I,2,3)*B
     1AH(I,3,1)+BAH(I,1,3)*BAH(I,3,2)*BAH(I,2,1)-BAH(I,1,3)*BAH(I,3,1
     2)*BAH(I,2,2)-BAH(I,2,3)*BAH(I,3,2)*BAH(I,1,1)-BAH(I,1,2)*BAH(I,
     32,1)*BAH(I,3,3)
        DO 310 J=1,3
        DO 310 K=1,3
          BAHINV(I,J,K)=ADJBAH(I,J,K)/DETBAH
  310   CONTINUE
        DO 320 J=1,3
        DO 320 K=1,3
```

```
         H(I,J,K)=0.
      DO 320 L=1,3
         H(I,J,K)=H(I,J,K)+BAHINV(I,J,L)*C(I,L,K)
  320 CONTINUE
      DO 330 J=1,3
         AG(I,J,1)=0.
      DO 330 L=1,3
         AG(I,J,1)=AG(I,J,1)+A(I,J,L)*G(I-1,L,1)
  330 CONTINUE
      DO 340 J=1,3
         DAG(I,J,1)=D(I,J,1)-AG(I,J,1)
  340 CONTINUE
      DO 350 J=1,3
         G(I,J,1)=0.0
      DO 350 L=1,3
         G(I,J,1)=G(I,J,1)+BAHINV(I,J,L)*DAG(I,L,1)
  350 CONTINUE
  360 CONTINUE
C
C     SOLUTION FOR THE W VECTOR
C
      W(N,1,1)=1.0
      W(N,2,1)=THETAE(M)
      W(N,3,1)=1.0
      F(N)=W(N,1,1)
      THETA(N)=W(N,2,1)
      CS(N)=W(N,3,1)
      DO 390 II=1,NN
         I=N-II
         DO 370 J=1,3
            HW(I,J,1)=0.
         DO 370 L=1,3
            HW(I,J,1)=HW(I,J,1)+H(I,J,L)*W(I+1,L,1)
  370    CONTINUE
         DO 380 J=1,3
            W(I,J,1)=G(I,J,1)-HW(I,J,1)
  380    CONTINUE
         F(I)=W(I,1,1)
         THETA(I)=W(I,2,1)
         CS(I)=W(I,3,1)
  390 CONTINUE
      CONVG=(F(2)-F(1)-F2(2)+F2(1))/(F(2)-F(1))
      PRINT 760, CONVG
      IF (KLOUP.LT.2) GO TO 400
      IF (ABS(CONVG).LE.0.0005) GO TO 470
  400 DO 410 I=1,N
         F2(I)=F(I)
         THETA2(I)=THETA(I)
         CS2(I)=CS(I)
  410 CONTINUE
      DO 420 I=1,N
         F(I)=0.5*(F2(I)+F1(I))
         THETA(I)=.5*(THETA2(I)+THETA1(I))
         T(I)=TTE(M)*THETA(I)
         CS(I)=0.5*(CS2(I)+CS1(I))
  420 CONTINUE
  430 CONTINUE
C
C     EVALUATION OF THE STREAM FUNCTION,FCL2
C
      DO 440 I=1,NN
         FF(I)=F(I)/(ALF*(1.0-FLOAT(I-1)*DELT))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DELST(I)=DELST(I-2)+FACTOR*((RHOE(M)/RHOM(I-2)-F(I-2))/(1.0-YN(
     1 I-2))+4.0*(RHOE(M)/RHOM(I-1)-F(I-1))/(1.0-YN(I-1))+(RHOE(M)/RHO
     2 M(I)-F(I))/(1.0-YN(I)))*DELT/(3.0*ALF)
      IF ((I+1).GT.NN) GO TO 540
      DELST(I+1)=DELST(I-1)+FACTOR*((RHOE(M)/RHOM(I-1)-F(I-1))/(1.0-Y
     1 N(I-1))+4.0*(RHOE(M)/RHOM(I)-F(I))/(1.0-YN(I))+(RHOE(M)/RHOM(I+
     2 1)-F(I+1))/(1.0-YN(I+1)))*DELT/(3.0*ALF)
  540 CONTINUE
      DSTAR(M)=DDSTR(M)+DELST(NN)
      THMOM(1)=0.0
      THMOM(2)=FACTOR*((F(1)-F(1)*F(1))+(F(2)-F(2)*F(2))/(1.0-DELT))*DEL
     1T/(2.0*ALF)
      DO 550 I=3,NN,2
      THMOM(I)=THMOM(I-2)+FACTOR*((F(I-2)-F(I-2)*F(I-2))/(1.0-YN(I-2)
     1 )+4.0*(F(I-1)-F(I-1)*F(I-1))/(1.0-YN(I-1))+(F(I)-F(I)*F(I))/(1.
     2 0-YN(I)))*DELT/(3.0*ALF)
      IF ((I+1).GT.NN) GO TO 550
      THMOM(I+1)=THMOM(I-1)+FACTOR*((F(I-1)-F(I-1)*F(I-1))/(1.0-YN(I-
     1 1))+4.0*(F(I)-F(I)*F(I))/(1.0-YN(I))+(F(I+1)-F(I+1)*F(I+1))/(1.
