N7716290

| | | |
|---|---|---|
| 1. Report No.<br>NASA CR-2742 | 2. Government Accession No. | 3. Recipient's Catalog No.<br>N77-16290 |
| 4. Title and Subtitle<br>STAN5 - A PROGRAM FOR NUMERICAL COMPUTATION OF TWO-DIMENSIONAL INTERNAL AND EXTERNAL BOUNDARY LAYER FLOWS | | 5. Report Date<br>November 1976<br>6. Performing Organization Code |
| 7. Author(s)<br>M. E. Crawford and W. M. Kays | | 8. Performing Organization Report No.<br>HMT-23<br>10. Work Unit No. |
| 9. Performing Organization Name and Address<br>Stanford University<br>Stanford, California 94305 | | 11. Contract or Grant No.<br>NGR-05-020-134, NAS3-14336, NSF GK534, NSF GK2201<br>13. Type of Report and Period Covered<br>Contractor Report |
| 12. Sponsoring Agency Name and Address<br>National Aeronautics and Space Administration<br>Washington, D.C. 20546 | and National Science Foundation<br>Washington, D.C. 20550 | 14. Sponsoring Agency Code |
| 15. Supplementary Notes<br>Final Report. Project Manager, Raymond S. Colladay, Fluid System Components Division, NASA Lewis Research Center, Cleveland, Ohio | | |

16. Abstract

This report describes a computer program for solution of the boundary layer equations. The program is an outgrowth of the original procedure developed by Patankar and Spalding at Imperial College, London. Included in the report is a listing of the program and sample data sets. A large variety of two-dimensional flows can be accommodated by the program, including boundary layers on a flat plate, flow inside nozzles and diffusers (for a prescribed potential flow distribution), flow over axisymmetric bodies, and developing and fully developed flow inside circular pipes and flat ducts. The flows may be laminar or turbulent, and provision is made to handle transition. Turbulence modeling includes (1) Prandtl mixing-length scheme throughout the flow; (2) a turbulent kinetic energy (TKE) scheme; or (3) an eddy diffusivity function. For the latter two models, the mixing-length scheme is used in the sublayer region. The program solves the momentum equation, as a minimum, plus any number of diffusion equations. The stagnation enthalpy equation and TKE equation are solved by using the concept of a turbulent Prandtl/Schmidt number. Fluid properties may be treated as constant or variable. Initial boundary layer profiles are user-supplied.

| | |
|---|---|
| 17. Key Words (Suggested by Author(s))<br>Boundary layer, Numerical<br>Heat transfer<br>Computer program<br>Turbulence | 18. Distribution Statement<br>Unclassified - unlimited<br>STAR category 34 |
| 19. Security Classif. (of this report)<br>Unclassified | 20. Security Classif. (of this page)<br>Unclassified |

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

## NOMENCLATURE

| | |
|---|---|
| a | constant in program correlation for $A^+$ or $B^+$, or constant in constant eddy diffusivity model, or coefficient in transformed equation (4.7). |
| $A_q$ | production constant, turbulent kinetic energy equation. |
| $A^+$ | damping constant, van Driest damping function (see equation 2.24 for correlation). |
| b | constant in program correlation for $A^+$ or $B^+$, or constant in constant eddy diffusivity model, or coefficient in transformed equation (4.7). |
| bf | generalized x-direction body force, momentum equation. |
| $B_q$ | dissipation constant, turbulent kinetic energy equation. |
| $B^+$ | damping constant, Evans damping function (see equation 2.24 for program correlation). |
| c | constant in program correlation for $A^+$ or $B^+$, or constant in variable turbulent Prandtl number model, or specific heat of fluid, or coefficient in transformed equation (4.7). |
| C | constant in differential lag equation to compute effective $P^+$ or $V_o^+$. |
| $C_f/2$ | friction coefficient, $g_c\tau_o/(\rho_\infty U_\infty^2)$, or $g_c\tau_o/\bar{\rho}\bar{U}^2)$ for pipe and channel flows. |
| d | coefficient in transformed equation (4.7). |
| D | damping function to suppress mixing length in the region immediately adjacent to a wall, equation (2.22) and (2.23). |
| $\mathcal{D}$ | dissipation term, turbulent kinetic energy equation. |
| E-surface | see Figure 4.1. |
| $\dot{E}_{total}$ | total energy flux boundary condition at a wall, $\dot{m}''_o I^*_o + \dot{q}''_o$ (see Figure 2.2). |
| g | local gravitational constant to determine free-convection body force. |
| $g_c$ | proportionality constant, Newton's Second Law. |
| i' | fluctuation in static enthalpy. |

| | |
|---|---|
| $i^{*'}$ | fluctuation in stagnation enthalpy. |
| I | static enthalpy of fluid. |
| I-surface | see Figure 4.1. |
| $I^*$ | stagnation enthalpy of fluid, $I + U^2/(2g_c J)$. |
| $I^{*+}$ | non-dimensional stagnation enthalpy, $(I_o^* - I^*)\ U_\tau/(\dot{q}_o''/\rho_o)$. |
| J | conversion constant, mechanical to thermal energy. |
| $J_q$ | diffusion term, turbulent kinetic energy equation. |
| k | thermal conductivity of fluid. |
| $\ell$ | mixing-length (see section 2.3.1) |
| $\dot{m}''$ | mass flux at I or E surface (see Figures 2.2 and 4.1). |
| Nu | Nusselt number, pipe and channel flow, $St \cdot \overline{Pr} \cdot Re$. |
| P | thermodynamic pressure. |
| $P^+$ | non-dimensional pressure, $g_c \nu_o (dP/dx)/(\rho_o U_\tau^3)$ |
| $Pe_t$ | turbulent Peclet number, program correlation for $Pr_t$ . |
| Pr | Prandtl number, $\mu c/k$ . |
| $Pr_{eff}$ | combined laminar and turbulent Prandtl number, equation (2.14). |
| $Pr_t$ | turbulent Prandtl number, $\varepsilon_M/\varepsilon_H$ (see equation 2.37 for program correlation). |
| $\dot{q}''$ | combined laminar and turbulent heat flux, Figure 2.2 and equation (3.2). |
| $q^+$ | non-dimensional heat flux, $\dot{q}''/\dot{q}_o''$ . |
| $q^2/2$ | turbulent kinetic energy |
| r | radius |
| Re | pipe or channel flow Reynolds number, equation (3.29). |
| $Re_H$ | enthalpy thickness Reynolds number, $\Delta_2\ U_\infty/\nu_\infty$ |
| $Re_M$ | momentum thickness Reynolds number, $\Delta_2\ U_\infty/\nu_\infty$ |
| $Re_{tran}$ | Reynolds number (Re or $Re_M$) for transition from laminar to turbulent flow. |
| s | generalized energy source, stagnation enthalpy equation. |

| | |
|---|---|
| $s^+$ | non-dimensional generalized energy source, $\nu_o s/(\dot{q}_o'' U_\tau)$ . |
| S | energy source term, stagnation enthalpy equation, $UX/J + s$ . |
| $Sc_q$ | turbulent Schmidt number, $\varepsilon_M/\varepsilon_q$ . |
| St | Stanton number, $\dot{q}_o''/\{\rho_\infty U_\infty (I_o^* - I_\infty^*)\}$ , or $\dot{q}_o''/\{\overline{\rho}\,\overline{U}(I_o^* - \overline{I}^*)\}$ . |
| $T_u$ | longitudinal free-stream turbulence intensity, $\sqrt{\overline{u'^2}} / U_\infty$ . |
| u' | fluctuation in U component of velocity . |
| U | velocity component in x-direction. |
| $U_\tau$ | shear velocity $\sqrt{g_c \tau_o/\rho_o}$ . |
| $U^+$ | non-dimensional U velocity component $U/U_\tau$ . |
| v' | fluctuation in V component of velocity . |
| V | velocity component in y-direction. |
| $V_o{}^+$ | non-dimensional V velocity component at wall, $V_o/U_\tau$ or $(\dot{m}_o''/\rho_o)/U_\tau$ . |
| W | $\rho_o\, U_\tau{}^3/(g_c\, J\, \dot{q}_o'')$ |
| x | distance along surface (see Figures 2.1 and 4.1). |
| $x^+$ | non-dimensional x distance, $xU_\tau/\nu_o$ . |
| X | body force term, momentum equation, $\frac{\rho g}{g_c} + bf$ . |
| $X^+$ | non-dimensional body force term, $g_c \nu_o X/(\rho_o U_\tau{}^3)$ . |
| y | distance normal to surface (see Figures 2.1a and 4.1). |
| $y^+$ | non-dimensional y distance, $yU_\tau/\nu_o$ . |
| $\alpha$ | angle between surface tangent and axis-of-symmetry line (see Figures 2.1a and 4.1), or constant in internal correlation for $Pr_t$ . |
| $\beta$ | power-law coefficient velocity equation slip scheme. |
| $\gamma$ | power-law coefficient, diffusion equation slip scheme. |
| $\delta_1$ | displacement thickness, equation (3.22a). |
| $\delta_2$ | momentum thickness, equation (3.22b). |
| $\delta_{.99}$ | boundary layer thickness where $U/U_\infty = 0.99$ . |

| | |
|---|---|
| $\Delta_2$ | enthalpy thickness, equation (3.22c). |
| $\varepsilon_H$ | eddy diffusivity for heat. |
| $\varepsilon_M$ | eddy diffusivity for momentum. |
| $\varepsilon_q$ | eddy diffusivity for turbulent kinetic energy. |
| $\kappa$ | Karman constant, mixing-length model. |
| $\lambda$ | outer length scale constant mixing-length model. |
| $\lambda_o$ | program input value of $\lambda$ . |
| $\mu$ | dynamic viscosity of fluid. |
| $\mu_{eff}$ | combined laminar and turbulent viscosity, equation (2.6). |
| $\mu^+$ | non-dimensional viscosity, $\mu_{eff}/\mu_o$ . |
| $\nu$ | kinematic viscosity of fluid. |
| $\rho$ | density of fluid. |
| $\tau$ | combined laminar and turbulent shear stress, equation (3.1). |
| $\tau^+$ | non-dimensional shear stress, $\tau/\tau_o$ . |
| $\phi$ | generalized dependent variable in transformed equation (4.7). |
| $\psi$ | stream function coordinate. |
| $\omega$ | non-dimensional stream function coordinate. |

<u>Subscripts</u>

| | |
|---|---|
| axi | axisymmetric (see section 3.6). |
| d | downstream edge of finite-difference control volume. |
| e | edge of shear layer, equation (2.18). |
| eff | effective value. |
| fp | "flat plate" value, without transpiration or pressure gradient. |
| eq | equilibrium value, equation (2.25). |
| N | number of stream tubes. |
| $\left.\begin{matrix} o \\ w \end{matrix}\right\}$ | wall value. |
| t | turbulent value. |

| | |
|---|---|
| u | upstream edge of finite-difference control volume. |
| $\infty$ | free-stream value. |
| 2.5 | join-point value. |

Superscript

| | |
|---|---|
| overbar | time averaged quantity, or bulk mean value (Section 3.7). |

## Chapter 1

## INTRODUCTION

In recent years it has become practicable and popular to compute turbulent boundary layers using finite-difference techniques and the digital computer. These techniques have now been developed to the point where one can readily develop one's own program for particular applications, and numerous workers have described their programs in the literature and have made listings or card decks available to others. There is no question that the development of one's own program is a tedious process and the programs become sufficiently complex that a great deal of development effort is usually required. For the user who doesn't expect to devote a great amount of time (and money) on a program it is often more practicable to make use of someone else's program, provided that the program is sufficiently well documented that it can be used intelligently.

It is the objective of this report to describe one such program which has gone through a considerable period of development, and which has been found useful in connection with an experimental turbulent boundary layer research program at Stanford University. Enough people have asked for copies of this program that it seems worthwhile to provide in a more formal way the documentation that is really necessary if the program is to be used properly.

No claim of superiority is made; in fact, there is no question that there are other programs developed for particular applications that are faster and are in some cases even more precise. However, this program is believed to be unique in its degree of generality, in the large variety of different kinds of problems that can be handled, and, in particular, in an input-output scheme that makes it possible to handle a great variety of problems without touching the deck. Very minor modifications in the deck open up a whole realm of additional possibilities.

The original basic program from which this one was developed was the Patankar/Spalding program described in their 1967 book [1]. Much of that program will be recognized in this present version, and a complete understanding of all the details of the present program may require reference to that publication. However, it is hoped that this description will be sufficiently complete to make further study unnecessary in most cases. A later revision of the Spalding program was published in 1970 [2] in which a number of important improvements were made. Some of these improvements have been incorporated in the present version, and it is our belief that the present version suffers in comparison only with respect to size and speed, and perhaps in accuracy for some unusual types of problems. The largest source of inaccuracy and uncertainty in turbulent boundary layer finite-difference procedures lies in the methods used to model the turbulence, and this has nothing to do with the computational procedure.

The basic features of the program will now be described, and then elaborated upon in the chapters that follow.

The program is designed to solve two-dimensional parabolic differential equations only, i.e., the boundary layer equations incorporating the usual boundary layer approximations. The eddy diffusivity concept must be used in modeling the turbulent stresses, although beyond that point there is great flexibility. The program does not handle re-circulating flows.

The program solves the momentum equation of the boundary layer, as a minimum, plus any number of diffusion equations, all simultaneously. The listing presented in the Appendix is dimensioned to a maximum of five diffusion equations, and the output routine handles only five, but it is a simple matter to increase this number if desired.

A coordinate system for axi-symmetric flows is used so that a large variety of flow types can be accommodated by simple manipulation of variables in the input routine. These include the boundary layer on a flat plate, flow inside nozzles and diffusers (for a prescribed potential flow distribution), flow over axi-symmetric bodies, both developing and fully developed flow inside circular pipes and flat ducts, circular and flat jets and free-shear flows. As presently set up, the program provides for one wall surface, and thus the duct-flow problems are limited to simple pipes and flat ducts with symmetrical boundary conditions. In principle there is no reason why two walls, such as are encountered in circular tube annuli, cannot be handled, but this does require some additional program modification.

The program solves laminar boundary layers as well as turbulent boundary layers, and provision is made for a transition from a laminar to a turbulent boundary layer based on a momentum thickness Reynolds number criterion. Solution of laminar boundary layers is of necessity slower than is possible with programs developed for laminar boundary layers alone, because the program was developed for the more complex turbulent problems.

Fluid properties are treated as variable with the properties of any particular fluid supplied through a separate subroutine. In the present program listing the only fluid properties provided are those of air (essentially the Keenan and Kaye Gas Tables). Properties of other fluids may be introduced by attaching additional property subroutines. Fluid properties may also be treated as constant, in which case the properties are introduced directly into the input routine. The types of problems that can be handled with the present listing are obviously limited by inclusion of only the properties of air. For example, the program could readily solve a binary diffusion problem, together with heat transfer, but it would be necessary to append an additional properties subroutine unless the constant properties mode is deemed adequate.

Viscous dissipation in the energy equation is included as an option controllable through the input routine, so high velocity flows can be readily solved. Provision is also made for introducing axial body forces and internal heat generation. A particular provision is made to introduce an axial gravity force, and this together with the variable property option allows solution of both laminar and turbulent free-convection problems.

In principle the chemically reacting boundary layer may be solved to various degrees of approximation, but this does require the addition of source terms which are not included in the present listing.

Any kind of initial conditions can be accommodated, and the boundary condition possibilities using the input routine, while not infinite, are nevertheless large. Free-stream velocity, rather than pressure, is treated as a variable boundary condition, and heat and mass flux along a wall may assume any values. Alternatively, wall enthalpy (or concentration in the case of mass diffusion) and mass flux may be treated as independent. In the case of duct flows there is no free-stream and pressure is computed as a dependent variable.

Several possibilities for turbulence modeling are included and can be activated in a simple manner in the input routine. The Prandtl mixing-length scheme may be used throughout, or, alternatively, a one-differential-equation turbulent kinetic energy scheme may be used for the flow outside the sublayer region. This alternative involves solution of the turbulent kinetic energy differential equation of the boundary layer, which is simply another diffusion equation. As another possibility, eddy diffusivity in the outer part of the boundary layer may be evaluated as an empirical function of Reynolds number. In all cases a mixing-length scheme is used to calculate the sublayer near the wall, and two possibilities are programmed. In one the Van Driest exponential damping function is used, while in the other the Evans linear damping function is used. Internal empirical correlations for the damping constants to account for effects of pressure gradient and transpiration are contained in the program, or, alternatively, the user can supply his own constants. Other variations in the turbulence physics can be quite easily made, but this does require some re-programming.

The energy equation, and any other type of diffusion equations, is solved through the concept of turbulent Prandtl number (or turbulent Schmidt number). The program contains an internal calculation for turbulent Prandtl number as a function of turbulent Peclet number, which gives reasonably good results over the entire spectrum of Prandtl number, including the liquid metal region. Alternatively, the user may specify his own turbulent Prandtl number.

The concepts of "slip" values at the wall and a "Wall Function" are employed, allowing the use of a relatively coarse grid in the direction normal to a wall surface. The region adjacent to the wall is computed by numerically integrating the Couette flow forms of the boundary layer equations, but with physics input identical to that used outside the wall region. This option can, however, be bypassed, but at the cost of a greatly increased number of grid points near the wall. The Wall Function is especially useful in high Reynolds number applications where the number of cross-stream grid points can otherwise become excessive.

The program is "almost" independent of any particular dimensioning system. It would be completely independent were it not for the fact that the property subroutine for air which is packaged with the program is based on Btu, ft, $lb_m$ units. The dimensioning system to be used is designated in the input routine by two constants.

Finally, a word about the differencing scheme employed is in order, because in this respect it differs from many other programs. A fully implicit scheme is employed for the main dependent variables (velocity, enthalpy, mass concentration, etc.), and this, together with the fact that the conservation equations are always satisfied, in principle allows large forward steps to be taken without stability problems. However, fluid properties and turbulence properties are handled explicitly, and if these are changing markedly in the flow direction it is not possible to take large forward steps without stability and accuracy problems. The advantage is that nowhere is iteration required. This restriction to relatively small forward steps (typically about one or two boundary layer thicknesses) is not necessarily disadvantageous, because one of the reasons for making finite-difference calculations is that variable boundary conditions can be easily handled, and there is often a need for output data, such as heat flux, at frequent intervals along a surface. Both of these requirements dictate a small forward step size anyway.

The remaining chapters of this report will now expand upon this brief description, culminating in detailed instructions about how to set up a problem and use the input routine. It might well be noted here, however, that the input subroutine (which is actually packaged at the end of the program) contains very extensive descriptive comments, suggestions, and instructions, and is thus a convenient summary of much of this report.

## Chapter 2

## DIFFERENTIAL EQUATIONS AND TURBULENCE MODELS

### 2.1 Convective Transport Equations

The types of flows modeled by STAN5 are those described by the parabolic boundary layer equations, which include the continuity, momentum, and stagnation enthalpy equations. They are written to describe flow of a turbulent, compressible fluid over an axi-symmetric body. All equations have been time-averaged, and in the equations all dependent variables and properties are either mean quantities or fluctuating quantities (as denoted by primes). They are also applicable to laminar flows, in which case the turbulent stress and heat flux are ignored. Figure 2.1 describes the coordinate system and typical velocity and stagnation enthalpy profiles. Note the coordinate system is written in terms of the independent variables, $x$ and $y$. The radius, $r$, is a transverse radius of curvature and is related to $y$ as shown in Figure 2.1(a), and the longitudinal radius of curvature is neglected (i.e., $\alpha(x)$ in Figure 2.1(a) varies slowly with $x$).

#### 2.1.1 The Continuity Equation

The time-averaged continuity equation for this coordinate system is given by

$$\frac{\partial}{\partial x}(r\rho U) + \frac{\partial}{\partial y}(r\rho V) = 0 \quad . \tag{2.1}$$

In the above equation and the momentum and energy equations which follow, thermodynamic quantity-velocity fluctuation correlations are neglected.

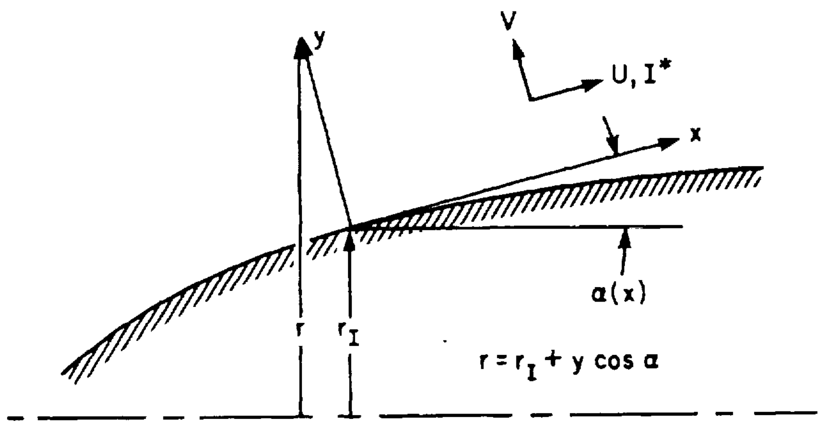
#### 2.1.2 The Momentum Equation

The time-averaged momentum equation in the x-direction is given by

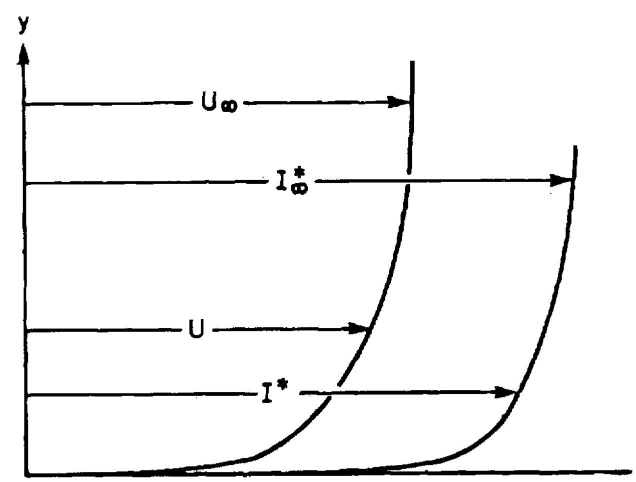
$$\rho U \frac{\partial U}{\partial x} + \rho V \frac{\partial U}{\partial y} = -g_c \frac{dP}{dx} + \frac{1}{r}\frac{\partial}{\partial y}\left[r\left(\mu \frac{\partial U}{\partial y} - \rho\, \overline{u'v'}\right)\right] + g_c X \quad . \tag{2.2}$$

In the program, the body-force term in equation (2.2) is decomposed into

$$X = \frac{\rho g}{g_c} + bf \quad , \tag{2.3}$$




(a) Coordinate system




(b) Velocity and stagnation enthalpy profiles

Figure 2.1. Notation for the differential equations and profiles.

where the first term is a free convection body force in the positive x direction and bf is a generalized, x-direction body force with units of (force/unit volume). The bf term might be used to model magnetohydrodynamic body forces.

Pressure gradient is computed for pipe/channel flows as described in [1,2]. For flows over a surface dP/dx is computed in terms of the free-stream velocity and body force,

$$- g_c \left(\frac{dP}{dx}\right) = \rho_\infty U_\infty \frac{dU_\infty}{dx} - g_c X_\infty \quad . \tag{2.4}$$

In the momentum equation, the turbulent shear stress, $-\overline{u'v'}$, is modeled using the eddy diffusivity for momentum, $\varepsilon_M$, as defined by

$$-\overline{u'v'} = \varepsilon_M \frac{\partial U}{\partial y} = \frac{\mu_t}{\rho} \frac{\partial U}{\partial y} \quad , \tag{2.5}$$

where $\mu_t$ is the turbulent viscosity. The laminar viscosity combines with the turbulent viscosity to obtain an effective viscosity

$$\mu_{eff} = (\mu + \mu_t) = \rho(\nu + \varepsilon_M) \quad . \tag{2.6}$$

Combining equations (2.2), (2.5), and (2.6) yields the final form for the momentum equation that is programmed.

$$\rho U \frac{\partial U}{\partial x} + \rho V \frac{\partial U}{\partial y} = - g_c \frac{dP}{dx} + \frac{1}{r} \frac{\partial}{\partial y}\left[r \mu_{eff} \frac{\partial U}{\partial y}\right] + g_c X \quad . \tag{2.7}$$

### 2.1.3. The Stagnation Enthalpy Equation

The time-averaged stagnation enthalpy equation is given by

$$\rho U \frac{\partial I^*}{\partial x} + \rho V \frac{\partial I^*}{\partial y} = \frac{1}{r} \frac{\partial}{\partial y}\left\{ r \left[ \frac{k}{c} \frac{\partial I}{\partial y} - \rho \overline{i^{*\prime} v'} + \frac{\mu}{g_c J} \frac{\partial}{\partial y}\left(\frac{U^2}{2}\right)\right]\right\} + S \tag{2.8}$$

where $I^*$ is the stagnation enthalpy of the fluid, defined as $I^* = I + U^2/2g_cJ$, and I is the static enthalpy.

In the program, the energy source term in equation (2.8) is decomposed into

$$S = \frac{UX}{J} + s \tag{2.9}$$

where the first term is work done against x-direction body forces and $s$ is a generalized source (energy rate/unit volume). The $s$ term might be used to model Joulean heating for an electrically conducting fluid or nuclear heating.

In equation (2.8), a model for $-\overline{i^{*'}v'}$ is required. The term is a correlation involving fluctuations in stagnation enthalpy and cross-stream velocity, and is approximated as

$$-\overline{i^{*'}v'} \simeq -\overline{i'v'} + U(-\overline{u'v'}) \quad , \tag{2.10}$$

where $i'$ is fluctuation in static enthalpy. The turbulent heat flux, $-\overline{i'v'}$, is modeled using the concept of eddy diffusivity for heat, $\varepsilon_H$, as defined by

$$-\overline{i'v'} = \varepsilon_H \frac{\partial I}{\partial y} = \left(\frac{k_t/c}{\rho}\right)\frac{\partial I}{\partial y} \quad , \tag{2.11}$$

where $k_t$ is the turbulent conductivity. The eddy diffusivities for heat and momentum are related through the turbulent Prandtl number,

$$Pr_t = \frac{\varepsilon_M}{\varepsilon_H} \quad . \tag{2.12}$$

The laminar conductivity combines with the turbulent conductivity to form an effective conductivity (divided by specific heat, $c$),

$$\left(\frac{k}{c}\right)_{eff} = \frac{k}{c} + \left(\frac{k}{c}\right)_t \quad . \tag{2.13}$$

Equations (2.6), (2.12), and (2.13) are combined to form an effective Prandtl number,

$$Pr_{eff} = \frac{\mu_{eff}}{(k/c)_{eff}} = \frac{1 + \frac{\varepsilon_M}{\nu}}{\frac{1}{Pr} + \frac{\varepsilon_M}{\nu}\frac{1}{Pr_t}} \quad . \tag{2.14}$$

Equations (2.5), (2.10), (2.11), and the definitions for $\mu_{eff}$ and $Pr_{eff}$ are combined with equation (2.8) to give the final form of the stagnation enthalpy equation that is programmed.

$$\rho U \frac{\partial I^*}{\partial x} + \rho V \frac{\partial I^*}{\partial y} = \frac{1}{r}\frac{\partial}{\partial y}\left\{ r \left[ \frac{\mu_{eff}}{Pr_{eff}} \frac{\partial I^*}{\partial y} + \frac{\mu_{eff}}{g_c J}\left(1 - \frac{1}{Pr_{eff}}\right) \frac{\partial}{\partial y}\left(\frac{U^2}{2}\right)\right]\right\} + S \quad . \tag{2.15}$$

## 2.2 Boundary Conditions

For boundary layer flows in which there are a wall and a free stream, e.g., flow over a flat surface or a body of revolution, the boundary conditions for the momentum equation are given by

$$U(x,0) = 0 \quad , \tag{2.16a}$$

$$V(x,0) = \dot{m}''_o(x)/\rho \quad , \tag{2.16b}$$

$$\lim_{y \to \infty} U(x,y) = U_\infty(x) \quad , \tag{2.16c}$$

where $\dot{m}''_o(x)$ is wall mass transfer per unit area due to fluid injection or suction.

Boundary conditions for the stagnation enthalpy equation are given by

$$I^*(x,0) = I^*_o(x) \quad , \quad \text{or}$$
$$\dot{q}''(x,0) = -\frac{k}{c}\frac{\partial I^*(x,0)}{\partial y} = \dot{q}''_o(x) \quad , \tag{2.16d}$$

$$\lim_{y \to \infty} I^*(x,y) = I^*_\infty \text{ (constant)} \quad . \tag{2.16e}$$

The wall boundary condition (2.16d) is either a level or a flux. For both cases, if there is transpiration at the surface, the transpired fluid is assumed to leave the surface in thermal equilibrium with it. If a flux boundary condition is specified, then the program requires specification of the <u>total</u> energy flux from the surface. This is related to the surface heat flux, $\dot{q}''_o(x)$ as shown in Figure 2.2 for a differential element of surface area.

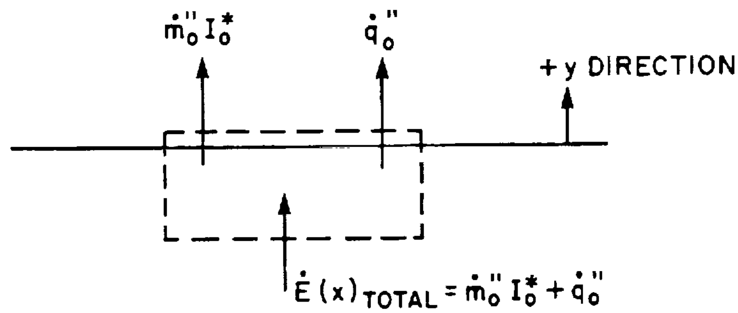



Figure 2.2. Wall flux boundary condition.

Boundary layer-type flows with a wall and a line of symmetry, e.g., flow in a circular pipe or a flat duct, have the following boundary conditions at the centerline, $y = 0$, and wall, $y = r_w$.

$$U(x,r_w) = 0 \quad , \tag{2.17a}$$

$$V(x,r_w) = 0 \quad , \tag{2.17b}$$

$$\frac{\partial U(x,0)}{\partial y} = 0 \quad , \tag{2.17c}$$

$$\left.\begin{aligned} I^*(x,r_w) &= I_o^*(x) \quad , \quad \text{or} \\ \dot{q}''(x,r_w) &= \dot{q}_o''(x) \quad , \end{aligned}\right. \tag{2.17d}$$

$$\frac{\partial I^*(x,0)}{\partial y} = 0 \quad . \tag{2.17e}$$

Because such flows are confined flows, the pressure gradient must be determined. This is accomplished indirectly in the program by linking it to conservation of mass: a pressure gradient is computed to conserve the mass flow rate as the momentum equation is integrated in the x-direction.

Boundary layer flows with a free surface and a line of symmetry, e.g., jets and free shear flows, have the following boundary conditions at the centerline, $y = 0$, and the edge of the shear layer, $r_e$.

$$\frac{\partial U(x,0)}{\partial y} = 0 \quad , \tag{2.18a}$$

$$V(x,0) = 0 \quad , \tag{2.18b}$$

$$\lim_{r \to r_e} U(x,r) = U_\infty(x) \quad , \tag{2.18c}$$

$$\frac{\partial I^*(x,0)}{\partial y} = 0 \quad , \tag{2.18d}$$

$$\lim_{r \to r_e} I^*(x,r) = I^*_\infty \text{ (constant)} \quad . \tag{2.18e}$$

## 2.3 Turbulent Shear Stress

Turbulent shear stress is modeled using the eddy diffusivity for momentum. The program incorporates three options for modeling $\varepsilon_M$, as follows.

### 2.3.1 Prandtl Mixing-Length Model for $\varepsilon_M$

The Prandtl mixing-length model relates eddy diffusivity for momentum to the mean velocity gradient by defining a mixing-length, $\ell$, such that

$$\varepsilon_M = \ell^2 \left| \frac{\partial U}{\partial y} \right| \quad . \tag{2.19}$$

The mixing-length for the region near the wall but outside the viscous region immediately adjacent to the wall is given by

$$\ell = \kappa y \quad . \tag{2.20}$$

A suggested value for $\kappa$ is 0.41.

Immediately adjacent to the wall, the viscous sublayer is modeled by introducing a damping function, $D$, that effectively suppresses the linear dependence of equation (2.20). With the damping function, the mixing-length for the viscous region becomes

$$\ell = \kappa y D \quad . \tag{2.21}$$

Two damping function options are available in the program. The first type is the Van Driest damping function,

$$D = 1.0 - \exp[-y^{+}(\nu_o/\nu)/A^{+}] \quad , \tag{2.22}$$

where $y^{+}(\nu_o/\nu)$ is the non-dimensional distance from the wall, expressed in "wall coordinates", defined in Section 3.2, and $A^{+}$ is an effective sublayer thickness defined in an analogous manner. The second type of damping function in the program is the Evans damping function,

$$D = \begin{cases} y^{+}(\nu_o/\nu)/B^{+} \quad , & y^{+}(\nu_o/\nu) \le B^{+} \\ 1.0 \quad , & y^{+}(\nu_o/\nu) > B^{+} \end{cases} \tag{2.23}$$

where $B^{+}$ is an effective sublayer thickness.

The effective thickness of the viscous sublayer is probably the single most important parameter in computation of turbulent boundary layers. The sublayer, though comprising a very small fraction of the total boundary layer thickness, is the region where the major change in velocity takes place and, except for very low Prandtl number fluids, is the region wherein most of the resistance to heat transfer resides. If this region is modeled accurately, only a very approximate scheme is needed throughout the rest of the boundary layer.

Thickness of the sublayer is evidently determined by viscous stability considerations. The experimental evidence is that a favorable pressure gradient (dP/dx negative) results in increased thickness, while an adverse pressure gradient has the opposite effect. Transpiration into the boundary layer (blowing) decreases the thickness, if it is expressed in non-dimensional wall coordinates, while suction has the opposite effect. Surface roughness, while not a subject of this paper, causes a thinning of the sublayer.

The effects of pressure gradient and transpiration on $A^{+}$ or $B^{+}$ are conveniently expressed in terms of a non-dimensional pressure gradient parameter, $P^{+}$, and a non-dimensional blowing parameter, $V_o^{+}$, both of which can be either positive or negative. In both of these parameters the main argument is normalized with respect to the same wall coordinate parameters as is the effective sublayer thickness $A^{+}$ or $B^{+}$.

The functional dependence of $A^+$ upon $P^+$ and $V_o^+$ has been deduced experimentally by examination of a very large number of velocity profiles obtained at Stanford [3]. This functional dependence can be directly related to $B^+$, and both can be expressed algebraically as

$$\left.\begin{matrix} A^+ \\ B^+ \end{matrix}\right\} = \frac{A^+_{fp} \text{ or } B^+_{fp}}{a\left[V_o^+ + b\left(\dfrac{P^+}{1 + cV_o^+}\right)\right] + 1.0}, \tag{2.24}$$

where

$a = 7.1$ if $V_o^+ \geq 0.0$, otherwise $a = 9.0$;

$b = 4.25$ if $P^+ \leq 0.0$, otherwise $b = 2.9$;

$c = 10.0$ if $P^+ \leq 0.0$, otherwise $c = 0.0$.

A recommended value for $A^+_{fp}$ and $B^+_{fp}$ are 25 and 35, respectively.

Equation (2.24) is plotted on Figure 2.3 for $A^+$, and in the graph the effects of pressure gradient and transpiration can be clearly seen. Note that a strong favorable pressure gradient forces $A^+$ to very high values, and that blowing lessens this effect, while suction increases it. If $A^+$ becomes very large, the viscous sublayer simply overwhelms the entire boundary layer, resulting in re-laminarization. The thickening of the sublayer caused by a favorable pressure gradient (accelerating flows) results in a decreased Stanton number simply because the major resistance to heat transfer is in the viscous sublayer.

$A^+$, as represented by equation (2.24) and Figure 2.3, has been evaluated under essentially equilibrium conditions, i.e., conditions under which $V_o^+$ and/or $P^+$ are invariant or, at worst, are varying only slowly along the surface. This is the case of inner region equilibrium. It is probable that when a sudden change of external conditions is imposed, the inner region comes to equilibrium more rapidly than the outer region, although this has not been proved. In any case, under non-equilibrium conditions where $V_o^+$ or $P^+$ are changing rapidly, it has been observed that the sublayer does not change instantaneously to its new equilibrium thickness, i.e., $A^+$ does not immediately

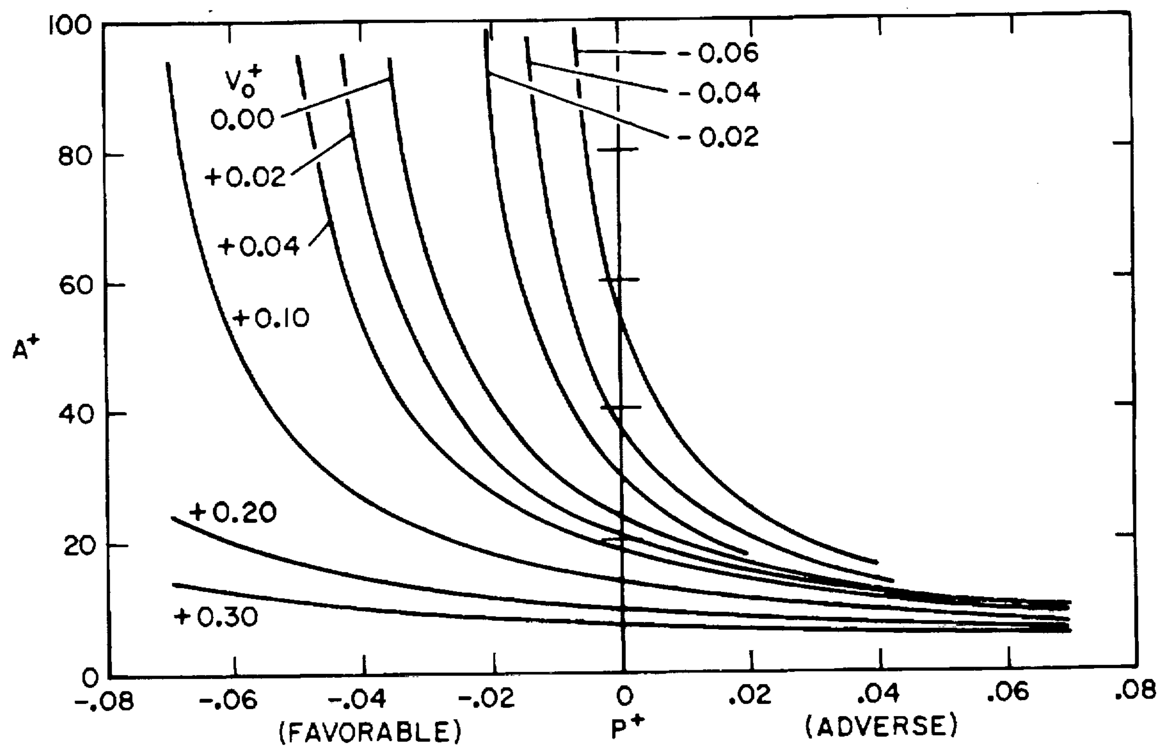



Figure 2.3. The variation of the damping constant, $A^+$, with $V_o^+$ and $P^+$.

assume its new equilibrium value. Since $A^+ = A^+(V_o^+, P^+)$, lag equations of the form (suggested by Launder and Jones [4])

$$\frac{dV^+_{o,eff}}{dx^+} = -\frac{(V^+_{o,eff} - V^+_{o,eq})}{C} \tag{2.25}$$

are solved to simulate the effect. The term $V^+_{o,eq}$ is the local blowing parameter, and $V^+_{o,eff}$ is its effective value, used to compute the damping constant. A similar equation is solved for $P^+$. The recommended value for C is 4000.

In the boundary layer momentum equation (2.7), the body force term, X, must exert some influence upon the viscous sublayer thickness. In the program it is assumed that the influence of X upon the damping coefficient is similar to the pressure gradient. Thus a non-dimensional body force, $X^+$, is computed, and the algebraic sum $(P^+ - X^+)$ is used in place of $P^+$ to evaluate an equation of the form of equation (2.25) for $P^+_{eff}$.

The outer region of the flow, referred to as the wake region, is modeled using a mixing-length directly proportional to the boundary layer thickness. The program input variable FR determines the thickness as $\delta_{(1.00-FR)}$, with a recommended value of 0.01 for FR.

$$\ell = \lambda \delta_{.99} \quad . \tag{2.26}$$

A recommended value of $\lambda$ is 0.085. The outer region is defined as $y > \lambda \delta_{.99}/\kappa$.

There is some evidence that the effective value of $\lambda$ is larger than 0.085 for boundary layers in which the momentum thickness Reynolds number is less than 5500. This may be a result of the fact that at low Reynolds numbers the sublayer is a larger fraction of the boundary layer and the approximation of a constant mixing-length over the remainder of the boundary layer is less valid. For strong blowing, even at low Reynolds numbers, $\lambda$ again appears to be close to 0.085, and this is consistent with the above explanation because the sublayer is then thinner. The following equation has been found to describe the observed low Reynolds behavior of $\lambda$ quite well.

$$\lambda = 2.942 \, \lambda_o \, Re_M^{-1/8} \, (1.0 - 67.5 \, F) \quad , \tag{2.27}$$

where $F = \rho_o V_o / \rho_\infty U_\infty$ and $\lambda_o$ is the program input value. If $\lambda$ becomes less than $\lambda_o$, it is set equal to $\lambda_o$.

### 2.3.2 <u>Turbulent Kinetic Energy Model for</u> $\varepsilon_M$

The Prandtl mixing-length is essentially an equilibrium model that can handle turbulent flows with slowly changing boundary conditions. For strongly non-equilibrium boundary layers (especially under adverse pressure gradient conditions or when there is an appreciable amount of free-stream turbulence), a higher level of closure model for the turbulent shear stress is desirable. The turbulent kinetic energy model (TKE model) relates a velocity scale-length scale product to the eddy diffusivity for momentum,

$$\varepsilon_M = \frac{\mu_t}{\rho} = \left(\frac{A_q}{\kappa}\right) \ell \sqrt{\frac{\overline{q^2}}{2}} \quad , \tag{2.28}$$

where $q^2/2$ is the turbulent kinetic energy of the flow and $\ell$ is the mixing-length, as defined by equations (2.21) or (2.26).

Actually, the TKE model incorporated into the program is a hybrid model; the Prandtl mixing-length model for $\varepsilon_M$ is used in the near-wall viscous region and the TKE model for $y^+ > 2A^+$ or $y^+ > B^+$. In principle, the TKE model may be applied in the viscous region, but this requires modification to the length scales for production and dissipation. At present there are no provisions in the program for computing TKE in the viscous sublayer region.

