An explicit finite difference analysis for developing turbulent internal flows with heat transfer and property variations

Ron Michael Nelson

Follow this and additional works at: http://lib.dr.iastate.edu/rtd

Part of the Fluid Dynamics Commons, and the Heat Transfer, Combustion Commons

Recommended Citation
An explicit finite difference analysis for developing turbulent internal flows with heat transfer and property variations

by

Ron Michael Nelson

A Thesis Submitted to the Graduate Faculty in Partial Fulfillment of The Requirements for the Degree of MASTER OF SCIENCE

Major: Mechanical Engineering

Approved:

________________________
In Charge of Major Work

________________________
For the Major Department

________________________
For the Graduate College

Iowa State University
Ames, Iowa
1972
TABLE OF CONTENTS

LIST OF SYMBOLS vi
ACKNOWLEDGEMENTS xiii
ABSTRACT xiv
INTRODUCTION 1
MATHEMATICAL STATEMENT OF THE PROBLEM 4
NON-DIMENSIONAL FORMS OF THE GOVERNING EQUATIONS 10
NUMERICAL SOLUTION 12

DuFort-Frankel Finite Difference Equations 14
Standard Explicit Finite Difference Equations 17
Method of Solution 19
Consistency and Stability 27

RESULTS 31
DISCUSSION 54
BIBLIOGRAPHY 59
APPENDIX 63

Main Program 64
Subroutine DELX1 92
Subroutine PRES 94
Subroutine UVEL 99
Subroutine TEMP 103
Subroutine EVISIC 107
Subroutine POLFIT 110
Subroutine FDITVP 113
Function F 117
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Cylindrical coordinate system used for flow in pipes.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Cartesian coordinate system used for flow between parallel plates or over a flat plate.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3</td>
<td>The finite difference grid.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4</td>
<td>A skeleton flow chart of the computer program.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Comparison of the present method with some predictions for velocity profiles for developing constant property laminar flow between parallel plates.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Comparison of the present method with some predictions for pressure drop for laminar constant property flow in the inlet section of a parallel plate channel.</td>
<td>33</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Comparison of the present method with some predictions for pressure drop for laminar constant property flow in the inlet section of a pipe.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Comparison of the present method with data for a fully developed constant property turbulent velocity profile.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Comparison of the present method with data for constant property turbulent velocity profiles in the inlet section of a pipe.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Comparison of the present method with data for pressure distribution for constant property turbulent flow in the inlet section of a pipe.</td>
<td>38</td>
</tr>
</tbody>
</table>
Figure 11. Comparison of the present method with a prediction for Nusselt numbers and wall to bulk temperature ratios for laminar flow of air between constant temperature parallel plates with uniform initial velocity and temperature profiles.

Figure 12. Comparison of the present method with predictions for Nusselt number, bulk to initial temperature ratio, and wall to bulk temperature ratio distributions in the inlet section of a pipe for laminar flow of air with constant wall heat flux and uniform initial velocity and temperature profiles.

Figure 13. Comparison of the present method with predictions for wall shear and pressure distributions in the inlet section of a pipe for laminar flow of air with constant wall heat flux and uniform initial velocity and temperature profiles.

Figure 14. Comparison of the present method with a prediction for velocity profiles for compressible laminar flow of air over a constant temperature flat plate.

Figure 15. Comparison of the present method with a prediction for temperature profiles for compressible laminar flow of air over a constant temperature flat plate.

Figure 16. Comparison of the present method with data for Nusselt number distributions in the inlet section of a pipe for turbulent flow of air with constant wall heat flux.

Figure 17. Comparison of the present method with data and another prediction for wall to initial temperature ratios for turbulent flow of nitrogen in the thermal entrance region of a pipe.
Figure 18. Comparison of the present method with data and another prediction for pressure distributions for turbulent flow of nitrogen in the thermal entrance region of a pipe.

Figure 19. Comparison of the present method with data for Nusselt number distributions for turbulent flow of nitrogen in the thermal entrance region of a pipe.

Figure 20. Comparison of the present method with data for a Stanton-Prandtl number group versus Reynolds number based on x for turbulent flow of air in the entrance region of a pipe with constant wall temperature and uniform initial velocity and temperature profiles.

Figure 21. Comparison of the present method with data and other predictions for skin friction coefficients and Stanton numbers versus Reynolds number based on x for compressible turbulent flow of air over a constant temperature flat plate.

Figure 22. Comparison of the present method with data for the convective coefficient distribution for turbulent flow of water (Pr=7) in the thermal entrance region of a pipe with constant wall heat flux.
LIST OF SYMBOLS

**Dimensional Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Dimensions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ft²</td>
<td>area</td>
</tr>
<tr>
<td>a</td>
<td>ft</td>
<td>half width of channel</td>
</tr>
<tr>
<td>Cₚ</td>
<td>BTU</td>
<td>lbₘ R</td>
</tr>
<tr>
<td>c</td>
<td>ft</td>
<td>sec</td>
</tr>
<tr>
<td>D</td>
<td>ft</td>
<td>diameter of pipe or width of channel</td>
</tr>
<tr>
<td>G</td>
<td>lbₘ</td>
<td>sec ft²</td>
</tr>
<tr>
<td>gₜ</td>
<td>lbₘ ft</td>
<td>lbₙ sec²</td>
</tr>
<tr>
<td>h</td>
<td>BTU</td>
<td>hr ft² R</td>
</tr>
<tr>
<td>i</td>
<td>BTU</td>
<td>lbₘ</td>
</tr>
<tr>
<td>J</td>
<td>ft lbₙ</td>
<td>BTU</td>
</tr>
</tbody>
</table>

constant (32.174)

constant (777.66)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Dimensions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>( \frac{\text{BTU}}{\text{hr ft } ^\circ \text{R}} )</td>
<td>thermal conductivity</td>
</tr>
<tr>
<td>l</td>
<td>ft</td>
<td>mixing length</td>
</tr>
<tr>
<td>m</td>
<td>( \frac{\text{lb}_m}{\text{sec}} )</td>
<td>mass flow rate</td>
</tr>
<tr>
<td>p</td>
<td>( \frac{\text{lb}_f}{\text{ft}^2} )</td>
<td>static pressure</td>
</tr>
<tr>
<td>q</td>
<td>( \frac{\text{BTU}}{\text{hr ft}^2} )</td>
<td>heat flux--equation (6b)</td>
</tr>
<tr>
<td>r</td>
<td>ft</td>
<td>distance from center line</td>
</tr>
<tr>
<td>t</td>
<td>( ^\circ \text{R} )</td>
<td>temperature</td>
</tr>
<tr>
<td>u</td>
<td>( \frac{\text{ft}}{\text{sec}} )</td>
<td>axial velocity</td>
</tr>
<tr>
<td>( u^* )</td>
<td>( \frac{\text{ft}}{\text{sec}} )</td>
<td>friction velocity, ( \sqrt{\frac{g_C}{\rho_w} \frac{T_w}{\rho_w}} )</td>
</tr>
<tr>
<td>v</td>
<td>( \frac{\text{ft}}{\text{sec}} )</td>
<td>radial velocity</td>
</tr>
<tr>
<td>x</td>
<td>ft</td>
<td>axial coordinate</td>
</tr>
<tr>
<td>Symbol</td>
<td>Dimensions</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>( y )</td>
<td>ft</td>
<td>coordinate normal to duct wall</td>
</tr>
<tr>
<td>( \Delta x )</td>
<td>ft</td>
<td>( x ) grid space</td>
</tr>
<tr>
<td>( \Delta y )</td>
<td>ft</td>
<td>( y ) grid space</td>
</tr>
<tr>
<td>( \delta )</td>
<td>ft</td>
<td>boundary layer thickness</td>
</tr>
<tr>
<td>( \mu )</td>
<td>( \frac{lb_m}{ft \ sec} )</td>
<td>viscosity</td>
</tr>
<tr>
<td>( \nu )</td>
<td>( \frac{ft^2}{sec} )</td>
<td>kinematic viscosity</td>
</tr>
<tr>
<td>( \rho )</td>
<td>( \frac{lb_m}{ft^3} )</td>
<td>density</td>
</tr>
<tr>
<td>( \tau )</td>
<td>( \frac{lb_f}{ft^2} )</td>
<td>shear stress--equation (1b)</td>
</tr>
</tbody>
</table>
### Dimensionless Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^+$</td>
<td>constant in damping factor</td>
</tr>
<tr>
<td>$\hat{A}$</td>
<td>dimensionless area, $\frac{A \rho_0 u_0^2}{\mu_0}$</td>
</tr>
<tr>
<td>$\hat{C}_p$</td>
<td>dimensionless specific heat, $\frac{C_p}{C_p \rho_0}$</td>
</tr>
<tr>
<td>$f_s$</td>
<td>dimensionless wall shear stress, $\frac{\tau_w}{(g^2/g_c \rho_b)}$</td>
</tr>
<tr>
<td>$\hat{i}$</td>
<td>dimensionless enthalpy, $\frac{i g_c J}{u_0^2}$</td>
</tr>
<tr>
<td>$K$</td>
<td>Von Karman constant ($\approx 0.40$)</td>
</tr>
<tr>
<td>$\hat{k}$</td>
<td>dimensionless thermal conductivity, $\frac{k}{3600 \mu_0 C_p \rho_0}$</td>
</tr>
<tr>
<td>$\hat{l}$</td>
<td>dimensionless mixing length, $\frac{\rho_0 u_0 l}{\mu_0}$</td>
</tr>
<tr>
<td>$M$</td>
<td>Mach number, $\frac{u}{c}$</td>
</tr>
<tr>
<td>$\hat{m}$</td>
<td>dimensionless mass flow, $\frac{\dot{m} \rho_0 u_0}{\mu_0}$</td>
</tr>
<tr>
<td>$\text{Nu}_D$</td>
<td>Nusselt number based on $D$, $\frac{h D}{k_b}$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Nu&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Nusselt number based on x, ( \frac{hx}{k_b} )</td>
</tr>
<tr>
<td>p</td>
<td>dimensionless pressure, ( \frac{pg_c}{\rho_o u_o^2} )</td>
</tr>
<tr>
<td>Pr</td>
<td>Prandtl number, ( \frac{\mu_b C_p b}{k_b} )</td>
</tr>
<tr>
<td>Pr&lt;sub&gt;t&lt;/sub&gt;</td>
<td>turbulent Prandtl number, ( \frac{\mu_t C_p}{k_t} )</td>
</tr>
<tr>
<td>Q</td>
<td>dimensionless heat flux, ( \frac{g_c J q}{3600 \rho_o u_o^3} )</td>
</tr>
<tr>
<td>R</td>
<td>dimensionless distance from center line, ( \frac{\rho_o u_o r}{\mu_o} )</td>
</tr>
<tr>
<td>Re&lt;sub&gt;D&lt;/sub&gt;</td>
<td>Reynolds number based on D, ( \frac{\rho D u_b D}{u_b} )</td>
</tr>
<tr>
<td>Re&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Reynolds number based on x, ( \frac{\rho D u_b x}{u_b} )</td>
</tr>
<tr>
<td>St</td>
<td>Stanton number, ( \frac{h}{\rho_b u_b C_p b} )</td>
</tr>
<tr>
<td>T</td>
<td>dimensionless temperature, ( \frac{g_c J c_p o t}{u_o^2} )</td>
</tr>
<tr>
<td>U</td>
<td>dimensionless axial velocity, ( \frac{u}{u_o} )</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$u^+$</td>
<td>dimensionless velocity, $\frac{u_x}{u}$</td>
</tr>
<tr>
<td>$V$</td>
<td>dimensionless radial velocity, $\frac{v}{u_0}$</td>
</tr>
<tr>
<td>$X$</td>
<td>dimensionless axial coordinate, $\frac{\rho_0 u_0 x}{\mu_0}$</td>
</tr>
<tr>
<td>$Y$</td>
<td>dimensionless coordinate normal to duct wall, $\frac{\rho_0 u_0 y}{\mu_0}$</td>
</tr>
<tr>
<td>$y^+$</td>
<td>dimensionless distance normal to wall, $\frac{\rho_w u^* y}{\mu_w}$</td>
</tr>
<tr>
<td>$\hat{\mu}$</td>
<td>dimensionless viscosity, $\frac{\mu}{\mu_0}$</td>
</tr>
<tr>
<td>$\hat{\rho}$</td>
<td>dimensionless density, $\frac{\rho}{\rho_0}$</td>
</tr>
</tbody>
</table>
Subscripts

aw  adiabatic wall
b  evaluated at bulk temperature and mean velocity (or edge conditions for flat plate)
c  evaluated at the center line (except in $g_c$)
eff  effective, sum of laminar and turbulent components
o  evaluated at initial bulk temperature and mean velocity
t  turbulent component or total property
w  evaluated at the wall

$\Delta X_+ = X_{i+1} - X_i$
$\Delta X_- = X_i - X_{i-1}$
$\Delta Y_+ = Y_{j+1} - Y_j$
$\Delta Y_- = Y_j - Y_{j-1}$

Superscripts

( )'  fluctuating components in turbulent flows
$\overline{()}$  time averaged

Quantities without prime are time averaged
I am very grateful to Dr. R. H. Pletcher, my committee chairman, for the many hours he has spent explaining ideas and methods to me. I truly appreciate all his helpful suggestions.

I am also thankful for the grant to Dr. Pletcher from the National Science Foundation which provided funds for my research assistantship and the computer time needed to work on this project.

I would also like to thank all my teachers for their part in all I have learned.

I also appreciate the help received from my wife, Dianna, in typing and keypunching this thesis.
A calculations method is presented which uses an always stable explicit finite difference technique to solve the coupled partial differential equations for the two-dimensional boundary layer in either cylindrical or Cartesian coordinates. The method can also be used to calculate the axial pressure gradients and satisfy total conservation of mass for internal flows. Turbulent shear and heat flux terms have been modeled with "effective" viscosities and thermal conductivities, and different models may easily be tested. The viscous dissipation term has been kept in the energy equation. Several boundary and initial conditions are available to solve most pipe, parallel plate channel, and flat plate flows. Various models for fluid properties are used in the program, and other models or tables may be added by making small changes. Results of the present method are presented and compared to experimental data and other predictions for laminar and turbulent flows in pipes and channels and over flat plates. A complete listing of the computer program with instructions for input is included in the appendix.
INTRODUCTION

Convective heat transfer is an important method of cooling or heating. Because of the high temperatures involved in nuclear reactors, rockets, and other devices, property variations greatly influence the temperature and velocity distributions, and constant property assumptions no longer yield good results. Analytical methods which readily yield results are severely restricted to the property variations they can handle. Numerical methods seem to be the easiest way to handle an arbitrary property variation. Once the basic computer program has been developed, a table of properties can be input for any substance, and a solution can be obtained. Also, arbitrary boundary and initial conditions can be handled by simply inputting the values.

Internal fluid flow with heat transfer occurs in many engineering applications. It will be shown that boundary layer assumptions apply, except very near the inlet, and lead to good predictions for heat transfer and pressure drop in pipe and channel flows. So, a numerical solution of the boundary layer equations would seem to be a good approach to solving internal flow cases with property variations, and arbitrary boundary and initial conditions. Similar approaches have been tried to some extent before, and have been found successful (3, 11, 26).
The first finite difference solutions for internal flows were plane Poiseuille and Couette flow developments by Bodoia and Osterle (5) in 1961. In 1964, Hornbeck (10) used finite difference methods to obtain a solution to laminar flow in the entrance region of a pipe. Laminar heat transfer solutions for internal flows were obtained numerically by Worsoe-Schmidt and Leppert (35) in 1965, and Presler (23, 24) and Schade and McEligot (26) in 1971. Finite difference methods have only recently been applied to turbulent flows. McEligot et al. (3, 14, 27) have obtained some solutions for turbulent internal heat transfer cases (without viscous dissipation) in 1970 and 1971. All previous internal flow calculations have used implicit finite difference techniques. Although explicit methods are algebraically simpler because they solve directly for one variable at a time, for turbulent flows, the ordinary explicit methods require very small streamwise steps for stability reasons. Implicit finite difference techniques are not as restricted in the step sizes they can take, but they must solve a set of simultaneous algebraic equations after each streamwise step. Pletcher (20, 21, 22) has used an always stable explicit method of DuFort and Frankel (8), which requires no iterations, for calculating external boundary layer flows. This method is algebraically simple, and the step sizes are not restricted for stability reasons.
This thesis presents the development of a computer program which extends this explicit formulation of DuFort and Frankel to solve the coupled partial differential equations for internal flows. The program is capable of handling laminar or turbulent pipe and channel flows with uniform or fully developed initial velocity profiles and uniform initial temperature profiles. Heat transfer boundary conditions can be any arbitrary distribution of wall temperature or heat flux.

A major challenge is the selection of a turbulent model. Once the basic program has been developed, only small changes must be made to test different models. Also, small changes can be made to calculate different flow problems, such as flows with blowing and suction and flows in annuli. The extension of the program to include area change looks promising.
The problem being considered is steady turbulent flow through a pipe or channel with heat transfer and property variations. The axial velocities and lengths are much greater than those normal to the duct wall. An order of magnitude analysis reduces Reynolds equations to the turbulent momentum boundary layer equation. This equation is valid everywhere except possibly very near the entrance where the second derivative of $u$ in the axial direction may not be negligibly small. With the assumptions of no body forces and no $\theta$ variations (see fig. 1), the momentum equation in cylindrical coordinates is:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -g \frac{\partial \rho}{\partial x} + \frac{1}{r} \frac{\partial}{\partial y} \left( \frac{\partial}{\partial r} \right)$$

(1a)

$$\tau = \mu \frac{\partial u}{\partial y} - \rho \overline{u'v'}$$

(1b)

A model must be chosen for the turbulent transport term $\rho \overline{u'v'}$. Prandtl's mixing length theory suggests using:

$$- \rho \overline{u'v'} = \rho \kappa^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial y}$$

The shear stress term in the momentum equation can then be written:

$$\tau = \mu \frac{\partial u}{\partial y} + \rho \kappa^2 \left| \frac{\partial u}{\partial y} \right| \frac{\partial u}{\partial y}$$
Figure 1. Cylindrical coordinate system used for flow in pipes.