     2 0-YN(I+1)))*DELT/(3.0*ALF)
  550 CONTINUE
      NSPACE=(51-N)/6
      IF (NSPACE.LT.0) NSPACE=0
      PRINT 690
      DO 560 I=1,NSPACE
      PRINT 700
  560 CONTINUE
      PRINT 770, M,X(M)
      PRINT 780
      PRINT 790, (I,YN(I),Y(I),ETA(I),F(I),THETA(I),CS(I),I=1,N)
      CPS(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,CPSC)
      VISS(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,VSC)
      TCONS(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,TCSC)
      IF (KEY.EQ.1) GO TO 570
      CPI(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,CPIC)
      TCONI(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,TCIC)
      VISI(1)=TWOVAR(NPRES,NTEMP,PLOG,T(1),PRES,TEMP,VIC)
      GO TO 580
  570 CPI(1)=CPS(1)
      TCONI(1)=TCONS(1)
      VISI(1)=VISS(1)
  580 CONTINUE
      CPM(1)=CPI(1)+CS(1)*(CPS(1)-CPI(1))
      IF (CS(1)-1.0) 590,600,600
  590 XS(1)=(CS(1)*WMI)/(WMS+CS(1)*(WMI-WMS))
      XI(1)=1.0-XS(1)
      IF (XI(1).EQ.0.0) GO TO 600
      G12(1)=((1.0+SQRT(VISS(1)/VISI(1))*((WMI/WMS)**0.25))**2)/(2.8284*
     1SQRT(1.+WMS/WMI))
      G21(1)=((1.0+SQRT(VISI(1)/VISS(1))*((WMS/WMI)**0.25))**2)/(2.8284*
     1SQRT(1.+WMI/WMS))
      VISM(1)=VISS(1)/(1.0+G12(1)*XI(1)/XS(1))+VISI(1)/(1.0+G21(1)*XS(1)
     1/XI(1))
      TCONM(1)=TCONS(1)/(1.0+1.065*G12(1)*XI(1)/XS(1))+TCONI(1)/(1.0+1.0
     165*G21(1)*XS(1)/XI(1))
      GO TO 610
  600 VISM(1)=VISS(1)
      TCONM(1)=TCONS(1)
  610 CONTINUE
      TAU=RHOM(1)*VISM(1)*UE(M)*UE(M)*(ROS(M)**KK)*ALF*(F(2)-F(1))/(DELT
     1*SQRT(2.*S(M)))
      CF=TAU/(0.5*RHOE(M)*UE(M)*UE(M))
```

```
      HEAT=-TCONM(1)*TTE(M)*UE(M)*RHOM(1)*(HDS(M)**KK)*ALF*(THETA(2)-THE
     1TA(1))/(DELT*SQRT(2.*S(M)))
      QDOT=-HEAT
      CPE=TWOVAR(NPRES,NTEMP,PLO.,TE(M),PRES,TEMP,CPSC)
      TCONE=TWOVAR(NPRES,NTEMP,P  G TE(M),PRES,TEMP,TCSC)
      PRNO(N)=VISE(M)*CPE/TCONE
      REC=SQRT(PRNO(N))
      TREC=REC*(TTE(M)-TE(M))+TE(M)
      STNO=QDOT/(RHOE(M)*UE(M)*CPM(N)*(TREC-THETAW(M)*TTE(M)))
      IF (ABS(THETAW-1.).LT.1.E-8) GO TO 620
      HCOEF=QDOT/(TTE(M)*(1.-THETAW(M)))
      GO TO 630
  620 HCOEF=999.
  630 HRATIO=HCOEF/HTREF
      PRINT 800, TAU,CF,HCOEF
      PRINT 810, TREC,QDOT,STNO,HRATIO
      PRINT 820, DELST(NN),THMOM(NN),DSTAR(M)
      DO 640 I=1,NN
         FF(I)=F(I)/(ALF*(1.0-FLOAT(I-1)*DELT))
  640 CONTINUE
      IF ((M+2).GT.MM) GO TO 680
      FLC(1)=0.5*(FLCW(M)+FLCW(M+1))
      FLC(2)=FLC(1)+(F(1)/ALF+F(2)/(ALF*(1.0-DELT)))*DELT/2.0
      DO 650 I=3,NN,2
         FLC(I)=FLC(I-2)+(FF(I-2)+4.0*FF(I-1)+FF(I))*DELT/3.0
         IF ((I+1).GT.NN) GO TO 650
         FLC(I+1)=FLC(I-1)+(FF(I-1)+4.0*FF(I)+FF(I+1))*DELT/3.0
  650 CONTINUE
      DO 660 I=1,NN
         DFLCDS(I)=(FLC(I)-FLC1(I))/DS(M)
  660 CONTINUE
      FLC(N)=0.0
      DO 670 I=1,N
         F1(I)=F(I)
         THETA1(I)=THETA(I)
         CS1(I)=CS(I)
         FLC1(I)=FLC(I)
  670 CONTINUE
      F2(1)=F1(1)
      F2(2)=F1(2)
      M=M+1
      GO TO 150
  680 CONTINUE
      RETURN
C
  690 FORMAT (1H1)
  700 FORMAT (/)
  710 FORMAT (52X,32HSTATION M=1, THE INITIAL PROFILE)
  720 FORMAT (43X,1HI,8X,4HF(I),8X,8HTHETA(I),8X,5HCS(I))
  730 FORMAT  .9X,I15,3E14.5)
  740 FORMAT (//44X,4HTAU=,E12.5,10X,3HCF=,E12.5)
  750 FORMAT (//30X,5HTREC=,E12.5,9X,5HQDOT=,E12.5,9X,5HSTNO=,E12.5//)
  760 FORMAT (10X,6HCONVG=,F12.5)
  770 FORMAT (50X,2HM=,I5,5X,5HX(M)=,F10.5,/)
  780 FORMAT (35X,1HI,2X,5HYN(I),2X,4HY(I),3X,6HFTA(I),5X,4HF(I),6X,8HTH
     1ETA(I),5X,5HCS(I))
  790 FORMAT (31X,I5,2X,F5.3,2F8.5,3E12.5)
  800 FORMAT (//30X,4HTAU=,E12.5,6X,3HCF=,E12.5,7X,6HHCOEF=,E12.5)
  810 FORMAT (//30X,5HTREC=,E12.5,5X,5HQDOT=,E12.5,5X,5HSTNO=,E12.5,5X,7
     1HHRATIO=,E12.5//)
  820 FORMAT (30X,6HDELST=,F12.5,8X,6HTHMOM=,E12.5,8X,6HDSTAR=,E12.5//)
  830 FORMAT (10X,I5,2(5X,F15.8))
```

```
C
      END
```

```
      FUNCTION TWOVAR(ITVMX,JTVMX,VALX,VALY,X,Y,FN)
      DIMENSION Y(JTVMX), X(ITVMX), FN(ITVMX,JTVMX)
      DO 10 J=1,JTVMX
         IF (Y(J)-VALY) 10,20,60
   10 CONTINUE
   20 DO 30 I=1,ITVMX
         IF (X(I)-VALX) 30,40,50
   30 CONTINUE
   40 FNMN=FN(I,J)
      TWOVAR=FNMN
      RETURN
   50 IP1=I
      I=I-1
      TWOVAR=FN(I,J)+(FN(IP1,J)-FN(I,J))*(VALX-X(I))/(X(IP1)-X(I))
      RETURN
   60 J=J-1
      DO 70 I=1,ITVMX
         IF (X(I)-VALX) 70,80,90
   70 CONTINUE
   80 FNMN=FN(I,J)
      GO TO 100
   90 IP1=I
      I=I-1
      FNMN=FN(I,J)+(FN(IP1,J)-FN(I,J))*(VALX-X(I))/(X(IP1)-X(I))
  100 JU=J+1
      DO 110 I=1,ITVMX
         IF (X(I)-VALX) 110,120,130
  110 CONTINUE
  120 FNMX=FN(I,JU)
      GO TO 140
  130 IP1=I
      I=I-1
      FNMX=FN(I,JU)+(FN(IP1,JU)-FN(I,JU))*(VALX-X(I))/(X(IP1)-X(I))
  140 CONTINUE
      TWOVAR=FNMN+(FNMX-FNMN)*(VALY-Y(J))/(Y(JU)-Y(J))
      RETURN
C
      END
```

```
      SUBROUTINE MOLIER (H,P,NOPT,T,Z,S,RHO,GAMMA)
C
C     NOPT=0  LOOK UP PROPS BASED ON P AND H
C     NOPT=1  LOOK UP PROPS BASED ON P AND S
C     NOPT=2  LOOK UP PROPS BASED ON P AND T
C     NOPT=3 LOOK UP PROPS BASED ON H AND S
C
      DIMENSION FLP(33,20), HZ(33,20), TT(33,20), ZT(33,20), GAME(33,20)
     1, ENTRO(33,20), FLPO(660), HZO(660), TTO(660), ZTO(660), GAMEO(660
     2), ENTROO(660), FLPZ(20), HTHL(33), ENTROV(20,33), FLPV(20,33)
      EQUIVALENCE (FLPO,FLP), (HZO,HZ), (TTO,TT), (ZTO,ZT), (GAMEO,GAME)
     1, (ENTROO,ENTRO)
      DATA PO,CPOR,HO,SOR,G,R,ALF,RTO,CP/2116.,.3,48158,117.346,23.6,32.2
     1,53.35,2.302585,33.705,.23866/
      IDEAL=0
      Z=1.0
      GAMMA=1.4
      IF (IZ.EQ.33) GO TO 30
      DO 10 K=1,20
         LL=33*(K-1)
      DO 10 L=1,33
         FLPO(LL+L)=FLPZ(K)
         IF (K.EQ.1) GO TO 10
         HZO(LL+L)=HZO(L)
   10 CONTINUE
      IZ=33
      JZ=20
      DO 20 I=1,IZ
         HTHL(I)=HZ(I,1)
      DO 20 J=1,JZ
         ENTROV(J,I)=ENTRO(I,JZ-J+1)
         FLPV(J,I)=FLP(I,JZ-J+1)
   20 CONTINUE
   30 IF (NOPT.EQ.3) GO TO 240
      PL=ALOG10(P/2116.)+10.
