Name: Michael Ramey

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: APPLICATION OF COMPUTATIONAL FLUID DYNAMICS TO INDOOR ROOM AIR FLOW

Pages in Study: 116

Candidate for the Degree of Master of Science

Major Field: Mechanical Engineering

Scope of Study: This study uses a FORTRAN computer code which simulates room air flow using a laminar model, a turbulent k-ε model with wall functions and a turbulent low-Reynolds k-ε model. The room dimensions are 4.6 x 2.75 x 2.75 m. A variable grid spacing option has been implemented into the code. Output data has been coded such that a number of post-processing software packages can be utilized for two and three-dimensional visualization. The prediction of convection coefficients and outlet temperatures were investigated and results were compared to experimental data. Global error numbers were obtained and comparisons were made to experimental results and to previous computational results. A two-dimensional error number was created to compare two-dimensional computational results to experimental sections.

Findings and Conclusions: Global error numbers were reduced over prior results using constant and variable grids in some cases. Two-dimensional error numbers were calculated at the z=0.127 m. section which demonstrated that the variable grid implementation helped the computational results match the experimental results somewhat better. The visualization software helped to determine the time of convergence for the computational runs. In most cases, the 100 ACH models converged at 100 seconds of simulation time, the 50 ACH models converged between 225 and 250 seconds, and the 15 ACH models took 450-500 seconds of simulation time for convergence. It was not possible to predict the room surface convection coefficients within acceptable limits, but when experimental convection coefficients were used accurate temperatures were computed.

Adviser's Approval: ________________________________
APPLICATION OF COMPUTATIONAL FLUID DYNAMICS
TO INDOOR ROOM AIRFLOW

By
MICHAEL RAMEY
Bachelor of Science
Oklahoma Christian University of Science and Arts
Oklahoma City, Oklahoma
1991

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1994
APPLICATION OF COMPUTATIONAL FLUID DYNAMICS
TO INDOOR ROOM AIRFLOW

Thesis Approved:

_______________________________________________
Thesis Adviser

_______________________________________________

_______________________________________________
Dean of the Graduate College
ACKNOWLEDGMENTS

I wish to thank my advisor, Dr. Jeff Spitler, for his supervision and guidance. My sincere appreciation also extends to the other members of my committee, Dr. Chambers and Dr. Lilley. Their comments were important to this project, and I am thankful for their help.

I am grateful to Jim Weathers for his insight and advice on this project.

I would like to thank my father, mother, sister and entire family for their continued support and encouragement. My completion of this degree is a success for them as much as it is for me.

Thank you to many other friends, classmates, and professors who have encouraged me along the way and provided helpful advice.

I would especially like to thank my wife, Lisa, for her encouragement at times of difficulty, and for her love and patience over the past few years. Her support has been invaluable, and I will be eternally grateful.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.0</strong> INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Literature Review</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1 CFD Technical Background</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1.1 Laminar Model</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1.2 Turbulent Model</td>
<td>4</td>
</tr>
<tr>
<td>1.2.2 Nielsen</td>
<td>8</td>
</tr>
<tr>
<td>1.2.3 Awbi</td>
<td>8</td>
</tr>
<tr>
<td>1.2.4 Chen</td>
<td>10</td>
</tr>
<tr>
<td>1.2.5 Murakami</td>
<td>12</td>
</tr>
<tr>
<td>1.2.6 Haghighat</td>
<td>14</td>
</tr>
<tr>
<td>1.2.7 Spitler</td>
<td>15</td>
</tr>
<tr>
<td>1.2.8 Weathers</td>
<td>17</td>
</tr>
<tr>
<td>1.2.9 Baker</td>
<td>19</td>
</tr>
<tr>
<td>1.2.10 Sakamoto</td>
<td>21</td>
</tr>
<tr>
<td>1.2.11 Kurabuchi</td>
<td>21</td>
</tr>
<tr>
<td>1.2.12 Li</td>
<td>22</td>
</tr>
<tr>
<td>1.2.13 Alamdari</td>
<td>22</td>
</tr>
<tr>
<td>1.2.14 Setrakian</td>
<td>23</td>
</tr>
<tr>
<td>1.2.15 Yau</td>
<td>23</td>
</tr>
<tr>
<td>1.2.16 Schaelin</td>
<td>24</td>
</tr>
<tr>
<td>1.2.17 Schachenmann</td>
<td>24</td>
</tr>
<tr>
<td>1.2.18 Tsutsumi</td>
<td>24</td>
</tr>
<tr>
<td>1.2.19 Vazquez</td>
<td>25</td>
</tr>
<tr>
<td>1.2.20 Yamazaki</td>
<td>25</td>
</tr>
<tr>
<td>1.2.21 Barozzi</td>
<td>25</td>
</tr>
<tr>
<td>1.2.22 Literature Review Summary &amp; Table</td>
<td>26</td>
</tr>
<tr>
<td>1.3 Objectives</td>
<td>29</td>
</tr>
<tr>
<td><strong>2.0</strong> SOLUTION METHODOLOGY</td>
<td>30</td>
</tr>
<tr>
<td>2.1 Solution Procedure</td>
<td>30</td>
</tr>
<tr>
<td>2.2 Marker-and-Cell Method</td>
<td>33</td>
</tr>
<tr>
<td>2.3 Finite Difference Methodology</td>
<td>36</td>
</tr>
<tr>
<td>2.3.1 Continuity Equation</td>
<td>36</td>
</tr>
<tr>
<td>2.3.2 Momentum Equations</td>
<td>38</td>
</tr>
<tr>
<td>2.3.3 Scalar Equations</td>
<td>42</td>
</tr>
<tr>
<td>2.3.4 Turbulent Kinetic Energy Equation</td>
<td>44</td>
</tr>
<tr>
<td>2.3.5 Turbulent Dissipation Equation</td>
<td>45</td>
</tr>
<tr>
<td>2.4 Boundary Conditions</td>
<td>46</td>
</tr>
<tr>
<td>2.4.1 Wall Boundary Conditions</td>
<td>46</td>
</tr>
<tr>
<td>2.4.1.1 Wall Function Boundary Conditions</td>
<td>47</td>
</tr>
<tr>
<td>2.4.1.2 Low-Reynolds Number Boundary Conditions</td>
<td>48</td>
</tr>
</tbody>
</table>
2.4.1.3 Wall Temperature Boundary Conditions.................50
2.4.2 Inlet Boundary Conditions .......................................................53
2.4.3 Outlet Boundary Conditions.....................................................55
2.4.4 Initial Conditions.....................................................................56
2.5 Turbulence Models.........................................................................56
2.6 Grid Resolution..............................................................................58
2.7 Error Analysis..................................................................................61
2.7.1 Global Error Number ...............................................................61
2.7.2 Slice Error Number ..................................................................62
2.7.3 Global Temperature Error Number ..........................................62
2.7.4 Extrapolation Problem .............................................................63

3.0 RESULTS..............................................................................................65
3.1 Overview of Computational Analysis ...............................................65
3.2 Convergence & Stability ..................................................................65
3.3 Prediction of Temperatures and Convection Coefficients ...............71
3.4 Simulation Results ...........................................................................78
3.4.1 15 ACH Results........................................................................78
3.4.2 50 ACH Results..........................................................................84
3.4.3 100 ACH Results.......................................................................89
3.4.4 Global Velocity Magnitude Error Summary .........................94
3.4.5 2-D Velocity Magnitude Error Summary...........................95

4.0 CONCLUSIONS & RECOMMENDATIONS ..............................................97
4.1 Conclusions.....................................................................................97
4.2 Recommendations.........................................................................98

5.0 REFERENCES.....................................................................................100

APPENDIX .................................................................................................109
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Name of Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>k-ε Wall Boundary Conditions</td>
<td>7</td>
</tr>
<tr>
<td>1.2</td>
<td>Weathers' Reported Global Error Numbers</td>
<td>19</td>
</tr>
<tr>
<td>1.3</td>
<td>Summary of Literature Review</td>
<td>27</td>
</tr>
<tr>
<td>2.1</td>
<td>Convection Coefficient Experimental Values</td>
<td>53</td>
</tr>
<tr>
<td>2.2</td>
<td>Inlet Boundary Condition for U-Normal</td>
<td>54</td>
</tr>
<tr>
<td>2.3</td>
<td>Initial Temperature Values (°C)</td>
<td>56</td>
</tr>
<tr>
<td>2.4</td>
<td>Constants Used in Turbulence Modeling</td>
<td>57</td>
</tr>
<tr>
<td>2.5</td>
<td>Functions Used in Turbulence Modeling</td>
<td>58</td>
</tr>
<tr>
<td>3.1</td>
<td>Convection Coefficient and Temperature Prediction</td>
<td>72</td>
</tr>
<tr>
<td>3.2</td>
<td>Global Error Numbers</td>
<td>94</td>
</tr>
<tr>
<td>3.3</td>
<td>Slice Error Numbers at Z=0.127 m.</td>
<td>95</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Experimental Facility</td>
<td>15</td>
</tr>
<tr>
<td>2.1</td>
<td>Solution Methodology Flow Chart</td>
<td>32</td>
</tr>
<tr>
<td>2.2a</td>
<td>Three-Dimensional View of MAC Velocities</td>
<td>33</td>
</tr>
<tr>
<td>2.2b</td>
<td>Three-Dimensional View of MAC Scalar Quantities</td>
<td>34</td>
</tr>
<tr>
<td>2.3</td>
<td>Two-Dimensional View of MAC Cell Layout</td>
<td>35</td>
</tr>
<tr>
<td>2.4</td>
<td>Temperature Designations and Locations</td>
<td>51</td>
</tr>
<tr>
<td>2.5a</td>
<td>Non-Uniform Grid Resolution in the X Direction</td>
<td>59</td>
</tr>
<tr>
<td>2.5b</td>
<td>Non-Uniform Grid Resolution in the Z (and Y) Directions</td>
<td>59</td>
</tr>
<tr>
<td>2.6a</td>
<td>Uniform Grid in X and Y Directions</td>
<td>60</td>
</tr>
<tr>
<td>2.6b</td>
<td>Variable Grid in X and Y Directions</td>
<td>60</td>
</tr>
<tr>
<td>2.7</td>
<td>Software Comparison: Experimental Velocity Magnitudes at 15 ACH</td>
<td>64</td>
</tr>
<tr>
<td>3.1</td>
<td>Transient Low-Reynolds Number Results at 50 ACH</td>
<td>69</td>
</tr>
<tr>
<td>3.2</td>
<td>Transient Low-Reynolds Number Results at 50 ACH</td>
<td>70</td>
</tr>
<tr>
<td>3.3</td>
<td>Transient Temperature Stratification Values</td>
<td>74</td>
</tr>
<tr>
<td>3.4</td>
<td>Transient Global Temperature Error Number</td>
<td>76</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparison of Computational and Experimental Stratification Values</td>
<td>76</td>
</tr>
<tr>
<td>3.6</td>
<td>Comparison of Air Heat Gain Values</td>
<td>77</td>
</tr>
<tr>
<td>3.7</td>
<td>Experimental Velocity Magnitudes at 15 ACH</td>
<td>80</td>
</tr>
<tr>
<td>3.8</td>
<td>Laminar Velocity Magnitudes at 15 ACH</td>
<td>81</td>
</tr>
<tr>
<td>3.9</td>
<td>Wall Function Velocity Magnitudes at 15 ACH</td>
<td>82</td>
</tr>
<tr>
<td>3.10</td>
<td>Low-Reynolds Number Velocity Magnitudes at 15 ACH</td>
<td>83</td>
</tr>
<tr>
<td>3.11</td>
<td>Experimental Velocity Magnitudes at 50 ACH</td>
<td>85</td>
</tr>
<tr>
<td>3.12</td>
<td>Laminar Velocity Magnitudes at 50 ACH</td>
<td>86</td>
</tr>
<tr>
<td>3.13</td>
<td>Wall Function Velocity Magnitudes at 50 ACH</td>
<td>87</td>
</tr>
</tbody>
</table>
3.14 Low-Reynolds Number Velocity Magnitudes at 50 ACH ...............................88
3.15 Experimental Velocity Magnitudes at 100 ACH ..............................................90
3.16 Laminar Velocity Magnitudes at 100 ACH......................................................91
3.17 Wall Function Velocity Magnitudes at 100 ACH.............................................92
3.18 Low-Reynolds Number Velocity Magnitudes at 100 ACH .............................93
NOMENCLATURE

\( A_{in} \)  Area of Inlet
\( A_{out} \)  Area of Outlet
ACH  Air Changes Per Hour
\( C_{1,2,\mu} \)  Constant Coefficients in k-\( \varepsilon \) Model
\( C_p \)  Specific Heat
D  Mass Divergence
\( E_{abs} \)  Absolute Error
\( F_{1,2,F\mu,E} \)  Empirical Functions in k-\( \varepsilon \) Model
\( g \)  Gravity
GEN  Global Error Number
H  Room Height
\( h_c \)  Convection Coefficient
\( I_u^2 \)  Turbulence Intensity of the X-Velocity
J  Jet Momentum Number
k  Turbulent Kinetic Energy
\( k_{in} \)  Turbulent Kinetic Energy at Inlet
P  Pressure
\( q_{c}^* \)  Convective Heat Flux
\( R_y \)  Local Reynolds number
\( R_{T} \)  Turbulent Reynolds number
T  Temperature
\( T_{O, T_{r}} \)  Reference Temperatures
\( T_s \)  Surface Temperature
t  Time
\( U, V, W \)  Velocity Components in X, Y, & Z Directions
$\bar{U}$  Mean Velocity Component
$U_{in}$  Inlet Velocity
$U_{out}$  Outlet Velocity
$U_n$  Normal Velocity
$U_t$  Tangential Velocity
$u'$  Turbulent Velocity Fluctuation

$\alpha$  Thermal Diffusivity
$\bar{\alpha}$  Donor Cell Coefficient
$\beta$  Thermal Expansion Coefficient
$\bar{\beta}$  Pressure Divisor
$\Delta x_i, \Delta y_j, \Delta z_k$  Distance Between Scalar Quantities
$\Delta x_c, \Delta y_c, \Delta z_c$  Width of MAC Cells
$\Delta t$  Time Step
$\delta_{ij}$  Kronecker Delta
$\varepsilon$  Turbulent Energy Dissipation Rate
$\varepsilon_{in}$  Turbulent Energy Dissipation Rate at Inlet
$\Gamma_x, \Gamma_y, \Gamma_z$  Turbulent Functions
$\Pi$  Turbulent Term in $k$ and $\varepsilon$ Equations
$\Theta$  Turbulent Function in $k$ and $\varepsilon$ Equations
$\rho$  Density
$\mu$  Dynamic Viscosity
$\nu$  Kinematic Viscosity
$\nu_t$  Turbulent Viscosity
1.0 INTRODUCTION

1.1 Overview

The simulation of room air flow has the potential to improve thermal comfort, indoor air quality, and the design of heating, ventilating, and air-conditioning systems. Information concerning the thermal condition of a room is helpful to the designer of a heating or cooling system, and can be provided by a computational analysis of air velocity and temperature in a room. Specific thermal information such as convection coefficients and temperatures throughout the model, especially at the outlet, are helpful to experimental researchers and designers. This thesis explores the possibility of predicting room surface convection coefficients and temperatures throughout the domain by using various temperature wall boundary conditions in a three dimensional, transient, turbulent, buoyant, computational fluid dynamics program.

Other goals of this project are to improve upon the previous modeling efforts through utilization of a variable grid model and improve confidence in the results by using transient visualization tools in the post-processing stages of analysis.
1.2 Literature Review

Airflow modeling and its application have progressed significantly in many areas within the last twenty years. Many researchers have contributed to the specific application of room airflow modeling, while a great many more have developed turbulent two equation models for analyzing a specific type of fluid flow such as boundary layer flow, channel flow, etc. This section will cover computational room airflow research and the development of the $k$-$\varepsilon$ model. The literature review is organized by author.

Because of the extensive amount of work which has been published on the CFD modeling of room airflow, a summary table is provided at the end of the literature review section which includes the designation of driving force, process studied, flow regime, dimension, methodology and grid information. Pertinent information will be given in the text, and Table 1.3 has been added to provide a quick summary and comparison of relevant work.

1.2.1 CFD Technical Background

This section presents the general CFD methodologies used in this study, and should help clarify some of the modeling terminology used later. Most of the technical background is also included in Weathers' [1992] thesis. This section focuses on the differences between the three models used in this study which are a laminar model, a turbulent $k$-$\varepsilon$ model with wall functions, and a turbulent $k$-$\varepsilon$ low-Reynolds number model.
1.2.1.1 Laminar Model

If the flow is assumed not to be turbulent, or if one wishes to obtain a general idea of a turbulent flow, then a laminar model can be implemented. Laminar models are the most basic CFD models, and are much simpler than any of the turbulent models. In some cases it makes sense to use a laminar model to save computational effort and gain basic information even though the flow is believed to be turbulent.

The main equations which need to be solved in a CFD model are conservation of mass and momentum. The conservation of mass, or continuity, equation is:

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (1.1)$$

The laminar, constant density, momentum equation based on the conservation of momentum in Eulerian form is:

$$\frac{\partial U_i}{\partial t} + U_i \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \nu \nabla^2 U_i + g_i \quad (1.2)$$

This is the Navier Stokes equation which together with the continuity equation provides a means for calculating the values of the three velocity components.

The conservation of energy equation for laminar flow can be written in Eulerian form as:

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial x_i} \left( U_i T \right) = \alpha \left( \nabla^2 T \right) \quad (1.3)$$
More about the previous equations will be presented later.

1.2.1.2 Turbulent Model

Turbulence modeling is much more complex than laminar modeling. One basic reason for the additional complexity is that when mean values plus turbulent fluctuations are substituted for velocities, Reynolds stresses result which lead directly to the well known closure problem of turbulence modeling. One good in-depth explanation of the closure problem is given by Hinze [1987]. What mathematically occurs is that one can derive an equation which describes the behavior of a velocity correlation, but in doing so one also creates a velocity correlation of a higher order. No matter how many equations one derives there will always be one too many dependent variables for a closed solution to be possible. Thus, the closure problem.

Because of the closure problem, many different types of models have been developed in order to obtain a turbulent solution. Most of the turbulent models in use today are included in one of the following categories of models: zero equation model, one equation model or two equation model. These models get their names based on the number of additional partial differential equations which are solved. Two equation models include the most popular turbulence models in use today and the models used in this study. Rodi [1980] describes the different classes of models and also provides information about the Reynolds stresses which illustrates the closure problem.

One type of two equation model is the k-ε model. k is the turbulent kinetic energy, and ε is the turbulent dissipation rate. The turbulent models used in this study are both versions of the k-ε model. The first type of turbulent k-ε model used is the standard k-ε model with
wall functions which was introduced by Launder and Spalding in 1974. This model will be referred to throughout this thesis as the wall function model. The second turbulent model used in this study is the low-Reynolds number k-ε model developed by Lam and Bremhorst [1981].

The following basic equations are written in Eulerian form and are valid for both the turbulent wall function model and the turbulent low-Reynolds number model.

The turbulent continuity equation can be written as:

\[
\frac{\partial \overline{U}_i}{\partial x_i} + \frac{\partial u_i}{\partial x_i} = 0
\]

(1.4)

where \( \overline{U}_i \) = Mean velocity components \\
\( u_i \) = Velocity component fluctuations

After velocity substitutions, time averaging, and assuming negligible viscous stresses, the turbulent conservation of momentum is:

\[
\frac{\partial \overline{U}_i}{\partial t} + \overline{U}_i \frac{\partial \overline{U}_j}{\partial x_i} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_i} \left[ \nu_t \left( \frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \right] + g_i
\]

(1.5)

where \( \nu_t \) is the turbulent viscosity which is defined by:

\[
\nu_t = C_\mu F_\mu \frac{k^2}{\varepsilon}
\]

(1.6)
\[ C_\mu = \text{constant coefficient and is usually considered to be 0.09} \]
\[ F_\mu = 1.0 \text{ (except when using the low-Reynolds model)} \]
\[ \varepsilon = \text{turbulent energy dissipation.} \]
\[ \delta_{ij} = \text{Kronecker delta defined by: } \delta_{ij} = 1 \text{ for } i = j \]
\[ \delta_{ij} = 0 \text{ for } i \neq j \]
\[ k = \text{turbulent kinetic energy, defined as: } k = \frac{1}{2} u_i u_i \]

The turbulent kinetic energy equation is:

\[
\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j} (U_j k) = \frac{\partial}{\partial x_j} \left[ \frac{\partial k}{\partial x_j} \right] + \nu \frac{\partial^2 U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \varepsilon \quad (1.7)
\]

The turbulent dissipation equation is:

\[
\frac{\partial \varepsilon}{\partial t} + \frac{\partial}{\partial x_j} (U_j \varepsilon) = \frac{\partial}{\partial x_j} \left[ \frac{\partial \varepsilon}{\partial x_j} \right] + \nu \frac{\partial^2 C_i}{\partial x_j} \frac{\partial \varepsilon}{\partial x_j} + \frac{\partial U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} \frac{\partial U_j}{\partial x_i} C_2 F_2 \frac{\varepsilon^2}{k} + E \quad (1.8)
\]

The basic differences between the two turbulent models are how the functions are applied and the wall boundary conditions used. The turbulent functions, designated as \( F_\mu, F_1, \) and \( F_2, \) are all equal to 1.0 when the wall function model is used. When using the wall function model or the low-Reynolds number model, \( E = 0.0. \) The empirical functions for the low-Reynolds number model are:

\[ F_\mu = e^{-(A_\mu R_e)} \]
\[ F_1 = 1 + 0.05 \mu \]
\[ F_2 = 1 - e^{-(A_\mu R_e)} \]
where \( A_\mu = \text{A constant (0.0165)} \)
\( A_t = \text{A constant (20.5)} \)

\[ R_y = \frac{\sqrt{ky_n}}{\nu} = \text{Local Reynolds number} \]
\( y_n = \text{the normal distance to the wall (m).} \)
\[ R_T = \frac{k^2}{\nu \varepsilon} = \text{Turbulent Reynolds number} \]

The differences in wall boundary conditions used in the two turbulent k-\( \varepsilon \) models are outlined in Table 1.1.

**TABLE 1.1**

**k-\( \varepsilon \) WALL BOUNDARY CONDITIONS**

<table>
<thead>
<tr>
<th>Tangential Wall Velocity:</th>
<th></th>
<th>Low-Reynolds Number Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{\partial U}{\partial x_{n_{\text{wall}}}} = \frac{m U_t}{y_n} )</td>
<td>Slide or No-Slip</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dissipation at the Wall:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon = \frac{(C_{0.5} \mu)^{1.5}}{\kappa y_n} )</td>
<td>( \varepsilon = \frac{2 \nu k}{y_n^2} )</td>
<td></td>
</tr>
</tbody>
</table>
1.2.2 Nielsen

One of the first researchers to use 2-D turbulence modeling to predict air movement and heat transfer in buildings was Nielsen. By 1974, Nielsen had used k-ε modeling to simulate air movement in a room. Nielsen et al. [1978] demonstrated that room airflow predictions were sufficient for design purposes through comparison with experimental values. But, additional research was required for general modeling. At that time, Nielsen's modeling could not handle the flow through rooms with small inlets or relatively long rooms where flow instabilities may be expected.

In 1980, Gosman, Nielsen, Restivo, and Whitelaw used a three-dimensional model to study velocities, pressures, turbulent kinetic energy, and the dissipation rate. They studied different geometries with their k-ε model and found that the turbulent kinetic energy and dissipation length scale varied between 0.1 and 0.3. They stated that an algebraic approximation might replace the differential equations in order to reduce computer usage. Three-dimensional information about the flow patterns of rooms with small inlet regions was provided. They incorporated wall jet assumptions at the inlet to eliminate the need for a denser grid in that region. They also claimed that their value of the maximum velocity was accurate to within 5 percent.

1.2.3 Awbi

Significant numerical research in the areas of wall jets and room ventilation systems has been performed by Awbi. The k-ε turbulence model was used to solve 2-dimensional ventilation problems [Awbi, 1989a].
To evaluate ceiling diffusers, velocity profiles and temperature distributions from the 2-D program were compared to measured values from an experimental room which was designed especially for evaluating the performance of ceiling diffusers. Good agreement between experimental and numerical results existed except for near the floor. With the 3-D program, results were obtained for experiments with obstacles interrupting a wall jet. Awbi found that the obstacles enhance jet diffusion. In some instances the jet reattaches, but as the height of the obstacle was increased, so was the chance for complete separation.

The turbulent kinetic energy and dissipation at the inlet are calculated by:

\[
k_{in} = \frac{3}{2} I_{u}^{2} U_{jet}^{2}
\]

\[
\varepsilon = \frac{k_{in}^{1.5}}{\lambda H}
\]

where \( I_{u}^{2} \) = turbulence intensity of the x-velocity

\( \lambda \) = a constant (\( \approx 0.005 \))

\( H \) = room height or \( \sqrt{\text{inlet area}} \) (m)

Also, for buoyancy-dominated flows, improved wall functions are needed [Awbi, 1989a].

CFD techniques were shown to be a useful tool with practical applications by Awolesi, Awbi, Seymour and Hiley [1991]. They studied ventilation and contamination problems in an industrial workshop, conducted measurements and performed computational analysis on the air movement. Further analysis is necessary to provide an evaluation of Awolesi’s solution to the ventilation problem.
In addition to industrial workshops, numerical simulations were performed on the air movement in offices, classrooms and clean rooms [Awbi, 1991]. Awbi concentrated on human comfort.

In 1989 Awbi noted that more work was needed before CFD programs could be used as design tools. Awbi also stated in 1991 that considerable progress was still needed to improve computational modeling in terms of grid coverage, stability algorithms and boundary conditions.

1.2.4 Chen

In 1990 Chen applied the k-ε model developed by Lam and Bremhorst to cavity flow. Most of his work dealt with the prediction of velocity and temperature distributions in an enclosure with natural convection flow. It was concluded that when one wishes to calculate dynamic indoor air movement near a wall, the low Reynolds number k-ε model will produce a more accurate velocity profile. He also found that more realistic convective heat transfer results occur from the walls to the room when one uses the low Reynolds number model.

The Boussinesq approximation could be used because the temperature difference in the room was relatively small. The Boussinesq approximation for buoyancy was implemented by adding the following term to the momentum equations.

\[
\beta (T - T_0) g_i
\]  

(1.14)
And, source terms were also added to the turbulent kinetic energy equation,

\[ S_k = \frac{\beta_v \partial (T-T_0)}{\sigma_k \partial x_i} g_i \]  \hspace{1cm} (1.15)

where \( h \) is enthalpy, and the dissipation source term is as follows.

\[ S_\varepsilon = \frac{1.44 \varepsilon}{k} S_k \]  \hspace{1cm} (1.16)

Chen validated the low Reynolds number turbulence model through comparison with experimental cavity flow data and showed that the high Reynolds number model with wall functions did a poor job of predicting the near-wall velocity profile.

Because the distance from the wall to the first grid point in most room simulations is greater than the thickness of the boundary layer, Chen claimed that the high Reynolds number \( k-\varepsilon \) model is not suitable for the room airflow application for heat transfer analysis. This is because the convective heat transfer coefficients from the high Reynolds number model are inaccurate [Chen, et al., 1990b]. The high Reynolds number model uses experimental approximations and does not determine the wall heat transfer in the inner layer of the boundary layer which makes it impossible to obtain the heat loss and heat gain through the boundaries computationally.

In addition to computational analysis of cavity flows, Chen and van der Kooi have performed combined energy analysis, indoor airflow, and air quality research. The ACCURACY program was developed to aid in this type of combined research. Chen and van der Kooi have used ACCURACY to analyze displacement ventilation systems [1990a].
Chen and Jiang [1992a] state that with simple thermal and flow boundary conditions, the standard k-ε model may correctly predict room airflow. They also point out that most standard k-ε models use semi-empirical wall function formulas when predicting the heat transfer on a solid surface. But for room airflow, the wall function method is not suitable. The low-Reynolds number k-ε models are also difficult to use properly because too many grid points are required in the near-wall region [Chen and Jiang, 1992a].

Thermal comfort and indoor air quality have been analyzed with a low Reynolds number k-ε model while varying the location and type of diffuser [Chen, Moser, Suter, 1992c]. Their study has shown that both the diffuser type and location have an impact on thermal comfort and indoor air quality. Therefore, if computational airflow simulations become more accurate and dependable, then air quality and thermal comfort can be better analyzed and improved.

1.2.5 Murakami

Many aspects of computational room airflow have been investigated by Murakami. Three dimensional simulations of ventilated rooms were performed [Murakami, et al., 1987] with "fairly good agreement" between computational and empirical results. According to Murakami, one of the most appropriate uses for the k-ε model is modeling airflow in clean rooms where the number of air changes per hour are very high and turbulent flow will most certainly occur. Murakami, et al. used the wall function method in their analysis. Dimensionless physical quantities were obtained which corresponded to experimental values "quite well". Murakami concluded that the k-ε model can appropriately be used to model airflow in clean rooms. It was also pointed out that the computational results have
been compared to empirical mean velocity data. Their turbulence model was only reliable for turbulent flows, and should not be applied out of context.

Murakami's simulation of clean rooms continued. In 1989, Murakami, Kato, and Suyama released results of research which dealt with turbulent diffusion fields and how they are affected by variations in the inlet and outlet. The geometry of the inlet openings and the location, size and number of outlets were varied. Techniques for modeling the boundary conditions at the wall from the log-law and the power law were recognized, outlined, and implemented. Three types of log-law techniques and a power law type boundary condition were studied and produced similar results. Therefore the power law type of boundary condition is preferred because of simplicity [Murakami et al., 1989c]. More about the power law boundary condition is given in section 2.4.1.1 of this report.

Murakami and Kato used the k-ε model to predict three-dimensional turbulent recirculating flows in ventilated rooms. In addressing problems of the k-ε model, Murakami and Kato suggested more complex turbulence modeling to handle the effect of buoyancy and boundary fitted coordinates to simulate complicated room geometries.

Murakami, Kato and Ishida have published details on how to simulate complex geometries with the k-ε model [1989d]. Their method of using generalized curvilinear coordinates is necessary for complex geometries because a normal, rectangular Cartesian grid does not sufficiently match the boundaries. Their grid system fits a curve to the boundary and alters the distance between nodes. The grid can then be transformed from generalized curvilinear coordinates to a rectangular grid.
1.2.6 Haghighat

Room configurations with interior partitions present a different set of problems for the field of CFD. Haghighat, Jiang, and Wang [1989] have explored the airflow patterns, natural convection, and contaminant dispersion of two-zone enclosures.

An open doorway in an interior partition was varied in height, size and location, and its effect on airflow and temperature were studied. Results have shown that the convective heat transfer rate is not sensitive to partition location, while room airflow patterns are sensitive to both door height and location [Haghighat, et al., 1989].

In additional experiments, the inlet and outlet locations along with door positions were varied. The most prominent results from these experiments show that the door location controls the direction and magnitude of air movement in the downstream zone.

Haghighat et al. mainly used the SIMPLE and PHOENICS codes to solve the finite difference equations and predict two and three-dimensional heat and mass transfer in building airflow. They note that at the wall, the turbulent viscosity no longer dominates over the viscous effects. Much like Murakami's approach, wall functions which use an equation for the momentum flux due to shear stress and an equation for the heat flux at solid surfaces were utilized in Haghighat's analysis. A grid sensitivity analysis revealed the most practical density of nodes to use. They used a 10x10x16 system for natural convection analysis of an enclosure with dimensions of 10m x 4m x 3m. But for forced convection, a smaller grid is recommended because of the more complicated boundary conditions [Haghighat, et al., 1992].
1.2.7 Spitler

Air movement and convective heat transfer in buildings were researched by Spitler [1990]. A full scale experimental facility, 2.74 x 2.74 x 4.57m, with the ability to change the inlet location, hold the room surfaces at a constant temperature and vary the flow rate between 2 and 100 air changes per hour was used to obtain experimental room airflow measurements. The experimental results from this facility were used to validate the CFD models of Williams, Baker, and Kelso [1994], Weathers [1993], and has also been used for comparison to this project's results.

The facility allowed for two inlet configurations. The first configuration was a sidewall inlet with dimensions of 0.4 m. x 0.9 m. A radial ceiling diffuser 0.4 m. x 0.4 m. was the other option. The outlet of the facility was 0.4 m. x 0.9 m. and was located in the lower corner of a sidewall. (see Figure 1.1).

Figure 1.1 Experimental Facility
Sixteen omnidirectional air velocity transducers and type T thermocouples were attached to a trolley system on rails which enabled the facility user to obtain velocity and temperature measurements at desired locations. For each experiment, values were determined approximately 1 ft. apart throughout the entire volume of the room.

In order to control the surfaces of the facility isothermally, 53 individually controlled panels covered the interior room surfaces. Each panel could be operated in a low or a high power mode and was heated with nickel chromium resistance wire. This technique allows the heat flux of the panels to be determined, which lead to an experimentally determined film coefficient from the following equation [Spitler, et al., 1991a, 1991b].

\[
h = \frac{q''}{(T_s - T_r)} \quad (1.17)
\]

where \( q'' \) = convective heat flux (W/m²) \\
\( T_s \) = surface temperature (°C) \\
\( T_r \) = reference temperature (°C)

The room outlet temperature, bulk air temperature, air temperature adjacent to a surface, and local air temperature as a function of height were all investigated for use as possible reference temperatures, but the room outlet temperature was used [Spitler, et al., 1991b].
1.2.8 Weathers

A comparative study of room airflow, which provides the basis on which the present research is being conducted, was performed by Weathers [1992]. This project uses Spitler's experimental data for CFD verification just as Weathers' research did. Weathers analyzed a laminar model, $k$-$\varepsilon$ model with wall functions, and the Lam-Bremhorst version of low-Reynolds $k$-$\varepsilon$ model. For each of these three models, he performed simulations at four different airflow rates, 15, 30, 50, and 100 air changes per hour (ACH).

In order to match Spitler's experimental data better, Weathers used measured inlet velocity profiles in his computational models. More information about the inlet profiles used in Weathers' study and in the present study is available in section 2.4.2 and in Table 2.2.

Weathers investigated grid independence with five different uniform grids which ranged from "very course", 10 x 6 x 6, to "ultra fine", 46 x 30 x 30. He concluded that a "very fine", 40 x 24 x 24 grid produced sufficient accuracy.

The wall boundary conditions were varied and the free slip wall boundary condition was found to match the experimental data better than the no slip condition. Weathers' free slip boundary condition recommendation makes sense because the uniform grid used in the computational analysis, 40 x 24 x 24, does not include grid points close enough to the walls. The uniform grid used by Weathers had 24 cells in the z-direction within the 2.75 (m) distance which made each full cell 11.46 cm. wide. Because the Marker-and-Cell method [Harlow and Welch, 1965] was used, the tangential velocity vectors in the cells closest to the wall occur half a cell away from the wall at the cell face. For diagrams and a more detailed explanation of the Marker-and Cell method, see section 2.2. Weathers' tangential wall velocities were 5.73 cm. away from the wall.
In order for a no slip condition to be necessary, the grid should be small enough so that the grid points closest to the wall are included in the boundary layer. If one is interested in boundary layer information, then one should use as small a grid as possible, because as the distance between the first node and the wall decreases, the accuracy of the computational model should increase. If the grid chosen does not contain nodes inside the boundary layer, then boundary layer and slip information is unattainable and the free slip boundary condition is the logical and appropriate choice.

A problem in a majority of prior computational room airflow experiments is that the comparison to experimental results is rather vague and subjective. In order to quantitatively evaluate his computational results, Weathers defined a global error number as:

\[
GEN = \frac{\bar{E}_{\text{abs}}}{V_{\text{max}}}
\]  

where \( \bar{E}_{\text{abs}} \) = average absolute error \( \left( \frac{1}{n} \sum_{i=1}^{n} E_{\text{abs,i}} \right) \)

\( V_{\text{max}} \) = maximum velocity magnitude from experimental data.

\( E_{\text{abs}} = |\text{numerical velocity magnitude} - \text{experimental velocity magnitude}| \)

The global error number compares the computational results to the entire experimental domain. Table 1.2 is presented below which summarizes Weathers' reported global error numbers for each of the 12 computational cases he performed.

**TABLE 1.2**

**WEATHERS' REPORTED GLOBAL ERROR NUMBERS**
Based on the GEN analysis, the low-Reynolds number k-ε model was determined to match the experimental data closer than the other computational models.