Figure 2. Cartesian coordinate system used for flow between parallel plates or over a flat plate.
An effective viscosity which is a sum of the laminar and turbulent components can be introduced such that:

\[ \tau = \mu_{\text{eff}} \frac{\partial u}{\partial y} \]  

(2)

\[ \mu_{\text{eff}} = \mu + \mu_t \]  

(3)

\[ \mu_t = \rho \gamma^2 \left| \frac{\partial u}{\partial y} \right| \]  

(4)

The turbulent component of viscosity \( \mu_t \), which is caused by globules of fluid exchanging momentum between "layers" of fluid moving at different velocities, is, except very near the wall, generally much larger than the laminar component. A mixing length model which has given good results is:

\[ l = K y (1 - \exp(-y^+/A^+)) \quad \text{for} \quad l \leq 0.089 \delta \]

\[ l = 0.089 \delta \quad \text{for all other} \quad y \]  

(5)

Prandtl originally suggested that \( l=Ky \). Van Driest added the damping factor \( (1-\exp(-y^+/A^+)) \) to account for the laminar sublayer. The cut off at 0.089 \( \delta \) has been found necessary because of the "wake" region in boundary layer flows.

An energy equation which is valid for any pure fluid has been derived using the same assumptions used in obtaining the momentum equation. It is:

\[ \rho u C_p \frac{\partial t}{\partial x} + \rho v \frac{C_p}{J} \frac{\partial t}{\partial y} = \frac{u \partial p}{\partial x} + \frac{1}{3600r} \frac{\partial}{\partial y} (r q) + \frac{\mu_{\text{eff}}}{g \sigma} J \left( \frac{\partial u}{\partial y} \right)^2 \]  

(6a)
\[ q = k \frac{\partial t}{\partial y} - 3600 \rho C_p \bar{t} \bar{v} \]  
\(6b\)

Since the increases in effective viscosity and effective thermal conductivity are due to the same turbulent mechanism, it seems reasonable to assume a Reynolds's analogy for the turbulent shear and heat transfer components. The effective thermal conductivity is then written as a sum of the laminar and turbulent components, and the heat flux term in the energy equation becomes:

\[ q = k_{\text{eff}} \frac{\partial t}{\partial y} \]  
\(7\)

\[ k_{\text{eff}} = k + k_t \]  
\(8\)

\[ k_t = \frac{\mu_t C_p}{Pr_t} = \frac{\rho}{Pr_t} \frac{k^2 C_p}{\bar{t} \bar{v}} \frac{\partial u}{\partial y} \]  
\(9\)

The turbulent component of thermal conductivity \(k_t\) is caused by the actual mixing of globules of fluid at different temperatures, and is generally, except very near the wall, much larger than the laminar component. The turbulent Prandtl number \((Pr_t = \mu_t C_p / k_t)\), which is needed to relate the turbulent components of viscosity and thermal conductivity, has been determined to be about 0.9 for the cases tried so far by comparing the results of the present method with experimental data.
The continuity equation is:

$$\frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0$$

(10)

The momentum, energy, and continuity equations are also valid for parallel plate channel flows if \( r \equiv 1 \) (see fig. 2). The equations valid in Cartesian coordinates \((r \equiv 1)\) may be compared with the turbulent boundary layer equations in reference (31).

An added necessary constraint for internal flows is overall conservation of mass. For no blowing or suction, this is:

$$\dot{m} = \int_A \rho u \, dA = \text{constant}$$

(11)

The boundary conditions are:

$$u(x,0) = 0 \quad v(x,0) = 0$$

$$\left. \frac{\partial u}{\partial y} \right|_c = 0$$

$$t(x,0) = t_w(x) \quad \text{or} \quad k_w \left. \frac{\partial t}{\partial y} \right|_w = q_w(x)$$

$$\left. \frac{\partial t}{\partial y} \right|_c = 0$$
The initial conditions are:

\[ u(0,y) = u(y) \]

\[ t(0,y) = t(y) \]

\[ p(0) = p_0 \]

The initial \( v \)'s, which should be determined from the continuity equation (knowing the initial \( u \)'s and \( \rho \)'s), have always been set equal to zero because only uniform or fully developed initial velocity profiles have been used.

Density is a function of temperature and pressure, and viscosity, thermal conductivity, and specific heat are considered as functions of temperature only. Thus, there are four unknowns \((u, v, t, p)\) and four equations (equations 1, 6, 10, 11). There are sufficient boundary and initial conditions so that the mathematical problem is well posed.
NON-DIMENSIONAL FORMS OF THE GOVERNING EQUATIONS

To keep the numbers in the computer the same order of magnitude, all variables were non-dimensionalized as indicated in the list of symbols. The resulting non-dimensional equations are:

**momentum**

\[
\beta U \frac{\partial U}{\partial x} + \beta V \frac{\partial U}{\partial y} = - \frac{\partial p}{\partial x} + \frac{1}{R} \frac{\partial}{\partial y} \left( R \hat{\mu}_{eff} \frac{\partial U}{\partial y} \right)
\]

\[
\hat{\mu}_{eff} = \mu + \beta \hat{k}^2 \left| \frac{\partial U}{\partial y} \right|
\]

**energy**

\[
\beta U \hat{C}_p \frac{\partial T}{\partial x} + \beta V \hat{C}_p \frac{\partial T}{\partial y} = U \frac{\partial p}{\partial x} + \frac{1}{R} \frac{\partial}{\partial y} \left( R \hat{k}_{eff} \frac{\partial T}{\partial y} + \hat{u}_{eff} \frac{\partial U}{\partial y} \right)
\]

\[
\hat{k}_{eff} = k + \frac{\beta \hat{k}^2 \hat{C}_p}{Pr} \left| \frac{\partial U}{\partial y} \right|
\]

**continuity**

\[
\frac{\partial (\hat{\beta} U R)}{\partial x} + \frac{\partial (\hat{\beta} V R)}{\partial y} = 0
\]

**global continuity**

\[
\hat{m} = \int_A \hat{\beta} U \, dA = \text{constant}
\]
The non-dimensional boundary and initial conditions are:

\[ U(X,0) = 0 \quad V(X,0) = 0 \]

\[ \frac{\partial U}{\partial X} = 0 \]

\[ T(X,0) = T_w(X) \quad \text{or} \quad k_w \frac{\partial T}{\partial Y} = Q_w(X) \]

\[ \frac{\partial T}{\partial Y} = 0 \]

\[ U(0,Y) = U(Y) \]

\[ T(0,Y) = T(Y) \]

\[ P(0) = P_0 \]
NUMERICAL SOLUTION

Obtaining a solution to the internal, turbulent flow problem with property variations is virtually impossible without the aid of a computer. An explicit finite difference technique which is always stable and requires no iterations has been chosen to solve this problem. The method is a Dufort-Frankel formulation used successfully by R. H. Pletcher for external flows with property variations (20, 22). The main difference between the Dufort-Frankel method and other explicit schemes is in the formulation of the derivatives. The Dufort-Frankel finite difference formulation is centered about the point \((i, j)\). Figure 3 shows the finite difference grid being used. Each of the four equations is applied, one at a time, to find one dependent variable. If the dependent variable being sought at the \((i+1, j)\) level appears at the point \((i, j)\) in a derivative term, the variable at \((i, j)\) is replaced by the average of the variable at points \((i+1, j)\) and \((i-1, j)\). This causes the finite difference formulation to become always stable, whereas other types of explicit schemes are restricted in allowed step sizes due to stability. More information may be found in references (8) and (25). The equation can then be solved directly for the variable at \((i+1, j)\). Each equation is used, in turn, to calculate a variable for all \(j\)'s at \(i+1\). Once everything is
Figure 3. The finite difference grid.
known at \( i+1 \), the solution is marched forward one step and all the variables at \( i+1 \) become quantities at \( i \), and those at \( i \) become quantities at \( i-1 \). The solution is thus stepped off until all the desired flow region has been calculated.

It should be noted that the DuFort-Frankel method requires information from two previous \( i \) steps. A standard explicit method which only requires information from one previous step is used for a few initial steps to start the DuFort-Frankel equations. Also, it has been found that the initial stability requirements for the ordinary explicit method allow it to take larger \( \Delta x \)'s for the first few steps and still produce good results. The DuFort-Frankel equations take over after the \( \Delta x \) steps for the ordinary method are smaller than the \( \Delta x \) needed for the DuFort-Frankel formulation. A \( \Delta x \) of a constant multiple (explained in the computer listing in the appendix, typically 0.65 for turbulent flows) of the boundary layer thickness has been found to lead to good predictions with the DuFort-Frankel equations. Using the largest \( \Delta x \) possible means that fewer steps will have to be calculated to cover the flow field and, therefore, less computer time will be used.

DuFort-Frankel Finite Difference Equations

The non-dimensional equations have been set up to be solved numerically. The DuFort-Frankel form of the momentum
The finite difference form of the momentum equation is a slight modification of a true DuFort-Frankel formulation. The \( \hat{\mu}_{\text{eff}, i,j+1} \) and \( \hat{\mu}_{\text{eff}, i,j-1} \) terms (from equation 13) each contain a \( U_{i,j} \) which would be in a derivative. In a strict application of the DuFort-Frankel formulation, this should be represented by an average of \( U_{i+1,j} \) and \( U_{i-1,j} \) which would lead to a slightly more complex solution for \( U_{i+1,j} \). It has been found that equation 12 may be used and will remain stable if the \( \hat{\mu}_{\text{eff}, i,j} \)'s used are the average of \( \hat{\mu}_{\text{eff}, i,j-1} \), \( \hat{\mu}_{\text{eff}, i,j} \), and \( \hat{\mu}_{\text{eff}, i,j+1} \) of equation 13.
\[ \hat{u}_{\text{eff} \, i, j} = \hat{u}_{i, j} + \beta_{i, j} \hat{v}_{i, j}^2 \left| \frac{(U_{i, j} + 1 - U_{i, j-1})}{(\Delta y_+ + \Delta y_-)} \right| \] (13)

The DuFort-Frankel form of the energy equation is:

\[ \beta_{i, j} U_{i, j} \hat{C}_P \hat{i, j} \left( \frac{T_{i+1, j} - T_{i-1, j}}{(\Delta x_+ + \Delta x_-)} \right) + \beta_{i, j} V_{i, j} \hat{C}_P \hat{i, j} \left( \frac{T_{i, j+1} - T_{i, j-1}}{(\Delta y_+ + \Delta y_-)} \right) = U_{i, j} \left( \frac{P_{i+1} - P_{i-1}}{(\Delta x_+ + \Delta x_-)} \right) + \left( \frac{1}{R_j} \frac{2}{(\Delta y_+ + \Delta y_-)} \right) \]

\[ \left\{ \begin{array}{l} \frac{1}{4} \left( R_{i+1} + R_i \right) \left( \hat{k}_{\text{eff} \, i, j+1} + \hat{k}_{\text{eff} \, i, j} \right) \left( T_{i, j+1} - 0.5 \left( T_{i+1, j} + T_{i-1, j} \right) \right) \frac{T_{i, j+1} - T_{i, j-1}}{\Delta y_+} \\ \frac{1}{4} \left( R_j + R_{i-1} \right) \left( \hat{k}_{\text{eff} \, i, j} + \hat{k}_{\text{eff} \, i, j-1} \right) \left( 0.5 \left( T_{i+1, j} + T_{i-1, j} \right) - T_{i, j-1} \right) \frac{T_{i, j+1} - T_{i, j-1}}{\Delta y_-} \end{array} \right\} + \hat{u}_{\text{eff} \, i, j} \left( \frac{U_{i, j+1} - U_{i, j-1}}{\Delta y_+ + \Delta y_-} \right)^2 \] (14)

Since \( \hat{k}_{\text{eff} \, i, j} \) is a function of temperature, statements similar to those about \( \hat{p}_{\text{eff} \, i, j} \) apply and the \( \hat{k}_{\text{eff} \, i, j} \)'s used in equation 14 are actually the average of \( \hat{k}_{\text{eff} \, i, j-1} \), \( \hat{k}_{\text{eff} \, i, j} \), and \( \hat{k}_{\text{eff} \, i, j+1} \) of equation 15.

\[ \hat{k}_{\text{eff} \, i, j} = \hat{k}_{i, j} + \frac{\hat{v}_{i, j} \hat{v}_{i, j}^2 \hat{C}_P \hat{i, j}}{Pr_t} \left( \frac{U_{i, j+1} - U_{i, j-1}}{(\Delta y_+ + \Delta y_-)} \right) \] (15)
The continuity equation used in the DuFort-Frankel method is:

\[
\frac{(R_{i+1} + R_j)}{4 (\Delta X_+ + \Delta X_-)} \left( (\delta U)_{i+1,j+1} - (\delta U)_{i-1,j+1} + (\delta U)_{i+1,j} - (\delta U)_{i-1,j} \right)
+ \frac{(\delta VR)_{i+1,j+1} - (\delta VR)_{i+1,j}}{\Delta Y_+} = 0
\]  

(16)

Standard Explicit Finite Difference Equations

The standard explicit form of the momentum equation (used in the starting procedure) is:

\[
\beta_{i,j} U_{i+1,j} \left( \frac{U_{i+1,j} - U_{i,j}}{\Delta X_+} \right) + \beta_{i,j} V_{i,j} \left( \frac{U_{i,j} - U_{i,j-1}}{\Delta Y_-} \right) * 
= \frac{(P_{i+1} - P_i)}{\Delta X_+} + \left( \frac{2}{R_j (\Delta Y_+ + \Delta Y_-)} \right)
\]

\[
\left( \frac{R_j + R_{j+1}}{4} \right) \left( \frac{(\beta_{eff} i,j + \beta_{eff} i,j)}{4} \right) \left( \frac{U_{i,j+1} - U_{i,j}}{\Delta Y_+} \right)
- \left( \frac{R_j + R_{j-1}}{4} \right) \left( \frac{(\beta_{eff} i,j + \beta_{eff} i,j)}{4} \right) \left( \frac{U_{i,j} - U_{i,j-1}}{\Delta Y_-} \right)
\]  

(17)

The starred term is used when \( V_{i,j} \) is positive, otherwise, to maintain stability it is replaced by:

\[
\beta_{i,j} V_{i,j} \left( \frac{U_{i,j+1} - U_{i,j}}{\Delta Y_+} \right)
\]
The standard explicit form of the energy equation is:

\[
\frac{\beta_{i,j} U_{i,j} C_{p_{i,j}}}{\Delta X_+} \left( \frac{T_{i+1,j} - T_{i,j}}{\Delta X_+} + \beta_{i,j} V_{i,j} C_{p_{i,j}} \frac{T_{i,j} - T_{i,j-1}}{\Delta Y_-} \right)
\]

\[
= U_{i,j} \frac{(P_{i+1} - P_i)}{\Delta X_+} + \left( \frac{2}{R_j \left( \Delta Y_+ + \Delta Y_- \right)} \right)
\]

\[
\left\{ \frac{(R_j + R_{j+1})}{4} \left( \hat{k}_{ef_{i,j}} + \hat{k}_{ef_{i,j+1}} \right) \frac{T_{i,j+1} - T_{i,j}}{\Delta Y_+} \right. 
\]

\[
- \left. \frac{(R_j + R_{j-1})}{4} \left( \hat{k}_{ef_{i,j}} + \hat{k}_{ef_{i,j-1}} \right) \frac{T_{i,j} - T_{i,j-1}}{\Delta Y_-} \right\}
\]

\[
+ \hat{\mu}_{ef_{i,j}} \left( \frac{U_{i,j} - U_{i,j+1}}{\Delta Y_+} \right) \right)^2
\]  

Both \( \hat{\mu}_{ef_{i,j}} \) and \( \hat{k}_{ef_{i,j}} \) are calculated the same way as in the DuFort-Frankel formulation, although the averaging is not really necessary.