      IF (NOPT-1) 40,140,210
   40 IF (H.LT.100.) GO TO 110
   50 CONTINUE
      CALL DINT2 (H,HZ,Z,ZT,PL,FLPZ,IZ,JZ,T,TT,SS,ENTRO,0,GAMMA,GAME)
      IF (NOPT.EQ.0) GO TO 80
      IF (ABS((S-SS)*10.**(4-IPLACE)).LT.1.) GO TO 90
      IF (S-SS) 60,90,70
   60 PLL=PL
      PL=(PL+PLU)/2.
      GO TO 50
   70 PLU=PL
      PL=(PL+PLL)/2.
      GO TO 50
   80 S=SS
   90 CONTINUE
      IF (Z-.2222E+30) 100,130,130
  100 RHO=P/(32.2*53.35*Z*T)
      GO TO 330
  110 IF (H.LT.0.) GO TO 130
      T=H/CP
      RHO=P/(G*R*T)
  120 S=(CPOR*ALOG10(H/HO)-ALOG10(P/PO))*ALF+SOR
      GO TO 330
  130 IFP=1
      IDEAL=1
      GO TO 330
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
140 IF (PL.LT.FLPZ(1)) GO TO 200
    DO 150 J=2,JZ
       IF (PL-FLPZ(J)) 170,160,150
150 CONTINUE
160 IF (S-ENTRO(2,J)) 190,180,180
170 IF (S-ENTRO(2,J-1)) 190,180,180
180 CALL DINT2 (S,ENTRO,H,HZ,PL,FLPZ,IZ,JZ,T,TT,Z,ZT,1,GAMMA,GAME)
    IF (H-.2222E+30) 100,200,200
190 H=RTO*CPOR*10.**((S-SOR+ALE*ALOG10(P/P0))/(CPOR*ALE))
    T=H/CP
    RHO=P/(G*R*T)
    GO TO 330
200 IFP=1
    IDEAL=1
    GO TO 330
210 IF (T.LT.419.) GO TO 220
    CALL DINT2 (T,TT,H,HZ,PL,FLPZ,IZ,JZ,Z,ZT,S,ENTRO,1,GAMMA,GAME)
    IF (H-.2222E+30) 100,230,230
220 IF (T.LT.0.) GO TO 230
    H=CP*T
    RHO=P/(G*R*T)
    GO TO 120
230 IFP=1
    IDEAL=1
    GO TO 330
240 IF (H.LT.100.) GO TO 300
    CALL DINT1 (S,ENTROV,PL,FLPV,H,HTBL,JZ,IZ)
    IF (PL-.2222E+30) 250,310,310
250 IF (PL.GT.12.69897) GO TO 320
    IPLACE=ALOG10(S)
    I=0
260 I=I+1
    IF (FLPZ(I)-PL) 260,270,280
270 PLU=FLPZ(I+1)
    PLL=FLPZ(I-1)
    GO TO 290
280 PLU=FLPZ(I)
    PLL=FLPZ(I-1)
290 P=10.**(PL-10.)*2116.
    GO TO 40
300 IF (H.LT.0.) GO TO 310
    ALP=(SOR-S)/ALE+CPOR*ALOG10(H/H0)
    P=P0*10.**ALP
    T=H/CP
    RHO=P/(G*R*T)
    GO TO 330
310 IFP=1
    IDFAL=1
    GO TO 330
320 IDEAL=2
330 RETURN
     DATA (FLPZ(J),J=1,20)/6.30103,6.69897,7.0,7.30103,7.69897,8.0,8.30
    1103,8.69897,9.0,9.30103,9.69897,10.0,10.30103,10.69897,11.0,11.301
    203,11.69897,12.0,12.30103,12.69897/
     DATA (TTO(L),L=1,204)/0.,419.,834.,1616.,2339.,3032.,3478.,3848.,4
    1025.,4175.,4320.,4466.,4600.,4880.,5148.,5730.,6238.,6565.,6836.,7
    2005.,7155.,7270.,7380.,7693.,7911.,8100.,8262.,8410.,8568.,8723.,8
    3900.,9140.,9450.,0.,419.,834.,1616.,2339.,3032.,3517.,3907.,4130.,
    44300.,4460.,4600.,4750.,4940.,5260.,5760.,6300.,6705.,6984.,7195.,
    57350.,7490.,7596.,7927.,8172.,8374.,8550.,8712.,8878.,9054.,9250.,
    69490.,9780.,0.,419.,834.,1616.,2339.,3032.,3546.,3964.,4224.,4390.
    7,4563.,4708.,4865.,5060.,5346.,5775.,6350.,6795.,7092.,7305.,7495.
```

```
8.7640.,7772.,8125.,8380.,8595.,8784.,8950.,9135.,9324.,9520.,9750.