Weathers provides information on the low-Reynolds number k-ε models, constants and functions which are used in the turbulent analysis by Launder-Sharma, Chien, and Lam-Bremhorst. Many researchers have their own opinion about what turbulent constants and functions should be used. This accounts for many of the modifications which have been made to the basic k-ε model.

1.2.9 Baker

Code validation is a legitimate concern when dealing with nearly any type of computational analysis. The CFD methodology for room airflow is no different. Baker and Kelso [1990] have attempted to define code validation issues.

Kelso, Wilkening, Schaub, and Baker [1992] have modeled commercial, kitchen exhaust hoods using finite element analysis because experimental analysis would be difficult and
costly. The concluded that their 2-D model must be extended to three dimensions and more detail in filter modeling and hood configurations should be implemented.

Baker, Williams and Kelso published a three-part article in 1994 which outline CFD methods, introduce a finite element method and use Spitler's experimental indoor room airflow data to compare to their CFD models. Therefore, some of the geometries investigated are identical to the geometry in the present study. Part I discusses CFD techniques, basic equations, and issues of modeling. Finite-volume and finite-element semi-discrete spatial constructions were examined.

Part II details the continuity constraint finite element method CFD algorithm. Computational results for benchmark problems were presented for validation. Two-dimensional and three-dimensional natural convection cavities and step-wall diffusers were simulated using the continuity constraint method (CCM), which is a pressure relaxation method.

In the third article of the series, Spitler's full-scale experimental room data is utilized for comparison at 15 and 30 air changes per hour (ACH). The non uniform 3-D CFD grids were generated with the INGRID mesh generator. The levels of turbulence were varied and the 3-D flow effects were studied. Williams, Baker and Kelso conclude that Spitler's 15 ACH data appears to be a low-turbulence-level room air experiment. The 30 ACH CFD velocity magnitude results did not match the experimental results as well as the 15 ACH velocity magnitude results. They also concluded that the computed flows were determined to be strongly three-dimensional and complex.

1.2.10 Sakamoto
In order to test the practicality of computationally predicting 3-D turbulent flow in a ventilated room, Sakamoto and Matsuo [1980] experimented with the Deardorff’s model of turbulence [Deardorff, 1970] and the k-ε model. The room was a 2 x 2 x 2m cube with a square ceiling inlet and a square exhaust outlet at one of the walls and was modeled with an 18 x 18 x 18 grid. The results from the two models were nearly identical, and compared well to experimental mean velocity data which was taken with an ultrasonic anemometer with a 10 cm. probe span. Because Deardorff’s model requires more computer usage, Sakamoto recommended the k-ε model.

1.2.11 Kurabuchi

Kurabuchi has performed indoor air movement and heat transfer simulation using the k-ε model and the large eddy simulation (LES) method while utilizing the Marker-and-Cell grid method [Harlow and Welch, 1965]. Large eddy simulation advantages are noted, but computational complexities have limited the use of the LES method. Experiments were performed for a laboratory with two open windows on one wall, an open doorway on the opposing wall, and a mechanical ventilation device, or fume hood on the sidewall [Kurabuchi 1987].
1.2.12 Li

A 3-D numerical code has been developed by Li which uses a k-\( \varepsilon \) model with the Boussinesq approximation and a two-band radiation model. This finite difference code simulates turbulent air flows in single and multi-rooms and solves the transient-state conservation equation for mass, momentum, and thermal energy. Non-uniform grids and local refinements are possible with the code. The effects of wall insulation and radiation for partially divided rectangular enclosures were studied [Li, et al., 1991a].

Experiments involving forced convection, natural convection and radiation were performed to evaluate the code. Refined grids were found to improve the accuracy of velocities and temperature predictions. Fine grids and a proper distribution of grid points are both needed to achieve accurate computational results. Like Chen [Chen, et al., 1990b], Li also concludes that the high-Reynolds number model with wall functions is not practical for indoor room airflow because it does not determine the wall heat transfer in the inner layer of the boundary layer. Most HVAC engineers are interested in obtaining the heat loss and heat gain through the boundaries [Li, et al., 1991b].

1.2.13 Alamdari

The need for computational fluid analysis is outlined and critiqued by Alamdari. Alamdari found that CFD models were capable of simulating complicated room airflow and heat transfer problems, but require a significant amount of computer time and are difficult to use [Alamdari 1991a]. Airflow and thermal analysis of the headquarters for Hoare, Lea and Partners, which are naturally-ventilated two-story buildings, helped the owners make heating and cooling decisions [Alamdari, et al., 1991b].
1.2.14 Setrakian

Because the range of applications for computational fluid programs varies greatly, a customized CFD room airflow program has important advantages. Setrakian and Morgan [1991a] describe these advantages while presenting airflow results from their "ARIA" program for a room with a single inlet, dual outlets, and interior obstacles.

ESP, a building energy simulation code, provides boundary condition information for the ARIA airflow simulation program. This approach was taken to increase the level of accuracy and confidence in the boundary conditions input to the CFD analysis. Further developments suggested by Setrakian [1991b] include the merging of the two programs so that the CFD program could automatically receive thermal information at every time step.

1.2.15 Yau

Computational fluid dynamics has also been applied to large spaces in buildings. Yau and Whittle [1991] modeled summer and winter operation of an airport terminal building to improve thermal comfort and air quality. They implemented curvilinear coordinates in order to model the curved ceiling of the airport terminal building. A scaled down model of the building was used to obtain experimental results. Smoke movements in the model helped to establish flow patterns at different temperatures, flow rates, and angles of trajectory relative to the roof. This smoke testing was then used to compare with the computational results Yau and Whittle concluded that CFD is a valuable tool in the design process.
1.2.16 Schaelin

Schaelin, van der Maas and Moser [1992] have simulated the coupling of indoor airflow to outdoor flow through large openings. The effect of wind on their configuration was studied in two and three dimensions, and their numerical data agreed well with analytical models which are well supported by experiments.

1.2.17 Schachenmann

In order to check room air currents, experimental measurements and numerical calculations were made for a 6.75 x 4.50 x 2.97 m room with mobile interior partitions. Instead of defining a fine grid at the inlet, Schachenmann et al. used an "equivalent boundary surfaces" technique. The radiation heat exchange is being added to the CFD program used for the [Schachenmann, et al., 1990] results.

1.2.18 Tsutsumi

3-D k-ε and large eddy simulation computational methods were used to obtain data for comparison to naturally ventilated room airflow. The experimental model was designed to measure air speed caused by natural cross-ventilation through large openings. Because the experimental measurement of wind through a building is rare, Tsutsumi's research is relatively unique. The k-ε model results were found to match the experimental values closer than the LES results [Tsutsumi, et al., 1988].
1.2.19 Vazquez

The effect of air inlet location on the ventilation patterns inside an auditorium was studied by Vazquez, Samano, and Yianneskis [1991]. The flow patterns and temperature fields were greatly impacted by the inlet location. An optimum inlet location, based on thermal comfort, was obtained for the auditorium [Vazquez, et al., 1991].

1.2.20 Yamazaki

According to Yamazaki, Komatsu and Otsubo, when modeling air conditioned flow into a room, the buoyant force has an important role in determining air temperature and flow distribution. Flow patterns, semi-open rooms, and large interior spaces can be modeled with a computational program and the results can be helpful in the design of an air conditioning system [Yamazaki 1987].

1.2.21 Barozzi

Many researchers have verified CFD results with experimental results from a scaled down model. Barozzi et al. [1991] used a 1/12th scale model to compare to numerical results for a bioclimatic building based on solar-driven ventilation. This study investigated the thermosyphon principle. The computational field was solved in two dimensions, and favorable agreement existed between CFD and experimental results.

1.2.22 Literature Review Summary & Table
The following summary table, 1.3, has been included to give the reader a comparative description of the numerical methods covered in this review. The table presents the driving force, the process studied, the flow regime, the number of dimensions, the solution methodology, and information about the grid.

### TABLE 1.3
SUMMARY OF LITERATURE REVIEW

<table>
<thead>
<tr>
<th>Author, Year</th>
<th>Driving Force</th>
<th>Process Studied</th>
<th>Flow Regime</th>
<th>Dim.</th>
<th>Methodology</th>
<th>Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alamdari, 1991a</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>3-D</td>
<td>FD</td>
<td>NU, BF</td>
</tr>
<tr>
<td>Alamdari, 1991b</td>
<td>N</td>
<td>B</td>
<td>T</td>
<td>3-D</td>
<td>FD</td>
<td>NU</td>
</tr>
<tr>
<td>Awbi, 1989a</td>
<td>F</td>
<td>B</td>
<td>T</td>
<td>B</td>
<td>FD, WF</td>
<td>NU, 20x20x11</td>
</tr>
<tr>
<td>Awbi, 1991</td>
<td>F</td>
<td>B</td>
<td>T</td>
<td>3-D</td>
<td>FD, WF</td>
<td>NU, 40x24x20</td>
</tr>
<tr>
<td>Awolesi, 1991</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, WF</td>
<td>NU, 45x52x33</td>
</tr>
<tr>
<td>Baker, 1990</td>
<td>B</td>
<td>B</td>
<td>T</td>
<td>B</td>
<td>WF, LR</td>
<td>U, NU</td>
</tr>
<tr>
<td>Author, Year</td>
<td>Driving Force</td>
<td>Process Studied</td>
<td>Flow Regime</td>
<td>Dim.</td>
<td>Methodology</td>
<td>Grid</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------</td>
<td>------</td>
<td>-------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Barozzi, 1991</td>
<td>N</td>
<td>C</td>
<td>L</td>
<td>2-D</td>
<td>FD</td>
<td>NU, 56x96</td>
</tr>
<tr>
<td>Chen, 1990a</td>
<td>N</td>
<td>B</td>
<td>T</td>
<td>3-D</td>
<td>FD, LR</td>
<td>NU</td>
</tr>
<tr>
<td>Chen, 1990b</td>
<td>B</td>
<td>B</td>
<td>T</td>
<td>B</td>
<td>FD, LR</td>
<td>NU, 53x53</td>
</tr>
<tr>
<td>Chen, 1992a</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>2-D</td>
<td>LES, LR</td>
<td>NU</td>
</tr>
<tr>
<td>Chen, 1992c</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, LR</td>
<td>NU</td>
</tr>
<tr>
<td>Gosman, 1980</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε</td>
<td>U, 13x15x8</td>
</tr>
<tr>
<td>Haghghat, 1989</td>
<td>B</td>
<td>C</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε</td>
<td>NU, 16x10x10</td>
</tr>
<tr>
<td>Haghghat, 1992</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>FD, WF</td>
<td>NU, 10x10x16</td>
</tr>
<tr>
<td>Kelso, 1992</td>
<td>F</td>
<td>A</td>
<td>B</td>
<td>2-D</td>
<td>Finite element</td>
<td>NU, 50x50</td>
</tr>
<tr>
<td>Kurabuchi, 1987</td>
<td>N</td>
<td>B</td>
<td>T</td>
<td>B</td>
<td>FD, LES</td>
<td>U, 30x20x15</td>
</tr>
<tr>
<td>Li, 1991a</td>
<td>B</td>
<td>A</td>
<td>T</td>
<td>2-D</td>
<td>FD, k-ε</td>
<td>NU</td>
</tr>
<tr>
<td>Reference</td>
<td>Model</td>
<td>Approach</td>
<td>Technique</td>
<td>Resolution</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>----------</td>
<td>-----------</td>
<td>------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Li, 1991b</td>
<td>B</td>
<td>B</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Murakami, 1987</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Murakami, 1989b</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Murakami, 1989c</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, NU, 46x46x17</td>
<td></td>
</tr>
<tr>
<td>Murakami, 1989d</td>
<td>B</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, CC</td>
<td></td>
</tr>
<tr>
<td>Nielsen, 1978</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>2-D</td>
<td>Stream function, 20x20</td>
<td></td>
</tr>
<tr>
<td>Sakamoto, 1980</td>
<td>F</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, LES, U, 18x18x18</td>
<td></td>
</tr>
<tr>
<td>Schachenmann, 1990</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>3-D</td>
<td>FD, k-ε, U, NU</td>
<td></td>
</tr>
<tr>
<td>Schaelin, 1992</td>
<td>N</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Setrakian, 1991a</td>
<td>F</td>
<td>B</td>
<td>B</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Setrakian, 1991b</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Tsutsumi, 1988</td>
<td>N</td>
<td>A</td>
<td>T</td>
<td>3-D</td>
<td>FD, LES, U, 5x5x5</td>
<td></td>
</tr>
<tr>
<td>Weathers, 1993</td>
<td>F</td>
<td>A</td>
<td>B</td>
<td>3-D</td>
<td>FD, WF, LR, U, 40x24x24</td>
<td></td>
</tr>
<tr>
<td>Yamazaki, 1987</td>
<td>B</td>
<td>B</td>
<td>T</td>
<td>3-D</td>
<td>FD, k-ε, NU</td>
<td></td>
</tr>
<tr>
<td>Yau, 1991</td>
<td>F</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>FD, k-ε, CC</td>
<td></td>
</tr>
</tbody>
</table>
1.3 Objectives

The objectives of this project include the production of a variable grid 3D transient turbulent buoyant code for the simulation of room airflow. Results of constant and variable grid models will be compared to experimental results and to previous computational results.

Improved visualization is necessary in order to add insight about the behavior of the transient simulations. Two and three dimensional visualization tools will be used in conjunction with model results to assure convergence and allow easier error detection.

The feasibility of predicting room surface convection coefficients will also be evaluated in this project. Wall temperature boundary conditions will be varied. Outlet temperatures and predicted convection coefficients will be monitored.
2.0 SOLUTION METHODOLOGY

The research for this project was performed with a finite difference FORTRAN program which uses 3-D Cartesian coordinates and calculates the velocity, temperature, pressure, contaminant distribution, turbulent kinetic energy, and turbulent dissipation. The program provides variable grid spacing and a choice of post processing options to the user. This program is based on the GEneral 3-Dimensional Flow (GE3DF) laminar, constant density code written by D. G. Lilley (1991). The GE3DF code was modified by J. W. Weathers in 1992 to handle turbulent, buoyant flows. Both of these predecessor programs use the Marker-and-Cell method [Harlow and Welch, 1965] which will be discussed in more detail later in this report.

2.1 Solution Procedure

Details of the approach used to solve for the velocities and scalar quantities are included in this section. First the geometry, grid spacing, initial conditions, some boundary specifications, physical properties, and other necessary information are read into the program. The program calculates additional geometric and numerical parameters such as the volume flow rate, the number of air changes per hour (ACH), and the areas of the inlet and outlet. After initial values and boundary conditions are imposed, time advanced velocity components, temperature, mixture fraction, turbulent energy and turbulent dissipation values are calculated for the entire 3-D domain. Boundary conditions are updated after the determination of time advanced values.

Next, the continuity equation is satisfied using the change in pressure as a correction factor for the time advanced velocity components. After continuity is satisfied, residual terms for
the velocities and mixture fraction are calculated based on the amount of change in the
values from the previous time step.

At this point in the solution process, a check for convergence is performed. If the values
have not converged, then another time step must be performed. So, the time advanced
values are shifted to the old memory locations, and time is advanced one step. Then, new
time advanced values can be calculated. This process continues until convergence has
been achieved.

Figure 2.1 is a flowchart which summarizes the solution methodology of the finite
difference code used in this study.
Figure 2.1 Solution Methodology Flow Chart
2.2 Marker-and-Cell Method

The Marker-and-Cell, (MAC), method was developed by Harlow and Welch [1965], and it forms the basis for the finite difference computational fluid dynamics techniques implemented in this research. The MAC method has been used by Lilley [1988] and Weathers [1992], whose work forms the basis of this project, and has also been utilized by other room airflow researchers such as Murakami [1987] and Awbi [1989a].

The Marker-and-Cell method defines all scalar quantities at the center of the cell, but the velocity vectors are defined at the center of the appropriate cell face as shown with the 3-dimensional Figure 2.2 and in 2-dimensions in Figure 2.3.

![Three-Dimensional View of MAC Velocities](image)

Figure 2.2a Three-Dimensional View of MAC Velocities
Figure 2.2b  Three-Dimensional View of MAC Scalar Quantities
In Figure 2.3, the w velocity component is not shown because it exists outside the plane of the page. Notice that the velocities are defined half a cell from the location of the scalar quantities, and that the velocity magnitudes represent the speed of the fluid moving across that particular boundary or cell face.
2.3 Finite Difference Methodology

The basic equations for laminar and turbulent numerical fluid dynamics and the finite difference equations which stem from the basic equations are contained within this section. The equations presented here apply to a constant density room airflow with Cartesian coordinates and variable grid spacing. For constant grid space equations, and additional discussion, see Weathers [1992].

As mentioned in previous sections, the program used to model room airflow for this research uses a transient finite difference model to solve a laminar or turbulent flow field. Buoyancy can be implemented in the constant density solution process by using a coefficient of thermal expansion to model the changes in density due to temperature differences. The turbulent models are solved using the k-ε model. The main equations necessary in this type of turbulence model are continuity, momentum, scalar equations, and of course, the k and ε equations. The discussion which follows states the equation, any changes for turbulence modeling, and the finite difference form of the equation. According to Rodi [1980], the mean flow quantities are of practical interest while specific information about the turbulent fluctuations is less useful. This is also true for the specific application to room airflow, and it should be kept in mind that accurate prediction of mean values are the main concern in this analysis.

2.3.1 Continuity Equation

Conservation of mass for a constant density situation can be written as
\[
\frac{\partial U_i}{\partial x_i} = 0 \quad (2.1)
\]

Whereas a turbulent form of the continuity equation is:

\[
\frac{\partial \bar{U}_i}{\partial x_i} + \frac{\partial u_i}{\partial x_i} = 0 \quad (2.2)
\]

The first term of equation (2.2) represents the change in mean velocity with position, and the second term represents the gradient of turbulent velocity fluctuations. Because the turbulent mean velocity and turbulent velocity fluctuations must both satisfy continuity, and this project deals with mean values, the second term of equation (2.2) can be subtracted out to yield the continuity equation used in this project.

\[
\frac{\partial \bar{U}_i}{\partial x_i} = 0 \quad (2.3)
\]

This approach to the continuity equation is especially convenient from the perspective of code writing because it means that the turbulent velocity fluctuations in the continuity equation can be ignored. This makes the form of equation (2.3) like equation (2.1) and therefore the solution procedures for turbulent and laminar continuity equations are able to be performed identically. The difference is that if the solution method is turbulent, then the velocities involved are always mean values.

Equation (2.1) in Cartesian form is:

\[
\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0 \quad (2.4)
\]
With the physical significance of the continuity equation and the Marker-and-Cell method of section 2.2 in mind, it is easy to see the transition to a finite difference equation. Each of the three terms of equation (2.4) can be written in finite difference terms by stating that the derivative of a velocity with respect to position is the difference between the velocity out of the cell minus the velocity in to the cell divided by the distance between the cell faces, and velocity vectors.

The finite difference form of the continuity equation is:

\[
\frac{U_{ijk} - U_{i-1,jk}}{\Delta x_{ci}} + \frac{V_{i+j,k} - V_{ij-1,k}}{\Delta y_{cj}} + \frac{W_{i,j+1,k} - W_{i,j,k-1}}{\Delta z_{ck}} = 0
\]  

(2.5)

Where \(\Delta x_{ci}, \Delta y_{cj},\) and \(\Delta z_{ck}\) are the distances between the appropriate velocity vectors.

2.3.2 Momentum Equations

The constant density momentum, or Navier Stokes, equations are given in Eulerian form in the following equation.

\[
\frac{\partial U_i}{\partial t} + U_i \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \nu \nabla^2 U_i + g_i
\]  

(2.6)

The equation used for turbulence modeling is different and is shown in the following equation. The bars over the velocities denote mean velocity values.

\[
\frac{\partial \bar{U}_i}{\partial t} + \bar{U}_i \frac{\partial \bar{U}_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_i} \left[ \nu_i \left( \frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \right] + g_i
\]  

(2.7)
Equation (2.7) above is obtained as a result of substituting the velocity components with the mean values and fluctuations and then taking the time average. Another important step in the derivation is the assumption that the viscous stresses are negligible.

The finite difference equations calculate a new velocity value based on the old values of velocities and pressures on the right hand side of the equation. Not that the P's in the finite difference equations represent the ratio of the pressure to the density, and not just the pressure as in the previous equations. A constant density finite difference form of equation (2.6) explicitly solved for the time advanced laminar U value is:

\[
U_{i,j,k}^{\text{new}} = U_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i+1,j,k}}{\Delta x} + g_x + \text{VISX} - \text{FUX} - \text{FUY} - \text{FUZ} \right]
\]

(2.8)

Where the viscous fluxes are included in the VISX term, and the convective fluxes are denoted by the FUX, FUY, and FUZ terms. The viscous flux term in expanded finite difference form is given below.

\[
\text{VISX} = \nu \left[ \left( \frac{U_{i+1,j,k} - U_{i,j,k}}{\Delta x_{i+1}} - \frac{U_{i,j,k} - U_{i-1,j,k}}{\Delta x_{i-1}} \right) \Delta x_{i} + \left( \frac{U_{i,j+1,k} - U_{i,j,k}}{\Delta y_{j+1}} - \frac{U_{i,j,k} - U_{i,j-1,k}}{\Delta y_{j-1}} \right) \Delta y_{j} + \left( \frac{U_{i,j,k+1} - U_{i,j,k}}{\Delta z_{k+1}} - \frac{U_{i,j,k} - U_{i,j,k-1}}{\Delta z_{k-1}} \right) \Delta z_{k} \right]
\]

(2.9)

Where \(\Delta x_{i+1}\) represents both the distance between \(U_{i+1,j,k}\) and \(U_{i,j,k}\) and the width of cell \(i+1\) in the x direction. The \(\Delta x\) term represents the distance between the midpoints of cells \(i\) and \(i+1\) in the x direction. The reason for the designation of separate distances is because the above finite difference equation has been written for a variable grid spacing. As
previously mentioned, Weathers [1992] describes the equivalent constant grid spaced equations.

The convective flux terms are as follows.

\[
F_{\text{UX}} = \frac{1}{4} \left\{ [(U_{i,j,k} + U_{i+1,j,k})^2 + \overline{\alpha} |U_{i,j,k} + U_{i+1,j,k}|(U_{i,j,k} - U_{i+1,j,k})] / \Delta x_{i+1} ight. \\
\left. - [(U_{i-1,j,k} + U_{i,j,k})^2 + \overline{\alpha} |U_{i-1,j,k} + U_{i,j,k}|(U_{i-1,j,k} - U_{i,j,k})] / \Delta x_i \right\} 
\]

\[
F_{\text{UY}} = \frac{1}{4} \left\{ [(V_{i,j,k} + V_{i+1,j,k})^2 + \overline{\alpha} |V_{i,j,k} + V_{i+1,j,k}|(U_{i,j,k} - U_{i+1,j,k})] / \Delta y_j ight. \\
\left. - [(V_{i,j-1,k} + V_{i+1,j-1,k})^2 + \overline{\alpha} |V_{i,j-1,k} + V_{i+1,j-1,k}|(U_{i,j-1,k} - U_{i+1,j-1,k})] / \Delta y_{j-1} \right\} 
\]

\[
F_{\text{UZ}} = \frac{1}{4} \left\{ [(W_{i,j,k} + W_{i+1,j,k})^2 + \overline{\alpha} |W_{i,j,k} + W_{i+1,j,k}|(U_{i,j,k} - U_{i+1,j,k})] / \Delta z_k ight. \\
\left. - [(W_{i,j,k-1} + W_{i+1,j,k-1})^2 + \overline{\alpha} |W_{i,j,k-1} + W_{i+1,j,k-1}|(U_{i,j,k-1} - U_{i+1,j,k-1})] / \Delta z_{k-1} \right\} 
\]

Where \( \overline{\alpha} \) is a coefficient which determines the amount of upstream or donor cell differencing, and should be based on the time step, grid spacing and velocity. More specifically, \( \overline{\alpha} \) should be about 1.5 times larger than the maximum value of

\[
\begin{vmatrix}
| U \Delta t / \Delta x | & | V \Delta t / \Delta y | & | W \Delta t / \Delta z |
\end{vmatrix}
\]

according to Hirt, Nichols, and Romero [1975].
Equations (2.10)-(2.12) are the convective flux terms used to obtain the time advanced \( U \) velocity. The velocity equations for the other two components, viscous flux equations, and convective flux equations are similar to the \( U \) velocity equations and are included in the Appendix.

The finite difference representation for equation (2.7) is solved for the turbulent time advanced value just as the laminar equation (2.8).

\[
U_{\text{new}}^{ijk} = U_{\text{old}}^{ijk} + \Delta t \left[ \frac{P_{ijk} - P_{i+1,j,k}}{\Delta x_i} + g_s + \text{GAMMAX} - \text{FUX} - \text{FUY} - \text{FUZ} \right]
\]  

(2.13)

The equation for \( \text{GAMMAX} \) is as follows.

\[
\Gamma_x = \nu_t \left[ 2 \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} + \frac{\partial^2 V}{\partial x \partial y} + \frac{\partial^2 W}{\partial x \partial z} \right] + 2 \frac{\partial \nu_t}{\partial x} \left[ \frac{\partial U}{\partial x} \right] \]  

(2.14)

2.3.3 Scalar Equations

The equation structure for scalar quantities such as temperature, mixture fraction, turbulent kinetic energy, and turbulent dissipation is very similar to the previously outlined momentum equations. The turbulent kinetic energy equation and the turbulent energy dissipation equation will be discussed in detail in the next sections. The temperature equation is presented here. The finite difference program used in this study is capable of
performing concentration analysis, but the mixture concentration results are not included in this analysis. Because the form of the mixture concentration equation is identical to the temperature equation, and mixture concentration results are not involved in this study, the mixture concentration equation will be presented in the Appendix.

\[
\frac{\partial T}{\partial t} + \frac{\partial}{\partial x_i} \left( U_i T \right) = \alpha \left( \nabla^2 T \right) \quad (2.15)
\]

Equation (2.15) is the heat transport equation used throughout the three dimensional domain for laminar and turbulent solution procedures.

The heat transport finite difference equation and the other scalar finite difference equations are derived in the same manner as the velocity component time advanced finite difference equations. The time advanced finite difference form of the heat transport equation is:

\[
T_{i,j,k}^{\text{new}} = T_{i,j,k}^{\text{old}} + \Delta t \left[ \text{VIST} - \text{FTX} - \text{FTY} - \text{FTZ} \right] \quad (2.16)
\]

where

\[
\text{VIST} = \alpha \left[ \left( \frac{T_{i+1,j,k} - T_{i,j,k}}{\Delta x_i} \right) - \left( \frac{T_{i,j,k} - T_{i-1,j,k}}{\Delta x_{i-1}} \right) \right] / \Delta x_i + \left( \frac{T_{i,j+1,k} - T_{i,j,k}}{\Delta y_j} \right) - \left( \frac{T_{i,j,k} - T_{i,j-1,k}}{\Delta y_{j-1}} \right) / \Delta y_j + \left( \frac{T_{i,j,k+1} - T_{i,j,k}}{\Delta z_k} \right) - \left( \frac{T_{i,j,k} - T_{i,j,k-1}}{\Delta z_{k-1}} \right) / \Delta z_k \right] (2.17)
\]

In equation (2.16), \( \alpha \) is the thermal diffusivity of air.
The convective flux terms are:

\[
FT_X = \frac{1}{2} \left\{ \left[ U_{i,j,k} (T_{i,j,k} + T_{i+1,j,k}) + \alpha \left[ U_{i,j,k} |(T_{i,j,k} - T_{i+1,j,k})/\Delta x_i \right] \right] \right. \\
- \left[ \left[ U_{i-1,j,k} (T_{i-1,j,k} + T_{i,j,k}) + \alpha \left[ U_{i-1,j,k} |(T_{i-1,j,k} - T_{i,j,k})/\Delta x_{i-1} \right] \right] \right. \\
\]

\[
FT_Y = \frac{1}{2} \left\{ \left[ V_{i,j,k} (T_{i,j,k} + T_{i,j+1,k}) + \alpha \left[ V_{i,j,k} |(T_{i,j,k} - T_{i,j+1,k})/\Delta y_j \right] \right] \right. \\
- \left[ \left[ V_{i,j-1,k} (T_{i,j-1,k} + T_{i,j,k}) + \alpha \left[ V_{i,j-1,k} |(T_{i,j-1,k} - T_{i,j,k})/\Delta y_{j-1} \right] \right] \right. \\
\]

\[
FT_Z = \frac{1}{2} \left\{ \left[ W_{i,j,k} (T_{i,j,k} + T_{i,j,k+1}) + \alpha \left[ W_{i,j,k} |(T_{i,j,k} - T_{i,j,k+1})/\Delta z_k \right] \right] \right. \\
- \left[ \left[ W_{i,j-1,k} (T_{i,j-1,k} + T_{i,j,k}) + \alpha \left[ W_{i,j-1,k} |(T_{i,j-1,k} - T_{i,j,k})/\Delta z_{k-1} \right] \right] \right. \\
\]

\[
(2.18)
\]

\[
(2.19)
\]

\[
(2.20)
\]

2.3.4 Turbulent Kinetic Energy Equation

For additional turbulence theory which is not included in this report, one should examine Hinze [1987], and for a thorough discussion of turbulence k-ε modeling consult Rodi [1980]. For an overview of turbulence theory, turbulent k-ε modeling, and more about how they apply to this project, see Weathers [1992]. All of these references discuss the nature of turbulence, turbulent properties, and the closure problem encountered in modeling. The turbulence model used in this research is a two-equation, k-ε model with the option of using wall functions or a low-Reynolds number algorithm. It should be noted that the k-ε model is not the best predictor of turbulent flow, and it has its limitations. For
the scope of this project the k-ε model has been chosen, but future studies will probably want to use a more detailed, complex model.

The turbulent kinetic energy equation is given below.

\[
\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j} (U_j k) = \frac{\partial}{\partial x_j} \left[ \frac{\nu_t \partial k}{\sigma_k \partial x_j} \right] + \nu_t \frac{\partial U_i}{\partial x_j} \left[ \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] - \mathcal{E} \quad (2.21)
\]

where \( \sigma_k \) is an empirical constant, and is usually set equal to 1.0.

For this study, terms \( \Theta \) and \( \Pi \) are defined as:

\[
\Theta(k) = \frac{\partial}{\partial x_j} \left[ \frac{\nu_t \partial k}{\sigma_k \partial x_j} \right] \quad (2.22)
\]

\[
\Pi = \frac{\partial U_i}{\partial x_j} \left[ \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] \quad (2.23)
\]

Previous equations (2.22) and (2.23) can be substituted into equation (2.21) for simplification.

The finite difference equation for the turbulent kinetic energy time advanced values is:

\[
k_{i,j,k}^{\text{new}} = k_{i,j,k}^{\text{old}} + \Delta t \left[ \text{THETAK} + \nu_t \cdot \Pi - \text{FKX} - \text{FKY} - \text{FKZ} - \mathcal{E} \right] \quad (2.24)
\]

The finite difference terms for the convective fluxes, \( \Theta \), and \( \Pi \) are in the Appendix.
2.3.5 Turbulent Dissipation Equation

The $\varepsilon$ equation, using the $\Theta$ function and $\Pi$ term, can be written as follows.

$$
\frac{\partial \varepsilon}{\partial t} + \frac{\partial}{\partial x_j} (U_j \varepsilon) = \Theta(\varepsilon) + \frac{\nu_1 C_1 F_1 \varepsilon}{k} \Pi - C_2 F_2 \frac{\varepsilon^2}{k} + E
$$

(2.25)

And the time advanced finite difference equation, which is again similar in form to all the other time advanced equations, is given below. Also, the flux terms can be found in the Appendix.

$$
\varepsilon_{i,j,k}^{\text{new}} = \varepsilon_{i,j,k}^{\text{old}} + \Delta t \left[ \Theta \varepsilon + \frac{\nu_1 C_1 F_1 \varepsilon}{k} \Pi - C_2 F_2 \frac{\varepsilon^2}{k} - F_{EX} - F_{EY} - F_{EZ} + E \right]
$$

(2.26)
2.4 Boundary Conditions

The value of the three velocity components, temperature, kinetic energy, and dissipation at the boundaries must be specified with some degree of accuracy in order to achieve an accurate solution. This section discusses the boundary conditions used at the solid walls, at the inlet and at the outlet. Of course for the laminar case, the turbulent kinetic energy and the turbulent dissipation do not need to be specified because they do not affect a laminar solution.

2.4.1 Wall Boundary Conditions

The boundary conditions at the walls have evolved along with turbulence modeling. Wall boundary conditions are also an area of CFD methodology which causes much discussion among researchers. The main difference between different types of two-equation turbulence models is often the way in which the boundary conditions are imposed at the wall. Many researchers have used k-ε turbulence models with both a wall function method and a low-Reynolds number model. These are generally recognized as the two main types of k-ε models.

The code used in this study has the capacity to obtain solutions through a laminar, turbulent wall function, or a low-Reynolds number algorithm. With that in mind, the boundary conditions involved in the wall function and low-Reynolds number model are noted here. More details about each of these turbulent solution methods will be presented in a later section.
All walls in the room are non-porous. Therefore the normal velocity, $U_n$, is always equal to zero. The classical no slip and free slip conditions can be imposed as follows.

**No Slip:**

$$U_t = 0$$

**Free Slip:**

$$\frac{\partial U_t}{\partial x_n} = 0$$

where $U_t = \text{velocity of the fluid tangential to the wall}$

So, when the no slip condition is in effect, the fluid near the wall does not "slip". While the free slip, or slip, condition is in effect, the fluid is free to move along the wall. Usually, in CFD modeling, the no slip boundary condition is imposed when nodes exist inside the boundary layer, but if the grid size is larger than the boundary layer thickness, then the slip condition should yield better results.

2.4.1.1 Wall Function Boundary Conditions

Because of problems in early CFD modeling, many variations of the k-$\varepsilon$ model were devised. Launder and Spalding [1974] devised a model which produced dependable results at a significant distance away from the wall. In order to extend their model to include the wall region, they developed wall functions. Since the wall function was first introduced, there have been many changes by researchers with different applications. The wall functions used in this project come from Murakami [1989a,b, & 1990]. It should be noted that when the wall function method is used, the previously mentioned slip and no-slip conditions are replaced by the following equation.
\[
\left. \frac{\partial U}{\partial x_n} \right|_{\text{wall}} = \frac{m U_t}{y_n} \quad (2.27)
\]

where \( m = \) a constant (usually \( 1/7 \))
\( y_n = \) the normal distance from the boundary to the first velocity vector defined inside the boundary
\( U_t = \) the value of the first tangential velocity vector inside the boundary

The wall function boundary conditions are used only with the turbulent wall function k-\( \varepsilon \) model. The laminar and turbulent low-Reynolds number k-\( \varepsilon \) models use either the slip or no slip conditions.

According to [Murakami, et. al., 1989a,b] the value of the dissipation at the boundary when wall functions are used can be defined as

\[
\varepsilon = \frac{(C_{u}^{0.5} k)^{1.5}}{k y_n} \quad (2.28)
\]

The turbulent kinetic energy wall boundary condition can be imposed as:

\[
\frac{\partial k}{\partial x_n} = 0 \quad (2.29)
\]

2.4.1.2 Low-Reynolds Number Boundary Conditions

A different type of k-\( \varepsilon \) turbulence model which does not use wall functions is the low-Reynolds number model. The low-Reynolds number model, as its name suggests, is generally not valid for high Reynolds number flows. According to Lam and Bremhorst
[1981] the boundary condition at the wall for the turbulent dissipation when using the low-Reynolds number model can be imposed as:

\[
\frac{\partial E}{\partial x_0} = 0
\]  

(2.30)

It should be noted that, when using the low-Reynolds number k-\(\varepsilon\) model, the turbulent kinetic energy wall boundary condition is the same as the boundary condition imposed in the wall function k-\(\varepsilon\) analysis.

When one uses the turbulent low-Reynolds number model, either the slip or no slip conditions are used, as mentioned earlier. If the no slip condition is used, then fundamentally this means that the user expects to predict the flow within the boundary layer. In order to predict the behavior of the boundary layer flow, one must know the thickness of the boundary layer and implement a grid with nodes close enough to the wall to be included inside the boundary layer. Conversely, if one uses the slip boundary condition, then the boundary layer information is not obtained or predicted because the size of the cell at the wall is larger than the thickness of the boundary layer. In this study and in Weathers' thesis, the slip boundary condition is used in the low-Reynolds number model simulations.