The continuity equation for the ordinary explicit method is:

\[
\frac{(R_{j+1} + R_j)}{\Delta X_+} \left\{ (\hat{\sigma}U)_{i+1,j+1} - (\hat{\sigma}U)_{i,j+1} + (\hat{\sigma}U)_{i+1,j} - (\hat{\sigma}U)_{i,j} \right\}
\]

\[
+ \frac{(\hat{\sigma}VR)_{i+1,j+1} - (\hat{\sigma}VR)_{i+1,j}}{\Delta Y_+} = 0
\]
Method of Solution

Once all the variables are known at an i-location, the explicit finite difference equations can be used, one at a time, to calculate each variable for all j’s at i+1. A skeleton flow chart is shown in figure 4 so the order of calculation may be seen. The ordinary explicit finite difference equations are used for the first few steps until the DuFort-Frankel equations take over. The Ax step for the ordinary explicit method is determined by a stability analysis, and for the DuFort-Frankel method a constant multiple of the boundary layer thickness has been found to work well. The development of the equation for $p_{i+1}$ in terms of quantities at i and i-1 will be explained later. The finite difference forms of the momentum equation, 12 and 17, have been solved for $U_{i+1,j}$. Since all the variables are at the i and i-1 levels, except for the pressure, they are all known, and U's can be calculated for all j’s at i+1. Then, the solutions of the finite difference forms of the energy equation, 14 and 18, for $T_{i+1,j}$ are used to calculate T’s for all j’s at i+1. The density is then found as a function of the temperature and pressure. The V's for all j's at i+1 are found by applying the finite difference forms of the continuity equation, 16 and 19. Then, all the variables are incremented to take the next step by making quantities at i+1 become quantities at i, and making quantities at i become quantities at
Figure 4. A skeleton flow chart of the computer program.
i-1. Specific heats, viscosities, and thermal conductivities are then found. So, all the variables are again known at i and i-1 locations, and another step may be taken by repeating this procedure. It should be emphasized that each variable is calculated explicitly at each location with all other variables known in the equation being used. No iterations or simultaneous solutions are required.

To obtain a pressure equation to be used with the DuFort-Frankel finite difference equations, equation 12 is first solved for \( U_{i+1,j} \) and then multiplied by \( \delta_{i,j} \). This equation is then integrated over the cross-section of the pipe or channel using Simpson’s rule for integration. The term containing the \( U_{i+1,j} \)'s, which are unknown at this point, can then be replaced with the mass flow by using the global continuity constraint. Since the pressure is only a function of the axial location, it can be factored out of the integration and the following equation results:

\[
\hat{m} = \int_A \hat{\rho}_{i,j} U_{i+1,j} \, d\hat{A} = \int_A \frac{PB}{PA} \, d\hat{A} + (P_{i+1} - P_{i-1}) \int_A \frac{PC}{PA} \, d\hat{A}
\]

where

\[
P_{\text{A}} = 4 \hat{\rho}_{i,j} U_{i,j} R_{i,j} (\Delta Y_+ + \Delta Y_-) \Delta Y_+ \Delta Y_- \\
+ (\Delta X_+ + \Delta X_-) \Delta Y_- (R_{j+1} + R_{j}) \left( \mu_{\text{eff} \, i,j+1} + \mu_{\text{eff} \, i,j} \right) \\
+ (\Delta X_+ + \Delta X_-) \Delta Y_+ (R_{j} + R_{j-1}) \left( \mu_{\text{eff} \, i,j} + \mu_{\text{eff} \, i,j-1} \right)
\]
\[
PB = -4 \beta_{i,j}^2 R_j \Delta Y_+ \Delta Y_- (\Delta X_+ + \Delta X_-) V_{i,j} (U_{i,j+1} - U_{i,j-1})
+ 2 \beta_{i,j} (\Delta X_+ + \Delta X_-) \Delta Y_- \left( \hat{U}_{\text{eff} i,j+1} + \hat{U}_{\text{eff} i,j} \right)

(U_{i,j+1} - 0.5 U_{i-1,j}) (R_{j+1} + R_j) + 2 \beta_{i,j} (\Delta X_+ + \Delta X_-)

\Delta Y_+ \left( \hat{U}_{\text{eff} i,j} + \hat{U}_{\text{eff} i,j-1} \right) (U_{i,j-1} - 0.5 U_{i-1,j})

(R_j + R_{j-1}) + 4 \beta_{i,j}^2 U_{i-1,j} U_{i,j} R_j (\Delta Y_+ + \Delta Y_-) \Delta Y_+ \Delta Y_-
\]

\[
PC = -4 \beta_{i,j} R_j (\Delta Y_+ + \Delta Y_-) \Delta Y_+ \Delta Y_-
\]

This equation can then be solved for the pressure at \(i+1\), since all the variables are known at the \(i\) and \(i-1\) locations.

To obtain a pressure equation to be used with the ordinary explicit finite difference equations, the above procedure is used, except equation 17 is used as a starting point instead of equation 12. In the development for the pressure equation, \(\beta_{i,j} U_{i+1,j}\) is being integrated over the cross section instead of \(\beta_{i+1,j} U_{i+1,j}\). The equation for \(U_{i+1,j}\) was multiplied by \(\beta_{i,j}\) because the \(\beta_{i+1,j}\)'s are not known at this point. Although integrating \(\beta_{i+1,j} U_{i+1,j}\) over the cross section would have been preferred, it is not really necessary because the finite difference approximations require a fine grid for convergence, so the difference between the value of a quantity at two adjacent points is negligible. An analytical expression was originally used for the pressure at \(i+1\). The pressures calculated this way did not lead to good results. The present method was tried in order to keep the
arrangement of the difference equations as consistent as possible. Very good results have been found using this method.

The differential equations in cylindrical coordinates have a singularity at \( r=0 \). L'Hopital's rule was used to obtain expressions valid at the centerline.

Wall slopes for temperature and velocity, which are required for engineering parameters like skin friction coefficients and Stanton and Nusselt numbers, are calculated from the velocities and temperatures already obtained from the finite difference solution by either a straight line fit between the wall and the first point out or a third degree polynomial fit using the four points nearest the wall. For the straight line fit, using temperature as an example, the wall slope is:

\[
\left. \frac{\partial T}{\partial Y} \right|_W = \frac{T_{i,2} - T_{i,1}}{\Delta Y_W}
\]

For the polynomial fit, the wall slope is:

\[
\left. \frac{\partial T}{\partial Y} \right|_W = (\text{PSA}) T_{i,1} + (\text{PSB}) T_{i,2} + (\text{PSC}) T_{i,3} + (\text{PSD}) T_{i,4}
\]

with

\[
\text{PSB} = \frac{1 + DYM + DYM^2}{\Delta Y_W DYM^2}
\]

\[
\text{PSC} = -\frac{1 + DYM + DYM^2}{\Delta Y_W (1 + DYM) DYM^3}
\]
\[
\text{PSD} = \frac{1}{\Delta Y_w (1 + \text{DYM} + \text{DYM}^2) \text{DYM}^3}
\]

\[
\text{PSA} = - \text{PSB} - \text{PSC} - \text{PSD}
\]

\(\text{DYM}\) is the multiplication factor used for the geometric grid spacings in turbulent flows \(\Delta Y_j = \text{DYM} \Delta Y_{j-1}\). The equation for the wall slope is valid for laminar flow grids (equal spacing) if \(\text{DYM} = 1\). When the slopes computed by the two methods closely agree, then the \(y\) grid spaces are fine enough such that a good approximation for the wall slope has been found.

Thermal entry cases require a fully developed velocity profile at the inlet. For laminar flows, a parabolic profile can easily be generated, while for turbulent flows, a subroutine \((\text{FDTVP})\) has been added. This subroutine solves the momentum equation with constant property and fully developed flow assumptions. The pressure gradient term has been replaced, by using a force balance, with a term involving wall shear. The resulting equation using the mixing length concept is:

\[
\frac{d\rho}{dx} = -\frac{2 r_w}{r_w} \frac{1}{r} \frac{d}{dr} \left[ r (\mu + \rho \kappa^2 |\frac{du}{dr}|) \frac{du}{dr} \right]
\] (20)
In order to obtain a numerical solution, the above second order equation was transformed into a system of two first order equations:

\[
\frac{dF}{dr} = \frac{-2 \frac{\tau_w}{r_w} - \mu \frac{F}{r} + (2 \rho \lambda \frac{d\lambda}{dr} + \frac{\rho \lambda^2}{r}) F^2}{(\mu - 2 \rho \lambda^2 F)}
\]

\[
\frac{du}{dr} = F
\]

The wall shear is determined by using the Karman-Prandtl universal friction factor equation for turbulent flow in smooth pipes. Knowing the velocity at the wall (zero) and the slope (from the wall shear), a solution is stepped off using a fourth order Runge-Kutta procedure. Although equation 20 could have been integrated once analytically and reduced to a first order equation before being solved numerically, the above method was used because a program, which was slightly modified to become subroutine FDTVP, was readily available to solve the equation as a first order system. Subroutine FDTVP has been set up so different mixing length models can easily be tried by changing subfunction F. This subroutine was very useful (and inexpensive) in finding a mixing length model which leads to a good fully developed turbulent velocity profile.
The present computer program can be used to calculate flows with or without property variations. If variable properties are desired, several models are available in the program. The perfect gas law, for which the gas constant must be input, is available for density variations. Power laws on temperature are available for viscosity, thermal conductivity, and specific heat variations. The constants which must be input for the power law (explained in the computer listing in the appendix) have been determined by comparing the results of the power law with data tabulated in NBS Circular 564 (9). Sutherland's equation, for which the constants are input (also explained in the appendix), is also available as a model for viscosity and thermal conductivity. The fluid property variations mentioned above are sufficient for most flow situations. If other property variations or tables are required, small changes could be made in the program to handle these cases. For the present analysis, the model which compared best with the tabulated data in NBS Circular 564 was used when the results of the present method were compared to experimental data. For gases, Sutherland's equation was usually used for viscosities and thermal conductivities, the perfect gas law was used for density variations, and the specific heat was usually kept constant. When comparing to other predictions, their models were used.
Consistency and Stability

Consistency and stability are major concerns when using finite difference methods. To satisfy consistency, the solution of the finite difference equations must also be the solution of the partial differential equations being approximated. To satisfy stability, any errors, usually due to rounding off by the computer, must not increase in magnitude as the solution is stepped off.

Consistency is usually studied by using a Taylor series expansion and observing the order of magnitude of the terms being neglected by the finite difference equations. The neglected terms are known as the truncation error. Using the derivative of \( U \) in the \( Y \) direction as an example, the truncation error may be seen for the standard explicit method:

\[
\frac{\partial U}{\partial Y} = \frac{U_{i,j+1} - U_{i,j}}{\Delta Y} - \frac{\partial^2 U}{\partial Y^2} \frac{\Delta Y}{2!} - \frac{\partial^3 U}{\partial Y^3} \frac{\Delta Y^2}{3!} - \ldots
\]

The truncation error for this formulation is the order of \( \Delta Y \).

The same derivative can be written using the DuFort-Frankel scheme:

\[
\frac{\partial U}{\partial Y} = \frac{U_{i,j+1} - U_{i,j-1}}{2 \Delta Y} - \frac{\partial^3 U}{\partial Y^3} \frac{\Delta Y^2}{3!} - \frac{\partial^5 U}{\partial Y^5} \frac{\Delta Y^4}{5!} - \ldots
\]

The truncation error here is seen to be the order of \( \Delta Y^2 \). It would be hoped that any truncation error is at most the order of \( \Delta Y \) or \( \Delta X \), so the solution of the difference equations would converge to the solution of the differential equations.
as the grid spacing is refined. This is true for all the derivatives being used by the present method, except the second derivative term in the DuFort-Frankel equations. This term can be written (for constant properties):

\[
\frac{\partial^2 U}{\partial Y^2} = \frac{U_{i,j+1} - U_{i+1,j} - U_{i-1,j} + U_{i,j+1}}{\Delta Y^2} + \frac{\partial^2 U}{\partial X^2} \frac{\Delta X^2}{\Delta Y^2} - \frac{\partial^4 U}{\partial Y^4} \frac{\Delta X^2}{\Delta Y^2} + \ldots
\]

The truncation error may be said to be the order of \((\Delta X^2 / \Delta Y^2)\), which would require that the mesh be refined in such a manner that \(\Delta X\) would go to zero faster than \(\Delta Y\). Luckily the whole truncation error term is really \(\partial^2 U / \partial X^2 (\Delta X^2 / \Delta Y^2)\), and for boundary layer flows \(\partial^2 U / \partial X^2\) is negligibly small, so \(\Delta X\) could be the order of \(\Delta Y\) and still satisfy the consistency conditions. To do a complete consistency analysis, the truncation error must be found for every derivative in every equation.

For variable properties this would be a laborious task. R. H. Pletcher (20, 21, 22) has successfully used these finite difference equations for boundary layer flows, so the analysis was never completely done for the present study (more details are shown in reference (21)). The proof that the consistency conditions were satisfied can be seen when the results of the present method are compared to experimental data and other predictions.

To study stability, von Neumann's condition (18) was applied to both the standard explicit and the DuFort-Frankel equations. This condition assumes an error of the form:
\[ \delta_{i,j} = e^{\alpha X} e^{i\beta Y} \]

The error \( \delta_{i,j} \) was substituted in the derivative terms for the dependent variable. The coefficients of the derivatives were assumed to be error free to simplify the analysis. After rearranging and taking the real part of the equation, it is desired that

\[ |e^{\alpha \Delta X}| < 1 \]

so errors do not propagate in the direction being stepped off. For the DuFort-Frankel equations, it is found that the above condition is always satisfied, so the method is always stable. For the ordinary explicit method, a maximum allowable step size (\( \Delta X \)) is found for which the error will not grow and the solution will remain stable. The equation for \( \Delta X \) used with the ordinary explicit method is:

\[
(\Delta X)_{\text{max}} \max_{j=1,N} \left\{ \frac{|V_{i,j}|}{U_{i,j}} \frac{(R_j + R_{j+1}) Z^+}{\Delta Y_-} + \frac{(R_j + R_{j-1}) Z^-}{\Delta Y_+ (\Delta Y_+ + \Delta Y_-)} \right\} \leq 1
\]
where \( z^+ = \max \left\{ \left( \rho_{\text{eff}, i, j} + \rho_{\text{eff}, i, j+1} \right) \left( \frac{k_{\text{eff}, i, j} + k_{\text{eff}, i, j+1}}{C_{\text{p}, i, j}} \right) \right\} \)

\( z^- = \max \left\{ \left( \rho_{\text{eff}, i, j} + \rho_{\text{eff}, i, j-1} \right) \left( \frac{k_{\text{eff}, i, j} + k_{\text{eff}, i, j-1}}{C_{\text{p}, i, j}} \right) \right\} \)

Even though certain assumptions were made to simplify this stability analysis, the actual proof that the equations will remain stable can be seen from the solutions obtained.
RESULTS

A computer program using the finite difference equations was developed to solve the turbulent internal flow problem with heat transfer and property variations. The program was tested and debugged by comparing the calculated results with experimental data and other predictions. The first case to be tested was laminar constant property flow in the entrance region between parallel plates. As can be seen in figures 5 and 6, good agreement between the present method (results of the computer program) and the predictions of Schlichting (29, 28), Bodoia (6), and Wang and Longwell (34), is found for some velocity profiles and the pressure drop. These results verified that the momentum, continuity, and pressure equations in Cartesian coordinates were well formulated. Since Wang and Longwell solved the full Navier-Stokes equations, this showed that the boundary layer assumptions were valid for this problem.

To check the momentum, continuity, and pressure equations in cylindrical coordinates, a laminar constant property pipe flow case was run. Figure 7 shows the agreement between the results for pressure drop by the present method and the predictions of Hornbeck (10) and Langhaar (12).

Once it was reasonably certain that the program was producing good results for the laminar hydrodynamic problem, a
Figure 5. Comparison of the present method with some predictions for velocity profiles for developing constant property laminar flow between parallel plates.
Figure 6. Comparison of the present method with some predictions for pressure drop for laminar constant property flow in the inlet section of a parallel plate channel.
Figure 7. Comparison of the present method with some predictions for pressure drop for laminar constant property flow in the inlet section of a pipe.
turbulent case was run. Several mixing length models were tried, and the one described earlier in this paper produced the best results. Figure 8 shows the comparison between the present method and the data of Nikuradse (17) for a fully developed velocity profile on "law of the wall" coordinates. Figures 9 and 10 show the results of the present method compared to the data of Barbin and Jones (4) for several velocity profiles and the pressure distribution in the inlet section of a pipe.

To test the formulation of the energy equation, a laminar parallel plate channel case was run with constant wall temperature and uniform velocity and temperature profiles at the entrance. Figure 11 presents comparisons between the present method and the predictions of Schade and McEligot (26) for Nusselt number and wall to bulk temperature ratios.

A constant heat flux case for developing laminar flow in a pipe was run. Figures 12 and 13 show comparisons between the present method and predictions of Bankston and McEligot (3) for Nusselt number, ratio of bulk to initial temperatures, ratio of wall to bulk temperatures, pressure drop, and friction factor distributions along the length of a pipe. Except for one point on the wall to bulk temperature plot, the agreements are well within four percent.
Figure 8. Comparison of the present method with data for a fully developed constant property turbulent velocity profile.
Figure 9. Comparison of the present method with data for constant property turbulent velocity profiles in the inlet section of a pipe.
Figure 10. Comparison of the present method with data for pressure distribution for constant property turbulent flow in the inlet section of a pipe.
Figure 11. Comparison of the present method with a prediction for Nusselt numbers and wall to bulk temperature ratios for laminar flow of air between constant temperature parallel plates with uniform initial velocity and temperature profiles.
Figure 12. Comparison of the present method with predictions for Nusselt number, bulk to initial temperature ratio, and wall to bulk temperature ratio distributions in the inlet section of a pipe for laminar flow of air with constant wall heat flux and uniform initial velocity and temperature profiles.
Figure 13. Comparison of the present method with predictions for wall shear and pressure distributions in the inlet section of a pipe for laminar flow of air with constant wall heat flux and uniform initial velocity and temperature profiles.
A compressible laminar flat plate case of Van Driest (32) was run on the present program by changing the pressure calculation to give no pressure drop. Figures 14 and 15 show results of the present method compared to Van Driest's predictions for velocity and temperature profiles. The good agreement on the temperature profile verified that the viscous dissipation term in the energy equation was properly formulated. Stanton numbers and skin friction coefficients calculated by the present method were within two to three percent of Van Driest's predictions.