9,10030.,0.,419.,834.,1616.,2339.,3032.,3571.,4035.,4295.,4500.,466
*9.,4830.,4990.,5180.,5450.,5890.,6450.,6870.,7218.,7450.,7640.,779
*0.,7920.,8323.,8604.,8827.,9018.,9216.,9405.,9594.,9780.,10030.,10
*350.,0.,419.,834.,1616.,2339.,3032.,3576.,4100.,4410.,4630.,4815.,
*4985.,5150.,5340.,5594.,5990.,6498.,6995.,7353.,7620.,7840.,8015.,
*8172.,8597.,8901.,9150.,9378.,9576.,9765.,9963.,10180.,10420.,1073
*. ,0.,419.,834.,1616.,2339.,3032.,3596.,4150.,4485.,4735.,4930.,51
*01.,5270.,5480.,5724.,6050.,6580.,7050.,7470.,7760.,8000.,8190.,83
*14.,8811.,9144.,9410.,9648.,9846.,10060.,10280.,10500.,10740.,1100
* ,0.,419.,834.,1616.,2339.,3032./
  ATA (TTO(L),L=205,400)/3618.,4208.,4555.,4829.,5040.,5240.,5410.,
 3610.,5872.,6195.,6650.,7165.,7596.,7908.,8150.,8355.,8532.,9030.,
.9414.,9657.,9927.,10160.,10380.,10590.,10820.,11100.,11400.,0.,419
5.,834.,1616.,2339.,3032.,3630.,4259.,4655.,4960.,5202.,5408.,5640.
4.5840.,6060.,6350.,6790.,7250.,7722.,8095.,8361.,8600.,8725.,9378.
5.9756.,10000.,10350.,10580.,10800.,11040.,11300.,11580.,11880.,0.,
6419.,834.,1616.,2339.,3032.,3644.,4300.,4750.,5060.,5323.,5550.,57
758.,5990.,6228.,6505.,6900.,7310.,7830.,8215.,8520.,8700.,9000.,96
850.,10040.,10390.,10680.,10920.,11180.,11420.,11680.,11970.,12290.
9 0.,419.,834.,1616.,2339.,3132.,3654.,4314.,4800.,5150.,5445.,5683
* ,5910.,6150.,6394.,6650.,7145.,7485.,7956.,8350.,8680.,8955.,9198
*.,9864.,10350.,10690.,11000.,11290.,11550.,11810.,12100.,12380.,12
*.00.,0.,419.,834.,1616.,2339.,3032.,3654.,4328.,4880.,5295.,5598.,
*1580.,6110.,6355.,6620.,6900.,7250.,7660.,8118.,8550.,8910.,9210.,
*5468.,10230.,10740.,11150.,11480.,11790.,12080.,12370.,12650.,1298
*0.,13300.,0.,419.,834.,1616.,2339.,3032.,3654.,4416.,4945.,5395.,5
*724.,6015.,6285.,6540.,6808.,7100.,7450.,7811.,8270.,8695.,9085.,9
*415.,9702.,10510.,11070.,11530.,11870.,12200.,12510.,12820.,13110.
*. 3450.,13800.,0.,419.,834.,1616./
 DATA (TTU(L),L=401,590)/2339.,3032.,3654.,4455.,4980.,5475.,5839.,
10150.,6450.,6735.,7002.,7300.,7640.,8000.,8433.,8848.,9245.,9603.,
29900.,10800.,11400.,11880.,12240.,12630.,12970.,13280.,13610.,1396
30.,14310.,0.,419.,834.,1616.,2339.,3032.,3654.,4469.,5010.,5548.,5
497c.,6348.,6655.,6950.,7281.,7575.,7910.,8275.,8658.,9100.,9496.,9
5865.,10188.,11200.,11880.,12020.,12820.,13271.,13600.,13960.,14310
6.,14690.,15080.,0.,419.,834.,1610.,2339.,3032.,3654.,4483.,5025.,5
7584.,6080.,6474.,6830.,7150.,7488.,7800.,8140.,8500.,8892.,9300.,9
8700.,10470.,10000.,11500.,12200.,12840.,13300.,13770.,14140.,14500
9.,14880.,15270.,15600.,0.,419.,834.,1616.,2339.,3032.,3654.,4497.,
*5038.,5675.,6169.,6580.,6975.,7340.,7680.,8008.,8350.,8740.,9100.,
*9525.,9930.,10310.,10683.,11800.,12050.,13280.,13810.,14270.,1470.
*.,15 00.,15490.,15890.,16290.,0.,419.,834.,1616.,2339.,3032.,3654.
*,4505.,5450.  725.,6266.,6753.,7 08.,7566.,7956.,8300.,8670.,9050.
*,9050.,9850.,10260.,10660.,11052.,12290.,13190.,13910.,14500.,1501
*0.,15480.,15930.,16350.,16800.,17220.,0.,419.,834.,1616.,2339.,303
*2.,3654.,4515.,5075.,5750.,6330.,6840.,7220.,7755.,8154.,8540.,894
*5.,9310.,9720.,10110.,10540.,10940.,11340.,12630.,13630.,14400.,15
*050.,15610.,16110./
 DATA (TTU(L),L=591,600)/16600.,17070.,17520.,17980.,0.,419.,834.,1
1616.,2339.,3032.,3654.,4425.,5110.,5774.,6386.,6935.,7440.,7905.,8
2352.,8755.,9155.,9570.,9990.,10400.,10846.,11230.,11630.,13010.,14
3090.,14940.,15640.,16240.,16830.,17320.,17800.,18300.,18820.,0.,41
49.,834.,1616.,2339.,3032.,3654.,4425.,5110.,5790.,6440.,7030.,7580
5.,8090.,8590.,9 0.,9490.,9905.,10370.,10800.,11210.,11640.,12060.
6.13540.,14710.,15680.,16460.,17140.,17770.,18360.,18910.,19450.,19
7980./
 DATA (ZT(   ),L=1,249)/6*1.0,1.007,1.024,1.049,1.076,1.103,1.130,1.
1156,1.180,1.201,1.210,1.216,1.226,1.238,1.251,1.265,1.280,1.296,1.
2358,1.422,1.486,1.551,1.617,1.681,1.746,1.810,1.874,1.934,6*1.0,1.
3004, .020,1.044,1.071,1.098,1.124,1.152,1.176,1.197,1.208,1.214,1.
4224,1.235,1.248,1.260,1.275,1.290,1.352,1.416,1.484,1.545,1.609,1.
```

```
5674,1.738,1.802,1.865,1.925,6*1.0,1.003,1.018,1.040,1.067,1.094,1.
6119,1.147,1.171,1.193,1.206,1.213,1.221,1.232,1.244,1.258,1.272,1.
7287,1.348,1.411,1.475,1.539,1.604,1.667,1.732,1.795,1.858,1.918,6*
81.0,1.001,1.016,1.037,1.060,1.088,1.115,1.140,1.165,1.188,1.204,1.
9212,1.219,1.229,1.241,1.255,1.268,1.283,1.343,1.406,1.470,1.534,1.