2.4.1.3 Wall Temperature Boundary Conditions
The temperature boundary conditions were varied for different cases because one of the objectives of this study is to investigate the feasibility of predicting the outlet temperature and/or the convection coefficient. Three different temperature boundary conditions were used in this analysis. The first type of boundary condition simply sets the temperature of the node just outside the boundary equal to the wall temperature, 30°C. In Figure 2.4, \( T_1 = 30°C \).

The second type of boundary condition is slightly more complex than the first boundary condition. Ideally, the wall temperature would be set to 30°C in order to match the experimental experiments. Therefore, this boundary condition sets the temperature of the node just outside the boundary based on the temperature of the temperature just inside the boundary such that the temperature at the wall is maintained at a specified constant value. The equation for this boundary condition is as follows.

\[
T_1 = 2 * T_{wall} - T_2
\]

where \( T_{wall} \) = Temperature of the boundary
\( T_1 \) = Temperature of the node just outside the boundary
\( T_2 \) = Temperature of the node just inside the boundary

Figure 2.4 shows the location of \( T_1, T_2 \) and \( T_{wall} \). Equation (2.31) was developed because scalar quantities in this program are not defined at the boundaries, but at one half cell outside and inside of the boundary, as discussed earlier. Therefore, even though \( T_{wall} \) is a constant value which is specified, it is up to the previous equation and the values of \( T_1 \) and \( T_2 \) to maintain a linear temperature gradient across the boundary and the effective wall temperature at a constant value.
The third type of temperature boundary condition used at the wall is based on a heat balance of the boundary:

\[ k \frac{\partial T}{\partial x_1} = h_c b_{ref} - T_{wall} \xi \]  

(2.32)

where  
\( k \) = Thermal conductivity  
\( h_c \) = Experimental convection coefficient  
\( T_{ref} \) = Reference temperature (see Spitler 1990)
For this study, the reference temperature is defined as the temperature at the outlet. The heat conduction through the wall is equal to the heat convection away from the wall. This heat balance with convection involved utilizes a known heat transfer coefficient. Therefore, this type of boundary condition requires prior knowledge of the convection coefficient and cannot be used to predict the convection coefficient.

The experimental values of the convection coefficients from Spitler [1990] were used in conjunction with this boundary condition. A separate convection coefficient is used for the floor, ceiling, and the four walls. The equations which yield the experimental convection coefficients for a wall inlet situation are taken from page 133 of Spitler's 1990 doctoral thesis and are given below.

For $0.002 < J < 0.011$

Walls: $h = 0.6 + 63.1 \ J^{0.4}$  \hspace{1cm} (2.33)

Ceiling: $h = -0.4 + 43.2 \ J^{0.4}$ \hspace{1cm} (2.34)

Floor: $h = 2.5 + 32.1 \ J^{0.4}$ \hspace{1cm} (2.35)

where $J$ is the jet momentum number which can be calculated as follows:

$J = ACH \ast \frac{U_{in}}{g}$  \hspace{1cm} (2.36)

ACH = the number of air changes per hour

The values for the convection coefficients used in this analysis are given in the following table.
TABLE 2.1
CONVECTION COEFFICIENT EXPERIMENTAL VALUES

<table>
<thead>
<tr>
<th></th>
<th>ACH: 15 (W/m² °C)</th>
<th>ACH: 50 (W/m² °C)</th>
<th>ACH: 100 (W/m² °C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walls:</td>
<td>2.448</td>
<td>6.249</td>
<td>10.018</td>
</tr>
<tr>
<td>Ceiling:</td>
<td>3.440</td>
<td>5.374</td>
<td>7.291</td>
</tr>
<tr>
<td>Floors:</td>
<td>1.225</td>
<td>3.828</td>
<td>6.408</td>
</tr>
</tbody>
</table>

2.4.2 Inlet Boundary Conditions

Inlet boundary conditions for a room airflow model are also important in obtaining an accurate solution as demonstrated by Weathers [1992]. First, he used uniform velocity profiles which set the u velocity component at every inlet cell equal to a constant value. Later he studied the experimental data and found that the experimental inlet jet was not uniform. Using the experimental data as a guide, he approximated inlet velocity profiles and used them as inputs into the CFD program rather than the uniform velocity profiles. He found that the approximated inlet profiles produced results which matched the experimental data better than the uniform velocity profiles. The uniform and approximated velocity profiles for the normal velocity are summarized in the following table where h is the height of the inlet.
TABLE 2.2
INLET BOUNDARY CONDITION FOR U-NORMAL
[Weathers, 1992]

<table>
<thead>
<tr>
<th>Flow Rate (ACH)</th>
<th>Uniform Profile (m/s)</th>
<th>Inlet Approximation (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.345</td>
<td>0.690 (0 &lt; y &lt; h/2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.000 (h/2 &lt; y &lt; h)</td>
</tr>
<tr>
<td>50</td>
<td>1.150</td>
<td>1.1 + (\frac{8y}{5h}) (0 &lt; y &lt; h/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.5 (h/4 &lt; y &lt; 3h/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.7 - (\frac{8y}{5h}) (3h/4 &lt; y &lt; h)</td>
</tr>
<tr>
<td>100</td>
<td>2.300</td>
<td>2.35 + (\frac{8y}{5h}) (0 &lt; y &lt; h/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.75 (h/4 &lt; y &lt; 3h/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.95 - (\frac{8y}{5h}) (3h/4 &lt; y &lt; h)</td>
</tr>
</tbody>
</table>

For a graphical representation of the inlet profile information of Table 2.2, see Weathers [1992]. The tangential velocity at the inlet is zero. The inlet temperature is specified, and equations (1.12) & (1.13) which have been taken from Awbi [1989] are used to obtain a value for \(k\) and \(\varepsilon\) at the inlet.

As mentioned earlier, the inlet for this project is 0.4 m. wide by 0.9 m. high. It is located on the sidewall approximately halfway between the ceiling and the floor (see Figure 1.1). Across the inlet there are 8 nodes in the y direction and 4 nodes in the z direction.

2.4.3 Outlet Boundary Conditions
Compared to the boundary conditions imposed at the walls and the inlet, the outlet boundary conditions are relatively simple. The equation used to calculate the normal u-velocity at the outlet is:

\[ U_n = U_i \frac{A_i}{A_{out}} \]  

(2.37)

When the areas of the inlet and outlet are both equal, the inlet velocity and outlet velocity are both equal. It should be noted that there are a few cases in this study when a variable grid is used and it is not possible to use an inlet area that is exactly equal to the outlet area. Again, the tangential velocity at the outlet is zero, just like the inlet condition.

The temperature, turbulent kinetic energy and turbulent dissipation of the fictitious nodes just outside the boundary at the outlet are each set equal to the appropriate values just inside the boundary. Therefore,

\[ \frac{\partial T}{\partial x_n} = 0, \quad \frac{\partial k}{\partial x_n} = 0, \quad \text{and} \quad \frac{\partial \varepsilon}{\partial x_n} = 0. \]  

(2.38)
2.4.4 Initial Conditions

It is rather simple to set the initial values of the variables throughout the domain, but it is also an important part of the modeling process, especially when performing a turbulent model. The initial values for all three velocity components, the mixture concentration and the pressure at every node inside the domain are set to zero. The initial values for the turbulent kinetic energy and the turbulent dissipation are set to a percentage of the inlet values. The initial temperature value is input by the user and changes with the air flow rate. The following table gives the initial temperature values used in the computational models.

<table>
<thead>
<tr>
<th>TABLE 2.3</th>
<th>INITIAL TEMPERATURE VALUES (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 ACH</td>
<td>30 ACH</td>
</tr>
<tr>
<td>28.5</td>
<td>26.5</td>
</tr>
</tbody>
</table>

The above values were obtained from experimental results presented in Spitler's 1990 thesis on page 93.

2.5 Turbulence Models

There are many different types of turbulence models which have been developed by a number of researchers for a wide range of fluid dynamic applications. As mentioned earlier, there is an option in the computer program which allows the user to perform a
laminar simulation, a turbulent k-ε simulation using wall functions or a low-Reynolds number turbulent k-ε simulation. For each of the k-ε turbulent simulations, there exists empirical constants and functions which distinguish the different types of turbulence models.

Equations for the Local Reynolds Number \( \left( R_y \right) \) and Turbulent Reynolds Number \( \left( R_T \right) \), which are used with the low-Reynolds number model, are given below.

\[
R_y = \frac{\sqrt{k y_n}}{\nu}
\]  

(2.39)

\[
R_T = \frac{k^2}{\nu \epsilon}
\]  

(2.40)

where \( y_n \) = the normal distance to the wall (m).

Constants for the standard, or wall function, k-ε model and the low-Reynolds number model used in this research are given below in Table 2.4.

**TABLE 2.4**

<table>
<thead>
<tr>
<th>( C_\mu )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( \sigma_k )</th>
<th>( \sigma_\epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09</td>
<td>1.44</td>
<td>1.92</td>
<td>1.0</td>
<td>1.3</td>
</tr>
</tbody>
</table>

The following table gives the empirical functions which have been used in this study for the standard k-ε model and the low Reynolds number k-ε model.
TABLE 2.5
FUNCTIONS USED IN TURBULENCE MODELING

<table>
<thead>
<tr>
<th>Model Type</th>
<th>$F_\mu$</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Low-Reynolds</td>
<td>$\mu$</td>
<td>$e^{-(A_\mu R_T)}$</td>
<td>$1 + 0.05 \mu$</td>
<td>$1 - e^{(-R_T)}$</td>
</tr>
</tbody>
</table>

As stated earlier, for more information about various turbulent constants and functions which have been used by other researchers, consult Weathers' [1992] thesis.

2.6 Grid Resolution

The uniform grid used in this study is equivalent to the 40 x 24 x 24 "very fine" grid used by Weathers [1992]. The variable grid used in this thesis is 41 x 25 x 25 and is only slightly non-uniform. It expands linearly from the edges to the center. In other words, the largest cell is the center cell, and the smallest cells are the first and the last cells. In the x direction the smallest cell is 10.04 cm. and the center cell is 12.45 cm. The smallest cell in the y and z directions is 10.3 cm. and the center cell is 11.75 cm. Figure 2.5a below shows the non-uniform cell resolution for the x direction in graphical terms. The non-uniform grid resolution in the z and y directions is identical and is displayed in Figure 2.5b.

Ideally, many other variable grids would have been used with a large range of cell sizes, but time limitations did not permit.
Figure 2.5a  Non-Uniform Grid Resolution in the X Direction

Figure 2.5b  Non-Uniform Grid Resolution in the Z (and Y) Directions
Figure 2.6a Uniform Grid in X and Y Directions

Figure 2.6b Variable Grid in X and Y Directions
Figures 2.6a and 2.6b show uniform and variable grid lines in x and y directions. The third direction (z) is omitted because it is the same as the y direction, as stated previously.

2.7 Error Analysis

Various error numbers were calculated in order to compare the experimental results to the computational results directly and quantitatively. Global velocity, global temperature, and two-dimensional velocity error numbers were used.

2.7.1 Global Error Number

The global error number used in Weathers’ study [1992] is also used here for help in quantitatively evaluating the accuracy of the computational velocity results.

\[
\text{GEN} = \frac{\bar{E}_{\text{abs}}}{V_{\text{max}}} \quad (2.41)
\]

where \( \bar{E}_{\text{abs}} \) = average absolute error for the velocity.
\( V_{\text{max}} \) = maximum experimental velocity.
2.7.2 Slice Error Number

A two-dimensional error number was also used in the same manner as the global error number in order to check the accuracy of a specific plane of data. The slice error number can be used to isolate a problem with a "slice" of the domain. The two-dimensional or slice error number is as follows:

\[ \text{SEN} = \frac{E_{\text{abs}}}{V_{\text{max}}} \]  \hspace{1cm} (2.42)

where \( E_{\text{abs}} \) = average absolute error for the velocity.  
\( V_{\text{max}} \) = maximum experimental velocity within the plane being studied.

In this study, the slice error number was mainly used to compare the computational wall jet to the experimental wall jet. Plots at \( z=0.127 \) m. and slice error numbers are provided in chapter 3.

2.7.3 Global Temperature Error Number

A quantity similar to the global error number was implemented in this study for the computational temperatures. The global temperature error number is as follows.

\[ \text{GTEN} = \frac{E_{\text{abs}}}{T_{\text{max}} - T_{\text{min}}} \]  \hspace{1cm} (2.43)

where \( E_{\text{abs}} \) = average absolute error for the temperature.  
\( T_{\text{max}} \) = maximum experimental temperature.
\[ T_{\text{min}} = \text{minimum experimental temperature.} \]

The global temperature error number was used to determine the appropriate time to make temperature comparisons between the experimental and computational results. An explanation of how the global temperature error number was used is detailed later in chapter 3.

2.7.4 Extrapolation Problem

Weathers encountered a problem in his study because of the statistical package that he used to plot his results, Systat™. Systat™ extrapolated values near the wall for the experimental and numerical results. Because the experimental results are not as dense as the numerical results, the extrapolation problem was magnified in the experimental plots. This extrapolation problem has been eliminated in the present study through the use of a software package which is better suited for displaying transient fluid dynamic data, NCSA Image 3.1.1 [NCSA, 1990]. The NCSA Image software is developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Weathers' extrapolation problem is most evident with his 15 ACH experimental plot. His plot showed unusually high velocity magnitudes in the center of the plot at the floor which is inconsistent with the experimental data. NCSA Image generates correct velocity magnitudes, as shown through the comparison made with Figure 2.7, and remedies the extrapolation problem.
Figure 2.7 Experimental Velocity Magnitudes at 15 ACH
3.0 RESULTS & DISCUSSION

3.1 Overview of Computational Analysis

Many different computational runs were performed at various air flow rates, the vast majority of which were performed at 15, 50, & 100 ACH. Three different temperature boundary conditions were investigated in order to improve the prediction of heat transfer. A variable grid was implemented in an attempt for better grid coverage and more accuracy at the inlet and outlet.

Section 3.2 discusses the importance of correctly recognizing convergence and stability concerns. In section 3.3, simulations made to investigate the feasibility of predicting convection heat transfer will be presented. Then, in section 3.4 simulation results will be discussed and converged plots will be shown. Finally, sections 3.4.4 and 3.4.5 include quantitative comparisons involving error numbers.

3.2 Convergence & Stability

Before specific, detailed results of this project are provided and discussed, it seems appropriate to discuss the more general concepts of convergence and stability and their importance. During the course of this thesis, it was discovered that convergence should be more carefully considered in each computational case. Prior to this research, the point of convergence probably was not attained. A discussion of convergence follows, and then stability techniques are addressed.
Ideally, the minimum global velocity magnitude error number, the minimum two-dimensional error number, and the minimum global temperature error number would occur simultaneously as the stratification values matched the experimental stratification values and the velocities converge. Unfortunately, most of these indications of convergence occur at different points in the transient solution process. Therefore, for any given computational case, it is difficult to determine the actual point of convergence.

If one bases convergence solely on the point in the computational simulation when velocity magnitudes are no longer changing significantly, then the 100 ACH models converge on average after 100 seconds of simulation. It should be made clear that the times reported here are not CPU times, and that actual run times of the simulations are much, much greater. The velocity components of the 50 ACH models converge at about 250 seconds of simulation, and the 15 ACH models require almost 500 seconds. All simulations were performed out to the times listed above.

The times given above were determined by watching transient, animated, color, computational simulations using NCSA Image 3.1.1 [1990]. The visualization tools were used in this project to help to determine the point of convergence. Prior quantitative techniques failed to require the simulations to run until convergence. With the older and less sophisticated visualization techniques, it was nearly impossible to recognize that convergence had not been achieved. Eventually, one should return to the quantitative techniques once stricter, proven convergence criteria has been defined.

In order to demonstrate the process of convergence, transient plots are included in this section. Discussion is provided in an attempt to present a larger portion of the transient effect, but it is difficult to do the transient color animation, which the plots were taken from, justice. Plots were chosen at the 25, 35, 70 and 100 second intervals of the 50 ACH
low Reynolds number variable grid model. The grid resolution is shown in Figures 2.5a and 2.5b. Later in section 3.3, the convergence question is approached from a different angle with the use of the global temperature error number. There the global temperature error number is monitored over time in order to determine when the computational temperatures best match the experimental temperatures.

First in the transient simulation, the effect of buoyancy is demonstrated as the wall jet enters the room and attaches itself to the floor. As the flow hits the opposing wall, it is forced upward towards the ceiling. Once it reaches the ceiling, a small recirculation zone is created. This zone begins to occur at about the 20-25 second mark in the solution process. The 25 second plot, in the top half of Figure 3.1, shows this small recirculation pattern in the upper right hand corner.

As the simulation continues two main things happen. 1.) The overall temperatures in the room are decreasing, and the cold inlet jet does not "see" as large a temperature differential which results in a smaller buoyant effect. Therefore, the jet wants to lift off the floor. 2.) The recirculation pattern continues to move along the ceiling back towards the inlet. These two effects are shown in the lower half of Figure 3.1, the 35 second plot.

While the wall jet continues to attempt to lift off the floor, the recirculation portion of the flow completes the cycle and reaches the inlet jet. When this happens, the wall jet is physically pushed back down to the floor, at about 40-45 seconds. From 45-70 seconds, the wall jet shape does not change drastically. The 70 second plot is shown in the top half of Figure 3.2.

After the 70 second mark, the strength of the recirculation region trails off, and the wall jet begins to lift off the floor once again. The 100 second plot, in Figure 3.2, shows a wall jet
which has already detached from the floor. After the 100 second plot, the jet reattaches to the floor, and begins to climb up the wall again. But this time, the effects are not as pronounced. The airflow patterns continue to change between something similar to the 70 second plot with strong recirculation and back to a profile similar to the 100 second plot with no apparent recirculation. The fluid must go through these cycles and the effects must damp out in order for one to conclude that convergence has been achieved.
Figure 3.1 Transient Low-Reynolds Number Results at 50 ACH
Figure 3.2 Transient Low-Reynolds Number Results at 50 ACH
Stability was a significant problem at the beginning of this study. At the present time, the only model which requires special attention is the 100 ACH low-Reynolds number model. For the 100 ACH low-Reynolds number model, the initial values of turbulent kinetic energy and turbulent dissipation are reduced in order to increase stability and allow the simulation to run. This essentially makes the model less turbulent initially. Awbi [1989] stated that changing the initial or inlet values does not significantly affect the computational results, but changing these values can increase stability. For better stability with the 100 ACH low-Reynolds number model, the initial turbulent dissipation was decreased by 84%, and the turbulent kinetic energy was decreased by 60%.

3.3 Prediction of Temperatures and Convection Coefficients

Different temperature boundary conditions were investigated, as previously discussed. The reason for varying the temperature boundary conditions was to explore the possibility of predicting a convection coefficient which matches the experimental convection coefficient.

Another option was to use the experimentally determined convection coefficients as inputs to the CFD models and monitor the calculated outlet temperature and mean temperatures to see if they matched experimental results.

The results for the laminar simulations made with varying boundary conditions for 50 ACH are shown in Table 3.1. The predicted mean temperatures may be compared to the experimental mean temperature of 25.8°C. The experimental convection coefficient for the wall at z=0.0 m. is 6.249 W/m²°C.
### TABLE 3.1

CONVECTION COEFFICIENT AND TEMPERATURE PREDICTION

<table>
<thead>
<tr>
<th>B. C. Type:</th>
<th>Description</th>
<th>G. E. N.</th>
<th>Mean Temperature</th>
<th>Predicted Convection Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$T_1 = 30^\circ C$</td>
<td>0.06563</td>
<td>21.0$^\circ$ C</td>
<td>17.363</td>
</tr>
<tr>
<td>2</td>
<td>$T_1 = 2*T_{wall} - T_2$</td>
<td>0.10234</td>
<td>30.0$^\circ$ C</td>
<td>29.679</td>
</tr>
<tr>
<td>3</td>
<td>$T_1$ uses experimental $h_c$</td>
<td>0.06559</td>
<td>26.0$^\circ$ C</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Two temperature boundary conditions were used without the experimentally determined convection coefficient. Both are much simpler than the boundary condition which uses experimental convection coefficients. The first temperature boundary condition, referred to as type 1 for clarity, simply sets the temperature just outside the wall equal to the temperature at the wall, 30$^\circ$C in all CFD cases performed. The global error number for the laminar 50 ACH model using this type 1 boundary condition is slightly higher than the global error number for the laminar 50 ACH model which used the experimental convection coefficient.

The initial temperature throughout the 50 ACH models is 25.5$^\circ$C, as mentioned earlier. The type 1 boundary condition causes the outlet temperature, and the overall temperature to decrease, and eventually converge at the inlet temperature of 21$^\circ$C.

The second type of temperature boundary condition used sets the gradient across the boundary such that the temperature at the boundary is held constant at 30$^\circ$C. The global
error number for the laminar 50 ACH model using the type 2 temperature boundary condition is the highest of the three. This boundary condition causes the temperatures to converge at 30°C, and the high predicted convection coefficient follows.

When one considers the first two temperature boundary conditions, even though the end result is not desirable, the trend is expected. The first temperature boundary condition does not predict enough heat transfer to the room, while the second temperature boundary condition over predicts the heat transfer. Setting the node just outside the boundary equal to the desired temperature at the boundary would actually give the temperature at the boundary a lower temperature than desired. Therefore less heat transfer to the room can be expected in this case. The second temperature boundary condition probably over predicts the heat transferred to the room because if the temperature just inside the wall is not 30°C, then the temperature just outside the boundary is set higher than 30°C. This should set the equivalent temperature at the boundary to 30°C, but the grid size might have hampered its effectiveness. More grid points closer to the wall might help one or both of these two boundary conditions to more accurately predict the convection coefficients.

The temperature boundary condition type 2 is more closely aligned with the physical significance of holding the wall temperature at a constant 30°C throughout the experiment than the type 1 boundary condition which sets the temperature outside the boundary equal to the temperature at the wall. Therefore one might expect that the type 2 boundary condition would produce more favorable results than the type 1 boundary condition. But, the type 2 temperature boundary condition does not produce better results, in terms of global error number analysis or convection coefficient prediction. The higher global error number can be explained when one considers the predicted convection coefficient, predicted outlet temperature, and the stratification plot for the model.
Since temperature boundary condition type 1 caused the temperatures to converge at 21°C and the type 2 boundary condition causes temperature convergence at 30°C, neither is satisfactory. A third type of boundary condition, which uses experimentally determined convection coefficients is preferred. The details of the use of this boundary condition and convection coefficients used were outlined in section 2.4.1.3.

Figure 3.3 Transient Temperature Stratification Values

Each point in Figure 3.3 represents the value calculated by averaging all temperatures at that height in the three-dimensional domain. A comparison to experimental results is shown below. Figure 3.3 shows that the average temperature profile continues to change past 100 seconds, which in the past would have been considered well past the point of convergence. All of the simulations performed included buoyant forces. If the temperatures in these buoyant simulations have not converged, then it is clear that the velocity components also have not converged.
Global temperature error numbers, as opposed to the global velocity magnitude error numbers previously discussed, were calculated for numerous reasons including help in the determination of when to compare the computational and experimental stratification data. The global temperature error numbers for the runs using temperature boundary conditions of type 1 and 2 increased steadily as the transient solution process continued. The global temperature error number for the type 1 temperature boundary condition is 0.333 at 95.0 seconds and continues to increase to 0.497 at 250 seconds. Likewise, the type 2 temperature boundary condition global temperature error number is 0.333 at 64.0 seconds and continues increasing to 250 seconds where the global temperature error number is 0.483.

The global temperature error number for the model using the type 3 temperature boundary condition is more worthy of discussion. Figure 3.4 shows the transient global temperature error number for the laminar 50 ACH model using the type 3 temperature boundary condition.

![Figure 3.4 Transient Global Temperature Error Number](image-url)
Note the minimum global temperature error number occurs at 126.0 seconds and is 0.087. Since the minimum global temperature error number in Figure 3.4 occurs at 126 seconds, this time was chosen for comparison of the computational temperatures against the experimental stratification values. Both plots are shown in Figure 3.5.

![Temperature Profile - 50 ACH](image)

**Figure 3.5** Comparison of Computational and Experimental Stratification Values

It was expected that the more complex boundary condition, type 3, which utilized convection coefficients calculated from experimental results, would match those experimental temperatures with greater accuracy than the other two boundary condition types. When one looks at the global error number analysis only, one could conclude that the type 1 boundary condition is a better choice because it produces an adequate global error number without requiring the user to know experimentally determined convection coefficients. But temperature prediction accuracy is also important. The type 3 boundary condition did not converge to either temperature extreme, in fact it matched the experimental stratification with acceptable accuracy as shown in Figure 3.5. It was also a
rather good predictor of the temperature at the outlet. Experimental air heat gain values are compared to computational air heat gain values in Figure 3.6. Note that exp designates experimental results, ru designates low Reynolds number uniform grid results, and rv designates low Reynolds number variable grid results. Figure 3.6 shows that the low Reynolds number model usually predicted air heat gain values lower than the experimental results.

![Figure 3.6 Comparison of Air Heat Gain Values](image)

3.4 Simulation Results

Other than the prediction of temperatures and convection coefficients, a main goal of this project was to implement a variable grid. Graphical results are presented below for a single "slice" of the room which contains the inlet jet. In each case, experimental results are presented in the first image; followed by simulation results for each model type (laminar, turbulent with wall functions, turbulent low-Reynolds number) and for uniform and variable grid. In the next section, quantitative results will be presented. In each air flow rate section the velocity magnitude sections have been plotted at the same scale which
permits direct wall jet comparisons between the experimental results, different fluid flow models, and grid types. The darker shades designate higher velocity magnitudes, of course. This section also allows one to study the shape of the wall jet and the amount of diffusion involved.

3.4.1 15 ACH Results

Experimental results at 15 ACH are shown in Figure 3.7. Figures 3.8, 3.9, and 3.10 show uniform and variable grid computational data for the laminar, turbulent wall function, and turbulent low-Reynolds number models, respectively.

The laminar sections, Figure 3.8, show that higher magnitudes were predicted with the non-uniform grid than with the uniform grid. The shape and spread of the jet are similar. The uniform plot shows higher velocities "climbing" the opposing wall. The transient analysis of the 15 ACH models showed this climbing effect quite clearly. When the wall jet moves into the room, it is affected by buoyancy and follows along the floor to the far wall. Air is forced up the wall, moves back towards the inlet along the ceiling, and eventually a full recirculation pattern occurs. This recirculation pattern begins to occur at about the 40 second simulation mark in most of these 15 ACH models. Because this is the slowest velocity studied, the effect of buoyancy on each model is more visible. The recirculation pattern is more likely to occur and is the easiest to view at this lower air flow rate with a greater buoyancy effect.

By the time in the simulation when the recirculating flow moves along the ceiling on its journey back towards the inlet, the temperatures in the domain have decreased due to the 21°C inlet jet. This decreases the effect of buoyancy on the inlet air. As this occurs, the
wall jet moves up and attempts to lift off the floor. When the recirculating flow returns to
the area of the inlet, it physically knocks the wall jet down towards the floor. This "lifting"
and "knocking" repeats as the simulation continues. But, the phenomenon is not as evident
further in the transient solution because the temperature domain is closer to convergence
and the simulation domain is three-dimensional.

The turbulent wall function model at 15 ACH, Figure 3.9, shows that the uniform grid plot
predicts a thinner wall jet than the variable grid section. The effect of buoyancy appears to
be greater in the non-uniform plot. The uniform grid profile is a unique 15 ACH plot
because the jet is not attached to the floor. In reality, the wall jet has been attached to
the floor during previous time steps but has eventually lifted off the floor.

When considering jet shape and thickness, the wall function model seems to be the poorest
match of the experimental results. Both the uniform and variable grid models would need
to display greater buoyancy in order to match the experimental data closer.

The two turbulent low-Reynolds number model sections, Figure 3.10, are the closest match
for the experimental data within their own grid types. The low-Reynolds number plots
show wall jets which are thicker than the wall function plots but are thinner than the
laminar plots. As stated earlier, the real test is a velocity magnitude analysis on a point by
point basis such as the 2-D error numbers presented later.
Figure 3.7 Experimental Velocity Magnitudes at 15 ACH
Figure 3.8 Laminar Velocity Magnitudes at 15 ACH
Figure 3.9 Wall Function Velocity Magnitudes at 15 ACH
Figure 3.10  Low-Reynolds Number Velocity Magnitudes at 15 ACH
3.4.2 50 ACH Results

Naturally, the 50 ACH plots do not show as much buoyant effect as the slower velocity plots. In the following 50 ACH experimental section at $z=0.127$ m, Figure 3.11, the highest velocity magnitudes are at the inlet, but there is another pocket of higher velocity airflow in the central region of the bottom half of the plot. Much like the 15 ACH computational results, none of the 50 ACH computational plots, Figures 3.12, 3.13, and 3.14, produce a section of a wall jet with this isolated pocket of higher velocity. In fact, the turbulent wall function model calculates a wall jet which does not include that section of the plot because the spread of the wall jet is not wide enough. In the experimental 50 ACH plot, it even appears as though this region could be an area of recirculation, which is definitely not present in the laminar or low-Reynolds number simulations.

In terms of velocity magnitude matching of the experimental results, the low-Reynolds number variable grid plot is best, with the variable grid laminar plot not far behind. The uniform low-Reynolds number and laminar plots follow, respectively, and the variable and uniform grid wall function plots are the worst velocity magnitude predictors, for reasons mentioned earlier.

In each of the computational figures, the effect of gravity is obviously present. In other words, the right half of the plot or tip of the wall jet has dipped down towards the floor. This visible buoyant effect is an improvement over Weathers' 50 ACH simulations. Weathers' 50 ACH data displays almost no buoyant effects. With each of his simulation types a wall jet is calculated which has completely separated from the floor. Wall jet profiles which more closely match the shape of the experimental wall jet have been produced with this study. The experimental data, Figure 3.11, shows a jet that has separated only slightly from the floor. Especially figures 3.12 and 3.13 display jets which
appear to be in the process of lifting off the floor or have just done so. The present results also have produced more pronounced wall jet differences between the model types than Weathers' results.
Figure 3.12  Laminar Velocity Magnitudes at 50 ACH
Figure 3.13 Wall Function Velocity Magnitudes at 50 ACH
Figure 3.14  Low-Reynolds Number Velocity Magnitudes at 50 ACH
3.4.3 100 ACH Results

Many of the comments which were made about the 50 ACH results could also be made about the 100 ACH results. Again, Weathers' results showed little or no effect of gravity. The low-Reynolds number results, Figure 3.18, and the laminar variable grid results of Figure 3.16 still display definite buoyant effects. The turbulent wall function model, Figure 3.17, has not predicted enough jet diffusion. The laminar results in Figure 3.16 showed much more jet diffusion than Weather's results.

The flow pattern of the 100 ACH experimental plot is interesting. Notice in Figure 3.15 that there is no immediate change of the incoming flow until the jet spreads both up and down in an S-shaped pattern. This S-shaped spreading begins to occur at about 1 meter into the room. It is possible that better experimental resolution might alter the shape of the spread. None of the 100 ACH computational data in this study has predicted an S-shaped spread. Either the jet starts to spread immediately like in Figures 3.16 and 3.17 or the jet doesn't really spread at all as in Figure 3.18.
Figure 3.15 Experimental Velocity Magnitudes at 100 ACH
Figure 3.16 Laminar Velocity Magnitudes at 100 ACH
Figure 3.17 Wall Function Velocity Magnitudes at 100 ACH
Turbulent Low-Reynolds Number Results

Section at $Z = 0.127 \text{m}$

Uniform Grid

Variable Grid

Figure 3.18 Low-Reynolds Number Velocity Magnitudes at 100 ACH
### 3.4.4 Global Velocity Magnitude Error Summary

A summary of velocity magnitude global error numbers is provided in Table 3.2 for direct comparison between the uniform grid the variable grid simulations. The global error number gives one a quantitative analysis of the how well the entire computational domain matches the experimental domain. The equation for the global error number has been presented earlier as equation (3.1).

Check marks note cases where the variable grid has shown improvements over the uniform grid simulations. The table below shows that over the entire domain the variable grid matches the experimental results about as well as the uniform grid. Using the variable grid seems to be more beneficial at the higher airflows.

**TABLE 3.2**

**GLOBAL ERROR NUMBERS**

<table>
<thead>
<tr>
<th>Model Type and Airflow Rate:</th>
<th>Uniform Grid</th>
<th>Variable Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laminar 15 ACH:</td>
<td>0.0371</td>
<td>0.0503</td>
</tr>
<tr>
<td>Wall Function 15 ACH:</td>
<td>0.0352</td>
<td>0.0378</td>
</tr>
<tr>
<td>Low-Reynolds 15 ACH:</td>
<td>0.0408</td>
<td>0.0421</td>
</tr>
<tr>
<td>Laminar 50 ACH:</td>
<td>0.0656</td>
<td>0.0633</td>
</tr>
<tr>
<td>Wall Function 50 ACH:</td>
<td>0.0828</td>
<td>0.0874</td>
</tr>
<tr>
<td>Low-Reynolds 50 ACH:</td>
<td>0.0595</td>
<td>0.0544</td>
</tr>
<tr>
<td>Laminar 100 ACH:</td>
<td>0.0700</td>
<td>0.0690</td>
</tr>
<tr>
<td>Wall Function 100 ACH:</td>
<td>0.0820</td>
<td>0.0774</td>
</tr>
<tr>
<td>Low-Reynolds 100 ACH:</td>
<td>0.0645</td>
<td>0.0639</td>
</tr>
</tbody>
</table>
3.4.5 2-D Velocity Magnitude Error Summary

As stated earlier, all two-dimensional, or "slice" error numbers reported here are for the z=0.127 m. section. This gives a measure of how much the variable grid is helping to improve the results at the inlet. As stated in section 3.2, all 15 ACH simulations were run to 500 seconds, all 50 ACH simulations were run to 250 seconds, and all 100 ACH models were run to 100 seconds. The uniform and variable grid slice error numbers are presented below in Table 3.3.

TABLE 3.3
SLICE ERROR NUMBERS AT Z=0.127 m.

<table>
<thead>
<tr>
<th>Model Type and Airflow Rate:</th>
<th>Uniform Grid</th>
<th>Variable Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laminar 15 ACH:</td>
<td>0.1276</td>
<td>0.0767</td>
</tr>
<tr>
<td>Wall Function 15 ACH:</td>
<td>0.1276</td>
<td>0.0736</td>
</tr>
<tr>
<td>Low-Reynolds 15 ACH:</td>
<td>0.1098</td>
<td>0.0644</td>
</tr>
<tr>
<td>Laminar 50 ACH:</td>
<td>0.1254</td>
<td>0.0979</td>
</tr>
<tr>
<td>Wall Function 50 ACH:</td>
<td>0.2144</td>
<td>0.1735</td>
</tr>
<tr>
<td>Low-Reynolds 50 ACH:</td>
<td>0.1090</td>
<td>0.0843</td>
</tr>
<tr>
<td>Laminar 100 ACH:</td>
<td>0.1520</td>
<td>0.1299</td>
</tr>
<tr>
<td>Wall Function 100 ACH:</td>
<td>0.2447</td>
<td>0.2055</td>
</tr>
<tr>
<td>Low-Reynolds 100 ACH:</td>
<td>0.0990</td>
<td>0.0934</td>
</tr>
</tbody>
</table>
The slice error numbers show slight improvement when the variable grid is used. Ideally, the variable grid would show improvement in every case of both the global error and slice error analysis. Improvement in the slice error number is more desirable because the reasons for using the variable grid are to improve resolution through the inlet, outlet, and near the wall.
4.0 CONCLUSIONS & RECOMMENDATIONS

4.1 Conclusions

The following conclusions were reached during this investigation:

1. The convergence of a computational model is often based on an arbitrary quantitative indicator and therefore convergence is sometimes assumed, when in fact it has not actually occurred. For this project, the convergence of each model was carefully analyzed in order to avoid past mistakes such as stopping the simulation early and assuming convergence. Many quantitative and qualitative indicators were studied to insure that convergence had been achieved and that the entire transient process was simulated and understood.