The application for which this method has the greatest advantage over other solutions is turbulent flow in a pipe with heat transfer and property variations. Figure 16 shows Nusselt numbers calculated by the present method compared to the data of Mills (15) for turbulent flow of air in a pipe with constant wall heat flux. One case starts with uniform initial temperature and velocity profiles in which both the temperature and velocity boundary layers develop simultaneously. The other case, a thermal entry problem, starts with a fully developed velocity profile and uniform temperature profile. The agreement found for these cases verified the capability of this method to calculate these flow situations.

A couple of high heating rate cases of Perkins and Worsoe-Schmidt (19) for turbulent flow of nitrogen in a pipe
Figure 14. Comparison of the present method with a prediction for velocity profiles for compressible laminar flow of air over a constant temperature flat plate.
Figure 15. Comparison of the present method with a prediction for temperature profiles for compressible laminar flow of air over a constant temperature flat plate.
Figure 16. Comparison of the present method with data for Nusselt number distributions in the inlet section of a pipe for turbulent flow of air with constant wall heat flux.
were run. Figures 17 and 18 show results of the present method compared to the data of Perkins and Worsoe-Schmidt and predictions of Bankston and McEligot (3) for wall temperature and pressure distributions along the axis of the pipe. Figure 19 shows the Nusselt number distribution calculated by the present method compared to the data of Perkins and Worsoe-Schmidt. Part of the reason for the disagreement near the entrance is because the wall temperatures calculated by the present method are too high there, which causes a larger wall to adiabatic wall temperature difference, which in turn causes a smaller convective coefficient and a smaller Nusselt number. Figure 17 shows results of the present method for wall temperature distributions for case 106 using two slightly different mixing length models. One method uses the standard model, shown in equation 5, and the other uses the local density in the damping factor. Great differences are noticed in the wall temperature distributions. This points out the importance of selecting a good turbulence model as well as the need for more work to be done in this area.

Figure 20 shows comparisons between the present method and the data of Back, Cuffel, and Massier (2) for turbulent flow of air through a tube with wall cooling. The results of the present method for both of their cases shown were so close that one line was used to represent both predictions.
Figure 17. Comparison of the present method with data and another prediction for wall to initial temperature ratios for turbulent flow of nitrogen in the thermal entrance region of a pipe.
Figure 18. Comparison of the present method with data and another prediction for pressure distributions for turbulent flow of nitrogen in the thermal entrance region of a pipe.
Figure 19. Comparison of the present method with data for Nusselt number distributions for turbulent flow of nitrogen in the thermal entrance region of a pipe.
Figure 20. Comparison of the present method with data for a Stanton-Prandtl number group versus Reynolds number based on $x$ for turbulent flow of air in the entrance region of a pipe with constant wall temperature and uniform initial velocity and temperature profiles.
A compressible turbulent flat plate case of Neal (16) at a Mach number of 6.8 with wall cooling was run. Figure 21 shows results of the present method compared to the data of Neal and predictions of Pletcher (20), Van Driest (33), and Spalding and Chi (7, 30) for skin friction and Stanton numbers. The predictions and data for this figure were found in reference (20). It might be expected that the present method and Pletcher's prediction would agree exactly since both use very similar finite difference formulations to solve the boundary layer equations. Some possible causes for the slight disagreement (five per cent at most) are that different methods were used to model the effective viscosity, and the energy equation used in Pletcher's formulation is based on total enthalpy, while the present method uses static temperature.

Figure 22 compares the present method with the data of Allen and Eckert (1) for the convective coefficient distribution for turbulent flow of water (Pr=7) in the inlet region of a pipe with constant wall heat flux. The good agreement indicates that the present method has potential for producing good results for flows in which the Prandtl number differs significantly from one. A turbulent oil flow case (Pr=75) of Malina and Sparrow (13) was also run. The predictions of the present method were not completely satisfactory for this case, so work is being done to see what the problems are.
Figure 21. Comparison of the present method with data and other predictions for skin friction coefficients and Stanton numbers versus Reynolds number based on $x$ for compressible turbulent flow of air over a constant temperature flat plate.
Figure 22. Comparison of the present method with data for the convective coefficient distribution for turbulent flow of water (Pr=7) in the thermal entrance region of a pipe with constant wall heat flux.
DISCUSSION

The original problem being analyzed was steady laminar or turbulent flow through a pipe or parallel plate channel with heat transfer and property variations. A simple modification of the computer program enabled flat plate boundary layers to also be calculated. The agreement between the results produced by the present program and experimental data and other predictions verified that reasonable assumptions had been made in the analysis. The results for laminar flows indicated that the boundary layer equations applied, and that the finite difference formulations were basically correct. The computer program can then be used as a tool to test turbulence models. It was chosen to model the shear and heat flux terms (equations 1b and 6b) with "effective" viscosities and thermal conductivities (as in equations 2 and 7). Any turbulence model fitting this form can be tested by changing a subroutine. Other models may be tested by modifying the main program. The only model considered so far has been Prandtl's mixing length model. Several mixing length formulas were tried, and the one which produced the best results is shown in equation 5 using $K=0.40$, $A^+=26$, the cut off for the "wake" region at $l/\delta=0.089$, and $y^+$ evaluated at wall conditions. A turbulent Prandtl number (used in equation 9) of 0.9 has been found to produce satisfactory results.
for all gas and water flow cases.

The size of the grid is an important consideration in any finite difference analysis since some of the approximations used are only valid in the limit as $\Delta X$ and $\Delta Y$ go to zero. For this reason a very fine grid (small $\Delta X$'s and $\Delta Y$'s) would be desirable so that the solution of the finite difference equations converges to the solution of the governing equations which are being approximated. With a finer grid, more points are needed to cover the flow field, and more calculations and computer time are required. So, to minimize computer time, the coarsest grid for which a good solution can be obtained is the most desirable. To determine the optimum grid spacing, a coarse grid may be used, and then the solutions obtained with increasingly finer grids may be compared. When the solutions no longer change, the grid spacing is fine enough so that the solution of the finite difference equations has converged to the solution of the governing equations. The grid spacing also affects the stability of the solution. Instability is caused by allowing round off errors to grow as the solution progresses.

The potential user of this program would probably not want to worry about convergence and stability of the solution, so some general guidelines for grid spacings have been determined. These are shown with an explanation of the input at the beginning of the program listing shown in the appen-
An equal spaced $Y$ grid has worked well for laminar flows, while a geometric progression on grid spaces with the finest part of the grid near the wall has worked best for turbulent flows. The geometric grid being used is such that $\Delta Y_j = DYM \Delta Y_{j-1}$. The values that may be used for $DYM$ are explained in the computer listing in the appendix, and are usually between 1.0 and 1.15 to produce good results. About 40 or 50 grid spaces in the radius of a pipe or half width of a channel has been found to work in most cases. For turbulent flows, the fine grid is needed near the wall, because the velocity and temperature gradients are larger there, while a coarse grid will do near the center where the gradients are small. Also, it is desirable to have a grid point within the laminar sublayer so the wall slopes may be calculated better. The standard explicit method calculates its own $x$ steps, and constant multiples ($DXF$) of the boundary layer thickness have been found to work well for the DuFort-Frankel method. For turbulent flows, $DXF$ is usually 0.65, although 0.5 was used for some heat transfer cases, and 0.3 was used for the high heat transfer case (case 140) of Perkins and Worsoe-Schmidt (19). Much larger values (explained in the appendix) may be taken for laminar flows.

The case of Allen and Eckert (1) for turbulent flow of water ($Pr=7$) in a pipe showed that liquids can be handled with the program. A case of Malina and Sparrow (13) was
tried for turbulent flow of oil (Pr=75) in a pipe. More time must be spent to find out why there was disagreement between their data and the results of the present method. Some possible causes for the disagreement are that the grid spacings must be refined more, a better turbulent model must be found for either shear or heat flux, the fluid properties could be modeled better, their data was misinterpreted, or the program may not work for some cases like this one. There is a good possibility that the cause for the disagreement can be found and corrected so that the present method can produce good results for this case.

The present program will solve the coupled partial differential equations in either cylindrical or Cartesian coordinates for two-dimensional boundary layer flows. It can also calculate the necessary axial pressure gradients and satisfy total conservation of mass for internal flows. The program has been found to conserve mass flow for all internal flow cases to within half a percent. The viscous dissipation term has been kept in the energy equation. Turbulent shear and heat flux terms have been modeled with "effective" viscosities and thermal conductivities, and different models may be tested by changing only one subroutine. Several boundary and initial conditions for temperature and velocity are available to solve most pipe, parallel plate channel, and flat plate flows. Small changes in the program would allow
flows in annuli or flows with blowing and suction to be calculated, although the results would have to be tested before confidence could be put in the predictions for these cases. Calculating flows with area change looks promising although the radial momentum equation would probably have to be included in the analysis. Several models for fluid properties are used in the program, and other models or tables may be added by making small changes. Good results have been obtained for laminar and turbulent flow of gases in pipes and channels and over flat plates. Work is being done at the present to verify the results obtained for high Prandtl number fluids.

This analysis has shown that the DuFort-Frankel type finite difference equations are well suited to solve flows governed by equations of boundary layer form. Except very near the leading edge, the DuFort-Frankel equations have been able to take much larger steps than the ordinary explicit method. Most turbulent flow cases with heat transfer and property variations take about one minute, and never more than two, on the IBM 360/65.


APPENDIX

The appendix contains a listing of the computer program which uses the explicit finite difference equations developed in this analysis to solve the boundary layer equations for laminar or turbulent pipe, channel, or flat plate flows with heat transfer and property variations.
Main Program

The main program reads in the data, initializes all quantities, coordinates the subroutines, and outputs the results.
THE FOLLOWING IS THE FORMAT TO BE USED FOR THE INPUT CARDS.
ALL FORMATS ARE 6G12.5 EXCEPT THE TITLE CARD WHICH IS 72H AND THE ONE CARD WITH INTEGERS ON IT WHICH IS 8110.

QUANTITIES TO BE INPUT

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>WHEN USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TITLE&quot;</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>DENS, VISC, USTART, PSTART, RAD, DELYI</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>VDK, VW, DYM, PRT, DXF, DXFM</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>RCON, TSTART, TWALL, CPCON, HTK, QW</td>
<td>DENS. LE. 0</td>
</tr>
<tr>
<td>IEND, IOUT, NSTA, KNC, NY, LORT, NT, NS</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>OREX, ORET, OXOUT, OINC, OSTOP</td>
<td>ALWAYS</td>
</tr>
<tr>
<td>OUT(K), K=1, NSTA</td>
<td>NSTA. GT. 0</td>
</tr>
<tr>
<td>UX, DD1, VV1</td>
<td>USTART. LE. 0</td>
</tr>
<tr>
<td>XVO, XVTO, AO, SPLV</td>
<td>VISC. LT. 0</td>
</tr>
<tr>
<td>CPO, CPTO, CO</td>
<td>CPCON. LT. 0</td>
</tr>
<tr>
<td>HTKO, HTKT0, BO, SPLK</td>
<td>HTK. LT. 0</td>
</tr>
<tr>
<td>TW(J), J=1,</td>
<td>TWALL. LT. 0</td>
</tr>
<tr>
<td>TX(J), J=1,</td>
<td>TWALL. LT. 0</td>
</tr>
</tbody>
</table>

THE FOLLOWING IS AN EXPLANATION OF THE INPUT

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DENS</td>
<td>=DENSITY(LBM/FT3) FOR CONSTANT PROPERTY FLOW</td>
</tr>
<tr>
<td></td>
<td>.LE.0 INDICATES VARIABLE PROPERTY FLOW</td>
</tr>
<tr>
<td>VISC</td>
<td>=VISCOSITY(LBM/FT-SEC) FOR CONSTANT VISCOSITY</td>
</tr>
<tr>
<td></td>
<td>.LT.0 INDICATES VARIABLE VISCOSITY (DENS MUST BE .LT.0)</td>
</tr>
<tr>
<td>USTART</td>
<td>=VELOCITY(FT/SEC) FOR UNIFORM INITIAL PROFILE</td>
</tr>
<tr>
<td></td>
<td>.LE.0 INDICATES FULLY DEVELOPED PROFILE WILL BE USED</td>
</tr>
<tr>
<td>PSTART</td>
<td>=INITIAL STATIC PRESSURE (LBF/FT2)</td>
</tr>
<tr>
<td>RAD</td>
<td>=RADIUS FOR PIPE (FT)</td>
</tr>
<tr>
<td></td>
<td>=0 INDICATES FLAT PLATE</td>
</tr>
<tr>
<td></td>
<td>.LT.0 INDICATES CHANNEL</td>
</tr>
<tr>
<td>DELYI</td>
<td>=Y GRID SPACE NEAREST THE WALL FOR TURBULENT FLOWS OR</td>
</tr>
<tr>
<td>LAMINAR FLAT PLATES. FOR TURBULENT FLOWS, DELYI SHOULD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BE A VALUE SUCH THAT Y*(2).LE.4, AND FOR TURBULENT HEAT</td>
</tr>
</tbody>
</table>
TRANSFER Y+(2).*E.1 FOR X/D 'S .GT.1
VDK = VAN DRIEST CONSTANT K (SUGGEST USING 0.40)
VW = NORMAL VELOCITY COMPONENT TO WALL
DYM = MINIMUM Y GRID GEOMETRIC FACTOR (USED IN TURBULENT FLOW)
A DYM OF 1.04 (LARGER FOR FLAT PLATE) FOR INPUT IS GOOD.
THE ADJUSTED DYM (BECAUSE THE SPACES MUST FILL THE WHOLE PIPE OR CHANNEL) SHOULD NOT BE GREATER THAN 1.15 FOR MOST CASES (1.08 OR LESS FOR HIGH PRESSURE GRADIENT OR HIGH HEAT TRANSFER)
PRT = TURBULENT PRANDTL NUMBER (SUGGEST USING 0.9)
DXF =DELX/(BOUNDARY LAYER THICKNESS) FOR D-F EQUATIONS
FOR MOST TURBULENT CASES 0.65 IS OK. FOR HIGHER PRESSURE GRADIENTS OR HIGHER HEAT TRANSFER 0.3 OR LESS MAY BE NEEDED. FOR LAMINAR CONSTANT PROPERTY FLOWS, 0.0008 TIMES THE TUBE REYNOLDS NUMBER MAY BE USED. MAY NEED SMALLER DXF FOR HEAT TRANSFER.
LT.0 FOR DXF INCREASES SLOWLY FROM |DXF|
DXFM =MAXIMUM DXF IF DXF.LT.0, OR STARTING METHOD IS USED
SAME AS DXF
RCON =GAS CONSTANT R FOR DENSITY=P/RT
.LE.0 FOR DENSITY=-RCON
TSTART =TEMPERATURE(DEG R) FOR UNIFORM INITIAL PROFILE
TSTART =WALL TEMPERATURE(NEG R) FOR CONST WALL TEMP
=0 INDICATES CONSTANT WALL HEAT FLUX
.LT.0 FOR TEMP OR HEAT FLUX INPUT ALONG WALL
CPCON =SPECIFIC HEAT (BTU/LBM-R)
.LT.0 INDICATES VARIABLE SPECIFIC HEAT
HTK = THERMAL CONDUCTIVITY (BTU/HR-FT-R)
.LE.0 INDICATES VARIABLE THERMAL CONDUCTIVITY
QW =WALL HEAT FLUX (BTU/HR-FT2) FOR CONSTANT HEAT FLUX
IEND =LAST I STATION TO BE COMPUTED
THIS IS A SAFETY FEATURE. SET IEND HIGH (1000 OR MORE)
IF A CASE MIGHT NEED MORE STEPS. MOST CASES TAKE ABOUT 400 STEPS
IOUT =NUMBER OF I STEPS BETWEEN OUTPUT
.LE.0 NOT USED
NSTA =NUMBER OF REX,RET, OR X STATIONS TO BE READ IN FOR OUTPUT
.LE.0 NOT USED

GT.0 READ IN ANOTHER CASE AFTER THIS ONE

LE.0 THIS IS THE LAST CASE TO BE READ IN

NY =NUMBER (MUST BE EVEN) OF Y GRID SPACES IN PIPE OR C-CHANNEL
SUGGEST USING 50. MORE MAY BE NEEDED FOR SOME CASES.