*597,1.661,1.725,1.788,1.851,1.909,7*1.0,1.013,1.032,1.056,1.081,1.
*108,1.132,1.156,1.181,1.197,1.208,1.216,1.226,1.237,1.249,1.264,1.
*277,1.337,1.398,1.461,1.526,1.589,1.652,1.715,1.778,1.840,1.899,7*
*1.0,1.012,1.028,1.052,1.076,1.100,1.128,1.152,1.175,1.193,1.205,1.
*214,1.223,1.234,1.246,1.259,1.273,1.332,1.393,1.455,1.518,1.582,1.
*644,1.707,'.770,1.831,1.891,7*1.0,1.011,1.024,1.047,1.070,1.096,1.
*120,1.144,1.169,1.188,1.201,1.211,1.220,1.231,1.242,1.256,1.269,1.
*326,1.387,1.449,1.511,1.574,1.636,1.699,1.761,1.822,1.880,7*1.0,1.
*010,1.022,1.042,1.063,1.088,1.112,1.136,1.160,1.180,1.196,1.206/
 DATA (ZTO(L),L=250,502)/1.215,1.226,1.237,1.249,1.263,1.318,1.378,
11.439,1.500,1.563,1.625,1.687,1.748,1.808,1.867,7*1.0,1.009,1.017,
21.037,1.058,1.080,1.104,1.128,1.152,1.172,1.190,1.201,1.211,1.220,
31.232,1.244,1.258,1.312,1.372,1.431,1.492,1.554,1.616,1.677,1.738,
41.798,1.855,7*1.0,1.008,1.015,1.032,1.052,1.076,1.098,1.121,1.144,
51.164,1.184,1.197,1.207,1.216,1.228,1.240,1.252,1.306,1.364,1.424,
61.484,1.545,1.606,1.667,1.727,1.787,1.844,7*1.0,1.007,1.012,1.026,
71.046,1.066,1.088,1.110,1.133,1.154,1.173,1.188,1.200,1.210,1.220,
81.232,1.245,1.297,1.353,1.412,1.472,1.531,1.592,1.653,1.711,1.770,
91.826,7*1.0,1.005,1.009,1.022,1.040,1.058,1.080,1.101,1.124,1.144,
*1.164,1.180,1.193,1.204,1.215,1.227,1.238,1.289,1.345,1.403,1.462,
*1.522,1.581,1.640,1.699,1.757,1.812,7*1.0,1.004,1.007,1.018,1.034,
*1.052,1.072,1.092,1.115,1.135,1.154,1.172,1.186,1.197,1.208,1.219,
*1.231,1.281,1.337,1.394,1.452,1.510,1.569,1.627,1.685,1.742,1.796,
*7*1.0,1.003,1.005,1.013,1.027,1.044,1.063,1.081,1.102,1.122,1.140,
*1.159,1.174,1.188,1.199,1.210,1.221,1.270,1.324,1.379,1.437,1.494,
*1.552,1.609,1.666,1.721,1.777,7*1.0,1.002,1.004,1.012,1.023,1.037,
*1.055,1.072,1.092,1.112,1.129,1.148,1.164,1.177,1.189,1.200,1.213,
*1.261,1.314,1.368,1.424,1.480,1.537,1.594,1.650,1.705,1.759,7*1.0/
 DATA (ZTO(L),L=503,600)/1.001,1.003,1.008,1.019,1.030,1.046,1.064,
11.082,1.098,1.117,1.136,1.152,1.165,1.179,1.191,1.203,1.251,1.302,
21.355,1.410,1.466,1.522,1.577,1.632,1.687,1.739,8*1.0,1.002,1.005,
31.014,1.024,1.037,1.052,1.069,1.085,1.103,1.119,1.136,1.150,1.164,
41.176,1.189,1.235,1.285,1.338,1.390,1.445,1.500,1.553,1.607,1.660,
51.712,8*1.0,1.001,1.003,1.010,1.019,1.030,1.044,1.059,1.075,1.091,
61.107,1.122,1.136,1.150,1.163,1.175,1.223,1.272,1.322,1.375,1.428,
71.480,1.534,1.586,1.639,1.692,9*1.0,1.001,1.007,1.015,1.024,1.037,
81.050,1.064,1.079,1.093,1.108,1.122,1.136,1.148,1.161,1.208,1.256,
91.306,1.357,1.410,1.460,1.512,1.564,1.615,1.665,10*1.,1.003,1.010,
*1.019,1.028,1.039,1.050,1.063,1.076,1.090,1.104,1.116,1.128,1.141,
*1.187,1.234,1.282,1.332,1.382,1.431,1.480,1.531,1.581,1.630/
 DATA (GAMEO(L),L=1,212)/2*1.4,1.390,1.344,1.317,1.295,1.206,1.143,
11.114,1.108,1.108,1.114,1.131,1.169,1.226,1.264,1.238,1.180,1.143,
21.121,1.110,1.103,1.099,1.094,1.092,1.093,1.094,1.096,1.099,1.104,
31.111,1.122,1.138,2*1.4,1.390,1.344,1.317,1.295,1.214,1.149,1.119,
41.113,1.113,1.118,1.131,1.162,1.212,1.262,1.248,1.190,1.152,1.128,
51.116,1.109,1.105,1.098,1.096,1.097,1.098,1.100,1.102,1.107,1.114,
61.124,1.138,2*1.4,1.390,1.344,1.317,1.295,1.220,1.154,1.124,1.117,
71.116,1.121,1.132,1.159,1.203,1.258,1.255,1.198,1.158,1.133,1.121,
81.114,1.109,1.101,1.099,1.100,1.101,1.103,1.105,1.110,1.116,1.126,
91.139,2*1.4,1.390,1.344,1.317,1.295,1.227,1.160,1.130,1.121,1.120,
*1.124,1.134,1.156,1.195,1.250,1.261,1.206,1.165,1.140,1.126,1.118,
*1.114,1.105,1.102,1.103,1.104,1.106,1.108,1.112,1.118,1.128,1.140,
*2*1.4,1.390,1.344,1.317,1.295,1.236,1.169,1.138,1.128,1.125,1.129,
*1.137,1.154,1.186,1.237,1.261,1.218,1.174,1.148,1.134,1.125,1.120,
*1.110,1.107,1.107,1.108,1.110,1.112,1.116,1.122,1.130,1.142,2*1.4,
*1.390,1.344,1.317,1.295,1.244,1.176,1.144,1.133,1.130,1.133,1.140,
*1.154,1.181,1.226,1.258,1.227,1.182,1.156,1.140,1.131,1.120,1.114,
```

```
*1.111,1.111,1.112,1.113,1.116,1.124,1.125,1.133,1.145,2*1.4,1.390,
*1.344,1.317,1.295,1.250,1.183,1.151,1.136,1.134,1.137,1.143,1.155/
 DATA (GAMEU(L),L=213,276)/1.179,1.218,1.252,1.236,1.191,1.163,1.14