Turbulent kinetic energy of a flow is computed in the program by solving a differential equation of the form

$$\rho U \frac{\partial(q^2/2)}{\partial x} + \rho V \frac{\partial(q^2/2)}{\partial y} = -\rho \overline{u'v'} \frac{\partial U}{\partial y} - \mathcal{D} + \frac{1}{r}\frac{\partial}{\partial y}(rJ_q) \quad . \qquad (2.29)$$

In the TKE equation, the production term (the first term to the right of the equal sign) is modeled from equations (2.5) and (2.28), and given by

$$-\rho \overline{u'v'} \frac{\partial U}{\partial y} = \rho\left(\frac{A_q}{\kappa}\right) \ell \sqrt{\frac{q^2}{2}} \left(\frac{\partial U}{\partial y}\right)^2 \quad . \qquad (2.30)$$

The dissipation term, $\mathcal{D}$, is modeled as

$$\mathcal{D} = \rho\,(B_q \kappa)\, \frac{\left(\sqrt{q^2/2}\right)^3}{\ell} \quad , \qquad (2.31)$$

where $\kappa$ is the von Karman constant.

$B_q$ is the dissipation constant, and it is related to $A_q$ by requiring production to equal dissipation in the logarithmic region near the wall.

$$B_q = \frac{A_q^3}{\kappa^4} \quad . \qquad (2.32)$$

For $\kappa = 0.41$, suggested values for $A_q$ and $B_q$ are 0.22 and 0.38, respectively.

The diffusion term, $J_q$, is modeled as

$$J_q = \rho\,(\nu + \varepsilon_q)\, \frac{\partial(q^2/2)}{\partial y} \quad , \qquad (2.33)$$

where $\nu$ is the laminar kinematic viscosity, and $\varepsilon_q$ is related to $\varepsilon_M$ by a turbulent Schmidt number,

$$Sc_q = \frac{\varepsilon_M}{\varepsilon_q} . \tag{2.34}$$

A suggested value for $Sc_q$ is 1.7.

Boundary conditions for equation (2.29), with a wall and a free stream, are

$$\frac{q^2}{2} = \left(\frac{\kappa}{A_q} \ell \frac{\partial U}{\partial y}\right)^2 \quad \text{at} \quad y^+ = \begin{cases} 2A^+ \\ B^+ \end{cases} \tag{2.35a}$$

and

$$\lim_{y\to\infty} \frac{q^2}{2} = \begin{pmatrix}\text{free stream} \\ \text{turbulence level}\end{pmatrix} = \frac{3}{2} T_u^2 U_\infty^2 . \tag{2.35b}$$

Equation (2.35b) assumes isotropic free-stream turbulence and $T_u = \sqrt{\overline{u'^2}} / U_\infty$.

### 2.3.3 Constant Eddy Diffusivity Model

An alternative to the assumption that mixing-length in the outer region is constant is the assumption that eddy diffusivity for momentum is constant. Eddy diffusivity in this region can be correlated to either displacement thickness or momentum thickness Reynolds number or diameter Reynolds number in the case of pipe-flow. In the program, this option is given by

$$\frac{\varepsilon_M}{\nu} = aRe_M^b . \tag{2.36}$$

In the above expression, suggested values of $a$ and $b$ for pipe-flow are 0.005 and 0.9, respectively. For pipe-flow this option is to be preferred to the constant mixing-length option.

## 2.4 Turbulent Heat Flux

Turbulent heat flux is modeled using the eddy diffusivity for heat. The program incorporates two options for modeling $\varepsilon_H$, a constant turbulent Prandtl and a variable turbulent Prandtl number.

### 2.4.1 Constant Turbulent Prandtl Number

The eddy diffusivity for heat is modeled by relating it to the eddy diffusivity for momentum,

$$Pr_t = \frac{\varepsilon_M}{\varepsilon_H} , \quad (2.12)$$

where $Pr_t$ is the turbulent Prandtl number.

A very simple physical model of the turbulent momentum and energy transfer processes leads to the conclusion that $\varepsilon_H = \varepsilon_M$, i.e., $Pr_t = 1.00$ (the "Reynolds Analogy"). Slightly more sophisticated models suggest that $Pr_t >$ 1.00 when the molecular Prandtl number is very much less than unity. A suggested value for gases is 0.90.

### 2.4.2 Variable Turbulent Prandtl Number

An improved model for $Pr_t$ is to allow it to vary with distance from the wall, as suggested from experimental data from Stanford [3]. Several conclusions can be drawn from the Stanford data. First, the turbulent Prandtl number, at least for air, apparently has an order of magnitude of unity. Thus the Reynolds Analogy ($Pr_t = 1.00$) is not a bad approximation.

The second conclusion is that $Pr_t$ seems to go to a value higher than unity very near the wall, but is evidently less than unity in the wake or outer region. The situation very close to the wall is especially vexing because it is extremely difficult to make accurate measurements in this region, and yet it seems evident that something interesting and important is happening in the range of $y^+$ from 10.0 to 15.0. The behavior of $Pr_t$ at values of $y^+$ less than about 10.0 is highly uncertain but fortunately not very important, because molecular conduction is the predominant transfer mechanism in this region. At the other extreme, in the wake region $Pr_t$ does not need to be known precisely, because the heat flux tends to be small there.

Another conclusion, for which the evidence is not yet very strong, is that there is some small effect of pressure gradient on $Pr_t$. Data suggest that an adverse pressure gradient tends to decrease $Pr_t$, and there seems a tendency for $Pr_t$ to be increased by a favorable pressure gradient (an accelerating flow). Transpiration, apparently, does not influence $Pr_t$ unless there is an effect very close to the wall that is hidden in the experimental uncertainty in this region.

Incorporated into the program to predict the general behavior of turbulent Prandtl number for gases, as well as low and high laminar Prandtl number fluids, is a conduction model for $Pr_t$. The model simulates the idea that an "eddy" exchanges energy both in transit in the vertical direction and while equilibrating with the surrounding fluid at the end of its travel. From analytical considerations, the model is expressed by

$$Pr_t = \left[\frac{\alpha^2}{2} + \alpha c Pe_t - (cPe_t)^2 \left(1.0 - \exp[-\alpha/cPe_t]\right)\right]^{-1} . \qquad (2.37)$$

In the above equation, $Pe_t$ is the turbulent Peclet number, $(\varepsilon_M/\nu)Pr$, and $\alpha = \sqrt{1/PRT}$, where PRT is the asymptotic value of $Pr_t$ for large $y^+$, in the wake region. The programmed value for c is 0.2, and the suggested value for PRT is 0.86. Equation (2.37) is plotted in Figure 2.4 for three values of Pr using these constants.

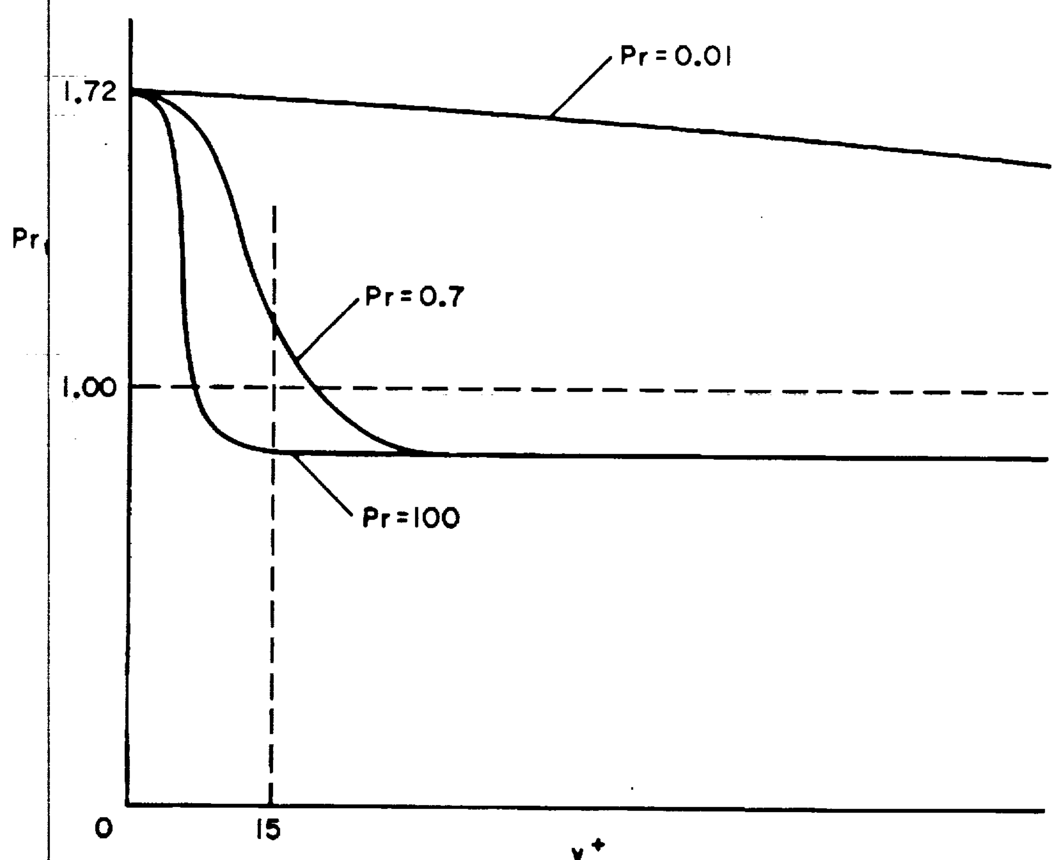



Figure 2.4. Variation of turbulent Prandtl number with Pr.

## 2.5 Laminar-Turbulent Transition

In laminar boundary layers, disturbances to the flow will either die out or grow; if the disturbances continue to grow, there will be a region downstream where transition occurs, beyond which fully turbulent flow will eventually be established. The onset of transition depends to a large extent upon whether the prevailing boundary conditions have a stabilizing or a destabilizing effect on the flow. Smooth surfaces and favorable pressure gradients (acceleration) can cause the former, and rough surfaces, adverse pressure gradients, and free-stream turbulence can cause the latter effect.

For two-dimensional boundary layer flows over a smooth surface, with a constant free stream velocity, and with moderate free-stream turbulence, the onset of transition is usually considered to be related to a critical momentum thickness Reynolds number, $Re_{tran}$. This is analogous to flow in a pipe where $Re_{tran} \simeq 2300$. Once transition commences, it will continue until the flow becomes completely turbulent.

Transition is modeled in the program by flagging the program to commence computation of turbulent shear stress and heat flux when the flow momentum thickness Reynolds number, $Re_M$, exceeds $Re_{tran}$. To effect a gradual transition, the local value of $A^+$ is modified according to the empirical equation

$$A^+ = A^+ + (300.0 - A^+) \times \left\{1.0 - \sin\left(\frac{1.57}{Re_{tran}}\left[Re_M - Re_{tran}\right]\right)\right\}^2 , \quad (2.38)$$

for the region in the downstream flow direction where $Re_M \leq 2Re_{tran}$. This equation has the effect of smoothly increasing the turbulent viscosity in the near-wall region. A suggested value for $Re_{tran}$ is 200. Transition with $B^+$ is handled in a similar manner.

## Chapter 3

## FLOW NEAR A WALL

### 3.1 Computation in the Near-Wall Region

Computation of a flow field involves solving the finite-difference equations at discrete nodes in the cross-stream direction. The nodal spacing, or grid, can be coarse if velocity and enthalpy profiles are slowly changing between nodes. For a turbulent flow, large gradients in velocity exist with the near-wall region requiring a fine nodal spacing. It is customary in most finite-difference turbulent calculations to have at least as many nodal points in the near-wall region (say the inner 20 per cent of the boundary layer) as are used in the remaining coarse part of the grid.

In computing near-wall flows in this program, the Couette flow form of the boundary layer equations are solved between the wall and a point near the wall, the join point. At the join point the Couette flow solutions are matched to the finite-difference solutions, in terms of velocity and shear stress, and enthalpy and heat flux, and the resulting unknowns, wall shear stress and wall heat flux, are thus determined.

In dealing with flow in the near-wall region, the program has two options. The first option is to "use the Wall Function." Here the Couette flow equations are numerically integrated over the region of high velocity gradient. A major advantage of this option is that it greatly reduces the required number of finite-difference nodes. Using the Wall Function is especially advantageous when computing high Reynolds number flows.

The second option in computing flow near a wall is to "bypass the Wall Function." Here the finite-difference mesh is carried down to the wall with a progressively finer spacing. Bypassing the Wall Function is recommended for large pressure gradients when the Couette flow approximation begins to lose its validity.

### 3.2 The Couette Flow Equations

In the near-wall region both velocity and stagnation enthalpy profiles can have large gradients in the cross-stream direction, but their streamwise gradients are usually small. By neglecting these streamwise gradients, the convectiv transport equations are simplified to ordinary differential equations, and the integrated form of these equations is the Couette flow equations.

To develop the Couette flow equations, the boundary layer equations will be recast in terms of shear stress and heat flux using

$$\tau = (\mu + \mu_t) \frac{\partial U}{\partial y} = \mu_{eff} \frac{\partial U}{\partial y} , \qquad (3.1)$$

and

$$\dot{q}'' = -\left[\frac{k}{c} + \left(\frac{k}{c}\right)_t\right] \frac{\partial I}{\partial y} = -\frac{\mu_{eff}}{Pr_{eff}} \frac{\partial}{\partial y}\left[I^* - \frac{U^2}{2g_c J}\right] . \qquad (3.2)$$

These definitions are substituted into the momentum equation (2.7) and stagnation enthalpy equation (2.15), and they are re-written, along with the continuity equation (2.1), for plane flow (no-radius effect included).

$$\frac{\partial(\rho U)}{\partial x} + \frac{\partial(\rho V)}{\partial y} = 0 , \qquad (3.3a)$$

$$\rho U \frac{\partial U}{\partial x} + \rho V \frac{\partial U}{\partial y} = g_c\left(- \frac{dP}{dx} + \frac{\partial \tau}{\partial y} + X\right) , \qquad (3.3b)$$

$$\rho U \frac{\partial I^*}{\partial x} + \rho V \frac{\partial I^*}{\partial y} = - \frac{\partial}{\partial y} [\dot{q}'' - U \tau] + \frac{UX}{J} + s . \qquad (3.3c)$$

These equations are non-dimensionalized using "wall coordinates". In the definitions which follow, the small zero subscript denotes a wall value.

$$U_\tau = \sqrt{g_c \tau_o / \rho_o} , \qquad (3.4a)$$

$$U^+ = U/U_\tau , \qquad (3.4b)$$

$$V_o^+ = V_o/U_\tau , \qquad (3.4c)$$

$$x^+ = xU_\tau/\nu_o , \qquad (3.4d)$$

$$y^+ = yU_\tau/\nu_o , \qquad (3.4e)$$

$$\tau^+ = \tau/\tau_o , \qquad (3.4f)$$

$$P^+ = \frac{g_c \nu_o}{\rho_o U_\tau^3} \frac{dP}{dx} \quad , \tag{3.4g}$$

$$X^+ = \frac{g_c \nu_o}{\rho_o U_\tau^3} X \quad , \tag{3.4h}$$

for the momentum equation, and, in addition,

$$I^{*+} = \frac{(I_o^* - I^*) U_\tau}{\dot{q}_o''/\rho_o} \quad , \tag{3.4i}$$

$$q^+ = \frac{\dot{q}''}{\dot{q}_o''} \quad , \tag{3.4j}$$

$$S^+ = \frac{\nu_o}{\dot{q}_o'' U_\tau} s \quad , \tag{3.4k}$$

$$W = \frac{\rho_o U_\tau^3}{g_c J \dot{q}_o''} \quad , \tag{3.4ℓ}$$

for the stagnation enthalpy equation.

Integration of equations (3.3a) and (3.3b) with respect to $y$, combining, and transforming to "wall coordinates" yields

$$\tau^+ = 1 + v_o^+ U^+ + (P^+ - X^+)\, y^+ \left[1 - \frac{1}{y}\int_o^y \left(\frac{\rho}{\rho_\infty}\right)\left(\frac{U}{U_\infty}\right)^2 dy\right] + f_x \quad , \tag{3.5}$$

where

$$f_x = \frac{\rho_\infty U_\infty}{\tau_o} \frac{dU_\infty}{dx} \left[\frac{\rho U}{\rho_\infty U_\infty} \int_o^y \left(\frac{\rho U}{\rho_\infty U_\infty}\right) dy - \int_o^y \left(\frac{\rho}{\rho_\infty}\right)\left(\frac{U}{U_\infty}\right)^2 dy\right]$$
$$+ \frac{\rho_\infty U_\infty^2}{\tau_o}\left[\frac{d}{dx}\int_o^y \left(\frac{\rho}{\rho_\infty}\right)\left(\frac{U}{U_\infty}\right)^2 dy - \frac{\rho U}{\rho_\infty U_\infty}\int_o^y \left(\frac{\rho U}{\rho_\infty U_\infty}\right) dy\right] .$$

The Couette flow form of the momentum equation used in the program is equation (3.5) with $f_x$ neglected. This form was developed by Julien et al. [5] ar retains an integral term to better approximate a departure from Couette flow when $P^+$ is large. The additional term is exact for asymptotic accelerating flows.

Integration of equations (3.3a) and (3.3c) with respect to y, combining, and transforming to "wall coordinates", yields

$$q^+ = 1 + V_o^+ I^{*+} + U^+ \tau^+ W + U^+ y^+ X^+ W + S^+ y^+ + g_x \quad , \qquad (3.6)$$

where

$$\begin{aligned} g_x = \frac{(I^* - I_\infty^*)}{\dot{q}_o''} & \left[ \frac{d}{dx} (\rho_\infty U_\infty) \cdot \int_o^y \left( \frac{\rho U}{\rho_\infty U_\infty} \right) dy \right. \\ & \left. + \rho_\infty U_\infty \frac{d}{dx} \int_o^y \left( \frac{\rho U}{\rho_\infty U_\infty} \right) dy \right] \\ & - \frac{1}{\dot{q}_o''} \frac{d}{dx} \left[ \rho_\infty U_\infty (I_o^* - I_\infty^*) \right] \int_o^y \frac{\rho U}{\rho_\infty U_\infty} \left( \frac{I^* - I_\infty^*}{I_o^* - I_\infty^*} \right) dy \\ & - \frac{\rho_\infty U_\infty (I_o^* - I_\infty^*)}{\dot{q}_o''} \frac{d}{dx} \int_o^y \frac{\rho U}{\rho_\infty U_\infty} \left( \frac{I^* - I_\infty^*}{I_o^* - I_\infty^*} \right) dy \quad . \end{aligned}$$

The Couette flow form of the stagnation enthalpy equation used in the program is equation (3.6) with $g_x$ neglected.

### 3.3 Using the Wall Function

In the previous section it was seen that the Couette flow equations are merely first integrals of the Couette flow form of the boundary layer equations, and they relate wall shear stress and wall heat flux to shear stress and heat flux at some point away from the wall. By replacing the shear stress and heat flux with their constitutive equations, the Couette flow equations become first-order ordinary differential equations describing the variation in velocity and stagnation enthalpy across the Couette layer adjacent to the wall. These equations are then numerically integrated across the layer and matched to the finite-difference solutions for velocity and stagnation enthalpy, resulting in explicit expressions for the wall shear stress and heat flux. The match-up point is located midway between the second and third finite-difference nodes from the wall and is referred to as the join point, or 2.5 point.

### 3.3.1 Momentum Equation

The constitutive equation (3.1) for shear stress is rewritten in terms of "wall coordinates" as

$$\tau^+ = \mu^+ \frac{\partial U^+}{\partial y^+} , \tag{3.7}$$

where $\mu^+ = (\mu + \mu_t)/\mu_o$.

From Section 3.2, the Couette flow equation for momentum is

$$\tau^+ = 1 + V_o^+ U^+ + (P^+ - X^+)\ y^+ \left[1 - y \int_o^y \left(\frac{\rho}{\rho_\infty}\right)\left(\frac{U}{U_\infty}\right)^2 dy\right]. \tag{3.8}$$

An ordinary differential equation describing momentum transport across the Couette layer is obtained by equating (3.7) and (3.8), along with using the mixing-length hypothesis to model $\mu^+$.

$$\frac{dU^+}{dy^+} = \frac{2\tau^+ \left(\frac{\mu_o}{\mu}\right)}{1 + \left[1 + 4\kappa^2 y^{+2} D^2 \tau^+ \left(\frac{\rho}{\rho_o}\right)\left(\frac{\mu_o}{\mu}\right)^2\right]^{1/2}} . \tag{3.9}$$

In the program the above equation is numerically integrated, using equation (3.8) for $\tau^+$, and equation (2.22 or 2.23) for D, from the wall outward to the join point.

The join point, or match-up point, is located at $y_{2.5}$, which is the arithmetic average of $y_2$ and $y_3$, locating nodal points 2 and 3. The required value of U at the join point is $U_{2.5}$, the arithmetic average of $U_2$ and $U_3$, as computed from the finite-difference solution.

Since the integration of equation (3.9) is in "wall coordinates", the upper limit to the integral needs to be in "wall coordinates". It is not yet possible to convert $U_{2.5}$ and $y_{2.5}$ to $U^+_{2.5}$ and $y^+_{2.5}$ because $\tau_o$ is still an unknown. However, a join-point Reynolds number can be formed which relates the "physical coordinates" to the "wall coordinates",

$$Re_{2.5} = \frac{U_{2.5}\ y_{2.5}}{\nu_o} = (U^+y^+)_{2.5} \tag{3.10}$$

As $U^+ = U^+(y^+)$ is evaluated from integration of equation (3.9), the $U^+y^+$ product is computed and compared to $Re_{2.5}$ . Integration is terminated when the $U^+y^+$ product equals $Re_{2.5}$. With the join-point values of $U^+$ and $y^+$ now known, the wall shear stress and friction factor are computed from $U_{2.5}$ and the definition of $U^+$ ,

$$\tau_o = \frac{\rho_o\ U_{2.5}^2}{g_c(U_{2.5}^+)^2} \tag{3.11a}$$

and

$$C_f/2 = \frac{g_c\tau_o}{\rho_\infty U_\infty^2} \tag{3.11b}$$

### 3.3.2 Stagnation Enthalpy Equation

The constitutive equation (3.2) for heat flux is rewritten in terms of wall coordinates as

$$q^+ = \frac{\mu^+}{Pr_{eff}}\ \frac{\partial I^{*+}}{\partial y^+} + W\frac{\mu^+}{Pr_{eff}}\ \frac{\partial}{\partial y}\left(\frac{U^{+2}}{2}\right) \tag{3.12}$$

From Section 3.2, the Couette flow equation for stagnation enthalpy is

$$\begin{aligned} q^+ = 1 + V_o^+\ I^{*+} + U^+\tau^+W \\ + U^+y^+X^+W + S^+y^+ \end{aligned} \tag{3.13}$$

An ordinary differential equation describing enthalpy transport across the Couette layer is obtained by equating (3.12) with (3.13),

$$\frac{dI^{*+}}{dy^+} = \frac{Pr_{eff}}{\mu^+}\ (1 + V_o^+\ I^{*+}) + (Pr_{eff}-1)W\frac{d}{dy^+}\left(\frac{U^{+2}}{2}\right) + \frac{Pr_{eff}}{\mu^+}(U^+y^+X^+W + S^+y^+)$$

(3.14)

In the program equation 3.14 is numerically integrated in the same loop as equation (3.9) for $U^+$ .

If the stagnation enthalpy boundary condition is a level type, i.e., $I^*(x,0) = I_o^*(x)$, then wall heat flux and Stanton number are computed from $I_{2.5}^*$ , the arithmetic average of $I_2^*$ and $I_3^*$ , and the definition of $I^{*+}$ ,

$$\dot{q}_o'' = \frac{\rho_o U_{2.5}}{U_{2.5}^+ I_{2.5}^{*+}} (I_o^* - I_{2.5}^*) \tag{3.15a}$$

and

$$St = \frac{\dot{q}_o''}{\rho_\infty U_\infty (I_o^* - I_\infty^*)} \tag{3.15b}$$

If the stagnation enthalpy boundary condition is a flux type, then the wall enthalpy and heat flux are linked through the total energy flux boundary condition (see Figure 2.2).

$$\dot{E}_{total}(x) = \dot{m}_o'' I_o^* + \dot{q}_o'' \tag{3.16}$$

For flux-type boundary conditions, equations (3.15a) and (3.16) are solved algebraically for $I_o^*$ and $\dot{q}_o''$ . The Stanton number is then formulated from equation (3.15b). Note that the Stanton number evaluated in the program, equation (3.15b), is based on <u>stagnation</u> enthalpy difference, and <u>not</u> recovery enthalpy difference. The latter would require knowledge of a "recover factor" which has no real significance or usefulness in the general problem, i.e., for other than constant free-stream velocity flows.

### 3.4 Bypassing the Wall Function

The second user option is to "bypass the Wall Function", implying the join point is in close proximity to the wall where laminar-like flow exists. For turbulent flows, this implies a join-point value $y^+$ of less than, say, 2.0. In this region the viscosity ratio $(\mu + \mu_T)/\mu_o$ is unity, and the Couette flow equations can be integrated in closed form. Match-up with the finite-difference solutions for velocity and stagnation enthalpy is similar to the procedure involved in "using the Wall Function".

### 3.4.1 Momentum Equation

To obtain an expression for $U^+$ at the edge of the Couette layer, the constitutive equation (3.7) for the shear stress is equated to the Couette flow equation for momentum (3.8) and integrated (with $\mu^+ = 1$) .

$$U^+ = y^+ + (V_o^+ + P^+ - X^+)\left[\frac{\exp(V_o^+ y^+) - 1. - V_o^+ y^+}{(V_o^+)^2}\right] \tag{3.17}$$

Recall that while $U^+$ and $y^+$ are unknown, their product is the join-point Reynolds number (see Section 3.3.1).

$$Re_{2.5} = \frac{U_{2.5}\, y_{2.5}}{\nu_o} = (U^+ y^+)_{2.5} \tag{3.10}$$

In the program, the solution to equation (3.17) is obtained by linearizing and solving in three successive steps:

$$y_{2.5}^+ = (Re_{2.5})^{1/2} \tag{3.18a}$$

$$y_{2.5}^+ = \left[\frac{Re_{2.5}}{1 + \frac{(P^+ - X^+)y_{2.5}^+}{2} + \frac{V_o^+ y_{2.5}^+}{2}}\right]^{1/2} \tag{3.18b}$$

$$y_{2.5}^+ = \left[\frac{Re_{2.5}}{1 + \frac{(P^+ - X^+)y_{2.5}^+}{2} + \frac{V_o^+ y_{2.5}^+}{2}}\right]^{1/2} \tag{3.18c}$$

After solving for $y_{2.5}^+$ , the value of $U_{2.5}^+$ is obtained from equation (3.10). The shear stress and friction factor are obtained from equations (3.11a-b).

### 3.4.2 Stagnation Enthalpy Equation

An expression for $I^{*+}$ at the edge of the Couette layer is obtained by integrating equation (3.14), which relates the constitutive equation for heat flux to the Couette flow equation for stagnation enthalpy. In the integration, the viscous dissipation, work against body forces, and energy source terms are neglected. The resulting expression for $I^{*+}$, with $\mu^+$ equal to unity and $Pr_{eff}$ equal to $Pr$, is

$$I^{*+} = \frac{\exp[PrV_o^+y^+] - 1}{V_o^+} \tag{3.19}$$

In the program, equation (3.19) is approximated by

$$I^{*+}_{2.5} = Pr\left(\frac{V_o^+ y_{2.5}^{+2}}{2} + y^+_{2.5}\right) \tag{3.20}$$

After solving for $I^*_{2.5}$, the wall heat flux and Stanton number are obtained as described at the end of Section 3.3.2 .

## 3.5 Routine LAMSUB

As indicated in the previous sections, the Couette flow equations are solved from the wall out to the join point where $y^+ = y^+_{2.5}$ . The main function of the LAMSUB routine is to assure the condition

$$\text{YPMIN} \leq y^+_{2.5} \leq \text{YPMAX} \tag{3.21}$$

where YPMIN and YPMAX are program input variables.

When "bypassing the Wall Function", YPMIN must be zero, and YPMAX should be less than two (unity is recommended). This will give a join-point Reynolds number of less than four, thus assuring the assumption that turbulent viscosity can be neglected in the Couette flow equations.

When "using the Wall Function" typical values for YPMIN and YPMAX are 20 and 40, respectively. These values bracket the upper limits of the integrals, and assure that the Couette flow equations are not applied outside their region of applicability. For a flat plate boundary layer, the upper limit might be 50 to 100, and for high Reynolds number flows, the upper limit might extend out to between 100 and 200. For boundary layer flows with strong pressure gradient, the limit of applicability can drop to near 15 -- thus the reason for the Wall Function bypass option.

Routine LAMSUB controls the join point value as follows: if $y^+_{2.5}$ drops below YPMIN, the routine removes the stream tube located at $y_3$, and if $y^+_{2.5}$ becomes larger than YPMAX, the routine inserts a new stream tube midway between $y_{2.5}$ and $y_3$. In both cases, after the grid has been readjusted, the wall function is again solved and the new $y^+_{2.5}$ is compared using equation (3.21).

## 3.6 Integral Parameters

At each integration step, when one surface is a wall, the velocity profile displacement and momentum thicknesses, $\delta_1$ and $\delta_2$, are calculated along with the enthalpy thickness, $\Delta_2$, for the stagnation enthalpy profile. These thicknesses are defined as follows:

$$\delta_1 = \int_o^\delta \left(1 - \frac{\rho U}{\rho_\infty U_\infty}\right) \frac{r}{r_o}\, dy \quad , \tag{3.22a}$$

$$\delta_2 = \int_o^\delta \frac{\rho U}{\rho_\infty U_\infty} \left(1 - \frac{U}{U_\infty}\right) \frac{r}{r_o}\, dy \quad , \tag{3.22b}$$

$$\Delta_2 = \int_o^\delta \frac{\rho U}{\rho_\infty U} \left(\frac{I^* - I^*_\infty}{I^*_o - I^*_\infty}\right) \frac{r}{r_o}\, dy \quad , \tag{3.22c}$$

where $r_o$ is the wall radius. Integration is carried out in the program using a trapezoidal rule.

In the program the boundary layer equations can be solved with or without consideration of transverse radius of curvature. Generally, transverse curvature effects are important for thick axisymmetric boundary layers. If these curvature effects are considered, then $\delta_1$ and $\delta_2$ are modified by solving the equations

$$\delta_{1,axi}\left(1 \pm \frac{\delta_{1,axi} \cos \alpha}{2r_o}\right) = \delta_1 \quad , \tag{3.23a}$$

$$\delta_{2,axi}\left(1 \pm \frac{\delta_{2,axi}\cos\alpha}{2r_o}\right) = \delta_2 \tag{3.23b}$$

for $\delta_{1,axi}$ and $\delta_{2,axi}$ after calculating $\delta_1$ and $\delta_2$ using equation (3.22). Figure 2.1 shows $\alpha$ and its relation to the wall radius. The proper sign choice is (+) for external flow over a body of revolution and (-) for flow inside a body of revolution (due to the coordinate system used in the program).

## 3.7 Pipe and Channel Flows

If the flow is a confined flow, a friction factor, Stanton number, and Nusselt number are computed using the following definitions.

$$\frac{C_f}{2} = \frac{g_c \tau_o}{\bar{\rho}\,\bar{U}^2} , \tag{3.24}$$

$$St = \frac{q''_o}{\bar{\rho}\,\bar{U}(I^*_o - \bar{I}^*)} , \tag{3.25}$$

$$Nu = St \cdot \overline{Pr} \cdot Re , \tag{3.26}$$

where the bar quantities are mean quantities.

The mean stagnation enthalpy is defined by

$$\bar{I}^* = \frac{\int_o^{r_w} \rho U I^* r dy}{\int_o^{r_w} \rho U r dy} . \tag{3.27}$$

The mean velocity is defined by

$$\bar{U} = \frac{2\pi \int_o^{r_w} \rho U r dy}{\bar{\rho}\pi r_w^2} = \frac{2\left(\begin{matrix}\text{mass flow}\\ \text{rate/radian}\end{matrix}\right)}{\bar{\rho} r_w^2} , \tag{3.28}$$

and the Reynolds number is defined as

$$\mathrm{Re} = \frac{\overline{\rho U D}}{\bar{\mu}} = \frac{4\begin{pmatrix}\text{mass flow}\\ \text{rate/radian}\end{pmatrix}}{\bar{\mu} r_w} \quad . \tag{3.29}$$

The mean values for density, viscosity and Prandtl number are those values at the $y$ location where $I^* = \bar{I}^*$.

## Chapter 4

## METHOD OF SOLUTION

### 4.1 Transformation of the Equations

The continuity, momentum, and stagnation enthalpy equations were developed in Chapter 2. The first step in transformation is to recast the convective transport equations into stream function coordinates using the von Mises transformation. In essence, the y-coordinate is replaced by a coordinate that is constant along streamlines, namely, the stream function $\psi$. The new independent variables become $x$ and $\psi$, and the $U$ velocity component is defined by

$$U = \frac{1}{r\rho}\frac{\partial\psi}{\partial y} \quad . \tag{4.1}$$

In stream function coordinates the momentum equation (2.7) and the stagnation enthalpy equation (2.15) become

$$\rho U \frac{\partial U}{\partial x} + \rho U \frac{\partial}{\partial\psi}\left[r^2 \rho U \mu_{eff} \frac{\partial U}{\partial\psi}\right] = -g_c \frac{dP}{dx} + g_c X \quad , \tag{4.2}$$

$$\rho U \frac{\partial I^*}{\partial x} + \rho U \frac{\partial}{\partial\psi}\left[r^2 \rho U \frac{\mu_{eff}}{Pr_{eff}} \frac{\partial I^*}{\partial\psi}\right] = \frac{\partial}{\partial\psi}\left[\frac{\mu_{eff}}{g_c J}\left(1 - \frac{1}{Pr_{eff}}\right) r^2 \rho U \frac{\partial}{\partial\psi}\left(\frac{U^2}{2}\right)\right] + S \quad . \tag{4.3}$$

Note that in the transformation the $V$ component of velocity disappears and the continuity equation is no longer used explicitly, due to the definition of the stream function.

In the stream function coordinate system, the boundary layer fluid flows between two surfaces, I and E. The I-surface originates at $y = 0$, and the E-surface forms the second bounding surface. Sign convention for a positive $y$ displacement is <u>always</u> from the I to E surface. Fluid crossing the I surface is $\dot{m}''_I$; this flow might be due to wall transpiration. Fluid crossing the E surface is $\dot{m}''_E$; this flow might be due to entrainment. The bounding solid surface is described by $\alpha$, related to the rate of change of surface curvature in the x-direction, and $r_I$, which describes the transverse curvature of the I-surface. Location of the E-surface, $r_E$, is related to $r_I$ and $\alpha$. Figure 4.1 shows the coordinate system.

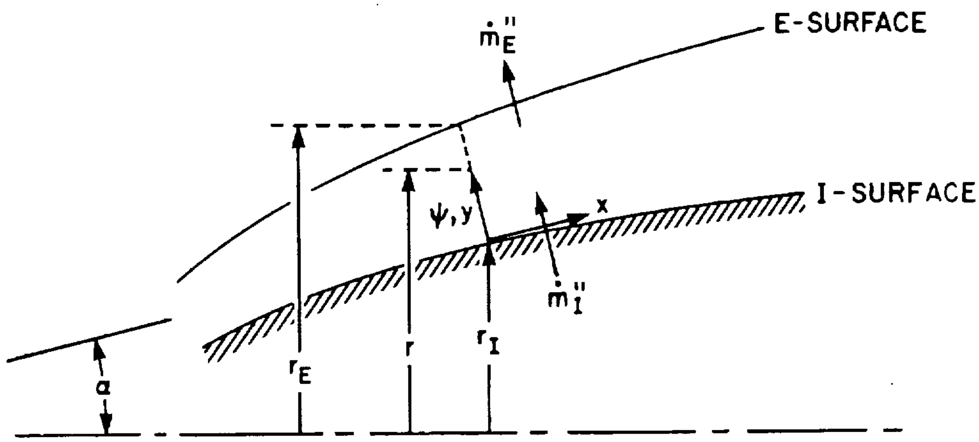



Figure 4.1. The stream-function coordinate system.

The sketch in Figure 4.1 depicts an external boundary layer over either a flat or conical surface, with I being a wall and E being a free stream. In the program, there is a limited freedom in defining these bounding surfaces. This will be discussed more thoroughly in Chapter 5.

The second and final step in the transformation is to recast equations (4.2) and (4.3) into the Patankar-Spalding coordinate system using the transformation

$$\omega = \frac{\psi - \psi_I}{\psi_E - \psi_I} \quad , \tag{4.4}$$

where $\psi_E$ and $\psi_I$ are the stream function values on the bounding surfaces.

In this non-dimensional stream function coordinate system, the momentum and stagnation enthalpy equations become

$$\frac{\partial U}{\partial x} + \left[\frac{r_I \dot{m}_I'' + \omega(r_E \dot{m}_E'' - r_I \dot{m}_I'')}{(\psi_E - \psi_I)}\right] \frac{\partial U}{\partial \omega} - \frac{\partial}{\partial \omega}\left[\frac{r^2 \rho U \mu_{eff}}{(\psi_E - \psi_I)^2} \frac{\partial U}{\partial \omega}\right] = \frac{g_c}{\rho U}\left[-\frac{dP}{dx} + X\right] \quad . \tag{4.5}$$

$$\frac{\partial I^*}{\partial x} + \left[\frac{r_I\dot{m}_I'' + \omega(r_E\dot{m}_E''-r_I\dot{m}_I'')}{(\psi_E-\psi_I)}\frac{\partial I^*}{\partial \omega}\right] - \frac{\partial}{\partial \omega}\left[\frac{r^2\rho U\mu_{eff}}{(\psi_E-\psi_I)^2\, Pr_{eff}}\frac{\partial I^*}{\partial \omega}\right]$$

$$= \frac{\partial}{\partial \omega}\left[\frac{r^2\rho U}{(\psi_E-\psi_I)^2}\mu_{eff}\left(1-\frac{1}{Pr_{eff}}\right)\frac{\partial}{\partial \omega}\left(\frac{U^2}{2}\right)\right] + \frac{S}{\rho U} \quad . \tag{4.6}$$

The transformed equations have the general form of a diffusion equation:

$$\frac{\partial \phi}{\partial x} + (a+b\omega)\frac{\partial \phi}{\partial \omega} - \frac{\partial}{\partial \omega}\left(c\,\frac{\mu_{eff}}{Pr_{eff}}\frac{\partial \phi}{\partial \omega}\right) = d \quad , \tag{4.7}$$

where a, b, c, d are constants.

In the program, equation (4.7) becomes the velocity equation when $Pr_{eff}$ is set equal to unity.

## 4.2 Finite-Difference Equations

As indicated in Chapter 1, the original basic program from which STAN5 has evolved is the Patankar/Spalding program, described in their 1967 book [1]. Only the numerics of the finite-difference equations and the concept of a wall function have been carried over into STAN5. It is our intent in this section to point out several facts regarding the finite-differencing scheme. These equations are well documented in Patankar and Spalding [1,2], and, for a revised version of the program, by Spalding [6].

The central theme in obtaining the finite-difference equations, hereafter referred to as FDE's, is twofold: (1) to form a miniature integral equation over a finite-control volume; and (2) to presume a linear variation of the dependent variable over the control volume to effect the integration. Figure 4.2 shows node locations and a control volume for three adjacent nodes at an upstream and a downstream station.

The first term in equation (4.7) is transformed into an FDE term, as follows:

$$\frac{\partial \phi}{\partial x} \simeq \frac{1}{\delta x \delta \omega}\int_{i-\frac{1}{2}}^{i+\frac{1}{2}}\int_{x_u}^{x_d}\left(\frac{\partial \phi}{\partial x}\right)dx\,d\omega \simeq \frac{1}{\delta x\delta \omega}\left[\int_{i-\frac{1}{2}}^{i}\left(\phi_{x_d}-\phi_{x_u}\right)d\omega + \int_{i}^{i+\frac{1}{2}}\left(\phi_{x_d}-\phi_{x_u}\right)d\omega\right]$$

$$\simeq \frac{1}{\delta x\delta \omega}\left[\left(\frac{1}{4}\phi_{i-1}+\frac{3}{4}\phi_i\right)\frac{1}{2}(\omega_i-\omega_{i-1}) + \left(\frac{3}{4}\phi_i+\frac{1}{4}\phi_{i+1}\right)\frac{1}{2}(\omega_{i+1}-\omega_i)\right]_{x_u}^{x_d} . \tag{4.8}$$

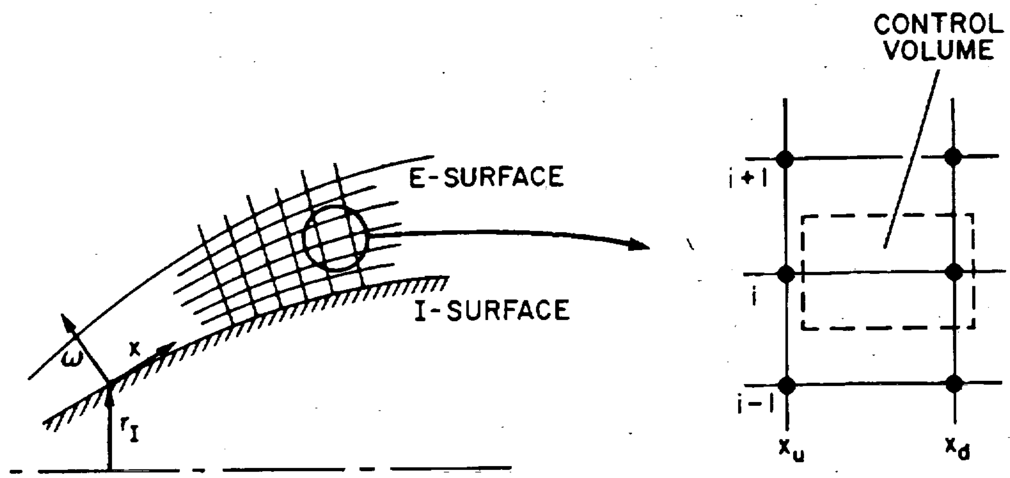



Figure 4.2. Typical nodal locations and control volume for finite-difference equations.