The animated visualization of velocity and temperature values was the key to discovering that convergence had not been previously reached with the quantitative indicator. Further, the visualization techniques used in this project allowed the user to check that convergence of the velocity magnitudes had been reached, which was not previously possible with the quantitative indicator used.

2. Improvements have been made in the prediction of temperature and heat transfer with this project. The prediction of temperatures is relatively easy and accurate as long as one uses the experimental convection coefficients available. Accurate convection coefficient predictions are not yet possible using the wall temperature boundary conditions implemented in the experiments for this study.
3. Temperature predictions need to be as accurate as possible when modeling buoyancy. One also needs to verify that the temperatures have converged. Questions about convergence were investigated in this project. One should keep in mind that if the temperature field has not converged, then the velocity field has not converged either.

4. This 3-D turbulent code has the capacity to utilize a variable grid for better grid coverage and more accuracy at the inlet and outlet of the model. Using the 2-D error numbers as a gage for the experiments performed in this thesis, the variable grid matched the experimental results somewhat better than the uniform grid at the z=0.127 m. section.

5. For Weathers' uniform grid study, the global error analysis was a valid quantitative technique for directly comparing the accuracy of the models. One purpose of using a variable grid is to improve the prediction of a model near the wall. The slice error number is more sensitive to the error at the wall and is therefore a preferable choice for comparing models and grids.

4.2 Recommendations

1. Ideally, numerous non-uniform grids with different degrees of non-uniformity would have been used in this thesis. For further studies, one should increase the degree of non-uniformity of the grid. Use the 2-dimensional or "slice" error number to gage one's progress. Hold tight constraints on the computational models and show with one case how the degree of grid non-uniformity effects the results. Note the convergence rate to observe whether one converges faster. Once the grid is chosen and proven, expand the experiment to include other model types and airflow rates.
2. Continued investigation of convergence issues might be aimed at improving automatic convergence criteria. Another quantitative indicator could be an inlet jet monitor. The direction of the inlet jet could be calculated at each time step based on the velocity magnitudes. Constraints on how much the direction of the inlet jet is allowed to change might aid in determining convergence. The constraints would have to be designed such that very small changes in the inlet jet direction do not declare convergence, such as during the 70-100 second time period shown in Figure 3.1 and discussed in section 3.2.

3. Expand the analysis to include other inlet and outlet configurations. Furniture or other obstructions could also be modeled.

4. More detailed experimental results would help the computational results in many different respects. More experimental information near the walls is needed. Also detailed experimental results might help to improve boundary conditions, at the outlet for example. The computational analysis could be coordinated with experimental research that will soon be performed in the school of Mechanical and Aerospace Engineering at Oklahoma State University.
5.0 REFERENCES


Finite Difference Equations for U Velocity Components

\[ U_{i,j,k}^{\text{new}} = U_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i+1,j,k}}{\Delta x} + g_x + \text{VISX} - \text{FUX} - \text{FUY} - \text{FUZ} \right] \]

The viscous flux term in expanded finite difference form is given below.

\[ \text{VISX} = \nu \left[ \left( \frac{U_{i+1,j,k} - U_{i,j,k}}{\Delta x_c^+} - \frac{U_{i,j,k} - U_{i-1,j,k}}{\Delta x_c^-} \right)/\Delta x_c \right] + \left( \frac{U_{i,j+1,k} - U_{i,j,k}}{\Delta y_c} - \frac{U_{i,j,k} - U_{i,j-1,k}}{\Delta y_c} \right)/\Delta y_c \]
\[ + \left( \frac{U_{i,j,k+1} - U_{i,j,k}}{\Delta z_c} - \frac{U_{i,j,k} - U_{i,j,k-1}}{\Delta z_c} \right)/\Delta z_c \]

The convective flux terms are as follows.

\[ \text{FUX} = \frac{1}{4} \left\{ [(U_{i,j,k} + U_{i+1,j,k})^2 + \bar{\alpha} \ |U_{i,j,k} + U_{i+1,j,k}|*(U_{i,j,k} - U_{i+1,j,k})]/\Delta x_{ci+1} \right. \]
\[ -[(U_{i-1,j,k} + U_{i,j,k})^2 + \bar{\alpha} \ |U_{i-1,j,k} + U_{i,j,k}|*(U_{i-1,j,k} - U_{i,j,k})]/\Delta x_{ci} \left\} \]

\[ \text{FUY} = \frac{1}{4} \left\{ [(V_{i,j,k} + V_{i+1,j,k})^2 + \bar{\alpha} \ |V_{i,j,k} + V_{i+1,j,k}|*(V_{i,j,k} - V_{i+1,j,k})]/\Delta y_{ji} \right. \]
\[ -[(V_{i-1,j,k} + V_{i,j,k})^2 + \bar{\alpha} \ |V_{i-1,j,k} + V_{i,j,k}|*(V_{i-1,j,k} - V_{i,j,k})]/\Delta y_{ji} \left\} \]

\[ \text{FUZ} = \frac{1}{4} \left\{ [(W_{i,j,k} + W_{i+1,j,k})^2 + \bar{\alpha} \ |W_{i,j,k} + W_{i+1,j,k}|*(W_{i,j,k} - W_{i+1,j,k})]/\Delta z_{kj} \right. \]
\[ -[(W_{i,j,k-1} + W_{i,j,k})^2 + \bar{\alpha} \ |W_{i,j,k-1} + W_{i,j,k}|*(W_{i,j,k-1} - W_{i,j,k})]/\Delta z_{kj} \left\} \]
Finite Difference Equations for V Velocity Components

\[
V_{i,j,k}^{\text{new}} = V_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i,j+1,k}}{\Delta y_j} + g_y + \text{VISY} - \text{FVX} - \text{FVY} - \text{FVZ} \right]
\]

The viscous flux term in expanded finite difference form is given below.

\[
\text{VISY} = \nu \left[ \left( \frac{V_{i+1,j,k} - V_{i,j,k}}{\Delta x_i} - \frac{V_{i,j,k} - V_{i-1,j,k}}{\Delta x_{i+1}} \right) \right] / \Delta x_i
+ \left( \frac{V_{i,j+1,k} - V_{i,j,k}}{\Delta y_{j+1}} - \frac{V_{i,j,k} - V_{i,j-1,k}}{\Delta y_{j}} \right) / \Delta y_j
+ \left( \frac{V_{i,j,k+1} - V_{i,j,k}}{\Delta z_{k+1}} - \frac{V_{i,j,k} - V_{i,j,k-1}}{\Delta z_{k}} \right) / \Delta z_{k}
\]

The convective flux terms are as follows.

\[
\text{FVX} = \frac{1}{4} \left\{ \left[ (U_{i,j,k} + U_{i,j+1,k})^* (V_{i,j,k} + V_{i,j,k+1}) + \bar{\alpha} \ |U_{i,j,k} + U_{i,j+1,k}|^* (V_{i,j,k} - V_{i,j,k+1}) \right] / \Delta x_i 
- \left[ (U_{i-1,j,k} + U_{i-1,j+1,k})^* (V_{i-1,j,k} + V_{i,j,k}) + \bar{\alpha} \ |U_{i-1,j,k} + U_{i-1,j+1,k}|^* (V_{i-1,j,k} - V_{i,j,k}) \right] / \Delta x_{i+1} \right\}
\]

\[
\text{FVY} = \frac{1}{4} \left\{ \left[ (V_{i,j,k} + V_{i,j+1,k})^2 + \bar{\alpha} \ |V_{i,j,k} + V_{i,j+1,k}|^* (V_{i,j,k} - V_{i,j+1,k}) \right] / \Delta y_{j+1} 
- \left[ (V_{i,j-1,k} + V_{i,j,k})^2 + \bar{\alpha} \ |V_{i,j-1,k} + V_{i,j,k}|^* (V_{i,j-1,k} - V_{i,j,k}) \right] / \Delta y_{j} \right\}
\]

\[
\text{FVZ} = \frac{1}{4} \left\{ \left[ (W_{i,j,k} + W_{i,j+1,k})^* (V_{i,j,k} + V_{i,j,k+1}) + \bar{\alpha} \ |W_{i,j,k} + W_{i,j+1,k}|^* (V_{i,j,k} - V_{i,j,k+1}) \right] / \Delta z_{k} 
- \left[ (W_{i,j-1,k} + W_{i,j+1,k})^* (V_{i,j-1,k} + V_{i,j,k}) + \bar{\alpha} \ |W_{i,j-1,k} + W_{i,j+1,k}|^* (V_{i,j-1,k} - V_{i,j,k}) \right] / \Delta z_{k+1} \right\}
\]
Finite Difference Equations for W Velocity Components

\[ W_{i,j,k}^{new} = W_{i,j,k}^{old} + \Delta t \left[ \frac{P_{i,j,k} - P_{i,j,k+1}}{\Delta z_k} + g_z + \text{VISZ} - FW X - FW Y - FW Z \right] \]

The viscous flux term in expanded finite difference form is given below.

\[ \text{VISZ} = \nu \left[ \frac{(W_{i+1,j,k} - W_{i,j,k} - W_{i-1,j,k})}{\Delta x_i} \right] / \Delta x_i \]
\[ + \left( \frac{W_{i,j+1,k} - W_{i,j,k} - W_{i,j-1,k}}{\Delta y_j} \right) / \Delta y_j \]
\[ + \left( \frac{W_{i,j,k+1} - W_{i,j,k} - W_{i,j,k-1}}{\Delta z_{ck+1}} \right) / \Delta z_{ck} \]

The convective flux terms are as follows.

\[ FW X = \frac{1}{4} \left\{ \left[ (U_{i,j,k} + U_{i,j,k+1}) \cdot W_{i,j,k} + W_{i-1,j,k} \cdot \bar{\alpha} \right] * (W_{i,j,k} - W_{i+1,j,k}) \right\} / \Delta x_i \]
\[ - \left[ (U_{i-1,j,k} + U_{i,j,k+1}) \cdot W_{i,j,k} + W_{i,j,k+1} \cdot \bar{\alpha} \right] * (W_{i,j,k} - W_{i,j,k-1}) / \Delta x_{i-1} \}

\[ FW Y = \frac{1}{4} \left\{ \left[ (W_{i,j,k} + W_{i,j,k+1}) \cdot V_{i,j,k} + V_{i,j,k+1} \cdot \bar{\alpha} \right] * (W_{i,j,k} - W_{i,j,k+1}) \right\} / \Delta y_j \]
\[ - \left[ (V_{i,j,k+1} + V_{i,j,k+1}) \cdot W_{i,j,k} + W_{i,j,k+1} \cdot \bar{\alpha} \right] * (V_{i,j,k+1} - W_{i,j,k}) / \Delta y_{j+1} \}

\[ FW Z = \frac{1}{4} \left\{ \left[ (W_{i,j,k} + W_{i,j,k+1}) \cdot W_{i,j,k} + \bar{\alpha} \right] * (W_{i,j,k} - W_{i,j,k+1}) \right\} / \Delta z_{ck+1} \]
\[ - \left[ (W_{i,j,k+1} + W_{i,j,k+1}) \cdot W_{i,j,k+1} \cdot \bar{\alpha} \right] * (W_{i,j,k+1} - W_{i,j,k}) / \Delta z_{ck} \} \]
Finite Difference Turbulent Velocity Equations

\[ U_{ijk}^{\text{new}} = U_{ijk}^{\text{old}} + \Delta t \left[ \frac{P_{ijk} - P_{i+1,j,k}}{\Delta x} + g_x + \text{GAMMAX} - \text{FUX} - \text{FUY} - \text{FUZ} \right] \]

\[ \Gamma_x = \nu_t \left[ 2 \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} + \frac{\partial^2 V}{\partial x \partial y} + \frac{\partial^2 W}{\partial x \partial z} \right] + 2 \frac{\partial \nu_t}{\partial x} \left[ \frac{\partial U}{\partial x} \right] 
+ \frac{\partial \nu_t}{\partial y} \left[ \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right] + \frac{\partial \nu_t}{\partial z} \left[ \frac{\partial U}{\partial z} + \frac{\partial W}{\partial x} \right] - \frac{2 \partial k}{3} \frac{\partial}{\partial x} \]

\[ V_{ij,k}^{\text{new}} = V_{ij,k}^{\text{old}} + \Delta t \left[ \frac{P_{ijk} - P_{i,j+1,k}}{\Delta y} + g_y + \text{GAMMAY} - \text{FVX} - \text{FY} - \text{FVZ} \right] \]

\[ \Gamma_y = \nu_t \left[ 2 \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial z^2} + \frac{\partial^2 U}{\partial x \partial y} + \frac{\partial^2 W}{\partial x \partial z} \right] + 2 \frac{\partial \nu_t}{\partial y} \left[ \frac{\partial V}{\partial y} \right] 
+ \frac{\partial \nu_t}{\partial x} \left[ \frac{\partial V}{\partial x} + \frac{\partial U}{\partial y} \right] + \frac{\partial \nu_t}{\partial z} \left[ \frac{\partial V}{\partial z} + \frac{\partial W}{\partial y} \right] - \frac{2 \partial k}{3} \frac{\partial}{\partial y} \]

\[ W_{ij,k}^{\text{new}} = W_{ij,k}^{\text{old}} + \Delta t \left[ \frac{P_{ijk} - P_{i,j,k+1}}{\Delta z} + g_z + \text{GAMMAZ} - \text{FWX} - \text{FWY} - \text{FWZ} \right] \]

\[ \Gamma_z = \nu_t \left[ 2 \frac{\partial^2 W}{\partial z^2} + \frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial y^2} + \frac{\partial^2 U}{\partial x \partial z} + \frac{\partial^2 V}{\partial x \partial y} \right] + 2 \frac{\partial \nu_t}{\partial z} \left[ \frac{\partial W}{\partial z} \right] 
+ \frac{\partial \nu_t}{\partial x} \left[ \frac{\partial W}{\partial x} + \frac{\partial U}{\partial z} \right] + \frac{\partial \nu_t}{\partial y} \left[ \frac{\partial W}{\partial y} + \frac{\partial V}{\partial z} \right] - \frac{2 \partial k}{3} \frac{\partial}{\partial z} \]
Finite Difference Mixture Fraction Equations

\[ S_{i,j,k}^{\text{new}} = S_{i,j,k}^{\text{old}} + \Delta t \left[ \text{VISS} - \text{FSX} - \text{FSY} - \text{FSZ} \right] \]

\[ \text{VISS} = \alpha \left[ \left( \frac{S_{i+1,j,k} - S_{i,j,k}}{\Delta x_i} - \frac{S_{i,j,k} - S_{i-1,j,k}}{\Delta x_{i-1}} \right) / \Delta x_i \right] + \left( \frac{S_{i,j+1,k} - S_{i,j,k}}{\Delta y_j} - \frac{S_{i,j,k} - S_{i,j-1,k}}{\Delta y_{j-1}} \right) / \Delta y_j + \left( \frac{S_{i,j,k+1} - S_{i,j,k}}{\Delta z_k} - \frac{S_{i,j,k} - S_{i,j,k-1}}{\Delta z_{k-1}} \right) / \Delta z_k \right] \]

\[ \text{FSX} = \frac{1}{2} \left\{ \left[ U_{i,j,k}(S_{i,j,k} + S_{i+1,j,k}) + \bar{\alpha} |U_{i,j,k}|(S_{i,j,k} - S_{i+1,j,k}) \right] / \Delta x_i \right\} - \left\{ \left[ U_{i-1,j,k}(S_{i-1,j,k} + S_{i,j,k}) + \bar{\alpha} |U_{i-1,j,k}|(S_{i-1,j,k} - S_{i,j,k}) \right] / \Delta x_{i-1} \right\} \]

\[ \text{FSY} = \frac{1}{2} \left\{ \left[ V_{i,j,k}(S_{i,j,k} + S_{i,j+1,k}) + \bar{\alpha} |V_{i,j,k}|(S_{i,j,k} - S_{i,j+1,k}) \right] / \Delta y_j \right\} - \left\{ \left[ V_{i,j-1,k}(S_{i,j-1,k} + S_{i,j,k}) + \bar{\alpha} |V_{i,j-1,k}|(S_{i,j-1,k} - S_{i,j,k}) \right] / \Delta y_{j-1} \right\} \]

\[ \text{FSZ} = \frac{1}{2} \left\{ \left[ W_{i,j,k}(S_{i,j,k} + S_{i,j,k+1}) + \bar{\alpha} |W_{i,j,k}|(S_{i,j,k} - S_{i,j,k+1}) \right] / \Delta z_k \right\} - \left\{ \left[ W_{i,j,k-1}(S_{i,j,k-1} + S_{i,j,k}) + \bar{\alpha} |W_{i,j,k-1}|(S_{i,j,k-1} - S_{i,j,k}) \right] / \Delta z_{k-1} \right\} \]
Finite Difference Convective Flux Equations

\[
FKX = \frac{1}{2} \left\{ \left[ U_{i,j,k}(k_{i,j,k} + k_{i+1,j,k}) + \bar{\alpha} \left| U_{i,j,k} \right| (k_{i,j,k} - k_{i+1,j,k}) \right] / \Delta x_i - \left[ (U_{i-1,j,k}(k_{i-1,j,k} + k_{i,j,k}) + \bar{\alpha} \left| U_{i-1,j,k} \right| (k_{i-1,j,k} - k_{i,j,k}) \right] / \Delta x_{i-1} \right\}
\]

\[
FKY = \frac{1}{2} \left\{ \left[ V_{i,j,k}(k_{i,j,k} + k_{i,j+1,k}) + \bar{\alpha} \left| V_{i,j,k} \right| (k_{i,j,k} - k_{i,j+1,k}) \right] / \Delta y_j - \left[ (V_{i,j-1,k}(k_{i,j-1,k} + k_{i,j,k}) + \bar{\alpha} \left| V_{i,j-1,k} \right| (k_{i,j-1,k} - k_{i,j,k}) \right] / \Delta y_{j-1} \right\}
\]

\[
FKZ = \frac{1}{2} \left\{ \left[ W_{i,j,k}(k_{i,j,k} + k_{i,j,k+1}) + \bar{\alpha} \left| W_{i,j,k} \right| (k_{i,j,k} - k_{i,j,k+1}) \right] / \Delta z_k - \left[ (W_{i,j,k-1}(k_{i,j,k-1} + k_{i,j,k}) + \bar{\alpha} \left| W_{i,j,k-1} \right| (k_{i,j,k-1} - k_{i,j,k}) \right] / \Delta z_{k-1} \right\}
\]

\[
FEX = \frac{1}{2} \left\{ \left[ U_{i,j,k}(\epsilon_{i,j,k} + \epsilon_{i+1,j,k}) + \bar{\alpha} \left| U_{i,j,k} \right| (\epsilon_{i,j,k} - \epsilon_{i-1,j,k}) \right] / \Delta x_i - \left[ (U_{i-1,j,k}(\epsilon_{i-1,j,k} + \epsilon_{i,j,k}) + \bar{\alpha} \left| U_{i-1,j,k} \right| (\epsilon_{i-1,j,k} - \epsilon_{i,j,k}) \right] / \Delta x_{i-1} \right\}
\]

\[
FEY = \frac{1}{2} \left\{ \left[ V_{i,j,k}(\epsilon_{i,j,k} + \epsilon_{i,j+1,k}) + \bar{\alpha} \left| V_{i,j,k} \right| (\epsilon_{i,j,k} - \epsilon_{i,j+1,k}) \right] / \Delta y_j - \left[ (V_{i,j-1,k}(\epsilon_{i,j-1,k} + \epsilon_{i,j,k}) + \bar{\alpha} \left| V_{i,j-1,k} \right| (\epsilon_{i,j-1,k} - \epsilon_{i,j,k}) \right] / \Delta y_{j-1} \right\}
\]

\[
FEZ = \frac{1}{2} \left\{ \left[ W_{i,j,k}(\epsilon_{i,j,k} + \epsilon_{i,j,k+1}) + \bar{\alpha} \left| W_{i,j,k} \right| (\epsilon_{i,j,k} - \epsilon_{i,j,k+1}) \right] / \Delta z_k - \left[ (W_{i,j,k-1}(\epsilon_{i,j,k-1} + \epsilon_{i,j,k}) + \bar{\alpha} \left| W_{i,j,k-1} \right| (\epsilon_{i,j,k-1} - \epsilon_{i,j,k}) \right] / \Delta z_{k-1} \right\}
\]
Finite Difference Equations for Theta Function and Pi

\[ \Theta(k) = \frac{\nu}{\sigma_k} \left( \frac{\partial^2 k}{\partial x^2} + \frac{\partial^2 k}{\partial y^2} + \frac{\partial^2 k}{\partial z^2} \right) + \frac{1}{\sigma_k} \left( \frac{\partial \nu \cdot \partial k}{\partial x \partial x} + \frac{\partial \nu \cdot \partial k}{\partial y \partial y} + \frac{\partial \nu \cdot \partial k}{\partial z \partial z} \right) \]

\[ \Theta(\varepsilon) = \frac{\nu}{\sigma_\varepsilon} \left( \frac{\partial^2 \varepsilon}{\partial x^2} + \frac{\partial^2 \varepsilon}{\partial y^2} + \frac{\partial^2 \varepsilon}{\partial z^2} \right) + \frac{1}{\sigma_\varepsilon} \left( \frac{\partial \nu \cdot \partial \varepsilon}{\partial x \partial x} + \frac{\partial \nu \cdot \partial \varepsilon}{\partial y \partial y} + \frac{\partial \nu \cdot \partial \varepsilon}{\partial z \partial z} \right) \]

\[ \Pi = \frac{2}{\sigma_\varepsilon} \left( \frac{\partial U}{\partial x} \frac{\partial U}{\partial y} + \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} + \frac{\partial V}{\partial x} \frac{\partial V}{\partial y} + \frac{\partial V}{\partial z} \frac{\partial V}{\partial z} + \frac{\partial W}{\partial x} \frac{\partial W}{\partial y} + \frac{\partial W}{\partial z} \frac{\partial W}{\partial z} \right) \]
USER'S GUIDE FOR FORTRAN CODE:

TFLOVG.F

( Turbulent FLOw with Variable Grid )

By

MICHAEL RAMEY
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2.0 Nomenclature</td>
<td>3</td>
</tr>
<tr>
<td>3.0 Geometry &amp; Background</td>
<td>6</td>
</tr>
<tr>
<td>4.0 FORTRAN Code Solution Procedure</td>
<td>6</td>
</tr>
<tr>
<td>5.0 FORTRAN Source Code: TFLOVG.F</td>
<td>8</td>
</tr>
<tr>
<td>6.0 Input Files for FORTRAN Source Code</td>
<td>44</td>
</tr>
<tr>
<td>6.1 Input File: grid.dat</td>
<td></td>
</tr>
<tr>
<td>6.2 Input File: room.dat</td>
<td></td>
</tr>
<tr>
<td>7.0 Description of Useful Modifications</td>
<td>49</td>
</tr>
<tr>
<td>7.1 How To Change the Model Type</td>
<td>49</td>
</tr>
<tr>
<td>7.2 How To Change the Air Flow Rate</td>
<td>49</td>
</tr>
<tr>
<td>7.3 How To Change the Location &amp; Size of the Inlet</td>
<td>50</td>
</tr>
<tr>
<td>7.4 How To Change the Location &amp; Size of the Outlet</td>
<td>51</td>
</tr>
<tr>
<td>7.5 How To Model Buoyancy</td>
<td>53</td>
</tr>
<tr>
<td>8.0 How To Run the Code</td>
<td>54</td>
</tr>
<tr>
<td>9.0 Summary of Equations</td>
<td>56</td>
</tr>
<tr>
<td>9.1 Finite Difference Equations for U Velocity Components</td>
<td>57</td>
</tr>
<tr>
<td>9.2 Finite Difference Equations for V Velocity Components</td>
<td>58</td>
</tr>
<tr>
<td>9.3 Finite Difference Equations for W Velocity Components</td>
<td>59</td>
</tr>
<tr>
<td>9.4 Finite Difference Turbulent Velocity Equations</td>
<td>60</td>
</tr>
<tr>
<td>9.5 Finite Difference Temperature Equations</td>
<td>61</td>
</tr>
<tr>
<td>9.6 k Finite Difference Equations</td>
<td>62</td>
</tr>
<tr>
<td>9.7 ε Finite Difference Equations</td>
<td>63</td>
</tr>
<tr>
<td>9.8 Finite Difference Equations for Theta Function and Pi</td>
<td>64</td>
</tr>
</tbody>
</table>
1.0 Introduction

This manual is to be used as an aid when using the FORTRAN program TFLOVG.F. Section 2.0 lists symbols, descriptions of those symbols and variable names used in the FORTRAN source code. Section 3.0 briefly describes the FORTRAN code solution procedure. A flowchart of the code is included. Section 4.0 gives a listing of the code itself. Section 5.0 presents the input files to TFLOVG.F, 'room.dat' & 'grid.dat'. Section 6.0 describes how to make changes to the input files to make the code more versatile. Section 7.0 gives the user some information about how to run the code on the rs6000 computers. Finally, section 8.0 lists important equations.

2.0 Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFLOVG.F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{in}$</td>
<td>Area of Inlet</td>
<td>AREAIN</td>
</tr>
<tr>
<td>$A_{out}$</td>
<td>Area of Outlet</td>
<td>AREAOUT</td>
</tr>
<tr>
<td>ACH</td>
<td>Air Changes Per Hour</td>
<td>ACH</td>
</tr>
<tr>
<td>$C_1$</td>
<td>Constant Coefficient in k-ε Model</td>
<td>TCFC1</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>$C_2$</td>
<td>Constant Coefficient in k-ε Model</td>
<td>TCFC2</td>
</tr>
<tr>
<td>$C_\mu$</td>
<td>Constant Coefficient in k-ε Model</td>
<td>TCFCMU</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Specific Heat</td>
<td>QCP</td>
</tr>
<tr>
<td>$D$</td>
<td>Mass Divergence</td>
<td>D</td>
</tr>
<tr>
<td>$F_1$</td>
<td>Empirical Function in k-ε Model</td>
<td>TFN1</td>
</tr>
<tr>
<td>$F_2$</td>
<td>Empirical Function in k-ε Model</td>
<td>TFN2</td>
</tr>
<tr>
<td>$F_\mu$</td>
<td>Empirical Function in k-ε Model</td>
<td>TFNMU</td>
</tr>
<tr>
<td>$g_x$</td>
<td>Gravity in X-Direction</td>
<td>GX</td>
</tr>
<tr>
<td>$g_y$</td>
<td>Gravity in Y-Direction</td>
<td>GY</td>
</tr>
<tr>
<td>$g_z$</td>
<td>Gravity in Z-Direction</td>
<td>GZ</td>
</tr>
<tr>
<td>GEN</td>
<td>Global Velocity Magnitude Error Number</td>
<td>GENV</td>
</tr>
<tr>
<td>GENT</td>
<td>Global Temperature Error Number</td>
<td>GENT</td>
</tr>
<tr>
<td>$h_{\text{ceiling}}$</td>
<td>Film Coefficient for the Ceiling</td>
<td>FILMC</td>
</tr>
<tr>
<td>$h_{\text{floor}}$</td>
<td>Film Coefficient for the Floor</td>
<td>FILMF</td>
</tr>
<tr>
<td>$h_{\text{wall}}$</td>
<td>Film Coefficient for the Wall</td>
<td>FILMW</td>
</tr>
<tr>
<td>$k$</td>
<td>Turbulent Kinetic Energy</td>
<td>TK &amp; TKN</td>
</tr>
<tr>
<td>$k_{\text{in}}$</td>
<td>Turbulent Kinetic Energy at Inlet</td>
<td>TKin</td>
</tr>
<tr>
<td>$P$</td>
<td>Pressure</td>
<td>P</td>
</tr>
<tr>
<td>$q''_c$</td>
<td>Convective Heat Flux</td>
<td>QAVG</td>
</tr>
<tr>
<td>$R_y$</td>
<td>Local Reynolds number</td>
<td>REYNY</td>
</tr>
<tr>
<td>$R_T$</td>
<td>Turbulent Reynolds number</td>
<td>REYNT</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
<td>T &amp; TN</td>
</tr>
<tr>
<td>$T_0$, $T_r$</td>
<td>Reference Temperatures</td>
<td>TREF</td>
</tr>
<tr>
<td>$T_{\text{wall}}$</td>
<td>Wall Temperature</td>
<td>QTWALL</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>t</td>
<td>Time</td>
<td>TIME</td>
</tr>
<tr>
<td>U</td>
<td>Velocity Components in X-Direction</td>
<td>U &amp; UN</td>
</tr>
<tr>
<td>V</td>
<td>Velocity Components in Y-Direction</td>
<td>V &amp; VN</td>
</tr>
<tr>
<td>W</td>
<td>Velocity Components in Z-Direction</td>
<td>W &amp; WN</td>
</tr>
<tr>
<td>U_{in}</td>
<td>Inlet Velocity For Each Cell</td>
<td>UINS()</td>
</tr>
<tr>
<td>U_{jet}</td>
<td>Inlet Jet Velocity</td>
<td>UJET</td>
</tr>
<tr>
<td>U_{out}</td>
<td>Outlet Velocity</td>
<td>UJOUT</td>
</tr>
<tr>
<td>\alpha</td>
<td>Thermal Diffusivity</td>
<td>TDIFUS</td>
</tr>
<tr>
<td>\bar{\alpha}</td>
<td>Donor Cell Coefficient</td>
<td>ALPHA</td>
</tr>
<tr>
<td>\beta</td>
<td>Thermal Expansion Coefficient</td>
<td>BUOBETA</td>
</tr>
<tr>
<td>\bar{\beta}</td>
<td>Pressure Divisor</td>
<td>BETA</td>
</tr>
<tr>
<td>\Delta x_i</td>
<td>Distance Between Scalar Quantities (X-Direction)</td>
<td>DX</td>
</tr>
<tr>
<td>\Delta y_j</td>
<td>Distance Between Scalar Quantities (Y-Direction)</td>
<td>DY</td>
</tr>
<tr>
<td>\Delta z_k</td>
<td>Distance Between Scalar Quantities (Z-Direction)</td>
<td>DZ</td>
</tr>
<tr>
<td>\Delta x_c_i</td>
<td>Width of Cell in the X-Direction</td>
<td>DXM</td>
</tr>
<tr>
<td>\Delta y_c_j</td>
<td>Width of Cell in the Y-Direction</td>
<td>DYM</td>
</tr>
<tr>
<td>\Delta z_c_k</td>
<td>Width of Cell in the Z-Direction</td>
<td>DZM</td>
</tr>
<tr>
<td>\Delta t</td>
<td>Time Step</td>
<td>DELT</td>
</tr>
<tr>
<td>\varepsilon</td>
<td>Turbulent Energy Dissipation Rate</td>
<td>TE &amp; TEN</td>
</tr>
<tr>
<td>\varepsilon_{in}</td>
<td>Turbulent Energy Dissipation Rate at Inlet</td>
<td>TEIN</td>
</tr>
<tr>
<td>\Gamma_x</td>
<td>Turbulent Function</td>
<td>GAMX</td>
</tr>
<tr>
<td>\Gamma_y</td>
<td>Turbulent Function</td>
<td>GAMY</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>$\Gamma_z$</td>
<td>Turbulent Function</td>
<td>GAMZ</td>
</tr>
<tr>
<td>$\Pi$</td>
<td>Turbulent Term in $k$ and $\varepsilon$ Equations</td>
<td>PI</td>
</tr>
<tr>
<td>$\Theta_k$</td>
<td>Turbulent Function in $k$ Equation</td>
<td>THETAK</td>
</tr>
<tr>
<td>$\Theta_\varepsilon$</td>
<td>Turbulent Function in $\varepsilon$ Equation</td>
<td>THETAE</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
<td>QDENS</td>
</tr>
<tr>
<td>$v$</td>
<td>Kinematic Viscosity</td>
<td>QVISC &amp; TVIS</td>
</tr>
<tr>
<td>$v_t$</td>
<td>Turbulent Viscosity</td>
<td>TVIS</td>
</tr>
</tbody>
</table>
3.0  Geometry & Background

The TFLOVG.F FORTRAN code is currently set up to model room airflow. The size of the room is 4.6 m x 2.75 m x 2.75 m in the x, y and z-directions, respectively. Figure 1 shows the room geometry. The origin of the model is at the bottom, left, front of the room. The origin is designated by the black dot in Figure 1. The inlet is included in the wall at x=0 (left wall in Figure 1) and the inlet air moves across the z=0 plane (front wall in Figure 1).
The Marker-and-Cell methodology is used in the TFLOVG.F program. Therefore, the scalar quantities are located at the cell centers, and the velocity vectors are located at the cell faces. A portion of the x-y plane is shown in Figure 2 below. The other dimensions are similar. Figure 2 shows the cell centers, the velocity vectors at the cell faces, and the distances between each of the like quantities in the x and y-directions. The distances between the cell centers are designated as DX(), DY(), and DZ(). The distances between the cell faces are designated as DXM(), DYM(), and DZM(). The DXM(1), DYM(1) and DZM(1) distances are never used. Also, the DX(IMAX), DY(JMAX) and DZM(KMAX) distances are never used.
Note: All Scalar Quantities Are Defined At Cell Centers

**Figure 2** Location of Scalars, Vectors, and Distances

- **DX** = Δx
- **DXM** = Δx_c
- **DY** = Δy
- **DYM** = Δy_c
4.0 FORTRAN Code Solution Procedure

Details of the approach used to solve for the velocities and scalar quantities are included in this section. First the geometry, grid spacing, initial conditions, some boundary specifications, physical properties, and other necessary information are read into the program. As stated in the previous section, the FORTRAN code uses the Marker-and-Cell methodology. The program calculates additional geometric and numerical parameters such as the volume flow rate, the number of air changes per hour (ACH), and the areas of the inlet and outlet. After initial values and boundary conditions are imposed, time advanced velocity components, temperature, mixture fraction, turbulent energy and turbulent dissipation values are calculated for the entire 3-D domain. Boundary conditions are updated after the determination of time advanced values.

Next, the continuity equation is satisfied using the change in pressure as a correction factor for the time advanced velocity components. After continuity is satisfied, residual terms for the velocities and mixture fraction are calculated based on the amount of change in the values from the previous time step.