17,1.138,1.131,1.118,1.114,1.114,1.115,1.117,1.124,1.123,1.128,1.13
26,1.147,2*1.4,1.390,1.344,1.317,1.295,1.258,1.193,1.161,1.146,1.14
30,1.142,1.148,1.157,1.177,1.209,1.244,1.244,1.207,1.177,1.159,1.14
48,1.139,1.124,1.120,1.119,1.120,1.122,1.124,1.128,1.133,1.141,1.15
50,2*1.4,1.390,1.344,1.317,1.295,1.263,1.201,1.169,1.152,1.146,1.14
67/
 DATA (GAMEO(L),L=277,424)/1.152,1.164,1.177,1.204,1.237,1.245,1.21
19,1.189,1.169,1.150,1.145,1.128,1.124,1.123,1.124,1.126,1.128,1.13
22,1.137,1.143,1.154,2*1.4,1.390,1.344,1.317,1.295,1.267,1.210,1.17
37,1.159,1.152,1.152,1.156,1.165,1.178,1.200,1.230,1.242,1.226,1.20
41,1.179,1.164,1.152,1.134,1.128,1.127,1.128,1.133,1.132,1.136,1.14
50,1.147,1.156,2*1.4,1.390,1.344,1.317,1.295,1.271,1.223,1.168,1.16
66,1.164,1.159,1.161,1.169,1.181,1.197,1.221,1.236,1.231,1.214,1.19
72,1.174,1.162,1.141,1.135,1.133,1.133,1.136,1.138,1.142,1.146,1.15
82,1.161,2*1.4,1.390,1.344,1.317,1.295,1.273,1.232,1.196,1.175,1.16
97,1.164,1.166,1.173,1.182,1.195,1.217,1.231,1.232,1.222,1.202,1.18
*3,1.170,1.147,1.140,1.138,1.136,1.140,1.142,1.146,1.154,1.156,1.16
*4,2*1.4,1.390,1.344,1.317,1.295,1.274,1.239,1.206,1.183,1.172,1.17
*0,1.171,1.178,1.186,1.197,1.214,1.228,1.232,1.226,1.214,1.192,1.17
*9,1.153,1.145,1.143,1.143,1.145/
 DATA (GAMEO(L),L=425,636)/1.147,1.15,1.154,1.160,1.166,2*1.4,1.390
1,1.344,1.317,1.295,1.275,1.244,1.218,1.195,1.182,1.170,1.178,1.183
2,1.190,1.200,1.213,1.225,1.231,1.228,1.218,1.204,1.191,1.162,1.153
3,1.150,1.150,1.150,1.153,1.156,1.160,1.160,1.173,2*1.4,1.390,1.344
4,1.317,1.295,1.275,1.247,1.226,1.204,1.189,1.184,1.184,1.188,1.194
5,1.242,1.213,1.224,1.230,1.230,1.224,1.212,1.200,1.169,1.159,1.155
6,1.155,1.156,1.158,1.161,1.165,1.171,1.177,2*1.4,1.390,1.344,1.317
7,1.295,1.275,1.250,1.232,1.211,1.196,1.190,1.190,1.192,1.198,1.205
8,1.214,1.224,1.230,1.231,1.227,1.219,1.208,1.176,1.165,1.161,1.160
9,1.161,1.163,1.166,1.169,1.175,1.182,2*1.4,1.390,1.344,1.317,1.295
*,1.276,1.253,1.237,1.219,1.205,1.199,1.198,1.199,1.204,1.210,1.216
*,1.224,1.230,1.232,1.231,1.226,1.216,1.186,1.174,1.169,1.167,1.168
*,1.170,1.173,1.177,1.181,1.187,2*1.4,1.390,1.344,1.317,1.295,1.276
*,1.254,1.240,1.223,1.212,1.205,1.203,1.200,1.208,1.214,1.220,1.225
*,1.230,1.233,1.232,1.228,1.221,1.194,1.181,1.176,1.173,1.174,1.176
*,1.179,1.182,1.186,1.192,2*1.4,1.390,1.344,1.317,1.295,1.276,1.255
*,1.241,1.226,1.217,1.211,1.208,1.209,1.212,1.218,1.223,1.226,1.229
*,1.233,1.232,1.229,1.224,1.203,1.188,1.182,1.179,1.180,1.182,1.184
*,1.187,1.191,1.196,2*1.4,1.390,1.344,1.317,1.295,1.276,1.255,1.242
*/
 DATA (GAMEO(L),L=637,660)/1.228,1.223,1.22,1.214,1.216,1.219,1.223
1,1.228,1.228,1.230,1.233,1.233,1.229,1.227,1.214,1.198,1.191,1.187
2,1.188,1.189,1.192,1.195,1.196,1.202/
 DATA (ENTROO(L),L=1,204)/23.68,31.56,33.97,36.43,37.92,39.01,39.88
1,40.87,41.78,42.65,43.50,44.33,45.16,45.94,46.67,47.33,47.94,48.50
2,49.04,49.58,50.10,50.59,51.08,53.05,54.88,56.70,58.46,60.16,61.93
3,63.61,65.22,66.85,68.44,22.70,30.64,33.05,35.51,37.00,38.09,38.96
4,39.93,40.84,41.70,42.54,43.32,44.08,44.84,45.56,46.22,46.81,47.39
5,48.01,48.41,48.95,49.43,49.93,51.84,53.58,55.32,57.08,58.72,60.36
6,62.00,63.58,65.10,66.66,22.17,29.95,32.36,34.62,36.31,37.40,38.25
7,39.21,40.11,40.99,41.79,42.57,43.34,44.01,44.75,45.40,45.99,46.53
8,47.10,47.56,48.02,48.51,49.02,50.85,52.61,54.32,55.99,57.64,59.24
9,60.80,62.33,63.84,65.34,21.37,29.26,31.66,34.12,35.61,36.70,37.56
*,38.50,39.40,40.24,41.01,41.79,42.52,43.22,43.92,44.56,45.15,45.70
*,46.20,46.70,47.17,47.65,48.12,49.94,51.65,53.31,54.96,56.52,58.06
*,59.61,61.10,62.57,63.99,20.46,28.54,30.75,33.21,34.70,35.79,36.63
*,37.57,38.43,39.23,40.01,40.77,41.45,42.14,42.82,43.44,44.02,44.57
*,45.09,45.55,46.03,46.49,46.93,48.73,50.35,51.98,53.54,55.06,56.56
*,58.04,59.46,60.86,62.28,19.77,27.65,30.06,32.52,34.01,35.10,35.96
```

```
*,36.89,37.72,38.53,39.27,40.00,40.72,41.36,42.00,42.62,43.20,43.74
*,44.23,44.70,45.18,45.63,46.04,47.77,49.38,50.94,52.49,53.97,55.42
*,56.87,58.23,59.60,60.98,19.07,26.96,29.36,31.82,33.31,34.40/
 DATA (ENTHOO(L),L=205,408)/35.27,36.17,37.01,37.80,38.51,39.22,39.