The second term in equation (4.7) is transformed into an FDE term using integration by parts:

$$(a+b\omega)\frac{\partial\phi}{\partial\omega} \simeq \frac{1}{\delta x\delta\omega}\int_{x_u}^{x_d}\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} (a+b\omega)\frac{\partial\phi}{\partial\omega}\, d\omega dx$$

$$\simeq \frac{1}{\delta\omega}\left[(a+b\omega)_{x_{u,i+\frac{1}{2}}} \cdot \phi_{x_{d,i+\frac{1}{2}}} - (a+b\omega)_{x_{u,i-\frac{1}{2}}} \cdot \phi_{x_{d,i-\frac{1}{2}}} - b\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \phi_{x_d}\, d\omega\right] . \quad (4.9)$$

In the above equation, the integral is evaluated in a like manner to equation (4.8). Several assumptions are built into equation (4.9): (1) the integrand of the integral is evaluated only at $x_d$; (2) the equation is "linearized" in that $(a+b\omega)$ is evaluated at $x_u$; and (3) the integrand is presumed to vary linearly with $\omega$ over the control volume. Assumption (3) implies small cross-stream convection; this was later changed by Patankar and Spalding [2] using a "high lateral flux modification", or "upwind-differencing" to more properly account for high lateral convection. The modification is not used in STAN5.

The third term in equation (4.7) is transformed into an FDE as follows:

$$\frac{\partial}{\partial\omega}\left(c\,\frac{\partial\phi}{\partial\omega}\right) \simeq \frac{1}{\delta x\delta\omega}\int_{x_u}^{x_d}\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \frac{\partial}{\partial\omega}\left(c\,\frac{\partial\phi}{\partial\omega}\right) d\omega dx$$

$$\simeq \frac{1}{\delta\omega}\left[(c)_{x_{u,i+\frac{1}{2}}}\ \frac{(\phi_{i+1}-\phi_i)_{x_d}}{(\omega_{i+1}-\omega_i)} - (c)_{x_{u,i-\frac{1}{2}}}\ \frac{(\phi_i-\phi_{i-1})_{x_d}}{(\omega_i-\omega_{i-1})}\right] . \tag{4.10}$$

The above equation is "linearized" in that $c$ is evaluated at $x_u$.

The fourth and final term in equation (4.7) is the source term. It is transformed into an FDE term as follows:

$$d \simeq \frac{1}{\delta x\delta\omega}\int_{x_u}^{x_d}\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} (d)\,d\omega dx$$

$$\simeq \frac{1}{\delta x\delta\omega}\int_{x_u}^{x_d}\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \left[(d)_{x_u} + \left(\frac{\partial d}{\partial\phi}\right)_{x_u} (\phi_d-\phi_u)\right] d\omega dx \quad . \tag{4.11}$$

In STAN5, the velocity source term is handled precisely as described by Patankar and Spalding [1]. Sources for stagnation enthalpy and turbulent kinetic energy are evaluated at $x_u$; the downstream contribution is neglected.

The FDE terms described by equations (4.8) to (4.11) are assembled into a form

$$\phi_{x_{d,i}} = A\phi_{x_{d,i+1}} + B\phi_{x_{d,i-1}} + C \quad , \tag{4.12}$$

where A, B, and C are coefficients evaluated at the upstream station, $x_u$. A set of $\phi$ equations is written for each dependent variable. In the text which follows, the velocity dependent variable is designated as U, and all other dependent variables are designated as $\phi$-equation variables.

### 4.3 Grid and Slip Scheme

A sketch of the finite-difference grid and nodal locations was previously given in Figure 4.2. Cross-stream grid lines in that sketch divide the region between the I-surface and the E-surface into non-dimensional stream tubes, or flow tubes (from consideration of the definition of $\omega$). The number of flow tubes that comprise the cross-stream grid is denoted by N. Two additional stream tubes (to define slip points) are inserted by the program near the I and E surfaces, making a total of N + 2 tubes, and thus N + 3 nodal points. A cross-stream grid is shown in Figure 4.3.

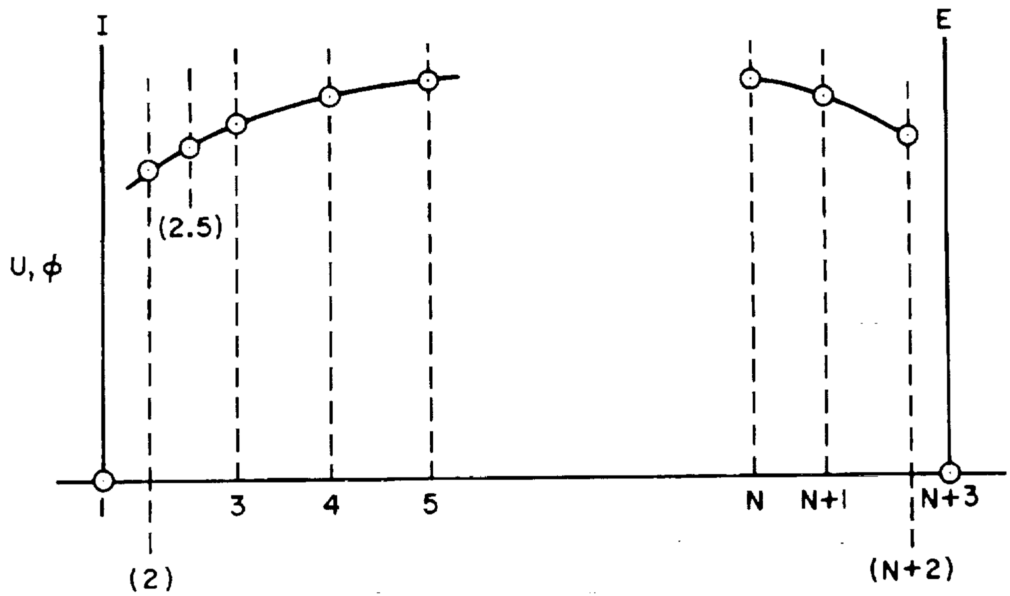



Figure 4.3. Cross-stream grid between the I and E surfaces.

In the above sketch, the 2.5 point on the grid is the join point, discussed in Section 3.3.1; the (2) point and the (N + 2) point are the slip points. Finite-difference equations of the form of equation (4.12) are solved for all nodes (2) through (N + 2). Boundary conditions for these equations are obtained through wall-function calculations, described in Chapter 3, if one surface is a wall.

The grid is established from the initial velocity profile, $U = U(y)$. The profile is integrated using equation (4.1) to obtain $U = U(\psi)$, where flow between consecutive y locations is $\Delta\psi$, or non-dimensionally $\Delta\omega$. The $\Delta\omega$ values, which represent the fractional amount of the initial flow, remain constant throughout the calculations, unless altered by routine LAMSUB, discussed in Section 3.5. The amount of boundary layer fluid can change, but the fractional percentages in each stream tube are fixed.

The slip points, along with "using the Wall Function", were developed by Patankar and Spalding [1] to allow use of a linear profile assumption (Section 4.1) in the near-wall region, thus eliminating the need to compute across a region of high velocity gradient. The scheme is an excellent "engineering tool" in terms of computational speed while preserving accuracy.

The idea behind the slip scheme is to presume power-law profiles for velocity and other $\phi$-equations in the near-wall region.

$$U = C_1 y^{\beta} \quad , \tag{4.13a}$$

$$(\phi - \phi_1) = C_2 y^{\gamma} \quad . \tag{4.13b}$$

Each of the above equations contains two unknowns, which are obtained by matching the function and its first derivative (e.g., shear stress or heat flux) at the join point. From these two criteria come defining FDE's for the slip points.

$$U_2 = U_2(U_3, \beta) \quad , \tag{4.14a}$$

$$\phi_2 = \phi_2(\phi_1, \phi_3, \gamma) \quad . \tag{4.14b}$$

The above equations are linearized in that the upstream values of $\beta$ and $\gamma$ are used. Similar types of equations can be developed for slip values near a free stream and near a symmetry line (see Patankar and Spalding [1 or 2] for a complete description).

The procedure described above to obtain slip values near a wall was later changed by Patankar and Spalding [2] to more accurately account for convection between the wall and the join point. In STAN5, this correction was accomplished by a modification to the join-point velocity and essentially accomplishes the same goal. The correction is needed for low values of $\beta$; for $\beta > 0.9$, i.e., a linear profile in the near-wall region due to laminar flow or "bypassing the Wall Function", the power-law slip scheme is adequate.

## 4.4 Entrainment and Grid Control

Entrainment is applicable to flows in which there are free surfaces. For example, the free surface for a wall boundary layer is the location where $U$ approaches $U_\infty$, i.e., its cross-stream gradient approaches zero. The function of entrainment is to introduce new fluid into the region between the I and E surfaces, thus expanding the grid outward into "fresh" fluid and thus preserving the near zero gradient at the outer edge of the computation region. The expansion can be easily seen by recalling that to increment $(\psi_E - \psi_I)$ with a constant $\Delta\omega$ spacing causes $\Delta\psi$ to increase and thus $\Delta y$. The entrained fluid is distributed to all flow tubes.

To determine if fluid should be entrained, the dependent variable difference near the free surface is compared with its free-stream value, e.g., $(U_{N+3} - U_{N+1})/U_{N+3}$ is computed and compared to ENFRA, a program input variable. This idea is depicted in Figure 4.4. The entrainment calculation for velocity in STAN5 is

$$\dot{m}''_{E\ \text{new}} = \dot{m}''_{E\ \text{old}} + \begin{bmatrix}\text{b.l.}\\ \text{mass}\\ \text{flux}\end{bmatrix}\left[\text{ENFRA} - \frac{U_{N+3} - U_{N+1}}{U_{N+3}}\right] . \tag{4.15}$$

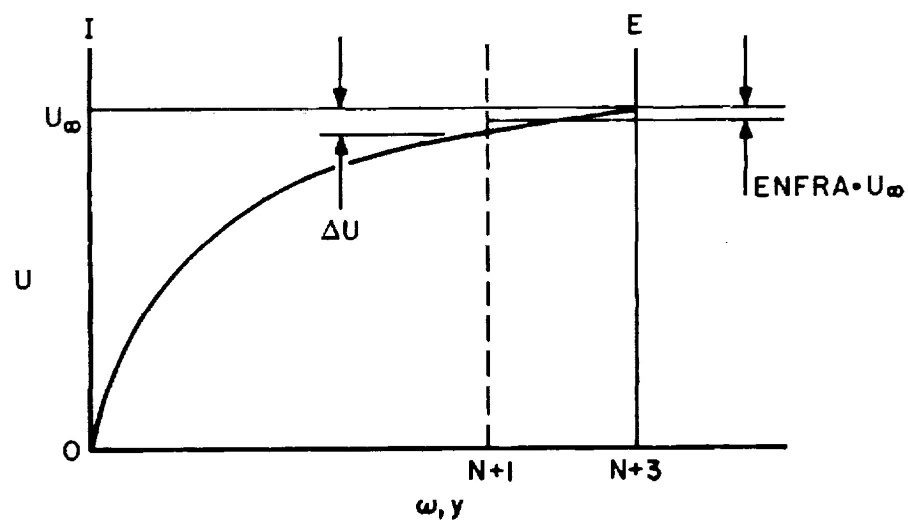



Figure 4.4. Entrainment at the free stream.

When there are $\phi$-equations being solved in addition to the momentum equation, each of these gradients near the free stream is checked to assure no defects in profiles develop. This is especially important in accelerating flows or low Prandtl number flows, where the thermal boundary layer grows outside of the momentum boundary layer. There is a flag in STAN5 that can be set to base entrainment on either the momentum equation or on the behavior of all equations. Note that in STAN5, fluid is never allowed to be detrained, due to stability considerations.

Integration stepsize, $\Delta x$, is partly determined by entrainment. The control is via an input variable FRA, say 5%, which requires that the mass flow rate into the boundary layer through the I and E surfaces be no more than FRA $\cdot$ $(\psi_E - \psi_I)$ over the distance $\Delta x$. This control, in effect, cuts back the stepsize if the boundary layer entrainment is large.

## 4.5 The Calculating Procedure

Equation (4.12) in Section 4.2 is the general form for the finite-difference equations. The equations couple all grid points in the cross-stream direction, and they are solved by a tri-diagonal matrix algorithm for $i = 2$ to $N + 2$. They have been linearized in the sense that the coefficients are calculated at the upstream stations. Thus, the program is "one step behind" in fluid properties, eddy viscosity, etc.

Because of linearization, the equations are only partially implicit, and this requires the use of a smaller $\Delta x$ stepsize than could be used by a fully implicit scheme. For heat transfer calculations this does not present much of a problem, though, because the stepsize must be small enough to follow variable boundary conditions.

# Chapter 5

# INPUT/OUTPUT

## 5.1 Introduction

To facilitate use of the program, a rather flexible input format has been developed which makes it possible to compile and link edit, and still accommodate a large number of input options merely by reading in numerical DATA. Other changes can be readily made in the core of the program, but the objective of this chapter will be to describe in detail how to access the program through DATA that are read directly by the computer.

All of the data input to the program are concentrated in the final subroutine which is labeled SUBROUTINE INPUT (KERROR). This subroutine contains a very large number of comments which in themselves constitute a set of instructions for its use. In reading this chapter it will be useful to refer to the input subroutine, and the present discussion will be based on the assumption that the reader has the input subroutine before him (her).

First it should be noted that each "read" statement is preceded by the symbols ******* extending across the page. Preceding these symbols the instructions for the "read" statement are given.

All of the "read" statements (except the title) are in the form of either a series of integer numbers or a series of decimal numbers. All of the integer numbers are in fields of five spaces. It is important to note that integers must be justified to the right side of these fields.

All decimal numbers are arranged in fields of 10 digits, and of course may be placed anywhere within that field.

## 5.2 Flow Descriptors and Controls

On the card following a title, eight integers are read, all of which convey rather fundamental information about the type of problem to be solved. Some of the program nomenclature will be introduced as these, and other variables and constants appearing below, are discussed.

GEOM is an integer, from 1 to 9, which signals in a general way the type of system geometry to be solved. GEOM = 1 is the simple boundary layer on a flat plate, but this case also applies for an axi-symmetric body so long as the boundary layer thickness is small relative to the body radius. Thus it can be used for flow in a nozzle (subsonic or supersonic), or for flow over an axi-symmetric body such as a missile, even including a stagnation region.

GEOM = 2 & 3 differ from 1 only in that radius is included in the boundary layer equations so that boundary layer thickness need not be small relative to body radius.

GEOM = 4 & 5 refer to flow in circular and flat ducts, respectively. Strictly speaking, the "boundary layer" is treated as if it filled the entire duct; however, a judicious choice of grid spacing makes it possible to handle entry-length problems with accuracy. It is also possible to solve pipes or ducts which have slightly convergent or slightly divergent walls.

GEOM = 6, 7, 8, 9 cover the cases of circular and flat jets, and free shear flows.

MODE refers to whether the flow is to be laminar or tubulent. MODE = 1 is a laminar flow, while MODE = 2 is turbulent. As will be seen below, it is possible to start with MODE = 1 and then shift to a turbulent flow on the basis of an input transition criterion.

FLUID refers to the type of fluid. FLUID = 1 is any constant-property fluid, such properties to be supplied in a later read statement. FLUID = 2 refers to air, the properties of which (based on the Keenan and Kaye Gas Tables) are provided as a separate subroutine in the program. The air properties cover temperatures from 180°R to 4620°R, but do not take into consideration dissociation at high temperatures. The program is not provided with the properties of any variable-property fluids other than air, but it is only necessary to designate some other fluid with a number (3 or higher) and then construct a subroutine similar to SUBROUTINE PROP2. The appropriate call for any other property subroutine must be inserted as indicated in the MAIN program.

NEQ refers to the total number of boundary layer equations to be solved. Thus, if the momentum equation alone is to be solved, NEQ = 1, but if momentum and energy are to be solved, NEQ = 2. Actually, the program dimensioning allows NEQ to be as high as 6, if, for example, a number of mass diffusion equations must be solved. Another related variable, NPH, will be found throughout the program. NPH = NEQ - 1, and is the number of diffusion equations (energy, mass, etc.) that must be solved. It is assumed that the momentum equation is <u>always</u> solved.

N defines the grid structure in the y-direction; it is the number of flow tubes. Thus the number of grid points in the y-direction is N + 1. Because of the "slip" scheme described earlier, the program inserts two more grid points, one near the I surface, and one near the E surface. Thus the total number of grid points with which the program works is N + 3. Within the program the grid points are numbered starting with 1 at the I (for internal) surface and extending to N + 3 at the E (for external) surface. The character I is used to index the grid points, and I then varies from 1 to N + 3. For convenience, the last three points are designated NP1 = N + 1, NP2 = N + 2, NP3 = N + 3. The two "slip" points, which have no real physical significance, are I = 2 and I = NP2.

The choice of N determines how fine or how coarse a grid structure is to be used, and only experience can tell what is necessary to achieve desired precision for a particular problem. For a turbulent boundary layer when "using the wall function" (this will be discussed further below), N in the range 15-20 is generally satisfactory. If "bypassing the wall function" is used, or if the flow is laminar, N should generally be greater than 30. If N is less than 12 the program will not operate, and N is limited to 50 by the dimensioning of the program. However, this limitation can be readily changed, if desired. Finally, it should be noted that the program will change N internally under special circumstances to be discussed later in connection with the input values of YPMIN and YPMAX.

KIN and KEX are indicators which determine the character of the I and E boundaries, respectively. If either is set equal to 1, that boundary is a wall; if set equal to 2, the boundary is a free stream; 3 indicates a line of symmetry, such as the centerline of a pipe or a free jet. As presently assembled, the program will handle only one wall surface, so, for example, it is not possible for both KIN and KEX to be equal to 1. Note that the I and E boundaries are literally "inner" and "outer" with respect to the axi-symmetric coordinate system, so, for example, for flow in a pipe the I boundary must be the centerline of the pipe and the E boundary must be the pipe wall; they are not interchangeable. On the other hand, for GEOM = 1 the I and E boundaries are interchangeable and either could be the wall.

KENT is an indicator for the entrainment calculation at a free boundary. If there is no free boundary, KENT can be left blank. If KENT = 0 entrainment is calculated based on the behavior of the momentum equation alone; if KENT = 1 all diffusion equations are tested. Since it is quite possible for the thermal boundary layer, for example, to extend outside the momentum boundary layer, and one generally wants to adjust entrainment so that the region of interest (the region enclosed by the I and E boundaries) encloses the thickest boundary layer, it is generally wise to set KENT = 1. On occasion this can lead to some instability, and this is the reason why the option to set KENT = 0 is provided.

The next card to be read contains more general information, all in the form of decimal numbers. XU is the present location of the calculations in the x-direction, and is one of the primary independent variables. Here XU is initialized, so this is where calculations start. Most often XU is 0.0, but it can be any positive number where it is desired to commence calculations. (Actually XU refers to the "upstream" side of the finite-difference step in the x-direction, as opposed to XD on the "downstream" side. The difference between XU and XD is DX, the step length.) XL is the x-distance where it is desired to stop calculations. Thus XU and XL, as read here, define the distance over which calculations are to take place. These are dimensional quantities and may be in feet, inches, meters, or whatever is desired. The actual dimensioning system to be used is designated later. Recall, as shown in Figure 4.1, that x is intrinsic, measured along the I-surface, and is <u>not</u> the projection onto the axis of symmetry.

DELTAX is a number (non-dimensional) from which DX, the step-length, is derived. It is the ratio of DX to boundary layer thickness, so DX grows as the boundary layer thickens. For a pipe-flow it is the ratio of DX to pipe radius. Actually, DELTAX determines a maximum value of DX and can be overridden by another number, FRA, which will be discussed shortly. DELTAX = 1.0 is a reasonable value when dealing with a gas for which properties are varying rapidly. If properties are nearly constant considerably larger values may be used and this is particularly true for laminar flows. For fully developed flow in a pipe DELTAX can sometimes be made equal to 10 or greater. If DELTAX is too large, a slight instability will be noted, with oscillation of the output data. It is often advantageous to use large values of DELTAX to reduce computation time. A further option is available using the constant K1 and the auxiliary boundary condition, AUX1(M) (see below), whereby DELTAX can be changed arbitrarily in the course of a calculation.

RETRAN provides a way to effect internally a transition from a laminar to a turbulent boundary layer. For a simple boundary layer, the momentum thickness Reynolds number is employed as a transition criterion, and RETRAN is the Reynolds number at which MODE will automatically shift from 1 to 2. Actually, the transition is made smoothly, rather than abruptly, over a range of momentum thickness Reynolds number from RETRAN to twice RETRAN by smoothly bringing the sublayer damping constant down from a large number to its equilibrium value (see SUBROUTINE WALL). Typically, a transition Reynolds number of 200-300 provides realistic results. If it is desired to make laminar boundary layer calculations only, care must be taken to make sure RETRAN is a number larger than any momentum thickness Reynolds numbers anticipated. For flow in a pipe or duct, RETRAN is interpreted as a diameter Reynolds number, but of course diameter Reynolds number does not vary in the x-direction in this case. For totally turbulent boundary layers and flows, RETRAN can be 0.0, or left blank, if desired. For free-convection boundary layers, or for flows for which there is no wall surface, Reynolds number has no useful significance, so RETRAN must be set to unity.

FRA, when multiplied times the total amount of flow between the I and E boundaries, specifies the maximum amount of new fluid that will be permitted to enter the region of interest between XU and XD either by entrainment or by mass transfer through a porous wall. If this amount is exceeded by the specified value of DELTAX, then DX is appropriately reduced in value. FRA = 0.05 is a reasonable value for most applications.

ENFRA is the entrainment fraction. It has significance only when there is a free-stream boundary, in which case it is the desired difference (expressed as a fraction of the total difference through the boundary layer) between the free-stream velocity, or the corresponding dependent variable in a diffusion equation, and the next closest grid point (excluding the slip point). This difference is maintained by automatically adjusting the rate of entrainment of free-stream fluid. The appropriate value of ENFRA differs somewhat for different kinds of problems, and is also related to the chosen grid spacing near the outer edge of the boundary layer. Calculated results are not necessarily highly sensitive to the value chosen for ENFRA, but a very inappropriate value can lead to either instability (wild oscillations in entrainment rate and in boundary layer thickness) or inaccurate overall results. For typical boundary layer calculations, turbulent or laminar, a value of 0.005 frequently works well, but a fine grid near the outer edge may suggest a value as low as 0.001. On the other hand, for a free-convection boundary layer or any case where free-stream velocity is at or near zero (for example, a jet) ENFRA should be very much larger, 0.01 to 0.05. One way to get a handle on ENFRA, in any case, is to plot the initial velocity profile, perhaps based on an appropriate analytic solution, and then superimpose the desired grid on the plot. The difference in velocity between the free-stream and the next adjacent grid line, divided by the maximum velocity difference for the whole boundary layer, is then usually a good value for ENFRA.

If there is no free-stream, as would be the case for pipe-flow, then ENFRA can be left blank.

GV is a gravity constant which should be either set at zero or left blank if gravity is not a relevant parameter. The only gravity effects that can be considered are those in the direction of flow (x-direction). Note that a positive value of GV represents a gravity force in the positive or flow direction; if simple free-convection on a vertical flat plate is being considered, remember that GV must be <u>negative</u>. Note also that gravity has no effect unless there are density gradients across the boundary layer; the free-convection boundary layer is a compressible flow boundary layer, and nothing will happen if FLUID = 1.

## 5.3 <u>Body Forces and Sources</u>

The next card read concerns some integer indicators having to do with body forces in the momentum equation, and energy and other types of sources in the diffusion equations. BODFOR can be 0, 1, or 2. If 0, there is no body force present other than pressure. If BODFOR = 1, the body force is the result of a gravity force acting upon density, and of course a value for GV must also be specified.

If BODFOR = 2, an external body force is present, and this force is introduced through a specified set of auxiliary boundary conditions AUX1(M), which will be discussed later. Provision is made only for a body force that is a function of x, and independent of y. BODFOR = 2 also includes BODFOR = 1. A body force has the dimensions force per unit of volume.

The source indicators, SOURCE(J), are not read unless there are one or more diffusion equations in addition to the momentum equation, i.e., unless NEQ is greater than 1, and NPH is greater than 0. The index J varies from 1 to NPH so that one value for SOURCE is read for each diffusion equation, reading across the card in integer fields of 5, after BODFOR.

If there is more than one diffusion equation it must be decided ahead of time which is which, and the designation of a source for each equation establishes what kind of a diffusion equation it is to be. Of course, the initial dependent variable profiles and the boundary conditions, both of which are discussed later, must be consistent with this choice.

If there is to be no source for a particular diffusion equation, set SOURCE = 0, or at least leave it blank. If SOURCE = 0, the equation could be the energy equation with viscous dissipation neglected, or it could be a mass diffusion equation with no chemical reaction. Only the initial and boundary conditions serve to make a distinction (together also with the Prandtl or Schmidt number), since the differential equations are identical.

SOURCE = 1 activates viscous dissipation as an energy source, as well as body-force work, and the equation is then definitely the energy equation.

Setting SOURCE = 2 for a particular diffusion equation has more extensive effects. It activates the source function for the turbulence energy equation (Production-Dissipation), but additionally it changes the method of calculation of eddy viscosity (and thus eddy conductivity) from the mixing-length method to the turbulent kinetic energy method, wherein eddy viscosity is proportional to the square root of the turbulent kinetic energy. However, the program still uses mixing-length in the Wall Function, and it still uses mixing-length out to the edge of the viscous sublayer if the Wall Function does not extend that far. If there is no wall, turbulent kinetic energy is used throughout.

SOURCE = 3 is the same as SOURCE = 1, except that an external energy source, as a function of x alone, may be introduced through AUX2(M). Such a source must have the dimensions (energy)/(volume * time).

SOURCE = 4 also implies that an external volume source is being introduced through AUX2(M), but viscous dissipation and body-force work are omitted, so this could be a source different from energy.

Note that all of these external body forces and sources which are introduced through the auxiliary functions AUX1(M) and AUX2(M) are functions of x only. This is obviously somewhat limiting, but the only practical way to introduce sources that vary in the y-direction is by modifications in the core of the program. However, this can be easily done in SUBROUTINE AUX, where some comments are given.

## 5.4 Fluid Properties

The next card is the one in which fluid properties are introduced. The amount of information actually read depends upon whether constant properties are to be used or whether the variable properties contained in a separate property subroutine are to be used. In any case the initial static pressure PO is always read, and for the variable property case this is all that is needed. For constant properties, density, RHOC, and viscosity, VISCOC, are next to be read; if only the momentum equation is to be solved this is all that is necessary. If one or more diffusion equations are to be solved, the only additional property is the Prandtl number for the energy equation, PRC(J), or a Schmidt number for each and every mass diffusion equation, making sure that they are read in the same order as has been established for designating each equation, i.e., $J = 1$ refers to a particular diffusion equation, and $J = 2$ to another, and this order must be maintained throughout the entire input routine. Note that although the symbol PRC(J) is used, this can be either a Prandtl or a Schmidt number. Finally, all units must comprise a consistent set. Note that the read statements are so arranged that it doesn't matter if there is a redundancy of information. Thus the program might be set up to solve both momentum and energy equations with constant properties; but if in the second card FLUID is changed to 2 the program will run with variable air properties and simply will not read the constant properties (except PO). Similarly, if NEQ is changed to 1, the program will not read Prandtl number or anything else having to do with a diffusion equation; it is not necessary to remove this input information if an abbreviated problem is to be run. As a word of caution, do not try to solve the momentum equation alone without setting FLUID = 1 and supplying the appropriate constant properties. There is no way to introduce variable properties without temperature or mass concentration distributions upon which to base them.

5.5 Boundary Conditions

The next card supplies some information about types of boundary conditions, and the number of entries read depends upon the number of differential equations to be solved. NXBC, an integer number, refers to the number of points along the boundaries at which boundary condition information is to be supplied. The cards following will contain that information. Internally, the program will determine boundary values at each XU position by linear interpolation between the x-positions of the input boundary data as specified here. Thus NXBC must have as a minimum a value of 2 so that there is something to interpolate between. If boundary values are varying with respect to x in other than a linear manner, many more than two boundary values may be required for an accurate representation. The program is dimensioned such that NXBC may be as large as 100. Free-stream velocity is evaluated from a cubic spline fit scheme rather than linear interpolation, except that when NXBC = 2 linear interpolation is used.

The other items read on this card refer to the type of boundary condition at a wall that is going to be supplied for any and all diffusion equations. If there is no wall nothing is read, and the same is true if only the momentum equation is to be solved. TYPBC(J) can be either 1 or 2, depending upon whether the boundary condition read is to be, respectively, a specification of the value of the dependent variable at the wall, or the flux of the dependent variable at the wall. In the case of the energy equation, the question is whether it is the enthalpy at the wall or the heat flux that is to be specified. For the turbulent kinetic energy equation set TYPBC(J) = 1. Of course a specification for every diffusion equation must be supplied, and in the proper order.

The following card continues boundary specifications. These items, all decimal numbers, are read in the form of a table. The number of lines in the table must be equal to NXBC. X(M) is the x location of the points where boundary information is to be supplied. The first entry, X(1), must be equal to or less than XU read earlier; the last entry, X(NXBC), must be equal to or greater than XL. For a variable velocity boundary condition, the value of XU must coincide with an X(M) in the table. Between the first and last point, the spacing of any other boundary condition points can be completely arbitrary. Discontinuities, for example, can be simulated by placing two points very close together. When free-stream velocity is changing rapidly, it is important to use a large number of points and not produce situations that a spline fit will have difficulty accommodating; abrupt changes of velocity are troublesome and can lead to unwanted velocities between the specified points.

RW(M) is a geometry specification for an axi-symmetric body. It is the transverse radius of the body at each specified x-location. Note that RW is a function of x, distance measured along the surface, and not the projection onto the axis of symmetry. The boundary layer can be either on the inside or the outside of the body for GEOM = 1. GEOM = 2 and 3 are restricted in this regard. For a pipe, GEOM = 4, RW(M) is the pipe radius. For a boundary layer on a non-axisymmetric body, for example a flat plate or an airfoil, use GEOM = 1 and set all values of RW(M) equal to any constant number, such as 1.0. For an axi-symmetric stagnation point use GEOM = 1 or 2 and set RW(M) = X(M). For a flat duct, GEOM = 5; RW(M) is the duct half-width.

Two additional pieces of boundary condition information can, if desired, be read on this card, AUX1(M) and AUX2(M). It has already been noted that these auxiliary items can be used for specified body forces or specified internal heat sources, if proper indicators are activated. AUX1(M) can also be used to provide a control on DELTAX. These functions, however, are provided in general so that the user can conveniently introduce any kind of information that is a function of x, and then appropriately modify the core of the program to make use of the information. If there is a wall present, the program additionally calculates two more functions, AUXM1 and AUXM2, which are linearly interpolated values of AUX1(M) and AUX2(M), and are always available in the COMMON.

The primary boundary condition data are read on the next cards, again in the form of a table in which the number of lines must equal NXBC. UG(M) is the free-stream velocity which must always be supplied if there is indeed a free stream. (In the case of a pipe or duct flow this column can be left blank.) A particular feature of this version of the program is the fact that free-stream velocity is treated as an independent boundary condition rather than pressure or pressure gradient. A minor modification of the basic program is necessary if pressure is to be the independent boundary condition. Note that UG is zero for simple free convection, or for a jet in a stagnant environment.

The second column (second field of 10 spaces) is the mass flux at the wall, AM(M). If there is no wall this column is simply not read. AM is positive in the positive direction of the coordinate system. Thus, if the I boundary is a wall, positive AM is mass transfer into the boundary layer, but if the E boundary is a wall (as in pipe-flow), <u>negative</u> AM is mass transfer <u>into</u> the boundary layer.

The next five columns are read only if there is a wall and if one or more diffusion equations are to be solved. FJ(J,M) is either the wall value of the dependent variable in a diffusion equation or it is the wall value of the dependent variable flux. Whether it is a wall value or a flux is determined by TYPBC(J), discussed above. Thus for the energy equation FJ is either a wall value of enthalpy or a wall value of heat flux. The sign of the flux is again positive in the positive direction of the coordinate system which goes in the direction from I to E. Thus, for flow in a pipe, a heat flux into the fluid results when FJ is negative. Care must be taken when FJ is a flux and there is mass transfer at the wall. FJ is then the product of the mass flux and the value of the property in question in a reservoir outside the wall. For example, for the energy equation with FJ as a flux, FJ is the product of AM and the enthalpy of the transferred fluid in an external reservoir. For the turbulent kinetic energy equation, FJ should be 0.0.

## 5.6 Initial Profiles

The next series of cards contains the initial or starting profiles for velocity and the dependent variables for each of the diffusion equations. These are read in the form of a table, as in the previous case. The number of entries in the vertical columns must be equal to N + 1. Each column again occupies 10 spaces.

The first column contains Y(I), the distance measured from the I-boundary for each of the grid points at which the other information is to be supplied. In Y(I), I is an integer which varies from 1 to NP3, but 2 and NP2 are omitted, since these are the slip positions which are evaluated within the program. Thus the table will contain N + 1 entries. Y(1) is always 0.0, since y is measured from the I-boundary.

The spacing of the various Y(I) is very important, since it establishes the cross-stream grid for the entire boundary layer calculation. First, the obvious fact should be noted that it is not possible to start finite-difference calculations with this program from a singularity; starting profiles are mandatory, but the boundary layer can be as thin as desired, although a very thin starting boundary layer may require a large number of calculations to progress very far in the x-direction. Generally, the starting profiles are where analyti boundary layer solutions can be used to great advantage. Typically, one knows something like momentum thickness Reynolds number at the start, and simple analytic solutions can then be used to establish initial total thickness and initia profile shapes. Actually, since boundary layers, and especially turbulent bound ary layers, come to equilibrium relatively quickly, the initial profile shapes are often not at all critical; it is important that the initial integral parameters (such as momentum and enthalpy thickness) be close to correct. For example, a laminar boundary layer calculation could be started with a simple linear velocity profile, and within a few downstream steps the correct profile will be closely approximated. An exception to this discussion is flow in a pipe or duct where the "boundary layer thickness" is always the distance from the wall to the centerline. It is possible to start such calculations with a uniform velocity profile and thus calculate the velocity entry length, but for accuracy this does require using a relatively fine grid spacing near the wall.

Now, to get back to the Y(I) spacing, the reason it is so important is that the program reads the initial data, calculates the fluid flow in each flow tube, totals this for the entire region from I to E, and then calculates the fraction of the flow in each flow tube. As the boundary layer grows, the total flow in the region I to E may grow due to entrainment and/or mass transfer, and the distance from I to E may grow, but the fraction of the total flow in each flow tube is maintained constant. The fraction of the flow from the I-boundary to some Y(I) is given the symbol OM(I) (omega). Thus the flow between the I and the I + 1 grid point is OM(I+1) - OM(I). It is these initial values of OM(I) that remain the same throughout the calculation (with an exception to be discussed below). Now there is no requirement that the OM spacings be uniform; on the contrary, it is generally more efficient if they are not. But it is important that the OM spacing differences between adjacent flow tubes not be too large. As a rule of thumb, differences greater than a factor of about 3 should be avoided. A good way to set up the initial velocity profile is to lay it out on a piece of graph paper and then superimpos lines for grid points, crowding them closer together in the regions where veloc ity is changing rapidly. A mental estimate of the relative flow rate between each pair of grid lines will usually suffice to make sure that large steps in flow rates are avoided. This graphical procedure was also recommended as a guide for specifying entrainment fraction.

If there is a wall and the boundary layer is turbulent, a decision must be made whether to use a small number of grid points, along with "using the Wall Function", or to "bypass the Wall Function" and use a fine grid down to the wall. For a great many calculations the results will not differ much, and "using the Wall Function" is a little simpler and cheaper in computation time. For very high Reynolds numbers there is really no choice; a grid fine enough to allow "bypassing the Wall Function" may require an excessive number of grid points. "Bypassing the Wall Function" does become useful where pressure gradients are large, or boundary conditions are changing rapidly along the wall, or it is simply desirable to have a print-out of the variables near the wall. The accuracy question comes down to the adequacy of the Couette flow approximation, which is used in the Wall Function. For large adverse pressure gradients, for example, the Couette flow approximation begins to yield a substantial error in local shear stress in a typical case when $y^+$ becomes larger than about 15 or 20.

When "bypassing the Wall Function", it is necessary that U and y at the first grid point (I = 3 if KIN = 1, or I = NP1 if KEX = 1) be so chosen that $y^+$ is about 1.00, or less. This can be checked by multiplying U by y and dividing by kinematic viscosity, which gives the product $U^+y^+$. In this region $U^+ = y^+$, approximately. The spacing of the grid points farther from the wall can then be gradually increased by steps of perhaps 20 percent out to about $y^+ = 20$, and 25-30 percent thereafter, i.e., $y^+ = 1.2, 1.4, 1.7, 2.1$, etc.

When "using the Wall Function" it is important that the first grid point be at a value of $y^+$ not less than about 20.0. The subsequent points can then be spaced at intervals that increase by 25 to 30 percent, i.e., 25.0, 31.0, 39.0, 49.0, etc.

For both cases, after $y^+$ becomes greater than about 200, quite large, equally spaced steps generally can be used because velocity is no longer changing rapidly. The important thing is to concentrate the grid where rapid changes are taking place.

It is important that the velocity at the free-stream edge of the boundary layer be precisely the same as the value of free-stream velocity introduced as a boundary condition earlier.

Having once established the initial velocity profile, the other columns are filled in with the corresponding initial dependent variable profiles for the diffusion equations, all in the same order as discussed earlier. Any of these can be totally zeros if desired, or all equal to the free-stream value, as would be the case for a heat transfer problem with an unheated starting length. For turbulent kinetic energy it is possible to start with all zeros and the program will generate its own kinetic energy. In any case, the wall value of turbulent kinetic energy should be 0.0. In the case of the energy equation the dependent variable is always stagnation enthalpy, <u>not</u> temperature.

The value of the dependent variable in the diffusion equations at the outer edge of the boundary layer is always constant, and is established by the value specified in the initial profiles.

## 5.7 <u>Turbulence Constants</u>

Some of the turbulence constants are read in the next cards. If the flow is laminar, dummy turbulent values can be used, or these entries can be left blank. If there is no wall, some of the constants are also redundant.

AK is the wall region mixing-length constant, kappa. There is not total agreement on the value of kappa, but 0.41 is extensively used. ALMGG is lambda, the outer region mixing-length constant (or outer region length-scale constant when turbulent kinetic energy is used). There is also a constant eddy diffusivity option available (see below) in which case ALMGG becomes a dummy. For boundary layers a value of 0.085 appears reasonable; for flow in a pipe 0.07 is suggested, but the constant diffusivity option is recommended for pipe-flow. For a boundary layer the value for ALMGG is overridden at momentum thickness Reynolds numbers below about 5500 by an internal correlation that yields a higher value. This override can be suppressed by setting K2 = 3 (see below).

ALMGG is a non-dimensional constant which yields a mixing-length when multiplied by boundary layer thickness. But "boundary layer thickness" must be defined, and FR provides this definition. If FR is set equal to 0.01, the boundary layer thickness upon which ALMGG is based is the distance from the wall to the point where the velocity is within 1 percent of free-stream velocity. The suggested values for ALMGG given above are based on FR = 0.01.

AQ and BQ are turbulence constants which are used for the turbulence energy equation, but also for the constant eddy diffusivity option. In the former case AQ is the eddy diffusivity constant while BQ is the dissipation constant. Of the three constants, AK, AQ, BQ, only two are independent. The three are related by the equation:

$$AK = (AQ ** 0.75)/(BQ ** 0.25)$$

If AK = 0.41, some reasonable values for AQ and BQ are 0.22 and 0.38.

When the constant eddy diffusivity option is used (set K2 = 2, see below), eddy diffusivity in the outer region is evaluated from the equation:

$$\frac{\varepsilon_M}{\nu} = AQ * (\text{Reynolds number}) ** BQ$$

For an external boundary layer, momentum thickness Reynolds number is used; for flow in a pipe or duct, diameter Reynolds number is used. Reasonable values for the pipe case are AQ = 0.005, BQ = 0.9.

YPMAX and YPMIN are controls on the values of $y^+$ at the outer edge of the Wall Function. They are operable whether the flow is laminar or turbulent, but are meaningless if there is no wall. Routine LAMSUB provides a scheme whereby extra grid points can be automatically inserted between the wall and the next point out, or grid points can be removed from the same region. In other words, the grid number N is changed. YPMAX sets a maximum limit on the value of $y^+$ at the outer edge of the Wall Function. If this limit is exceeded an extra grid point will be inserted. YPMIN sets a minimum limit on the value of $y^+$ at the outer edge of the Wall Function. If $y^+$ at the outer edge is less than this limit, the grid point nearest the wall will be removed.

When using the Wall Function, a typical procedure is to set YPMIN = 20.0 and YPMAX = 50.0 to 100.0. When bypassing the Wall Function, set YPMIN = 0.0 and YPMAX = 1.0. This scheme is also useful in setting up the initial profiles when it is desired to bypass the Wall Function. For example, a rather coarse grid can be introduced in which $y^+$ at the innermost point is, say, 50.0. Then if YPMIN = 0.0 and YPMAX = 1.0, the program will insert a series of points down to near $y^+ = 1.0$, with optimal spacing.

The damping function constant for the viscous sublayer is read in the next card. Two options are available, together with some variations. APL refers to $A^+$ in the Van Driest exponential damping function scheme; BPL refers to $B^+$ in the Evans linear damping function scheme. The program will use the scheme for which the larger number is indicated, i.e., if APL is larger than BPL, the Van Driest scheme will be used, and vice versa. In either case an empirical internal correlation is used to modify the value of APL or BPL to account for the effects of pressure gradient and transpiration. For the Van Driest scheme, $A^+ = 25.0$ is suggested for external boundary layers, and $A^+ = 26.0$ for flow in a circular pipe. $B^+ = 35.0$ appears to be about correct for the Evans scheme. In any event, the user is urged to experiment with the constants and compare results against proven experimental data. If it is desired to not use the internal correlation for transpiration and pressure gradient, SIGNAL should be set to 1.0; otherwise SIGNAL may simply be left blank. For example, the internal equation for the effects of pressure gradient is probably not valid for a free-convection boundary layer, or for any boundary layer involving body forces in the flow direction, so in such a case set SIGNAL = 1.0.

The next card contains a lag constant, PPLAG, to account for the time required for the sublayer to adjust to different externally imposed conditions, such as pressure gradient or transpiration. PPLAG = 4000.0 has been found to be reasonably satisfactory.

Also in this card is read the turbulent Prandtl number, PRT(J), for each of the diffusion equations. PRT(J) is not read if the flow is laminar, nor is it read if only the momentum equation is being solved. The program contains an internal calculation for turbulent Prandtl number near a wall, based on a conduction model. The value of turbulent Prandtl number read here is the value for a region far removed from the wall. However, this value is used in the near-wall analysis and does affect it directly and importantly. Right at the wall, turbulent Prandtl number is computed to be twice the value far removed from the wall. For the energy equation it has been found that PRT(J) = 0.86 gives reasonable results for air, and is also quite satisfactory for liquid metals. In the latter case the internal analysis yields a turbulent Prandtl number over the entire region of interest considerably greater than the value of PRT(J) read in the input.