LORT .GE.0 TURBULENT CASE

LT.0 LAMINAR CASE

NT =NUMBER OF DATA POINTS INPUT FOR WALL TEMPERATURE

LT.0 -NT=NUMBER FOR HEAT FLUX

NS =ICOUNT WHEN O-F EQUATIONS START

SUGGEST USING 20

OREX =VALUE OF REX BETWEEN OUTPUT

=0 NOT USED

LT.0 INPUT REX STATIONS (NSTA.LT.0)

ORET SAME AS OREX, BUT FOR RET

OXOUT SAME AS OREX, BUT FOR X

OINC =GEOMETRIC PROGRESSION FACTOR FOR REX, RET, OR X OUTPUTS

.LE.0 NOT USED

OSTOP =LARGEST VALUE OF REX, RET, OR X TO BE OUTPUT

.LE.0 NOT USED

OUT(K) =VALUES OF REX, RET, OR X TO BE OUTPUT

TW(J) =WALL TEMP OR HEAT FLUX VALUES TO BE INPUT

TX(J) =X VALUES FOR TW(J)

UO =AVERAGE VELOCITY FOR FULLY DEVELOPED PROFILE

DD1 =DENSITY AT INITIAL PRESSURE AND BULK TEMPERATURE

VV1 =VISCOITY AT INITIAL BULK TEMPERATURE

C FOR SPLV.LT.0, VISC(J)=XVO*[T(J)/XVTO]**AO

C FOR SPLV.GE.0, VISC(J)=SPLV*SQRT(T(J))/(1.0+AO/T(J))

C FOR SPLK.LT.0, K(J)=HTKOCT(J)/HTKTO)**BO

C FOR SPLK.GE.0, K(J)=SPLK*SQRT(T(J))/(1.0+BO/T(J))

DIMENSION VP(102),DP(102),DWR(102),DWRP(102),DLYM(102),
1UPLUS(102),OUT(102),DM(102),TTW(50),TX(50),UUJ(102),UUY(102)
CCMON/MVAR/U1(102),UM(102),UP(102),V(102),R(102),EV(102),
1EK(102),D(102),C(102),DLYP(102),DLYT(102),C0EF(102),C0EFM(102),
2SMUP(102),SMUM(102),SKCP(102),SKCM(102),SPMAX(102),SMMAX(102),
3DELXP,DELXT,DELPT,DELPP,FP,PDOP,PP,PSO,SUMPO,CHPF,ICOUNT,N,NPI,
!NM1,JDEL,KJDEL,NS
COMMOM/MEF/XV(102),XK(102),Y(102),YPLUS(102),POUT(102),YPMOD(102),
1XL(102),VDK,DENS,PRT,TW,UST,VDWALL,UCONV
COMMOM/MTEMP/T(102),TP(102),TM(102),H(102),TSTART,TCONV,JTDEL
5
KNC=-1
PRST=-1.0
GCON=32.174
XJCON=777.66
BU=1.0
BT=1.0
BXL=1.0
CHPF=8.0
CH=1.0
FP=1.0
TCNV=0.0
TO=0.0
CO=0.0
CPTO=1.0
XVT0=0.0
HTKTO=0.0
SPLV=-1.0
SPLK=-1.0
HIDX=0.0
DXFT=1.0
NTT=1
OXDIST=0.0
T1=0.0
CREADING IN THE INPUT
10 READ(5,1001)
1001 FORMAT(72H
1
1
WRITE(6,1900)
1900 FORMAT('1*)
1901 FORMAT('0*,,,//)
WRITE(6,1001)
20 READ(5,1002) DENS,VISC,USTART,PSTART,RAD,DELYI
1002 FORMAT(6G12.5)
WRITE(6,1002) DENS,VISC,USTART,PSTART,RAD,DELY
READ(5,1002) VDK,VWALL,DMY,PRT,DXF,DXFM
WRITE(6,1002) VDK,VWALL,DMY,PRT,DXF,DXFM
IF (DENS.GT.0.0) GO TO 40
30 READ(5,1002) RCON,TSTART,TWALL,CPCON,HTK,QW
WRITE(6,1002) RCON,TSTART,TWALL,CPCON,HTK,QW
40 READ(5,1003) IEND,IOUT,NSTA,KNC,NC,LY,LORT,NT,NS
1003 FORMAT(8110)
WRITE(6,1003) IEND,IOUT,NSTA,KNC,NC,LY,LORT,NT,NS
IF (NT.LE.0) NTT=-1
NT=1ABS(NT)
READ(5,1002) OREX,ORET,OXOUT,OINC,OSTOP
WRITE(6,1002) OREX,ORET,OXOUT,OINC,OSTOP
IF (NS.GT.1) DXFT=-1.0
IF (DXF.LT.0.0) DXFT=-1.0
DXF=ABS(DXF)
IF (NSTA.LE.0) GO TO 50
READ(5,1004) (OUT(K),K=1,NSTA)
1004 FORMAT(6G12.5)
WRITE(6,1004) (OUT(K),K=1,NSTA)
CINITIALIZING AND NONDIMENSIONALIZING
50 IF (USTART.LE.0.0) GO TO 60
UO=USTART
JDEL=1
51 IF (DENS.LE.0.0) GO TO 65
DO=DENS
DMM=DO
XVO=VISC
GO TO 80
60 READ (5,1002) UO,DD1,VV1
WRITE(6,1002) UO,DD1,VV1
BU=-1.0
GO TO 51
65 IF (TSTART.LE.0.0) GO TO 70
TO=TSTART
66 IF (RCON.LE.0.0) GO TO 72
DO=PSTART/(RCON*TO)
IF(VISC.LT.0.0) GO TO 75
   XVO=VISC
68 IF(CPCON.LE.0.0) GO TO 76
   CPO=CPCON
69 IF(HTK.LE.0.0) GO TO 77
   HTKO=HTK
   GO TO 80
70 CONTINUE
C 70 READ TEMP PROFILE, NON-DIM, TO, BT=-1, GO TO 66
72 DO=-RCON
   DMM=DO
   GO TO 67
75 READ(5,1004)XVO,XVTO,AO,SPLV
   WRITE(6,1004)XVO,XVTO,AO,SPLV
   IF(SPLV.LT.0.0) XVO=XVO*(TO/XVTO)**AO
   IF(SPLV.GE.0.0) XVO=SPLV*SQR(TO)/(1.0+AO/TO)
   XVTO=TO
   GO TO 68
76 READ(5,1004) CPO,CPTO,CO
   WRITE(6,1004) CPO,CPTO,CO
   CPO=CPO*(TO/CPTO)**CO
   CPTO=TO
   GO TO 69
77 READ(5,1004)HTKO,HTKTO,BO,SPLK
   WRITE(6,1004)HTKO,HTKTO,BO,SPLK
   IF(SPLK.LT.0.0) HTKO=HTKO*(TO/HTKTO)**BO
   IF(SPLK.GE.0.0) HTKO=SPLK*SQR(TO)/(1.0+BO/TO)
   HTKTO=TO
CCOMPUTING THE NON-DIMENSIONALIZING CONVERSION FACTORS
80 UCONV=UO
   XCONV=XVO/(DO*UO)
   PCONV=(DO*UO*UO)/GCON
   DCONV=DO
   XVCONV=XVO
   XMCONV=(XVO*XVO)/(DO*UO)
   WRITE(6,1005) UCONV,XCONV,PCONV,DCONV,XVCONV,XMCONV
1005 FORMAT('UCONV=',G12.5, 12X,'XCONV=',G12.5,12X,'PCONV=',G12.5,
112X,' DCONV='G12.5,12X,' XVCONV='G12.5,11X,' XMCONV='G12.5)
IF(DENS.GT.0.0) GO TO 41
TCONV=(UO*UO)/(GCON*XJCON*CPO)
HCONV=(UO*UO)/(GCON*XJCON)
CPCONV=CPO
XKCONV=3600.0*XVO*CPO
DDD=PCONV/(TCONV*DCONV)
XVTO=XVTO/TCONV
CPTO=CPTO/TCONV
HTKTO=HTKTO/TCONV
TOZ=SQR(TCONV)
SPLV=SPLV*TOZ/XVO
SPLK=SPLK*TOZ/XKCONV
IF(SPLV.GE.0.0) AO=AO/TCONV
IF(SPLK.GE.0.0) BO=BO/TCONV
WRITE(6,1006)TCONV,HCONV,CPCONV,XKCONV
1006 FORMAT(1X,TCONV='G12.5,12X,'HCONV='G12.5,12X,'CPCONV='G12.5,12X,'XKCONV='G12.5)
CSETTING THE Y-GRID SPACING
41 N=NY+1
DELY=DELY/XCONV
IF(RAD) 37,38,39
37 CH=-1.0
CHPF=2.0
RAD=ABS(RAD)
GO TO 39
38 FP=-1.0
RAD=0.1
N=101
GO TO 44
39 DELY=RAD/(XCONV*NY)
44 NP1=N+1
NM1=N-1
NM2=N-2
XRAD=RAD/XCONV
IF(LORT.GE.0.0) GO TO 43
DYM=1.0
DO 42 J=1,N
Y(J)=(J-1)*DELY
DELYP(J)=DELY
DELYM(J)=DELY
DELYT(J)=2.0*DELY
R(J)=XRAD-Y(J)
42 CONTINUE
Y(N+1)=Y(N-1)
R(N+1)=R(N-1)
GO TO 34
43 DELY=DELY1/XCONV
DELYP(1)=DELY
DELYM(1)=DELY
DELYT(1)=2.0*DELY
Y(1)=0.0
47 DO 45 J=2,N
DELYP(J)=DYM*DELYP(J-1)
DELYM(J)=DELYP(J-1)
DELYT(J)=DELYM(J)+DELYP(J)
Y(J)=Y(J-1)+DELYP(J-1)
R(J)=XRAD-Y(J)
45 CONTINUE
R(N+1)=R(N-1)
Y(N+1)=Y(N-1)
IF(FP.LE.0.0) GO TO 34
IF(Y(N).GE.XRAD) GO TO 46
DYM=DYM+0.005
GO TO 47
46 YCH=XRAD/Y(N)
DO 48 J=2,N
48 Y(J)=YCH*Y(J)
Y(N+1)=Y(N-1)
DO 49 J=2,N
DELYP(J)=Y(J+1)-Y(J)
DELYM(J)=Y(J)-Y(J-1)
DELYT(J)=Y(J+1)-Y(J-1)
R(J)=XRAD-Y(J)
49 CONTINUE
DELTY(N)=2.0*DELYM(N)
DELYP(1)=Y(2)
DELYM(1)=DELYP(1)
DELYT(1)=2.0*DELYP(1)
R(1)=Y(N)
WRITE(6,1009) DYM
1009 FORMAT('ODYM=',GL12.5)
34 WRITE(6,1007)
1007 FORMAT('DY(J)')
DO 52 J=1,N
52 POUT(J)=Y(J)*XCONV
WRITE(6,1008)(POUT(J),J=1,N)
1008 FORMAT(10G12.5)
CINITIALIZING
DYMS=DYM*DYM
DYMC=DYM*DYMS
DYM4=1.0+DYM+DYMS
PSB=DYM4/(Y(2)*XCONV*DYMS)
PSC=-DYM4/(Y(2)*XCONV*(1.0+DYM)*DYMC)
PSD=1.0/(Y(2)*XCONV*DYM4*DYMC)
PSA=-PSB-PSC-PSD
VW=VWALL/UCONV
IF(CH) 53,53,54
54 IF(FP) 53,53,85
53 DO 35 J=1,102
35 R(J)=1.0
85 IF(BU.LE.0.0) GO TO 86
U(1)=0.0
UM(1)=0.0
DO 87 J=2,N
U(J)=1.0
UM(J)=1.0
87 CONTINUE
GO TO 61
86 JDEL=N
KJDEL=N
IF(LORT.GE.0) GO TO 62
DO 63 J=1,NP1
   U(J)=4.0*Y(J)/XRAD-2.0*Y(J)*Y(J)/(XRAD*XRAD)
  63 UM(J)=U(J)
GO TO 61
62 CALL FDTVP (U0,DD1,VV1,RAD,Y,UUY,UUJ,UCONV,XCONV,NP1)
   UUY(1)=0.0
   UUJ(1)=0.0
   U(1)=0.0
   UM(1)=0.0
DO 64 J=2,N
   CALL POLFIT (NP1,UUY,UUJ,Y(J),U(J))
  64 UM(J)=U(J)
   U(NP1)=U(NM1)
   UM(NP1)=U(NM1)
61 IF(DENS.GT.0.0) GO TO 99
   IF(BT.LE.0.0) GO TO 88
   DO 89 J=1,NP1
      T(J)=TSTART/TCONV
      TM(J)=T(J)
  89 CONTINUE
   IF(TWALL.LE.0.0) GO TO 90
   T(1)=TWALL/TCONV
   TM(1)=T(1)
GO TO 88
90 IF(TWALL.LT.0.0) GO TO 81
   T(1)=-(PSB*T(2)+PSC*T(3)+PSD*T(4)+QW/(HTKO*TCONV))/PSA
   TM(1)=T(1)
GO TO 88
81 READ(5,1002)(TTW(J),J=1,NT)
   WRITE(6,1002)(TTW(J),J=1,NT)
READ(5,1002)(TX(J),J=1,NT)
   WRITE(6,1002)(TX(J),J=1,NT)
   CALL POLFIT(NT,TX,TTW,OXDIST,T1)
   IF(NNT.LE.0) GO TO 82
   T(1)=T1/TCONV
   TM(1)=T1/TCONV
GO TO 88
82 T(1)=-(PSB*T(2)+PSC*T(3)+PSD*T(4)+T1/(HTK0*TCONV))/PSA
TM(1)=T(1)
88 DO 91 J=1,NPl
   V(J)=VW
   VP(J)=VW
   D(J)=PSTART/(RCON*T(J)*TCONV*DCONV)
   DM(J)=D(J)
   XK(J)=HTKO/XKCONV
   XV(J)=1.0
   C(J)=1.0
91 CONTINUE
   IF(RCON.LE.0.0) GO TO 95
92 IF(HTK.LE.0.0) GO TO 96
93 IF(VISC.LT.0.0) GO TO 97
94 IF(CPCON.LE.0.0) GO TO 98
95 GO TO 101
96 DO 36 J=1,NPl
   D(J)=1.0
36 DM(J)=1.0
   GO TO 92
96 IF(SPLK.GE.0.0) GO TO 56
   DO 55 J=1,NPl
55 XK(J)=HTKO*(T(J)/HTK0)**BO/XKCONV
   GO TO 93
56 DO 57 J=1,NPl
57 XK(J)=SPLK*SQRRT(T(J))/(1.0+BO/T(J))
   GO TO 93
97 IF(SPLV.GE.0.0) GO TO 59
   DO 58 J=1,NPl
58 XV(J)=(T(J)/XVTO)**AO
   GO TO 94
59 DO 78 J=1,NPl
78 XV(J)=SPLV*SQRRT(T(J))/(1.0+AO/T(J))
   GO TO 94
98 DO 84 J=1,NPl
84 C(J)=(T(J)/CPTO)**CO
GO TO 101
99 DO 100 J=1, NP1
DP(J)=1.0
D(J)=1.0
DM(J)=1.0
XV(J)=1.0
EK(J)=0.0
C(J)=1.0
V(J)=VW
VP(J)=VW
100 CONTINUE
101 EXRF=0.33
IF(LORT.GE.0.0) GO TO 103
EXRF=0.5
104 DO 102 J=1, N
XL(J)=0.0
102 CONTINUE
GO TO 110
103 IF(LU.GE.0.0) GO TO 104
DO 105 J=1, N
IF(BXL.LE.0.0) GO TO 106
XL(J)=V DK*Y(J)
IF(XL(J).GE.0.089*Y(JDEL)) GO TO 107
GO TO 105
107 BXL=-1.0
106 XL(J)=0.089*Y(JDEL)
105 CONTINUE
110 P=PSTART/PCONV
PM=P
PSO=P
U(N+1)=U(N-1)
IF(USTART.LE.0.0) GO TO 111
EV(1)=XV(1)
EV(2)=XV(2)+D(2)*XL(2)*XL(2)*ABS(U(3)/DELTY(2))
DO 108 J=3, N
EV(J)=XV(J)
108 CONTINUE
EV(N+1)=EV(N-1)
IF(DENS.GT.0.0) GO TO 115
EK(1)=XK(1)
EK(2)=XK(2)+D(2)*C(2)*XL(2)*XL(2)*ABS(U(3)/DELYT(2))
DO 109 J=3,N
EK(J)=XK(J)
109 CONTINUE
EK(N+1)=EK(N-1)
GO TO 115
111 DO 112 J=2,N
EV(J)=XV(J)+D(J)*C(J)*XL(J)*XL(J)*ABS((U(J+1)-U(J-1))/DELYT(J))
112 CONTINUE
EV(N+1)=EV(N-1)
IF(DENS.GT.0.0) GO TO 115
DO 114 J=2,N
EK(J)=XK(J)+D(J)*C(J)*XL(J)*XL(J)*ABS((U(J+1)-U(J-1))/DELYT(J))
114 CONTINUE
EK(N+1)=EK(N-1)
115 I COUNT=C
SUMPO=0.0
DO 120 J=1,NM2,2
Y A1=DELYP(J)
Y B1=DELYP(J+1)
Y AB=Y A1+Y B1
W 2=Y AB/Y B1*(Y AB/3.0-Y A1/2.0)
W 1=Y AB**2.0/(2.0*Y A1)-W 2*Y AB/Y A1
W O=Y AB-W 1-W 2
SUM1=D(J)*U(J)*R(J)
SUM2=D(J+1)*U(J+1)*R(J+1)
SUM3=D(J+2)*U(J+2)*R(J+2)
120 SUMPO=W O*SUM1+W 1*SUM2+W 2*SUM3+SUMPO
JDEL=2
J TDEL=2
DEL XP=DXF*Y(JDEL)
XDIST=0.0
IPRINT=0
PDROP=0.0
K=1
XDOUT=0.0
TXOUT=0*XOUT/XCONV
TSTOP=0*STOP/XCONV
RED=(2.0*DO*UO*RAD)/XVO
IF(CH.GE.0.0) GO TO 117
XMDOT=2.0*SUMPO*DCONV*UCONV*XCONV
GO TO 118
117 XMDOT=6.28319*SUMPO*XCONV
118 WRITE(6,1030)RED,XMDOT
1030 FORMAT('RED=',G12.5,'OMASS FLOW=',G12.5)
XMDOT1=XMDOT
IF(DENS.GT.0.0) GO TO 116
PR=XVO*CP0*3600.0/HTKO
WRITE(6,1031)PR
1031 FORMAT('OPR=',G12.5)
116 WRITE(6,1016)
1016 FORMAT(/,/'**** BEGINNING OF COMPUTATION LOOP ****/')
CBEGINNING COMPUTATION LOOP
200 ICOUNT=ICOUNT+1
IPI=ICOUNT+1
CCALCULATING DELX
DELXM=DELXP
IF(DXFT.GE.0.0) GO TO 204
201 IF(ICOUNT.GE.NS) GO TO 202
CALL DELX1
DXF=DELXP/Y(JDEL)
GO TO 205
202 IF(DXF.LT.(DXFM+0.02)) GO TO 203
NS=IPI
GO TO 201
203 IF(ICOUNT.EQ.NS) WRITE(6,1017)ICOUNT
1017 FORMAT(/,/'O*** D-F EQUATIONS STARTED AT ICOUNT EQUAL TO',I6)
IF(DXF.GE.0.05) GO TO 206
DXF=1.1*DXF
GO TO 204
206 JDELM=MAXO(JDELM, JTDEL)
    JDELM=2.0*JDELM
    IF(ICCUNT.GE.JDELM OR BU.LE.0.0) DXF=DXF+0.02*DXFM
    IF(DXF.GE.DXFM)DXFT=1.0
204 DELXP=DXF*Y(JDELM)
205 DELXT=DELXM+DELXP
    XDIST=XDIST+DELXP
    OXDIST=XDIST*XCONV
CCALCULATING PRESSURE GRADIENT
    IF(FP.LE.0.0) GO TO 235
    IF(CH.GE.0.0) GO TO 220
    XMDOT1=XMDOT1+2.0*D(1)*VW*DELXP*XCONV
    SUMPO=XMDOT1/(2.0*DCONV*UCONV*XCONV)
    GO TO 221
220 XMDOT1=XMDOT1+D(1)*VW*XRAD*DELXP*5.28319*XCONV
    SUMPO=XMDOT1/(6.28319*XCONV)
    CALL PRES
    GO TO 240
235 PP=PM
    DELPT=0.0
    DELPP=0.0
    IF(JDELM.GE.100) GO TO 600
CCALCULATION OF UP(J)
    CALL UVEL
    JDELM=JDELM+10
    KC=MINO(N, JDELM)
    IF(JDELM.GE.N)KJDEL=N
CCALCULATION OF TP(J)
    IF(DENS.GT.0.0) GO TO 360
    CALL TEMP
    JDELM=JDELM+10
    KCT=MINO(N, JDELM)
    IF(TWALL)330,320,310
    330 TP(1)=ThALL/TCONV
    H(1)=C(1)*TP(1)
GO TO 340
320 CONTINUE
   TP(1) = -(PSB*TP(2)+PSC*TP(3)+PSD*TP(4)+QW/(XK(1)*XKCONV*TCONV))/PSA
   H(1) = C(1)*TP(1)
   GO TO 340
330 CALL POLFIT (NT, TX, TTW, OXDIST, T1)
   IF(NTT.LE.0) GO TO 331
   TP(1) = T1/TCONV
   H(1) = C(1)*TP(1)
   GO TO 340
331 QW = T1
   GO TO 320
CCALCULATION CF DP(J)
340 IF(RCON.GT.0.0) GO TO 350
   DO 341 J = 1, NP1
      DP(J) = 1.0
   GO TO 360
350 DO 352 J = 1, NP1
      DP(J) = PP/(RCON*TP(J)) *DDD
   CONTINUE
CCALCULATION CF VP(J)
360 IF(ICOUNT.GE.NS) GO TO 400
   DO 401 J = 1, N
      DM(J) = D(J)
      UM(J) = U(J)
   CONTINUE
400 DPJP = DP(1)
      DJP = D(1)
      DMJP = DM(1)
      RJP = R(1)
      UPJP = UP(1)
      UMJP = UM(1)
      VP(1) = VW
      KD = KC - 1
   DO 410 J = 1, NM2
      DPJ = DPJP
      DPJP = DP(J+1)
DMJ=DMJP
DMJP=DM(J+1)
RJ=RJP
RJP=R(J+1)
UPJ=UPJP
UPJP=UP(J+1)
UMJ=UMJP
UMJP=UM(J+1)
VV1=((RJP+RJ)*DELYP(J))/(DPJP*RJP*DELXT*4.0)
VV2=VV1*DPJP*UPJP-VV1*DMJP*UMJP+VV1*DPJ*UPJ-VV1*DMJ*UMJ
VP(J+1)=DPJ*VP(J)*RJ/DPJP*RJP-VV2