190,40.57,41.17,41.80,42.37,42.90,43.39,43.86,44.32,44.76,45.17,46.
280,48.42,49.93,51.40,52.87,54.31,55.66,57.00,58.36,59.68,18.16,26.
30,28.45,30.91,32.40,33.49,34.35,35.25,36.08,36.85,37.56,38.25,38.
488,39.53,40.12,40.64,41.25,41.80,42.30,42.74,43.16,43.61,44.00,45.
561,47.11,48.62,50.00,51.40,52.80,54.13,55.40,56.68,57.90,17.46,25.
634,27.75,30.21,31.70,32.79,33.64,34.55,35.38,36.12,36.81,37.50,38.
713,38.75,39.33,39.90,40.43,40.95,41.42,41.89,42.32,42.72,43.13,44.
870,46.19,47.62,49.00,50.33,51.64,52.97,54.24,55.42,56.60,16.77,24.
965,27.06,29.52,31.41,32.10,32.96,33.88,34.66,35.40,36.00,36.72,37.
*37,37.97,38.54,39.11,39.63,40.12,40.60,41.02,41.48,41.88,42.27,43.
*[illegible],45.25,46.64,47.96,49.28,50.52,51.60,53.00,54.20,55.32,15.85,23.
*74,26.14,28.60,30.49,31.18,32.03,32.97,33.75,34.45,35.11,35.76,36.
*35,36.93,37.48,38.04,38.54,39.02,39.48,39.94,40.35,40.75,41.13,42.
*06,43.99,45.33,46.65,47.87,49.06,50.26,51.45,52.58,53.67,15.16,23.
*04 15.45,27.91,29.40,30.49,31.34,32.26,33.04,33.74,34.41,35.00,35.
*61,36.16,36.69,37.23,37.73,38.19,38.66,39.10,39.50,39.89,40.28,41.
*70,43.04,44.35,45.60,46.80,47.98,49.10,50.27,51.35,52.40,14.47,22.
*35,24.76,27.22,28.71,29.83,30.67,31.57,32.35,33.02,33.68,34.28/
 DATA (ENTHOO(L),L=409,612)/34.85,35.40,35.92,36.45,36.92,37.40,37.
184,38.27,38.68,39.03,39.40,40.81,42.13,43.36,44.56,45.73,46.88,47.
298,49.07,50.15,51.15,13.55,21.43,23.84,26.30,27.79,28.88,29.75,30.
364,31.42,32.10,32.72,33.30,33.87,34.39,34.90,35.40,35.86,36.32,36.
475,37.17,37.57,37.95,38.30,39.65,40.93,42.10,43.27,44.34,45.45,46.
551,47.53,48.53,49.50,12.86,20.74,23.15,25.61,27.10,28.19,29.01,29.
697,30.71,31.39,32.00,32.60,33.15,33.68,34.14,34.62,35.10,35.54,35.
795,36.35,36.77,37.11,37.44,38.77,40.02,41.16,42.27,43.32,44.37,45.
842,46.38,47.36,48.28,12.16,20.05,22.45,24.91,26.40,27.49,28.36,29.
925,30.03,30.68,31.30,31.86,32.41,32.93,33.40,33.88,34.31,34.73,35.
*14,35.53,35.91,36.27,36.62,37.90,39.10,40.22,41.31,42.30,43.31,44.
*20,45.27,46.18,47.09,11.25,19.13,21.54,24.00,25.49,26.58,27.44,28.
*34,29.11,29.77,30.37,30.95,31.46,31.96,32.41.32.88,33.30,33.72,34.
*11,34.50,34.84,35.19,35.53,36.77,37.90,38.99,40.00,41.00,41.95,42.
*87,43.79,44.63,45.49,10.55,18.44,20.84,23.30,24.79,25.88,26.75,27.
*04,28.42,29.06,29.69,30.22,30.76,31.22,31.68,32.12,32.54,32.95,33.
*31,33.68,34.05,34.38,34.71,35.93,37.06,38.05,39.08,40.00,40.92,41.
*80,42.64,43.51,44.30,9.86,17.76,20.15,22.61,24.10,25.19,26.08,26.9
*5,27.72,28.37,28.98,29.53,30.04,30.50,30.95,31.40,31.80,32.20/
 DATA (ENTHOO(L),L=613,660)/32.55,32.91,33.24,33.59,33.90,35.09,36.
110,37.18,38.12,39.04,39.90,40.76,41.57,42.37,43.18,8.95,16.83,19.2
24,21.70,23.19,24.28,25.15,26.03,26.79,27.44,28.05,28.59,29.10,29.5
36,30.00,30.41,30.80,31.18,31.54,31.89,32.23,32.54,32.84,33.94,35.0
41,35.96,36.90,37.77,38.57,39.33,40.12,40.92,41.68/
 DATA (H2O(L),L=1,33)/0.,100.,200.,400.,600.,800.,1000.,1250.,1500.
1,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,3750.,4000.,4250.
2,4500.,4750.,5000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13
3000.,14000.,15000./
C
 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE DINT2 (XX,XT,Y1,YT1,ZZ,ZT,ML,NL,Y2,YT2,Y3,YT3,IL,YE,YT4
     1)
C
C     DOUBLE INTERPOLATION SUBROUTINE.  IF IL=1, THE SECOND INDEPENDENT
C     VARIABLE IS NOT CONSTANT WITH THE FIRST, (THE RANGE OF THE SECOND
C     IS NOT THE SAME AT EACH VALUE OF THE FIRST.)