For the turbulent kinetic energy equation the internal correlation is not used, and a value of PRT(J) = 1.7 may be about right, although there is great uncertainty about this figure.

If it is desired to suppress the internal calculation for turbulent Prandtl number and thus use a constant turbulent Prandtl number (or turbulent Schmidt number) throughout, set K3 = 3, as described later.

## 5.8 Other Constants and Output

The dimensioning system used is established in the next card. GC is the constant in Newton's Second Law ($g_c$). If SI units are being used, GC = 1.0. If English Engineering units are used GC = 32.2 ($lb_m$ ft)/($lb_f$ $sec^2$), etc. CJ is the proportionality factor in the First Law of Thermodynamics (J). Again, if SI units are used, CJ = 1.0; but with English Engineering units, J = 778 ft-$lb_f$/Btu. The other quantities read on this card, AXX, etc., are merely auxiliary constants which may be employed by the user for special purposes, after making appropriate adjustments inside the program.

The final card reads some integer numbers concerned with a number of different things. The first, NUMRUN, is the number of sets of DATA that are to be read. Ordinarily this would be 1, but DATA sets may be stacked if desired. SPACE designates the number of integrations between output prints, i.e., if SPACE = 10, the program will print out a complete set of results every 10 integrations in the x-direction. There are two special cases. If SPACE = 11, a one-line set of abbreviated results will be printed out for every integration; if SPACE = 21, a complete set of results will be printed every 20 integrations, and a one-line abbreviated set will be printed for every integration.

OUTPUT is a number designating the particular output format to be used. Three are presently available, designated by the integer numbers 2, 4, 6.

OUTPUT 6 is a general-purpose routine usable for any kind of problem. Complete profiles of all dependent variables are printed, together with numerous other pieces of information such as shear stress at a wall, heat flux, entrainment rates, eddy viscosity, etc. This routine is the only one which is usable for KEX = 1, as well as KIN = 1, and it is the only one which should be used when free-stream velocity is at or near zero.

OUTPUT 2 is especially designed for external boundary layers when the I-boundary is a wall. $U^+$ and $y^+$ are printed, as well as the dimensional profiles; and the non-dimensional parameters $C_f/2$, St, momentum thickness Reynolds number, enthalpy thickness Reynolds number, are all printed.

OUTPUT 4 is a routine for flow in a pipe or duct. Parameters peculiar to this type of problem, such as mean velocity, mixed-mean enthalpy, and diameter Reynolds number, are printed along with the pertinent profiles.

The options SPACE = 11 and 21 are available only for output routines 2 and 4.

Some additional data may be printed with any of the output routines by setting the indicator K1 (see below) to any number greater than 10. Five specially designated pieces of information, SP(1), ... SP(5), will be printed, but they must first be assigned at some point in the body of the program. This option simply provides the user with a simple method of capturing additional information of his own choosing.

The integer indicators K1, K2, K3, have been mentioned several times in this chapter. These indicators provide the user with a convenient scheme for causing particular things to happen within the program. They have already been used for a number of purposes, but the user still has the option for other uses. The uses already programmed are as follows:

| | |
|---|---|
| K1 greater than 10: | Five specially defined pieces of information will be printed in all of the output routines. |
| K1 equal to 9 or 20: | DELTAX becomes equal to AUX1(M), and the input value of DELTAX is overridden. |
| K2 equal to 2: | Program will use the constant eddy diffusivity option in the outer region, rather than mixing-length. |
| K2 equal to 3: | An internal empirical equation for ALMGG will be suppressed, and the input value of ALMGG will be used throughout. |
| K3 equal to 3: | An internal calculation for turbulent Prandtl number will be suppressed, and the input values of turbulent Prandtl number will be used as a constant throughout. |

## Chapter 6

## PROGRAM ORGANIZATION

### 6.1 Structure of the Program

Program STAN5 consists of a driver program and six subroutines.

The driver program, MAIN, sets all boundary conditions and conducts the integration. In addition, fluid properties, entrainment, DX stepsize, and integral parameters are calculated in this routine.

SUBROUTINE STEP is a package containing five subsections. In STEP(1), the initial slip points and $\beta$ and $\gamma$ near the I and E surfaces are computed. STEP(2) computes the initial radii and converts the initial $y$'s to $\psi$'s and then to $\omega$'s. These two routines are required only at the start of integration or if LAMSUB readjusts the profiles. STEP(3) computes $y$'s from the velocity profile, and the $\psi$ distribution and the radii associated with these $y$ locations. Also, the velocity profile is searched for its maximum and minimum values, and the boundary layer thickness is determined. In STEP(4), all finite-difference coefficients are formed and the resulting FDE's are solved. STEP(5) is used to initialize variables at the start of integration.

If one of the bounding surfaces is a wall, SUBROUTINE WALL computes wall shear stress and heat flux, along with $C_f/2$ and St. The internal correlation for $A^+$ or $B^+$ as a function of $V_o^+$, $P^+$, and $BF^+$, and LAMSUB, are contained in this subroutine.

Effective viscosities and effective Prandtl numbers for turbulent flow calculations are computed in SUBROUTINE AUX, and, in addition, all source terms for the $\phi$-equations, e.g., viscous dissipation or TKE production and dissipation.

Printing during integration is via SUBROUTINE OUT, which contains three subsections, with the first designed primarily for external boundary layers, the second for pipe flows, and the third for a general output.

SUBROUTINE PROP2 is a variable-properties table for air at moderate temperatures, to be used with compressible flow calculations.

SUBROUTINE INPUT reads and prints all input variables. In addition, it performs diagnostics on these variables to look for "pitfalls" associated with setting up a problem or incompatibilities among the variables.

## 6.2 MAIN

The driver of any program is generally the most complex routine, and the one contained in STAN5 is no exception. Therefore, it has been diagrammed and is given in Figure 6.1. Since the flow chart presents the sequence of events straightforwardly, no further discussion is felt necessary.

## 6.3 STEP

Five sections comprise STEP(K), with STEP1, STEP3, and STEP4 very similar in content to that found in Patankar and Spalding [1,2].

STEP1 computes slip-point quantities near the I and E surfaces and $\beta$ and $\gamma$ (see Section 4.3). This routine is used only for the initial profiles and for profiles readjusted by LAMSUB (see Section 3.5).

STEP2 has two functions, and is used only for the initial profiles and for profiles readjusted by LAMSUB. It computes the radii that correspond to initial values of $y$ in the velocity profile. It also converts the initial $y$ table to $\psi$, using equation (4.1), and finally to $\omega$, OM(I), using equation (4.4), with $\psi_I$ arbitrarily set to zero. Note that integration of equation (4.1) between the I and E surfaces gives mass flow rate per radian (or unit depth for two-dimensional flows). The variable PEI is this quantity. For internal flows, PEI remains constant (unless there is mass transfer at the wall), and for external flows PEI is increased at each integration step due to entrainment or wall mass transfer.

STEP3 has three functions; it is called at each integration step. This routine computes $y$ locations of the nodes by integrating the velocity profile using equations (4.1) and (4.4) and the mass flow rate per unit radian, PEI. The radii are then calculated from the $y$'s. Finally, the velocity profile is searched to obtain maximum and minimum velocities, UMAX and UMIN, and the input variable FR is multiplied times (UMAX - UMIN). The $y$ table is then interpolated to obtain the location for this product, YL; this variable is the boundary layer thickness, defined as delta sub (1.000 - FR). For pipe flows YL is the wall radius.

STEP 4 has two functions; it is called at each integration step. It computes the velocity finite-difference coefficients AU(I), BU(I), and CU(I), and those for the $\phi$ equations, A(J,I), B(J,I), and C(J,I). The FDE's are then assembled and solved to obtain profiles for velocity, U(I), and $\phi$-dependent variables, F(J,I).

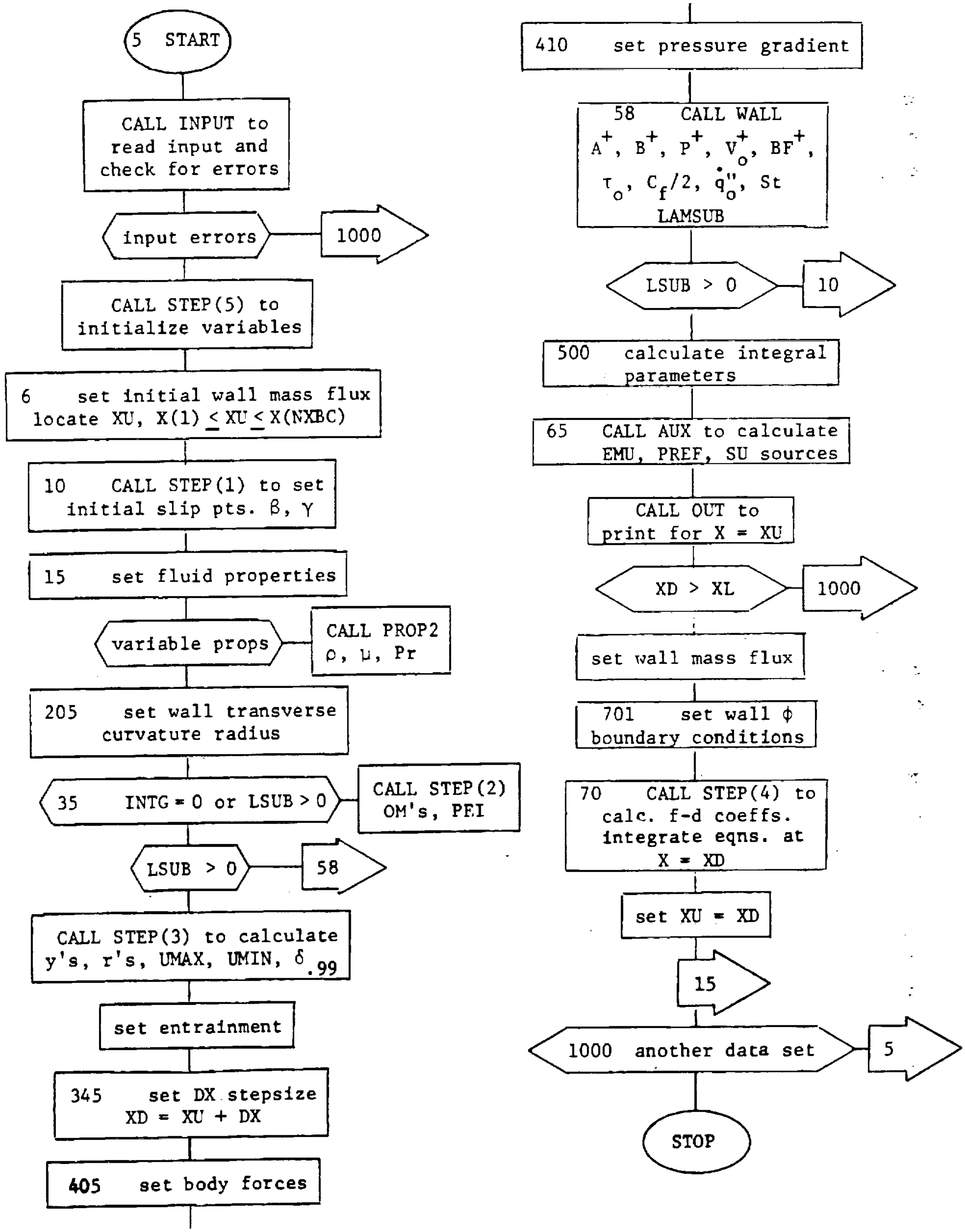



Figure 6.1. Flow chart of the driver routine in STAN5.

STEP5 is called at the beginning of the program to zero the arrays and initialize parameters.

6.4 WALL

SUBROUTINE WALL performs the functions described in Sections 3.1 through 3.5 to determine friction factor and Stanton number. It is called one or more times per integration (depending on whether LAMSUB is invoked) providing one surface is a wall.

The first part of the routine sets up the join point conditions for velocity and stagnation enthalpy: $y_{2.5}$ is YI; $U_{2.5}$ is UI; $I^*_{2.5}$ is FI(J); and $Re_{2.5}$ is REW. The shear velocity $U_\tau$, UTAUW, is also computed using the wall shear stress from the previous integration step.

The second part of the routine sets up various source terms for the stagnation enthalpy Wall Function equation (3.14). The variable C3 is W, C4 is $W \cdot X^+/2$, and C5 is the term in equation (3.4k) to convert s to $S^+$. Since the non-dimensionalizations contain $\dot{q}''_o$ in the denominator; an adiabatic wall should be simulated with a very small but non-zero heat flux.

In the third section Couette flow quantities are formed: PPL is $P^+$; GPL is $V^+_o$; and BFPLUS is $X^+$. These quantities are then converted into effective values by solving a lag equation (2.25) for $V^+_{o,eff}$, GPLE; and for $(P^+-X^+)_{eff}$, PPLE. The constant in equation (2.25) is the input variable PPLAG. Finally, the $A^+$ or $B^+$ equation (2.24) is evaluated using these effective values. If transition from laminar to turbulent flow is in progress, $A^+$ or $B^+$ is modified according to equation (2.38).

The fourth section examines the join-point Reynolds number. If it is less than 4 (which is synonymous with setting the input variable YPMAX < 2), the Wall Function is bypassed (section six below); otherwise section five is used.

The fifth section of SUBROUTINE WALL is "using the Wall Function". Here equation (3.9) is solved for $U^+$ and equation (3.14) is solved for $I^{*+}$. Both equations are numerically integrated by a trapezoidal rule using progressively larger $\Delta y^+$ steps, DYPL. In the output from this section $y^+_{2.5}$ is YPL, $U^+_{2.5}$ is UPL, and $I^{*+}_{2.5}$ is HPS(J). When the $U^+y^+$ product equals $Re_{2.5}$, control is transferred to section seven, described below. During integration $\tau^+$ and $y^+$ are continuously monitored, and if $\tau^+$ becomes less than 0.1 or

$y^+$ becomes greater than YPMAX, control is transferred to LAMSUB to insert a new point near the wall.

The Wall Function bypass option is contained in the sixth subsection of WALL. Here equations (4.18a-c) are solved for $y^+_{2.5}$, with $U^+_{2.5}$ computed from the definition of $Re_{2.5}$.

Outputs from either section five or six are used in section seven to compute wall shear stress, TAUW, using equation (3.11a). The friction factor, CF2, is then formed following equation (3.11b). If there are no $\phi$ equations being solved, control is passed to section ten of WALL.

If $\phi$ equations are being considered, section eight is used, providing the Wall Function is being bypassed, and equation (3.20) is solved for $I^{*+}_{2.5}$.

Section nine uses $I^{*+}_{2.5}$ from either section five or eight to compute wall heat flux, QW(J), and Stanton number, ST(J). If there is a total flux boundary condition, the wall value of $\phi$ is computed (see Section 3.3.2).

Routine LAMSUB is contained in the tenth section. It is invoked in accordance with equation (3.21), which is fully described in Section 3.5.

## 6.5 AUX

In the first part of subroutine AUX, the turbulent viscosity and conductivity for each node is computed and added to its laminar counterpart to obtain an effective viscosity and conductivity.

Computation of the turbulent viscosity at each node begins with evaluating the damping function, DV(I), as described by equation (2.22) or (2.23). Then the $\lambda\delta_{.99}$ mixing-length, AL, is evaluated according to equation (2.26), with $\lambda$, ALMG, obtained from equation (2.27). If the flow is in the near-wall region, the mixing-length is switched to $\kappa yD$, equation (2.21)

Once a mixing-length for the node is established, the turbulent viscosity $\mu_t$, EMUT, is evaluated using either the Prandtl mixing-length model, equation (2.19), or the constant eddy viscosity model, equation (2.36), or the turbulent kinetic energy model, equation (2.28). The turbulent viscosity is added to the laminar viscosity to form an effective viscosity, EMU(I), as defined by equation (2.6).

If the stagnation entahlpy equation is being solved, the turbulent Prandtl number, PRTJ, is set either to its input value, PRT(J), or to a value calculated using the variable turbulent Prandtl number model, equation (2.37).

For TKE the input variable is the turbulent Schmidt number. An effective Prandtl/Schmidt number, PREF(J,I), is formed according to equation (2.14).

In the second part of subroutine AUX all source terms for the $\phi$-equations are formulated. The sources are defined as all terms to the right of the equal sign after a $\phi$-equation is transformed using equation (4.7), and finite-differenced according to equation (4.11).

References

1. Patankar, S. V., and Spalding, D. B., Heat and Mass Transfer in Boundary Layers, 1st Ed., Morgan-Grampian, London, 1967.

2. Patankar, S. V., and Spalding, D. B., Heat and Mass Transfer in Boundary Layers, 2nd Ed., International Textbook Company Ltd., London, 1970.

3. Kays, W. M., and Moffat, R. J., "The Behavior of Transpired Turbulent Boundary Layers," Studies in Convection, Vol. 1: Theory, Measurement and Application, Academic Press, London, 1975.

4. Launder, B., and Jones, W. P., "On the Prediction of Laminarization," presented at the ARC Heat and Mass Transfer Subcommittee Meeting of April 5, 1968.

5. Julien, H. L., Kays, W. M., and Moffat, W. M., "Experimental Hydrodynamics of the Accelerated Turbulent Boundary Layer With and Without Mass Addition," Jn. Heat Transfer, Trans. ASME, Series C, Vol. 93, No. 4, p. 373, November 1971.

6. Spalding, D. B., "A General Computer Program for Two-Dimensional Boundary-Layer Problems," Report HTS/73/48, Dept. of Mech. Engrg., Imperial College of Science and Technology, September, 1973.

## Appendix I

## PROGRAM NOMENCLATURE

A(J,I) Finite-difference coefficient for $\phi$-equations.

A2 Integral term in Couette flow form of momentum equation.

AJE(J) Linear-interpolated value of FJ(J,M) if flux type boundary condition and E-surface is a wall.

AJI(J) Linear-interpolated value of FJ(J,M) if flux type boundary condition and I-surface is a wall.

AK Kappa in Prandtl mixing-length model.

ALMG Outer layer constant in Prandtl mixing-length constant, modified if low Reynolds number (K2$\neq$3).

ALMGG Input value of outer layer constant in Prandtl mixing-length model.

AM(M) Wall mass flux boundary condition, positive in direction of increasing y.

AME Linear-interpolated value of AM(J) if wall mass flux and E-surface is a wall.

AMI Linear-interpolated value of AM(J) if wall mass flux and I-surface is a wall.

APL Van Driest damping coefficient in mixing-length model, input value SIGNAL=1.) or computed from internal correlation (SIGNAL=0.).

AQ Production constant in TKE model or constant in eddy diffusivity model.

AU(I) Finite-difference coefficient for velocity equation.

AUX1(M) Generalized x-direction body force for momentum equation (BODFOR=2) in units of force/unit volume, specified at each X(M).

AUX2(M) Generalized energy equation source [SOURCE(J)=2,3] in units of energy rate/unit volume, specified at each X(M).

AUXM1 Linear-interpolated value of AUX1(M).

AUXM2 Linear-interpolated value of AUX2(M).

AXX Not used by program.

| | |
|---|---|
| B(J,I) | Finite-difference coefficient for $\phi$-equation. |
| BETA | Power of y in slip scheme, near-wall region. |
| BF(I) | Body force term for momentum equation (gravity, AUXM1 for BODFOR$\neq$0). |
| BFPLUS | Body force in "wall coordinates" $(X^+)$. |
| BODFOR | Type of body force for momentum equation. |
| BPL | Evans damping coefficient in mixing-length model, input value (SIGNAL=1.) or computed from internal correlation (SIGNAL=0.). |
| BQ | Dissipation constant in TKE model or constant in eddy diffusivity model. |
| BU(I) | Finite-difference coefficient for velocity equation. |
| BXX | Not used by program. |
| C(J,I) | Finite-difference coefficient for $\phi$-equation. |
| CAY | Acceleration parameter, $(\nu/U_\infty^2)\,dU_\infty/dx$. |
| CF2 | Wall friction coefficient, $C_f/2$. |
| CJ | Conversion factor, mechanical to thermal energy. |
| CSALFA | Cosine $\alpha$, to relate y and r. |
| CU(I) | Finite-difference coefficient for velocity equation. |
| CXX | Not used by program. |
| DEL1 | Boundary layer displacement thickness. |
| DEL2 | Boundary layer momentum thickness. |
| DEL3 | Boundary layer enthalpy thickness. |
| DELTAX | Maximum integration stepsize (DELTAX * YL). |
| DPDX | Pressure gradient due to free-stream velocity variation and free-stream body force (pressure gradient to conserve continuity and momentum if pipe/channel flow). |
| DX | Integration stepsize (computed by program). |
| DXX | Not used by program. |

| | |
|---|---|
| EMU(I) | Effective dynamic viscosity, sum of laminar and turbulent contributio |
| ENFRA | Entrainment fraction to control boundary layer entrainment. |
| EXX | Not used by program. |
| F(J,I) | $\phi$-dependent variable in $\phi$-equations (e.g., stagnation enthalpy or TKE equations) at Y(I). |
| FI(J) | Join-point value of F(J,I). |
| FJ(J,I) | Boundary value of F(J,I), specified at each X(M) [level if TYPBC(J)=1 and flux if TYPBC(J)=2]. |
| FLUID | Type of free-stream fluid. |
| FMEAN | Bulk-mean stagnation enthalpy for pipe flow, to adjust Stanton number |
| FR | Defines boundary layer thickness. |
| FRA | Fraction to determine DX stepsize. |
| GAMA(J) | Power of y in slip scheme, near-wall region. |
| GC | Proportionality constant, Newton's 2nd Law. |
| GEOM | Geometry descriptor. |
| GPL | Blowing parameter in "wall coordinates" ($V_o^+$). |
| GV | Gravity constant for momentum body force. |
| H | Boundary layer shape factor. |
| I | Cross-stream index for dependent variable (I = 1 at y = 0). |
| INDE(J) | Type of boundary condition at E-surface (TYPBC(J)). |
| INDI(J) | Type of boundary condition at I-surface (TYPBC(J)). |
| INTG | Integration step counter. |
| ITKE | I index value at edge of mixing-length model/TKE model boundary. |
| J | Index for $\phi$-equations (all J loops bypassed if only solving velocit equation). |

| | |
|---|---|
| KASE | Flag to identify if one surface is a wall. |
| KD | Flag to determine how damping coefficient will be determined for Prandtl mixing-length model. |
| K1 | Flag to control print of SP(I) and changes in DELTAX. |
| K2 | Flag to suppress corrections to ALMGG or to use eddy diffusivity model. |
| K3 | Flag to suppress use of internal correlation of turbulent part of PREF(J,I). |
| KENT | Flag to control the entrainment calculation. |
| KERROR | Flag to terminate program if input data error detected. |
| KEX | Type of E-surface. |
| KIN | Type of I-surface. |
| KRAD | Flag to identify if transverse radius effects are to be included in equations. |
| LSUB | Flag to activate the LAMSUB routine in subroutine WALL. |
| LVAR | Flag to prematurely terminate program (e.g., if dimensioning exceeded, negative pressure, etc.). |
| M | Index for boundary condition location. |
| MODE | Flag to signal laminar or turbulent flow. |
| N | Number of initial stream tubes (which requires specification N + 1 initial profile points). |
| NEQ | Number of equations to be solved. |
| NIND | Counter for number of data sets executed. |
| NPH | Number of $\phi$-equations to be solved (NEQ-1). |
| NP1 | N + 1. |
| NP2 | N + 2. |
| NP3 | N + 3. |
| NUMRUN | Number of consecutive data sets to be processed. |
| NXBC | Number of boundary condition locations (X(1) < X(M) < X(NXBC)). |

| | |
|---|---|
| OM(I) | Non-dimensional stream function. |
| OMD(I) | OM(I+1) - OM(I). |
| OUTPUT | Flag to signal type of print format, related to GEOM. |
| PEI | Boundary layer mass flow rate per unit radian (or per unit depth if transverse radius not considered). |
| PO | Initial free-stream static pressure. |
| PPLAG | Lag constant for changing $P^+$, $X^+$, $V_o^+$. |
| PPL | Pressure gradient parameter in wall coordinates ($P^+$). |
| PR(J,I) | Laminar Prandtl number. |
| PRC(J) | Constant property laminar Prandtl/Schmidt number. |
| PRE | Pressure at X = XD. |
| PREF(J,I) | Effective Prandtl number, combining the laminar and turbulent Prandtl numbers. |
| PRO | Pressure at X = XU. |
| PRT(J) | Initial value of turbulent Prandtl number for $\phi$-equation (asymptote if variable turbulent Prandtl number model used). |
| QW(J) | Flux of $\phi$-equation at a wall (positive in positive y-direction). |
| QWF(J) | Flux of $\phi$-equation at a wall/ [F(J,wall) - FI(J)]. |
| R(I) | Transverse radius of finite-difference node at Y(I). |
| RBOM(I) | 1./[OM(I+1) - OM(I-1)]. |
| REH | Enthalpy thickness Reynolds number. |
| REM | Momentum thickness Reynolds number (diameter Reynolds number for pipe flow) |
| RETRAN | Reynolds number for laminar-to-turbulent transition. |
| RHO(I) | Fluid density. |
| RHOC | Constant property fluid density. |
| RHOM | Fluid density at location of FMEAN for pipe flow. |
| ROMD(I) | 1./[OM(I+1) - OM(I)]. |

RW(M) Distance from axis of symmetry to body surface, specified at each X(M).

RWO Wall radius for pipe flows.

SC(I) Diffusion term for velocity equation (small c).

SD Source term at X = XD, not used by program.

SOURCE(J) Type of source function for $\phi$-equation.

SP(I) Special print array, user supplied.

SPACE Print spacing.

ST(J) Wall Stanton number (based on FMEAN if pipe flow).

SU(J,I) Source term for $\phi$-equation.

T(I) Static temperature if stagnation enthalpy equation (FLUID = 2); shear stress if no $\phi$-equations.

TAU Shear stress at join-point location.

TAUW Wall shear stress.

TYPBC(J) Type of boundary condition for $\phi$-equations (level or flux).

U(I) Velocity-dependent variable in momentum equation at Y(I).

UG(M) Free-stream velocity, specified at each X(M), except for pipe/channel flows.

UGD Free-stream velocity at XD, obtained using 3rd order spline fit to UG(M).

UGU Free-stream velocity at XU, obtained using 3rd order spline fit to UG(M).

UI Join-point velocity.

UMAX Maximum U(I) in velocity profile.

UMIN Minimum U(I) in velocity profile.

VISCO(I) Laminar dynamic viscosity.

VISCOC Constant property laminar dynamic viscosity.

X(M) Location along wall (centerline if no wall) where boundary values are given.

XD Downstream value of $x$ where differential equations are solved.

XL Value of $x$ where integration terminated.

XU Value of $x$ where integration begins; during integration the upstrear value of $x$.

Y(I) Independent variable, perpendicular to $x$, measured from I-surface.

YEM Location for (1 - FR) · UMAX.

YIP Location for (1 - FR) · UMIN.

YPMAX Maximum $y^+$ at outer edge of Wall Function.

YPMIN Minimum $y^+$ at outer edge of Wall Function.

## Appendix II

### OUTPUT NOMENCLATURE

| | |
|---|---|
| AME | $\dot{m}''_E$, wall mass flux, Figure 4.1, or entrainment, equation (4.15). |
| AMI | $\dot{m}''_I$, wall mass flux, Figure 4.1. |
| APL | $A^+$, Van Driest damping constant, equation (2.22). |
| BETA | $\beta$, slip constant, equation (4.13a); or $-H * Re_M * K/C_f/2$, acceleration parameter, OUTPUT = 2. |
| BPL | $B^+$, Evans damping constant, equation (2.23). |
| CF2 | $C_F/2$, friction factor, equations (3.11b) or (3.24). |
| EDR | $\mu_{eff}/\mu$, effective/laminar viscosity, equation (2.6). |
| EMU(I) | $\mu_{eff}$, effective viscosity at y location, equation (2.6). |
| F | $\dot{m}''_{(wall)}/\rho U_{(free\ stream)}$, blowing fraction. |
| F(1,I) | dependent variable at y location for first $\phi$-equation. |
| F(2,I) | dependent variable at y location for second $\phi$-equation. |
| F(1,wall) | dependent variable, wall value. |
| FM | $\bar{I}^*$, mean stagnation enthalpy, equation (3.27). |
| FW | $I^*$ at wall, stagnation enthalpy. |
| G | Clauser parameter, $(H-1.)/(H\sqrt{C_f/2})$, OUTPUT = 2. |
| GAMA(J) | $\gamma$, slip constant, equation (4.13b). |
| H | $\delta_1/\delta_2$, shape parameter, equations (3.22a-b). |
| HPLUS(I) | $I^{*+}$ at y location, equation (3.4i). |
| I | y location. |
| INTG | integration number. |
| K | acceleration parameter, $(\nu/U_\infty^2)dU_\infty/dx$. |
| NU | Nu, Nusselt number, equation (3.26). |
| OM(I) | $\omega$, equation (4.4). |

| | |
|---|---|
| PEI | $(\psi_E-\psi_I)$, boundary layer mass flow rate/radian (on unit depth), equation (4.4). |
| PPLUS | $P^+$, pressure gradient parameter, equation (3.4g). |
| PRESS, PRESSURE | fluid thermodynamic pressure. |
| QWALL | $\dot{q}_o''$, wall heat flux, equation (3.15a). |
| R(I) | r, radius at y location. |
| RE | Reynolds number, equation (3.29), OUTPUT = 4. |
| REM | $Re_M$, momentum thickness Reynolds number, $\delta_2 U_\infty/\nu$, equation (3.22b |
| REH | $Re_H$, enthalpy thickness Reynolds number, $\Delta_2 U_\infty/\nu$, equation (3.22c |
| RHO(1) | $\rho$, fluid density at I-surface. |
| RHO(NP3) | $\rho$, fluid density at E-surface. |
| SP(I) | special output array, user supplied. |
| SQRT(K)/UG | $\sqrt{q^2/2}/U_\infty$, turbulent kinetic energy equation. |
| ST(J) | St, Stanton number, equation (3.15b). |
| T(I) | static temperature, degrees Rankine; or $\tau^+$, equation (3.4f), if NEQ = 1 and OUTPUT = 2. |
| TAUPLUS | $\tau^+$, equation (3.4f). |
| TAUWALL | $\tau$, wall shear stress, equation (3.11a). |
| U(I) | U, velocity at y location. |
| UGU | $U_\infty$ at print location. |
| UM | $\overline{U}$, mean velocity equation (3.28). |
| UPL, UPLUS(I) | $U^+$ at y location, equation (3.4b). |
| VWPLUS | $V_o^+$, equation (3.4c). |
| XU | x at print location. |
| Y(I) | y, dependent variable location. |
| YPL, YPLUS(I) | $y^+$ at y location, equation (3.4e). |