410 CONTINUE
VP(N)=0.0
CINCREMETING
451 IF(DENS.GT.0.0) GO TO 460
DO 452 J=1,N
TM(J)=T(J)
T(J)=TP(J)
DM(J)=D(J)
D(J)=DP(J)
452 CONTINUE
T(NP1)=TP(NP1)
460 DO 462 J=1,N
UM(J)=U(J)
U(J)=UP(J)
V(J)=VP(J)
462 CONTINUE
U(N+1)=UP(N+1)
PM=P
P=PP
Ccalculating new properties
TW=XV(1)*U(2)*PCONF/DELYM(2)
TWP=XV(1)*PSB*U(2)+PSPC*U(3)+PSD*U(4)*XCONV*PCONF
IF(TW.LE.0.0) GO TO 600
464 CONTINUE
UST=TW*GCONF/(D(1)*DCONV)
UST=SQRT(UST)
DO 463 J=1,N
UPLUS(J)=U(J)*UCNV/UST
YPLUS(J)=Y(J)*D(1)/(XV(1)*UCNV)*UST
463 CONTINUE
470 IF(DENS.GT.0.0) GO TO 490
DTDYW=(T(2)-T(1))*TCNV/(DELYM(2)*XCONV)
DTDYP=(PSA*T(1)+PSB*T(2)+PSC*T(3)+PSD*T(4))*TCNV
DHDYW=(H(2)-H(1))*HCONV/(DELYM(2)*XCONV)
DHDYP=(PSA*H(1)+PSB*H(2)+PSC*H(3)+PSD*H(4))*HCONV
DTTA=T(1)*TCNV-TO
IF(DTTA.EQ.0.0) DTTA=(1.0E-50)
HHA=-XK(1)*DTDYW*XKCONV/DTTA
HDX=HHA*DELXP*XCONV
HIDX=HIDX+HDX
HAVE=HIDX/(XDIST*XCONV)
IF(CPCON.GT.0.0) GO TO 472
DO 471 J=1,NP1
471 C(J)=(T(J)/CPTO)**CO
472 IF(VISC.GT.0.0) GO TO 474
IF(SPLV.GE.0.0) GO TO 477
DO 473 J=1,NP1
473 XV(J)=(T(J)/XVTO)**AO
GO TO 474
474 IF(HTK.GT.0.0) GO TO 480
IF(SPLK.GE.0.0) GO TO 476
DO 475 J=1,NP1
475 XK(J)=HTK0/XKCONV*(T(J)/HTK0)**BO
GO TO 480
476 DO 479 J=1,NP1
479 XK(J)=SPLK*SQR(T(J))/(1.0+BO/T(J))
480 IF(RCON.GT.0.0) GO TO 485
C DWRT(J) AND DWRP(J) FOR NON-IDEAL GAS, GO TO 490
485 DO 486 J=1,N
DWRT(J)=-D(J)/T(J)
486 DWRP(J)=D(J)/P
490 IF(LORT.LT.0.0) GO TO 441
   CALL EFVISC
   GO TO 442
441 DO 443 J=1,NP1
443 EV(J)=XV(J)
   IF (DENS.GT.0.0) GO TO 442
   DO 444 J=1,NP1
444 EK(J)=XK(J)
442 CONTINUE
CFINDING OUTPUT STATIONS
500 IF(ICOUNT.GE.IEND) GO TO 600
   IF(IOUT.LE.0) GO TO 502
   IF((ICOUNT-IOUT).GE.IPRINT) GO TO 605
502 IF(OXOUT). LE. 503, 510, 505
503 IF(XDIST*XCONV). LE. OUT(K)) GO TO 504
   GO TO 200
504 IF(NSTA.LE.K) GO TO 600
   K=K+1
   GO TO 605
505 IF(XDIST.GE.TSTOP) GO TO 600
   IF(XDIST.GE.(XOUT+TXOUT)) GO TO 506
   GO TO 200
506 IF(OINC.LE.0.0) GO TO 605
   TXOUT=OINC*TXOUT
   GO TO 605
510 IF(OREX) 615, 511, 615
511 IF(ORET) 615, 200, 615
COUTPUT
600 PRST=1.0
   GO TO 610
605 PRST=-1.0
610 IPRINT=ICOUNT
   XDOUT=XDIST
615 DN=O(N)
   UN=U(N)
   SUM1=0.0
   SUM2=0.0
SUM3=0.0
SUM4=0.0
SUM5=0.0
SUM6=0.0
DO 620 J=1,NM2,2
YA1=DELYP(J)
YB1=DELYP(J+1)
YAB=YA1+YB1
W2=YAB/YB1*(YAB/3.0-YA1/2.0)
W1=YAB**2.0/(2.0*YA1)-W2*YAB/YA1
WO=YAB-W1-W2
UOUN1=U(J)/UN
UOUN2=U(J+1)/UN
UOUN3=U(J+2)/UN
UMR1=UOUN1*R(J)
UMR2=UOUN2*R(J+1)
UMR3=UOUN3*R(J+2)
DUON1=UOUN1*D(J)/DN
DUON2=UOUN2*D(J+1)/DN
DUON3=UOUN3*D(J+2)/DN
Y1=1-DUCN1
Y2=1-DUCN2
Y3=1-DUCN3
YY1=DUON1*(1-UOUN1)
YY2=DUON2*(1-UOUN2)
YY3=DUON3*(1-UOUN3)
YW1=6.28319*D(J)*U(J)*R(J)
YW2=6.28319*D(J+1)*U(J+1)*R(J+1)
YW3=6.28319*D(J+2)*U(J+2)*R(J+2)
IF (DENS.GT.0.0) GO TO 619
UT1=2.0*R(J)*U(J)*T(J)*D(J)*C(J)
UT2=2.0*R(J+1)*U(J+1)*T(J+1)*D(J+1)*C(J+1)
UT3=2.0*R(J+2)*U(J+2)*T(J+2)*D(J+2)*C(J+2)
SUM6=SUM6+WO*UT1+W1*UT2+W2*UT3
619 CONTINUE
SUM1=SUM1+WO*Y1+W1*Y2+W2*Y3
SUM2=SUM2+WO*YY1+W1*YY2+W2*YY3
SUM3 = SUM3 + W0 * YW1 + W1 * YW2 + W2 * YW3
SUM4 = SUM4 + W0 * UOUN1 + W1 * UOUN2 + W2 * UOUN3
SUM5 = SUM5 + W0 * UMR1 + W1 * UMR2 + W2 * UMR3

620 CONTINUE
DST = SUM1
THETA = SUM2
IF (FP GE 0.0) GO TO 618
TMM = TO
UMean = UO
GO TO 623

618 CONTINUE
IF (CH GE 0.0) GO TO 621
XMDOT = SUM3 / (3.14159 * XCONV)
UMean = SUM4 * UCONV * XCONV / RAD
TMM = ((CPTO * TCONV) ** (CO / (CO + 1.0))) * ((SUM6 * TCONV / (XMDOT * XCONV)) ***
1.0 / (CO + 1.0))
GO TO 623

621 XMDOT = SUM3
UMean = SUM5 * UN * UCONV * 2.0 / (XRAD * XRAD)
TMM = (((CPTO * TCONV) ** (CO / (CO + 1.0))) * ((SUM6 * TCONV * 3.14159
1 / XMDOT) *** (1.0 / (CO + 1.0)))
623 REX = DO * UMEAN * XDIST * XCONV / XVO
RET = DO * UMEAN * THETA * XCONV / XVO
RED = 2.0 * DO * UMEAN * RAD / XVO
IF (PRST GE 0.0) GO TO 627
IF (OREX) 661, 660, 663
660 IF (ORET) 665, 627, 667
661 IF (REX GE OUT(K)) GO TO 662
GO TO 200
662 IF (NSTA LE K) GO TO 624
K = K + 1
GO TO 625
663 IF (REX GE OSTOP) GO TO 624
IF (REX GE (POREX + OREX)) GO TO 664
GO TO 200
664 IF (OINC LE 0.0) GO TO 625
OREX = OINC * OREX
GO TO 625
665 IF(RET.GE.OUT(K)) GO TO 666
GO TO 200
666 IF(NSTA.LE.K) GO TO 624
K = K + 1
GO TO 625
667 IF(RET.GE.OSTOP) GO TO 624
IF(RET.GE.(PORET+ORET)) GO TO 668
GO TO 200
668 IF(OINC.LE.0.0) GO TO 625
ORET = OINC*ORET
GO TO 625
624 PRST = 1.0
GO TO 626
625 PRST = -1.0
626 POREX = REX
PORET = RET
IPRINT = ICOUNT
627 OXMDOT = XMDOT*XCONV
628 CONTINUE
622 F = ODROP*4.0*GCON*RAD/(OXDIST*DO*UMEAN*UMEAN)
CF = TW/(0.5*D(N)*U(N)*U(N)*PCONV)
IF(DENS.LE.0.0.AND.RCON.GT.0.0) DMM = OP/(RCON*TMM)
CFM = TW*GCON/(0.5*DMM*UMEAN*UMEAN)
US2 = TW*GCON/(D(1)*DCONV)
WRITE(6,1901)
WRITE(6,1001)
WRITE(6,1051) RAD,RED,QXMDOT
WRITE(6,1052) QDIST,QXOD,QDELX
WRITE(6,1053) OP,OPDROP,OPWRX
WRITE(6,1054) QDEL,QDIST,QTHETA
WRITE(6,1055) TW,F,CF
WRITE(6,1056) US,JDEL,ICOUNT

1051 FORMAT(' RADIUS=',G12.5,14X,'MASS FLOW=',G12.5)
1052 FORMAT(' X=',G12.5,14X,'X/D=',G12.5,8X,'DEL=',G12.5)
1053 FORMAT(' P=',G12.5,13X,'P0-P=',G12.5,8X,'DEL/DELX=',G12.5)
1054 FORMAT(' DEL=',G12.5,10X,'DELSTAR=',G12.5,12X,'THETA=',G12.5)
1055 FORMAT(' TAU=',G12.5,16X,'F=',G12.5,15X,'CF=',G12.5)
1056 FORMAT(' USTAR=',G12.5,10X,'JDEL=',G17,16X,'ICOUNT=',G7)

ZZZ=CF*SQRT(REX)
WRITE(6,2000) REX,RET,ZZZ

2000 FORMAT(' REX=',G12.5,11X,'RET=',G12.5,14X,'ZZZ=',G12.5)

XB=XVO*QDIST*XCONV/(RAD*RAD*DO*UMEAN)
DPD1=OPDROP/(DO*UMEAN*UMEAN)*3.174
DPD2=2.0*DPD2
WRITE(6,1080) UMEAN,DPD2,XB

1080 FORMAT(' UMEAN=',G12.5,13X,'DPD2=',G12.5,15X,'XB=',G12.5)
WRITE(6,2001) TWP,VWALL,QM0T1

2001 FORMAT(' TWP=',G12.5,15X,'VW=',G12.5,12X,'MDOT1=',G12.5)
WRITE(6,2002) CFM

2002 FORMAT(' CFM=',G12.5)
WRITE(6,1060)

1060 FORMAT('O**U(J)**')
DO 630 J=1,KC
   POUT(J)=U(J)*UCONV
WRITE(6,1061)(POUT(J),J=1,KC)

630 WRITE(6,1061)(POUT(J),J=1,KC)

1061 FORMAT(' **10G13.5')
WRITE(6,1062)

1062 FORMAT('O**V(J)**')
DO 631 J=1,N
   POUT(J)=V(J)*UCONV
WRITE(6,1061)(POUT(J),J=1,N)
IF (LORT.LT.0) GO TO 644
WRITE(6,1063)
1063 FORMAT('O**U+(J)*•')
WRITE(6,1061)(UPLUS(J),J=1,KC)
WRITE(6,1064)
1064 FORMAT('O**Y+(J)*•')
WRITE(6,1061)(YPLUS(J),J=1,KC)
DO 640 J=1,KC
640 POUT(J)=EV(J)*XVCONV
WRITE(6,1065)
1065 FORMAT('O**EFF VISC(J)**')
WRITE(6,1061)(POUT(J),J=1,KC)
DO 642 J=1,KC
642 POUT(J)=XL(J)*XCONV
WRITE(6,1066)
1066 FORMAT('O**L(J)**')
WRITE(6,1061)(POUT(J),J=1,KC)
644 CONTINUE
650 IF(DENS.GT.0.0) GO TO 700