C
      DIMENSION ZT(NL), XT(ML,NL), YT1(ML,NL), YT2(ML,NL), YT3(ML,NL)
      DIMENSION YT4(ML,NL)
      IF (ZZ-ZT(1)) 40,10,10
   10 IF (IL.EQ.1) GO TO 20
      IF (XX.LT.XT(1,1)) GO TO 40
   20 DO 30 I=1,NL
         L=I
         LL=I-1
         IF (ZZ-ZT(I)) 50,60,30
   30 CONTINUE
   40 Y1=.2222E+30
      RETURN
   50 RATIP=(ZZ-ZT(LL))/(ZT(L)-ZT(LL))
      IF (IL) 100,100,140
   60 DO 70 J=1,ML
         LM=J
         LLM=J-1
         IF (XX-XT(J,L)) 80,90,70
   70 CONTINUE
      GO TO 40
   80 RATIO=(XX-XT(LLM,L))/(XT(LM,L)-XT(LLM,L))
      Y1=YT1(LLM,L)+RATIO*(YT1(LM,L)-YT1(LLM,L))
      Y2=YT2(LLM,L)+RATIO*(YT2(LM,L)-YT2(LLM,L))
      Y3=YT3(LLM,L)+RATIO*(YT3(LM,L)-YT3(LLM,L))
      YE=YT4(LLM,L)+RATIO*(YT4(LM,L)-YT4(LLM,L))
      RETURN
   90 Y1=YT1(LM,L)
      Y2=YT2(LM,L)
      Y3=YT3(LM,L)
      YE=YT4(LM,L)
      RETURN
  100 DO 110 J=1,ML
         LM=J
         LLM=J-1
         IF (XX-XT(J,LL)) 130,120,110
  110 CONTINUE
      GO TO 40
  120 Y1=YT1(LM,LL)-RATIP*(YT1(LM,LL)-YT1(LM,L))
      Y2=YT2(LM,LL)-RATIP*(YT2(LM,LL)-YT2(LM,L))
      Y3=YT3(LM,LL)-RATIP*(YT3(LM,LL)-YT3(LM,L))
      YE=YT4(LM,LL)-RATIP*(YT4(LM,LL)-YT4(LM,L))
      RETURN
  130 RATIO=(XX-XT(LLM,L))/(XT(LM,L)-XT(LLM,L))
      Y4=YT1(LLM,LL)+RATIO*(YT1(LM,LL)-YT1(LLM,LL))
      Y5=YT1(LLM,L)+RATIO*(YT1(LM,L)-YT1(LLM,L))
      Y1=Y4-RATIP*(Y4-Y5)
      Y4=YT2(LLM,LL)+RATIO*(YT2(LM,LL)-YT2(LLM,LL))
      Y5=YT2(LLM,L)+RATIO*(YT2(LM,L)-YT2(LLM,L))
      Y2=Y4-RATIP*(Y4-Y5)
      Y4=YT3(LLM,LL)+RATIO*(YT3(LM,LL)-YT3(LLM,LL))
      Y5=YT3(LLM,L)+RATIO*(YT3(LM,L)-YT3(LLM,L))
      Y3=Y4-RATIP*(Y4-Y5)
      Y4=YT4(LLM,LL)+RATIO*(YT4(LM,LL)-YT4(LLM,LL))
      Y5=YT4(LLM,L)+RATIO*(YT4(LM,L)-YT4(LLM,L))
```

```
      YE=Y4-RATIP*(Y4-Y5)
      RETURN
  140 DO 150 J=1,ML
         IF (XX.LT.XT(1,L)) GO TO 40
         LM=J
         LLM=J-1
         IF (XX-XT(J,L)) 170,160,150
  150 CONTINUE
      GO TO 40
  160 Y5=YT1(LM,L)
      Y7=YT2(LM,L)
      Y9=YT3(LM,L)
      Y0=YT4(LM,L)
      GO TO 180
  170 RATIO=(XX-XT(LLM,L))/(XT(LM,L)-XT(LLM,L))
      Y5=YT1(LLM,L)+RATIO*(YT1(LM,L)-YT1(LLM,L))
      Y7=YT2(LLM,L)+RATIO*(YT2(LM,L)-YT2(LLM,L))
      Y9=YT3(LLM,L)+RATIO*(YT3(LM,L)-YT3(LLM,L))
  180 DO 190 J=1,ML
         Y0=YT4(LLM,L)+RATIO*(YT4(LM,L)-YT4(LLM,L))
         LM=J
         LLM=J-1
         IF (XX-XT(J,LL)) 210,200,190
  190 CONTINUE
      GO TO 40
  200 Y4=YT1(LM,LL)
      Y6=YT2(LM,LL)
      Y8=YT3(LM,LL)
      YA=YT4(LM,LL)
      GO TO 220
  210 RATIO=(XX-XT(LLM,LL))/(XT(LM,LL)-XT(LLM,LL))
      Y4=YT1(LLM,LL)+RATIO*(YT1(LM,LL)-YT1(LLM,LL))
      Y6=YT2(LLM,LL)+RATIO*(YT2(LM,LL)-YT2(LLM,LL))
      Y8=YT3(LLM,LL)+RATIO*(YT3(LM,LL)-YT3(LLM,LL))
      YA=YT4(LLM,LL)+RATIO*(YT4(LM,LL)-YT4(LLM,LL))
  220 Y1=Y4-RATIP*(Y4-Y5)
      Y2=Y6-RATIP*(Y6-Y7)
      Y3=Y8-RATIP*(Y8-Y9)
      YE=YA-RATIP*(YA-Y0)
      RETURN
C
      END
```

```
      SUBROUTINE DINT1 (XX,XT,Y1,YT1,ZZ,ZT,ML,NL)
C
C     DOUBLE INTERPOLATION SUBROUTINE.
C
      DIMENSION ZT(NL), XT(ML,NL), YT1(ML,NL)
      IF (ZZ-ZT(1)) 20,10,10
   10 IF (XX-XT(1,1)) 20,30,30
   20 Y1=.2222E+30
      RETURN
   30 DO 40 I=1,NL
         L=I
         LL=I-1
         IF (ZZ-ZT(I)) 50,60,40
   40 CONTINUE
      GO TO 20
   50 RATIP=(ZZ-ZT(LL))/(ZT(L)-ZT(LL))
      GO TO 100
   60 DO 70 J=1,ML
         LM=J
         LLM=J-1
         IF (XX-XT(J,L)) 80,90,70
   70 CONTINUE
      GO TO 20
   80 RATIO=(XX-XT(LLM,L))/(XT(LM,L)-XT(LLM,L))
      Y1=YT1(LLM,L)+RATIO*(YT1(LM,L)-YT1(LLM,L))
      RETURN
   90 Y1=YT1(LM,L)
      RETURN
  100 DO 110 J=1,ML
         LM=J
         LLM=J-1
         IF (XX-XT(J,L)) 130,120,110
  110 CONTINUE
      GO TO 20
  120 Y5=YT1(LM,L)
      GO TO 140
  130 RATIO=(XX-XT(LLM,L))/(XT(LM,L)-XT(LLM,L))
      Y5=YT1(LLM,L)+RATIO*(YT1(LM,L)-YT1(LLM,L))
  140 DO 150 J=1,ML
         LM=J
         LLM=J-1
         IF (XX-XT(J,LL)) 170,160,150
  150 CONTINUE
      GO TO 20
  160 Y4=YT1(LM,LL)
      GO TO 180
  170 RATIO=(XX-XT(LLM,LL))/(XT(LM,LL)-XT(LLM,LL))
      Y4=YT1(LLM,LL)+RATIO*(YT1(LM,LL)-YT1(LLM,LL))
  180 Y1=Y4-RATIP*(Y4-Y5)
      RETURN
C
      END
```