Appendix III

STAN5 PROGRAM

```
C.....TURBULENT BOUNCARY LAYER PREDICTION---PATANKAR/SPALDING METHOD,
C.....KAYS/STANFORC VERSION, DESIGNATION STAN5, DECEMBER, 1975
C.....
C.....IN THIS VERSICN ECDY VISCOSITY AND THE EDDY CONDUCTIVITIES ARE
C.....CALCULATED EITHER BY THE MIXING-LENGTH METHOD OR FROM SOLUTION
C.....CF THE TURBULENT KINETIC ENERGY EQUATION.  IF THE LATTER METHOD IS
C.....TO BE USED IT IS MERELY NECESSARY TO ACTIVATE ONE ADDITIONAL DIFF-
C.....USION EQUATICN IN THE INPUT ROUTINE, AND TO SET SOURCE = 2 FOR
C.....THAT EQUATION.  THE SHIFT OF METHOD IS THEN AUTOMATIC.  NOTE THAT
C.....THE PROGRAM IS SET UP ONLY FOR A ZERO OR A ONE-EQUATION MODEL OF
C.....TURBULENCE.  ADDITIONAL THOUGH MINOR MODIFICATION IS NECESSARY FOR
C.....MULTI-EQUATICN MODELS.
C.....THERE IS ALSC AN CPTION FOR WHICH EDDY DIFFUSIVITY IN THE OUTER
C.....REGION OF THE BOUNDARY LAYER IS EVALUATED DIRECTLY FROM A REYNOLDS
C.....NUMBER CORRELATICN.
C.....
      INTEGER GEOM,FLUIC,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      COMMON/GEN/PEI,AMI,AME,DPDX,XU,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,NXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SCURCE,RETRAN,NUMRUN,SPACE,RWD,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,OWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     6/W/SC(54),AU(54),BU(54),CU(54),A(5,54),B(5,54),C(5,54),SU(5,54),SD
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     8/L1/YL,UMAX,UMIN,FR,YIP,YEM,ENFRA,KENT,AUXM2
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHOM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,OW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AUX1(100),AUX2(100),YPMAX
     3/ADD/RBOM(54),OMC(54),ROMD(54),ITKE
      DIMENSION AMEF(5),AMIF(5)
      DIMENSION FPP(100),AFPP(100),BFPP(100),CFPP(100)
C.....
C.....PROGRAM DIMENSIONED FOR 50 FLOW TUBES
C.....IF DIMENSIONING CHANGED, SEE *** CARDS IN SUBS WALL AND OUT
C.....
      NUMRUN=1
      NIND=0
    5 KERROR=0
      NIND=NIND+1
C ------------------------------------------------------------ INPUT ----
      CALL INPUT(KERROR)
      IF(KERROR.GT.0)GC TO  1000
C ------------------------------------------------------------ STEP5 ----
      CALL STEP(5)
      M=1
    6 M=M+1
      IF (XU.GT.X(M)) GO TO 6
      AMIE=AM(M-1)+(AM(M)-AM(M-1))*(XU-X(M-1))/(X(M)-X(M-1))
      AUXM1=AUX1(M-1)+(AUX1(M)-AUX1(M-1))*(XU-X(M-1))/(X(M)-X(M-1))
      AUXM2=AUX2(M-1)+(AUX2(M)-AUX2(M-1))*(XU-X(M-1))/(X(M)-X(M-1))
      IF (KEX.EQ.1) AME=AMIE
      IF (KIN.EQ.1) AMI=AMIE
C ------------------------------------------------------------ STEP1 ----
   10 CALL STEP(1)
   15 CONTINUE
C---------------------------------------------- FLUID PROPERTIES ----
C.....FLUID PROPERTIES EITHER SET EQUAL TO INPUT DATA
C.....OR COMPUTED BY CALLING A VARIABLE PROPERTIES
```

```
C.....SUBROUTINE SUPPLIED BY USER.
      IF(FLUID.NE.1) GO TO 115
      IF(INTG.GT.0) GO TO 205
      DO 110 I=1,NP3
      VISCO(I)=VISCOC
      RHO(I)=RHOC
      T(I)=1.
      IF (NPH.EQ.0) GO TO 110
      DO 105 J=1,NPH
      PR(J,I)=PRC(J)
  105 CONTINUE
  110 CONTINUE
      GO TO 205
  115 DO 130 I=1,NP3
      IPROP=FLUID-1
      GO TO (120,122,124),IPROP
  120 J=1
      IF(SOURCE(J).EQ.2)J=2
      CALL PROP2(I,F(J,I),T(I),VISCO(I),PR(1,I),RHO(I))
      IF (LVAR.EQ.7) GC TO 1000
      GO TO 130
C.....CALLS FOR OTHER PROPERTY SUBROUTINES CAN BE INSERTED HERE.  IT IS
C.....ALSO NECESSARY TO CHANGE PROPERTY CALLS IN SUBROUTINE WALL.
  122 CONTINUE
  124 CONTINUE
  130 CONTINUE
C--------------------------------------------------------- WALL RADIUS ----
C.....WALL RADIUS AT EACH X LOCATION EVALUATED BY
C.....BY LINEAR INTERPCLATION OF INPUT DATA
  205 IF (LSUB.GT.0) GC TO 35
      R12=RW(M)
      R11=RW(M-1)
      X2=X(M)
      X1=X(M-1)
      RUU=R11+(R12-R11)*(XU-X1)/(X2-X1)
      IF(GEOM.EQ.7)GO TC 225
      IF(GEOM.EQ.9)GC TC 230
      RWO=RUU
      CSALFA=(SQRT(ABS((X2-X1)*(X2-X1)-(R12-R11)*(R12-R11))))/(X2-X1)
      IF(KIN.EQ.2.ANC.KRAD.EQ.1)RUU=RUU-Y(NP3)*CSALFA
      IF(GEOM.EQ.4.OR.GEOM.EQ.6)GO TO 220
      GO TO 30
  220 CSALFA=1.00
      RUU=0.0
      GO TO 30
  225 CSALFA=1.00
      IF(INTG.EQ.0)PI=0.5*RUU*RUU*U(1)*RHO(1)
      IF(INTG.EQ.0)GO TC 30
      PI=PI-AMI*RUU*CX
      IF(PI.LE.0.0)RUU=0.0
      IF(PI.LE.0.0)GEOM=6
      IF(GEOM.EQ.6)KIN=3
      IF(GEOM.EQ.6)WRITE(6,228)
      IF(GEOM.EQ.6)GO TC 30
      RUU=SQRT(PI*2./(U(1)*RHO(1)))
      GO TO 30
  230 CSALFA=1.00
      IF(INTG.EQ.0)EN=0.0
      IF(INTG.EQ.0)RWO=0.0
      IF(INTG.EQ.0)GO TO 30
```

```
      EN=EN+AMI*RUU*CX
      RWD=EN/(RUU*U(1)*RHO(1))
   30 R(1)=RUU
C ------------------------------------------------------------ STEP2 ----
   35 IF (INTG.EQ.0.OR.LSUB.GT.0) CALL STEP(2)
      IF (LSUB.GT.0) GO TO 58
C ------------------------------------------------------------ STEP3 ----
      CALL STEP(3)
C-------------------------------------------------- ENTRAINMENT CONTROL ----
      IF (GEOM.EQ.4.OR.GEOM.EQ.5) GO TO 340
      IF (INTG.EQ.0) GO TO 345
      UMM=UMAX-UMIN
      PEIE=PEI/(R(NP3)*Y(NP3))
      PEII=PEIE*R(NP3)/R(1)
      UDIFF=ENFRA*UMM
      UACTI=ABS(U(3)-U(1))/UMM
      UACTE=ABS(U(NP3)-U(NP1))/UMM
      AMEN=AME + (ENFRA-UACTE)*PEIE
      IF(ENFRA.GT.2.*UACTE)AMEN=AME/2.
      IF(ABS(U(NP1)-U(N)).LE.UDIFF/2.)AMEN=AME/2.
      AMIN=0.
      IF (KIN.EQ.2) AMIN=AMI - (ENFRA-UACTI)*PEII
      IF(ENFRA.GT.2.*UACTI)AMIN=AMI/2.
      IF(ABS(U(3)-U(4)).LE.UDIFF/2.)AMIN=AMI/2.
      IF(NPH.EQ.0)GO TO 330
      IF(KENT.EQ.0)GO TO 330
      DO 312 J=1,NPH
      FMAX=F(J,1)
      FMIN=F(J,1)
      DO 305 I=3,NP3
      IF(I.EQ.NP2)GO TO 305
      IF(F(J,I).GT.FMAX)FMAX=F(J,I)
      IF(F(J,I).LT.FMIN)FMIN=F(J,I)
      IF(SOURCE(J).EQ.2)FMAX=1.
  305 CONTINUE
      FMM=FMAX-FMIN
      FDIFF=ENFRA*FMM
      IF(FMM.LT.0.1)GO TO 310
      FACTI=ABS(F(J,1)-F(J,3))/FMM
      FACTE=ABS(F(J,NP3)-F(J,NP1))/FMM
      AMEF(J)=AME + (ENFRA-FACTE)*PEIE
      AMIF(J)=0.
      IF (KIN.EQ.2) AMIF(J)=AMI - (ENFRA-FACTI)*PEII
  310 CONTINUE
      IF(INDI(J).EQ.2.AND.ABS(AJI(J)).LT..0001)AMEF(J)=0.0
      IF(INDE(J).EQ.2.AND.ABS(AJE(J)).LT..0001)AMIF(J)=0.0
      IF(FMM.LT.0.1)AMEF(J)=0.0
      IF(FMM.LT.0.1)AMIF(J)=0.0
  312 CONTINUE
      DO 325 J=1,NPH
      IF(J.GT.1)GO TO 315
      AMEMAX=AMEN
      IF(KIN.EQ.2)AMIMAX=AMIN
  315 IF(SOURCE(J).EQ.2)GO TO 320
      IF(-AMEF(J).GT.-AMEN)AMEMAX=AMEF(J)
      IF(KIN.EQ.2.AND.AMIF(J).GT.AMIN)AMIMAX=AMIF(J)
  320 IF(KEX.EQ.2)AME=AMEMAX
      IF(KIN.EQ.2)AMI=AMIMAX
  325 CONTINUE
      GO TO 335
```

```
  330 IF(KEX.EQ.2)AME=AMEN
      IF(KIN.EQ.2)AMI=AMIN
  335 IF(KEX.EQ.2.AND.AME.GT.0.)AME=0.
      IF(KIN.EQ.2.AND.AMI.LT.0.)AMI=0.
      AMMAX=PEIE*0.1
      IF(KEX.EQ.2.AND.AME.LT.-AMMAX)AME=-AMMAX
      IF(KIN.EQ.2.AND.AMI.GT.AMMAX)AMI=AMMAX
  340 IF(KIN.EQ.3)AMI=0.
      IF(KEX.EQ.3)AME=0.
C ------------------------------------------------------ DX STEPSIZE ----
      IF(K1.EQ.20.OR.K1.EQ.9)DELTAX=AUXM1
  345 IF(AMI - 0.0)42,40,42
   40 IF(AME - 0.0)42,44,42
   42 DX=FRA*PEI/(ABS(R(1)*AMI-R(NP3)*AME))
      IF(DX.GT.DELTAX*YL)DX=DELTAX*YL
      GO TO 46
   44 DX = DELTAX*YL
   46 IF(INTG.EQ.0)GO TO 49
      IF(DX.GT.20.*DXOLD)WRITE(6,48)
   49 IF(INTG.LT.10)DX=0.2*DX
      INTG=INTG+1
      XD = XU + DX
      DXOLD=DX
      IF(REM.GT.RETRAN)MODE=2
      IF (XD.GE.(X(NXBC)-1.5*DX)) XL=X(NXBC)-1.5*DX
      IF(XD.GT.X(NXBC))PRO=PRE
      IF(XD.GT.X(NXBC)) GO TO 55
C------------------------------------------------------- BODY FORCE ----
C..... BODY FORCE(OTHER THAN PRESSURE GRADIENT), SUCH AS
C..... BOUYANCY OR A BODY FORCE PER UNIT VOLUME, FOR THE
C..... MOMENTUM EQUATION. POSITIVE IN THE POSITIVE X-DIR.
      IF (BODFOR.EQ.0) GO TO 410
      DO 405 I=1,NP3
      BF(I)=GV*RHO(I)/GC
      IF(BODFOR.EQ.2)BF(I)=BF(I)+AUXM1
  405 CONTINUE
      BFG=BF(1)
      IF (KEX.EQ.2) BFG=BF(NP3)
C---------------------------- PRESSURE GRADIENT - EXTERNAL FLOW ----
C.....PRESSURE GRADIENT FOR EXTERNAL FLOW COMPUTED BY
C.....FITTING A 3RD ORDER SPLINE-FIT TO FREE-STREAM VELOCITY
C.....INTRODUCED IN THE INPUT DATA.
  410 IF (INTG.EQ.1) DPSUM=0.0
      IF (GEOM.EQ.4)GO TO 435
      IF(GEOM.EQ.5)GO TO 440
      IF (INTG.NE.1) GO TO 415
      M=1
      IF(KEX.EQ.2)UGUIDE=U(NP3)
      IF(KIN.EQ.2.AND.KEX.NE.2)UGUIDE=U(1)
      RHOLD=RHO(1)
      IF (KEX.EQ.2) RHOLD=RHO(NP3)
      RHO2=RHOLD
      BFG=BF(1)
      IF (KEX.EQ.2) BFG=BF(NP3)
      FPP(1)=0.
      FPP(NXBC)=0.
      NXBCM1=NXBC-1
      IF (NXBC.EQ.2) GO TO 425
      DELI=X(2)-X(1)
      DO 411 I=2,NXBCM1
```

```
      DELM=DELI
      DELI=X(I+1)-X(I)
      DELSUM=DELI+DELM
      AFPP(I)=-DELI/DELSUM/2.
      BFPP(I)=-DELM/DELSUM/2.
  411 CFPP(I)=3.*(UG(I-1)*DELI-UG(I)*DELSUM+UG(I+1)*DELM)/(DELI*DELM
     1*DELSUM)
      BFPP(2)=BFPP(2)*FPP(1)+CFPP(2)
      IF (NXBC.EQ.3) GO TO 414
      DO 412 I=3,NXBCM1
      TRATIO=1./(1.-BFPP(I)*AFPP(I-1))
      AFPP(I)=AFPP(I)*TRATIO
  412 BFPP(I)=(BFPP(I)*BFPP(I-1)+CFPP(I))*TRATIO
  414 DO 413 I=2,NXBCM1
      J=NXBCM1-I+2
  413 FPP(J)=AFPP(J)*FPP(J+1)+BFPP(J)
      GO TO 425
C.....ADJUSTMENT OF PRESSURE FOR CORRECT DENSITY
  415 IF (FLUID.EQ.1) GO TO 425
      RHO2=RHO(1)
      IF (KEX.EQ.2) RHO2=RHO(NP3)
      DPDXN=DPDX*RHO2/RHOLD
      PRE=PRE+(DPDXN-DPDX)*0.5*DX/GC
      RHOLD=RHO2
      GO TO 425
  418 M=M+1
      DXM=X(M)-X(M-1)
      AA=0.1666666*FPP(M-1)/DXM
      BB=0.1666666*FPP(M)/DXM
      CC=UG(M-1)/DXM-0.1666666*DXM*FPP(M-1)
      D=UG(M)/DXM-0.1666666*DXM*FPP(M)
  425 CONTINUE
      PRO=PRE
      IF (XD.GT.X(M)) GO TO 418
      IF (INTG.EQ.1) UGD=AA*(X(M)-XU)*(X(M)-XU)*(X(M)-XU)
     1+BB*(XU-X(M-1))*(XU-X(M-1))*(XU-X(M-1))+CC*(X(M)-XU)+D*(XU-X(M-1))
      IF(INTG.EQ.1.AND.ABS(UGUIDE-UGD).GT.0.001*UGUIDE)LVAR=5
      UGU=UGD
      XMXD=X(M)-XD
      XDMX=XD-X(M-1)
      UGD=AA*XMXD*XMXD*XMXD+BB*XDMX*XDMX*XDMX+CC*XMXD+D*XDMX
C.....IF IT IS DESIRED TO INTRODUCE THE FREE-STREAM VELOCITY AS AN
C.....ANALYTIC FUNCTION, RATHER THAN A TABULATION, THIS IS THE PLACE
C.....TO PUT IT IN.  IT THEN OVERRIDES THE PRECEDING STATEMENT.
C.....THE FOLLOWING IS SPECIAL FOR EQUILIBRIUM ADVERSE PG FLOWS.
C.....IF(XD.GT.BXX.AND.K1.EQ.15)UGD=AXX*((XD-CXX)/(BXX-CXX))**DXX
      DPDX = (UGU+UGD)*(UGU-UGD)*0.5*RHO2/(XD-XU)+ GC*BFG
C.....NOTE THAT THIS PRESSURE GRADIENT IS ACTUALLY THE TRUE
C.....PRESSURE GRADIENT TIMES G SUB C
      DPSUM=DPDX*DX+DPSUM
      PRE =PO+DPSUM/GC
C.....THIS PRESSURE IS USED ONLY IN THE PROPERTY SUBROUTINE, AND THERE
C.....ONLY TO EVALUATE DENSITY.
      GO TO 451
C--------------------------------- PRESSURE GRADIENT - INTERNAL FLOW ----
  435 CONTINUE
      PRO=PRE
      IF(INTG.EQ.1)RHOM=RHO(1)
      UGU=2.*PEI/(R(NP3)*R(NP3)*RHOM   )
      DPDX=-PEI*UGU*(R(NP3)-RWO)/(RWO*RWO*RWO*DX)-2.*CF2*RHOM*
```

```
     1UGU*UGU/R(NP3)+GC*BF(1)
      DPSUM=DPDX*DX+DPSUM
      PRE =PO+DPSUM/GC
      GO TO 445
  440 CONTINUE
      PRO=PRE
      IF(INTG.EQ.1)RHOM=RHO(1)
      UGU=PEI/(R(NP3)*Y(NP3)*RHOM)
      DPDX=-PEI*UGU*(Y(NP3)-RWO)/(2.*RWO*RWO*RWO*DX)-CF2*RHOM*UGU*UGU/
     1Y(NP3)+GC*BF(1)
      DPSUM=DPDX*DX+DPSUM
      PRE =PO+DPSUM/GC
  445 IF (XD.LE.X(M)) GO TO 451
      M=M+1
      GO TO 445
  451 CONTINUE
      IF(LVAR.EQ.5) GO TO 1020
      IF (PRE.LT.0.) LVAR=4
      IF (LVAR.EQ.4) GO TO 1010
   55 IF(KASE.EQ.2) GO TO 65
C ------------------------------------------------------------ WALL ----
   58 CALL WALL
      IF (LVAR.EQ.6) GO TO 1000
      IF(N.LT.12) GO TO 1030
      IF(LSUB.GT.0)GO TO 10
C---------------------------- INTEGRAL PARAMETERS - EXTERNAL FLOW ----
C.....CALCULATION OF   DISPLACEMENT THICKNESS,MOMENTUM THICKNESS,
C.....SHAPE FACTOR, MOMENTUM THICKNESS REYNOLDS NUMBER, AND THE ENTHALPY
C.....THICKNESS REYNOLDS NUMBER.
      GO TO (500,500,500,538,538,560,560,560,560), GEOM
  500 VISG=VISCO(NP3)
      RHG=RHO(NP3)
      IF(KIN.EQ.1)GO TO 505
      RHG=RHO(1)
      VISG=VISCO(1)
  505 IF(UGU.LT.0.001)GO TO 520
      SUM1=0.
      DO 510 I=2,NP3
  510 SUM1=SUM1+(Y(I)-Y(I-1))*(R(I)+R(I-1))/2.
      DEL1=SUM1/RWO-PEI/(RWO*RHG*UGU)
      SUM=0.
      DO 515 I=3,NP2
  515 SUM=SUM+(1.-(U(I)+U(I-1))/(2.*UGU))*OMD(I-1)
      DEL2=PEI*SUM/(RWO*RHG*UGU)
C.....CORRECTION OF INTEGRAL PARAMETERS FOR TRANSVERSE CURVATURE
      IF (KRAD.NE.1) GO TO 519
      IF (KIN.EQ.2) GO TO 517
      DEL1=RWO*(-1.+SQRT(1.+2.*CSALFA*DEL1/RWO))/CSALFA
      DEL2=RWO*(-1.+SQRT(1.+2.*CSALFA*DEL2/RWO))/CSALFA
      GO TO 519
  517 DEL1=RWO*(+1.-SQRT(1.-2.*CSALFA*DEL1/RWO))/CSALFA
      DEL2=RWO*(+1.-SQRT(1.-2.*CSALFA*DEL2/RWO))/CSALFA
  519 CONTINUE
      H=DEL1/DEL2
      REM=DEL2*UGU*RHG/VISG
  520 CONTINUE
      IF(NPH.EQ.0) GO TO 560
      REH=0.
      IF(SOURCE(1).EQ.2.AND.NPH.EQ.1)GO TO 535
      J=1
```

```
      IF(SOURCE(1).EQ.2)J=2
      FW=F(J,1)
      FG=F(J,NP3)
      IF(KIN.EQ.1)GO TO 525
      FW=F(J,NP3)
      FG=F(J,1)
  525 SUM=0.
      DO 530 I=3,NP2
      DEL=OMD(I-1)*0.5*(F(J,I)+F(J,I-1)-2.*FG)
  530 SUM=SUM+DEL
      IF(ABS(UGU*(FW-FG)).LT.0.00001)GO TO 535
      DEL3=PEI*SUM/(RWC*RHG*UGU*(FW-FG))
      IF (KRAD.NE.1) GO TO 534
      IF (KIN.EQ.2) GO TO 532
      DEL3=RWO*(-1.+SQRT(1.+2.*CSALFA*DEL3/RWO))/CSALFA
      GO TO 534
  532 DEL3=RWO*(+1.-SQRT(1.-2.*CSALFA*DEL3/RWO))/CSALFA
  534 REH=DEL3*UGU*RHG/VISG
  535 CONTINUE
      GO TO 560
C---------------------------- INTEGRAL PARAMETERS - INTERNAL FLOW ----
  538 FMEAN=0.0
      VISCOM=VISCO(1)
      RHOM=RHO(1)
      IF(NEQ.EQ.1)GO TO 555
      IF(SOURCE(1).EQ.2.AND.NPH.EQ.1)GO TO 555
      J=1
      IF(SOURCE(1).EQ.2)J=2
      IF(ABS((F(J,1)-F(J,NP3))).LT..0001)ST(J)=0.0
      IF(ABS((F(J,1)-F(J,NP3))).LT..0001)GO TO 555
      DO 540 I=3,NP2
      DEL=(OM(I)-OM(I-1))*(F(J,I)+F(J,I-1))/2.
  540 FMEAN=FMEAN+DEL
      MM=0
  545 MM=MM+1
      RATIO=1.0
      IF(INTG.EQ.1)GO TO 550
      IF(MM.EQ.NP2)GO TO 550
      IF(ABS(F(J,MM)-FMEAN).GT.ABS(F(J,MM)-F(J,MM+1)))GO TO 545
      RATIO=(F(J,MM)-FMEAN)/(F(J,MM)-F(J,MM+1))
  550 RHOM=RHO(MM)-(RHO(MM)-RHO(MM+1))*RATIO
      VISCOM=VISCO(MM)-(VISCO(MM)-VISCO(MM+1))*RATIO
C.....STANTON NUMBER AND CF2 ARE CALCULATED HERE BY SIMPLY
C.....MODIFYING THE VALUES CALCULATED IN THE WALL FUNCTION
C.....WHERE FREE-STREAM U AND F ARE USED.
      ST(J)=ST(J)*(RHO(1)/RHOM)*(F(J,1)-F(J,NP3))/(FMEAN-F(J,NP3))
  555 REM=4.*PEI/(Y(NP3)*VISCOM)
      CF2=CF2*RHO(1)/(RHOM)
  560 CONTINUE
   65 CONTINUE
C ---------------------------------------------------------- AUX ----
      CALL AUX
C ---------------------------------------------------------- OUT ----
      CALL OUT
      IF(LVAR.GT.1)GO TO  1000
C.....  THE TERMINATION CONDITION
      IF(XD.GT.XL) GO TO  1000
      IF (KASE.EQ.2) GO TO 70
C---------------------------------------------- WALL MASS TRANSFER ----
C.....LINEAR INTERPOLATION OF WALL MASS TRANSFER DATA,
```

```
C.....BODY  FORCE SOURCE DATA, AND ENERGY SOURCE DATA
C.....FROM INPUT DATA.
      AMIE=AM(M-1)+(AM(M)-AM(M-1))*(XD-X(M-1))/(X(M)-X(M-1))
      IF (KEX.EQ.1) AME=AMIE
      IF (KIN.EQ.1) AMI=AMIE
      AUXM1=AUX1(M-1)+(AUX1(M)-AUX1(M-1))*(XD-X(M-1))/(X(M)-X(M-1))
      AUXM2=AUX2(M-1)+(AUX2(M)-AUX2(M-1))*(XD-X(M-1))/(X(M)-X(M-1))
C.....IF IT IS DESIRED TO INTRODUCE THE WALL BOUNDARY CONDITION AS
C.....AN ANALYTIC FUNCTION, RATHER THAN A TABULATION, THIS IS THE
C.....PLACE TO PUT IT IN.  IT THEN OVER-RIDES THE PRECEDING STATEMENT.
C-------------------------------- WALL F BOUNDARY CONDITION ----
C.....LINEAR INTERPOLATION OF WALL BOUNDARY DATA
C.....FROM INPUT DATA
C.....
      IF (NEQ.EQ.1) GO TO 70
      DO 710 J=1,NPH
      FQ = FJ(J,M-1) + (FJ(J,M)-FJ(J,M-1))*(XD-X(M-1))/(X(M)-X(M-1))
C.....NOTE THAT FQ  IS EITHER A SURFACE PROPERTY, SUCH AS ENTHALPY,
C.....OR IT IS A SURFACE FLUX, SUCH AS HEAT FLUX.
C.....IF IT IS DESIRED TO INTRODUCE THE WALL BOUNDARY CONDITION AS
C.....AN ANALYTIC FUNCTION, RATHER THAN A TABULATION, THIS IS THE
C.....PLACE TO PUT IT IN.  IT THEN OVER-RIDES THE PRECEDING STATEMENT.
      GO TO (701,704,704), KIN
  701 NINDJ=INDI(J)
      GO TO (702,703), NINDJ
  702 F(J,1)=FQ
      GO TO 704
  703 AJI(J)=FQ
  704 GO TO (705,708,708), KEX
  705 NINDJ=INDE(J)
      GO TO (706,707), NINDJ
  706 F(J,NP3)=FQ
      GO TO 708
  707 AJE(J)=FQ
  708 CONTINUE
  710 CONTINUE
C ------------------------------------------------------ STEP4 ----
   70 CALL STEP(4)
      XU=XD
      PEI=PEI+DX*(R(1)*AMI-R(NP3)*AME)
      GO TO 15
 1000 IF(NIND.LT.NUMRUN)GO TO 5
      RETURN
 1010 WRITE (6,455)
      GO TO 1000
 1020 WRITE (6,450)
      GO TO 1000
 1030 WRITE (6,64)
      GO TO 1000
  228 FORMAT(//' PROGRAM HAS SHIFTED TO GEOM=6 '//)
   64 FORMAT(//' PROGRAM TERMINATED BECAUSE THE NUMBER OF SPACES'/
     1' IN THE GRID (N) GOT BELOW 12.  ADD MORE GRID POINTS TO'/
     2' THE OUTER PART OF THE BOUNDARY LAYER'//)
  450 FORMAT(//' PROGRAM MAY HAVE TERMINATED BECAUSE INITIAL VELOCITY'/
     1' PROFILE IS INCOMPATIBLE AT EITHER Y(NP3) OR Y(1) WITH'/
     2' THE INPUT FREE-STREAM VELOCITY PROFILE'//)
  455 FORMAT (/10X,'PRESSURE HAS GONE NEGATIVE,PROGRAM TERMINATED')
   48 FORMAT(//' DX HAS TAKEN A LARGE STEP FORWARD WHICH LOOKS'/
     1' LIKE NOTHING BUT TROUBLE.  PERHAPS Y(NP3) HAS BLOWN UP.'//)
      END
```

```
      SUBROUTINE STEP(KSTEP)
C.....
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      COMMON/GEN/PEI,API,AME,DPDX,XU,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,KXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SOURCE,RETRAN,NUMRUN,SPACE,RWD,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,QWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     6/W/SC(54),AU(54),BU(54),CU(54),A(5,54),B(5,54),C(5,54),SU(5,54),SD
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     8/L1/YL,UMAX,UMIN,FR,YIP,YEM,ENFRA,KENT,AUXM2
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHOM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AUX1(100),AUX2(100),YPMAX
     3/ADD/RBOM(54),CMC(54),ROMD(54),ITKE
C.....
      GO TO (100,200,300,400,500), KSTEP
C.....
C------------------------------------------------------------- STEP 1 ----
C.....STEP1 COMPUTES SLIP POINT QUANTITIES,BETA,GAMA(J) AT
C.....BEGINNING OF INTEGRATION OR AFTER LAMSUB HAS
C.....BEEN ACTIVATED.
C.....
  100 GO TO (22,24,26),KIN
   22 IF(LSUB.GT.0)GO TO 23
      BETA=(U(4)/U(3)-1.)/(Y(4)/Y(3)-1.)
   23 U(2)=U(3)/(1.+2.*BETA)
      Y(2)=Y(3)*BETA/(2.+BETA)
      GO TO 30
C.....    FREE BOUNDARY
   24 U11=U(1)*U(1)
      U13=U(1)*U(3)
      U33=U(3)*U(3)
      SQ=84.*U11-12.*U13+9.*U33
      U(2)=(16.*U11-4.*U13+U33)/(2.*(U(1)+U(3))+SQRT(SQ))
      Y(2)=Y(3)*(U(2)+U(3)-2.*U(1))*.5/(U(2)+U(3)+U(1))
      GO TO 30
   26 IF(KRAD.NE.0) GO TO 28
C.....    SYMMETRY LINE, PLANE FLOW
      U(2)=(4.*U(1)-U(3))/3.
      Y(2)=0.
      GO TO 30
C.....    SYMMETRY LINE, AXIALLY SYMMETRICAL FLOW
   28 U(2)=U(1)
      Y(2)=Y(3)/3.
C.....    SAME AS ABOVE BUT FOR KEX
   30 GO TO (32,34,36),KEX
C.....    WALL
   32 IF(LSUB.GT.0)GO TO 33
      BETA=(U(N)/U(NP1)-1.)/((Y(NP3)-Y(N))/(Y(NP3)-Y(NP1))-1.)
   33 U(NP2)=U(NP1)/(1.+2.*BETA)
      Y(NP2)=Y(NP3)-(Y(NP3)-Y(NP1))*BETA/(2.+BETA)
      GO TO 38
   34 U11=U(NP1)*U(NP1)
      U13=U(NP1)*U(NP3)
      U33=U(NP3)*U(NP3)
      SQ=84.*U33-12.*U13+9.*U11
C.....    FREE BOUNDARY
      U(NP2) =(16.*U33-4.*U13+U11)/(2.*(U(NP1)+U(NP3))+SQRT(SQ))
```

```
      Y(NP2)=Y(NP3)-(Y(NP3)-Y(NP1))*(U(NP2)+U(NP1)-2.*U(NP3))*.5/
     1(U(NP2)+U(NP1)+U(NP3))
      GO TO 38
C.....     SYMMETRY LINE
   36 U(NP2)=(4.*U(NP3)-U(NP1))/3.
      Y(NP2)=Y(NP3)
   38 CONTINUE
      IF(NEQ.EQ.1) GO TO 58
C.....CALCULATION OF SLIP VALUES FOR OTHER DEPENDENT VARIABLES
      DO 56 J=1,NPH
      IF(LSUB.EQ.0)GAMA(J)=0.0
      GO TO (42,44,46),KIN
   42 IF(SOURCE(J).EQ.2.OR.ABS(F(J,3)-F(J,1)).LE..001)GO TO 43
      IF(LSUB.GT.0)GO TO 43
      GAMA(J)=((F(J,4)-F(J,1))/(F(J,3)-F(J,1))-1.)/(Y(4)/Y(3)-1.)
   43 F(J,2)=F(J,1)+(F(J,3)-F(J,1))*(1.+BETA-GAMA(J))/(1.+BETA+GAMA(J))
      GO TO 48
   44 G=(U(2)+U(3)-8.*U(1))/(5.*(U(2)+U(3))+8.*U(1))
      GF=(1.-PRT(J))/(1.+PRT(J))
      GF = (G+GF)/(1.+G*GF)
      F(J,2)=F(J,3)*GF+(1.-GF)*F(J,1)
      GO TO 48
   46 F(J,2)=F(J,1)
      IF(KRAD.EQ.0)F(J,2)=(4.*F(J,1)-F(J,3))/3.
   48 GO TO (50,52,54),KEX
   50 IF(SOURCE(J).EQ.2.OR.ABS(F(J,NP1)-F(J,NP3)).LE..001)GO TO 51
      IF(LSUB.GT.0)GO TO 51
      GAMA(J)=((F(J,N)-F(J,NP3))/(F(J,NP1)-F(J,NP3))-1.)/(((Y(NP3)-Y(N))/
     1(Y(NP3)-Y(NP1))-1.)
   51 F(J,NP2)=F(J,NP3)+(F(J,NP1)-F(J,NP3))*(1.+BETA-GAMA(J))/(1.+BETA+
     1GAMA(J))
      GO TO 56
   52 G=(U(NP2)+U(NP1)-8.*U(NP3))/(5.*(U(NP2)+U(NP1))+8.*U(NP3))
      GF=(1.-PRT(J))/(1.+PRT(J))
      GF = (G+GF)/(1.+G*GF)
      F(J,NP2)=F(J,NP1)*GF+(1.-GF)*F(J,NP3)
      GO TO 56
   54 F(J,NP2)=(4.*F(J,NP3)-F(J,NP1))/3.
   56 CONTINUE
   58 CONTINUE
      RETURN
C-------------------------------------------------------- STEP 2 ----
C.....STEP2 COMPUTES OM(I) ARRAYS AND PEI
C.....AT BEGINNING OF INTEGRATION OR AFTER LAMSUB HAS
C.....BEEN ACTIVATED.
C.....
C.....CALCULATION OF RADII
  200 IF(KRAD.EQ.0) GO TO 220
      DO 210 I=2,NP3
  210 R(I)=R(1)+Y(I)*CSALFA
      GO TO 240
  220 DO 230 I=2,NP3
C.....     R(1) CANNOT EQUAL ZERO
  230 R(I)=R(1)
  240 CONTINUE
C.....CALCULATION OF OMEGA VALUES.  THESE VALUES ARE ESTABLISHED BY THE
C.....INITIAL VELOCITY PROFILE AND REMAIN UNCHANGED THEREAFTER.
      OM(1)=0.0
      OM(2)=0.0
      DO 250 I=3,NP2
```

```
C.....     FOLLOWING EQUATION RESULTS FROM CONTINUITY                    STEP1200
  250 OM(I)=OM(I-1)+.5*(RHO(I)*U(I)*R(I)+RHO(I-1)*U(I-1)*R(I-1))*         STEP1210
     1(Y(I)-Y(I-1))                                                      STEP1220
      PEI=OM(NP2)                                                        STEP1230
C.....     OMEGA IS NORMALIZED AT THIS POINT                             STEP1240
      DO 260 I=3,NP1                                                     STEP1250
  260 OM(I)=OM(I)/PEI                                                    STEP1260
      OM(NP2)=1.0                                                        STEP1270
      OM(NP3)=1.0                                                        STEP1280
      DO 270 I=2,NP1                                                     STEP1290
      RBOM(I)=1./(OM(I+1)-OM(I-1))                                       STEP1300
      OMD(I)=OM(I+1)-OM(I)                                               STEP1310
      ROMD(I)=1./OMD(I)                                                  STEP1320
  270 CONTINUE                                                           STEP1330
      IF(LSUB.GT.0)RETURN                                                STEP1340
      DO 280 I = 5,NP1                                                   STEP1350
      KXERR = 0                                                          STEP1360
      OMRAT = (OM(I) -OM(I-1))/(OM(I-1) - OM(I-2))                       STEP1370
      IF(OMRAT.GT.4.0.OR.OMRAT.LT.0.25)KXERR  = 7                        STEP1380
      IF(KXERR.EQ.7)WRITE(6,275) I                                       STEP1390
      IF(KXERR.EQ.7)LVAR=8                                               STEP1400
      IF (LVAR.EQ.8) OUTPUT=6                                            STEP1410
  280 CONTINUE                                                           STEP1420
  275 FORMAT(/,37H PROGRAM TERMINATED BECAUSE THE OMEGA,/                STEP1430
     121H SPACING BETWEEN I = ,I2,1X,17HAND THE PRECEDING,/              STEP1440
     249H NODE IS EITHER MORE THAN FOUR TIMES OR LESS THAN,/             STEP1450
     334H ONE QUARTER THE PRECEDING SPACING)                             STEP1460
      RETURN                                                             STEP1470
C----------------------------------------------------- STEP 3 ----STEP1480
C.....STEP3 COMPUTES Y'S,R'S                                             STEP1490
C.....AND COMPUTES UMAX,UMIN,AND YL                                      STEP1500
C.....                                                                   STEP1510
  300 IF (INTG.EQ.0) GO TO 360                                           STEP1520
C.....Y NEAR THE I BOUNDARY                                              STEP1530
      GO TO (312,314,316),KIN                                            STEP1540
  312 Y(2)=(1.+BETA)*OM(3)*4./((3.*RHO(2)+RHO(3))*(U(2)+U(3)))           STEP1550
      GO TO 320                                                          STEP1560
  314 Y(2)=12.*OM(3)/((3.*RHO(2)+RHO(3))*(U(2)+U(3)+4.*U(1)))            STEP1570
      GO TO 320                                                          STEP1580
  316 Y(2)=.5*OM(3)/(RHO(1)*U(1))                                        STEP1590
  320 Y(3)=Y(2)+.25*OM(3)*(1./(RHO(3)*U(3))+2./(RHO(3)*U(3)+RHO(2)*      STEP1600
     1U(2)))                                                             STEP1610
      IF(BETA.GE.0.9.AND.KIN.EQ.1)Y(3)=3.0*OM(3)/(RHO(3)*U(3)+RHO(2)     STEP1620
     1*U(2))                                                             STEP1630
C.....Y 'S FOR INTERMEDIATE GRID POINTS                                  STEP1640
      DO 330 I=4,NP1                                                     STEP1650
  330 Y(I)=Y(I-1)+2.*OMD(I-1)/(RHO(I)*U(I)+RHO(I-1)*U(I-1))              STEP1660
C.....Y NEAR THE E BOUNDARY                                              STEP1670
      Y(NP2)=Y(NP1)+.25*OMD(NP1)*(1./(RHO(NP1)*U(NP1))+2./               STEP1680
     1(RHO(NP1)*U(NP1)+RHO(NP2)*U(NP2)))                                 STEP1690
      GO TO (332,334,336),KEX                                            STEP1720
  332 Y(NP3)=Y(NP2)+(1.+BETA)*OMD(NP1)*4./((RHO(NP1)+3.*RHO(NP2          STEP1730
     1))*(U(NP1)+U(NP2)))                                                STEP1740
      IF (BETA.GE.0.9) Y(NP3)=Y(NP1)+1.5*(OM(NP2)-OM(NP1))/              STEP1741
     1 (0.5*(RHO(NP1)*U(NP1)+RHO(NP2)*U(NP2)))                           STEP1742
      GO TO 340                                                          STEP1750
  334 Y(NP3)=Y(NP2)+12.*OMD(NP1)/((RHO(NP1)+3.*RHO(NP2))*(U(NP2          STEP1760
     1)+U(NP1)+4.*U(NP3)))                                               STEP1770
      GO TO 340                                                          STEP1780
  336 Y(NP3)=Y(NP2)+.5*OMD(NP1)/(RHO(NP3)*U(NP3))                        STEP1790
```

```
  340 IF(KRAD.EQ.0) GO TO 344
      DO 342 I=2,NP3
  342 Y(I)=2.*Y(I)*PEI/(R(1)+SQRT(R(1)*R(1)+2.*Y(I)*PEI*CSALFA))
      GO TO 350
  344 DO 346 I=2,NP3
  346 Y(I)=PEI*Y(I)/R(1)
  350 Y(2)=2.*Y(2)-Y(3)
      IF(BETA.GE.0.9.AND.KIN.EQ.1)Y(2)=Y(3)/3.
      IF(BETA.GT.0.9.AND.KEX.EQ.1)GO TO 351
      Y(NP2)=2.*Y(NP2)-Y(NP1)
      GO TO 352
  351 Y(NP2)=(2.*Y(NP3)+Y(NP1))/3.
C.....CALCULATION OF RADII
  352 DO 355 I=2,NP3
      IF(KRAD.EQ.0)R(I)=R(1)
      IF(KRAD.NE.0)R(I)=R(1)+Y(I)*CSALFA
  355 CONTINUE
C.....CALCULATION OF THE BOUNDARY LAYER THICKNESS
C.....BASED ON THE FR CRITERION USED IN 'INPUT'.  I.E., IF FR
C.....IS 0.01, THIS ROUTINE CALCULATES YL WHICH IS THE 99 PER-
C.....CENT THICKNESS OF THE BOUNDARY LAYER.  YL IS THEN A DISTANCE UPON
C.....WHICH A TURBULENCE LENGTH SCALE IS BASED.
C  SEARCH FOR THE MAXIMUM AND MINIMUM VELOCITIES
  360 UMAX=U(1)
      UMIN=U(1)
      DO 375 J=3,NP3
      IF(J.EQ.NP2) GO TO 375
      IF(U(J).GT.UMAX)UMAX=U(J)
      IF(U(J).LT.UMIN)UMIN=U(J)
  375 CONTINUE
      DIF=ABS(UMAX-UMIN)*FR
C.....SEARCH NEAR THE I BOUNDARY
      IF(KIN.NE.2) GO TO 386
      U21=ABS(.5*(U(2)+U(3))-U(1))
      IF(U21.LT.DIF) GO TO 380
      YIP=SQRT(DIF/U21)*(Y(2)+Y(3))*.5
      GO TO 388
  380 J=2
  382 J=J+1
      UJ1=U(J)-U(1)
      IF(ABS(UJ1).GE.DIF) GO TO 384
      GO TO 382
  384 A1=1.
      IF(UJ1.LT.0.)A1=-1.
      YIP=Y(J-1)+(Y(J)-Y(J-1))*(U(1)+A1*DIF-U(J-1))/(U(J)-U(J-1))
      GO TO 388
  386 YIP=0.
C.....SEARCH NEAR THE E BOUNDARY
  388 IF(KEX.NE.2) GO TO 396
      U21=ABS(.5*(U(NP1)+U(NP2))-U(NP3))
      IF(U21.LT.DIF) GO TO 390
      YEM=SQRT(DIF/U21)*(.5*(Y(NP1)+Y(NP2))-Y(NP3))+Y(NP3)
      GO TO 398
  390 J=NP2
  392 J=J-1
      UJ1=U(J)-U(NP3)
      IF(ABS(UJ1).GE.DIF) GO TO 394
      GO TO 392
  394 A1=1.
      IF(UJ1.LT.0.)A1=-1.
```

```
      YEM=Y(J+1)+(Y(J)-Y(J+1))*(U(NP3)+A1*DIF-U(J+1))/(U(J)-U(J+1))
      GO TO 398
  356 YEM=Y(NP3)
  398 YL=YEM-YIP
      RETURN
C--------------------------------------------------------- STEP 4 ----
C.....STEP4 CALCULATES ALL OF THE COEFFICIENTS FOR THE
C.....FINITE DIFFERENCE EQUATIONS.
C.....THE FDE'S ARE THEN INTEGRATED.
C.....
C.....SOURCE TERMS, SU(J,I), AND SMALL C'S ARE COMPUTED IN
C.....SUBROUTINE AUX.
C.....THE CONVECTICN TERM
  400 KXX=0
      KXXX=0
  401 RFDX=1./4./DX
      SA=R(1)*AMI/PEI
      SBDF=(R(NP3)*AME-R(1)*AMI)/PEI/4.
      P2=3.*RFDX
      DO 420 I=3,NP1
      P1=OMD(I)*RFDX*RBOM(I)
      P3=OMD(I-1)*RFDX*RBOM(I)
      G1=P1+(SA+SBDF*(OM(I+1)+3.*OM(I)))*RBOM(I)
      G2=P2-SBDF
      G3=P3-(SA+SBDF*(OM(I-1)+3.*OM(I)))*RBOM(I)
      AU(I)=SC(I)*ROMD(I)*2.*RBOM(I)
      BU(I)=SC(I-1)*ROMD(I-1)*2.*RBOM(I)
      CU(I)=-P1*U(I+1)-P2*U(I)-P3*U(I-1)
      IF(NEQ.EQ.1) GO TO 410
      DO 405 J=1,NPH
      C(J,I)=-P1*F(J,I+1)-P2*F(J,I)-P3*F(J,I-1)
      C(J,I)=-C(J,I)+SU(J,I)-F(J,I)*SD
      A(J,I)=AU(I)/PREF(J,I)
      B(J,I)=BU(I)/PREF(J,I-1)
  405 CONTINUE
  410 CONTINUE
C.....SOURCE TERM FOR VELOCITY EQUATION
      S1=DPDX*DX-GC*BF(I)*DX
      S2=P2*S1/(RHO(I)*U(I))
      S3=P3*S1/(RHO(I-1)*U(I-1))
      S1=P1*S1/(RHO(I+1)*U(I+1))
      CU(I)=-CU(I)-2.*(S1+S2+S3)
      S1=S1/U(I+1)
      S2=S2/U(I)
      S3=S3/U(I-1)
C.....COEFFICIENTS IN THE FINAL FORM
      RL=1./(G2+AU(I)+BU(I)-S2)
      AU(I)=(AU(I)+S1-G1)*RL
      BU(I)=(BU(I)+S3-G3)*RL
      CU(I)=CU(I)*RL
      IF(NEQ.EQ.1) GO TO 420
      DO 415 J=1,NPH
      RL=1./(G2+A(J,I)+B(J,I)-SD)
      A(J,I)=(A(J,I)-G1)*RL
      B(J,I)=(B(J,I)-G3)*RL
  415 C(J,I)=C(J,I)*RL
  420 CONTINUE
C.....SLIP COEFFICEINTS NEAR THE I BOUNDARY FOR VELOCITY EQUATION
      CU(2)=0.
      CU(NP2)=0.
```

```
      GO TO (422,424,426),KIN
  422 BU(2)=0.
      AU(2)=1./(1.+2.*BETA)
      GO TO 430
  424 SQ=84.*U(1)*U(1)-12.*U(1)*U(3)+9.*U(3)*U(3)
      BU(2)=8.*(2.*U(1)+U(3))/(2.*U(1)+7.*U(3)+SQRT(SQ))
      IF(U(5).LE.U(1))BU(2)=1.
      AU(2)=1.-BU(2)
      GO TO 430
  426 BU(2)=0.
      AK1=1./DX-(DPDX-GC*BF(2))/(RHO(1)*U(1)*U(1))
      AK2=-U(1)*AK1+(DPDX-GC*BF(2))/(RHO(1)*U(1))
      AJ=RHO(1)*U(1)*.25*(Y(2)+Y(3))**2/EMU(2)
      IF(KRAD.EQ.0) GO TO 428
      AU(2)=2./(2.+AJ*AK1)
      CU(2)=-.5*AJ*AK2*AU(2)
      GO TO 430
  428 CU(2)=1./(2.+3.*AJ*AK1)
      AU(2)=CU(2)*(2.-AJ*AK1)
      CU(2)=-CU(2)*4.*AJ*AK2
C.....SLIP COEFFICIENTS NEAR THE E BOUNDARY FOR VELOCITY EQUATION
  430 GO TO (432,434,436),KEX
  432 AU(NP2)=0.
      BU(NP2)=1./(1.+2.*BETA)
      GO TO 440
  434 SQ=84.*U(NP3)*U(NP3)-12.*U(NP3)*U(NP1)+9.*U(NP1)*U(NP1)
      AU(NP2)=8.*(2.*U(NP3)+U(NP1))/(2.*U(NP3)+7.*U(NP1)+SQRT(SQ))
      BU(NP2)=1.-AU(NP2)
      GO TO 440
  436 AU(NP2)=0.
      BK1=1./DX-(DPDX-GC*BF(NP2))/(RHO(NP3)*U(NP3)*U(NP3))
      BK2=-U(NP3)*BK1+(DPDX-GC*BF(NP2))/(RHO(NP3)*U(NP3))
      BJ=RHO(NP3)*U(NP3)*.25*(2.*Y(NP3)-Y(NP1)-Y(NP2))**2/EMU(NP1)
      CU(NP2)=1./(2.+3.*BJ*BK1)
      BU(NP2)=CU(NP2)*(2.-BJ*BK1)
      CU(NP2)=-CU(NP2)*4.*BJ*BK2
  440 IF(NEQ.EQ.1) GO TO 471
C.....SLIP COEFFICIENTS NEAR THE I BOUNDARY FOR OTHER EQUATIONS
      DO 470 J=1,NPH
      C(J,2)=0.
      C(J,NP2)=0.
      GO TO (452,454,456),KIN
  452 IF (INDI(J).EQ.1) GO TO 453
      VA=GAMA(J)/((1.+BETA)*(QWF(J)+AMI))
      A(J,2)=(1.-VA*AMI)/(1.+VA*AMI)
      B(J,2)=0.
      C(J,2)=2.*AJI(J)*VA/(1.+VA*AMI)
      GO TO 460
  453 A(J,2)=(1.+BETA-GAMA(J))/(1.+BETA+GAMA(J))
      B(J,2)=1.-A(J,2)
      IF(SOURCE(J).NE.2)GO TO 460
      A(J,2)=-1.
      B(J,2)=0.
      C(J,2)=2.*FI(J)
      GO TO 460
  454 A(J,2)=(U(2)+U(3)-8.*U(1))/(5.*(U(2)+U(3))+8.*U(1))
      GF=(1.-PREF(J,2))/(1.+PREF(J,2))
      A(J,2)=(A(J,2)+GF)/(1.+A(J,2)*GF)
      B(J,2)=1.-A(J,2)
      GO TO 460
```

```
  456 B(J,2)=0.
      DS=0.
      AK1=1./DX-DS
      CS=0.
      AK2=-AK1*F(J,1)-CS
      AJF=AJ*PREF(J,2)
      IF(KRAD.EQ.0) GO TO 457
      A(J,2)=2./(2.+AJF*AK1)
      C(J,2)=-.5*AJF*AK2*A(J,2)
      GO TO 460
  457 C(J,2)=1./(2.+3.*AJF*AK1)
      A(J,2)=C(J,2)*(2.-AJF*AK1)
      C(J,2)=-C(J,2)*4.*AJF*AK2
C.....SLIP COEFFICIENTS NEAR THE E BOUNDARY FOR OTHER EQUATIONS
  460 GO TO (462,464,466),KEX
  462 IF (INDE(J).EQ.1) GO TO 463
      VA=GAMA(J)/((1.+BETA)*(QWF(J)-AME))
      B(J,NP2)=(1.+VA*AME)/(1.-VA*AME)
      A(J,NP2)=0.
      C(J,NP2)=-2.*AJE(J)*VA/(1.-VA*AME)
      GO TO 470
  463 B(J,NP2) = (1.+BETA-GAMA(J))/(1.+BETA + GAMA(J))
      A(J,NP2) = 1. - B(J,NP2)
      IF(SOURCE(J).NE.2)GO TO 470
      A(J,NP2)=0.
      B(J,NP2)=-1.
      C(J,NP2)=2.*FI(J)
      GO TO 470
  464 B(J,NP2)=(U(NP2)+U(NP1)-8.*U(NP3))/(5.*(U(NP2)+U(NP1))+8.*U(NP3))
      GF=(1.-PREF(J,NP1))/(1.+PREF(J,NP1))
      B(J,NP2)=(B(J,NP2)+GF)/(1.+B(J,NP2)*GF)
      A(J,NP2)=1.-B(J,NP2)
      GO TO 470
  466 A(J,NP2)=0.
      DS=0.
      BK1=1./DX-DS
      CS=0.
      BK2=-BK1*F(J,NP3)-CS
      BJF=BJ*PREF(J,NP1)
      C(J,NP2)=1./(2.+3.*BJF*BK1)
      B(J,NP2)=C(J,NP2)*(2.-BJF*BK1)
      C(J,NP2)=-C(J,NP2)*4.*BJF*BK2
  470 CONTINUE
C.....SETTING UP VELOCITIES AT A FREE BOUNDARY
  471 IF(KEX.EQ.2)U(NP3)=SQRT(U(NP3)*U(NP3)-2.*DX*(DPDX-GC*BF(NP3))
     1/RHO(NP3))
      IF(KIN.EQ.2)U(1)=SQRT(U(1)*U(1)-2.*DX*(DPDX-GC*BF(1))/RHO(1))
C.....THIS IS THE TRI-DIAGONAL ROUTINE WHERE THE FINITE
C.....DIFFERENCE EQUATIONS ARE ACTUALLY SOLVED.
C.....INTEGRATE VELOCITY
      BU(2) = BU(2)*U(1) + CU(2)
      DO 472 I=3,NP2
      TT = 1./(1.-BU(I)*AU(I-1))
      AU(I) = AU(I)*TT
  472 BU(I) = (BU(I)*BU(I-1) + CU(I))*TT
      DO 474 I=2,NP2
      JJ=NP2-I+2
  474 U(JJ)=AU(JJ)*U(JJ+1)+BU(JJ)
      DO 476 I=3,NP1
      IF(U(I).GT.0.0)GO TO 476
```

```
      U(I)=-U(I)                                                        STEP4200
      KXX=1                                                             STEP4210
  476 CONTINUE                                                          STEP4220
      KXXX=KXXX+1                                                       STEP4230
      IF(KXX.EQ.0)GO TO 478                                             STEP4240
      IF(KXXX.GT.2)GO TO 478                                            STEP4250
C.....ATTEMPT TO RE-SOLVE IF NEGATIVE VELOCITY APPEARS                  STEP4260
      IF(KEX.EQ.2)AME=AME/1.3                                           STEP4270
      IF(KIN.EQ.2)AMI=AMI/1.3                                           STEP4280
      DO 4777 I=2,NP1                                                   STEP4290
      RAVG=0.5*(R(I+1)+R(I))                                            STEP4300
      RHOAV=0.5*(RHO(I+1)+RHO(I))                                       STEP4310
C.....ADJUSTMENT OF EMU AT 2.5 AND N+1.5                                STEP4320
      IF (I.GT.2) GO TO 4777                                            STEP4330
      IF (KIN.NE.1) GO TO 4778                                          STEP4340
      IF (BETA.LT.0.02.OR.BETA.GT.0.9) GO TO 4777                       STEP4350
      EMU(2)=TAU*(Y(2)+Y(3))/(BETA*(U(2)+U(3)))                         STEP4360
 4778 IF (KEX.NE.1) GO TO 4777                                          STEP4370
      IF (BETA.LT.0.02.OR.BETA.GT.0.9) GO TO 4777                       STEP4380
      EMU(NP1)=TAU*(Y(NP3)-0.5*(Y(NP1)+Y(NP2)))/                        STEP4390
     1(BETA*0.5*(U(NP1)+U(NP2)))                                        STEP4400
C.....COMPUTE SMALL C'S                                                 STEP4410
 4777 SC(I)=RAVG*RAVG*RHOAV*0.5*(U(I+1)+U(I))*EMU(I)/(PEI*PEI)          STEP4420
      WRITE (6,477) INTG                                                STEP4430
  477 FORMAT (/10X,'VELOCITY NEGATIVE, RE-SOLVE, INTG=',I4/)            STEP4440
      GO TO 401                                                         STEP4450