At this point in the solution process, a check for convergence is performed. If the values have not converged, then another time step must be performed. So, the time advanced values are shifted to the old memory locations, and time is advanced one step. Then, new time advanced values can be calculated. This process continues until convergence has been achieved.
Figure 2 is a flowchart which summarizes the solution methodology of the finite difference code used in this study.
Read in Grid Spacing
Read in All Other Input Values
Set Initial Conditions
Set Boundary Conditions
Determine Time Advanced Values
Update Boundary Conditions
Calculate Cell Divergence
Calculate Change in Pressure
Correct Time Advanced Values with Pressure Change
If No, Do Another Iteration
Check Divergence: Has Continuity Been Satisfied?
If Yes
Update Boundary Conditions
Advance a Time Step
Calculate Residual Terms
Check for Convergence: Is the Error Value Acceptable?
If Yes
Print Specified Outputs
End Program
5.0 FORTRAN Source Code: TFLOVG.F

*******************************************************************************
**********
** PROGRAM: TFLOVG.F
**
** TURBULENT FLOW WITH A VARIABLE GRID
**
** = === = =
**
**
** DEVELOPER: MIKE RAMEY
**
**
** PURPOSE: CALCULATION OF 3-D TRANSIENT, TURBULENT, BUOYANT **
** FLOW WITHIN A ROOM UTILIZING A VARIABLE GRID **
**
** REFERENCE: GE3DF.FOR - A 3-D TRANSIENT FLOW PROGRAM **
** BY D.G. LILLEY **
** TURBFLO.F - TURBULENT, BUOYANT ADDITIONS TO GE3DF.F **
** BY JIM WEATHERS **

*******************************************************************************
**********
*** 1.0 PRELIMINARIES & DEFINITION OF VARIABLES
***
************************************************************************************
************
IMPLICIT REAL*8 (A-H,O-Z)
************************************************************************************ COMMON
BLOCK ***
COMMON
**************************************************************************************
************

** /GEOM/ --GEOMETRY
** IMAX = NUMBER OF CELLS IN X-DIRECTION
** JMAX = NUMBER OF CELLS IN Y-DIRECTION
** KMAX = NUMBER OF CELLS IN Z-DIRECTION
** IMM1 = NUMBER OF CELLS IN X-DIRECTION - 1
** JMM1 = NUMBER OF CELLS IN Y-DIRECTION - 1
** KMAX = NUMBER OF CELLS IN Z-DIRECTION - 1
** X = DISTANCE IN X-DIRECTION FROM ORIGIN TO CELL CENTER
** XM = DISTANCE IN X-DIRECTION FROM ORIGIN TO CELL WALL
** DX = DISTANCE BETWEEN CELL CENTERS IN THE X-DIRECTION
** DXM = DISTANCE BETWEEN CELL WALLS IN THE X-DIR (CELL WIDTHS)
** Y = DISTANCE IN Y-DIRECTION FROM ORIGIN TO CELL CENTER
** YM = DISTANCE IN Y-DIRECTION FROM ORIGIN TO CELL WALL
** DY = DISTANCE BETWEEN CELL CENTERS IN THE Y-DIRECTION
** DYM = DISTANCE BETWEEN CELL WALLS IN THE Y-DIR (CELL WIDTHS)
** Z = DISTANCE IN Z-DIRECTION FROM ORIGIN TO CELL CENTER
** ZM = DISTANCE IN Z-DIRECTION FROM ORIGIN TO CELL WALL
** DZ = DISTANCE BETWEEN CELL CENTERS IN THE Z-DIRECTION
** DZM = DISTANCE BETWEEN CELL WALLS IN THE Z-DIR (CELL WIDTHS)
**/GEOM/ IMAX, IMM1, JMAX, JMM1, KMAX, KMM1, DX(50), DXM(50),
X(50), Y(50), Z(50), XM(50), YM(50), ZM(50),
DY(50), DZ(50), DYM(50), DZM(50),
XDIMEN, YDIMEN, ZDIMEN

**********************************************************
************
** /VELS/ --VELOCITIES
** U = X-DIRECTION VELOCITY
** V = Y-DIRECTION VELOCITY
** W = Z-DIRECTION VELOCITY
** UN = X-DIRECTION VELOCITY AT PREVIOUS TIME STEP
** VN = Y-DIRECTION VELOCITY AT PREVIOUS TIME STEP
** WN = Z-DIRECTION VELOCITY AT PREVIOUS TIME STEP

1/VELS/U(50,50,50), V(50,50,50), W(50,50,50), UN(50,50,50),
2 VN(50,50,50), WN(50,50,50)

**********************************************************
************
** /MIX/ --MIXTURE CONCENTRATION ARRAYS
** S = MIXTURE CONCENTRATION
** SN = MIXTURE CONCENTRATION AT PREVIOUS TIME STEP
1/MIX/S(50,50,50), SN(50,50,50)

**********************************************************
************
** /NBCS/ --WALL BOUNDARY CONDITION SPECIFICATIONS
** NSI1 = B. C. SPECIFICATION AT THE I=1 WALL
** NSIMAX= B. C. SPECIFICATION AT THE I=IMAX WALL
** NSJ1 = B. C. SPECIFICATION AT THE J=1 WALL
** NSJMAX= B. C. SPECIFICATION AT THE J=JMAX WALL
** NSK1 = B. C. SPECIFICATION AT THE K=1 WALL
** NSKMAX= B. C. SPECIFICATION AT THE K=KMAX WALL
**(IF /NBCS/ VALUES ARE SET TO 0, THEN NO SLIP AT THE WALL.)
**(IF /NBCS/ VALUES ARE SET TO 1, THEN FREE SLIP AT THE WALL.)
1/NBCS/NSI1, NSIMAX, NSJ1, NSJMAX, NSK1, NSKMAX

**********************************************************
************
** /INNOUT/ --INLET AND OUTLET CELLS (NUMBER AND LOCATION)  
**  NINCELS = TOTAL NUMBER OF INLET CELLS  
**  NOUTCELS= TOTAL NUMBER OF OUTLET CELLS  
**  JIN() = CELL NUMBERS OF Y-DIRECTION INLET CELLS  
**  KIN() = CELL NUMBERS OF Z-DIRECTION INLET CELLS  
**  JOUT() = CELL NUMBERS OF Y-DIRECTION OUTLET CELLS  
**  KOUT() = CELL NUMBERS OF Z-DIRECTION OUTLET CELLS  

1/INNOUT/NINCELS,NOUTCELS,JIN(90),KIN(90),JOUT(90),KOUT(90)  

************************************************************  
************  
** /TREFCEL/ --CELL NUMBERS USED TO CALCULATE THE OUTLET  
**  TEMPERATURE.  
**  JTREF1, JTREF2 = OUTLET CELL NUMBERS IN THE Y-  
**  DIRECTION  
**  KTREF1, KTREF2 = OUTLET CELL NUMBERS IN THE Z-  
**  DIRECTION  
1/TREFCEL/JTREF1,JTREF2,KTREF1,KTREF2  

************************************************************  
************  
** /JETS/ --JETS, ACH, AND INLET/OUTLET VALUES  

1/JETS/ACH,UJET,UJOUT,UINS(90),JOUTTEMP(20),KOUTTEMP(20)  

************************************************************  
************  
** /RESIDUALS/ --RESIDUAL VALUES (USED IN RESID  
**  SUBROUTINE)  
**  RESU, RESUN, RESU0 = RESIDUAL U-VELOCITY VALUES  
**  RESV, RESVN, RESV0 = RESIDUAL V-VELOCITY VALUES  
**  RESW, RESWN, RESW0 = RESIDUAL W-VELOCITY VALUES  
**  RESS, RESSN, RESS0 = RESIDUAL MIXTURE FRACTION VALUES  
**  DSUMN, DSUM0 = RESIDUAL MASS DIVERGENCE VALUES  

1/RESIDUALS/RESU,RESV,RESW,RESS,DSUMN,RESUN,RESVN,RESWN,RESSN,  
2           DSUM0,RESU0,RESV0,RESS0
**********************************************************
************
** /TEMPS/ --TEMPERATURES
** T, TN = TEMPERATURE & TEMPERATURE AT PREVIOUS TIME
STEP
** QT0, QTIN= INITIAL TEMPERATURE & INLET TEMPERATURE
** QTWALL, QTCOND= WALL TEMPERATURE & THERMAL
CONDUCTIVITY
1/TEMPS/T(50,50,50),TN(50,50,50),QT0,QTIN,QTWALL,QTCOND
**********************************************************
************
** /TURB/ --TURBULENT VARIABLES
** TK, TKN= TURBULENT KINETIC ENERGY (PRESENT AND
PREVIOUS TIME STEP)
** TE, TEN= TURBULENT DISSIPATION (PRESENT AND PREVIOUS
TIME STEP)
** TVIS= TURBULENT VISCOSITY
1/TURB/TK(50,50,50),TKN(50,50,50),TE(50,50,50),TEN(50,50,5
0),
2 TVIS(50,50,50)
**********************************************************
************
** /TCOEFF/ --TURBULENT CONSTANT COEFFICIENTS
** TCFCMU = C SUB MU
** TCFC1 = C SUB 1
** TCFC2 = C SUB 2
** TSIGK = SIGMA SUB k
** TSIGE = SIGMA SUB EPSILON
1/TCOEFF/TCFCMU,TCFC1,TCFC2,TSIGK,TSIGE
**********************************************************
************
** /DTRMS/ --TURBULENT FUNCTIONS PI AND GAMMA
** (CALCULATED IN SECTION 5.0 OF DERIVS
SUBROUTINE)
** PI = PI FUNCTION
** GAMX, GAMY, GAMZ = GAMMAx, GAMMAy, & GAMMAz
1/DTRMS/PI(50,50,50),GAMX(50,50,50),GAMY(50,50,50),GAMZ(50,50,50)

**********************************************************
************
** /TURBBC/ --TURBULENT k and EPSILON INLET BOUNDARY
CONDITIONS
  1/TURBBC/TKIN(90),TEIN(90)

**********************************************************
************
** /TURBS/ --TURBULENT EMPIRICAL FUNCTIONS (IN CALFUNCT
SUBROUTINE)
** REYNT = TURBULENT REYNOLDS NUMBER
** TFNMU = EMPIRICAL FUNCTION F SUB MU
** TFN1 = EMPIRICAL FUNCTION F SUB 1
** TFN2 = EMPIRICAL FUNCTION F SUB 2

1/TURBS/REYNT(50,50,50),TFNMU(50,50,50),TFN1(50,50,50),
  2 TFN2(50,50,50)

**********************************************************
************
** /FILM/ --CONVECTION COEFFICIENTS FOR ROOM SURFACES
** FILMW, FILMF, FILMC= CONV. COEFS. FOR THE WALLS,
FLOOR, & CEILING
** NTEMP = TEMPERATURE BOUNDARY CONDITION TYPE: 1, 2, OR
3
  **(IF NTEMP=1, THEN TEMPERATURE JUST OUTSIDE THE WALL= THE
WALL TEMP.)
  **(IF NTEMP=2, THEN THE TEMP. GRADIENT ACROSS THE WALL IS
SET SUCH
  ** THAT THE TEMP. AT THE WALL IS QTWALL)
  **(IF NTEMP=3, THEN EXPERIMENTAL COEFS. FILMW, FILMF &
FILMC ARE USED)
  1/FILM/FILMW,FILMF,FILMC,NTEMP

**********************************************************
************

********************************************************************
STRING VARIABLE
DEFINITIONS ***
CHARACTER*1 LET(2)
CHARACTER*11 PLABL(8),VTRANS(160),TTRANS(160)
CHARACTER*20

FNAMEOUT,FNAMEVEC,SYSLAT(20),SYSLON(20),FILMDAT
CHARACTER*20

HDLAT(20),HDFLON(20),HDF3D(10),EXPFILE,TEXPFILE
CHARACTER*40 FLOWTYPE(3)
CHARACTER*50 TITLE,COMMENTS

****************************************** DIMENSION
STATEMENTS ***
DIMENSION
P(50,50,50),NOPT(10,3),F1(50,50),F2(50,50),F3(50,50,50)
DIMENSION
NLATI(10),NLONGK(10),FLUX(50,50),FLUX2(50,50)
DIMENSION DZRO(50,50),BETA(50,50,50),INTRVLS(160)
DIMENSION
VZ127(50,50),TZ127(50,50),XTMP(50),YTMP(50),ZTMP(50)

********************************************** DATA
STATEMENTS ***
DATA EPSI,OMG,ALPHA/1.E-6,1.7,1.0/
DATA NOLATSP,NOLONGSP,NOPROFPL,NPLOT,NPTS
/1,1,0,0,0/
DATA ERO1 /1.0E6/
DATA ERRMAX /1.0E-5/
DATA LET /'N','Y' /
DATA PLABL /'U-FIELD ','V-FIELD ','W-FIELD
', 'VEL MAG ',
c'S-FIELD ','T-FIELD ','K-FIELD ','Ep-FIELD '/

****************************************** FLOWTYPE
DEFINITION ***
FLOWTYPE(1)= 'LAMINAR
FLOW'
FLOWTYPE(2)= 'TURBULENT FLOW - WALL
FUNCTIONS'
FLOWTYPE(3)= 'TURBULENT FLOW - LOW-REYNOLDS
NUMBER'

*************
*** 1.1 OBTAIN PARAMETERS FROM INPUT FILE ***

**********************************************************
************
**********************************************************
************
** XDIMEN,YDIMEN,ZDIMEN = ROOM DIMENSIONS IN THE X, Y & Z DIRECTIONS **
OPEN(UNIT=1,FILE='grid.dat')
READ(1,*) XDIMEN,YDIMEN,ZDIMEN
READ(1,*) IMAX,JMAX,KMAX
   DO 20 I=1,IMAX
   20 READ(1,*) X(I),XM(I),DX(I),DXM(I)
   WRITE(*,99) I,XM(I),DXM(I),X(I),DX(I)
   DO 21 J=1,JMAX
   21 READ(1,*) Y(J),YM(J),DY(J),DYM(J)
   WRITE(*,99) J,YM(J),DYM(J),Y(J),DY(J)
   DO 22 K=1,KMAX
   22 READ(1,*) Z(K),ZM(K),DZ(K),DZM(K)
   WRITE(*,23) K,ZM(K),DZM(K),Z(K),DZ(K)
CLOSE(1)
   C 23 FORMAT(1X,I5,2X,4(E10.4))

**********************************************************
************
25 FORMAT(A20)
26 FORMAT(A50)
27 FORMAT(I5,2X,2(A11))
OPEN(UNIT=3,FILE='room.dat')

** TITLE = TITLE GIVEN TO THE SIMULATION (50 CHARACTERS MAX) **
READ(3,26) TITLE

** NFLOW = FLOW TYPE: 1) LAMINAR 2) TURB WALL FUNCT 3) TURB LOW-RE # **
READ(3,*) NFLOW,NTEMP

** COMMENTS - USED AS A COMMENT OR TO SEPARATE DATA **
READ(3,26) COMMENTS
READ(3,*) NINCELS
DO 30 N=1,NINCELS
30 READ(3,*) JIN(N),KIN(N),UINS(N),TKIN(N),TEIN(N)

** NOUTJ,NOUTK = NUMBER OF OUTLET CELLS IN THE Y and Z DIRECTIONS  **
READ(3,*) NOUTJ, NOUTK
DO 32 N=1,NOUTJ
32 READ(3,*) JOUTTEMP(N)
DO 34 N=1,NOUTK
34 READ(3,*) KOUTTEMP(N)
READ(3,26) COMMENTS
READ(3,*) JTREF1
READ(3,*) JTREF2
READ(3,*) KTREF1
READ(3,*) KTREF2
READ(3,26) COMMENTS

** QDENS,QVISC = DENSITY & KINEMATIC VISCOSITY  **
READ(3,*) QDENS,QVISC

** QTCOND, QCP = THERMAL CONDUCTIVITY & SPECIFIC HEAT  **
READ(3,*) QTCOND,QCP
READ(3,*) FILMW,FILMF,FILMC
READ(3,*) QT0,QTIN,QTWALL

** GX,GY,GZ = BODY ACCELERATION IN X, Y & Z DIRECTIONS  **
** BUOBETA = THERMAL EXPANSION COEFFICIENT  **
READ(3,*) GX,GY,GZ,BUOBETA
READ(3,26) COMMENTS

** QZETA = CRITERION FOR STEADY-STATE CONVERGENCE  **
** MAXSTPQ = MAXIMUM PRESSURE ITERATIONS  **
READ(3,*) QZETA,MAXSTPQ
** ISTMAX = MAXIMUM TIME STEPS  
**  
** TIMEMAX = MAXIMUM SIMULATION TIME  
**  
** USERDT = USER-DETERMINED TIME STEP  
**  
READ(3,*) ISTMAX, TIMEMAX, USERDT  
READ(3,26) COMMENTS  
READ(3,*) NSI1, NSIMAX, NSJ1, NSJMAX, NSK1, NSKMAX  
READ(3,26) COMMENTS  
READ(3,*) TCFCMU, TCFC1, TCFC2, TSIGK, TSIGE  
READ(3,26) COMMENTS  

** FNAMEOUT = OUTPUT FILE FOR INTERMEDIATE INFORMATION  
**  
READ(3,25) FNAMEOUT  

** FNAMEVEC = OUTPUT FILE FOR VELOCITY VECTOR INFORMATION  
**  
READ(3,25) FNAMEVEC  

** FILMDAT = CONvection COEFFICIENT & OUTLET TEMPERaTURE OUTPUT FILE **  
READ(3,25) FILMDAT  

** EXPFILE = EXPERIMENTAL VELOCITY MAGNITUDE DATA FILE  
**  
READ(3,25) EXPFILE  

** TEXPFILE = EXPERIMENTAL TEMPERATURE DATA FILE  
**  
READ(3,25) TEXPFILE  

** NUMLATS = TOTAL NUMBER OF CONSTANT X-PLOTS  
**  
** NLATI() = CELL VALUE FOR CONSTANT X- PLOT  
**  
READ(3,*) NUMLATS  
DO 36 N=1, NUMLATS  
36 READ(3,*) NLATI(N)
** NUMLONGS = TOTAL NUMBER OF CONSTANT Z-PILOTS
**
** NLONGK() = CELL VALUE FOR CONSTANT Z-PILOTS
**
READ(3,*) NUMLONGS
DO 38 N=1,NUMLONGS
  38 READ(3,*) NLONGK(N)
READ(3,26) COMMENTS
  NUMTYPES=0
  NUM3DPLTS=0
** NOPT(N,M) = OUTPUT PRINTING OPTIONS (1=yes, 0=no)
**
** NOPT(N,1) -> TABULAR PRINTOUT OF DATA
**
** NOPT(N,2) -> GRAPHICAL PRINTOUT OF DATA
**
** NOPT(N,3) -> 3-D OUTPUT FILE FOR USE WITH NCSA XDATASLICE
**
** NOPT(1,M) -> U-VELOCITY OUTPUT
**
** NOPT(2,M) -> V-VELOCITY OUTPUT
**
** NOPT(3,M) -> W-VELOCITY OUTPUT
**
** NOPT(4,M) -> VELOCITY MAGNITUDE
**
** NOPT(5,M) -> MIXTURE FRACTION
**
** NOPT(6,M) -> TEMPERATURE
**
** NOPT(7,M) -> TURBULENT KINETIC ENERGY
**
** NOPT(8,M) -> TURBULENT DISSIPATION
**
DO 40 N=1,8
  READ(3,*) NOPT(N,1),NOPT(N,2),NOPT(N,3)
  IF (NOPT(N,3).EQ.1) NUM3DPLTS=NUM3DPLTS+1
  IF (NOPT(N,2).EQ.1) NUMTYPES=NUMTYPES+1

NLATPLTS=NUMLATS*NUMTYPES
NLONPLTS=NUMLONGS*NUMTYPES
** SYSLAT() = LATERAL OUTPUT FILE FOR USE WITH SYSTAT SOFTWARE **
** HDFLAT() = LATERAL OUTPUT FILE FOR USE WITH NCSA IMAGE3.1.1 **

    DO 42 N=1,NLATPLTS
    READ(3,25) SYSLAT(N)
  42 READ(3,25) HDFLAT(N)

** SYSLON() = LONGITUDINAL OUTPUT FILE FOR SYSTAT SOFTWARE **
** HDFLON() = LONGITUDINAL OUTPUT FILE FOR NCSA IMAGE3.1.1 **

    DO 44 N=1,NLONPLTS
    READ(3,25) SYSLON(N)
  44 READ(3,25) HDFLON(N)

** HDF3D() = 3D OUTPUT FILE FOR USE WITH NCSA XDATASLICE **

    DO 46 N=1,NUM3DPLTS
  46 READ(3,25) HDF3D(N)

READ(3,26) COMMENTS
** NUMPLTS = NUMBER OF INTERMEDIATE 2D LONGITUDINAL PLOTS **
** INTRVLS() = TIME STEP NUMBER TO PRINT INTERMEDIATE FILE **
** VTRANS() = NAME OF VELOCITY MAGNITUDE TRANSIENT FILE **
** TTRANS() = NAME OF TEMPERATURE TRANSIENT FILE **

    READ(3,*) NUMPLTS
    DO 48 N=1,NUMPLTS
  48 READ(3,27) INTRVLS(N), VTRANS(N), TTRANS(N)
C 48 WRITE(*,27) INTRVLS(N), VTRANS(N), TTRANS(N)
    CLOSE(3)

**************************************************************************************
**************************************************************************************
**************************************************************************************
OPEN INTERMEDIATE FILES ***
C    OPEN(UNIT=10, FILE=FNAMEOUT)
    OPEN(UNIT=4, FILE=FILMDAT)
    OPEN(UNIT=5, FILE='genv.out')
OPEN(UNIT=9,FILE='gent.out')
OPEN(UNIT=8,FILE='eror1.dat')
OPEN(UNIT=11,FILE='strat.dat')

**********************************************************
************
*** 1.2 DETERMINE GEOMETRICAL AND NUMERICAL PARAMETERS
***
**********************************************************
*******************************************************
GEOMETRY ***
IMM1=IMAX-1
JMM1=JMAX-1
KMM1=KMAX-1
IMM2=IMAX-2
JMM2=JMAX-2
KMM2=KMAX-2

** IF THE WALL FUNCTION MODEL IS USED, THEN THE POWER LAW
IS ALSO **
** USED. (RATHER THAN NO SLIP OR FREE SLIP- SEE UPDATEBC
SUBROUTINE)**
IF(NFLOW .EQ. 2) THEN
NSI1=2
NSIMAX=2
NSJ1=2
NSJMAX=2
NSK1=2
NSKMAX=2
ENDIF

************************* CALCULATE VOLFLO: THE VOLUME
FLOW RATE ***
UMAX=-9999.99
VOLFLO=0.0
WCOUNT=KIN(1)
HCOUNT=JIN(1)
DO 120 N=1,NINCELS
IF (UINS(N).GT.UMAX) UMAX=UINS(N)
IF (N .GE. 2) WCOUNT=WCOUNT+1
IF (WCOUNT .GT. KIN(NINCELS)) THEN
WCOUNT=KIN(1)
HCOUNT=HCOUNT+1
ENDIF
VOLFLO=VOLFLO+UINS(N)*DYM(HCOUNT)*DZM(WCOUNT)
120 CONTINUE

***** CALCULATE AREAIN & AREAOUT: AREA OF INLET & AREA OF OUTLET  ***
   AREAIN=0.0
   AREAOUT=0.0
   DO 122 K=KIN(1),KIN(NINCELS)
       DO 122 J=JIN(1),JIN(NINCELS)
   122 AREAIN=AREAIN+DYM(J)*DZM(K)
   NOUTCELS=NOUTJ*NOUTK
   N=1
   DO 124 J=1,NOUTJ
       DO 124 K=1,NOUTK
   JOUT(N)=JOUTTEMP(J)
       KOUT(N)=KOUTTEMP(K)
   124 N=N+1
   DO 126 K=KOUT(1),KOUT(NOUTCELS)
       DO 126 J=JOUT(1),JOUT(NOUTCELS)
   126 AREAOUT=AREAOUT+DYM(J)*DZM(K)

***********************************************************************
CALCULATE ACH, UJET & UJOUT  ***
*** ACH = NUMBER OF AIR CHANGES PER HOUR  ***
*** UJET = AVERAGE INLET VELOCITY  ***
*** UJOUT = OUTLET VELOCITY  ***

ACH=3600.0*VOLFLO/(XDIMEN*YDIMEN*ZDIMEN)
UJET=ACH*XDIMEN*YDIMEN*ZDIMEN/(3600.0*AREAIN)
UJOUT=UJET*AREAIN/AREAOUT
VIS=QVISC
SC=0.7
EXCH=VIS/SC

***********************************************************************
CALCULATE THERMAL DIFFUSIVITY  ***
TDIFUS=QTCOND/(QDENS*QCP)
1.3 SET INITIAL VALUES, CALC TIME STEP CRITERIA & PRESSURE CORRECTION VALUES

**INITIAL VALUES**

\[
\begin{align*}
V_I &= 0.0 \\
W_I &= 0.0 \\
U_I &= 0.0 \\
S_I &= 0.0 \\
T_{KI} &= T_{KIN}(1) \times 0.1 \\
T_{EI} &= T_{EIN}(1) \times 0.1
\end{align*}
\]

**INITIAL NUMERICAL VALUES**

\[
\begin{align*}
\text{TIME} &= 0.0 \\
\text{ITER} &= 0 \\
\text{ISTEP} &= 0 \\
\text{ISTRAT} &= 0
\end{align*}
\]

**INITIAL RESIDUAL VALUES**

\[
\begin{align*}
D_{SUMN} &= 1.0 \\
R_{SUN} &= 1.0 \\
R_{SVN} &= 1.0 \\
R_{SNN} &= 1.0 \\
R_{ESSN} &= 1.0
\end{align*}
\]

**TIME STEP CRITERIA**

\[
\text{DELT} = \text{TIME STEP}
\]

The time step is determined by the smallest value of:

1) \(D_{TMAX1} \)  
2) \(D_{TMAX2} \)  
3) \(D_{TMAX3} \)  
4) \(D_{TMAX4} \)  
5) \(D_{USERDT} \)

\[
D_{TMAX1} = 0.333 \times D_{MIN1}(D_X(2), D_Y(2), D_Z(2)) / U_{MAX}
\]
DTMAX2=0.5*DMIN1(DX(2)**2,DY(2)**2,DZ(2)**2)/VIS

DTMAX3=(0.5/(1./DX(2)**2+1./DY(2)**2+1./DZ(2)**2))/VIS

DTMAX4=(0.5/(1./DX(2)**2+1./DY(2)**2+1./DZ(2)**2))/TDIFUS

DELT=DTMAX1
IF (DELT.GT.DTMAX2) DELT=DTMAX2
IF (DELT.GT.DTMAX3) DELT=DTMAX3
IF (DELT.GT.USERDT) DELT=USERDT

************************** PRESSURE ITERATION
        DIVISOR
*** DZRO = DIVISOR USED IN P-U-V-W ITER. FOR CONTINUITY IMPOSITION ***

DO 137 I=2,IMM1
DO 137 J=2,JMM1
137 DZRO(I,J)=UJET/(DXM(I)*DYM(J))

************************** PRESSURE CORRECTION VALUE
*** BETA() = PRESSURE CORRECTION VALUE TO BE USED IN SECTION 6.0 ***
*** OMG = OVERRELAXATION FACTOR (BETWEEN 1 & 2) ***

DO 139 I=2,IMM1
DO 139 J=2,JMM1
DO 139 K=2,KMM1
139
BETA(I,J,K)=OMG/(2*DELT*(1/(DXM(I)**2.)+1/(DYM(J)**2.)
          + 1/(DZM(K)**2.)))

******************************************************************************

*** 1.4 PRINT INITIAL HEADING
***
******************************************************************************

140 FORMAT(///,1X,130('='),/,1X,A50,40X,A40,/,1X,130('='),//,1X,
        c'GEOMETRY:',
        c6X,'IMAX: ',I3,' JMAX: ',I3,' KMAX: ',I3,' DX(2): ',

cE10.4,' m   DY(2): ',E10.4,' m   DZ(2): 
',E10.4,' m'//,59X,  
c'X-DIM: ',E10.4,' m   Y-DIM: ',E10.4,' m   Z-DIM: ',E10.4,' m'//,1X,'AIR FLOW:    ',F7.2,' ACH
UINLET: ',E10.4,
c' m/s   OUTLET: ',E10.4,' m/s   UI: ',F6.2,'  
VI: ',
cF6.2,'  WI: ',F6.2)

142 FORMAT(/,1X,A15,10(I2,',',I2,4X))
143 FORMAT(16X,10(I2,',',I2,4X))
144 FORMAT(/,1X,'FLUID: DENSITY: ',E11.5,' kg/m^3',
c' VISCOSITY: ',E11.5,' m^2/s   TCOND: 
',E11.5,' W/(m*K)',
,c//,21X,'Cp: ',E11.5,' J/(kg*K)   TDIFUS: 
',E11.5,' m^2/s',
c//,1X,'TEMPERATURE:    ',
c'INITIAL: ',F7.2,' C   INLET: ',F7.2,' C
WALLS: ',
c1X,F6.2,' C',//,1X,'BUOYANCY:   GX: ',F5.2,'  
m/s^2   GY: ',
cF5.2,' m/s^2   GZ: ',F5.2,' m/s^2   EXP.COEFF: 
',
cE11.5,' 1/K',//)
146 FORMAT(1X,'BOUNDARY',//,1X,'CONDITIONS: NSI1: 
',',I1,
c' NSIMAX: ',',I1,' NSJ1: ',',I1,' NSJMAX: ',',I1,' NSK1: ',
c'I1,' NSKMAX: ',',I1,/
147 FORMAT(1X,'TURBULENCE: Cmu: ',F6.3,5X,'C1: 
',',F6.3,5X,  
148 FORMAT(1X,'TIME STEP: ',E9.3,'s   DTMAX1:',E9.3,
c'DTMAX2:',E9.3,' DTMAX3:',E9.3,' DTMAX4:',E9.3,
c'USERDT:',E9.3, 
c//,1X,'NUMERICAL: OMEGA: ',F6.4,'   ALPHA: 
',',F6.4,  
c//,1X,'CONVERGENCE: EPSI: ',
cE11.5,'   DZRO: ',E11.5,'   ZETA: ',E11.5,
c'   MAX PRESS ITS: ',I7,
c/,16X,'MAX TIME STEPS: ',I6,' MAX TIME: ',F9.3)
149 FORMAT(/,1X,'OUTPUT [D,P]: U['',2(I1),'], V['',2(I1),'], W['',2(I1),'], Vmag['',2(I1),'], S['',2(I1),'], T['',2(I1),'], K['',2(I1),'], E['',2(I1),']',//,1X,130('='))

WRITE(*,140)
TITLE,FLOWTYPE(NFLOW),IMAX,JMAX,KMAX,DX(2),DY(2),

DZ(2),XDIMEN,YDIMEN,ZDIMEN,ACH,UJET,UJOUT,UI,VI, WI
WRITE(10,140)
TITLE,FLOWTYPE(NFLOW),IMAX,JMAX,KMAX,DX(2),DY(2),

DZ(2),XDIMEN,YDIMEN,ZDIMEN,ACH,UJET,UJOUT,UI,VI, WI
L1=INT(NINCELS/10)
L2=MIN(NINCELS,10)
WRITE(*,142) 'INLET (J,K):
'

C WRITE(10,142) 'INLET (J,K):
'
DO 150 N=1,L1
LS=1+N*10
LE=MIN((10+N*10),NINCELS)
WRITE(10,143) (JIN(NN),KIN(NN),NN=LS,LE)
150 WRITE(*,143) (JIN(NN),KIN(NN),NN=LS,LE)

L1=INT(NOUTCELS/10)
L2=MIN(NOUTCELS,10)
WRITE(*,142) 'OUTLET (J,K):
'

C WRITE(10,142) 'OUTLET (J,K):
'
DO 152 N=1,L1
LS=1+N*10
LE=MIN((10+N*10),NOUTCELS)
WRITE(10,143) (JOUT(NN),KOUT(NN),NN=LS,LE)
152 WRITE(*,143) (JOUT(NN),KOUT(NN),NN=LS,LE)
WRITE(*,144)
QDENS,VIS,QTCOND,QCP,TDIFUS,QT0,QTIN,QTWALL,

C GX,GY,GZ,BUOBETA
WRITE(*,146) NSI1,NSIMAX,NSJ1,NSJMAX,NSK1,NSKMAX

147
WRITE(*,147) TCFCMU, TCFC1, TCFC2, TSIGK, TSIGE
WRITE(*,148) DELT, DMAX1, DMAX2, DMAX3, DMAX4, USERDT, OMG, ALPHA,
C EPSI, DZRO(2,2), QZETA, MAXSTPQ, ISTMAX, TIMEMAX
WRITE(*,149) NOPT(1,1), NOPT(1,2), NOPT(2,1), NOPT(2,2),
C NOPT(3,1), NOPT(3,2), NOPT(4,1), NOPT(4,2),
C NOPT(5,1), NOPT(5,2), NOPT(6,1), NOPT(6,2),
C NOPT(7,1), NOPT(7,2), NOPT(8,1), NOPT(8,2)
C WRITE(10,144) QDENS, VIS, QTCOND, QCP, TDIFUS, QT0, QTIN, QTWALL,
C GX, GY, GZ, BUOBETA
C WRITE(10,146) NSI1, NSIMAX, NSJ1, NSJMAX, NSK1, NSKMAX
C WRITE(10,147) TCFCMU, TCFC1, TCFC2, TSIGK, TSIGE
C WRITE(10,148) DELT, DMAX1, DMAX2, DMAX3, DMAX4, USERDT, OMG, ALPHA,
C EPSI, DZRO(2,2), QZETA, MAXSTPQ, ISTMAX, TIMEMAX
WRITE(10,149) NOPT(1,1), NOPT(1,2), NOPT(2,1), NOPT(2,2),
C NOPT(3,1), NOPT(3,2), NOPT(4,1), NOPT(4,2),
C NOPT(5,1), NOPT(5,2), NOPT(6,1), NOPT(6,2),
C NOPT(7,1), NOPT(7,2), NOPT(8,1), NOPT(8,2)
C WRITE(*,*)' ( 1-CONTINUE 0-STOP)'
NCONTINU=1
IF (NCONTINU.EQ.0) STOP

**********************************************************************
*************
*** 2.0 INITIALIZE FLOW FIELD
***
**********************************************************************
*************
200 CONTINUE
   DO 202 I=1,IMAX
   DO 202 J=1,JMAX

148
DO 202 K=1,KMAX
U(I,J,K)=UI
V(I,J,K)=VI
W(I,J,K)=WI
S(I,J,K)=SI
P(I,J,K)=0.0
T(I,J,K)=QT0
TK(I,J,K)=TKI
TE(I,J,K)=TEI
202 CONTINUE

**********************************************************
************
*** 3.0 SET BOUNDARY CONDITIONS AND INITIAL PRINTOUT***
************
**********************************************************
************
CALL UPDATEBC(NFLOW,VIS,QTCOND)
IF (ITSTEP.EQ.0) GO TO 690
************
************
*** 4.0 TIME-ADVANCE LOOP BEGINS***
************
400 CONTINUE
ITER=0
FLG=1.0

************
*******
*** 5.0 COMPUTE TIME-ADVANCED VALUES***
************
************
IF (NFLOW.NE.1) THEN
CALL CALTFUNC(NFLOW,VIS)
CALL TURBVISC(VIS)
CALL DERIVS(ITSTEP)
ENDIF

DO 500 I=2,IMM1
DO 500 J=2,JMM1
DO 500 K=2,KMM1
5.1 U-VELOCITY

\[
\begin{align*}
\text{VISX} &= \text{VIS} \times \left( \frac{(\text{UN}(I+1,J,K) - \text{UN}(I,J,K))}{\text{DXM}(I+1)} - \frac{(\text{UN}(I,J,K) - \text{UN}(I-1,J,K))}{\text{DX}(I)} \right) \\
&+ \frac{((\text{UN}(I,J+1,K) - \text{UN}(I,J,K))}{\text{DYM}(J)} - \frac{((\text{UN}(I,J,K) - \text{UN}(I,J-1,K))}{\text{DYM}(J-1)})}{\text{DYM}(J)} \\
&+ \frac{((\text{UN}(I,J,K+1) - \text{UN}(I,J,K))}{\text{DZM}(K)} - \frac{((\text{UN}(I,J,K) - \text{UN}(I,J,K-1))}{\text{DZM}(K)} \\
\text{FUX} &= 0.25 \times \left( \frac{((\text{UN}(I,J,K) + \text{UN}(I+1,J,K))}{\text{DXM}(I+1)} - \frac{((\text{UN}(I-1,J,K) + \text{UN}(I,J,K))}{\text{DXM}(I)} \\
&+ \frac{(\text{UN}(I,J,K) + \text{UN}(I+1,J,K))}{\text{DY}(J)} - \frac{(\text{UN}(I,J,K) + \text{UN}(I+1,J,K))}{\text{DY}(J-1)})}{\text{DY}(J)} \\
&+ \frac{(\text{UN}(I,J,K) + \text{UN}(I+1,J,K))}{\text{DZ}(K)} - \frac{(\text{UN}(I,J,K) + \text{UN}(I+1,J,K))}{\text{DZ}(K-1)})}{\text{DZ}(K)} \\
\text{U}(I,J,K) &= \text{UN}(I,J,K) + \text{DELT} \times \left( \frac{(\text{P}(I,J,K) - \text{P}(I+1,J,K))}{\text{DX}(I)} + \frac{\text{GX} \times \text{BUO} \beta \times (\text{QT0} - \text{T}(I,J,K))}{\text{DX}(I)} - \frac{\text{FUX} - \text{FUY} - \text{FUZ}}{\text{DX}(I)} \\
\text{UTURB} &= \text{UN}(I,J,K) + \text{DELT} \times \left( \frac{(\text{P}(I,J,K) - \text{P}(I+1,J,K))}{\text{DX}(I)} + \frac{\text{GX} \times \text{BUO} \beta \times (\text{QT0} - \text{T}(I,J,K))}{\text{DX}(I)} + \text{GAMX}(I,J,K) - \text{FUX} - \text{FUY} - \text{FUZ} \\
\end{align*}
\]
IF(NFLOW.NE.1) U(I,J,K)=UTURB

**********************************************************
************
*** 5.2 V-VELOCITY
***
**********************************************************