ADDITIONAL OUTPUT FOR HEAT TRANSFER CASES

XVMM=XVO
IF(VISC.LE.0.0.AND.SPLV.LT.0.0)XVMM=((TMM/(XVTO*TCONV))**AO)
1*XVCONV
IF(SPLV.GE.0.0)XVMM=SPLV*SQRT(TMM/TCONV)*XVCONV/(1.0+AO*TCONV/TMM)
XKMM=HTKO
IF(HTK.LE.0.0.AND.SPLK.LT.0.0)XKMM=HTKO*(TMM/(HTKTO*TCONV))**BO
IF(SPLK.GE.0.0)XKMM=SPLK*SQRT(TMM/TCONV)*XCONV/(1.0+BO*TCONV/TMM)
CPMM=CPO
IF(CPCON.LE.0.0)CPMM=((TMM/(CPTO*TCONV))**CO)*CPCON
PRMM=XVMM*CPMM*3600.0/XKMM
REXMM=DMM*UMEAN*OXDIST/XVMM
RETMM=DMM*UMEAN*THETA*XCONV/XVMM
REDMM=2.0*DMM*UMEAN*RAD/XVMM
QWALL=-XK(1)*DTDYP*XKCONV
RECF=PR**EXRF
TAW=TMM+RECF*UMEAN*UMEAN/(2.0*CPMM*GCON*XJCON)
DTTM=T(1)*TCONV-TMM
IF(DTTM.EQ.0.0) DTTM=(1.0E-50)
DTTAW=T(1)*TCONV-TAW
IF(DTTAW.EQ.0.0) DTTAW=(1.0E-50)
HHH=-XK(1)*DTDYP*XKCONV/DTTM
STMTP=HHH/(DMM*UMEAN*CPMM*3600.0)
XNUX=HHH*OXDIST/(XK(N)*XKCONV)
XNUD=HHH*2.0/(XK(N)*XKCONV)
XNUXMM=HHH*OXDIST/XKMM
XNUDMM=HHH*2.0*RAD/XKMM
XNDTAW=XNUDMM*DTTM/DTTAW
XNUDA=HAVE*RAD*2.0/(XK(N)*XKCONV)
XNUXA=HAVE*OXDIST/(XK(N)*XKCONV)
XVN=HAVE*OXDIST/(XK(N)*XKCONV)
XV1=HAVE*RAD/XKCONV
CPN=C(N)*CPCONV
CP1=C(1)*CPCONV
XKN=XK(N)*XKCONV
XK1=XK(N)*XKCONV
STMtw=STMTP*DTDWY/DTDYP
STMHW=STMTP*DHDYW/(DTDYP*CPMM)
STMHP=STMTP*DHCPY/(DTDYP*CPMM)
STAWTP=STMTP*DTTAW
STAWtw=STAWTP*DTDYP/DTTAW
STAWHW=STAWTP*DHDYW/(DTDYP*CPMM)
STAWHP=STAWTP*DHCYP/(DTDYP*CPMM)
WRITE(6,1071) TMM,TAW,DMM
WRITE(6,1072) XVMM,CPMM,XKMM
WRITE(6,1073) XVN,CPN,XKN
WRITE(6,1074) XV1,CP1,XK1
WRITE(6,1075) PRMM,DTTM,DTTAW
WRITE(6,1076) DTDYW,STMtw,STAWtw
WRITE(6,1077) DTDYP,STMHP,STAWHP
WRITE(6,1078) DHDYW,STMHW,STAWHW
WRITE(6,1079) DHCYP,STMHP,STAWHP
WRITE(6,1090) XNUX,XNUD,QWALL
WRITE(6,1091) XNUXMM,XNUDMM,JTDEL
WRITE(6,1092) HAVE,XNUDA,XNUXA
WRITE(6,1093) REXMM,RETMM,REDMM
WRITE(6,1094) XNDTAW
1071 FORMAT(' O TMEAN=',G12.5,F14.1,' TAW=',G12.5,F10.1,' RHOMEAN=',G12.5)
1072 FORMAT(%15X,'MUMEAN=',G12.5,F11.1,' CPMEAN=',G12.5,F12.1,' KMEAN=',G12.5)
1074 FORMAT(' MUEEDGE=',G12.5,F11.1,' CPEDGE=',G12.5,F12.1,' KEDGE=',G12.5)
1075 FORMAT(' MUEWALL=',G12.5,F11.1,' CPWALL=',G12.5,F12.1,' KWALL=',G12.5)
1077 FORMAT(' DTDYW=',G12.5,F12.1,' STMTW=',G12.5,F11.1,' STAWTW=',G12.5)
1078 FORMAT(' DHDYW=',G12.5,F12.1,' STMHW=',G12.5,F11.1,' STAHW=' ,G12.5)
1079 FORMAT(' DHDYP=',G12.5,F12.1,' STMPH=',G12.5,F11.1,' STAWHP=',G12.5)
1083 FORMAT(' PRMEAN=',G12.5,F10.1,' NUX=',G12.5,F14.1,' NUO=',G12.5,F12.1)
WRITE(6,1067)
1067 FORMAT(' O**T(J)**')
DO 652 J=1,KCT
652 POUT(J)=T(J)*TCONV
WRITE(6,1061)(POUT(J),J=1,KCT)
WRITE(6,1068)
1068 FORMAT(' O**I(J)**')
KCO=MAX0(KC,KCT)
DO 654 J=1,KCO
654 POUT(J)=HI(J)*HCONV
WRITE(6,1061)(POUT(J),J=1,KCO)
DO 656 J=1,KCT
656 POUT(J)=DJ(J)*DCONV
WRITE(6,1069)
1069 FORMAT(' O**RHO(J)**')
WRITE(6,1061)(POUT(J),J=1,KCT)
IF(LRT.LT.0.0) GO TO 700
DO 658 J=1,KCO
658 POUT(J)=EK(J)*XKCONV
WRITE(6,1070)
1070 FORMAT(' O**EFF K(J)**')
WRITE(6,1061)(POUT(J),J=1,KCO)
700 IF(PRST.LE.0.0) GO TO 200
   IF (KNC.GT.0) GO TO 5
STOP
END
Subroutine DELX1

Subroutine DELX1 calculates the $X$ step for the ordinary explicit method based on the stability requirements.
SUBROUTINE DELX1
CCALCULATES DELX FOR STANDARD EXPLICIT EQUATIONS USED AS STARTING CMETHOD
COMMON/MVAR/U(102),UM(102),UP(102),V(102),R(102),EV(102),
1EK(102),D(102),C(102),DELYP(102),DELYT(102),COEFP(102),COEFM(102),
2SMUP(102),SMUM(102),SKCP(102),SKCM(102),SPMAX(102),SMMAX(102),
3DELXP,DELXT,DELP,P,F PDROP,PP,PSO,SUMPO,CHPF,ICOUNT,N,NP1,
4NM1,JDEL,KJDEL,NS
SMUP(1)=EV(1)+EV(2)
SKP=EK(1)+EK(2)
CC3=0.0
DO 5 J=2,NM1
SMUM(J)=SMUP(J-1)
SMUP(J)=EV(J)+EV(J+1)
SKM=SKP
SKP=EK(J)+EK(J+1)
SKCM(J)=SKM/C(J)
SKCP(J)=SKP/C(J)
SPMAX(J)=AMAX1(SMUP(J),SKCP(J))
SMMAX(J)=AMAX1(SMUM(J),SKCM(J))
CC1=2.0*D(J)*U(J)*DELYT(J)
COEFP(J)=(R(J)+R(J+1))/(CC1*DELYP(J))
COEFM(J)=(R(J)+R(J-1))/(CC1*DELYP(J-1))
CC2=(ABS(V(J)))/(U(J)*DELYP(J-1)+COEFP(J)*SPMAX(J)+COEFM(J)*
1SMMAX(J)
CC3=AMAX1(CC2,CC3)
5 CONTINUE
DELXP=0.5/CC3
RETURN
END
Subroutine PRES

Subroutine PRES calculates the axial pressure gradients for pipe or parallel plate channel flows.
SUBROUTINE PRES
CCMMON/MVAR/UM(102),UP(102),V(102),R(102),EV(102),
1EK(102),D1(102),C(102),DELYP(102),DELYT(102),COEFP(102),COEFM(102),
SMUP(102),SMUM(102),SKCP(102),SKCM(102),SPMAX(102),SMMAX(102),
3DELPX,DELT,DELP,DELPF,FP,PDROP,PP,PSO,SUMP,CHPF,ICOUNT,N,NP1,
4NM1,JDEL,KJDEL,NS
DIMENSION SUM1(102),SUM2(102)
NM2=N-2
IF(ICOUNT.GE.NS) GO TO 220
CSTANDARD EXPLICIT EQUATIONS
U(1)=U(2)/(1.0+(1.0-U(2)/U(3)))*DELYP(1)/DELYP(2)
IF(U(1).EQ.0.0) U(1)=(1.0E-50)
AXX=0.0
BXX=0.0
CXX=0.0
DXX=0.0
UDl=0.0
CMl=0.0
DO 5 J=l,NM2,2
 YA1=DELYP(J)
 YB1=DELYP(J+1)
 YAB=YA1+YB1
 W2=YAB/YB1*(YAB/3.0-YA1/2.0)
 W1=YAB**2.0/(2.0*YA1)-W2*YAB/YA1
 W0=YAB-W1-W2
 AX1=R(J)/U(J)
 AX2=R(J+1)/U(J+1)
 AX3=R(J+2)/U(J+2)
 IF(J.EQ.1) GO TO 6
 UD1=(U(J)-U(J-1))/(DELYP(J)*U(J))
 CMl=(R(J-1)+R(J))*EV(J-1)+EV(J))/U(J-1)/U(J))/(2.0*DELYP(J-1))
 6 UD2=(U(J+1)-U(J))/(DELYP(J)*U(J+1))
 UD3=(U(J+2)-U(J+1))/(DELYP(J+1)*U(J+2))
 IF(V(J).LT.0.0) UD1=(U(J+1)-U(J))/(DELYP(J)*U(J))
 IF(V(J+1).LT.0.0) UD2=(U(J+2)-U(J+1))/(DELYP(J+2)*U(J+1))
 IF(V(J+2).LT.0.0) UD3=(U(J+3)-U(J+2))/(DELYP(J+2)*U(J+2))
 BXl=CELXP*V(J)*D(J)*R(J)*UD1
BX2 = DELXP*V(J+1)*D(J+1)*R(J+1)*U02  
BX3 = DELXP*V(J+2)*D(J+2)*R(J+2)*U03  
CP1 = (R(J)+R(J+1))*(EV(J)+EV(J+1))*(U(J+1)-U(J))/(2.0*DELTP(J))  
CP2 = (R(J+1)+R(J+2))*(EV(J+1)+EV(J+2))*(U(J+2)-U(J+1))/(2.0*DELTP(J+1))  
CP3 = (R(J+2)+R(J+3))*(EV(J+2)+EV(J+3))*(U(J+3)-U(J+2))/(2.0*DELTP(J+2))  
CX1 = DELXP*(CP1-CM1)/(DELYT(J)*U(J))  
CX2 = DELXP*(CP2-CP1)/(DELYT(J+1)*U(J+1))  
CX3 = DELXP*(CP3-CP2)/(DELYT(J+2)*U(J+2))  
DX1 = C(J)*R(J)*U(J)  
DX2 = D(J+1)*R(J+1)*U(J+1)  
DX3 = D(J+2)*R(J+2)*U(J+2)  
AXX = W0*AX1+W1*AX2+W2*AX3+AXX  
BXX = W0*BX1+W1*BX2+W2*BX3+BXX  
CXX = W0*CXX1+W1*CXX2+W2*CXX3+CXX  
DXX = W0*DXX1+W1*DXX2+W2*DXX3+DXX  
DELPP = (DXX-SUMPO-BXX+CXX)/AXX  
U(J) = 0.0  
GO TO 10

CDUFORT-FRANKEL EQUATIONS

220 SUMP1 = 0.0  
SUMP2 = 0.0  
DEN = (R(1)+R(2))*(EV(1)+EV(2)) + 2.0*R(1)*EV(1)  
SUM1(1) = 2.0*(R(1)+R(2))*(EV(1)+EV(2))*U(2)*D(1)*R(1)/DEN  
SUM2(1) = 8.0*R(1)*R(1)*D(1)*(DELTP(1)**2.0)/(DELTXT*DEN)  
RJP = R(2)  
RJ = R(1)  
UJP = U(2)  
UJ = U(1)  
EVJP = EV(2)  
EVJ = EV(1)  
DELTPJ = DELTP(1)  
DO 221 J = 2, N  
RJM = RJ  
RJ = RJP  
RJP = R(J+1)
UJM=UJ
UJ=UJP
UJP=U(J+1)
EVJM=EVJ
EVJ=EVJP
EVJP=EV(J+1)
DELMJ=DELYPJ
DELYPJ=DELMJ+DELPJ
VJ=V(J)
UMJ=UM(J)
DJ=D(J)

PA=-4.0*DJ*RJ*DELYPJ*DELMJ*DELXT*VJ*( WJP-UJM)
PC=2.0*DELXT*DELMJ*(RJP+RJ)*(EVJP+EVJ)*(UJP-0.5*UMJ)
PD=2.0*DELXT*DELYPJ*(RJ+RJM)*(EVJ+EVJM)*(UJM-0.5*UMJ)
PE=4.0*UMJ*DJ*UJ*RJ*DELYTJ*DELYPJ*DELMJ
PF=4.0*DJ*UJ*RJ*DELYTJ*DELMJ*DELYPJ
PG=DELXT*DELMJ*(RJP+RJ)*(EVJP+EVJ)
PH=DELXT*DELYPJ*(RJ+RJM)*(EVJ+EVJM)
PPP=4.0*RJ*DELYTJ*DELMJ*DELYPJ
SUM1(J)=((PA+PC+PD+PE)*DJ*RJ)/(PF+PG+PH)
SUM2(J)=(PPP*DJ*RJ)/(PF+PG+PH)

221 CONTINUE
DO 50 J=1,NM2,2
YAl=DELYP(J)
YBl=DELYP(J+1)
YAB=YAl+YBl
W2=YAB/YBl*(YAB/3.0-YAl/2.0)
W1=YAB**2.0/(2.0*YAl)-W2*YAB/YAl
WO=YAB-W1-W2
SUMP1=WO*SUM1(J)+W1*SUM1(J+1)+W2*SUM1(J+2)+SUMP1
50 SUMP2=WO*SUM2(J)+W1*SUM2(J+1)+W2*SUM2(J+2)+SUMP2
DELP=(SUMP1-SUMP0)/SUMP2
DELPP=DELXP*DELP/DELXT

10 CONTINUE
PDROP=PDROP+DELPP
PP=PS0+PDROP
Subroutine UVEL

Subroutine UVEL applies the finite difference forms of the momentum equation to calculate the axial velocity components $U_{i+1,j}$ for each $j$ at $i+1$. 
SUBROUTINE UVEL
COMMON/MVAR/U(102),UM(102),UP(102),V(102),R(102),EV(102),
1EK(102),D(102),C(102),DELYP(102),DELYT(102),COEFP(102),COEFM(102),
2SMUP(102),SMUM(102),SKCP(102),SKCM(102),SPMAX(102),SMMAX(102),
3DELXP,DELXT,DELP,FP,PDROP,PP,PSO,SUMPO,CHPF,ICOUNT,N,NPI,
4NM1,JDEL,KJDEL,NS
IF(ICOUNT.GE.NS) GO TO 40
CSTANDARD EXPLICIT EQUATIONS
IF(FP.LE.0.0) GO TO 20
UP(N)=U(N)+(DELXP/(D(N)*U(N)))*(4.00*EV(N)*(U(NM1)-U(N))/(DELYP(N)
1*DELYP(N))-DELP/DELXP)
GO TO 30
20 UP(N)=1.0
30 UDEL=0.9999*UP(N)
UP(1)=0.0
UJ=U(1)
UJP=U(2)
DO 35 J=2,N
IF(JDEL.EQ.N) GO TO 33
IF(UP(J-1).LE.0.99*UP(N)) KJDEL=J
IF(UP(J-1).GE.99*UDEL) GO TO 32
JDEL=J
IF(J.EQ.N) KJDEL=N
33 IF(J.EQ.N) GO TO 35
UJM=UJ
UJ=UJP
UJP=U(J+1)
IF(V(J).LE.0.0)
UP(J)=UJ-DELXP*V(J)*(UJ-UJM)/(DELYP(J-1)*UJ)-DELP/(D(J)*UJ)+
1DELXP*(COEFP(J)*SMUP(J)*(UJP-UJ)-COEFM(J)*SMUM(J)*(UJ-UJM))+
GO TO 35
31 UP(J)=UJ-DELXP*V(J)*(UJP-UJ)/(DELYP(J)*UJ)-DELP/(D(J)*UJ)+
1DELXP*(COEFP(J)*SMUP(J)*(UJP-UJ)-COEFM(J)*SMUM(J)*(UJ-UJM))+
GO TO 35
32 UP(J)=UP(N)
35 CONTINUE
GO TO 250
CDUFORT-FRANKEL EQUATIONS

40 IF (FP .LE. 0.0) GO TO 244

240 DA=D(N)
   DELYMN=DELYP(N)
   UN=U(N)
   EVN=EV(N)
   EVNM=EV(N-1)
   UP(N)=((-1.0*DELYMN*DELYMN*(DELPT))+(1.0*DELT*(EVNM+EVN)*U(N-1))
      1-(0.5*DELT*(EVNM+EVN)*UM(N))
      2+(1.0*DELYMN*DELYMN*UM(N)*DN*UN))/((1.0*DELYMN*DELYMN*DN*UN)
      3+(0.5*DELT*(EVNM+EVN))))
GO TO 245

244 UP(N)=1.00

245 UDEL=0.9999*UP(N)
   UP(1)=0.0
   RJP=R(2)
   RJ=R(1)
   UJP=U(2)
   UJ=U(1)
   EVJP=EV(2)
   EVJ=EV(1)
   DELYPJ=DELYP(1)