  478 CONTINUE                                                          STEP4460
C.....SETTING UP VELOCITIES AT A SYMMETRY LINE                          STEP4470
      IF(KIN.NE.3) GO TO 480                                            STEP4480
      U(1)=U(2)                                                         STEP4490
      IF(KRAD.EQ.0)U(1)=0.75*U(2)+.25*U(3)                              STEP4500
  480 IF(KEX.EQ.3)U(NP3)=.75*U(NP2)+.25*U(NP1)                          STEP4510
      IF(NEQ.EQ.1) GO TO 494                                            STEP4520
C.....INTEGRATE F EQUATIONS                                             STEP4530
      DO 492 J=1,NPH                                                    STEP4540
      DO 482 I=2,NP2                                                    STEP4550
      AU(I)=A(J,I)                                                      STEP4560
      BU(I)=B(J,I)                                                      STEP4570
  482 CU(I)=C(J,I)                                                      STEP4580
      IF (SOURCE(J).NE.2) GO TO  4886                                   STEP4581
      IF (ITKE.EQ.1) GO TO 4886                                         STEP4582
      IF (KEX.EQ.1) GO TO 4884                                          STEP4583
      DO 4882 I=1,ITKE                                                  STEP4584
      AU(I)=0.                                                          STEP4585
      BU(I)=0.                                                          STEP4586
 4882 CU(I)=F(J,ITKE)                                                   STEP4587
      GO TO 4886                                                        STEP4588
 4884 DO 4885 I=ITKE,NP3                                                STEP4589
      AU(I)=0.                                                          STEP4590
      BU(I)=0.                                                          STEP4591
 4885 CU(I)=F(J,ITKE)                                                   STEP4592
 4886 CONTINUE                                                          STEP4593
      BU(2) = BU(2)*F(J,1) + CU(2)                                      STEP4594
      DO 484 I=3,NP2                                                    STEP4600
      TT = 1./(1.-BU(I)*AU(I-1))                                        STEP4610
      AU(I) = AU(I)*TT                                                  STEP4620
  484 BU(I) = (BU(I)*BU(I-1) + CU(I))*TT                                STEP4630
      DO 486 I=2,NP2                                                    STEP4640
      JJ=NP2-I+2                                                        STEP4650
  486 F(J,JJ)=AU(JJ)*F(J,JJ+1)+BU(JJ)                                   STEP4660
```

```
C.....SETTING UP SYMMETRY-LINE VALUES OF F
      IF(KIN.NE.3) GO TC 490
      F(J,1)=F(J,2)
      IF(KRAD.EQ.0)F(J,1)=.75*F(J,2)+.25*F(J,3)
  490 IF(KEX.EQ.3)F(J,NP3)=.75*F(J,NP2)+.25*F(J,NP1)
  492 CONTINUE
  494 RETURN
C--------------------------------------------------------------- STEP 5 ----
C.....STEP5 INITIALIZES PARAMETERS AND SETS UP INITIAL CONDITIONS.
C.....
  500 XD=XU
      PRE=PO
      AME=0.
      AMI=0.
      INTG=0
      BETA=0.
      CAY=0.0
      PPL=0.0
      YL=0.0
      REM=1.
      REH=1.
      H=1.
      CF2=0.002
      TAUW=0.02
      IF(NPH.EQ.0)SOURCE(1)=0
      DO 540 I=1,NP3
      EMU(I)=0.0
      BF(I)=0.0
      SP(I)=0.0
      IF (NPH.EQ.0) GO TO 540
      DO 530 J=1,NPH
      ST(J)=0.002
      QW(J)=0.0
      IF (SOURCE(J).EQ.2) PR(J,I)=1.0
  530 CONTINUE
  540 CONTINUE
      FMEAN=0.0
      LSUB=0
      LVAR=0
      ALMG=ALMGG
      KRAD = 1
      IF(GEOM.EQ.1)KRAC=0
      IF(GEOM.EQ.5)KRAD=0
      IF(GECM.EQ.8)KRAC=0
      IF(GEOM.EQ.9)KRAC=0
      IF(NEQ.EQ.1) GC TC 560
      DO 550 J=1,NPH
      AJI(J)=0.0
      AJE(J)=0.0
      INDI(J)=0
      INDE(J)=0
      IF(KIN.EC.1)INDI(J)=TYPBC(J)
      IF(KEX.EQ.1)INDE(J)=TYPBC(J)
  550 CONTINUE
  560 CONTINUE
      RETURN
      END

      SUBROUTINE WALL
```

```
C.....
C.....
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      DIMENSION HW(5),HG(5),PRW(5),PRI(5),HPS(5),DHY(5)
      COMMON/GEN/PEI,AMI,AME,DPDX,XJ,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,NXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SCURCE,RETRAN,NUMRUN,SPACE,RWJ,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,QWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     8/L1/YL,UMAX,UMIN,FR,YIP,YEM,ENFRA,KENT,AUXM2
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHOM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AUX1(100),AUX2(100),YPMAX
C.....
      IF(INTG.GT.1.OR.LSUB.GT.0)GO TO 8
      KSTART=1
      MARKER=0
      C3=0.0
      C4=0.0
      C3N=0.
      C5=0.0
      BFOLD=0.0
      C5N=0.0
      AJO=0.0
      AJN=0.0
      KKK=0
      TPLUS=1.
      CLDDPX=0.0
      DUDY=1.
      APLO=APL
      BPLO=BPL
      EE=0.04
      IF(NPH.LT.1)GO TO 8
      DO  6 J=1,NPH
      DHY(J)=1.
    6 QW(J)=100.
    8 LSUB=0
      LTPL=0
C ------------------------------------------------ SECTION ONE ----
C.....THE JOIN POINT CONDITIONS ARE SET UP HERE
      IF(KEX.NE.1)GO TO 20
      RHW=RHO(NP3)
      VISG=VISCO(1)
       VISW=VISCO(NP3)
      RHG=RHO(1)
      RHI=0.5*(RHO(NP1)+RHO(NP2))
      VISI=0.5*(VISCO(NP1)+VISCO(NP2))
      BFOR=(BF(NP2)+BF(NP1))/2.
      UGG = UGU
      YI=Y(NP3)-.5*(Y(NP1)+Y(NP2))
      UI=.5*(U(NP2)+U(NP1))
      UE=U(N)-(U(N)-U(NP1))/(Y(N)-Y(NP1))*(Y(N)-.5*(Y(NP1)+Y(NP2)))
      IF(MODE.EQ.1)UE=UI
      IF(BETA.GE.0.9)UE=UI
      UI=(UI+UE)/2.
      REW=ABS(UI*YI*RHW/VISW)
      AMW = -AME
      IF(NEQ.EQ.1)GO TO 40
```

```
      DO 10 J=1,NPH
      HW(J)=F(J,NP3)
      HG(J)=F(J,1)
      PRW(J)=PR(J,NP3)
      PRI(J)=0.5*(PR(J,NP1)+PR(J,NP2))
      FI(J)=.5*(F(J,NP2)+F(J,NP1))
      FE=F(J,N)-(F(J,N)-F(J,NP1))/(Y(N)-Y(NP1))*(Y(N)-.5*(Y(NP1)+Y(NP2)
     1 ))
      IF(MODE.EQ.1)FE=FI(J)
      IF(GAMA(J).GE.0.9.AND.GAMA(J).LE.1.1)FE=FI(J)
      IF(SOURCE(J).EQ.2)FE=FI(J)
   10 FI(J)=(FI(J)+FE)/2.
      GO TO 40
   20 IF(KIN.NE.1)RETURN
       RHW=RHO(1)
       VISW=VISCO(1)
      VISG=VISCO(NP3)
       RHG=RHO(NP3)
       RHI=0.5*(RHO(2)+RHO(3))
      VISI=0.5*(VISCO(2)+VISCO(3))
      BFOR=(BF(2)+BF(3))/2.
      UGG = UGU
      YI=.5*(Y(2)+Y(3))
      UI=.5*(U(2)+U(3))
      UE=U(3)+0.5*(Y(2)-Y(3))*(U(4)-U(3))/(Y(4)-Y(3))
      IF(MODE.EQ.1)UE=UI
      IF(BETA.GE.0.9)UE=UI
      UI = (UI + UE)/2.
      REW=ABS(UI*YI*RHW/VISW)
      AMW = AMI
      IF(NEQ.EQ.1)GO TO 40
      DO 30 J=1,NPH
      HW(J)=F(J,1)
      HG(J)=F(J,NP3)
      PRW(J)=PR(J,1)
      PRI(J)=0.5*(PR(J,2)+PR(J,3))
      FI(J)=.5*(F(J,2)+F(J,3))
      FE=F(J,3)+0.5*(Y(2)-Y(3))*(F(J,4)-F(J,3))/(Y(4)-Y(3))
      IF(MODE.EQ.1)FE=FI(J)
      IF(GAMA(J).GE.0.9.AND.GAMA(J).LE.1.1)FE=FI(J)
      IF(SOURCE(J).EQ.2)FE=FI(J)
   30 FI(J)=(FI(J)+FE)/2.
   40 UTAUW=SQRT(GC*TAUW/RHW)
      UTAUG=SQRT(GC*TAUW/RHG)
      IF(REW.LT.4.)GC TO 160
C ---------------------------------------------------- SECTION TWO ----
C.....SOURCE TERMS FOR COUETTE FLOW STAG ENTHALPY EQUATION
      S=0.0
      IF(NEQ.EQ.1)GO TO 160
      DO 150 J=1,NPH
      IF(SOURCE(J).EQ.3.OR.SOURCE(J).EQ.4)S=AUX42
      IF(SOURCE(J).EQ.1.OR.SOURCE(J).EQ.3)GO TO 130
      GO TO 150
  130 IF(UGG.LT.0.01)GO TO 140
      DENOM=QW(J)*CJ
      IF(ABS(DENOM).LT.0.00001)GO TO 140
C.....NOTE: IF WALL HEAT FLUX IS NEAR ZERO, VISCOUS DISSIPATION
C.....   IS NOT PROPERLY HANDLED.  ALWAYS USE AT LEAST A SMALL
C.....   HEAT FLUX. SAME TRUE OF HEAT SOURCE, S.
      C3N=TAUW*UTAUW/DENOM
```

```
      C4=VISW*BFOR/(2.*DENOM*RHW)
      C5N=VISW*CJ/(RHW*DENOM*UTAUW)
  140 C3=(C3+C3N)/2.
      C5=(C5+C5N)/2.
      IF(INTG.LT.5)C3=0.0
      IF(TYPBC(J).EQ.1)GO TO 150
      AJN=AJE(J)
      IF(KIN.EQ.1)AJN=AJI(J)
      IF(ABS(AJN).GT.ABS(1.1*AJO))KKK=INTG
      IF(INTG.LE.(KKK+1))C3=0.0
      AJO=AJE(J)
      IF(KIN.EQ.1)AJO=AJI(J)
  150 CONTINUE
C ------------------------------------------------------ SECTION THREE ----
C.....COUETTE FLOW EQUATION TERMS ARE COMPUTED HERE
  160 IF (INTG.LT.2) OLDDPX=0.
      IF(UGG.LE.0.01)GC TO 165
      CAY=-0.5*(DPDX+OLDDPX-2.*GC*BFOR)/(UGG*UGG*UGG)*VISG/(RHG*RHG)
  165 PPL=.5*(DPDX+OLDDPX)*VISW/(TAUW*RHW*GC*UTAUW)
      GPL=AMW/(RHW*UTAUW)
      BFPLUS=BFOR*VISW/(TAUW*RHW*UTAUW)
      BFPLUS=(BFPLUS+BFOLD)/2.
      BFOLD=BFPLUS
      IF(KSTART.EQ.1)PPL=0.
      IF(KSTART.EQ.1)BFPLUS=0.
      IF(KSTART.EQ.1)GPL=0.
      AKK=AK
      IF(MODE.EQ.1)AKK=0.0
C.....TURBULENT FLOW DAMPING TERMS ARE COMPUTED HERE
      IF(KD.EQ.1)GO TO 180
      IF(KD.EQ.3)GC TO 180
      IF(INTG.EQ.1)PPLE=PPL-BFPLUS
      IF(INTG.EQ.1)GPLE=GPL
      IF(PPLAG.LT.400.) PPLE=PPL-BFPLUS
      IF(PPLAG.LT.400.)GPLE=GPL
      IF(PPLAG.LT.400.)GO TO 170
      IF(MARKER.EQ.1)GC TO 170
      DIR=1.0
      IF((PPL-BFPLUS).GT.PPLE)DIR=0.3
      PPLE=PPL-BFPLUS-(PPL-BFPLUS-PPLE)*EXP(-(RHW*DX*UTAUG)/
     1(VISW*PPLAG*DIR))
      GPLE=GPL-(GPL-GPLE)*EXP(-(RHW*DX*UTAUG)/
     1(VISW*PPLAG))
  170 CONTINUE
C.....THE FOLLOWING ARE EMPIRICAL CORRELATIONS FOR THE DAMPING TERM
C.....IN THE MIXING-LENGTH EXPRESSION.
      AC=7.1
      BC=4.25
      CC=10.0
      IF(PPLE.GT.0.)BC=2.9
      IF(PPLE.GT.0.)CC=0.
      IF(GPLE.LT.0.)AC=9.0
      APL=APLO/(AC*(GPLE+BC*(PPLE/(1.+CC*GPLE)))+1.)
      BPL=BPLO/(AC*(GPLE+BC*(PPLE/(1.+CC*GPLE)))+1.)
      IF(APL.LT.-.001)APL=1000.
      IF(BPL.LT.-.001)BPL=1000.
      IF(INTG.EQ.1.OR.REM.GT.2.*RETRAN)GO TO 180
C.....THE FOLLOWING IS A GIMMICK TO SIMULATE A GRADUAL TRANSITION.
      IF(KD.LE.1.AND.MODE.EQ.2)APL=APL+(300.-APL)*(1.-SIN(1.57
     1*(REM-RETRAN)/RETRAN))**2
```

```
      IF(KD.GE.2.AND.MODE.EQ.2)BPL=BPL+(400.-BPL)*(1.-SIN(1.57
     1*(REM-RETRAN)/RETRAN))**2
C ----------------------------------------------------- SECTION FOUR ----
  180 IF(REW.LT.4..AND.INTG.EQ.22)WRITE(6,900)
      IF(REW.LT.4..AND.INTG.EQ.92)WRITE(6,900)
  900 FORMAT(/' THE PROGRAM IS BYPASSING THE WALL FUNCTION, AT LEAST AT
     1THIS INTEGRATION.'/)
      IF(REW.LT.4.)GO TO 290
C ----------------------------------------------------- SECTION FIVE ----
C.....THIS IS THE BEGINNING OF A LOOP IN WHICH THE MOMENTUM
C.....AND ANY NUMBER OF DIFFUSION COUETTE FLOW ORDINARY DIFFERENTIAL
C.....EQUATIONS ARE SOLVED.
      IF(INTG.EQ.22.OR.INTG.EQ.92)WRITE(6,910)
  910 FORMAT(/' THE PROGRAM IS EMPLOYING THE WALL FUNCTION, AT LEAST AT
     1THIS INTEGRATION.'/)
      YPL=0.0
      DYPL=0.1*SQRT(REW)
      IF (DYPL.GT.0.25) DYPL=0.25
      UPL=0.0
      DUDY=1.
      A2=0.0
      IF(NPH.LT.1)GO TO 200
      DO 190 J=1,NPH
      HPS(J)=0.0
  190 DHY(J)=PRW(J)
  200 KCHECK=0
  210 IF(YPL.GT.2.5)DYPL=YPL/10.
  220 TPLUS=(1.0+GPL*(UPL+DUDY*DYPL/2.)+(PPL-BFPLUS)*(YPL+DYPL/2.)-
     1(PPL-BFPLUS)*CF2*A2/RHG)
      IF(TPLUS.LT.0.0)TPLUS=0.0
      IF(TPLUS.LT.0.1)LTPL=1
      IF(TPLUS.LT.0.1)LSUB=2
      RR=1.+((RHI/RHW)-1.)*(UPL*UTAUW/UI)
      IF(RR.LE.0.)RR=1.
      VR=1.+((VISI/VISW)-1.)*(UPL*UTAUW/UI)
      IF(VR.LE.0.)VR=1.
      IF(KD.GT.1)GO TO 230
      YLOAP=(YPL+DYPL/2.)/APL/(VR/SQRT(RR))
      IF(YLOAP.GT.10.)EE=1.
      IF(YLOAP.GT.10.)GO TO 240
      EE=1.-1./EXP(YLOAP)
      GO TO 240
  230 EE=(YPL+DYPL/2.)/BPL/(VR/SQRT(RR))
      IF(EE.GT.1.)EE=1.
  240 DD=1.+4.*TPLUS*AKK*AKK*EE*EE*(YPL+DYPL/2.)*(YPL+DYPL/2.)*
     1(RR/(VR*VR))
      DUDY=(2.*TPLUS/(1.+SQRT(DD)))/VR
      UPL=UPL+DUDY*DYPL
      DI2=RR*RHW*.5*((UPL-DUDY*DYPL)*(UPL-DUDY*DYPL)+UPL*UPL)*DYPL
      A2=DI2+A2
      IF(NEQ.EQ.1)GO TO 260
      EDR=(RR/VR)*AKK*AKK*(YPL+DYPL/2.)*(YPL+DYPL/2.)*EE*EE*DUDY
      EDRT=EDR+1.0
      DO 250 J=1,NPH
      PRR=1.+((PRI(J)/PRW(J))-1.)*(UPL*UTAUW/UI)
C.....THE FOLLOWING IS THE FREE CONSTANT IN THE TURBULENT PRANDTL
C.....NUMBER EQUATION. EXPERIENCE MAY SUGGEST A DIFFERENT VALUE.
      CT = 0.2
      PRTJ=PRT(J)
      PETC=EDR*CT*PRR*PRW(J)
```

```
      IF(PETC.LT..001)PETC=.001
      IF(PETC.GT.100..OR.MODE.EQ.1)GO TO 245
      ALPHA=SQRT(1./PRTJ)
      AOP=ALPHA/PETC
      IF(AOP.GT.10.)AOP=10.
      PRTJ=1./(1./(2.*PRTJ)+ALPHA*PETC-PETC*PETC*(1.-EXP(-AOP)))
      IF(K3.EQ.3.OR.SOURCE(J).EQ.2)PRTJ=PRT(J)
  245 PREFW=EORT/(1./(FRR*PRW(J))+EDR/PRTJ)
      DHY(J)=(PREFW/(ECRT*VR))*(1.+GPL*(HPS(J)+DHY(J)*DYPL/2.)+S*C5*
     1YPL+C4*UPL*YPL)+C3*
     2(PREFW-1.)*(UPL-DUDY*DYPL/2.)*DUDY
      HPS(J)=HPS(J)+DHY(J)*DYPL
  250 CONTINUE
  260 YPL=YPL+DYPL
      REL=UPL*YPL
      IF(INTG.EQ.1)GO TC 270
      IF(YPL.GT.YPMAX)LSUB=2
      IF(LSUB.EQ.2) GO TO 340
  270 IF(KCHECK.EQ.1) GC TO 280
C.....AT THIS POINT THE PRODUCT UPL*YPL IS COMPARED TO U*Y*RHO/MU AT THE
C.....JOIN-POINT IN THE MAIN PROGRAM GRID.
      IF(REL.LT.REW)GO TO 210
  280 ERR=REL-REW
      AERR=ABS(ERR)
      ER2=AERR/REW
      IF(ER2.LT.0.01)GO TO 300
      KCHECK=1
      IF(ERR.LT.0.0)DYPL=ABS(DYPL/2.0)
      IF(ERR.GT.0.0)DYPL=-ABS(DYPL/2.0)
      GO TO 220
C ----------------------------------------------------- SECTION SIX ----
C.....THIS SECTION IS USED IF THE NUMERICAL INTEGRATION
C.....OF THE COUETTE FLOW EQUATIONS IS BYPASSED
  290 YPL=SQRT(REW)
      YPL=SQRT(ABS(REW/(1.+(PPL-BFPLUS)*YPL/2.+GPL*YPL/2.)))
      YPL=SQRT(ABS(REW/(1.+(PPL-BFPLUS)*YPL/2.+GPL*YPL/2.)))
      UPL=REW/YPL
      TPLUS=1.+GPL*UPL+(PPL-BFPLUS)*YPL
      IF(TPLUS.LT.0.0)TPLUS=0.0
      IF(TPLUS.LT.0.1)LTPL=1
      IF(TPLUS.LT.0.1)LSUB=2
      DUDY=TPLUS
C --------------------------------------------------- SECTION SEVEN ----
C.....WALL SHEAR STRESS AND FRICTION FACTOR ARE COMPUTED HERE
  300 IF(YPL.LT.YPMIN.AND.MODE.EQ.2)LSUB=1
      IF(YPL.GT.YPMAX)LSUB=2
      IF(LSUB.NE.2)CLOCFX=DPDX
      BETA=DUDY*YPL/UPL
C.....THE FOLLOWING IS AN APPROXIMATE CORRECTION FOR USE
C.....OF PLANE WALL FUNCTION EQUATIONS FOR AXI-SYMMETRIC PROBLEMS
      RADRAT=(R(NP2)+R(NP1))/(2.*R(NP3))
      IF (KIN.EQ.1) RADRAT=(R(2)+R(3))/(2.*R(1))
      TAUW=ABS(RADRAT*RHW*UI*UI/(UPL*UPL*GC))
      TAU=TAUW*TPLUS*GC
      IF(UGG.LT.0.001)GO TO 310
      CF2=GC*TAUW/(RHG*UGG*UGG)
  310 CONTINUE
      IF(NEQ.EQ.1)GO TC 340
C.....
      DO 330 J=1,NPH
```

```
      IF (REW.GE.4.) GC TO 320
C ---------------------------------------------- SECTION EIGHT ----
C.....THIS SECTION IS USED IF THE NUMERICAL INTEGRATION
C.....OF THE COUETTE FLOW EQUATIONS IS BYPASSED
      HPS(J)=PRW(J)*(YPL+GPL*YPL*YPL/2.)
      DHY(J)=PRW(J)*(1.+GPL*HPS(J))
      PREFW=PRW(J)
C ---------------------------------------------- SECTION NINE ----
C.....WALL HEAT TRANSFER AND STANTON NUMBER ARE COMPUTED HERE
  320 GAMA(J)=DHY(J)*YFL/HPS(J)
      QWF(J)=RADRAT*RHW*UI/(UPL*HPS(J))
      IF(SOURCE(J).EQ.2)GAMA(J)=0.0
      IF(SOURCE(J).EQ.2.AND.MODE.EQ.2)FI(J)=(AKK*AKK*EE*BETA*UI/AQ)**2
      IF(SOURCE(J).EQ.2)GO TO 328
      IF(INDI(J).EQ.1.CR.INDE(J).EQ.1)GO TO 325
      IF(KEX.EQ.1)GO TO 322
      F(J,1)=(FI(J)+AJI(J)/QWF(J))/(1.+AMI/QWF(J))
      HW(J)=F(J,1)
      QW(J)=AJI(J)-AMI*HW(J)
      IF(FLUID.EQ.2)CALL PROP2(1,F(J,1),T(1),VISCO(1),PR(J,1),RHO(1))
C.....IF VARIABLE PROPERTY ROUTINE OTHER THAN 2 IS USED, CHANGE THIS
C.....CALL AS APPROPRIATE.  ALSO AFTER STATEMENT 325
      GO TO 326
  322 F(J,NP3)=(FI(J)-AJE(J)/QWF(J))/(1.-AME/QWF(J))
      HW(J)=F(J,NP3)
      QW(J)=-AJE(J)+AME*HW(J)
      IF(FLUID.EQ.2)CALL PROP2(NP3,F(J,NP3),T(NP3),VISCO(NP3),PR(J,NP3),
     1RHO(NP3))
      GO TO 326
  325 CONTINUE
      QW(J)=QWF(J)*(HW(J)-FI(J))
  326 IF(ABS(HG(J)-HW(J)).LT.0.000001.OR.UGG.LT..001)ST(J)=0.0
      IF(ABS(HG(J)-HW(J)).LT.0.000001.OR.UGG.LT..001)GO TO 328
      ST(J)=QW(J)/(RHG*UGG*(HW(J)-HG(J)))
      IF(KEX.EQ.1.AND.INDE(J).EQ.1)QW(J)=-QW(J)
  328 IF(KIN.EQ.1)PREF(J,2)=PREFW
      IF(KEX.EQ.1)PREF(J,NP1)=PREFW
  330 CONTINUE
C ---------------------------------------------- SECTION TEN ----
  340 KSTART=KSTART+1
      IF(INTG.EQ.1.AND.KSTART.LT.4)GO TO 40
      MARKER=0
      IF(LSUB.GT.0)MARKER=1
      IF (LSUB.EQ.0) RETURN
C.....
C.....LAMSUB ROUTINE
C.....IF LSUB EQUALS 1, (SEE CONDITIONS IN MAIN PROGRAM),
C.....THIS ROUTINE HAS AS ITS FUNCTION THE DELETION OF THE FIRST
C.....GRID LINE NEAREST THE WALL.  IN EFFECT IT COMBINES THE
C.....FIRST TWO SPACES AND REDUCES THE NUMBER OF SPACES, N, BY ONE.
C.....ALTERNATIVELY, IF LSUB EQJALS 2, IT INSERTS ANOTHER GRID
C.....POINT BETWEEN 1 AND 3.
      INTGE=INTG-1
      WRITE(6,930)
  930 FORMAT(/' ROUTINE LAMSUB HAS BEEN CALLED')
      IF(LSUB.GT.1) GO TO 390
      N=N-1
      WRITE(6,920)N,INTGE
      NP1=N+1
      NP2=N+2
```

```
      NP3=N+3
      IF(KIN.EQ.1)GO TO 360
      U(NP3)=U(N+4)
      U(NP2)=U(N+3)
      Y(NP3)=Y(N+4)
      Y(NP2)=Y(N+3)
      IF(NEQ.EQ.1)GO TC 460
      DO 350 J=1,NPH
      F(J,NP3)=F(J,N+4)
  350 F(J,NP2)=F(J,N+3)
      GO TO 460
  360 CONTINUE
      DO 380 I=3,NP3
      U(I)=U(I+1)
      Y(I)=Y(I+1)
      IF(NEQ.EQ.1)GO TO 380
      DO 370 J=1,NPH
  370 F(J,I)=F(J,I+1)
  380 CONTINUE
      GO TO 460
  390 N=N+1
      WRITE(6,920)N,INTGE
C.....CHANGE IF PRCGRAM DIMENSIONING IS CHANGED. ************************
      IF(N.GT.50)GC TO 470
      NP1=N+1
      NP2=N+2
      NP3=N+3
      IF(KIN.EQ.1)GO TC 420
      Y(NP3)=Y(NP2)
      Y(NP2)=Y(NP1)
      U(NP3)=U(NP2)
      U(NP2)=U(NP1)
      YI=0.5*(Y(NP2)+Y(N))
      Y(NP1)=0.5*(Y(N)+YI)
      U(NP1)=0.5*(U(N)+UI)
      IF(NEQ.EQ.1)GO TO 410
      DO 400 J=1,NPH
      F(J,NP3)=F(J,NP2)
      F(J,NP2)=F(J,NP1)
  400 F(J,NP1)=0.5*(F(J,N)+FI(J))
  410 CONTINUE
      GO TO 455
  420 DO 440 K=1,N
      I=NP3+(1-K)
      U(I)=U(I-1)
      Y(I)=Y(I-1)
      IF(NEQ.EQ.1) GC TO 440
      DO 430 J=1,NPH
  430 F(J,I)=F(J,I-1)
  440 CONTINUE
      YI=0.5*(Y(2)+Y(3))
      Y(3)=0.5*(YI+Y(3))
      U(3)=0.5*(UI+U(3))
      IF(NEQ.EQ.1) GO TO 455
      DO 450 J=1,NPH
  450 F(J,3)=0.5*(FI(J)+F(J,3))
  455 EMU(NP1)=EMU(N)
      BF(NP3)=BF(NP2)
      VISCO(NP3)=VISCO(NP2)
      RHO(NP3)=RHO(NP2)
```

```
      T(NP3)=1.
      IF (NPH.EQ.0) GO TO 460
      DO 458 J=1,NPH
  458 PR(J,NP3)=PR(J,NP2)
  460 CONTINUE
      IF(LTPL.EQ.1)WRITE(6,62)
      RETURN
  470 WRITE (6,940)
      LVAR=6
   62 FORMAT(' LSUB=2 WAS INVOKED BECAUSE SHEAR STRESS RATIO'/
     1' IN WALL FUNCTION GOT LESS THAN 0.1, DUE PROBABLY'/
     2' TO EXCESSIVE PRESSURE GRADIENT OR SUCTION'/)
  940 FORMAT(//' PROGRAM TERMINATED BECAUSE N HAS EXCEEDED THE'/
     1' NUMBER OF FLOW TUBES FOR WHICH THE PROGRAM IS DIMENSIONED'//)
  920 FORMAT(1X,17HN HAS SHIFTED TO ,I2,1X,9HAT INTG =,I4/)
      RETURN
      END


      SUBROUTINE AUX
C.....
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      COMMON/GEN/PEI,AMI,AME,DPDX,XU,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,NXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SOURCE,RETRAN,NUMRUN,SPACE,RHO,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,QWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     6/W/SC(54),AU(54),BU(54),CU(54),A(5,54),B(5,54),C(5,54),SU(5,54),SD
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     8/L1/YL,UMAX,UMIN,FR,YIP,YEM,ENFRA,KENT,AUXM2
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHOM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AUX1(100),AUX2(100),YPMAX
     3/ADD/RBOM(54),CMD(54),ROMD(54),ITKE
      DIMENSION DV(54)
C.....
       ITKE=1
      UGG=U(1)+FLOAT(KEX-1)*(U(NP3)-U(1))
      RHG=RHO(1)+FLOAT(KEX-1)*(RHO(NP3)-RHO(1))
      AMW=-AME+FLOAT(KEX-1)*(AME+AMI)
      RHW=RHO(NP3)+FLOAT(KEX-1)*(RHO(1)-RHO(NP3))
      VISW=VISCO(NP3)+FLOAT(KEX-1)*(VISCO(1)-VISCO(NP3))
      UTAU=SQRT(GC*TAUW/RHW)
      YPUT=RHW*UTAU/VISW
      IF (INTG.GT.1) GO TO 10
      KOUNT=0
      IF (MODE.EQ.2) KOUNT=1
    1 RAVG=R(1)
      RHOAV=RHO(1)
      VISAV=VISCO(1)
      KTURB=0
      IF (NPH.EQ.0.AND.MODE.EQ.2.AND.K2.NE.2)WRITE(6,6)
      IF(K2.EQ.2.AND.MODE.EQ.2)WRITE(6,9)
      IF (NPH.EQ.0.AND.MODE.EQ.2.AND.KD.LT.2) WRITE(6,7)
      IF (NPH.EQ.0.AND.MODE.EQ.2.AND.KD.GE.2) WRITE(6,8)
      IF (NPH.EQ.0) GO TO 10
      JTKE=0
      DO 5 J=1,NPH
      IF (SOURCE(J).EQ.2) JTKE=J
```

```
      IF (J.EQ.JTKE.AND.MODE.EQ.2) WRITE (6,4)
      IF (J.EQ.JTKE.AND.MODE.EQ.2.AND.K2.EQ.2) WRITE (6,3)
    3 FORMAT(//' K2 SHOULD NOT BE SET EQUAL TO 2 '//)
    4 FORMAT(' FLOW IS TURBULENT AND PROGRAM IS USING TURBULENT'
     1/' KINETIC-ENERGY TO EVALUATE EDDY VISCOSITY, EXCEPT IN THE'/
     1' WALL FUNCTION WHERE MIXING-LENGTH IS USED.  NOTE THAT THE'/
     1' PRINTED-OUT VALUES OF TKE HAVE NO MEANING IN THE NEAR-WALL'/
     1' REGION, I.E., FOR Y+ LESS THAN B+, OR 2*A+.'/)
    5 IF (SOURCE(J).EQ.2) KTURB=1
      IF (MODE.EQ.1) GO TO 10
      IF (KTURB.EQ.0.AND.K2.NE.2)WRITE (6,6)
      IF (KD.LT.2) WRITE (6,7)
      IF (KD.GE.2) WRITE (6,8)
    6 FORMAT(' FLOW IS TURBULENT AND PROGRAM IS USING THE PRANDTL MIX-'
     1/' ING-LENGTH HYPOTHESIS TO EVALUATE EDDY-VISCOSITY'/)
    7 FORMAT(' THE VAN DRIEST SCHEME IS BEING USED TO EVALUATE'
     1/' THE MIXING-LENGTH OR LENGTH-SCALE DAMPING NEAR THE WALL.'/)
    8 FORMAT(' THE EVANS SCHEME IS BEING USED TO EVALUATE THE'
     1/' MIXING-LENGTH OR LENGTH-SCALE DAMPING NEAR THE WALL.'/)
    9 FORMAT(' FLOW IS TURBULENT AND PROGRAM IS USING THE CONSTANT'/
     1' EDDY DIFFUSIVITY OPTION IN THE OUTER REGION'/)
C.....
   10 DO 100 I=2,NP1
      YM=0.5*(Y(I+1)+Y(I))
      IF (KEX.EQ.1) YM=Y(NP3)-YM
      IF (FLUID.EQ.1) GO TO 12
      RAVG=0.5*(R(I)+R(I+1))
      RHOAV=0.5*(RHO(I)+RHO(I+1))
      VISAV=0.5*(VISCO(I)+VISCO(I+1))
   12 EMUT=0.
      DV(I)=1.
      IF (MODE.EQ.1) GO TO 50
      KOUNT=KOUNT+1
      IF (KOUNT.EQ.1) GO TO 1
      IF (KASE.EQ.2) GO TO 25
C------------------------------------ EDDY VISCOSITY DAMPING TERM ----
C.....VAN DRIEST DAMPING FUNCTION
C.....APL, BPL COMPUTED IN WALL
      IF (FLUID.NE.1) YPUT=SQRT(RHOAV*TAUW*GC)/VISAV
      YLOC=YM*YPUT
      IF (KD.GT.1) GO TO 15
      IF (YLOC/APL.GT.10.) GO TO 25
      DV(I)=1.-1./EXP(YLOC/APL)
      GO TO 22
C.....EVANS DAMPING FUNCTION
   15 DV(I)=YLOC/BPL
   20 IF(DV(I).GT.1.) DV(I)=1.
C.....LOWER LIMIT VALUE DAMPING TERM
   22 IF (DV(I).LT.0.0001) DV(I)=0.0001
   25 CONTINUE
C------------------------------------------------ PRANDTL MIXING LENGTH ----
      IF (I.GT.2) GO TO 30
      IF (GEOM.EQ.4.OR.GEOM.EQ.5) GO TO 30
      IF (REM.LE.100..OR.K2.EQ.3) GO TO 30
C.....EMPIRICAL CORRELATION FOR ALMG FOR WALL FLOWS
C.....THIS CORRELATION THEN OVERRIDES THE INPUT ALMGG
      AMOR=AME/RHO(1)
      IF (KIN.EQ.1) AMOR=AMI/RHO(NP3)
      ALMG=ALMGG*2.942/REM**0.125*(1.-67.5*AMOR/UGU)
      IF (ALMG.LT.ALMGG) ALMG=ALMGG
```

```
C.....COMPUTE MIXING LENGTH
   30 AL=ALMG*YL
      IF (KASE.EQ.1.AND.YM.LT.AL/AK) AL=AK*YM
      IF(KASE.EQ.1.AND.K2.EQ.2)AL=AK*YM
      IF (KTURB.EQ.1.AND.KASE.EQ.2) GO TO 40
      IF(KASE.EQ.2)GO TO 35
      YTKE=Y(I)*YPUT
      IF (KEX.EQ.1)YTKE=(Y(NP3)-Y(I+1))*YPUT
      IF(KTURB.EQ.1.AND.KD.LE.1.AND.YTKE.GE.2.*APL)GO TO 40
      IF(KTURB.EQ.1.AND.KD.GE.2.AND.YTKE.GE.BPL)GO TO 40
   35 EMUT=RHOAV*AL*AL*ABS((U(I+1)-U(I))/(Y(I+1)-Y(I)))*DV(I)*DV(I)
      IF(K2.NE.2.OR.KASE.EQ.2)GO TO 36
      EMUTC=(AQ*REM**BQ)*VISAV
      IF(EMUT.GT.EMUTC)EMUT=EMUTC
      IF(YM.GT.0.4*YL)EMUT=EMUTC
   36 IF(KTURB.NE.1)GO TO 50
C.....ADJUSTMENT OF TKE IN NEAR-WALL REGION
      FJJAVE=((AK*EMUT)/(AQ*RHOAV*AL*DV(I)))**2
      F(JTKE,I)=FJJAVE
      ITKE=I
      IF (KEX.EQ.1.AND.ITKE.EQ.1) ITKE=I
      GO TO 50
C.....COMPUTE EDDY VISCOSITY USING TURBULENT KINETIC ENERGE EQN
   40 FJJAVE=ABS(0.5*(F(JTKE,I+1)+F(JTKE,I)))
      EMUT=AQ*RHOAV*AL*DV(I)*SQRT(FJJAVE)/AK
C---------------------------------------------- EFFECTIVE VISCOSITY ----
   50 EMU(I)=EMUT+VISAV
      IF (NPH.EQ.0.AND.KASE.EQ.1) T(I)=ABS(EMU(I)*(U(I+1)-U(I))/
     1(Y(I+1)-Y(I)))/(GC*TAUW)
      IF (NPH.EQ.0) GO TO 100
C-------------------------------- TURBULENT PRANDTL/SCHMIDT NUMBER ----
      EDR=EMUT/VISAV
      DO 90 J=1,NPH
      IF (MODE.EQ.1) GO TO 80
      JPHI=1
      IF (SOURCE(J).GT.0) JPHI=SOURCE(J)
      GO TO (62,68,62,62), JPHI
C.....
C.....STAGNATION ENERGY EQN, TURBULENT PRANDTL NUMBER
   62 PRTJ=PRT(J)
      IF (KASE.EQ.2.OR.K3.EQ.3) GO TO 70
C.....THE FOLLOWING IS THE FREE CONSTANT IN THE TURBULENT PRANDTL
C.....NUMBER EQUATION. EXPERIENCE MAY SUGGEST A DIFFERENT VALUE.
      CT = 0.2
      PETC=EDR*CT*(PR(J,I+1)+PR(J,I))/2.
      IF(PETC.LT..001)PETC=.001
      IF(PETC.GT.100.)GO TO 69
      ALPHA=SQRT(1./PRTJ)
      AOP=ALPHA/PETC
      IF(AOP.GT.10.)AOP=10.
      PRTJ=1./(1./(2.*PRTJ)+ALPHA*PETC-PETC*PETC*(1.-EXP(-AOP)))
      GO TO 69
C.....TURBULENT KINETIC ENERGY EQN, TURB PRANDTL NUMBER
   68 PRTJ=PRT(J)
C------------------------------- EFFECIVE PRANDTL/SCHMIDT NUMBER ----
   69 IF(KIN.EQ.1.AND.I.EQ.2)GO TO 90
      IF(KEX.EQ.1.AND.I.EQ.NP1)GO TO 90
   70 PREF(J,I)=(1.0+EDR)/(EDR/PRTJ+1.0/(0.5*(PR(J,I+1)+PR(J,I))))
      GO TO 90
C.....LAMINAR EFFECTIVE PRANDTL NUMBER
```

```
   80 PREF(J,I)=0.5*(PR(J,I+1)+PR(J,I))
   90 CONTINUE
  100 CONTINUE
      DO 110 I=2,NP1
      RHOAV=(RHO(I)+RHO(I+1))/2.
      RAVG=(R(I)+R(I+1))/2.
C.....ADJUSTMENT OF EMU AT 2.5 AND N+1.5
      IF (I.GT.2) GO TO 110
      IF (KIN.NE.1) GO TO 105
      IF (BETA.LT.0.02.OR.BETA.GT.0.9) GO TO 110
      EMU(2)=TAU*(Y(2)+Y(3))/(BETA*(U(2)+U(3)))
  105 IF (KEX.NE.1) GO TO 110
      IF (BETA.LT.0.02.OR.BETA.GT.0.9) GO TO 110
      EMU(NP1)=TAU*(Y(NP3)-0.5*(Y(NP1)+Y(NP2)))/
     1         (BETA*0.5*(U(NP1)+U(NP2)))
C.....COMPUTE SMALL C'S
  110 SC(I)=RAVG*RAVG*RHOAV*0.5*(U(I+1)+U(I))*EMU(I)/(PEI*PEI)
      IF (NEQ.EQ.1) GO TO 300
C------------------------------------------------------ SOURCE TERMS ----
      DO 200 I=3,NP1
      DO 150 J=1,NPH
      SU(J,I)=0.
      SD=0.
      IF (SOURCE(J).EQ.0) GO TO 150
      NSOR=SOURCE(J)
      GO TO (115,130,115,120), NSOR
C---------------------------------------- STAGNATION ENERGY EQN SOURCE ----
  115 IF (I.EQ.2) PREF(J,1)=PREF(J,2)
      PREFF=(PREF(J,I)+PREF(J,I-1))*0.5
      CS=SC(I)*(U(I+1)*U(I+1)-U(I)*U(I))*ROMD(I)
      CS = CS-SC(I-1)*(U(I)*U(I)-U(I-1)*U(I-1))*ROMD(I-1)
      CS=(1.-1./PREFF  )*CS*RBOM(I)
      SU(J,I) = CS/(GC*CJ)+BF(I)/(CJ*RHO(I))
  120 IF (U(I).LT.0.0001) GO TO 125
      IF (SOURCE(J).EQ.3) SU(J,I)=SU(J,I)+AUXM2/(RHO(I)*U(I))
      IF (SOURCE(J).EQ.4) SU(J,I)=AUXM2/(RHO(I)*U(I))
  125 SD=0.
      GO TO 150
C----------------------- TURBULENT KINETIC ENERGY EQUATION SOURCE ----
  130 AL=ALMG*YL
      IF (KASE.EQ.2) GO TO 140
      YMQ=Y(I)
      IF (KEX.EQ.1) YMQ=Y(NP3)-Y(I)
      IF (YMQ.LT.AL/AK) AL=AK*YMQ
  140 DU2DOM=.5*((U(I+1)*U(I+1)-U(I)*U(I))*ROMD(I)+
     1  (U(I)*U(I)-U(I-1)*U(I-1))*ROMD(I-1))
      DVQ=.5*(DV(I)+DV(I-1))
      FJ2=ABS(F(J,I))
      PROD=AQ*AL*DVQ*SQRT(FJ2)*(RHO(I)*R(I)/PEI)**2/(U(I)*AK*4.)*
     1 DU2DOM**2
      DISS=BQ*AK*FJ2**1.5/(AL*DVQ*U(I))
      IF(DISS*DX.GT.FJ2)DISS=FJ2/DX
      SU(J,I)=PROD-DISS
      FGT=F(J,NP3)
      IF(KIN.EQ.2)FGT=F(J,1)
      IF(KIN.EQ.3)FGT=0.0
      IF(F(J,I).LT.FGT)SU(J,I)=PROD
      SD=0.0
      GO TO 150
C.....ADD OTHER SOURCE FUNCTIONS HERE
```

```
C.....CHANGE 'COMPUTED GO TO' STATEMENT TO INCLUDE
C.....SOURCE FUNCTION STATEMENT NUMBERS.  LIKEWISE,
C.....CHANGE TURBULENT PR/SC NUMBER 'COMPUTED GO TO '
C.....STATEMENT NUMBERS.
  150 CONTINUE
  200 CONTINUE
  300 CONTINUE
      RETURN
      END


      SUBROUTINE OUT
C.....
      DIMENSION UPLUS(54),YPLUS(54),HP(54),QRAT(54)
      INTEGER FLAG,FLAG2
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      COMMON/GEN/PEI,AMI,AME,DPDX,XU,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,AXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SOURCE,RETRAN,NUMRUN,SPACE,RWD,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,QWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHCM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AJX1(100),AUX2(100),YPMAX
C.....
      GO TO (100,200,300,400,500,600), OUTPUT
  100 CONTINUE
      GO TO 1000
  200 CONTINUE
C.....
C.....THIS ROUTINE WORKS PROPERLY ONLY FOR KIN=1.  IT IS DESIGNED
C.....PRIMARILY FOR EXTERNAL MOMENTUM AND THERMAL BOUNDARY LAYERS,
C.....WITH OR WITHOUT THE TURBULENT KINETIC ENERGY EQUATION.  TO WORK
C.....PROPERLY, IT MUST BE THE LAST EQUATION SOLVED.
      IF(KIN.NE.1)GO TO 600
      IF(INTG.NE.1)GO TO 205
      IF(INTG.EQ.1)KSPACE=SPACE
      IF(KSPACE.EQ.11.OR.KSPACE.EQ.21)SPACE=1
      IF(NPH.EQ.0)ST(1)=0.0
      IF(NPH.EQ.0)ST(2)=0.
      IF(NPH.EQ.1.AND.SOURCE(1).EQ.2)ST(1)=0.
      IF(NPH.EQ.1.AND.SOURCE(1).EQ.2)REH=0.
      IF(NPH.EQ.0)REH=0.0
      IF(NPH.EQ.0)F(1,1)=0.
      IF(INTG.EQ.1) FLAG=1
      IF(INTG.EQ.1)FLAG2=1
  205 FAM=AMI/(RHO(NP3)*U(NP3))
      STA=ST(1)
      G=(H-1.)/(H*SQRT(CF2))
      BTA=-H*REM*CAY/CF2
      IF(SOURCE(1).EQ.2.AND.NPH.GT.1)STA=ST(2)
      IF(XD.GE.XL)GO TO 210
      IF(INTG.NE.FLAG) GO TO 278
  210 CONTINUE
      NINTG=INTG-1
      IF(INTG.EQ.1.AND.KSPACE.NE.11)GO TO 215
      IF(INTG.EQ.1)WRITE(6,282)
      IF(INTG.EQ.2.AND.KSPACE.EQ.11)WRITE(6,282)
```

```
      IF(INTG.EQ.2.AND.KSPACE.EQ.21)WRITE(6,282)
      IF(SPACE.NE.1)GO TO 215
      IF(KSPACE.EQ.1)GO TO 215
      CPL=APL
      IF(KD.GT.1)CPL=BPL
      WRITE(6,284)NINTG,XU,UGU,CAY,FAM,REM,CF2,H,REH,STA,F(1,1),CPL,AME
      IF(KSPACE.EQ.11.AND.XD.LT.XL)GO TO 279
      IF(INTG.EQ.FLAG2)GO TO 215
      IF(KSPACE.EQ.21.AND.XD.LT.XL)GO TO 279
      IF(XD.GE.XL) UGU=UGD
  215 CONTINUE
      WRITE(6,280)
      WRITE(6,282)
      CPL=APL
      IF(KD.GT.1)CPL=BPL
      WRITE(6,284)NINTG,XU,UGU,CAY,FAM,REM,CF2,H,REH,STA,F(1,1),CPL,AME
      IF(K1.GT.10)WRITE(6,286)(SP(I),I=1,5),G,BTA
  286 FORMAT(1X,23HSPECIAL OUTPUT - SP(1)=,E9.3,1X,6HSP(2)=,E9.3,1X,6HSP
     1(3)=,E9.3,1X,6HSP(4)=,E9.3,1X,6HSP(5)=,E9.3,1X,2HG=,
     2F5.2,1X,5HBETA=,F5.2)
  288 FORMAT(/,5X,64H  I         Y(I)        U(I)       YPLUS(I)     UPLUS(I)
     1     HPLUS(I),5X,10HSQRT(K)/UG,/)
      IF(NEQ.GT.1)WRITE(6,288)
      IF(NEQ.EQ.1)WRITE(6,290)
  290 FORMAT(/,5X,66H  I         Y(I)        U(I)       YPLUS(I)     UPLUS(I)
     1        TAUPLUS ,/)
  292 FORMAT(6X,I2,4X,F8.6,2X,F7.2,3X,F8.2,6X,F5.2,9X,F9.4,5X,F7.4,
     15X,F6.3,2X,F7.3)
      YPUT=U(NP3)*SQRT(CF2*RHO(NP3)*RHO(1))/VISCO(1)
      UPUT=1./(U(NP3)*SQRT(CF2*RHO(NP3)/RHO(1)))
      IF(NPH.EQ.0)GO TO 293
      IF(ABS((F(1,1)-F(1,NP3))*ST(1)).LT..0001)GO TO 293
      IF (NEQ.GT.1.AND.SOURCE(1).NE.2)HPUT=SQRT(CF2*RHO(1)/RHO(NP3))/
     1 ((F(1,1)-F(1,NP3))*ST(1))
C.....
C.....CHANGE MU IF DIMENSIONING CHANGED **********************************
  293 MU=54
C.....
      DO 274 I=1,NP3
      M=I
      YPLUS(I)=Y(I)*YPUT
      UPLUS(I)=U(I)*UPUT
      IF(I.NE.2)GO TO 245
      GO TO (240,225,220,225,225),NEQ
  220 IF(SOURCE(1).EQ.2)GO TO 235
      IF(SOURCE(2).NE.2)GO TO 225
      QRAT(MU)=SQRT(ABS(FI(2)))/U(NP3)
  225 IF(SOURCE(1).EQ.2)GO TO 235
      HP(MU)=0.0
      IF(ABS((F(1,1)-F(1,NP3))*ST(1)).LT..0001)GO TO 230
      HP(MU)=(F(1,1)-FI(1))*HPUT
  230 F(1,MU)=FI(1)
      GO TO 240
  235 QRAT(MU)=SQRT(ABS(FI(1)))/U(NP3)
  240 Y(MU)=0.5*(Y(2)+Y(3))
      U(MU)=UI
      YPLUS(MU)=0.5*(Y(2)+Y(3))*YPUT
      UPLUS(MU)=UI*UPUT
      M=MU
  245 IF(I.EQ.NP2)GO TO 274
```

```
      GO TO (250,255,265,255,255),NEQ
  250 TAUPL=1.0
      IF(I.NE.1.AND.M.NE.MU)TAUPL=0.5*(T(M)+T(M-1))
      IF(M.EQ.MU)TAUPL=TAU/(GC*TAUW)
      IF(I.EQ.NP3)TAUPL=0.0
      WRITE(6,292) M,Y(M),U(M),YPLUS(M),UPLUS(M),TAUPL
      GO TO 274
  255 IF(SOURCE(1).EQ.2)GO TO 272
      HP(I)=0.0
      IF(ABS((F(1,1)-F(1,NP3))*ST(1)).LT..0001)GO TO 260
      HP(I)=(F(1,1)-F(1,I))*HPUT
  260 WRITE(6,292) M, Y(M),U(M),YPLUS(M),UPLUS(M)
     1,HP(M)
      GO TO 274
  265 IF(SOURCE(1).EQ.2)GO TO 272
      IF(SOURCE(2).NE.2)GO TO 255
      QRAT(I)=SQRT(ABS(F(2,I)))/U(NP3)
      HP(I)=0.0
      IF(ABS((F(1,1)-F(1,NP3))).LT..0001)GO TO 270
      HP(I)=(F(1,1)-F(1,I))*HPUT
  270 WRITE(6,292) M,Y(M),U(M),YPLUS(M),UPLUS(M),HP(M),QRAT(M)
      GO TO 274
  272 QRAT(I)=SQRT(ABS(F(1,I)))/U(NP3)
      DUMMY=0.0
      WRITE(6,292)M,Y(M),U(M),YPLUS(M),UPLUS(M),DUMMY,QRAT(M)
  274 CONTINUE
      WRITE(6,280)
      IF(XD.GT.XL)GO TC 276
      IF(INTG.EQ.1)GO TC 276
      IF(KSPACE.EQ.11.OR.KSPACE.EQ.21)WRITE(6,282)
  276 CONTINUE
  280 FORMAT(//)
      FLAG2=FLAG2+KSPACE-1
  279 FLAG=FLAG + SPACE
  278 CONTINUE
  282 FORMAT(/,115H  INTG      XJ        UGU         K              F        REM
     1  CF2        H           REH          ST       F(1,WALL) APL OR BPL   AME)
  284 FORMAT(3X,I3,2X,F7.4,2X,F7.2,1X,E10.3,2X,F6.4,1X,F7.1,2X,F8.6,2X,
     1F5.3,3X,F7.1,2X,F8.6,2X,F8.2,2X,F6.2,2X,F8.4)
      RETURN
C.....
  300 CONTINUE
      GO TO 1000
  400 CONTINUE
C.....
C.....THIS OUTPUT ROUTINE IS DESIGNED PRIMARILY FOR FLOW IN A TUBE.
      IF(INTG.NE.1)GO TO 404
      KSPACE=SPACE
      IF(NPH.GT.0)GO TO 403
      ST(1)=0.0
      PR(1,7)=0.0
      F(1,NP3)=0.0
      SOURCE(1)=0
  403 FLAG=1
      FLAG2=1
  404 IF(KSPACE.EQ.11.OR.KSPACE.EQ.21)SPACE=1
      IF(XD.GE.XL)GO TO 405
      IF(INTG.NE.FLAG)GO TO 425
  405 CONTINUE
  480 FORMAT(2X,5HINTG=,I3,1X,3HXU=,F6.3,1X,3HRE=,F9.1,
```

```
     11X,4HCF2=,F7.5,1X,3HST=,F7.5,1X,3HNU=,F7.2,1X,3HUM=,F7.2,1X,
     13HFM=,F9.2,1X,6HPRESS=,F10.3,1X,3HFW=,F8.2)
      NINTG=INTG-1
      J=1
      IF(SOURCE(J).EQ.2.AND.NPH.EQ.2)J=2
      IF(SOURCE(J).EQ.2.AND.NPH.EQ.1)ST(1)=0.0
      ANU=ST(J)*PR(J,7)*REM
C.....NOTE THAT NUSSELT HERE IS CALCULATED FROM I=7  PR, WHEREAS THE
C.....OTHER PARAMETERS ARE BASED ON MIXED MEAN TEMPERATURE.
      WRITE(6,480)NINTG,XU,REM,CF2,ST(J),ANU,UGJ,FMEAN,PRO,F(J,NP3)
      IF(K1.GT.10)WRITE(6,482)(SP(I),I=1,5)
  482 FORMAT(12X,23HSPECIAL OUTPUT - SP(1)=,E10.3,1X,6HSP(2)=,E10.3,1X,6
     1HSP(3)=,E10.3,1X,6HSP(4)=,E10.3,1X,6HSP(5)=,E10.3)
      IF(XD.GT.XL)GC TC 410
      IF(KSPACE.EQ.11)GC TO 420
      IF(INTG.EQ.FLAG2)GO TO 410
      IF(KSPACE.EQ.21)GC TO 420
  484 FORMAT(/,5X,45H  I        Y(I)       U(I)       F(1,I)      F(2,I)
     1,5X,'YPL',5X,'UPL',7X,'EDR',6X,'T(I)'/)
  410 WRITE(6,484)
  486 FORMAT(6X,I2,3X,F9.6,2X,F7.2,F10.2,F10.2,3X,F8.2,2X,F6.2,
     14X,F6.2,4X,F7.2)
      DUM=0.0
      YPUT=UGU*SQRT(CF2)
      DO 415 I=1,NP3
      IF(KEX.EQ.1)YPL=(Y(NP3)-Y(I))*RHO(NP3)*YPUT/VISCO(NP3)
      IF(KIN.EQ.1)YPL=(Y(I)-Y(1))*RHO(1)*YPUT/VISCO(1)
      UPL=U(I)/YPUT
      IF(NPH.EQ.1)F(2,I)=0.0
      IF(I.GT.2.AND.I.LT.NP2)EDR=(EMU(I)+EMU(I-1))/(2.*VISCO(1))
      IF(I.LT.3.OR.I.GT.NP1)EDR=1.0
      IF(NPH.EQ.0)WRITE(6,486)I,Y(I),U(I),DUM,DUM,YPL,UPL,EDR,T(I)
      IF(NPH.GT.0)WRITE(6,486)I,Y(I),U(I),F(1,I),F(2,I),YPL,UPL,EDR,T(I)
  415 CONTINUE
      WRITE(6,488)
  488 FORMAT(//)
      FLAG2=FLAG2+KSPACE-1
  420 CONTINUE
      FLAG=FLAG+SPACE
  425 CONTINUE
      RETURN
C.....
  500 CONTINUE
      GO TO 1000
  600 CONTINUE
C  THIS IS A GENERAL PURPOSE OUTPUT ROUTINE
C.....
      IF(INTG.EQ.1)FLAG=1
      FAM=0.0
      IF(XD.GE.XL)GO TO 605
      IF(INTG.NE.FLAG)GC TO 620
  605 CONTINUE
      NINTG=INTG-1
  680 FORMAT(//,2X,5HINTG=,I3,2X,3HXU=,F8.5,2X,4HPEI=,F8.5,2X,4HAMI=,
     1F8.4,2X,4HAME=,F8.4,2X,9HPRESSURE=,F9.3,2X,5HBETA=,F7.4,2X,
     22HK=,E10.3)
  682 FORMAT(12X,4HREM=,F9.1,2X,4HREH=,F9.1,1X,4HCF2=,F8.6,2X,9HA OR BPL
     1=,F6.2,2X,2HH=,F6.3,2X,7HRHO(1)=,F7.4,2X,9HRHO(NP3)=,F7.4)
  684 FORMAT(12X,28HDISPLACEMENT OF I-SURFACE = ,F7.5)
  686 FORMAT(12X,9HST(J)=    ,5F9.6)
```

```
  688 FORMAT(/'      I      Y(I)      R(I)       OM(I)      U(I)         EMU(I)
     1  T(I)      F(1,I)      F(2,I)      F(3,I)      F(4,I)      F(5,I)'/)
  690 FORMAT(4X,I2,1X,F8.6,2X,F7.4,3X,F7.5,1X,F7.2,2X,F10.7,2X,F6.1,2X,
     15F10.3)
  692 FORMAT(4X,I2,1X,F8.6,2X,F7.4,3X,F7.5,1X,F7.2,2X,F10.7,2X,F6.3)
      WRITE(6,680)NINTG,XU,PEI,AMI,AME,PRO,BETA,CAY
      CPL=APL
      IF(KASE.EQ.2) GO TO 610
      IF(KD.GT.1)CPL=BPL
      IF(KASE.EQ.1)WRITE(6,682)REM,REH,CF2,CPL,H,RHO(1),RHO(NP3)
  694 FORMAT(12X,2HF=,F6.3,2X,7HVWPLUS=,F7.4,2X,6HPPLUS=,F7.4,2X,8HTAUWA
     1LL=,F9.6)
      IF(KASE.NE.1)GO TO 610
      IF(KIN.EQ.1.AND.U(NP3).GT.0.001)FAM=AMI/(U(NP3)*RHO(NP3))
      IF(KEX.EQ.1.AND.U(1).GT.0.001)FAM=AME/(U(1)*RHO(1))
      WRITE(6,694)FAM,CPL,PPL,TAUW
  610 IF(GEOM.EQ.9)WRITE(6,684)RWO
      EMU(1)=0.0