************

VISY=VIS*(((VN(I+1,J,K)-VN(I,J,K))/DX(I) +
          ((VN(I,J,K)-VN(I-1,J,K))/DX(I-1))/DXM(I)+
          ((VN(I,J+1,K)-VN(I,J,K))/DYM(J+1) -
          (VN(I,J,K)-VN(I,J-1,K))/DYM(J))/DY(J)+
          ((VN(I,J,K+1)-VN(I,J,K))/DZM(K) -
          (VN(I,J,K)-VN(I,J,K-1))/DZM(K))/DZM(K))

FVX=0.25*(((UN(I,J,K)+UN(I,J+1,K))*(VN(I,J,K)+VN(I+1,J,K)) +
          ALPHA*ABS(UN(I,J,K)+UN(I,J+1,K))*(VN(I,J,K) -
          VN(I+1,J,K)))/DX(I) -((UN(I-1,J,K)+UN(I-1,J+1,K))*(VN(I-1,J,K)+VN(I,J,K)) +
          ALPHA*ABS(UN(I-1,J,K)+UN(I-1,J+1,K))*(VN(I-1,J,K) -
          VN(I,J,K)))/DX(I-1))

FVY=0.25*(((VN(I,J,K)+VN(I,J+1,K))**2 +
          ALPHA*ABS(VN(I,J,K)+VN(I,J+1,K))*(VN(I,J,K) -
          VN(I,J+1,K)))/DYM(J+1) -((VN(I,J-1,K)+VN(I,J,K))**2 +
          ALPHA*ABS(VN(I,J-1,K)+VN(I,J,K))*(VN(I,J-1,K) -
          VN(I,J,K)))/DYM(J))

FVZ=0.25 *((WN(I,J,K)+WN(I,J+1,K)) *(VN(I,J,K)+
          WN(I,J,K+1))+ALPHA*ABS(WN(I,J,K)) +
          WN(I,J,K)+WN(I,J+1,K))/DZM(K) -((WN(I,J,K-1)+WN(I,J+1,K-1)) *(WN(I,J,K-1)+
          WN(I,J+1,K)) +ALPHA*ABS(WN(I,J,K-1)+WN(I,J+1,K-1))*(WN(I,J,K-1) -
          WN(I,J,K)))/DZM(K))

V(I,J,K)=VN(I,J,K)+DELT*((P(I,J,K)-P(I,J+1,K))/DY(J) +
          -FVX-FVY-FVZ+VISY+GY*BUOBETA*(QT0-T(I,J,K)))

VTURB=VN(I,J,K)+DELT*((P(I,J,K)-P(I,J+1,K))/DY(J) +
          -FVX-FVY-FVZ+GAMY(I,J,K)+GY*BUOBETA*(QT0-T(I,J,K)))
IF (NFLOW.NE.1)  V(I,J,K)=VTURB

**************************************************************
************
*** 5.3 W-VELOCITY
***
**************************************************************

**********

VISZ=VIS*(((WN(I+1,J,K)-WN(I,J,K))/DX(I)
+ (WN(I,J+1,K)-WN(I,J,K))/DY(J)
+ (WN(I,J,K+1)-WN(I,J,K))/DZM(K+1))
+ ((WN(I,J,K)-WN(I,J,K))/DXM(I)
+ (WN(I,J,K)-WN(I,J,K))/DYM(J)
+ (WN(I,J,K)-WN(I,J,K))/DZM(K))/DXM(I)
+ DYM(J)
+ DZM(K))

FWX=0.25*(((UN(I,J,K)+UN(I,J,K+1))*(WN(I,J,K)+WN(I+1,J,K))
+ALPHA*ABS(UN(I,J,K)+UN(I,J,K+1))*(WN(I,J,K)
- WN(I+1,J,K)))/DX(I)
-((UN(I-1,J,K)+UN(I-1,J,K+1))*(WN(I-1,J,K)+WN(I,J,K))
+ALPHA*ABS(UN(I-1,J,K)+UN(I-1,J,K+1))*(WN(I-1,J,K)
- WN(I,J,K)))/DZM(K))

FWY=0.25*(((VN(I,J,K)+VN(I,J,K+1))*(WN(I,J,K)+WN(I,J+1,K))
+ALPHA*ABS(VN(I,J,K)+VN(I,J,K+1))*(WN(I,J,K)
- WN(I,J+1,K)))/DY(J)
-((VN(I,J-1,K)+VN(I,J-1,K+1))*(WN(I,J-1,K)+WN(I,J,K))
+ALPHA*ABS(VN(I,J-1,K)+VN(I,J-1,K+1))*(WN(I,J-1,K)
- WN(I,J,K)))/DYM(J))

FWZ=0.25*(((WN(I,J,K)+WN(I,J,K+1))**2 +ALPHA*ABS(WN(I,J,K)+WN(I,J,K))*(WN(I,J,K)
- WN(I,J,K+1)))/DZM(K+1)
- WN(I,J,K)))/DZM(K))

W(I,J,K)=WN(I,J,K)+DELT*((P(I,J,K) - P(I,J,K+1))/DZ(K)
+GZ*BUOBIETA*(QT0-T(I,J,K)) -FWX -FWY -FWZ
+VISZ)

WTURB=WN(I,J,K) + DELT*((P(I,J,K) - P(I,J,K+1))/DZ(K)
+152
1 +GZ*BUO+GAMZ(I,J,K))
   IF (NFL.1) W(I,J,K)=WTURB

**************************************************************
************
*** 5.4 TEMPERATURE
***
**************************************************************

VIST=TDIFUS*(((TN(I+1,J,K)-TN(I,J,K))/DX(I)
1 + (TN(I-1,J,K)-TN(I,J,K))/DX(I-1))/DXM(I) +
2 ((TN(I,J+1,K)-TN(I,J,K))/DY(J)
3 + (TN(I,J-1,K)-TN(I,J,K))/DY(J-1))/DYM(J) +
4 ((TN(I,J,K+1)-TN(I,J,K))/DZ(K)
5 + (TN(I,J,K-1)-TN(I,J,K))/DZM(K))

FTX=((UN(I,J,K)*(TN(I,J,K)+TN(I+1,J,K))+ALPHA*ABS(UN(I,J,K)
1 ) (TN(I,J,K)-TN(I+1,J,K)))/DX(I)
2 -(UN(I-1,J,K)*(TN(I-1,J,K)+TN(I,J,K)
3 +ALPHA*ABS(UN(I-1,J,K))*(TN(I-1,J,K)-
TN(I,J,K))/DX(I-1))/2.

FTY=((VN(I,J,K)*(TN(I,J,K)+TN(I,J+1,K))+ALPHA*ABS(VN(I,J,K)
1 ) (TN(I,J,K)-TN(I,J+1,K)))/DY(J)
2 -(VN(I,J-1,K)*(TN(I,J-1,K)+TN(I,J,K)
3 +ALPHA*ABS(VN(I,J-1,K))*(TN(I,J-1,K)-
TN(I,J,K))/DY(J-1))/2.

FTZ=((WN(I,J,K)*(TN(I,J,K)+TN(I,J,K+1))+ALPHA*ABS(WN(I,J,K)
1 ) (TN(I,J,K)-TN(I,J,K+1)))/DZ(K)
2 -(WN(I,J,K-1)*(TN(I,J,K-1)+TN(I,J,K)
3 +ALPHA*ABS(WN(I,J,K-1))*(TN(I,J,K-1)-
TN(I,J,K))/DZ(K-1))/2.

T(I,J,K)=TN(I,J,K)+DELT*(-FTX-FTY-FTZ+VIST)

**************************************************************
************
*** 5.5 MIXTURE FRACTION (S)
***
VISS = EXCH*(((SN(I+1,J,K) - SN(I,J,K))/DX(I) +
  ((SN(I,J,K+1) - SN(I,J,K))/DZ(K) +
  (SN(I,J,K-1) - SN(I,J,K))/DZ(K-1))/DZM(K))
FSX = ((UN(I,J,K) * (SN(I,J,K) + SN(I+1,J,K)) + ALPHA*ABS(UN(I,J,K)) *
  (SN(I,J,K) - SN(I+1,J,K)))/DX(I) -
  (UN(I-1,J,K) * (SN(I-1,J,K) + SN(I,J,K)) +
  ALPHA*ABS(UN(I-1,J,K)) * (SN(I-1,J,K) -
  SN(I,J,K)))/DX(I-1))/2.
FSY = ((VN(I,J,K) * (SN(I,J,K) + SN(I,J+1,K)) + ALPHA*ABS(VN(I,J,K)) *
  (SN(I,J,K) - SN(I,J+1,K)))/DY(J) -
  (VN(I,J-1,K) * (SN(I,J-1,K) + SN(I,J,K)) +
  ALPHA*ABS(VN(I,J-1,K)) * (SN(I,J-1,K) -
  SN(I,J,K)))/DY(J-1))/2.
FSZ = ((WN(I,J,K) * (SN(I,J,K) + SN(I,J,K+1)) + ALPHA*ABS(WN(I,J,K)) *
  (SN(I,J,K) - SN(I,J,K+1)))/DZ(K) -
  (WN(I,J,K-1) * (SN(I,J,K-1) + SN(I,J,K)) +
  ALPHA*ABS(WN(I,J,K-1)) * (SN(I,J,K-1) -
  SN(I,J,K)))/DZ(K-1))/2.
S(I,J,K) = SN(I,J,K) + DELT*(-FSX-FSY-FSZ+VISS)

*** 5.6 TURBULENT ENERGY (K) ***

IF (NFLOW.NE.1) THEN
  THETAK = CALTHETA(ITSTEP, TKN, TSIGK, I, J, K)
\[
\begin{align*}
FKX &= ((UN(I,J,K) \cdot (TKN(I,J,K) + TKN(I+1,J,K)) + \alpha \cdot \text{ABS}(UN(I,J,K)) \cdot \text{ABS}(TKN(I,J,K) - TKN(I+1,J,K))) / DX(I)) \\
&\quad - (UN(I-1,J,K) \cdot (TKN(I-1,J,K) + TKN(I,J,K)) + \alpha \cdot \text{ABS}(UN(I-1,J,K)) \cdot \text{ABS}(TKN(I-1,J,K) - TKN(I,J,K))) / DX(I-1)) / 2.
\end{align*}
\]

\[
FKY = ((VN(I,J,K) \cdot (TKN(I,J,K) + TKN(I,J+1,K)) + \alpha \cdot \text{ABS}(VN(I,J,K)) \cdot \text{ABS}(TKN(I,J,K) - TKN(I,J+1,K))) / DY(J)) \\
&\quad - (VN(I,J-1,K) \cdot (TKN(I,J-1,K) + TKN(I,J,K)) + \alpha \cdot \text{ABS}(VN(I,J-1,K)) \cdot \text{ABS}(TKN(I,J-1,K) - TKN(I,J,K))) / DY(J-1)) / 2.
\]

\[
FKZ = ((WN(I,J,K) \cdot (TKN(I,J,K) + TKN(I,J,K+1)) + \alpha \cdot \text{ABS}(WN(I,J,K)) \cdot \text{ABS}(TKN(I,J,K) - TKN(I,J,K+1))) / DZ(K)) \\
&\quad - (WN(I,J,K-1) \cdot (TKN(I,J,K-1) + TKN(I,J,K)) + \alpha \cdot \text{ABS}(WN(I,J,K-1)) \cdot \text{ABS}(TKN(I,J,K-1) - TKN(I,J,K))) / DZ(K-1)) / 2.
\]

\[
TK(I,J,K) = TKN(I,J,K) + \Delta T \cdot (\Theta_K + TVIS(I,J,K) \cdot PI(I,J,K)) \\
&\quad - F\dot{X} - F\dot{Y} - F\dot{Z} - TEN(I,J,K).
\]

**********************************************************
************
*** 5.7 TURBULENT DISSIPATION (EPSILON)
***
************
**********************************************************

\[
\theta_{AE} = CALTHETA(ITSTEP,TEN,TSIGE,I,J,K)
\]

\[
FEX = ((UN(I,J,K) \cdot (TEN(I,J,K) + TEN(I+1,J,K)) + \alpha \cdot \text{ABS}(UN(I,J,K)) \cdot \text{ABS}(TEN(I,J,K) - TEN(I+1,J,K))) / DX(I)) \\
&\quad - (UN(I-1,J,K) \cdot (TEN(I-1,J,K) + TEN(I,J,K)) + \alpha \cdot \text{ABS}(UN(I-1,J,K)) \cdot \text{ABS}(TEN(I-1,J,K) - TEN(I,J,K))) / DX(I-1)) / 2.
\]
FEY = \((VN(I,J,K) \times (TEN(I,J,K) + TEN(I,J+1,K)) + \text{ALPHA} \times \text{ABS}(VN(I,J,K)) \times (TEN(I,J, K) - TEN(I,J+1,K))) / DY(J)\)
\(- (VN(I,J-1,K) \times (TEN(I,J-1,K) + TEN(I,J,K))) - \text{ALPHA} \times \text{ABS}(VN(I,J-1,K)) \times (TEN(I,J-1,K) - TEN(I,J,K)) / DY(J-1)) / 2.\)

FEZ = \((WN(I,J,K) \times (TEN(I,J,K) + TEN(I,J,K+1)) + \text{ALPHA} \times \text{ABS}(WN(I,J,K)) \times (TEN(I,J,K) - TEN(I,J,K+1))) / DZ(K)\)
\(- (WN(I,J,K-1) \times (TEN(I,J,K-1) + TEN(I,J,K))) - \text{ALPHA} \times \text{ABS}(WN(I,J,K-1)) \times (TEN(I,J,K-1) - TEN(I,J,K)) / DZ(K-1)) / 2.\)

TE(I,J,K) = TEN(I,J,K) + DELT \times (\text{THETAE} - \text{FEX} - \text{FEY} - \text{FEZ} + (\text{TFN1(I,J,K) \times TCFC1 \times TVIS(I,J,K) \times TEN(I,J,K) \times PI(I,J,K) / TKN(I,J,K)}\)
\(- (\text{TFN2(I,J,K) \times TCFC2 \times (TEN(I,J,K) \times 2.0) / TKN(I,J,K)}))\)

ENDIF

500 CONTINUE

CALL UPDATEBC(NFLOW,VIS,QTCOND)

**********************************************************
************
*** 6.0 P-U-V-W ITERATION FOR CONTINUITY IMPOSITION
***
**********************************************************

600 CONTINUE

IF ((\text{FLAG6 EQ 0.) \&\& (ITER GE 5))) GO TO 690
ITER = ITER + 1

IF (ITER LT MAXSTPQ) GO TO 610

GOTO 690

610 \text{FLAG6} = 0.0

DSUM = 0.0

DO 650 I = 2, IMM1
DO 650 J = 2, JMM1
DO 650 K = 2, KMM1
D = 1/DMX(I) * (U(I, J, K) - U(I-1, J, K)) + 1/DYM(J) * (V(I, J, K) - V(I, J-1, K)) + 1/DZM(K) * (W(I, J, K) - W(I, J, K-1))

DSUM = DSUM + ABS(D)

IF (ABS(D/DZRO(I, J)).GE.EPSI) FLAG6 = 1.

DELP = -D * BETA(I, J, K)

P(I, J, K) = P(I, J, K) + DELP

U(I, J, K) = U(I, J, K) + DELT * DELP / DMX(I)

U(I-1, J, K) = U(I-1, J, K) - DELT * DELP / DMX(I)

V(I, J, K) = V(I, J, K) + DELT * DELP / DYM(J)

V(I, J-1, K) = V(I, J-1, K) - DELT * DELP / DYM(J)

W(I, J, K) = W(I, J, K) + DELT * DELP / DZM(K)

W(I, J, K-1) = W(I, J, K-1) - DELT * DELP / DZM(K)

650 CONTINUE

661 FORMAT(I4, 2X, 2(F12.8))

CALL UPDATEBC(NFLOW, VIS, QTCOND)

GO TO 600

690 CONTINUE

*********************************************************************************************************

************

*** 7.0 RESIDUALS - CHANGE IN U, V, W, S OVER LAST TIME STEP ***

************

*********************************************************************************************************

************

IF (ISTEP.NE.0) THEN

CALL RESID(ISTEP, DSUM, IBIG, JBIG, KBIG, THEERR)

ERROR1 = THEERR / (IMAX * JMAX * KMAX)

ENDIF

************

************

*** 8.0 INTERMEDIATE PRINTING ***

************

800 CONTINUE

801 FORMAT(1X, F6.2, 4X, E12.5)

805 FORMAT(//, 19X, '|------------------ VALUES AT (', I2, ',', I2, ',', cI2, ') -----------------|', 43X, 'AVERAGE', ',//,)
&c1X,'STEP TIME ITER | U V W S',
c' K E | DSUMN RESUN RESVN RESWN',
c' RESSN ERROR',/,,1X,130('='))

810 FORMAT(1X,I5,1X,F6.2,1X,I4,1X,6E10.2,5(1X,F7.4),1X,E12.5,3I4)
811 FORMAT(5(3X,F10.4))

****************************** PRINT THE ABSOLUTE ERROR (EROR1) ***
WRITE(8,801)TIME,EROR1
C
IF (ISTEP.EQ.0) WRITE(10,805) 4,4,3
IF (ISTEP.EQ.0) WRITE(*,805) 4,4,3
IF (ISTEP.LE.10) GO TO 850
IF (ISTEP/50*50.NE.ISTEP) GO TO 848

********** CALCULATE CONVECTION COEFFICIENT & OUTLET TEMPERATURE ***
CALL CCOEF(TIME,QAVG,FTOT,TNUMOUT,HNUM)
WRITE(4,811) TIME,QAVG,FTOT,TNUMOUT,HNUM

******************************************************************************
CALCULATE GLOBAL ERROR NUMBERS ***
CALL GERROR(TIME,EXPFILE,TEXPFILE)

******************************************************************************
CALCULATE STRATIFICATION VALUES ***
ISTRAT=ISTRAT+1
CALL STRAT(TIME,ISTRAT)

848 IF ((ISTEP/50*50.NE.ISTEP).AND.(ISTEP.NE.1)) GO TO 851

850 WRITE(*,810)
ISTEP,TIME,ITER,U(4,4,3),V(4,4,3),W(4,4,3),
S(4,4,3),TK(4,4,3),TE(4,4,3),DSUMN,RESUN,RESVN,RESSN
,EROR1,IBIG,JBIG,KBIG
WRITE(10,810)
ISTEP,TIME,ITER,U(4,4,3),V(4,4,3),W(4,4,3),
S(4,4,3),TK(4,4,3),TE(4,4,3),DSUMN,RESUN,RESVN,RESWN,RESSN,
C c ERROR1,IBIG,JBIG,KBIG

************
*** PRINT TRANSIENT 2-D NCSA IMAGE3.1.1 LONGITUDINAL
*** AND TEMPERATURE/COEFFICIENT DATA
***
************
851 FLAG8=0.0
DO 852 N=1,NUMPLTS
852 IF (ISTEP .EQ. INTRVLS(N)) FLAG8=1.0
IF (FLAG8 .NE. 1.0) GO TO 900
K=1
855 IF (Z(K)-ZM(1) .LT. 0.127) THEN
K=K+1
GO TO 855
ENDIF
K=K-1
ZDIST=0.127-(Z(K)-ZM(1))
RAT1=ZDIST/DZM(K+1)
RAT2=1.0-RAT1
DO 860 I=1,IMAX
DO 860 J=1,JMAX
VFRONT=DSQRT(U(I,J,K+1)**2.0+V(I,J,K+1)**2.0+W(I,J,K+1)**2.0)
VBEHIND=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0+W(I,J,K)**2.0)
VZ127(I,J)=VFRONT*RAT1+VBEHIND*RAT2
TZ127(I,J)=T(I,J,K+1)*RAT1+T(I,J,K)*RAT2
860 CONTINUE
AMAX=VZ127(1,1)
AMIN=VZ127(1,1)
NPTS = NPTS + 1
DO 870 I=1,IMAX
DO 870 J=1,JMAX
IF (AMIN.GE.VZ127(I,J)) AMIN=VZ127(I,J)
870 IF (AMAX.LE.VZ127(I,J)) AMAX=VZ127(I,J)
OPEN(UNIT=9,FILE=VTRANS(NPTS))
WRITE(9,*) JMM2,IMM2
WRITE(9,*) AMAX, AMIN
WRITE(9,1398)(Y(J),J=2,JMM1)
WRITE(9,1398)(X(I),I=2,IMM1)
DO 880 J=JMM1,2,-1
WRITE(9,1399)(VZ127(I,J),I=2,IMM1)
880 CONTINUE
CLOSE(9)
AMAX=QTWALL
AMIN=QTIN
OPEN(UNIT=9,FILE=TTRANS(NPTS))
WRITE(9,*) JMM2,IMM2
WRITE(9,*) AMAX, AMIN
WRITE(9,1398)(Y(J),J=2,JMM1)
WRITE(9,1398)(X(I),I=2,IMM1)
DO 890 J=JMM1,2,-1
WRITE(9,1399)(TZ127(I,J),I=2,IMM1)
890 CONTINUE
CLOSE(9)

************************************************************************************
************
*** 9.0 CONVERGENCE CHECK & STOP CRITERIA
***
************************************************************************************
************
900 CONTINUE
905 FORMAT(1X,130(', '),'PROGRAM TERMINATED',
      c//,1X, 'TIME STEPS: ',I4,40X,'TIME: ',F8.3,' s',40X,
      c'ERROR: ',E11.5,/',1X,130(', '])/)
915 FORMAT(1X,3(F7.4,3X),4(E13.5,3X))

IF((ISTEP.LT.ISTMAX).AND.(ERROR1.GT.QZETA).AND.(TIME.LT.TIMEMAX))
c THEN

CALL REPACK
ISTEP=ISTEP+1
TIME=TIME+DELT
GO TO 400
ENDIF

WRITE(*,905) ISTEP,TIME,EROR1
C WRITE(10,905) ISTEP,TIME,EROR1

******************************************************************************
****** 10.0 PRINT THE DESIRED CONTOUR PLOTS AND MATRIX VALUES ******
****** FOR CONSTANT X PLANES (LATERAL PLOTS) ******
******************************************************************************

1000 CONTINUE
1001 FORMAT(//,1X,'|',44('-'),' RESULTS AT I = ',I3,' |',44('-'),'|',/)  
1002 FORMAT(//,2X,A10,/)  
1003 FORMAT(2X,20(F9.3))  
1004 FORMAT(7X,'I = ',I3,' LATERAL PLOT -- ',A10,//)  
NPLOT = 0
DO 1090 ILAT=1,NUMLATS
I=NLATI(ILAT)
   C WRITE(10,1001) I
   DO 1006 N=1,8
   IF (NOPT(N,1).NE.1) GO TO 1006
   C WRITE(10,1002) PLABL(N)
   C DO 1005 JJ=1,JMAX
   C     J=JMAX-JJ+1
   C     IF (N.EQ.1) WRITE(10,1003) (U(I,J,K),K=1,KMAX)
   C     IF (N.EQ.2) WRITE(10,1003) (V(I,J,K),K=1,KMAX)
   C     IF (N.EQ.3) WRITE(10,1003) (W(I,J,K),K=1,KMAX)
   C     IF (N.EQ.4) WRITE(10,1003)
   (DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0)
+W(I,J,K)**2.0),K=1,KMAX)
C    IF (N.EQ.5) WRITE(10,1003) (S(I,J,K),K=1,KMAX)
C    IF (N.EQ.6) WRITE(10,1003) (T(I,J,K),K=1,KMAX)
C    IF (N.EQ.7) WRITE(10,1003) (TK(I,J,K),K=1,KMAX)
C    IF (N.EQ.8) WRITE(10,1003) (TE(I,J,K),K=1,KMAX)
C 1005 CONTINUE
1006 CONTINUE
1030 N=1,8
XRANGE=ZM(KMM1)-ZM(1)
YRANGE=YM(JMM1)-YM(1)
ISIZE=KMAX
JSIZE=JMAX
IF (NOPT(N,2).NE.1) GO TO 1030
DO 1007 J=1,JMAX
DO 1007 K=1,KMAX
IF (N.EQ.1) F1(K,J)=U(I,J,K)
IF (N.EQ.2) F1(K,J)=V(I,J,K)
IF (N.EQ.3) F1(K,J)=W(I,J,K)
IF (N.EQ.4)
F1(K,J)=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0+
   W(I,J,K)**2.0)
IF (N.EQ.5) F1(K,J)=S(I,J,K)
IF (N.EQ.6) F1(K,J)=T(I,J,K)
IF (N.EQ.7) F1(K,J)=TK(I,J,K)
IF (N.EQ.8) F1(K,J)=TE(I,J,K)
1007 CONTINUE
    CALL CPLOT(F1,ISIZE,JSIZE,XRANGE,YRANGE,1,0,0,0)
C    WRITE(10,1004) I,PLABL(N)
C***********************************************************************
C************
*** 10.1 PRINT 2-D NCSA IMAGE3.1.1 LATERAL PLOTS
***
C***********************************************************************
************
AMAX=F1(1,1)
AMIN=F1(1,1)
NPLOT = NPLOT + 1
DO 1010 K=1,KMAX
DO 1010 J=1,JMAX
IF (AMIN.GE.F1(K,J)) AMIN=F1(K,J)
1010 IF (AMAX.LE.F1(K,J)) AMAX=F1(K,J)
OPEN(UNIT=9,FILE=HDFLAT(NPLOT))
WRITE(9,*), JMM2, KMM2
WRITE(9,*), AMAX, AMIN
WRITE(9,1398)(Y(J),J=2,JMM1)
WRITE(9,1398)(Z(I),K=2,KMM1)
DO 1020 J=JMM1,2,-1
WRITE(9,1399)(F1(I,J),K=KMM1,2,-1)
1020 CONTINUE
CLOSE(9)

**********************************************************
************
*** 10.2 PRINT 2-D SYSTAT LATERAL PLOTS ***
************
**********************************************************
************
OPEN(UNIT=2,FILE=SYSLAT(NPLOT))
DO 1028 K=KMM1,2,-2
DO 1028 J=2,JMM1,2
1028 WRITE(2,1099)Z(K),Y(J),F1(K,J)
CLOSE(2)
1030 CONTINUE
1090 CONTINUE
1099 FORMAT(1X,3(F7.4,3X))
**********************************************************
************
*** 11.0 PRINT THE DESIRED CONTOUR PLOTS AND MATRIX VALUES ***
FOR CONSTANT Z PLANES (LONGITUDINAL PLOTS) ***
**********************************************************
1100 CONTINUE
1101 FORMAT(//,1X,'|',44('-'),' RESULTS AT K = ',I3,'|',44('-'),'|',/)
1103 FORMAT(2X,20(F9.3))
1104 FORMAT(7X,'K = ',I3,' LONGITUDINAL PLOT -- ',A10,/'
NPLOT = 0
DO 1190 KLONG=1,NUMLONGS
K=NLONGK(KLONG)
IF (K.EQ.1) GO TO 1190
1190 CONTINUE
C WRITE(10,1101) K

DO 1106 N=1,8
   IF (NOPT(N,1).NE.1) GO TO 1106
   WRITE(10,1002) PLABL(N)
C DO 1105 JJ=1,JMAX
   J=JMAX-JJ+1
   IF (N.EQ.1) WRITE(10,1103) (U(I,J,K),I=1,IMAX)
   IF (N.EQ.2) WRITE(10,1103) (V(I,J,K),I=1,IMAX)
   IF (N.EQ.3) WRITE(10,1103) (W(I,J,K),I=1,IMAX)
   IF (N.EQ.4) WRITE(10,1103) (DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0
C +W(I,J,K)**2.0),I=1,IMAX)
   IF (N.EQ.5) WRITE(10,1103) (S(I,J,K),I=1,IMAX)
   IF (N.EQ.6) WRITE(10,1103) (T(I,J,K),I=1,IMAX)
   IF (N.EQ.7) WRITE(10,1103) (TK(I,J,K),I=1,IMAX)
   IF (N.EQ.8) WRITE(10,1103) (TE(I,J,K),I=1,IMAX)
C 1105 CONTINUE
1106 CONTINUE

DO 1130 N=1,8
   XRANGE=XM(IMM1)-XM(1)
   YRANGE=YM(JMM1)-YM(1)
   ISIZE=IMAX
   JSIZE=JMAX
   IF (NOPT(N,2).NE.1) GO TO 1130
   DO 1108 I=1,IMAX
      DO 1108 J=1,JMAX
         IF (N.EQ.1) F1(I,J)=U(I,J,K)
         IF (N.EQ.2) F1(I,J)=V(I,J,K)
         IF (N.EQ.3) F1(I,J)=W(I,J,K)
         IF (N.EQ.4) F1(I,J)=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0
C +W(I,J,K)**2.0)
         IF (N.EQ.5) F1(I,J)=S(I,J,K)
         IF (N.EQ.6) F1(I,J)=T(I,J,K)
         IF (N.EQ.7) F1(I,J)=TK(I,J,K)
         IF (N.EQ.8) F1(I,J)=TE(I,J,K)
1108 CONTINUE
   CALL CPLOT(F1,ISIZE,JSIZE,XRANGE,YRANGE,1,0,0,0,1.0)
   WRITE(10,1104) K,PLABL(N)
** 11.1 PRINT 2-D IMAGE LONGITUDINAL PLOTS

```
AMAX=F1(1,1)
AMIN=F1(1,1)
NPLOT = NPLOT + 1
DO 1110 I=1,IMAX
   DO 1110 J=1,JMAX
      IF (AMIN.GE.F1(I,J)) AMIN=F1(I,J)
   1110 IF (AMAX.LE.F1(I,J)) AMAX=F1(I,J)
OPEN(UNIT=9,FILE=HDFLON(NPLOT))
WRITE(9,*) JMM2, IMM2
WRITE(9,*) AMAX, AMIN
WRITE(9,1398)(Y(J),J=2,JMM1)
WRITE(9,1398)(X(I),I=2,IMM1)
DO 1120 J=JMM1,2,-1
   WRITE(9,1399)(F1(I,J),I=2,IMM1)
  1120 CONTINUE
CLOSE(9)
```

** 11.2 PRINT 2-D SYSTAT LONGITUDINAL PLOTS

```
OPEN(UNIT=2,FILE=SYSLON(NPLOT))
DO 1128 I=2,IMM1,2
   DO 1128 J=2,JMM1,2
      WRITE(2,1099)X(I),Y(J),F1(I,J)
  1128 CONTINUE
  1190 CONTINUE
```

** 12.0 PRINT VELOCITY VECTOR FILE

```
C   OPEN(UNIT=3,FILE=FNAMEVEC)
```
C WRITE(3,*) IMM2, JMM2, KMM2
C DO 1200 I=2, IMM1
C XX=X(I)-X(M(I))
C DO 1200 J=2, JMM1
C YY=Y(J)-Y(M(J))
C DO 1200 K=2, KMM1
C ZZ=Z(K)-Z(M(K))
C VMAG=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0+W(I,J,K)**2.0)
C THETEMP=T(I,J,K)
C 1200 WRITE(3,915)
XX, YY, ZZ, VMAG, THETEMP, TK(I,J,K), TE(I,J,K)
C CLOSE(3)
******************************************************************************
************
*** 13.0 PRINT 3D DATA FILE FOR NCSA XDATASLICE
***
************
******************************************************************************

NPLOT=0
DO 1330 N=1, 8
IF (NOPT(N,3).NE.1) GO TO 1330
DO 1300 I=2, IMM1
DO 1300 J=2, JMM1
DO 1300 K=2, KMM1
IF (N.EQ.1) F3(I,J,K)=U(I,J,K)
IF (N.EQ.2) F3(I,J,K)=V(I,J,K)
IF (N.EQ.3) F3(I,J,K)=W(I,J,K)
IF (N.EQ.4) F3(I,J,K)=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0+W(I,J,K)**2.0)
IF (N.EQ.5) F3(I,J,K)=S(I,J,K)
IF (N.EQ.6) F3(I,J,K)=T(I,J,K)
IF (N.EQ.7) F3(I,J,K)=TK(I,J,K)
IF (N.EQ.8) F3(I,J,K)=TE(I,J,K)
1300 CONTINUE
AMAX=F3(2,2,2)
AMIN=F3(2,2,2)
DO 1310 I=2, IMM1
DO 1310 J=2, JMM1
DO 1310 K=2, KMM1
IF (AMAX .LE. F3(I,J,K)) AMAX=F3(I,J,K)
IF (A\textsubscript{MIN} .GT. F3(I,J,K)) A\textsubscript{MIN}=F3(I,J,K)

1310 CONTINUE
NPLOT = NPLOT + 1
OPEN(UNIT=12,FILE=HDF3D(NPLOT))
WRITE(12,1396)'TEXT'
WRITE(12,1397) KMM2, JMM2, IMM2
WRITE(12,1399) A\textsubscript{MAX}, A\textsubscript{MIN}
WRITE(12,1398) (Z(K),K=2,KMM1)
WRITE(12,1398) (Y(J),J=2,JMM1)
WRITE(12,1398) (X(I),I=2,IMM1)
DO 1320 K=KMM1,2,-1
DO 1320 J=2,JMM1
WRITE(12,1399) (F3(I,J,K),I=2,IMM1)
1320 CONTINUE

1330 CONTINUE
CLOSE (12)
1330 CONTINUE
1396 FORMAT(A4)
1397 FORMAT(I2,1X,I2,1X,I2)
1398 FORMAT(50(F5.3,1X))
1399 FORMAT(50(F7.3,1X))

************************************************************
************
***
*** 14.0 TERMINATE THE PROGRAM
***
***
***
************************************************************
************
CLOSE(10)
CLOSE(4)
STOP
END

************************************************************
************
** SUBROUTINE: CPLOT
**
** PURPOSE: CREATE CONTOUR PLOTS OF DATA **

******************************************************************************
************
SUBROUTINE CPLOT(Q,IMAX,JMAX,XRANGE,YRANGE,NOPT1,NOPT2,NOPT3,
C     NOPT4)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION Q(50,50),QP(50,50),RANGES(21),QHOLD(50)
CHARACTER*1 GRAPH(130),PCHAR(12),BORDER(130)

DATA PCHAR /' ','E',' ','D',' ','C',' ','B',' ','A',' ','#'/
DATA BORDER,GRAPH /260*'#'/
DATA NADJ /0/
DATA RANGES
/0.01,0.1,0.5,1.0,5.0,10.,20.,50.,100.,150.,200.,
C
250.,300.,400.,500.,600.,800.,1000.,2000.,5000.,10000./
DATA QHOLD,QP /2550*0.0/

50 FORMAT(/,10X,F9.3,' < ',A1,' < 
   F9.3,/,10X,F9.3,' < ',A1,
C     ' < ',F9.3,/,10X,F9.3,' < ',A1,' < 
   ',F9.3,/,10X,F9.3,' < ',A1,' < 
C   ',F9.3,/,10X,F9.3,') < ',A1,' < 
   ',F9.3)
52 FORMAT(2X,130(A1))
60 FORMAT(/,10X,'(DOMAIN RATIO: ','F7.4,' - ACTUAL 
   RATIO: ',
C     'F7.4,')')
******************************************************************************
************
** NOPT1 1 - ONLY INTERNAL DOMAIN IS CONSIDERED **
** 0 - ENTIRE DOMAIN IS CONSIDERED **
******************************************************************************
QMAX=-1.0E6
IF (NOPT1.EQ.1) NADJ=1
INTX=IMAX-2*NADJ-1
INTY=JMAX-2*NADJ-1
DO 100 I=1,(INTX+1)
II=I+NADJ
DO 100 J=1,(INTY+1)
JJ=J+NADJ
QP(I,J)=Q(II,JJ)
IF (QP(I,J).GT.QMAX) QMAX=QP(I,J)
100 CONTINUE

**********************************************************
************
** NOPT2 1 - MIRRORS X-DOMAIN
**
** 0 - NORMAL X-DOMAIN
**
**********************************************************
************
IF (NOPT2.EQ.1) THEN
DO 120 J=1,(INTY+1)
DO 125 I=1,(INTX+1)
125 QHOLD(I)=QP(I,J)
DO 120 I=1,(INTX+1)
II=INTX+2-I
QP(II,J)=QHOLD(I)
120 CONTINUE
ENDIF

**********************************************************
************
** NOPT3 1 - MIRRORS Y-DOMAIN
**
** 0 - NORMAL Y-DOMAIN
**
**********************************************************
************
IF (NOPT3.EQ.1) THEN
DO 130 I=1,(INTX+1)
DO 135 J=1,(INTY+1)
135 QHOLD(J)=QP(I,J)
DO 130 J=1,(INTY+1)
JJ=INTY+2-J
QP(I,JJ)=QHOLD(J)
** NOPT4 0 - PLOT SIZE DETERMINED BY SUBROUTINE
** 1 - SMALL PLOT SIZE ( 80 x 48 )
** 2 - LARGE PLOT SIZE ( 120 x 72 )