251 DO 255 J=2,N
   IF(JJDEL.EQ.N) GO TO 254
   IF(UP(J-1).LE.0.99*UP(N)) KJDEL=J
   IF(UP(J-1).GE.UDEL) GO TO 253
   JDEL=J
   IF(J.EQ.N) KJDEL=N
254 IF(J.EQ.N) GO TO 255
   RJM=RJ
   RJ=RJP
   RJP=R(J+1)
   UJM=UJ
   UJ=UJP
   UJP=U(J+1)
   EVJM=EVJ
   EVJ=EVJP
EVJP=EV(J+1)
DELYMJ=DELYPJ
DELYPJ=DELYPJ(J)
DELYTJ=DELYMJ+DELYPJ
VJ=V(J)
UMJ=UM(J)
D(J)=D(J)

252 UA=-4.0*RJ*DELYPJ*DELYMJ*DELXT*VJ*(UJP-UJM)
UB=-4.0*RJ*DELYTJ*DELYMJ*DELYPJ*DELYT
UC=2.0*DELXT*DELYMJ*(RJP+RJ)*(EVJP+EVJ)*(UJP-0.5*UMJ)
UD=2.0*DELXT*DELYPJ*(RJ+RJM)*(EVJ+EVJM)*(UJM-0.5*UMJ)
UE=4.0*UMJ*DJ*UJ*RJ*DELYTJ*DELYPJ*DELYMJ
UF=4.0*DJ*UJ*RJ*DELYTJ*DELYMJ*DELYPJ
UG=DELXT*DELYMJ*(RJP+RJ)*(EVJP+EVJ)
UH=DELXT*DELYPJ*(RJ+RJM)*(EVJ+EVJM)
UP(J)=(UA+UB+UC+UD+UE)/(UF+UG+UH)
GO TO 255

253 UP(J)=UP(N)
255 CONTINUE
250 CONTINUE
UP(N+1)=UP(N-1)
RETURN
END
Subroutine TEMP

Subroutine TEMP applies the finite difference forms of the energy equation to calculate the temperatures $T_{i+1,j}$ for each $j$ at $i+1$. 
SUBROUTINE TEMP
COMMON/MVAR/U(102), UM(102), UP(102), V(102), R(102), EV(102),
1EK(102), D(102), C(102), DELYP(102), DELYT(102), COEFP(102), COEFM(102),
2SMUP(102), SMUM(102), SKCP(102), SKCM(102), SMAX(102), SMMAX(102),
3DELPX, DELXT, DELPT, DELPP, FP, PDROP, PP, PSO, SUMPO, CHPF, ICOUNT, N, NP1,
4NM1, JDEL, KJDEL, NS
COMMON/TEMP/T(102), TP(102), TM(102), H(102), TSTART, TCONV, JTDEL
IF(ICOUNT .GE. NS) GO TO 80

CSTANDARD EXPLICIT EQUATIONS
IF(FP .LE. 0.0) GO TO 50
TP(N) = T(N) + DELXP/(D(N)*U(N)*C(N))*(U(N)*DELPP/DELXP + 4.0*EK(N)*
1(T(NM1) - T(N))/((DELYP(N)*DELYP(N))
GO TO 60
50 TP(N) = TSTART/TCOV
60 TDEL = T(N)
JTD = 0.9*JTDL
DO 70 J = 2, NM1
IF(J .LE. JTD) GO TO 90
TTEST = ABS(T(J-1) - TOEL)
IF(TTEST .LE. 0.0001*TDEL) GO TO 65
90 CONTINUE
JTDL = J
TP(J) = T(J) - DELXP*V(J)*(T(J) - T(J-1))/((DELYP(J-1)*U(J)) + DELPP/
1(D(J)*C(J)) + EV(J)*DELPX*(1(U(J) - U(J-1))/DELYP(J-1)**2.0)/
2(D(J)*C(J)*U(J)) + DELXP*(COEFP(J)*SKCP(J) + (T(J+1) - T(J)) - COEFM(J)*
3SKCM(J)*(T(J) - T(J-1))
GO TO 67
65 TP(J) = TP(N)
67 H(J) = C(J)*TP(J) + UP(J)*UP(J)/2.0
70 CONTINUE
H(N) = C(N)*TP(N) + UP(N)*UP(N)/2.0
GO TO 371

COUFRONT-FRANKEL EQUATIONS
80 CONTINUE
IF(FP .LE. 0.0) GO TO 312
TP(N) = TM(N) + (4.0*EK(N)*DELT*(T(NM1) - T(N))/((DELYP(N)*DELYP(N)*
1U(N)) + DELPT)/(D(N)*C(N))
GO TO 313
312 TP(N)=TSTART/TCONV
313 TDEL=T(N)
    JTD=0.9*JTD
    RJP=R(2)
    RJ=R(1)
    TJP=T(2)
    TJ=T(1)
    UJP=U(2)
    UJ=U(1)
    EKJP=EK(2)
    EKJ=EK(1)
    DELYPJ=DELYP(1)
    H(1)=0.0
301 DO 305 J=2,N
   IF(J.LE.JTD) GO TO 91
   TTEST=ABS(T(J-1)-TDEL)
   IF(TTEST.LE.0.0001*TDEL) GO TO 302
91 CONTINUE
   JTD=J
   IF(J.EQ.N) GO TO 305
   RJM=RJ
   RJ=RJP
   RJP=R(J+1)
   TJM=TJ
   TJ=TJP
   TJP=T(J+1)
   UJM=UJ
   UJ=UJP
   UJP=U(J+1)
   EKJM=EKJ
   EKJ=EKJP
   EKJP=EK(J+1)
   DELYMJ=DELYPJ
   DELYPJ=DELYP(J)
   DELYTJ=DELYPJ+DELYMJ
   TMJ=TM(J)
DJ=D(J)
CJ=C(J)
VJ=V(J)
EVJ=EV(J)
TA=4.0*TMJ*DJ*CJ*UJ*RJ*DELYTJ*DELYTJ*DELYPJ*DELYMJ
TB=4.0*(DELPT)*UJ*RJ*DELYTJ*DELYTJ*DELYPJ*DELYMJ
TC=4.0*VJ*(TJM-TJP)*DJ*CJ*RJ*DELYTJ*DELYTJ*DELYMJ*DELYPJ
TD=4.0*EVJ*(UJP-UJM)*(UJP-UJM)*RJ*DELYTJ*DELYPJ*DELYMJ
TE=2.0*(RJP+RJ)*(EKJP+EKJ)*(TJP-0.5*TMJ)*DELYMJ*DELYTJ
TF=2.0*(RJ*RJM)*(EKJ+EKJM)*(TJM-0.5*TMJ)*DELYPJ*DELYTJ*DELYTJ
TG=4.0*DJ*CJ*UJ*RJ*DELYTJ*DELYPJ*DELYMJ
TH=DELYTJ*DELYTJ*DELYTJ*(RJP+RJ)*(EKJP+EKJ)
TI=DELYTJ*DELYTJ*DELYPJ*(RJ*RJM)*(EKJ+EKJM)
TP(J)=(TA+TB+TC+TD+TE+TF)/(TG+TH+TI)
GO TO 303
302 TP(J)=TP(N)
303 H(J)=C(J)*TP(J)+UP(J)*UP(J)/2.0
305 CONTINUE
H(N)=C(N)*TP(N)+UP(N)*UP(N)/2.0
371 CONTINUE
TP(NP1)=TP(NM1)
RETURN
END
Subroutine EFVISC

Subroutine EFVISC calculates the effective viscosities and thermal conductivities for turbulent flows. This subroutine may be changed to try different turbulent models.
SUBROUTINE EFVISC
CCOMPUTES EFFECTIVE VISCOSITIES AND THERMAL CONDUCTIVITIES FOR
CTURBULENT FLOWS

COMMON/MVAR/U(102),UM(102),V(102),R(102),EV(102),
1EK(102),D(102),C(102),DELYP(102),DELYT(102),COEFP(102),COEFM(102),
2SMUP(102),SMUM(102),SKCP(102),SKCM(102),SPMAX(102),SMMAX(102),
3DELP,DELT,DELP,Th,PDROPP,PP,PSO,SUMPO,CPH,F,ICOUNT,N,NP1,
4NM1,JDEL,KJDEL,NS
COMMON/MEF/XV(102),XK(102),Y(102),YPLUS(102),POUT(102),YMOD(102),
1XL(102),VDK,DENS,PRT,TW,UST,VPWALL,UCONV
BXL=1.0
DO 468 J=1,N
   IF(BXL.LE.0.0) GO TO 467
   YP26=YPLUS(J)/26.0
   IF(YP26.GE.150.0) YP26=150.0
   XL(J)=VDK*Y(J)*(1.0-EXP(-YP26))
   IF(XL(J).GE.0.089*Y(KJDEL)) GO TO 466
   GO TO 468
466 BXL=-1.0
467 XL(J)=0.089*Y(KJDEL)
468 CONTINUE
   IF(DENS.GT.0.0) GO TO 490
476 DO 477 J=2,N
   EK(J)=XK(J)+D(J)*C(J)*XL(J)*XL(J)*ABS((U(J+1)-U(J-1))/DELYT(J))
   1/PRT
   POUT(J)=EK(J)
477 CONTINUE
   POUT(N+1)=POUT(N-1)
   POUT(1)=XK(1)
   DO 494 J=2,N
   EK(J)=(POUT(J+1)+POUT(J)+POUT(J-1))/3.0
494 CONTINUE
   EK(1)=XK(1)
   EK(N+1)=EK(N-1)
490 DO 492 J=2,N
   EV(J)=(XV(J)+D(J)*XL(J)*XL(J)*ABS((U(J+1)-U(J-1))/DELYT(J))
   POUT(J)=EV(J)
CONTINUE
POUT(N+1)=POUT(N-1)
POUT(1)=XV(1)
DO 493 J=2,N
EV(J)=(POUT(J+1)+POUT(J)+POUT(J-1))/3.0
CONTINUE
EV(1)=XV(1)
EV(N+1)=EV(N-1)
RETURN
END
Subroutine POLFIT

Subroutine POLFIT uses a second degree curve to interpolate between three data points. It is used when wall boundary conditions are input.
SUBROUTINE POLFIT(N,X,Y,AX,AY)
CUSED TO CURVE FIT BOUNDARY CONDITIONS
C INTERPOLATOR SUBROUTINE
C AX=X-INTERCEPT OF DESIRED POINT
C AY=Y-INTERCEPT OF DESIRED POINT
C X=X-INTERCEPTS OF THE DATA POINTS
C Y=Y-INTERCEPTS OF THE DATA POINTS
C N=NUMBER OF DATA POINTS
DIMENSION X(N),Y(N)

IF(AX.LT.X(1)) GO TO 10
DO 14 I=2,N
JJ=I
IF(JJ.EQ.2) JJ=3
XONE=X(I-2)
XTWO=X(I-1)
XTHREE=X(I)
M=I
WRITE(6,15) |
' ***WARNING*** Y IS EXTRAPOLATED'
GO TO 16
13 AY = Y(JJ)
RETURN
12 IF(JJ.EQ.2) I=3
XONE=X(I-2)
XTWO=X(I-1)
XTHREE=X(I)
M=I
GO TO 16
10 XONE=X(1)
XTWO=X(2)
XTHREE=X(3)
M=3
WRITE(6,15)
16 AL1=(AX-XTWO)*(AX-XTHREE)/((XONE-XTWO)*(XONE-XTHREE))
AL2=(AX-XTHREE)*(AX-XONE)/((XTWO-XTHREE)*(XTWO-XONE))
AL3=(AX-XONE)*(AX-XTWO)/((XTHREE-XONE)*(XTHREE-XTWO))
AY = A1 * Y(M-2) + A2 * Y(M-1) + A3 * Y(M)
RETURN
END
Subroutine FDTVP

Subroutine FDTVP solves the fully developed constant property momentum equation for turbulent flows to obtain a velocity profile. It is used for turbulent thermal entry problems.
SUBROUTINE FOTVP (UO, DD1, VV1, RAD, YM, UUY, UUJ, UCONV, XCONV, NPl)

COMMutes FULLY DEVELOPED CONSTANT PROPERTY TURBULENT VELOCITY PROFILE

USED FOR THERMAL ENTRY CASES

DIMENSION X(800), Y(800), V(800), YE(800), YM(102), UUY(102), UUJ(102), YPP(102)

FP(V) = V
F2 = 0.01
RE = 2.0 * DD1 * UO * RAD / VV1

20 F1 = SQRT(F2)
F3 = F2
F2 = (-0.8 + 2.0 * ALOG10(RE * F1)) ** (-2.0)
DIF = ABS(F3 - F2)
IF (DIF .GE. 0.00001) GO TO 20
WRITE(6, 101) F2

101 FORMAT(*, 'FULLY DEVELOPED F= ', G12.5)

UST = SQRT(UO * UO * F2 / 8.0)
DELX = 0.0
Y(1) = 0.0
X(1) = DD1 * UST * RAD / VV1
V(1) = -1.0
XX = 0.01
YE(1) = 0.0
TP = 1.0
R = X(1)
IF (R .LT. 600.0) DELX = -1.0
IF (DELX) 4, 5, 4

4 N = IFIX((XX - X(1)) / DELX)
DELX2 = DELX / 2.0
GO TO 6

5 TP = -1.0
N = 311 + IFIX((R - 600.0) / 20.0)
DELX = -1.0
DELX2 = DELX / 2.0

6 DO 1 I = 1, N
R1 = DELX * FP(X(I), Y(I), V(I), R)
Q1 = DELX * FP(V(I))
Q2 = DELX * FP(V(I) + R1 / 2.0)

1
\[ R_2 = \text{DELX} \cdot F(X(I) + \text{DELX} \cdot Y(I) + Q_1/2.0, V(I) + R_1/2.0, R) \]
\[ R_3 = \text{DELX} \cdot F(X(I) + \text{DELX} \cdot Y(I) + Q_2/2.0, V(I) + R_2/2.0, R) \]
\[ Q_3 = \text{DELX} \cdot F(P(V(I) + R_3)) \]
\[ Q_4 = \text{DELX} \cdot F(P(V(I) + R_3, V(I) + R_4)) \]
\[ V(I+1) = V(I) + (R_1 + 2.0 \cdot R_2 + 2.0 \cdot R_3 + R_4)/6.0 \]
\[ Y(I+1) = Y(I) + (Q_1 + 2.0 \cdot Q_2 + 2.0 \cdot Q_3 + Q_4)/6.0 \]
\[ X(I+1) = X(I) + \text{DELX} \]
\[ YE(I+1) = R - X(I+1) \]
\[ IF(TP)7,1,1 \]
7 
[IF(I.LT.101) GO TO 1] 
[IF(I.GT.301) GO TO 9] 
DELX=-2.0 
DELX2=DELX/2.0 
GO TO 1 
9 
[IF(I.GT.311) GO TO 10] 
DELX=-10.0 
DELX2=DELX/2.0 
GO TO 1 
10 
DELX=-20.0 
DELX2=DELX/2.0 
CONTINUE 
DO 30 J=1, NPl 
30 YPP(J)=YM(J)*XCONV*DDI*UST/VV1 
K=1 
NO1=NPl-1 
NO2=NPl-2 
NO3=NPl-3 
UUY(NO1)=RAD/XCONV 
UUJ(NO1)=Y(N)*UST/UCONV 
DO 40 J=1, N 
IF(K.GE.NO1) GO TO 40 
IF (YPP(K).GT.YE(J)) GO TO 40 
UUY(K)=YE(J)*VVI/(DDI*UST*XCONV) 
UUJ(K)=Y(J)*UST/UCONV 
K=K+1 
40 CONTINUE
UUJ(NP1)=UUJ(NO2)
UUJ(NP1)=UUJ(NO2)
RRR=UUJ(NO1)
UMEAN=0.0
DO 50 J=1,N03,2
YA1=UUJ(J+1)-UUJ(J)
YB1=UUJ(J+2)-UUJ(J+1)
YAB=YA1+YB1
W2=YAB/YB1*(YAB/3.0-YA1/2.0)
W1=YAB**2.0/(2.0*YA1)-W2*YAB/YA1
WO=YAB-W1-W2
U1=UUJ(J)*(RRR-UUJ(J))
U2=UUJ(J+1)*(RRR-UUJ(J+1))
U3=UUJ(J+2)*(RRR-UUJ(J+2))
50 UMEAN=W0*U1+W1*U2+W2*U3+UMEAN
UMEAN=UMEAN*2.0*UCONV/(RRR*RRR)
RE=2.0*CD1*UMEAN*RAD/VV1
WRITE(6,102) UMEAN,RE
102 FORMAT(' UMEAN=',G12.5,/, ' RE=',G12.5)
RETURN
END
Function F

Function F is used in subroutine PDTVP. Different mixing lengths for the initial turbulent velocity profile may be tried by changing function F.
FUNCTION F(X,Y,V,R)
XLK=0.89*R
XPF=(R-X)/26.0
IF(XPF GT 100.0) XPF=100.0
XPF=1.0/EXP(XPF)
XL=0.40*(R-X)*(1.0-XPF)
IF(XL LE XLK) GO TO 2
XL=XLK
DXL=0.0
GO TO 1
2 DXL=0.40*(XPF-1.0)-0.40*(R-X)*XPF/26.0
1 F=((-2.0/R+(2.0*XL*DXL+XL*XL/X)*V*V-V/X)/(1.0-2.0*XL*XL*V)
RETURN
END