      IF(KASE.EQ.1.AND.NPH.NE.0)WRITE(6,686)(ST(J),J=1,NPH)
      IF(KASE.EQ.1.AND.NPH.NE.0)WRITE(6,696)(QW(J),J=1,NPH)
  696 FORMAT(12X,8HQWALL=  ,5F10.5)
  698 FORMAT(12X,9HGAMA(J)= ,5F9.5)
      IF(KASE.EQ.1.AND.NPH.NE.0)WRITE(6,698)(GAMA(J),J=1,NPH)
      IF(K1.GT.10)WRITE(6,699)(SP(I),I=1,5)
  699 FORMAT(12X,23HSPECIAL OUTPUT - SP(1)=,E10.3,1X,6HSP(2)=,E10.3,1X,6
     1HSP(3)=,E10.3,1X,6HSP(4)=,E10.3,1X,6HSP(5)=,E10.3)
      WRITE(6,688)
      EMU(NP2)=0.0
      EMU(NP3)=0.0
      DO 615 I=1,NP3
      IF(NPH.EQ.0)WRITE(6,692)I,Y(I),R(I),OM(I),U(I),EMU(I),T(I)
      IF(NPH.GT.0)WRITE(6,690)I,Y(I),R(I),OM(I),U(I),EMU(I),T(I),
     1(F(J,I),J=1,NPH)
  615 CONTINUE
      FLAG=FLAG+SPACE
  620 CONTINUE
      RETURN
 1000 CONTINUE
      RETURN
      END


      SUBROUTINE PROP2(K,FX,TI,VISCOI,PRA,RHOA)
C.....
C.....THIS PROGRAM CALULATES THE PROPERTIES OF AIR AT ABSOLUTE STATIC
C.....ENTHALPY DETERMINED FROM F(J,I) AND U(I).  IT IS ESSENTIALLY A
C.....TABULATION OF THE ECKERT AND DRAKE TABLES.     IT IS ASSUMED
C.....IN USING THIS SUBROUTINE THAT THE DEPENDENT VARIABLE IN THE
C.....THERMAL ENERGY ECUATION IS STAGNATION ENTHALPY.
C..... HERE:
C.....    K=1 IMPLIES START WITH LOWEST TABULATED STATIC ENTHALPY
C.....     =2 IMPLIES START WITH PREVIOUSLY USED TABULATED STATIC.....
C.....          ENTHALPY
C.....    HI=ABSOLUTE STATIC ENTHALPY(B/LBM)
C.....    PRE=STATIC PRESSURE (LBF/SQ.FT.)
C.....    RHOA=CALCULATED DENSITY (LBM/(CU.FT.))
C.....    VISCOI=CALCULATED DYNAMIC VISCOSITY (LBM/(SEC.FT.))
C.....    PRA=CALCULATED PRANDTL NUMBER
C.....    TI=CALCULATED TEMPERATURE (DEG. RANKINE)
C.....
```

```
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC
      COMMON/GEN/PEI,AMI,AME,DPDX,XJ,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,NXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SOURCE,RETRAN,NUMRUN,SPACE,RWD,PPLAG,OUTPUT,DELTAX,GV
     3/V/U(54),F(5,54),R(54),OM(54),Y(54),UGJ,UGD,UI,FI(5),FMEAN,TAUW
     4/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
      DIMENSION HA(34),TA(34),VS(34),PA(34)
      DATA HA(1),HA(2),HA(3),HA(4),HA(5),HA(6),HA(7),HA(8),HA(9),HA(10),
     1HA(11),HA(12),HA(13),HA(14),HA(15),HA(16),HA(17),HA(18),HA(19),
     2HA(20),HA(21),HA(22),HA(23),HA(24),HA(25),HA(26),HA(27),HA(28),
     3HA(29),HA(30),HA(31),HA(32),HA(33),HA(34)/42.89,64.43,85.97,
     4107.50,129.06,150.68,172.39,194.25,216.26,238.50,260.97,283.68,
     5306.65,329.88,353.37,377.11,401.09,425.29,449.71,499.17,549.35,
     6600.16,651.51,703.35,755.61,808.28,861.28,914.61,968.21,
     71022.09,1076.20,1130.56,1185.11,1276.47/
      DATA TA(1),TA(2),TA(3),TA(4),TA(5),TA(6),TA(7),TA(8),TA(9),TA(10),
     1TA(11),TA(12),TA(13),TA(14),TA(15),TA(16),TA(17),TA(18),TA(19),
     2TA(20),TA(21),TA(22),TA(23),TA(24),TA(25),TA(26),TA(27),TA(28),
     3TA(29),TA(30),TA(31),TA(32),TA(33),TA(34)/180.0,270.0,360.0,450.0,
     4540.0,630.0,720.0,810.0,900.0,990.0,1080.0,1170.0,1260.0,1350.0,
     51440.0,1530.0,1620.0,1710.0,1800.0,1980.0,2160.0,2340.0,2520.0,
     62700.0,2880.0,3060.0,3240.0,3420.0,3600.0,3780.0,3960.0,4140.0,
     74320.0,4620.0/
      DATA VS(1),VS(2),VS(3),VS(4),VS(5),VS(6),VS(7),VS(8),VS(9),VS(10),
     1VS(11),VS(12),VS(13),VS(14),VS(15),VS(16),VS(17),VS(18),VS(19),
     2VS(20),VS(21),VS(22),VS(23),VS(24),VS(25),VS(26),VS(27),VS(28),
     3VS(29),VS(30),VS(31),VS(32),VS(33),VS(34)/46.53,69.10,89.30,107.4,
     4124.1,139.4,153.6,166.9,179.5,191.4,202.8,213.5,223.9,233.9,243.6,
     5253.0,262.0,270.3,279.0,295.5,310.9,325.8,339.8,353.3,366.8,379.2,
     6391.5,402.9,416.8,430.1,439.8,451.3,461.1,475.0/
      DATA PA(1),PA(2),PA(3),PA(4),PA(5),PA(6),PA(7),PA(8),PA(9),PA(10),
     1PA(11),PA(12),PA(13),PA(14),PA(15),PA(16),PA(17),PA(18),PA(19),
     2PA(20),PA(21),PA(22),PA(23),PA(24),PA(25),PA(26),PA(27),PA(28),
     3PA(29),PA(30),PA(31),PA(32),PA(33),PA(34)/0.770,0.753,0.739,0.722,
     40.708,0.697,0.689,0.683,0.680,0.680,0.680,0.682,0.684,0.686,0.689,
     50.692,0.696,0.699,0.702,0.706,0.714,0.722,0.726,0.734,0.741,0.749,
     60.759,0.767,0.783,0.803,0.831,0.863,0.916,0.972/
      HI=FX-(U(K)*U(K))/(2.0*GC*CJ)
      IF(HA(34).LT.HI.OR.HA(1).GT.HI)LVAR=7
      IF (LVAR.EQ.7) WRITE (6,6)
      IF(HA(34).LT.HI)HI=HA(34)
      IF(HA(1).GT.HI)HI=HA(1)
    6 FORMAT(//' ENTHALPY IS OUT OF THE RANGE OF'/
     1' VALUES TABULATED IN PROP2'//)
      IF(K.EQ.1) L=1
      DO 1 I=L,34
      IF(HA(I).GT.HI) GO TO 2
    1 CONTINUE
    2 M=I-1
      IF(HA(M).LE.HI) GO TO 5
      DO 3 J=1,M
      MB=M-J
      IF(HA(MB).LE.HI) GO TO 4
    3 CONTINUE
    4 M=MB
      I=M+1
    5 L=I
      TI=TA(M)+(TA(I)-TA(M))*(HI-HA(M))/(HA(I)-HA(M))
      VISCOI=(VS(M)+(VS(I)-VS(M))*(HI-HA(M))/(HA(I)-HA(M)))*0.0000001
      PRA=PA(M)+(PA(I)-PA(M))*(HI-HA(M))/(HA(I)-HA(M))
```

```
      PRES=PRE
      IF(LSUB.GT.0) PRES=PRO
      RHOA=PRES/(53.34*TI)
      RETURN
      END

      SUBROUTINE INPUT(KERROR)
C.....
      INTEGER GEOM,FLUID,SOURCE(5),SPACE,BODFOR,OUTPUT,TYPBC,TITLE(18)
      COMMON/GEN/PEI,AMI,AME,DPDX,XU,XD,XL,DX,INTG,CSALFA,TYPBC(5),
     1MODE,PRT(5),PRE,NXBC,X(100),RW(100),FJ(5,100),GC,CJ,AM(100),PRO,
     2UG(100),PO,SOURCE,RETRAN,NUMRUN,SPACE,RWD,PPLAG,OUTPUT,DELTAX,GV
     3/E/N,NP1,NP2,NP3,NEQ,NPH,KEX,KIN,KASE,KRAD,GEOM,FLUID,BODFOR,YPMIN
     4/GG/BETA,GAMA(5),AJI(5),AJE(5),INDI(5),INDE(5),TAU,QWF(5)
     5/V/U(54),F(5,54),R(54),OM(54),Y(54),UGU,UGD,UI,FI(5),FMEAN,TAUW
     6/W/SC(54),AU(54),BU(54),CJ(54),A(5,54),B(5,54),C(5,54),SU(5,54),SD
     7/L/AK,ALMG,ALMGG,FRA,APL,BPL,AQ,BQ,EMU(54),PREF(5,54),AUXM1
     8/L1/YL,UMAX,UMIN,FR,YIP,YEM,ENFRA,KENT,AUXM2
     9/P/RHO(54),VISCO(54),PR(5,54),RHOC,VISCOC,PRC(5),T(54),RHCM,BF(54)
     1/O/H,REM,CF2,ST(5),LSUB,LVAR,CAY,REH,PPL,GPL,QW(5),KD
     2/CN/AXX,BXX,CXX,DXX,EXX,K1,K2,K3,SP(54),AUX1(100),AUX2(100),YPMAX
C.....
C.....EACH 'READ' STATEMENT IS INDICATED BY THE SUMBOLS ****************
C.....ALL INTEGERS ARE IN FIELDS OF 5 SPACES. BE SURE TO JUSTIFY TO
C.....RIGHT. ALL DECIMAL NUMBERS ARE IN SUCCESSIVE FIELDS OF 10 SPACES.
C ************************** READ IN A TITLE OF UP TO 72 CHARACTERS
      READ(5,505)TITLE
C.....
      WRITE(6,506)TITLE
C..... THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....    GEOM= GENERAL STATEMENT OF THE SYSTEM GEOMETRY
C.....        =1 IMPLIES AXI-SYMMETRIC BODY--RADIUS NOT INCLUDED IN
C.....                   BOUNDARY LAYER EQUATIONS, APPLICABLE TO EITHER
C.....                   INTERNAL OR EXTERNAL BOUNDARY LAYERS WHERE
C.....                   BOUNDARY LAYER THICKNESS IS SMALL RELATIVE TO
C.....                   BODY RADIUS.  ALSO APPLICABLE TO FLAT-PLATE
C.....                   GEOMETRY (SIMPLY SET RW(M) CONSTANT).
C.....        =2 IMPLIES AXI-SYMMETRIC BODY--RADIUS INCLUDED IN
C.....                   BOUNDARY LAYER EQUATIONS, APPLICABLE ONLY TO
C.....                   EXTERNAL BOUNDARY LAYERS (KIN=1,KEX=2)
C.....        =3 IMPLIES AXI-SYMMETRIC BOUNDARY LAYER, APPLICABLE ONLY
C.....                   TO INTERNAL BOUNDARY LAYERS  (KIN=2,KEX=1)
C.....        =4 IMPLIES CIRCULAR TUBE-FLOW PROBLEM (KIN=3,KEX=1)
C.....        =5 IMPLIES FLOW BETWEEN PARALLEL PLANES, SYMMETRICAL
C.....           BOUNDARY CONDITIONS (KIN=3,KEX=1)
C.....        =6 IMPLIES AXIALLY-SYMMETRIC JETS
C.....        =7 IMPLIES AXIALLY-SYMMETRIC FREE SHEAR FLOW
C.....        =8 IMPLIES TWO-DIMENSIONAL SYMMETRIC JET
C.....        =9 IMPLIES TWO-DIMENSIONAL FREE SHEAR FLOW
C.....    MODE=  TYPE OF FLOW SYSTEM CONSIDERED INITIALLY
C.....        =1 IMPLIES LAMINAR FLOW
C.....        =2 IMPLIES TURBULENT FLOW
C.....           NOTE: IF MODE = 1 THE PROGRAM AUTOMATICALLY CHANGES
C.....           TO TURBULENT FLOW WHEN THE MOMENTUM THICKNESS RE
C.....           NUMBER EXCEEDS VALUES INSERTED AS 'RETRAN' BELOW.
C.....    FLUID= TYPE OF MAINSTREAM FLUID/SELECTS APPROPRIATE SUBROU-
C.....           TINES
C.....         =1 IMPLIES CONSTANT PROPERTY FLUID
C.....         =2 IMPLIES AIR AT MODERATE TEMPERATURES
```

```
C.....         =3(OR HIGHER) IMPLIES OTHER FLUIDS NOT YET SPECIFIED.
C.....    NEQ= NUMBER CF CONSERVATION EQUATIONS CONSIDERED
C.....         INCLUDING MOMENTJM EQUATION
C.....    N= NUMBER OF STRIPS ACROSS LAYER (LIMITED TO 50, THIS VERSION)
C.....    KEX= DEFINES TYPE OF BOUNDARY AT ARBITRARILY NOMINATED
C.....         EXTERNAL BOUNDARY
C.....    KIN= DEFINES TYPE OF BOUNDARY AT ARBIRARILY NOMINATED
C.....         INTERNAL BOUNDARY
C.....    KIN,KEX= 1 IMPLIES WALL BOUNDARY
C.....           = 2 IMPLIES FREE BOUNDARY
C.....           = 3 IMPLIES LINE OF SYMMETRY
C.....    KENT= 0 IF ENTRAINMENT IS BASED ON MOMENTUM EQUATION ONLY.
C.....        = 1 IF ENTRAINMENT IS BASED ON ALL EQUATIONS.
C ***********************************************************************
C (INTEGERS)
      READ(5,585) GEOM,MODE,FLUID,NEQ,N,KEX,KIN,KENT
C.....
      WRITE(6,510)
      IF(N.GT.40)WRITE(6,980)
      WRITE(6,520) GEOM,MODE,FLUID,NEQ,N,KEX,KIN
      IF(GEOM.EQ.4.OR.GEOM.EQ.5)GO TO 20
      IF(KENT.EQ.0)WRITE(6,525)
      IF(KENT.GT.0)WRITE(6,526)
C..... THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....    XU= INITIAL VALUE OF X CHOSEN TO DEFINE POSITION OF INITIAL
C.....        PROFILES.  TYPICALLY XU=0.0, BUT NEED NOT BE.
C.....    XL= VALUE CF X WHERE COMPUTATIONS ARE TERMINATED
C.....    DELTAX= MAXIMUM STEP IN X-DIRECTION, EXPRESSED AS FRACTION OF
C.....            BOUNDARY LAYER THICKNESS (SUGGEST 0.5, BUT CAN BE MADE
C.....            MUCH LARGER FOR CONSTANT PROPERTY FLOWS AND LAM FLOWS)
C.....    RETRAN = MOMENTUM THICKNESS REYNOLDS NUMBER (OR DIAMETER
C.....             REYNOLDS NUMBER IN TUBE-FLOW PROBLEM) AT WHICH
C.....             TRANSITION FROM LAMINAR TO TURBULENT BOUND-
C.....             ARY LAYER IS DESIRED (USE DUMMY NUMBER IF
C.....             PROBLEM IS ALL TURBULENT.) (SUGGEST 200.0)
C.....    FRA= FRACTION FOR DETERMINATION OF DX TO NEXT POSITION (SUG-
C.....         GESTED VALUE=0.05).
C.....    ENFRA= DESIRED FRACTIONAL DIFFERENCE BETWEEN FREE-STREAM AND
C.....           NEXT-TO-LAST GRID POINT.  CONTROLS ENTRAINMENT RATE.
C.....           (SUGGESTED VALUE=0.005).  THIS VALUE IS RELATED TO THE
C.....           CHOSEN GRID SPACING.  IN SOME CASES 0.01 WORKS BETTER,
C.....           BUT WITH A FINE GRID IT MAY BE NECESSARY TO GO AS LOW
C.....           AS 0.001.  IF THERE IS NO FREE-STREAM
C.....           LEAVE ENFRA BLANK, OR USE ANY DUMMY NUMBER.
C.....    GV= GRAVITY CONSTANT, POSITIVE IN POSITIVE X DIRECTION.
C.....        LEAVE 0.0 OR BLANK IF GRAVITY IS NOT CONSIDERED.
C ***********************************************************************
C (DECIMAL NUMBERS)
   20 READ(5,580)XU,XL,DELTAX,RETRAN,FRA,ENFRA,GV
C.....
      WRITE(6,540)
      WRITE(6,550) XU,XL,DELTAX,RETRAN,FRA,ENFRA,GV
      NPH=NEQ-1
      KASE=1
      IF(KIN.NE.1.AND.KEX.NE.1)KASE=2
C.....THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....BODFOR= TYPE OF BODY-FORCE (OTHER THAN PRESSURE GRADIENT)
C.....       =0 IMPLIES NO EXTRA BODY FORCES
C.....       =1 IMPLIES FREE-CONVECTION BODY-FORCE.
C.....       =2 IMPLIES AN EXTERNAL BODY-FORCE (IN ADDITION TO FREE
```

```
C.....         CONVECTICN), INTRODUCED THRU AUX1(M).
C....SOURCE(J)= TYPE OF SOURCE FUNCTION IN THE DIFFUSION EQUATIONS.
C.....         = 0  IMPLIES NO SOURCE FUNCTION.
C.....         = 1  IMPLIES VISCOUS DISSIPATION, PLUS WORK OF ANY BODY
C.....              FORCES, IN THE ENERGY EQUATION.
C.....         = 2  IMPLIES THE SOURCE FUNCTION FOR THE TURBULENT ENERGY
C.....              EQUATION.  SETTING SOURCE EQUAL TO 2 FOR ANY DIFFUS-
C.....              ION EQUATION AUTOMATICALLY MAKES THAT EQUATION BE THE
C.....              TURBULENT KINETIC ENERGY EQUATION, AND AT THE SAME
C.....              TIME THE EDDY VISCOSITY AND EDDY CONDUCTIVITIES WILL
C.....              BE CALCULATED BY THE TURBULENT KINETIC ENERGY METHOD
C.....              INSTEAD OF FROM THE MIXING LENGTH.
C.....         = 3  IMPLIES VISCOUS DISSIPATION PLUS AN EXTERNAL VOLUME
C.....              SOURCE, INTRODUCED THROUGH AUX2(M), PLUS BODY FORCE
C.....              WORK, IN THE ENERGY EQUATION.  AUX2(M) HAS DIMENSIONS
C.....              (ENERGY)/(VOLUME*TIME).
C.....         = 4  IMPLIES AN EXTERNAL VOLUME SOURCE, INTRODUCED THROUGH
C.....              AUX2(M).  DIMENSIONS, (QUANTITY)/(VOLUME*TIME).
C.....SOURCE WILL NOT BE READ UNLESS NEQ IS GREATER THAN 1.
C *************************** READ EITHER OF THE NEXT TWO
C (INTEGERS)
      IF(NEQ.GT.1)READ(5,585) BODFOR,(SOURCE(J),J=1,NPH)
      IF(NEQ.EQ.1)READ(5,585)BODFOR
C.....
      WRITE(6, 820)
      IF(NEQ.GT.1)WRITE(6, 830) BODFOR,(SOURCE(J),J=1,NPH)
      IF(NEQ.EQ.1)WRITE(6,830)BODFOR
C..... THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....    PO= INITIAL FREESTREAM STATIC PRESSURE
C.....    RHOC=DENSITY OF CONSTANT PROPERTY FLUID
C.....    VISCOC=VISCOSITY OF CONSTANT PROPERTY FLUID(IF ENGLISH UNITS,
C.....           USE DYNAMIC VISCOSITY, LBM/(SEC-FT)).
C.....    PRC = PRANDTL NUMBER OF CONSTANT PROPERTY FLUID (FOR TURBULENT
C.....          KINETIC ENERGY EQUATION USE PRC=1.00)
C..... (THE CONSTANT PRCPERTIES MAY BE OMITTED IF FLUID NOT EQUAL 1)
      IF(FLUID.EQ.2)WRITE(6,800)
      IF(FLUID.EQ.1)WRITE(6,750)
C ********************* READ ONLY ONE OF THE FOLLOWING THREE
C (DECIMAL NUMBERS)
      IF(FLUID.NE.1)READ(5,580)PO
      IF(FLUID.EQ.1.AND.NEQ.GT.1)READ(5,580) PO,RHOC,VISCOC,(PRC(J),J=1,
     1NPH)
      IF(FLUID.EQ.1.AND.NEQ.EQ.1)READ(5,580)PO,RHOC,VISCOC
C.....
      WRITE(6,700)
      WRITE(6, 900) PO
      IF(FLUID.NE.1)GO TO 50
      WRITE(6,680)
      IF(NPH.EQ.0) GO TO  40
      WRITE(6,690) RHOC,VISCOC,(PRC(J),J=1,NPH)
      GO TO  50
   40 WRITE(6, 870) RHOC,VISCOC
   50 CONTINUE
      WRITE(6,770)
C.....BOUNDARY CONDITIONS ALONG I AND E BOUNDARIES (ONLY ONE MAY BE
C.....A WALL)
C..... THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....    NXBC= NUMBER OF POINTS USED TO SPECIFY BOUNDARY CONDITIONS AT
C.....          EITHER INTERNAL OR EXTERNAL BOUNDARY
C.....    TYPBC(J)= IMPLIES TYPE OF BOUNDARY CONDITION GIVEN FOR THE
```

```
C.....                J-TH CONSERVED QUANTITY AT A WALL SURFACE
C.....              =1 IMPLIES LEVEL SPECIFICATION
C.....              =2 IMPLIES FLUX SPECIFICATION
C.....                 NOTE: FOR THE TURBULENT KINETIC ENERGY EQUATION
C.....                 USE TYPBC(J)=1
C.....      (TYPBC WILL NOT BE READ UNLESS NEQ IS GREATER THAN 1)
C ****************** READ ONLY ONE OF THE FOLLOWING THREE
C.....NOTE: KASE=1 MEANS A WALL; KASE=2 MEANS THERE ARE NO WALLS.
C (INTEGERS)
      IF(KASE.EQ.1.AND.NEQ.GT.1)READ(5,585) NXBC,(TYPBC(J),J=1,NPH)
      IF(KASE.EQ.1.AND.NEQ.EQ.1)READ(5,585)NXBC
      IF(KASE.EQ.2)READ(5,585)NXBC
C.....
      IF(KASE.EQ.2.OR.NEQ.EQ.1)GO TO 70
      WRITE(6,570)
      WRITE(6,590) NXBC,(TYPBC(J),J=1,NPH)
      GO TO 80
   70 WRITE(6, 890)
      WRITE(6, 585) NXBC
C..... THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....    X(M)= POSITION AT WHICH THE BOUNDARY VALUES ARE GIVEN.  NOTE
C.....          THAT X(1) MUST BE LESS THAN (OR EQUAL TO) XU, AND THE
C.....          LARGEST VALUE OF X(M) MUST BE GREATER THAN
C.....          (OR EQUAL TO) XL.
C.....    RW(M)= DISTANCE FROM AXIS OF SYMMETRY TO BODY SURFACE.
C.....           SET= CONSTANT IF PLANE BOUNDARY LAYER. (SUGGEST 1.0)
C.....           FOR GEOM=4, RW(M) IS THE PIPE RADIUS, MAY VARY WITH X.
C.....           FOR GEOM=5, RW(M) IS THE HALF-WIDTH OF THE DUCT, WHICH
C.....           MAY BE A FUNCTION OF X.
C.....           FOR GEOM=6,8, OR,9, RW(M) IS TOTALLY A DUMMY.
C.....           FOR GEOM=7, RW(M) IS THE INITIAL RADIUS OF THE I-
C.....           BOUNDARY, BUT IS A DUMMY THEREAFTER.
C.....    AUX1(M), AUX2(M)= AUXILIARY FUNCTIONS, FOR SPECIAL PURPOSES:
C.....             INTERPOLATED VALUES WILL APPEAR IN THE COMMON AS AUXM1
C.....             & AUXM2 IF THERE IS A WALL.  LEAVE COLUMN BLANK IF
C.....             NOT USED.
   80 CONTINUE
      IF(NXBC.LT.2)WRITE(6,920)
      DO 90 M=1,NXBC
C *************************************************************************
C (DECIMAL NUMBERS, IN THE FORM OF A TABLE.)
   90 READ(5,580) X(M),RW(M),AUX1(M),AUX2(M)
C.....
C.....THE QUANITIES READ AT THIS POINT ARE DEFINED:
C.....UG(M)= FREESTREAM VELOCITY AT POSITION X(M).   IF BOTH THE
C.....        I AND E SURFACES ARE FREE-STREAM BOUNDARIES. UG(M) IS
C.....        THE FREE-STREAM VELOCITY ON THE E-SIDE AND MUST START
C.....        OUT THE SAME AS U(NP3).
C.....           (IF THERE IS NO FREE-STREAM READ IN A DUMMY
C.....           NUMBER FOR UG, OR ELSE LEAVE A BLANK)
C.....AM = MASS FLUX AT WALL, POSITIVE IN THE POSITIVE DIRECTION OF Y
C.....     (AM WILL NOT BE READ UNLESS THERE IS A WALL.)
C.....FJ(J,M)= VALUE OF PROPERTY OR FLUX OF PROPERTY AT BOUNDARY
C.....    NOTE THAT IF FJ IS A PROPERTY AT THE
C.....    WALL (SUCH AS ENTHALPY), TYPBC(J) , ABOVE, MUST BE EQUAL
C.....    TO 1.  IF FJ IS A FLUX AT THE WALL (SUCH AS HEAT FLUX),
C.....    TYPBC(J) MUST BE SET EQUAL TO 2.  IN THE LATTER CASE, FJ
C.....    IS THE TOTAL FLUX OF THE PROPERTY IN QUESTION, I.E., THAT
C.....    EVALUATED AT THE 'T-STATE' CONTROL SURFACE.  THIS BECOMES OF
C.....    PARTICULAR SIGNIFICANCE WHEN THERE IS MASS TRANSFER AT THE
```

```
C.....     SURFACE. IF FJ IS A WALL FLUX, IT SHOULD BE POSITIVE
C.....     IN THE POSITIVE DIRECTION OF THE COORDINATE SYSTEM.
C.....     FOR THE TURBULENT KINETIC ENERGY EQUATION SET FJ=0.0
C.....     (FJ WILL NOT BE READ UNLESS THERE IS A WALL, AND WILL NOT
C.....     BE READ IF NEC IS 1)
C.....     NOTE: IF FJ IS A WALL FLUX, AND IS ZERO (ADIABATIC WALL), SOME
C.....     ERROR MAY BE INTRODUCED BECAUSE THE DEPENDENT VARIABLE IN THE
C.....     WALL FUNCTION IS NORMALIZED WITH RESPECT TO THE WALL FLUX.
C.....     IT IS BETTER TO INTRODUCE A SMALL WALL FLUX. (SUGGEST 0.0001)
C.....(NOTE THAT M IS AN INTEGER VARYING FROM 1 TO NXBC.)
      WRITE(6,600)
      DO 110 M=1,NXBC
C *************************READ ONLY ONE OF THE FOLLOWING THREE.
C (DECIMAL NUMBERS, IN THE FORM OF A TABLE)
      IF(KASE.EQ.1.AND.NEQ.GT.1)READ(5,580)UG(M),AM(M),(FJ(J,M),J=1,NPH)
      IF(KASE.EQ.1.AND.NEQ.EQ.1)READ(5,580)UG(M),AM(M)
      IF(KASE.EQ.2)READ(5,580)UG(M)
C.....
      IF(KASE.EQ.1.AND.NEQ.GT.1)WRITE(6,610) M,X(M),RW(M),UG(M),AM(M),
     1AUX1(M),AUX2(M),(FJ(J,M),J=1,NPH)
      IF(KASE.EQ.1.AND.NEQ.EQ.1)WRITE(6,610)M,X(M),RW(M),UG(M),AM(M),
     1AUX1(M),AUX2(M)
  110 IF(KASE.EQ.2)WRITE(6,610)M,X(M),RW(M),UG(M),AUX1(M),AUX2(M)
      NP1=N+1
      NP2=N+2
      NP3=N+3
C.....INITIAL PROFILE SPECIFICATION
C.....          THE INITIAL VELOCITY PROFILE ESTABLISHES THE GRID SPACING
C.....          AND THUS SOME CARE SHOULD BE EXERCISED IN LAYING IT OUT.
C.....          THE PROGRAM IS NOT PARTICULARLY SENSITIVE TO UNEVENNESS
C.....          IN THE Y-INCREMENTS, BUT BIG CHANGES IN DELTA-Y SHOULD
C.....          BE AVOIDED.
C.....          FOR TURBULENT FLOW NEAR A WALL THE VALUE OF
C.....          U*Y*DENSITY/VISCOSITY AT THE FIRST POINT NEXT TO THE
C.....          WALL SHOULD BE NOT LESS THAN ABOUT 200, UNLESS IT IS DESIRED
C.....          TO BY-PASS THE WALL FUNCTION.  IN THAT CASE THIS VALUE
C.....          SHOULD BE LESS THAN 1.0, AND ABOUT 20 POINTS RATHER EVENLY
C.....          SPACEDSHOULD BE USED OUT TO YPLUS EQUAL ABOUT 20.0.
C.....     Y(I)= DISTANCE ALONG NORMAL TO BOUNDARY
C.....               NOTE THAT Y IS MEASURED FROM THE I-BOUNDARY,
C.....               I.E., Y(1)= 0.0.
C.....     U(I)= VELOCITY IN X-DIRECTION AT Y(I)
C.....     F(J,I)= VALUE OF CONSERVED QUANTITY AT Y(I)
C.....                  NOTE: FOR THE TURBULENT KINETIC ENERGY EQUATION USE
C.....                  F(J,I)=0.0 AT THE WALL (IF ANY).  THE REMAINDER OF
C.....                  THE INITIAL TURBULENT KINETIC PROFILE DEPENDS ON THE
C.....                  PROBLEM SPECIFICATIONS.  IT CAN BE ALL ZERO.
      WRITE(6,760)
      WRITE(6,630)
C ********************** READ IN A TABLE OF Y AND U, OR Y, U, AND F'S.
C (DECIMAL NUMBERS)
      IF(NEQ.EQ.1)GO TO 240
      READ(5,580) Y(1),U(1),(F(J,1),J=1,NPH)
      DO  220 I=3,NP1
  220 READ(5,580) Y(I),U(I),(F(J,I),J=1,NPH)
      READ(5,580) Y(NP3),U(NP3),(F(J,NP3),J=1,NPH)
      GO TO 255
  240 READ(5,580) Y(1),U(1)
      DO  250 I=3,NP1
  250 READ(5,580) Y(I),U(I)
```

```
      READ(5,580) Y(NP3),U(NP3)
C.....
  255 NDUMB=1
      IF(NEQ.EQ.1)GO TO 265
      WRITE(6, 850)NDUMB,Y(1),U(1),(F(J,1),J=1,NPH)
      DO 230 I=3,NP1
  230 WRITE(6, 850)I,Y(I),U(I),(F(J,I),J=1,NPH)
      WRITE(6, 850)NP3,Y(NP3),U(NP3),(F(J,NP3),J=1,NPH)
      GO TO 270
  265 WRITE(6, 850) NDUMB,Y(1),U(1)
      DO 260 I=3,NP1
  260 WRITE(6, 850) I,Y(I),U(I)
      WRITE(6, 850) NP3,Y(NP3),U(NP3)
  270 CONTINUE
C.....TURBULENT TRANSPORT CONSTANTS
C.....IF LAMINAR B.L. ONLY, READ IN DUMMY DATA.
C.....IF THERE IS NO WALL, READ IN DUMMY VALUES FOR AK, APL, BPL
C..... THE QUANTITIES READ AT THIS POINT ARE DEFINED:
C.....      AK= MIXING LENGTH CONSTANT KAPPA ( SUGGESTED VALUE=0.41)
C.....      ALMGG= VALUE OF LAMBDA=YL/YG (TRY 0.085)(FOR A
C.....      BOUNDARY LAYER ON A WALL THIS VALUE IS OVERRIDEN AT LOW
C.....      REYNOLDS NUMBERS (BELOW APPROXIMATELY 6000) EXCEPT WHEN K2=3.
C.....      FOR PIPE-FLOW TRY 0.07) (WHEN THE CONSTANT EDDY
C.....      DIFFUSIVITY OPTION IS USED THIS NUMBER IS A DUMMY)
C.....      FR= DEFINES BOUNDARY LAYER THICKNESS (99% POINT=0.01) USED IN
C.....      THE DEFINITION OF ALMG (SUGGESTED VALUE=0.01).
C.....      AQ,BQ= CONSTANTS IN THE TURBULENT KINETIC ENERGY EQUATION, OR
C.....             CONSTANTS IN THE EDDY DIFFUSIVITY EQUATION.  WHEN THE
C.....             CONSTANT EDDY DIFFUSIVITY OPTION IS USED,
C.....             SET K2=2. (FOR PIPE-FLOW TRY K2=2, AQ=.005, BQ=0.9)
C.....             NOTE: TO BE CONSISTENT FOR TURBULENT K.E., AK MUST BE
C.....             EQUAL TO (AQ**.75)/(BQ**.25)
C.....             (SUGGEST 0.22 AND 0.377 FOR TKE.  ALSO SUGGEST USE
C.....             PRT(J)=1.7 FOR THE TURBULENT KINETIC ENERGY EQUATION.)
C.....      YPMAX= MAXIMUM VALUE OF YPLUS TO BE ALLOWED AT OUTER EDGE OF
C.....             WALL FUNCTION (SAY 50.0 FOR TURBULENT BL; USE 1.0 IF
C.....             DESIRED TO BYPASS WALL FUNCTION, BUT THEN SET
C.....             YPMIN = 0.0.  FOR STRONG PRESSURE GRADIENTS IT IS MORE
C.....             ACCURATE TO SET YPMAX NO GREATER THAN 15 SINCE THE DE-
C.....             PARTURE FROM COUETTE FLOW OCCURS AT VERY LOW Y+.  BEST
C.....             ACCURACY IS OBTAINED WHEN YMAX=1.0 AND YPMIN=0.0, BUT
C.....             THE NUMBER OF FLOW TUBES MAY THEN BE VERY LARGE.)
C.....      YPMIN= MINIMUM VALUE OF YPLUS TO BE ALLOWED AT OUTER EDGE
C.....             OF WALL FUNCTION FOR A TURBULENT BL (CAN BE 0.0)
C ***********************************************************************
C (DECIMAL NUMBERS)
      READ(5,580) AK,ALMGG,FR,AQ,BQ,YPMAX,YPMIN
C.....
C  READ IN A+ OR B+.  IF A+ IS GREATER THAN B+, PROGRAM WILL USE VAN
C  DRIEST SCHEME FOR SUBLAYER, AND B+ IS READ AS MERELY A DUMMY NUM-
C  BER.  IF B+ IS GREATER THAN A+, PROGRAM WILL USE THE EVANS SCHEME.
C  THE PROGRAM WILL USE AN INTERNAL EMPIRICAL CORRELATION FOR EFFECTS
C  OF PRESSURE GRADIENT, TRANSPIRATION, ETC., BUT THIS ADDITIONAL COR-
C  RECTION CAN BE SUPPRESSED IF DESIRED BY SETTING 'SIGNAL' AT ANY NUM-
C  BER EQUAL TO 1.0.       SUGGEST A+=25 FOR FLAT SURFACE, 26 FOR FLOW
C  INSIDE A CIRCULAR TUBE.
C ***********************************************************************
C (DECIMAL NUMBERS)
      READ(5,580)APL,BPL,SIGNAL
C.....
```

```
      KD=0
      IF(APL.GE.BPL.ANC.SIGNAL.GE.1.0)KD=1
      IF(BPL.GE.APL)KD=2
      IF(BPL.GE.APL.ANC.SIGNAL.GE.1.0)KD=3
C.....THE QUANTITIES READ AT THIS POINT ARE DEFINED:
C.....     PPLAG= A LAG CONSTANT IN THE EFFECTIVE VALUE OF PPLUS, GPLUS,
C.....            USED IN THE EVALUATION OF APL, OR BPL.
C.....            (SUGGESTED VALUE = 4000.)
C.....     PRT(J) = TURBULENT TRANSFER RATIO FOR F(J) (TRY .86)
C.....              NEAR A WALL THIS VALUE IS OVERRIDEN INSIDE THE
C.....              PROGRAM UNLESS K3 IS SET EQUAL TO 3,
C.....              SEE INFORMATION ON K3 BELOW.
C **************************** READ ONE OF THE FOLLOWING TWO
C (DECIMAL NUMBERS)
      IF(NEQ.GT.1)READ(5,580) PPLAG,(PRT(J),J=1,NPH)
      IF(NEQ.EQ.1)READ(5,580)PPLAG
C.....
      WRITE(6,780)
      WRITE(6,640)
      WRITE(6,650) AK,ALMGG,FR,PPLAG,AQ,BQ,YPMAX,YPMIN
      IF(PPLAG.LT.400.)WRITE(6,990)
      IF(BQ.LE.0.0)GO TC 275
      IF(NPH.LT.1)GO TC 275
      AKCHEC=(AQ**.75)/(BQ**.25)
      AKERR=ABS(AK-AKCHEC)
      K20=0
      DO 274 J=1,NPH
  274 IF(SOURCE(J).EQ.2)K20=1
      IF(AKERR/AK.GT.0.01.AND.K20.EQ.1)WRITE(6,710)
  275 WRITE(6,930)
      WRITE(6,940)APL,BPL,SIGNAL
      IF(KD.EQ.0)WRITE(6,950)
      IF(KD.EQ.2)WRITE(6,955)
      IF(NEQ.GT.1)WRITE(6,660)
      IF(NEQ.GT.1)WRITE(6,650) ( PRT(J),J=1,NPH)
C.....READ IN CONVERSICN CONSTANTS, AND ANY OTHER ARBITRARY DECIMAL CON-
C.....STANTS THAT ARE CESIRED.
C.....      GC= 32.2  IN BRITISH SYSTEM
C.....      CJ= 778.0 IN BRITISH SYSTEM
C.....IF YOU USE A SYSTEM SUCH AS MKS, GC=1.0 AND CJ=1.0.  JUST USE
C.....A CONSISTENT SYSTEM.  THE PROGRAM WORKS IN REAL WORLD DIMEN-
C.....SIONS, NOT NCNDIMENSIONAL VARIABLES.  BE CAREFUL ABOUT THE
C.....DIMENSIONS OF VISCOSITY -- IN ENGLISH UNITS USE LBM/(SEC-FT).
C.....THE CONSTANTS AXX, BXX, ETC.,MAY BE LEFT BLANK IF THEY ARE NOT
C.....BEING EMPLOYEC FCR SOME SPECIAL PURPOSE INSIDE THE PROGRAM.
C ***********************************************************************
C (DECIMAL NUMBERS)
      READ(5,580) GC,CJ,AXX,BXX,CXX,DXX,EXX
C.....
      WRITE(6,790)
      WRITE(6,720)
      WRITE(6, 910) GC,CJ,AXX,BXX,CXX,DXX,EXX
C.....READ IN THE NUMBER OF RUNS OF DATA THAT YOU WANT USED (NUMRUN),
C.....AND THE SPACING (NUMBER OF INTEGRATIONS) OF THE OUTPUT DATA THAT
C.....YOU WANT PRINTED (SPACE).  IF YOU SET SPACE=11, AN ABBREVIATED
C.....DATA SET WILL BE PRINTED OUT, OMITTING ALL PROFILES, BUT INCLUDING
C.....ALL OTHER DATA AT EACH INTEGRATION.  SETTING SPACE=21 WILL CAUSE
C.....A COMPLETE DATA SET TO BE PRINTED EVERY 20 INTEGRATIONS, AS WELL
C.....AS AN ABBREVIATEC SET EVERY INTEGRATION.  (THESE OPTIONS LIMITED
C.....TO OUT2, OUT4).
```

```
C.....READ IN DESIRED OUTPUT SUBROUTINE (2, 4, 6, ETC.)
C.....SOME ADDITIONAL ARBITRARY INTEGERS, K1,K2,K3, MAY BE READ IN HERE
C.....IF DESIRED - OTHERWISE LEAVE BLANK.  IF K1 IS SET GREATER THAN 10,
C.....ALL OUTPUTS WILL PRINT OUT ONE TO FIVE SPECIALLY DESIGNATED PIECES
C.....OF INFORMATION, DESIGNATED AS SP(I).  IF K3 IS SET EQUAL TO
C.....3 A VARIATION OF TURBULENT PR NEAR A WALL WILL BE SUPRESSED, SEE
C.....PRT(J) ABOVE.  SETTING K2 EQUAL TO 3 WILL DO THE SAME THING FOR
C.....ALMGG.
C.....IF K2 IS SET EQUAL TO 2, PROGRAM WILL USE A CONSTANT EDDY DIF-
C.....FUSIVITY IN THE OUTER REGION, INSTEAD OF A CONSTANT MIXING-
C.....LENGTH.  IT WILL BE EVALUATED FROM THE EQUATION, EDR=AQ*REM**BQ,
C.....WHERE REM IS MOMENTUM THICKNESS REYNOLDS NUMBER, OR PIPE DIAMETER
C.....REYNOLDS NUMBER.  DONT USE THIS OPTION IF FREE-STREAM VELOCITY
C.....IS ZERO (SEE COMMENT ON AQ,BQ).
C.....IF K1 IS SET EQUAL TO 9 OR 20, DELTAX BECOMES EQUAL TO AUX1(M),
C.....AND THE ORIGINAL INPUT VALUE OF DELTAX IS OVERRIDDEN.  THIS ALLOWS
C.....DELTAX TO VARY WITH X.
C.....(SEE PRESSURE GRADIENT CALCULATION IN MAIN FOR K1=15)
C ************************************************************************
C (INTEGERS)
      READ(5,585) NUMRUN,SPACE,OUTPUT,K1,K2,K3
C.....
      WRITE(6,730)
      WRITE(6,740) NUMRUN,SPACE,OUTPUT,K1,K2,K3
      IF(GEOM.EQ.4.OR.GEOM.EQ.5)GO TO 11
      IF(K2.NE.3.AND.KASE.EQ.1.AND.K2.NE.2.AND.MODE.EQ.2)WRITE(6,960)
   11 CONTINUE
      K10=0
      IF(K3.NE.3.AND.KASE.EQ.1)K10=1
      IF(K10.EQ.1.AND.NEQ.GT.1.AND.MODE.EQ.2)WRITE(6,970)
      IF(K1.EQ.20.OR.K1.EQ.9)WRITE(6,992)
C.....
C.....INPUT DATA ERROR CHECK
C.....
      IF(XU.LT.X(1).OR.XL.GT.X(NXBC))WRITE(6,503)
      IF(XU.LT.X(1).OR.XL.GT.X(NXBC))KERROR=3
      IF(XU.GE.XL)WRITE(6,500)
      IF(XU.GE.XL)KERROR=3
      IF(Y(NP3).GT.0.1*(X(2)-X(1)))WRITE(6,522)
      IF(BQ.LT.0.)KERROR=5
      IF(GC.LT.0.5)KERROR=5
      IF(OUTPUT.LT.1)KERROR=5
      IF(KIN.EQ.1.AND.U(1).GT.0.0)KERROR=5
      IF(KEX.EQ.1.AND.U(NP3).GT.0.0)KERROR=5
      IF(NEQ.GT.0)GO TO 13
      IF(TYPBC(1).LT.1)KERROR=5
   13 IF(GEOM.LT.6.AND.RW(1).EQ.0.0)KERROR=5
      IF(GC.EQ.0.0.OR.CJ.EQ.0.0)KERROR=5
      IF(SPACE.EQ.0)KERROR=5
      IF(KERROR.EQ.5)WRITE(6,502)
      IF(MODE.EQ.1)GO TO 15
      IF(AK.LT..25.OR.AK.GT..6)WRITE(6,504)
      IF(AK.LT..25.OR.AK.GT..6)KERROR=4
   15 DO 16 M=2,NXBC
   16 IF(X(M).LT.X(M-1))KERROR=1
      IF(Y(3).LT.Y(1))KERROR=1
      DO 18 I=4,NP1
   18 IF(Y(I).LE.Y(I-1))KERROR=1
      IF(Y(NP3).LT.Y(NP1))KERROR=1
      IF(KERROR.EQ.1)WRITE(6,507)
```

```
      IF(GEOM.EQ.4.AND.KIN.NE.3)KERROR=2
      IF(GEOM.EQ.5.AND.KIN.NE.3)KERROR=2
      IF(GEOM.EQ.3.AND.KIN.NE.2)KERROR=2
      IF(GEOM.EQ.2.AND.KIN.NE.1)KERROR=2
      IF(KIN.EQ.1.AND.KEX.EQ.1)KERROR=2
      IF(MODE.GT.2.OR.GEOM.LT.1)KERROR=2
      IF(KERROR.EQ.2)WRITE(6,508)
      IF((YPMAX-YPMIN).LE.0.0)KERROR=6
      IF((YPMAX-YPMIN).LE.YPMIN)KERROR=6
      IF(KERROR.EQ.6)WRITE(6,527)
  500 FORMAT(//' PROGRAM TERMINATED BECAUSE EITHER XU OR XL WERE'/
     1' OUTSIDE OF THE RANGE OF THE INPUT DATA, OR ELSE XU'/
     2' WAS INPUT AS GREATER THAN XL'///)
  502 FORMAT(//' PROGRAM TERMINATED BECAUSE OF INSUFFICIENT OR TOO '/
     1' MANY DATA CARDS, OR SOME OTHER INPUT ERROR'///)
  504 FORMAT(//' PROGRAM TERMINATED BECAUSE AK HAS ASSUMED AN ABSURD'/
     1' VALUE, CAUSED EITHER BY WRONG INPUT OR IMPROPER ATTENTION TO'/
     2' FORMATING OF SOME OF THE INPUT DATA'///)
  505 FORMAT(18A4)
  506 FORMAT(1H1,1X,18A4)
  507 FORMAT(//' PROGRAM TERMINATED BECAUSE THE INPUT VALUES OF EITHER'/
     1' X(M) OR Y(I) ARE NOT IN MONOTONIC SEQUENCE, OR ELSE THERE IS'/
     2' SOME OTHER INPUT FORMATING ERROR THAT HAS FORCED THESE'/
     3' QUANTITIES OUT OF ORDER'///)
  508 FORMAT(//' PROGRAM TERMINATED BECAUSE IT IS EITHER NOT'/
     1' YET COMPLETELY SET UP TO HANDLE THIS PARTICULAR'/
     2' GEOMETRY, OR ELSE THE COMBINATION OF KEX AND KIN'/
     3' IS NOT POSSIBLE IN THIS VERSION OF THE PROGRAM'///)
  510 FORMAT(/50H GEOMETRY MODE FLUID  NEQ  N  KEX  KIN              )
  520 FORMAT(4X,I2,5X,I1,4X,I2,4X,I2,2X,I2,3X,I1,4X,I1)
  522 FORMAT(//' THE INITIAL BOUNDARY LAYER IS RATHER THICK RELATIVE'/
     1' TO THE SPACING OF THE BOUNDARY CONDITION POINTS.  THIS MAY'/
     2' LEAD TO TROUBLES, ESPECIALLY WITH PRESSURE GRADIENT.'///)
  525 FORMAT(/' ENTRAINMENT BASED ON MOMENTUM EQUATION ONLY.')
  526 FORMAT(/' ENTRAINMENT BASED ON BEHAVIOR OF ALL EQUATIONS.')
  527 FORMAT(//' PROGRAM TERMINATED BECAUSE YPMAX IS TOO SMALL,'/
     1' OR THERE IS SOME OTHER RELATED INPUT ERROR.'///)
  540 FORMAT(/107H     XU          XL        DELTAX   TRANSITION REYNOLDS NO.
     1  FRA    ENTRAINMENT FRACTION   GRAVITY CONSTANT       )
  550 FORMAT(2X,F6.3,3X,F7.3,4X,F5.2,9X,F7.1,12X,F5.3,8X,F7.4,11X,F7.2,
     18X,F7.1,/)
  570 FORMAT(60H  NXBC       TYPBC1     TYPBC2     TYPBC3     TYPBC4     TYPB
     1C5  )
  580 FORMAT(7F10.0)
  585 FORMAT(8I5)
  590 FORMAT(I4,10X,I2,8X,I2,8X,I2,8X,I2,8X,I2)
  600 FORMAT(/,120H    M         X(M)        RW(M)        UG(M)        AM(M)       A
     1UX1(M)    AUX2(M)     FJ(1,M)     FJ(2,M)     FJ(3,M)     FJ(4,M)     FJ(5,M)
     2 )
  610 FORMAT(3X,I2,6X,F7.3,1X,F9.3,1X,F10.3,F10.3,F10.3,F10.3,F10.3,
     1F10.3,F10.3,F10.3,F10.3)
  630 FORMAT(80H     I          Y(I)        U(I)        F(1,I)      F(2,I)      F(3,
     1I)      F(4,I)      F(5,I)     ,/)
  640 FORMAT(/,95H   KAPPA     LAMBDA        FR    LAG CONSTANT      AQ
     1EQ     MAX. YPLUS IN WF   MIN. YPLUS IN WF)
  650 FORMAT(1X,F7.4,3X,F7.4,3X,F7.4,3X,F7.2,3X,F7.4,3X,F7.4,4X,F8.3,
     111X,F7.4,/)
  660 FORMAT(/,50H    PRT(1)     PRT(2)     PRT(3)     PRT(4)     PRT(5)    )
  680 FORMAT(/68H   DENSITY    VISCOSITY     PRC(1)     PRC(2)     PRC(3)     PR
     1C(4)    PRC(5))
```

```
690 FORMAT(1X,F8.4,3X,F9.7,3X,F8.3,3X,F6.3,3X,F6.3,3X,F6.3,3X,F6.3,/) INPU5350
700 FORMAT(/'      INITIAL STATIC PRESSURE ')                             INPU5360
710 FORMAT(//' WARNING: THE INPUT VALUES OF AK, AQ, AND BQ, ARE'/         INPU5370
   1' INCONSISTENT'//)                                                    INPU5380
720 FORMAT(/,107H G-SUB-C        J                                AXX     INPU5390
   1       BXX             CXX             DXX             EXX)           INPU5400
730 FORMAT(/,81H   NO. OF RUNS OF DATA       PRINTOUT SPACING      OUTPU INPU5410
   1T OPTION        K1    K2    K3)                                       INPU5420
740 FORMAT(13X,I2,22X,I2,15X,I2,10X,I5,I5,I5,/)                           INPU5430
750 FORMAT(/'  CONSTANT FLUID PROPERTIES ARE BEING USED')                 INPU5440
760 FORMAT(/' INITIAL PROFILES'/)                                         INPU5450
770 FORMAT(/' BOUNDARY CONDITIONS ALONG I- AND E-SURFACES'/)              INPU5460
780 FORMAT(/'  TURBULENCE CONSTANTS                                   ')  INPU5470
790 FORMAT(/'  DIMENSIONING SYSTEM CONSTANTS                  ARBITRARY INPU5480
   1CONSTANTS')                                                           INPU5490
800 FORMAT(/'  THE FLUID IS AIR (KEENAN AND KAYE GAS TABLES)')            INPU5500
820 FORMAT(/63H BODY-FORCE SOURCE(1) SOURCE(2) SOURCE(3) SOURCE(4) SOU INPU5510
   1RCE(5)   )                                                            INPU5520
830 FORMAT(5X,I1,9X,I1,9X,I1,9X,I1,9X,I1,9X,I1,/)                         INPU5530
850 FORMAT(3X,I2,7X,F7.5,2X,F7.2,F10.3,1X,F10.3,1X,F10.3,F11.3,1X,F10  INPU5540
   1.3)                                                                   INPU5550
870 FORMAT(1X,F8.4,3X,F9.7,/)                                             INPU5560
890 FORMAT(38H  NXBC (NUMBER OF SPECIFIED BC POINTS))                     INPU5570
900 FORMAT(10X,F10.2)                                                     INPU5580
910 FORMAT(2X,F4.1,7X,F5.1,18X,5F15.4)                                    INPU5590
920 FORMAT(/' PROGRAM TERMINATED BECAUSE NXBC WAS READ AS A'/             INPU5600
   1' NUMBER LESS THAN 2, WHICH IS NOT ALLOWED.'//)                       INPU5610
930 FORMAT(/'      APLUS      BPLUS     SIGNAL ')                         INPU5620
940 FORMAT(4X,F6.2,4X,F6.2,4X,F6.2)                                       INPU5630
950 FORMAT(/' THE PROGRAM IS USING AN INTERNAL CORRELATION TO         '/INPU5640
   1' ACCOUNT FOR THE INFLUENCE OF PRESSURE GRADIENT AND TRANSPIR-'/     INPU5650
   2' ATION ON APLUS')                                                    INPU5660
955 FORMAT(/' THE PROGRAM IS USING AN INTERNAL CORRELATION TO     '/     INPU5670
   1' ACCOUNT FOR THE INFLUENCE OF PRESSURE GRADIENT AND TRANSPIR-'/     INPU5680
   2' ON BPLUS')                                                          INPU5690
960 FORMAT(' IF REM IS LESS THAN ABOUT 6000,'/                            INPU5700
   1' LAMBDA IS BEING COMPUTED BY AN INTERNAL EQUATION.'/)                INPU5710
970 FORMAT(' PRT NEAR THE WALL IS BEING'/                                 INPU5720
   1' EVALUATED BY AN INTERNAL EQUATION, EXCEPT WHEN PRT IS FOR'/        INPU5730
   2' THE TURBULENT KE EQUATION.'/)                                       INPU5740
980 FORMAT(/' PROGRAM WILL BOMB OUT BECAUSE N IS GREATER THAN 40.'/)      INPU5750
990 FORMAT(/' IF THE LAG CONSTANT IS LESS THAN 400, IT IS TREATED'/      INPU5760
   1' AS IF IT WERE ZERO.'/)                                              INPU5770
992 FORMAT(/' DELTAX IS BEING OVERRIDDEN BY AJX1(M).'/)                   INPU5780
    RETURN                                                                INPU5790
    END                                                                   INPU5800
```