**

NXP=80
NYP=48
IF ((NOPT4.EQ.0).AND.((YRANGE/XRANGE).GT.1.5)) THEN
  NXP=120
  NYP=72
ENDIF
IF (NOPT4.EQ.2) THEN
  NXP=120
  NYP=72
ENDIF

RATIO=XRANGE/YRANGE
IF (RATIO.GT.3.0) RATIO=3.0
IF (RATIO.LT.(1./3.0)) RATIO=1./3.0
IF (XRANGE.GT.YRANGE) NYP=INT(NYP/RATIO)
IF (YRANGE.GT.XRANGE) NXP=INT(NXP*RATIO)

**

DO 150 N=21,1,-1
  IF (0.98*QMAX.LE.RANGES(N)) NRANGE=N
150 CONTINUE
QRANGE=RANGES(NRANGE)

WRITE(10,50)
(0.9*QRANGE),PCHAR(10),QRANGE,(0.7*QRANGE),PCHAR(8),
C
C \( c(0.8*QRANGE), (0.5*QRANGE), PCHAR(6), (0.6*QRANGE), (0.3*QRANGE) \),
C
C \( CPCHAR(4), (0.4*QRANGE), (0.1*QRANGE), PCHAR(2), (0.2*QRANGE) \)
C
IF (RATIO.NE.(XRANGE/YRANGE)) THEN
WRITE(10,60) RATIO,(XRANGE/YRANGE)
ENDIF

******************************************************************************
******
** PRINT CONTOUR PLOT
**
******************************************************************************
C
WRITE(10,52) (BORDER(N),N=1,(NXP+6))
C
DO 200 J=NYP,1,-1
    Y=(J-0.5)*YRANGE/REAL(NYP)
    JP=INT(Y/(YRANGE/INTY))+1
    JP1=JP+1
DO 250 I=1,NXP
    X=(I-0.5)*XRANGE/REAL(NXP)
    IP=INT(X/(XRANGE/INTX))+1
    IP1=IP+1
    XL1=X-(IP-1)*XRANGE/INTX
    XL2=IP*XRANGE/INTX-X
    YL1=Y-(JP-1)*YRANGE/INTY
    YL2=JP*YRANGE/INTY-Y
    A1=XL2*YL2
    A2=XL1*YL2
    A3=XL1*YL1
    A4=XL2*YL1
    A=A1+A2+A3+A4
    NPRINT=INT((0.98*QQ/QRANGE)*10.0)+1
    GRAPH(I+3)=PCHAR(NPRINT)
250 CONTINUE
C
WRITE(10,52) (GRAPH(N),N=1,(NXP+6))
200 CONTINUE
SUBROUTINE CCOEF

IMPLICIT REAL*8 (A-H,O-Z)

COMMON
1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50),
4 XDIMEN,YDIMEN,ZDIMEN

1/TEMPS/T(50,50,50),TN(50,50,50),QT0,QTIN,QTWALL,QTCOND
DIMENSION FLUX(50,50)

TNUMOUT=(T(IMM1,JTREF1,KTREF1)+T(IMM1,JTREF2,KTREF1)+
+ T(IMM1,JTREF1,KTREF2)+T(IMM1,JTREF2,KTREF2))/4.0

FTOT=0.0
DO 1500 I=2,IMM1
DO 1500 J=2,JMM1
FLUX(I,J)=-QTCOND*(T(I,J,2)-T(I,J,1))/DZ(1)
1500 FTOT=FTOT+FLUX(I,J)
QAVG=FTOT/(XDIMEN*YDIMEN)
HNUM=QAVG/(QTWALL-TNUMOUT)
RETURN
END
** SUBROUTINE: GERROR
**
** PURPOSE: CALCULATE THE GLOBAL ERROR NUMBER (GENV) & **
** THE GLOBAL TEMPERATURE ERROR NUMBER (GENT) **
**-------------------------------------------------------------
**
SUBROUTINE GERROR(TIME,EXPFILE,TEXPFILE)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON 1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50),
4 XDIMEN,YDIMEN,ZDIMEN
1/VELS/U(50,50,50),V(50,50,50),W(50,50,50),UN(50,50,50),
2 VN(50,50,50),WN(50,50,50)
1/TEMPS/T(50,50,50),TN(50,50,50),QT0,QTIN,QTWALL,QTCOND
DIMENSION IVL(15),JVL(15),KVL(15),XMD(50),YMD(50),ZMD(50)
DIMENSION EVM(15,15,15),ETP(15,15,15),AVM(15,15,15)
DIMENSION DI(15),DJ(15),DK(15),VM(50,50,50),ATP(15,15,15)
DIMENSION AVM2(15,15,15),ATP2(15,15,15)
CHARACTER*20 EXPFILE,TEXPFILE
DO 100 I=1,IMAX
DO 100 J=1,JMAX
DO 100 K=1,KMAX
100 VM(I,J,K)=DSQRT(U(I,J,K)**2.0+V(I,J,K)**2.0+W(I,J,K)**2.0)
C WRITE(*,499)(VM(I,J,K),I=2,IMM1)
DO 430 I=1,IMAX
XMD(I)=X(I)-XM(1)
YMD(I)=Y(I)-YM(1)
430 ZMD(I)=Z(I)-ZM(1)
497 FORMAT(I2,1X,I2,1X,I2)
OPEN(UNIT=12,FILE=EXPFILE)
READ(12,*)
READ(12,497) KEMAX, JEMAX, IEMAX
READ(12,499) AMAX, AMIN
READ(12,498)(DK(K),K=1,KEMAX)
READ(12,498)(DJ(J),J=1,JEMAX)
READ(12,498)(DI(I),I=1,IEMAX)
DO 500 K=KEMAX,1,-1
DO 500 J=1,JEMAX
READ(12,499)(EVM(I,J,K),I=1,IEMAX)
500 CONTINUE
CLOSE (12)

OPEN(UNIT=12,FILE=TEXPFILE)
READ(12,*)
READ(12,497) KEMAX, JEMAX, IEMAX
READ(12,499) ATMAX, ATMIN
READ(12,498)(DK(K),K=1,KEMAX)
READ(12,498)(DJ(J),J=1,JEMAX)
READ(12,498)(DI(I),I=1,IEMAX)
DO 501 K=KEMAX,1,-1
DO 501 J=1,JEMAX
READ(12,499)(ETP(I,J,K),I=1,IEMAX)
501 CONTINUE
CLOSE (12)

DO 802 I=1,IEMAX
  J=1
800 IF (DI(I).GT.XMD(J)) THEN
    J=J+1
    GO TO 800
ENDIF
IVL(I)=J
IF((XMD(J)-DI(I)) .GT. DX(J-1)/2.0) IVL(I)=J-1
802 CONTINUE
DO 806 I=1,JEMAX
  J=1
804 IF (DJ(I).GT.YMD(J)) THEN
    J=J+1
806 CONTINUE
GO TO 804
ENDIF
JVL(I)=J
IF((YMD(J)-DJ(I)) GT. DY(J-1)/2.0) JVL(I)=J-1
806 CONTINUE
DO 812 I=1,KEMAX
  J=1
810 IF (DK(I).GT.ZMD(J)) THEN
    J=J+1
    GO TO 810
ENDIF
KVL(I)=J
IF((ZMD(J)-DK(I)) GT. DZ(J-1)/2.0) KVL(I)=J-1
812 CONTINUE

GENV=0.0
TOTERR=0.0
GENT=0.0
TPTERR=0.0
DO 830 I=1,IEMAX
  DO 830 J=1,JEMAX
    DO 830 K=1,KEMAX
      DISTX=XMD(IVL(I))-XMD(IVL(I)-1)
      XDIST=XMD(IVL(I))-DI(I)
      XD=XDIST/DISTX
      DISTY=YMD(JVL(J))-YMD(JVL(J)-1)
      YDIST=YMD(JVL(J))-DJ(J)
      YD=YDIST/DISTY
      DISTZ=ZMD(KVL(K))-ZMD(KVL(K)-1)
      ZDIST=ZMD(KVL(K))-DK(K)
      ZD=ZDIST/DISTZ
      VM1=(VM(IVL(I),JVL(J),KVL(K))*(1-XD)) +
           (VM(IVL(I)-1,JVL(J),KVL(K))*XD)
      VM2=(VM(IVL(I),JVL(J)-1,KVL(K))*(1-XD)) +
           (VM(IVL(I)-1,JVL(J)-1,KVL(K))*XD)
      VM3=(VM(IVL(I),JVL(J),KVL(K)-1)*(1-XD)) +
           (VM(IVL(I)-1,JVL(J),KVL(K)-1)*XD)
      VM4=(VM(IVL(I),JVL(J)-1,KVL(K)-1)*(1-XD)) +
           (VM(IVL(I)-1,JVL(J)-1,KVL(K)-1)*XD)
      AVM(I,J,K)=((VM1*(1-YD)+VM2*YD)*(1-ZD) +
                  (VM3*(1-YD)+VM4*YD)*ZD)
      AVM2(I,J,K)=VM(IVL(I),JVL(J),KVL(K))
TP1 = (T(IVL(I), JVL(J), KVL(K)) * (1 - XD)) +
+ (T(IVL(I) - 1, JVL(J), KVL(K)) * XD)
TP2 = (T(IVL(I), JVL(J) - 1, KVL(K)) * (1 - XD)) +
+ (T(IVL(I) - 1, JVL(J) - 1, KVL(K)) * XD)
TP3 = (T(IVL(I), JVL(J), KVL(K) - 1) * (1 - XD)) +
+ (T(IVL(I) - 1, JVL(J), KVL(K) - 1) * XD)
TP4 = (T(IVL(I), JVL(J) - 1, KVL(K) - 1) * (1 - XD)) +
+ (T(IVL(I) - 1, JVL(J) - 1, KVL(K) - 1) * XD)

ATP(I, J, K) = ((TP1 * (1 - YD) + TP2 * YD) * (1 - ZD) +
+ (TP3 * (1 - YD) + TP4 * YD) * ZD)

ATP2(I, J, K) = T(IVL(I), JVL(J), KVL(K))

TPTERR = ABS(ATP(I, J, K) - ETP(I, J, K)) + TPTERR

830 TOTERR = ABS(AVM(I, J, K) - EVM(I, J, K)) + TOTERR

C TOTERR = ABS(0.0 - EVM(I, J, K)) + TOTERR
N = 896
GENV = TOTERR / (N * AMAX)
GENT = TPTERR / (N * ATMAX)

OPEN(UNIT=5, FILE='genv.out', STATUS='OLD')
OPEN(UNIT=9, FILE='gent.out', STATUS='OLD')
WRITE(5, 906) TIME, GENV, TOTERR, AMAX
WRITE(9, 906) TIME, GENT, TPTERR, ATMAX

C WRITE(*,*) TOTERR, GENV, GENT

900 FORMAT(3(F5.3, 2X), 4(F7.3, 2X))
905 FORMAT(F5.3, 2X, F5.3, 2X, F5.3, 2X, F5.3, 2X, F8.3, 2X, F8.3)
906 FORMAT(F8.3, ',', 2X, F10.8, ',', 2X, F7.3, ',', 2X, F6.3)
RETURN
END

*******************************************************************************************************

************
** SUBROUTINE: STRAT
**
**
** PURPOSE: CALCULATE STRATIFICATION VALUES
**
***********************************************************************

SUBROUTINE STRAT(TIME, ISTRAT)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
1/GEOM/IMAX, IMM1, JMAX, JMM1, KMAX, KMM1, DX(50), DXM(50),
DIMENSION STRATIF(900,50),TOTTP(50)

DO 100 J=1,JMAX
100 TOTTP(J)=0.0

OPEN(UNIT=11,FILE='strat.dat',STATUS='OLD')
IF (ISTRAT.EQ.1) THEN
   DO 300 J=2,JMM1
300 WRITE(11,905)Y(J)
ENDIF

DO 500 J=2,JMM1
   DO 500 I=2,IMM1
      DO 500 K=2,KMM1
500 TOTTP(J)=T(I,J,K)+TOTTP(J)
   STRATIF(ISTRAT,1)=TIME
   DO 502 J=2,JMM1
502 STRATIF(ISTRAT,J)=TOTTP(J)/((IMM1-1)*(KMM1-1))
   IF (ISTRAT/10*10.NE.ISTRAT) GO TO 907
   WRITE(11,*)
   DO 506 J=1,JMM1
506 WRITE(11,906)(STRATIF(I,J), I=ISTRAT-9,ISTRAT)
905 FORMAT(F6.4)
906 FORMAT(10(1X,F7.4))
907 RETURN
END

***********************************************************************************************
************
** SUBROUTINE: UPDATEBC
**
**
** PURPOSE: UPDATES AND IMPOSES BOUNDARY CONDITIONS
**
***********************************************************************************************

SUBROUTINE UPDATEBC(NFLOW,VIS,QTCOND)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
  1/GEOM/IMAX, IMM1, JMAX, JMM1, KMAX, KMM1, DX(50), DXM(50),
  2 X(50), Y(50), Z(50), XM(50), YM(50), ZM(50),
  3 DY(50), DZ(50), DYM(50), DZM(50)

  1/VELS/U(50,50,50), V(50,50,50), W(50,50,50), UN(50,50,50),
  2 VN(50,50,50), WN(50,50,50)
  1/MIX/S(50,50,50), SN(50,50,50)
  1/NBCS/NSI1, NSIMAX, NSJ1, NSJMAX, NSK1, NSKMAX

  1/INNOUT/NINCELS, NOUTCELS, JIN(90), KIN(90), JOUT(90), KOUT(90)
  1/TREFCEL/JTREF1, JTREF2, KTREF1, KTREF2
  1/JETS/ACH, UJET, UJOUT, UINS(90)
  1/TEMPS/T(50,50,50), TN(50,50,50), QT0, QTIN, QTWALL

  1/TURB/TK(50,50,50), TKN(50,50,50), TE(50,50,50), TEN(50,50,50),
  2 TVIS(50,50,50)
  1/TCOEFF/TCFCMU, TCFC1, TCFC2, TSIGK, TSIGE
  1/TURBBC/TKIN(90), TEIN(90)
  1/FILM/FILMW, FILMF, FILMC, NTEMP

    IMM2=IMM1-1
    JMM2=JMM1-1
    KMM2=KMM1-1
    PLAW=1.0/7.0
    TREF=(T(IMM1,JTREF1,KTREF1)+T(IMM1,JTREF2,KTREF1)+
    + T(IMM1,JTREF1,KTREF2)+T(IMM1,JTREF2,KTREF2))/4.0

******************************************************************************
******************************************************************************
*** 1.0 WALLS AT K=1 AND K=KMAX
***
******************************************************************************
******************************************************************************

    DO 100 J=1, JMAX
    DO 100 I=1, IMAX

    W(I,J,1)=0.0
    S(I,J,1)=S(I,J,2)
C T(I,J,1)=T(I,J,2)
  T(I,J,1)=QTWALL
IF (NTEMP.NE.1) THEN
  IF (NTEMP.EQ.2) T(I,J,1)=2*QTWALL-T(I,J,2)
  IF (NTEMP.EQ.3) T(I,J,1)=(QTWALL-TREF)*FILMW*DZ(1)/QTCOND+T(I,J,2)
ENDIF
IF (NFLOW.NE.1) THEN
  TK(I,J,1)=TK(I,J,2)
  TE(I,J,1)=TE(I,J,2)
  IF (NFLOW.EQ.2)
    TE(I,J,1)=(2.0*(TCFCMU**0.75)*(TK(I,J,2)**1.5)
               + / (0.4*DZM(2)/2.0)) - TE(I,J,2)
  C IF (NFLOW.EQ.3) TE(I,J,1)= 4.0*VIS*TK(I,J,2)
  C + /((DZM(2)/2.0)**2.0)-
  TE(I,J,2)
ENDIF
U(I,J,1)=-U(I,J,2)
IF (NSK1.EQ.1) U(I,J,1)=U(I,J,2)
IF (NSK1.EQ.2) U(I,J,1)=(1.0-(PLAW*DZ(1))/(0.5*DZM(2))) *U(I,J,2)
V(I,J,1)=-V(I,J,2)
IF (NSK1.EQ.1) V(I,J,1)=V(I,J,2)
IF (NSK1.EQ.2) V(I,J,1)=(1.0-(PLAW*DZ(1))/(0.5*DZM(2))) *V(I,J,2)
C
W(I,J,KMM1)=0.0
W(I,J,KMAX)=0.0
S(I,J,KMAX)=S(I,J,KMM1)
C
T(I,J,KMAX)=T(I,J,KMM1)
T(I,J,KMAX)=QTWALL
IF (NTEMP.NE.1) THEN
  IF (NTEMP.EQ.2) T(I,J,KMAX)=2*QTWALL-T(I,J,KMM1)
  IF (NTEMP.EQ.3) T(I,J,KMAX)=(QTWALL-TREF)*FILMW*DZ(KMM1)/QTCOND+T(I,J,KMM1)
ENDIF
IF (NFLOW.NE.1) THEN
  TK(I,J,KMAX)=TK(I,J,KMM1)
TE(I,J,KMAX)=TE(I,J,KMM1)

IF (NFLOW.EQ.2) TE(I,J,KMAX)=(2.0*(TCFCMU**0.75) +
      (TK(I,J,KMM1)**1.5)/(0.4*DZM(2)/2.0)) -
      TE(I,J,KMM1)
C   IF (NFLOW.EQ.3) TE(I,J,KMAX)=4.*VIS*
C         TK(I,J,KMM1)/((DZM(2)/2.)**2.)- TE(I,J,KMM1)
ENDIF

U(I,J,KMAX)=-U(I,J,KMM1)
IF(NSKMAX.EQ.1) U(I,J,KMAX)=U(I,J,KMM1)
IF(NSKMAX.EQ.2) U(I,J,KMAX)=(1.0-(PLAW*DZ(KMM1))/
      +
      (0.5*DZM(KMM1))) *U(I,J,KMM1)
V(I,J,KMAX)=-V(I,J,KMM1)
IF(NSKMAX.EQ.1) V(I,J,KMAX)=V(I,J,KMM1)
IF(NSKMAX.EQ.2) V(I,J,KMAX)=(1.0-(PLAW*DZ(KMM1))/
      +
      (0.5*DZM(KMM1))) *V(I,J,KMM1)

100 CONTINUE

**********************************************************
************
*** 2.0 WALLS AT J=1 AND J=JMAX
***
************

*****************************************************************
DO 200 I=1,IMAX
DO 200 K=1,KMAX

V(I,1,K)=0.0
S(I,1,K)=S(I,2,K)
C T(I,1,K)=T(I,2,K)
T(I,1,K)=QTWALL
IF (NTEMP.NE.1) THEN
  IF (NTEMP.EQ.2) T(I,1,K)=2*QTWALL-T(I,2,K)
  IF (NTEMP.EQ.3) T(I,1,K)=(QTWALL-
      TREF)*FILMF*DY(1)/QTCOND+T(I,2,K)
ENDIF

IF (NFLOW.NE.1) THEN
  TK(I,1,K)=TK(I,2,K)
  TE(I,1,K)=TE(I,2,K)
IF (NFLOW.EQ.2)
    TE(I,1,K) = (2.0* (TCFCMU**0.75) * (TK(I,2,K)**1.5)
    + (0.4*DYM(2)/2.0)) -
    TE(I,2,K)
C IF (NFLOW.EQ.3) TE(I,1,K) = 4.0*VIS*TK(I,2,K)/
    C + ((DYM(2)/2.)**2.1 - TE(I,2,K)
ENDIF
U(I,1,K) = -U(I,2,K)
IF (NSJ1.EQ.1) U(I,1,K) = U(I,2,K)
IF (NSJ1.EQ.2) U(I,1,K) = (1.0 -
    (PLAW*DY(1)) / (0.5*DYM(2))) * U(I,2,K)

W(I,1,K) = -W(I,2,K)
IF (NSJ1.EQ.1) W(I,1,K) = W(I,2,K)
IF (NSJ1.EQ.2) W(I,1,K) = (1.0 -
    (PLAW*DY(1)) / (0.5*DYM(2))) * W(I,2,K)

V(I,JMM1,K) = 0.
V(I,JMAX,K) = 0.
S(I,JMAX,K) = S(I,JMM1,K)
C T(I,JMAX,K) = T(I,JMM1,K)
T(I,JMAX,K) = QTWALL
IF (NTEMP.NE.1) THEN
    IF (NTEMP.EQ.2) T(I,JMAX,K) = 2*QTWALL - T(I,JMM1,K)
    IF (NTEMP.EQ.3) T(I,JMAX,K) = (QTWALL -
        TREF) * FILMC*DY(JMM1)/QTCOND
        C + T(I,JMM1,K)
ENDIF

IF (NFLOW.NE.1) THEN
    TK(I,JMAX,K) = TK(I,JMM1,K)
    TE(I,JMAX,K) = TE(I,JMM1,K)
    IF (NFLOW.EQ.2) TE(I,JMAX,K) = (2.0* (TCFCMU**0.75) *
        + (TK(I,JMAX,K)**1.5) / (0.4*DYM(2)/2.0)) -
        TE(I,JMM1,K)
C IF (NFLOW.EQ.3) TE(I,JMAX,K) = 4.0*VIS*TK(I,JMM1,K)
C + ((DYM(2)/2.)**2.1 - TE(I,JMM1,K)
ENDIF
U(I,JMAX,K) = -U(I,JMM1,K)
IF (NSJMAX.EQ.1) U(I,JMAX,K) = U(I,JMM1,K)
IF (NSJMAX.EQ.2) U(I,JMAX,K) = (1.0 - (PLAW*DY(JMM1))) / + (0.5*DYM(JMM1)) * U(I,JMM1,K)

W(I,JMAX,K) = -W(I,JMM1,K)
IF (NSJMAX.EQ.1) W(I,JMAX,K) = W(I,JMM1,K)
IF (NSJMAX.EQ.2) W(I,JMAX,K) = (1.0 - (PLAW*DY(JMM1))) / + (0.5*DYM(JMM1)) * W(I,JMM1,K)

200 CONTINUE
********************************************************************
************
*** 3.0 WALLS AT I=1 AND I=IMAX
***
********************************************************************
************
DO 300 J=1,JMAX
DO 300 K=1,KMAX

U(1,J,K) = 0.0
S(1,J,K) = S(2,J,K)

T(1,J,K) = T(2,J,K)
T(1,J,K) = QTWALL
IF (NTEMP.NE.1) THEN
IF (NTEMP.EQ.2) T(1,J,K) = 2*QTWALL - T(2,J,K)
ENDIF
IF (NFLOW.NE.1) THEN
TK(1,J,K) = TK(2,J,K)
TE(1,J,K) = TE(2,J,K)
IF (NFLOW.EQ.2) TE(1,J,K) = (2.0*(TCFCMU**0.75)*(TK(2,J,K)**1.5) + /(0.4*DXM(2)/2.0)) - TE(2,J,K)
C IF (NFLOW.EQ.3)
TE(1,J,K) = 4.0*VIS*TK(2,J,K)/((DXM(2)/2.0)**2.0) + - TE(2,J,K)
ENDIF

END
\[ V(1, J, K) = -V(2, J, K) \]
\[ \text{IF (NSI1.EQ.1) } V(1, J, K) = V(2, J, K) \]
\[ \text{IF (NSI1.EQ.2) } V(1, J, K) = (1.0 - (PLAW \times DX(1)) \div (0.5 \times DXM(2))) \times V(2, J, K) \]

\[ W(1, J, K) = -W(2, J, K) \]
\[ \text{IF (NSI1.EQ.1) } W(1, J, K) = W(2, J, K) \]
\[ \text{IF (NSI1.EQ.2) } W(1, J, K) = (1.0 - (PLAW \times DX(1)) \div (0.5 \times DXM(2))) \times W(2, J, K) \]

\[
\text{DO } 310 \ N=1, \text{NINCELS} \\
\text{IF ( (J.EQ.JIN(N)) .AND. (K.EQ.KIN(N)) ) THEN} \\
\quad U(1, J, K) = UINS(N) \\
\quad S(1, J, K) = 1.0 \\
\quad T(1, J, K) = QTIN \\
\quad TK(1, J, K) = TKIN(N) \\
\quad TE(1, J, K) = TEIN(N) \\
\text{ENDIF} \\
310 \text{ CONTINUE} \\
300 \text{ CONTINUE} \\
\]

\[
\text{DO } 350 \ J=1, \text{JMAX} \\
\text{DO } 350 \ K=1, \text{KMAX} \\
\quad U(IMM1, J, K) = 0.0 \\
\quad U(IMAX, J, K) = 0.0 \\
\quad S(IMAX, J, K) = S(IMM1, J, K) \\
\quad T(IMAX, J, K) = QTWALL \\
\text{IF ( NTEMP.NE.1 ) THEN} \\
\text{IF ( NTEMP.EQ.2 ) } T(1, J, K) = 2 \times QTWALL - T(IMM1, J, K) \\
\text{IF ( NTEMP.EQ.3 ) } T(IMAX, J, K) = (QTWALL - TREF) \times FILMw \times DX(IMM1) \div QTCOND \\
\quad + T(IMM1, J, K) \\
\text{ENDIF} \\
\text{IF ( NFLOW.NE.1 ) THEN} \\
\quad TK(IMAX, J, K) = TK(IMM1, J, K) \\
\quad TE(IMAX, J, K) = TE(IMM1, J, K) \\
\text{IF ( NFLOW.EQ.2 ) } TE(IMAX, J, K) = (2.0 \times (TCFCMU**0.75) \div (0.4 \times DXM(2)/2.0)) \div (0.4 \times DXM(2)/2.0)) - \\
\quad TE(IMM1, J, K) \]
C IF (NFLOW.EQ.3) TE(IMAX,J,K)=4.*VIS*TK(IMM1,J,K) + /((DXM(2)/2.)**2.)- TE(IMM1,J,K)
ENDIF

V(1,J,K)=-V(2,J,K)
IF(NSI1.EQ.1) V(1,J,K)=V(2,J,K)
IF(NSI1.EQ.2) V(1,J,K)=(1.0-2.0*PLAW)*V(2,J,K)

V(IMAX,J,K)=-V(IMM1,2,K)
IF(NSIMAX.EQ.1) V(IMAX,J,K)=V(IMM1,J,K)
IF(NSIMAX.EQ.2) V(IMAX,J,K)=(1.0-(PLAW*DX(IMM1)))/ + (0.5*DXM(IMM1)))*V(IMM1,J,K)

W(IMAX,J,K)=-W(IMM1,J,K)
IF(NSIMAX.EQ.1) W(IMAX,J,K)=W(IMM1,J,K)
IF(NSIMAX.EQ.2) W(IMAX,J,K)=(1.0-(PLAW*DX(IMM1)))/ + (0.5*DXM(IMM1)))*W(IMM1,J,K)

DO 360 N=1,NOUTCELS
IF ((J.EQ.JOUT(N)).AND.(K.EQ.KOUT(N))) THEN
  T(IMAX,J,K)=T(IMM1,J,K)
  S(IMAX,J,K)=S(IMM1,J,K)
  U(IMM1,J,K)=UJOUT
  U(IMAX,J,K)=UJOUT
  TK(IMAX,J,K)=TK(IMM1,J,K)
  TE(IMAX,J,K)=TE(IMM1,J,K)
ENDIF
360 CONTINUE
350 CONTINUE
RETURN
END

********************************************************************
************
** SUBROUTINE: RESID
**
**
** PURPOSE: DETERMINE RESIDUAL VALUES
**
SUBROUTINE RESID(ISTEP,DSUM,IBIG,JBIG,KBIG,THEERR)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50)

1/VELS/U(50,50,50),V(50,50,50),W(50,50,50),UN(50,50,50),
2 VN(50,50,50),WN(50,50,50)

1/MIX/S(50,50,50),SN(50,50,50)

1/JETS/ACH,UJET,UJOUT

1/RESIDUALS/RESU,RESV,RESS,DSUMN,RESUN,RESVN,RESWN,RESSN,
2 DSUM0,RESU0,RESV0,RESS0

RESU=0.0
RESV=0.0
RESS=0.0

VMAXERR=-99.99
THEERR=0.0
DO 100 I=1,IMAX
DO 100 J=1,JMAX
DO 100 K=1,KMAX
RESU=RESU+ABS(U(I,J,K)-UN(I,J,K))
RESV=RESV+ABS(V(I,J,K)-VN(I,J,K))
RESS=RESS+ABS(S(I,J,K)-SN(I,J,K))
VCHANGE=ABS((U(I,J,K)-UN(I,J,K))/UJET)+
        ABS((V(I,J,K)-VN(I,J,K))/UJET)+
        ABS((W(I,J,K)-WN(I,J,K))/UJET)
THEERR=THEERR+VCHANGE
IF (VCHANGE.GT.VMAXERR) THEN
   IBIG=I
   JBIG=J
   KBIG=K
   VMAXERR=VCHANGE
   THEERR=VMAXERR/UJET
ENDIF
100 CONTINUE
   IF (ISTEP.EQ.1) DSUM0=DSUM
   IF (ISTEP.EQ.1) RESU0=RESU
   IF (ISTEP.EQ.1) RESV0=RESV
   IF (ISTEP.EQ.1) RESW0=RESW
   IF (ISTEP.EQ.1) RESS0=RESS
   DSUMN=DSUM/DSUM0
   RESUN=RESU/RESU0
   RESVN=RESV/RESV0
   RESWN=RESW/RESW0
   RESSN=RESS/RESS0
   RETURN
END

**********************************************************
************
** SUBROUTINE: REPACK
**
**
**
** PURPOSE: RE-PACKAGES THE VARIABLES
**
**********************************************************
SUBROUTINE REPACK
   IMPLICIT REAL*8 (A-H,O-Z)
   COMMON
      1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
          2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
          3 DY(50),DZ(50),DYM(50),DZM(50)
1/VELS/U(50,50,50),V(50,50,50),W(50,50,50),UN(50,50,50),
   2 VN(50,50,50),WN(50,50,50)
1/MIX/S(50,50,50),SN(50,50,50)
1/TEMPS/T(50,50,50),TN(50,50,50),QTO,QTIN,QTWALL
1/TURB/TK(50,50,50),TKN(50,50,50),TE(50,50,50),TEN(50,50,50),
   2 TVIS(50,50,50)
   DO 100 I=1,IMAX
   DO 100 J=1,JMAX
DO 100 K=1,KMAX
UN(I,J,K)=U(I,J,K)
VN(I,J,K)=V(I,J,K)
WN(I,J,K)=W(I,J,K)
SN(I,J,K)=S(I,J,K)
TN(I,J,K)=T(I,J,K)
TKN(I,J,K)=TK(I,J,K)
TEN(I,J,K)=TE(I,J,K)
100 CONTINUE
RETURN
END

**********************************************************
************
** SUBROUTINE:  DERIVS
**
**
** PURPOSE:  CALCULATES THE NECESSARY DERIVATIVES
AND
**
** DERIVATIVE TERMS. FORWARD AND BACKWARD
**
** DIFFERENCE METHODS ARE USED TO
APPROXIMATE FIRST
**
** DERIVATIVES, DEPENDING ON THE TIME
STEP.
**
**********************************************************
************
SUBROUTINE DERIVS(IT)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50)

1/VELS/U(50,50,50),V(50,50,50),W(50,50,50),UN(50,50,50),
2 VN(50,50,50),WN(50,50,50)

1/MIX/S(50,50,50),SN(50,50,50)

1/TURB/TK(50,50,50),TKN(50,50,50),TE(50,50,50),TEN(50,50,50),
2 TVIS(50,50,50)
C=REAL(IT-2*INT(IT/2.0))
A=C-1.0
B=(-1.0)**C
DO 100 I=2,IMM1
DO 100 J=2,JMM1
DO 100 K=2,KMM1

DUDX=(A*UN(I-1,J,K)+B*UN(I,J,K)+C*UN(I+1,J,K))/DXM(I+INT(C))
DUDY=(A*UN(I,J-1,K)+B*UN(I,J,K)+C*UN(I,J+1,K))/DY(J+INT(A))
DUDZ=(A*UN(I,J,K-1)+B*UN(I,J,K)+C*UN(I,J,K+1))/DZ(K+INT(A))

DUDX2=((UN(I+1,J,K)-UN(I,J,K))/DXM(I+1)
        -(UN(I,J,K)-UN(I-1,J,K))/DXM(I))/DX(I)
DUDY2=((UN(I,J+1,K)-UN(I,J,K))/DY(J)
        -(UN(I,J,K)-UN(I,J-1,K))/DY(J-1))/DYM(J)
DUDZ2=((UN(I,J,K+1)-UN(I,J,K))/DZ(K)
        -(UN(I,J,K)-UN(I,J,K-1))/DZ(K-1))/DZM(K)

DU2DXDY=(A*((UN(I-1,J-1,K)-UN(I,J-1,K))/DXM(I)
         +(UN(I-1,J,K)-UN(I,J,K))/DXM(I))/DY(J-1))
         +(C*((UN(I+1,J+1,K)-UN(I,J+1,K))/DXM(I+1)
         -(UN(I+1,J,K)-UN(I,J,K))/DXM(I+1))/DY(J))

DU2DZDZ=(A*((UN(I-1,J,K-1)-UN(I,J,K-1))/DZ(K-1)
          -(UN(I-1,J,K)-UN(I,J,K))/DZ(K))/DZM(K)
          -(C*((UN(I+1,J,K+1)-UN(I,J,K+1))/DXM(I+1)
          -(UN(I+1,J,K)-UN(I,J,K))/DXM(I+1))/DZ(K))

***********
*** 1.0 DERIVATIVES OF U
***
***********

***********
*** 2.0 DERIVATIVES OF V
***
***********
\[ \text{DVDX} = (A \ast \text{VN}(I-1,J,K) + B \ast \text{VN}(I,J,K) + C \ast \text{VN}(I+1,J,K))/DX(I+\text{INT}(A)) \]
\[ \text{DVDY} = (A \ast \text{VN}(I,J-1,K) + B \ast \text{VN}(I,J,K) + C \ast \text{VN}(I,J+1,K))/DYM(J+\text{INT}(C)) \]
\[ \text{DVDZ} = (A \ast \text{VN}(I,J,K-1) + B \ast \text{VN}(I,J,K) + C \ast \text{VN}(I,J,K+1))/DZM(K+\text{INT}(A)) \]

\[ \text{DVDX}^2 = ((\text{VN}(I+1,J,K) - \text{VN}(I,J,K))/DX(I) - (\text{VN}(I,J,K) - \text{VN}(I-1,J,K))/DX(I-1))/DXM(I) \]
\[ \text{DWDY}^2 = ((\text{VN}(I,J+1,K) - \text{VN}(I,J,K))/DYM(J) - (\text{VN}(I,J,K) - \text{VN}(I,J-1,K))/DYM(J-1))/DYM(J) \]
\[ \text{DWDZ}^2 = ((\text{VN}(I,J,K+1) - \text{VN}(I,J,K))/DZM(K+1) - (\text{VN}(I,J,K) - \text{VN}(I,J,K-1))/DZM(K))/DZM(K) \]