## Appendix IV

## SAMPLE DATA SETS

```
EXTERNAL LAMINAR BOUNDARY LAYER, NO PRESSURE GRADIENT OR TRANSPIRATION
    1       1      1      1    24      2      1
0.023      0.320      1.         300.      0.05      0.003
    0
2117.      0.075      0.000012
    2
0.023      1.
0.320      1.
50.
50.
0.0        0.0
0.000062   3.1
0.000123   6.2
0.000185   9.3
0.000247   12.3
0.000308   15.3
0.000370   18.3
0.000432   21.1
0.000493   23.9
0.000555   26.7
0.000617   29.2
0.000678   31.7
0.000740   34.3
0.000802   36.4
0.000863   38.5
0.000925   40.7
0.000987   42.3
0.001050   43.9
0.001110   45.6
0.001170   46.7
0.001230   47.8
0.001300   48.9
0.001360   49.3
0.001420   49.7
0.001480   50.0
0.41       0.085      0.01       0.        0.        1.        0.
25.        0.
4000.      0.
32.2       778.
    1    21     2



NOTE:  MOMENTUM EQUATION ONLY IS BEING SOLVED.  A TRANSITION TO A
       TURBULENT BOUNDARY LAYER WILL OCCUR WHEN REM REACHES 300.
       AN ENTRAINMENT FRACTION OF 0.003 IS RECOMENDED FOR LAMINAR
       BOUNDARY LAYER FLOWS.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
EXTERNAL TURBULENT BOUNDARY LAYER, NO PRESSURE GRADIENT OR TRANSPIRATION
     1      2     1     2    15     2     1     1
0.0        4.0        1.0        0.0        0.05       .005
     0      0
2117.      0.075      0.000012   0.7
     2      1
0.0        1.0
4.0        1.0
110.                  100.
110.                  100.
0.0        0.0        100.
0.0024     73.        173.
0.0034     81.        181.
0.0044     85.        185.
0.0056     88.        188.
0.0066     91.        191.
0.0078     93.        193.
0.0088     96.        196.
0.01       98.        198.
0.0112     99.        199.
0.0122     101.       201.
0.0134     102.       202.
0.0144     104.       204.
0.0166     106.       206.
0.0188     108.       208.
0.0232     110.       210.
0.41       0.085      .01        0.0        0.0        1.0        0.0
25.
4000.      0.86
32.2       778.0
     1    21     2



NOTE:   THIS DATA IS SET UP FOR CONSTANT PROPERTIES; IF FLUID IS CHANGED
        TO 2, VARIABLE PROPERTIES OF AIR WILL BE USED, CONSISTANT WITH
        THE SPECIFIED PRESSURE AND ENTHALPIES.

        WITH YPMAX SET TO 1.0, THE WALL FUNCTION IS BEING BYPASSED AND
        THE PROGRAM WILL INSERT A NUMBER OF ADDITIONAL GRID POINTS.  IF
        IT IS DESIRED TO USE THE WALL FUNCTION OPTION, CHANGE YPMAX.

        THE MIXING-LENGTH SCHEME, WITH VAN DRIEST DAMPING IS USED THROUGH
        OUT THE BOUNDARY LAYER.
```

```
EXTERNAL TURBULENT BOUNDARY LAYER, USING TURBULENT KINETIC ENERGY
       1     2     1     2   15     2     1     1
0.0          4.0        1.0         300.       .05         .005
       0     2
2117.        0.075      0.000012  1.0
       2     1
0.0          1.0
4.0          1.0
110.                    0.
110.                    0.
0.0          0.         0.
0.0024       73.        73.
0.0034       81.        71.
0.0044       85.        67.
0.0056       88.        62.
0.0066       91.        57.
0.0078       93.        52.
0.0088       96.        47.
0.01         98.        41.
0.0112       99.        36.
0.0122       101.       31.
0.0134       102.       26.
0.0144       104.       21.
0.0166       106.       13.
0.0188       108.       5.
0.0232       110.       0.
0.41         0.085      .01         .22        .377        1.0        0.0
25.0
4000.        1.7
32.2         778.0
       1   21     2



NOTE:  MOMENTUM AND TURBULENT KINETIC ENERGY EQUATION ARE BEING SOLVED.
       THE THERMAL ENERGY EQUATION CAN BE ADDED AS A THIRD EQUATION AS
       DESIRED.  SEE PREVIOUS NOTE ON YPMAX.
```

```
LAMINAR FLOW IN A CIRCULAR PIPE, ENTRY LENGTH PROBLEM
     4     1     1     2   31     1     3
0.0         20.         1.0         2000.       .05
     0     0
21.17       0.0075      0.000012   0.7
     6     1
0.0         0.1         0.1
0.5         0.1         0.5
2.0         0.1         1.0
4.0         0.1         2.0
10.0        0.1         5.0
20.         0.1         5.0
                        240.
                        240.
                        240.
                        240.
                        240.
                        240.
0.0         8.          120.
0.022       8.          120.
0.034       8.          120.
0.039       8.          120.
0.043       8.          120.
0.047       8.          120.
0.051       8.          120.
0.055       8.          120.
0.058       8.          120.
0.061       8.          120.
0.064       8.          120.
0.067       8.          120.
0.070       8.          120.
0.073       8.          120.
0.076       8.          120.
0.078       8.          120.
0.080       8.          120.
0.082       8.          120.
0.084       8.          120.
0.086       8.          120.
0.088       8.          120.
0.090       8.          120.
0.092       8.          120.
0.094       8.          120.
0.096       8.          120.
0.098       8.          120.
0.0985      8.          120.
0.0988      8.          120.
0.0991      8.          120.
0.0994      7.          120.
0.0997      4.          180.
0.100       0.0         240.
0.0         0.0         0.01        0.0         0.0         1.0         0.0
0.0         0.0
0.0         0.0
32.2        778.0
     1    21     4     9



NOTE:  THE FOREWARD STEP SIZE IS HERE BEING VARIED USING KI = 9 AND
       A TABLE OF AUX1(M).
```

```
 1.    TURBULENT FLOW IN A CIRCULAR PIPE
 2.          4       2       1       2    17      1       3
 3.    0.0           4.0             1.0              2000.         .05
 4.          0       0
 5.    2117.         0.075           0.000012         0.7
 6.          2       2
 7.    0.0           0.1
 8.    9.            0.1
 9.                                  -2.
10.                                  -2.
11.    0.0           113.            137.
13.    0.010         112.            139.
15.    0.020         109.            143.
17.    0.030         106.            148.
18.    0.035         105.            149.
19.    0.040         103.            151.
20.    0.045         102.            153.
21.    0.050         100.            155.
22.    0.055         99.             158.
23.    0.060         95.             162.
24.    0.065         94.             164.
25.    0.070         91.             168.
26.    0.075         89.             172.
27.    0.080         87.             175.
28.    0.085         84.             179.
29.    0.090         80.             185.
30.    0.095         73.             193.
31.    0.10          0.              200.
32.    0.41          0.075           0.01             0.005         0.9           40.           0.0
33.    26.
34.    4000.         0.86
35.    32.2          778.0
36.          1     21       4               2
37.
38.
39.
40.    NOTE:  HERE EDDY VISCOSITY IN THE OUTER REGION IS BEING COMPUTED
41.           AS A CONSTANT BASED ON REYNOLDS NUMBER RATHER THAN USING
42.           A MIXING-LENGTH.  HEAT FLUX IS SPECIFIED AT THE WALL (AS A
43.           CONSTANT VALUE) RATHER THAN WALL ENTHALPY AS IN THE PREVIOUS
44.           EXAMPLES.  WALL FUNCTION IS BEING USED IN NEAR-WALL REGION.
```

```
LAMINAR FREE CONVECTION FROM A VERTICAL FLAT PLATE
    1    1    2    2   31    2    1    0
0.0       3.0       .5        1.0       .01       .01       -32.2
    1    0
2117.
    2    1
0.0       4.0
3.0       4.0
.0001     0.0       138.66
.0001     0.0       138.66
 0.0        0.0       138.66
 .2525E-03  .0165     138.47
 .4208E-03  .0276     138.34
 .5049E-03  .0331     138.26
 .6350E-03  .041      138.16
 .7651E-03  .0507     138.06
 1.017E-03  .0672     137.87
 1.270E-03  .0838     137.66
 1.530E-03  .1014     137.46
 1.783E-03  .1179     137.27
 2.035E-03  .1344     137.06
 2.295E-03  .1521     136.74
 2.548E-03  .1686     136.67
 2.800E-03  .1851     136.46
 3.060E-03  .2028     136.14
 3.313E-03  .2193     136.07
 3.565E-03  .2358     135.86
 3.826E-03  .2424     135.66
 4.208E-03  .2645     135.36
 4.591E-03  .2755     135.06
 5.356E-03  .2865     134.46
 6.121E-03  .2975     133.98
 7.651E-03  .3053     132.90
 9.181E-03  .2975     132.06
 12.24E-03  .2535     130.38
 16.07E-03  .1708     128.82
 19.89E-03  .1102     127.74
 23.72E-03  .0628     127.26
 27.54E-03  .0331     127.02
 31.37E-03  .0198     126.88
 35.19E-03  .0088     126.72
 38.26E-03  .0001     126.66
0.0       0.0       .01       0.0       0.0       1.0       0.0
0.0
0.0       0.0
32.2      778.0
    1   20    6
```

NOTE: A LARGE ENTRAINMENT FRACTION IS USED, AS DISCUSSED IN THE TEXT. A FINITE FREE-STREAM VELOCITY IS USED, BUT THE PROGRAM WILL GENERALLY OPERATE SATISFACTORILY WITH A ZERO VALUE IF THE ENTRAINMENT FRACTION IS NOT TOO SMALL. AN E-FORMAT IS HERE USED FOR THE Y-DISTANCES, FOR CONVENIENCE (IT OVERRIDES THE F-FORMAT SPECIFICATION, BUT THE NUMBERS MUST BE JUSTIFIED TO THE RIGHT OF THE 10 SPACE FIELD). SET RETRAN=1.0 FOR LAMINAR FREE CONVECTION FLOWS.

```
FLOW IN A SUPERSONIC NOZZLE, PRESCRIBED CORE VELOCITY DISTRIBUTION
     3      2     2      2    20      1     2     0
-0.1783    1.6358    0.25       200.       0.01       0.01
     0      1
2980.8
    40      1
-.1783     0.2083
0.0208     0.2083
0.0416     0.2083
0.1692     0.1864
0.2888     0.1656
0.3767     0.1503
0.4448     0.1385
0.4994     0.1291
0.5443     0.1213
0.5805     0.1150
0.61       0.1098
0.6356     0.1054
0.658      0.1015
0.6952     0.0951
0.7249     0.089
0.7491     0.0857
0.7691     0.0822
0.7931     0.0781
0.8117     0.0748
0.8261     0.0723
0.8457     0.0689
0.8545     0.0674
0.8603     0.0664
0.8616     0.0664
0.8655     0.0671
0.8731     0.0684
0.8882     0.071
0.9071     0.0743
0.9388     0.0798
0.9775     0.0865
1.0229     0.0944
1.0751     0.1035
1.1038     0.1085
1.1341     0.1137
1.1999     0.1251
1.2726     0.1378
1.3524     0.1516
1.4394     0.1667
1.5334     0.183
1.6358     0.2008
0123.78               150.69
0123.78               150.69
0123.78               150.69
0151.85               150.69
0189.74               150.69
0227.59               150.69
0265.38               150.69
0303.13               150.69
0340.77               150.69
0378.36               150.69
0415.85               150.69
0453.25               150.69
0490.53               150.69
0564.75               150.69
```

```
0638.42                150.69
0711.56                150.69
0784.02                150.69
0891.38                150.69
0996.99                150.69
1100.7                 150.69
1301.7                 150.69
1446.5                 150.69
1631.                  150.69
1805.2                 150.69
1969.                  150.69
2122.2                 150.69
2310.7                 150.69
2471.4                 150.69
2663.2                 150.69
2831.4                 150.69
2978.6                 150.69
3107.4                 150.69
3165.7                 150.69
3220.1                 150.69
3318.6                 150.69
3405.3                 150.69
3481.4                 150.69
3548.5                 150.69
3607.8                 150.69
3660.4                 150.69
0.00000   123.780      369.17
0.00373   122.31       369.17
0.00746   120.73       369.17
0.0112    119.02       366.42
0.01493   117.14       363.02
0.01867   115.07       359.26
0.02053   113.94       357.21
0.0224    112.74       355.04
0.02427   111.46       352.72
0.02614   110.08       350.22
0.028     108.59       347.52
0.02987   106.97       344.58
0.03174   105.18       341.35
0.03360   103.2        337.74
0.03547   100.95       333.67
0.03734   98.354       328.96
0.0392    95.268       323.37
0.04107   91.432       316.42
0.04294   86.286       307.09
0.0448    78.152       292.35
0.04667   00.000       150.69
0.41      0.085        0.01      0.0       0.0       1.0       0.0
25.
4000.     0.86
32.2      778.
    1    5    6


NOTE:   VARIABLE PROPERTIES OF AIR ARE BEING COMPUTED USING SUBROUTINE
        PROP2.  THE FLOW IS AN AXI-SYMMETRIC BOUNDARY LAYER AND THE
        WALL RADIUS IS BEING CONSIDERED (GEOM=3).  NOTE CHANGE IN FRA,
        ENFRA.
```



NASA-Langley, 1976