**3.0 DERIVATIVES OF W**

\[ \text{DWDX} = (A \ast \text{WN}(I-1,J,K) + B \ast \text{WN}(I,J,K) + C \ast \text{WN}(I+1,J,K))/DX(I+\text{INT}(A)) \]
\[ \text{DWDY} = (A \ast \text{WN}(I,J-1,K) + B \ast \text{WN}(I,J,K) + C \ast \text{WN}(I,J+1,K))/DYM(J+\text{INT}(A)) \]
\[ \text{DWDZ} = (A \ast \text{WN}(I,J,K-1) + B \ast \text{WN}(I,J,K) + C \ast \text{WN}(I,J,K+1))/DZM(K+\text{INT}(C)) \]

\[ \text{DWDX}^2 = ((\text{WN}(I+1,J,K) - \text{WN}(I,J,K))/DX(I) - (\text{WN}(I,J,K) - \text{WN}(I-1,J,K))/DX(I-1))/DXM(I) \]
\[ \text{DWDY}^2 = ((\text{WN}(I,J+1,K) - \text{WN}(I,J,K))/DYM(J) - (\text{WN}(I,J,K) - \text{WN}(I,J-1,K))/DYM(J-1))/DYM(J) \]
\[ \text{DWDZ}^2 = ((\text{WN}(I,J,K+1) - \text{WN}(I,J,K))/DZM(K+1) - (\text{WN}(I,J,K) - \text{WN}(I,J,K-1))/DZM(K))/DZM(K) \]

\[ \text{DWD}^2_{DZDX} = (A \ast ((\text{WN}(I-1,J,K) - \text{WN}(I,J,K))/DX(I-1) + (\text{WN}(I,J,K) - \text{WN}(I,J,K))/DX(I-1))/DZM(K)) \]
\[ \text{DWD}^2_{DWDY} = (A \ast ((\text{WN}(I,J-1,K) - \text{WN}(I,J,K))/DYM(J-1) + (\text{WN}(I,J,K) - \text{WN}(I,J,K))/DYM(J))/DYM(J)) \]
\[ \text{DWD}^2_{DWDZ} = (A \ast ((\text{WN}(I,J,K+1) - \text{WN}(I,J,K))/DZM(K+1) + (\text{WN}(I,J,K) - \text{WN}(I,J,K))/DZM(K))/DZM(K)) \]
DW2DZDY = (A* ((WN(I, J-1, K-1) - WN(I, J, K-1)) / DYM(J))
c + (WN(I, J-1, K) - WN(I, J, K)) / DYM(J)) / DZ(K-1))
c + (C* ((WN(I, J+1, K+1) - WN(I, J, K+1)) / DYM(J+1))
c - (WN(I, J+1, K) - WN(I, J, K)) / DYM(J+1)) / DZ(K))

**********************************************************************
************
*** 4.0 DERIVATIVES OF K AND TURBULENT VISCOSITY
***
************
************
DKDX = (A*TKN(I-1, J, K) + B*TKN(I, J, K) + C*TKN(I+1, J, K)) / DX(I+INT(A))
DKDY = (A*TKN(I, J-1, K) + B*TKN(I, J, K) + C*TKN(I, J+1, K)) / DY(J+INT(A))
DKDZ = (A*TKN(I, J, K-1) + B*TKN(I, J, K) + C*TKN(I, J, K+1)) / DZ(K+INT(A))

DNUTDX = (A*TVIS(I-1, J, K) + B*TVIS(I, J, K) + C*TVIS(I+1, J, K)) / DX(I+INT(A))
DNUTDY = (A*TVIS(I, J-1, K) + B*TVIS(I, J, K) + C*TVIS(I, J+1, K)) / DY(J+INT(A))
DNUTDZ = (A*TVIS(I, J, K-1) + B*TVIS(I, J, K) + C*TVIS(I, J, K+1)) / DZ(K+INT(A))

********************************************************************************
************
*** 5.0 CALCULATION OF PI & GAMMA FUNCTIONS
***
************
************
PI(I, J, K) = 2.0*(DUDX**2.0) + DUDY*(DUDY+DVDX) + DUDZ*(DUDZ+DWDX)
c + DVDX*(DVDX+DUDY) + 2.0*(DVDY**2.0) + DVDZ*(DVDZ+DWDY)
c + DWDX*(DWDX+DUDZ) + DWDY*(DWDY+DWDZ) + 2.0*(DWDZ**2.0)
GAMX(I, J, K) = (-2.0*DKDX/3.0)
c + TVIS(I, J, K) * ((2.0*DUDX2)+DUDY2+DUDZ2+DV2DYDX+DW2DZDX)
c + (2.0*DNUTDX*DVDX) + DNUTDY*(DVDY+DWDY) + DNUTDZ*(DVDZ+DWDY)
GAMY(I, J, K) = (-2.0*DKDY/3.0)
c + TVIS(I, J, K) * (DV2DX+DUDY2)+DVDZ2+DU2DXDY+DW2DZDY)
c + (2.0*DNUTDY*DVDY) + DNUTDX*(DVDX+DUDY) + DNUTDZ*(DVDZ+DWDY)
GAM2(I,J,K)=(-2.0*DKDZ/3.0)
c + TVIS(I,J,K)*(DWDX2+DWDY2+(2.0*DWDZ2)+DU2DXDZ+DV2DYDZ)
c + (2.0*DNUTDZ*DWDZ) + DNUTDX*(DWDX+DUDZ) + DNUTDY*(DWDY+DVDZ)
100 CONTINUE
RETURN
END

**********************************************************
************
** SUBROUTINE: TURBVISC
**
**
** DEVELOPER: JIM WEATHERS
**
**
** PURPOSE: CALCULATES THE TURBULENT VISCOSITY
**
**********************************************************

SUBROUTINE TURBVISC(VIS)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50)
1/TURB/TK(50,50,50),TKN(50,50,50),TE(50,50,50),TEN(50,50,50),
2 TVIS(50,50,50)
1/TCOEFF/TCFCMU,TCFC1,TCFC2,TSIGK,TSIGE
1/TURBS/REYNT(50,50,50),TFNMU(50,50,50),TFN1(50,50,50),
2 TFN2(50,50,50)
DO 100 I=2,IMM1
DO 100 J=2,JMM1
DO 100 K=2,KMM1
   TVISC=TFNMU(I,J,K)*TCFCMU*(TKN(I,J,K)**2.0)/TEN(I,J,K)
   TVIS(I,J,K)=VIS+TVISC
100 CONTINUE
DO 200 J=1,JMAX
   DO 200 I=1,IMAX
   TVIS(I,J,1)=TVIS(I,J,2)
   TVIS(I,J,KMAX)=TVIS(I,J,KMM1)
200 CONTINUE
DO 210 I=1,IMAX
   DO 210 K=1,KMAX
   TVIS(I,1,K)=TVIS(I,2,K)
   TVIS(I,JMAX,K)=TVIS(I,JMM1,K)
210 CONTINUE
DO 220 J=1,JMAX
   DO 220 K=1,KMAX
   TVIS(1,J,K)=TVIS(2,J,K)
   TVIS(IMAX,J,K)=TVIS(IMM1,J,K)
220 CONTINUE
RETURN
END

********************************************************************
************
** FUNCTION: CALTHETA
**
**
** PURPOSE: CALCULATES THE VALUE OF THE FUNCTION 'THETA'
**
********************************************************************

REAL*8 FUNCTION CALTHETA(IT,RR,SIGMAR,I,J,K)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON
1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
3 DY(50),DZ(50),DYM(50),DZM(50)
1/TURB/TK(50,50,50), TKN(50,50,50), TE(50,50,50), TEN(50,50,50),
        2               TVIS(50,50,50)

DIMENSION RR(50,50,50)
C=REAL(IT-2*INT(IT/2.0))
A=C-1.0
B=(-1.0)**C
DRDX=(A*RR(I-1,J,K)+B*RR(I,J,K)+C*RR(I+1,J,K))/DX(I+INT(A))
DRDY=(A*RR(I,J-1,K)+B*RR(I,J,K)+C*RR(I,J+1,K))/DY(J+INT(A))
DRDZ=(A*RR(I,J,K-1)+B*RR(I,J,K)+C*RR(I,J,K+1))/DZ(K+INT(A))

DRDX2=((RR(I+1,J,K)-RR(I,J,K))/DX(I+INT(A))
1 -(RR(I,J,K)-RR(I-1,J,K))/DX(I-1))/DXM(I)
DRDY2=((RR(I,J+1,K)-RR(I,J,K))/DY(J+INT(A))
1 -(RR(I,J,K)-RR(I,J-1,K))/DY(J-1))/DYM(J)
DRDZ2=((RR(I,J,K+1)-RR(I,J,K))/DZ(K+INT(A))
1 -(RR(I,J,K)-RR(I,J,K-1))/DZ(K-1))/DZM(K)

DNUTDX=(A*TVIS(I-1,J,K)+B*TVIS(I,J,K)+C*TVIS(I+1,J,K))/DX(I+INT(A))
DNUTDY=(A*TVIS(I,J-1,K)+B*TVIS(I,J,K)+C*TVIS(I,J+1,K))/DY(J+INT(A))
DNUTDZ=(A*TVIS(I,J,K-1)+B*TVIS(I,J,K)+C*TVIS(I,J,K+1))/DZ(K+INT(A))

CALTHETA=(TVIS(I,J,K)/SIGMAR)*(DRDX2+DRDY2+DRDZ2)
1 +1.0/SIGMAR)*(DRDX*DNUTDX+DRDY*DNUTDY+DRDZ*DNUTDZ)
RETURN
END

******************************************************
** SUBROUTINE: CALTFUNC
**
** PURPOSE: CALCULATES THE VALUES OF THE EMPIRICAL
FUNCTIONS **

193
SUBROUTINE CALTFUNC(NFLOW,VIS)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON 
 1/GEOM/IMAX,IMM1,JMAX,JMM1,KMAX,KMM1,DX(50),DXM(50),
 2 X(50),Y(50),Z(50),XM(50),YM(50),ZM(50),
 3 DY(50),DZ(50),DYM(50),DZM(50)
1/TURB/TK(50,50,50),TKN(50,50,50),TE(50,50,50),TEN(50,50,50),
 2 TVIS(50,50,50)
1/TURBS/REYNT(50,50,50),TFNMU(50,50,50),TFN1(50,50,50),
 2 TFN2(50,50,50)
DATA AMU,AT /0.0165,20.5/
DO 100 I=2,IMM1
  XDIST=(X(I)-XM(1))
    IF((XM(IMM1)-X(I)) .LT. XDIST) XDIST=XM(IMM1)-X(I)
    DO 100 J=2,JMM1
      YDIST=(Y(J)-YM(1))
        IF((YM(JMM1)-Y(J)) .LT. YDIST) YDIST=YM(JMM1)-Y(J)
        DO 100 K=2,KMM1
          ZDIST=(Z(K)-ZM(1))
            IF((ZM(KMM1)-Z(K)) .LT. ZDIST) ZDIST=ZM(KMM1)-Z(K)
            IF (NFLOW.EQ.2) THEN
              TFNMU(I,J,K)=1.0
              TFN1(I,J,K)=1.0
              TFN2(I,J,K)=1.0
            ELSE
              REYNT(I,J,K)=(TKN(I,J,K)**2.0)/(VIS*TEN(I,J,K))
              YNORM=DMIN1(XDIST,YDIST,ZDIST)
              REYN=DSQRT(TKN(I,J,K))*YNORM/VIS
              TFNMU(I,J,K)=((1.0-DEXP(-AMU*REYNY)**2.0) +
                (1.0+AT/REYNT(I,J,K))
              TFN1(I,J,K)=1.0+(0.05/TFNMU(I,J,K))**3.0
              TFN2(I,J,K)=1.0-DEXP(-1.0*(REYNT(I,J,K)**2.0))
            ENDIF
 100 CONTINUE
RETURN
END
6.0 Input Files for FORTRAN Source Code

This section gives example grid.dat and room.dat input files. File inputs and corresponding variable names are separated by colons. The nomenclature section, section 2.0, lists variable names, descriptions and corresponding symbols.

6.1 Input File: grid.dat

4.6000 2.7500 2.7500 :XDIMEN, YDIMEN, ZDIMEN
42 26 26 :IMAX, JMAX, KMAX
0.00000 0.05750 0.11500 0.05750: X(1), XM(1), DX(1), DXM(1)
0.11500 0.17250 0.11500 0.11500: X(2), XM(2), DX(2), DXM(2)
0.23000 0.28749 0.11500 0.11500: X(3), XM(3), DX(3), DXM(3)
0.34499 0.40249 0.11500 0.11500: X(4), XM(4), DX(4), DXM(4)
0.45999 0.51749 0.11500 0.11500: X(5), XM(5), DX(5), DXM(5)
0.57499 0.63249 0.11500 0.11500: X(6), XM(6), DX(6), DXM(6)
0.68999 0.74749 0.11500 0.11500: X(7), XM(7), DX(7), DXM(7)
0.80499 0.86249 0.11500 0.11500: X(8), XM(8), DX(8), DXM(8)
0.91999 0.97749 0.11500 0.11500: X(9), XM(9), DX(9), DXM(9)
1.03499 1.09249 0.11500 0.11500: X(10), XM(10), DX(10), DXM(10)
1.14999 1.20749 0.11500 0.11500: X(11), XM(11), DX(11), DXM(11)
<table>
<thead>
<tr>
<th>X</th>
<th>XM</th>
<th>DX</th>
<th>DXM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.26499</td>
<td>1.32249</td>
<td>0.11500</td>
<td>0.11500: X(12), XM(12), DX(12), DXM(12)</td>
</tr>
<tr>
<td>1.37999</td>
<td>1.43749</td>
<td>0.11500</td>
<td>0.11500: X(13), XM(13), DX(13), DXM(13)</td>
</tr>
<tr>
<td>1.49499</td>
<td>1.55249</td>
<td>0.11500</td>
<td>0.11500: X(14), XM(14), DX(14), DXM(14)</td>
</tr>
<tr>
<td>1.60999</td>
<td>1.66749</td>
<td>0.11500</td>
<td>0.11500: X(15), XM(15), DX(15), DXM(15)</td>
</tr>
<tr>
<td>1.72499</td>
<td>1.78249</td>
<td>0.11500</td>
<td>0.11500: X(16), XM(16), DX(16), DXM(16)</td>
</tr>
<tr>
<td>1.83999</td>
<td>1.89749</td>
<td>0.11500</td>
<td>0.11500: X(17), XM(17), DX(17), DXM(17)</td>
</tr>
<tr>
<td>1.95499</td>
<td>2.01249</td>
<td>0.11500</td>
<td>0.11500: X(18), XM(18), DX(18), DXM(18)</td>
</tr>
<tr>
<td>2.06999</td>
<td>2.12749</td>
<td>0.11500</td>
<td>0.11500: X(19), XM(19), DX(19), DXM(19)</td>
</tr>
<tr>
<td>2.18499</td>
<td>2.24249</td>
<td>0.11500</td>
<td>0.11500: X(20), XM(20), DX(20), DXM(20)</td>
</tr>
<tr>
<td>2.29999</td>
<td>2.35749</td>
<td>0.11500</td>
<td>0.11500: X(21), XM(21), DX(21), DXM(21)</td>
</tr>
<tr>
<td>2.41499</td>
<td>2.47249</td>
<td>0.11500</td>
<td>0.11500: X(22), XM(22), DX(22), DXM(22)</td>
</tr>
<tr>
<td>2.52999</td>
<td>2.58749</td>
<td>0.11500</td>
<td>0.11500: X(23), XM(23), DX(23), DXM(23)</td>
</tr>
<tr>
<td>2.64499</td>
<td>2.70249</td>
<td>0.11500</td>
<td>0.11500: X(24), XM(24), DX(24), DXM(24)</td>
</tr>
<tr>
<td>2.75999</td>
<td>2.81749</td>
<td>0.11500</td>
<td>0.11500: X(25), XM(25), DX(25), DXM(25)</td>
</tr>
<tr>
<td>2.87499</td>
<td>2.93249</td>
<td>0.11500</td>
<td>0.11500: X(26), XM(26), DX(26), DXM(26)</td>
</tr>
<tr>
<td>2.98999</td>
<td>3.04749</td>
<td>0.11500</td>
<td>0.11500: X(27), XM(27), DX(27), DXM(27)</td>
</tr>
<tr>
<td>3.10499</td>
<td>3.16249</td>
<td>0.11500</td>
<td>0.11500: X(28), XM(28), DX(28), DXM(28)</td>
</tr>
<tr>
<td>3.21999</td>
<td>3.27749</td>
<td>0.11500</td>
<td>0.11500: X(29), XM(29), DX(29), DXM(29)</td>
</tr>
<tr>
<td>3.33499</td>
<td>3.39249</td>
<td>0.11500</td>
<td>0.11500: X(30), XM(30), DX(30), DXM(30)</td>
</tr>
<tr>
<td>3.44999</td>
<td>3.50749</td>
<td>0.11500</td>
<td>0.11500: X(31), XM(31), DX(31), DXM(31)</td>
</tr>
<tr>
<td>3.56499</td>
<td>3.62249</td>
<td>0.11500</td>
<td>0.11500: X(32), XM(32), DX(32), DXM(32)</td>
</tr>
</tbody>
</table>
3.67999 3.73749 0.11500 0.11500: X(33), XM(33), DX(33), DXM(33)
3.79499 3.85249 0.11500 0.11500: X(34), XM(34), DX(34), DXM(34)
3.90999 3.96749 0.11500 0.11500: X(35), XM(35), DX(35), DXM(35)
4.02499 4.08249 0.11500 0.11500: X(36), XM(36), DX(36), DXM(36)
4.13999 4.19749 0.11500 0.11500: X(37), XM(37), DX(37), DXM(37)
4.25499 4.31249 0.11500 0.11500: X(38), XM(38), DX(38), DXM(38)
4.36999 4.42749 0.11500 0.11500: X(39), XM(39), DX(39), DXM(39)
4.48499 4.54249 0.11500 0.11500: X(40), XM(40), DX(40), DXM(40)
4.59999 4.65749 0.11500 0.11500: X(41), XM(41), DX(41), DXM(41)
4.71499 4.77249 0.11500 0.11500: X(42), XM(42), DX(42), DXM(42)
0.00000 0.05729 0.11458 0.11458: Y(1), YM(1), DY(1), DYM(1)
0.11458 0.17187 0.11458 0.11458: Y(2), YM(2), DY(2), DYM(2)
0.22916 0.28645 0.11458 0.11458: Y(3), YM(3), DY(3), DYM(3)
0.34375 0.40104 0.11458 0.11458: Y(4), YM(4), DY(4), DYM(4)
0.45833 0.51562 0.11458 0.11458: Y(5), YM(5), DY(5), DYM(5)
0.57291 0.63020 0.11458 0.11458: Y(6), YM(6), DY(6), DYM(6)
0.68750 0.74479 0.11458 0.11458: Y(7), YM(7), DY(7), DYM(7)
0.80208 0.85937 0.11458 0.11458: Y(8), YM(8), DY(8), DYM(8)
0.91666 0.97395 0.11458 0.11458: Y(9), YM(9), DY(9), DYM(9)
1.03125 1.08854 0.11458 0.11458: Y(10), YM(10), DY(10), DYM(10)
1.14583 1.20312 0.11458 0.11458: Y(11), YM(11), DY(11), DYM(11)
<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.26041</td>
<td>1.31770</td>
<td>0.11458</td>
<td>0.11458: Y(12), YM(12),</td>
</tr>
<tr>
<td>DY(12), DYM(12)</td>
<td>1.37500</td>
<td>1.43229</td>
<td>0.11458: Y(13), YM(13),</td>
</tr>
<tr>
<td>DY(13), DYM(13)</td>
<td>1.48958</td>
<td>1.54687</td>
<td>0.11458: Y(14), YM(14),</td>
</tr>
<tr>
<td>DY(14), DYM(14)</td>
<td>1.60416</td>
<td>1.66145</td>
<td>0.11458: Y(15), YM(15),</td>
</tr>
<tr>
<td>DY(15), DYM(15)</td>
<td>1.71875</td>
<td>1.77604</td>
<td>0.11458: Y(16), YM(16),</td>
</tr>
<tr>
<td>DY(16), DYM(16)</td>
<td>1.83333</td>
<td>1.89062</td>
<td>0.11458: Y(17), YM(17),</td>
</tr>
<tr>
<td>DY(17), DYM(17)</td>
<td>1.94791</td>
<td>2.00520</td>
<td>0.11458: Y(18), YM(18),</td>
</tr>
<tr>
<td>DY(18), DYM(18)</td>
<td>2.06250</td>
<td>2.11979</td>
<td>0.11458: Y(19), YM(19),</td>
</tr>
<tr>
<td>DY(19), DYM(19)</td>
<td>2.17708</td>
<td>2.23437</td>
<td>0.11458: Y(20), YM(20),</td>
</tr>
<tr>
<td>DY(20), DYM(20)</td>
<td>2.29166</td>
<td>2.34895</td>
<td>0.11458: Y(21), YM(21),</td>
</tr>
<tr>
<td>DY(21), DYM(21)</td>
<td>2.40625</td>
<td>2.46354</td>
<td>0.11458: Y(22), YM(22),</td>
</tr>
<tr>
<td>DY(22), DYM(22)</td>
<td>2.52083</td>
<td>2.57812</td>
<td>0.11458: Y(23), YM(23),</td>
</tr>
<tr>
<td>DY(23), DYM(23)</td>
<td>2.63541</td>
<td>2.69270</td>
<td>0.11458: Y(24), YM(24),</td>
</tr>
<tr>
<td>DY(24), DYM(24)</td>
<td>2.75000</td>
<td>2.80729</td>
<td>0.11458: Y(25), YM(25),</td>
</tr>
<tr>
<td>DY(25), DYM(25)</td>
<td>2.86458</td>
<td>2.92187</td>
<td>0.11458: Y(26), YM(26),</td>
</tr>
<tr>
<td>DY(26), DYM(26)</td>
<td>0.00000</td>
<td>0.05729</td>
<td>0.05729: Z(1), ZM(1),</td>
</tr>
<tr>
<td>DZ(1), DZM(1)</td>
<td>0.11458</td>
<td>0.17187</td>
<td>0.11458: Z(2), ZM(2),</td>
</tr>
<tr>
<td>DZ(2), DZM(2)</td>
<td>0.22916</td>
<td>0.28645</td>
<td>0.11458: Z(3), ZM(3),</td>
</tr>
<tr>
<td>DZ(3), DZM(3)</td>
<td>0.34375</td>
<td>0.40104</td>
<td>0.11458: Z(4), ZM(4),</td>
</tr>
<tr>
<td>DZ(4), DZM(4)</td>
<td>0.45833</td>
<td>0.51562</td>
<td>0.11458: Z(5), ZM(5),</td>
</tr>
<tr>
<td>DZ(5), DZM(5)</td>
<td>0.57291</td>
<td>0.63020</td>
<td>0.11458: Z(6), ZM(6),</td>
</tr>
<tr>
<td>DZ(6), DZM(6)</td>
<td>0.63020</td>
<td>0.68749</td>
<td>0.11458: Z(7), ZM(7),</td>
</tr>
</tbody>
</table>
0.68750 0.74479 0.11458 0.11458: Z(7), ZM(7),
DZ(7), DZM(7)
0.80208 0.85937 0.11458 0.11458: Z(8), ZM(8),
DZ(8), DZM(8)
0.91666 0.97395 0.11458 0.11458: Z(9), ZM(9),
DZ(9), DZM(9)
1.03125 1.08854 0.11458 0.11458: Z(10), ZM(10),
DZ(10), DZM(10)
1.14583 1.20312 0.11458 0.11458: Z(11), ZM(11),
DZ(11), DZM(11)
1.26041 1.31770 0.11458 0.11458: Z(12), ZM(12),
DZ(12), DZM(12)
1.37500 1.43229 0.11458 0.11458: Z(13), ZM(13),
DZ(13), DZM(13)
1.48958 1.54687 0.11458 0.11458: Z(14), ZM(14),
DZ(14), DZM(14)
1.60416 1.66145 0.11458 0.11458: Z(15), ZM(15),
DZ(15), DZM(15)
1.71875 1.77604 0.11458 0.11458: Z(16), ZM(16),
DZ(16), DZM(16)
1.83333 1.89062 0.11458 0.11458: Z(17), ZM(17),
DZ(17), DZM(17)
1.94791 2.00520 0.11458 0.11458: Z(18), ZM(18),
DZ(18), DZM(18)
2.06250 2.11979 0.11458 0.11458: Z(19), ZM(19),
DZ(19), DZM(19)
2.17708 2.23437 0.11458 0.11458: Z(20), ZM(20),
DZ(20), DZM(20)
2.29166 2.34895 0.11458 0.11458: Z(21), ZM(21),
DZ(21), DZM(21)
2.40625 2.46354 0.11458 0.11458: Z(22), ZM(22),
DZ(22), DZM(22)
2.52083 2.57812 0.11458 0.11458: Z(23), ZM(23),
DZ(23), DZM(23)
2.63541 2.69270 0.11458 0.11458: Z(24), ZM(24),
DZ(24), DZM(24)
2.75000 2.80729 0.11458 0.11458: Z(25), ZM(25),
DZ(25), DZM(25)
2.86458 2.92187 0.11458 0.11458: Z(26), ZM(26), DZ(26),
DZM(26)
6.2 Input File: room.dat

15 ACH: :TITLE
3  2 :NFLOW, NTEMP
*************************************:COMMENTS
32 :NINCELS
10  2  0.69  0.10  2.30 :JIN(1), KIN(1), UINS(1), TKIN(1), TEIN(1)
10  3  0.69  0.10  2.30 :JIN(2), KIN(2), UINS(2), TKIN(2), TEIN(2)
10  4  0.69  0.10  2.30 :JIN(3), KIN(3), UINS(3), TKIN(3), TEIN(3)
10  5  0.69  0.10  2.30 :JIN(4), KIN(4), UINS(4), TKIN(4), TEIN(4)
11  2  0.69  0.10  2.30 :JIN(5), KIN(5), UINS(5), TKIN(5), TEIN(5)
11  3  0.69  0.10  2.30 :JIN(6), KIN(6), UINS(6), TKIN(6), TEIN(6)
11  4  0.69  0.10  2.30 :JIN(7), KIN(7), UINS(7), TKIN(7), TEIN(7)
11  5  0.69  0.10  2.30 :JIN(8), KIN(8), UINS(8), TKIN(8), TEIN(8)
12  2  0.69  0.10  2.30 :JIN(9), KIN(9), UINS(9), TKIN(9), TEIN(9)
12  3  0.69  0.10  2.30 :JIN(10), KIN(10), UINS(10), TKIN(10), TEIN(10)
12  4  0.69  0.10  2.30 :JIN(11), KIN(11), UINS(11), TKIN(11), TEIN(11)
12  5  0.69  0.10  2.30 :JIN(12), KIN(12), UINS(12), TKIN(12), TEIN(12)
13  2  0.69  0.10  2.30 :JIN(13), KIN(13), UINS(13), TKIN(13), TEIN(13)
13  3  0.69  0.10  2.30 :JIN(14), KIN(14), UINS(14), TKIN(14), TEIN(14)
13  4  0.69  0.10  2.30 :JIN(15), KIN(15), UINS(15), TKIN(15), TEIN(15)
13  5  0.69  0.10  2.30 :JIN(16), KIN(16), UINS(16), TKIN(16), TEIN(16)
14 2 0.00 0.00 0.00
:JIN(17), KIN(17), UINS(17), TKIN(17), TEIN(17)
14 3 0.00 0.00 0.00
:JIN(18), KIN(18), UINS(18), TKIN(18), TEIN(18)
14 4 0.00 0.00 0.00
:JIN(19), KIN(19), UINS(19), TKIN(19), TEIN(19)
14 5 0.00 0.00 0.00
:JIN(20), KIN(20), UINS(20), TKIN(20), TEIN(20)
15 2 0.00 0.00 0.00
:JIN(21), KIN(21), UINS(21), TKIN(21), TEIN(21)
15 3 0.00 0.00 0.00
:JIN(22), KIN(22), UINS(22), TKIN(22), TEIN(22)
15 4 0.00 0.00 0.00
:JIN(23), KIN(23), UINS(23), TKIN(23), TEIN(23)
15 5 0.00 0.00 0.00
:JIN(24), KIN(24), UINS(24), TKIN(24), TEIN(24)
16 2 0.00 0.00 0.00
:JIN(25), KIN(25), UINS(25), TKIN(25), TEIN(25)
16 3 0.00 0.00 0.00
:JIN(26), KIN(26), UINS(26), TKIN(26), TEIN(26)
16 4 0.00 0.00 0.00
:JIN(27), KIN(27), UINS(27), TKIN(27), TEIN(27)
16 5 0.00 0.00 0.00
:JIN(28), KIN(28), UINS(28), TKIN(28), TEIN(28)
17 2 0.00 0.00 0.00
:JIN(29), KIN(29), UINS(29), TKIN(29), TEIN(29)
17 3 0.00 0.00 0.00
:JIN(30), KIN(30), UINS(30), TKIN(30), TEIN(30)
17 4 0.00 0.00 0.00
:JIN(31), KIN(31), UINS(31), TKIN(31), TEIN(31)
17 5 0.00 0.00 0.00
:JIN(32), KIN(32), UINS(32), TKIN(32), TEIN(32)
8 4 :
:JOUTTEMP (1)
3 :
:JOUTTEMP (2)
4 :
:JOUTTEMP (3)
5 :
:JOUTTEMP (4)
6 :
:JOUTTEMP (5)
7 :
:JOUTTEMP (6)
8 :
:JOUTTEMP (7)
9 :
:JOUTTEMP (8)
17 :
:KOUTTEMP (1)
18 :KOUTTEMP(2)
19 :KOUTTEMP(3)
20 :KOUTTEMP(4)

******************************************************************************
:COMMENTS
5 :JTREF1
6 :JTREF2
18 :KTREF1
19 :KTREF2

******************************************************************************
:COMMENTS
1.2  1.5e-5 :QDENS, QVISC
0.0261 1005. :QTCOND, QCP
2.447 3.440 1.225 :FILMW, FILMF, FILMC
28.5 21.0 30.0 25.7 :QT0, QTIN, QTWALL, TOUT
0.0 -9.80 0.0 3.47e-3 :GX, GY, GZ, BUOBETA

******************************************************************************
:COMMENTS
1.0e-7 1500 :QZETA, MAXSTPQ
50000 999.0 0.010 :ISTMAX, TIMEMAX, USERDT

******************************************************************************
:COMMENTS
1 1 1 1 1 1 :NSI1, NSIMAX, NSJ1, NSJMAX, NSK1, NSKMAX

******************************************************************************
:COMMENTS
0.09 1.44 1.92 1.0 1.3 :TCFCMU, TCFC1, TCFC2, TSIGK, TSIGE

******************************************************************************
:COMMENTS
examp2.out :FNAMEOUT
examp2.vec :FNAMEVEC
filmcoef.dat :FILMDAT
/w/ramey/g/v15.dat :EXPFILE
/w/ramey/g/t15.dat :TEXPFILE
0 :NUMLATS
0 :NUMLONGS

******************************************************************************
:COMMENTS
0 0 0 :NOPT(1,1), NOPT(1,2), NOPT(1,3)
0 0 0 :NOPT(2,1), NOPT(2,2), NOPT(2,3)
0 0 0 :NOPT(3,1), NOPT(3,2), NOPT(3,3)
0 0 1 :NOPT(4,1), NOPT(4,2), NOPT(4,3)
0 0 0 :NOPT(5,1), NOPT(5,2), NOPT(5,3)
0 0 1 :NOPT(6,1), NOPT(6,2), NOPT(6,3)
0 0 0 :NOPT(7,1), NOPT(7,2), NOPT(7,3)
0 0 0 :NOPT(8,1), NOPT(8,2), NOPT(8,3)
vmag3d.dat :HDF3D(1)
temp3d.dat :HDF3D(2)
************:COMMENTS
20 :NUMPLTS
500  vm05.dat  tp05.dat :INTRVLS(1), VTRANS(1),
     TTRANS(1)
1000 vm10.dat  tp10.dat :INTRVLS(2), VTRANS(2),
     TTRANS(2)
1500 vm15.dat  tp15.dat :INTRVLS(3), VTRANS(3),
     TTRANS(3)
2000 vm20.dat  tp20.dat :INTRVLS(4), VTRANS(4),
     TTRANS(4)
3000 vm30.dat  tp30.dat :INTRVLS(5), VTRANS(5),
     TTRANS(5)
4000 vm40.dat  tp40.dat :INTRVLS(6), VTRANS(6),
     TTRANS(6)
5000 vm50.dat  tp50.dat :INTRVLS(7), VTRANS(7),
     TTRANS(7)
6000 vm60.dat  tp60.dat :INTRVLS(8), VTRANS(8),
     TTRANS(8)
7000 vm70.dat  tp70.dat :INTRVLS(9), VTRANS(9),
     TTRANS(9)
8000 vm80.dat  tp80.dat :INTRVLS(10), VTRANS(10),
     TTRANS(10)
9000 vm90.dat  tp90.dat :INTRVLS(11), VTRANS(11),
     TTRANS(11)
10000 vm100.dat tp100.dat :INTRVLS(12), VTRANS(12),
    TTRANS(12)
15000 vm150.dat tp150.dat :INTRVLS(13), VTRANS(13),
    TTRANS(13)
20000 vm200.dat tp200.dat :INTRVLS(14), VTRANS(14),
    TTRANS(14)
25000 vm250.dat tp250.dat :INTRVLS(15), VTRANS(15),
    TTRANS(15)
30000 vm300.dat tp300.dat :INTRVLS(16), VTRANS(16),
    TTRANS(16)
35000 vm350.dat tp350.dat :INTRVLS(17), VTRANS(17),
    TTRANS(17)
40000 vm400.dat tp400.dat :INTRVLS(18), VTRANS(18),
    TTRANS(18)
45000 vm450.dat tp450.dat :INTRVLS(19), VTRANS(19),
    TTRANS(19)
50000 vm500.dat tp500.dat :INTRVLS(20), VTRANS(20),
    TTRANS(20)
7.0 Description of Useful Modifications

This section attempts to demonstrate how to change the program so that it can be used to model different configurations.

7.1 How To Change the Model Type

The easiest change to make with the TFLOVG.F code is the model type.

1. Locate the NFLOW variable in the input file, 'room.dat'.
   (First number on the 2nd line of 'room.dat')

2. Change NFLOW to the desired number.
   If you set NFLOW = 1, then you are using a laminar model.
   If you set NFLOW = 2, then you are using a turbulent k-ε model with wall functions.
   If you set NFLOW = 3, then you are using a turbulent low-Reynolds number k-ε model.

7.2 How To Change the Air Flow Rate

1. Locate the UINS() array in the input file, 'room.dat'.
2. Edit the values in the input file to match the desired inlet flow rate for each inlet cell.

7.3 How To Change the Location & Size of the Inlet

1. Locate NINCELS on the 4th line of 'room.dat'.
   Locate JIN() & KIN(), (Starting on the fifth line of 'room.dat'). JIN() & KIN() are the integers in the first two columns of the section of 'room.dat' which follows NINCELS.

2. Determine the cell numbers at the location where you want the inlet, and edit the values appropriately.

Example: You want the inlet to be in the lowest, nearest corner of the room.
   You also want the inlet to be 3 cells tall (in the y-direction) and 5 cells wide (in the z-direction).
   Then you would modify NINCELS to be 15, (3 x 5).
   Also edit JIN() & KIN() to be the cell numbers you desire.
   The fourth line and the first two columns of the next 15 lines of the 'room.dat' file would then look like this:
Note that the cell numbers start at 2 because cell number 1 in all directions is outside the boundary.

7.4 How To Change the Location & Size of the Outlet

Changing the size and location of the outlet is similar to changing the size and location of the inlet.

1. Locate the NOUTJ & NOUTK variables in 'room.dat'.
   Locate the JOUTTEMP() & KOUTTEMP() arrays in 'room.dat'.
   (This section of the input file is located after the inlet specification variables.)
2. Note that NOUTJ & NOUTK control the number of outlet cells in the y and z directions, respectively. JOUTTEMP() & KOUTTEMP() store the outlet cell numbers in the y and z directions.

3. Set NOUTJ (in 'room.dat') to the desired number of outlet cells in the y-direction.

4. Set NOUTK (in 'room.dat') to the desired number of outlet cells in the z-direction.

5. Set JOUTTEMP() (in 'room.dat') to the desired cell numbers in the y-direction.

6. Set KOUTTEMP() (in 'room.dat') to the desired cell numbers in the z-direction.

Example: You want the outlet to be in the lowest, nearest corner of the room. You also want the outlet to be 5 cells tall (in the y-direction) and 4 cells wide (in the z-direction).

Then you would modify NOUTJ to be 5, and NOUTK to be 4. Also edit JOUTTEMP() & KOUTTEMP() to the cell numbers you desire.

The outlet specification section of the 'room.dat' file would then look like this:
7.5 How To Model Buoyancy

1. Locate the GX, GY, GZ variables in the input file, 'room.dat'.

2. Modify the gravitational acceleration amounts.
   For Example:  GX = 0.0
                 GY = 0.0
                 GZ = 0.0  (No Buoyancy)

   Setting GY = -9.8 (m/sec²) models buoyancy in the normal fashion, with the Y-direction being the height of the domain.
8.0  How To Run the Code

First, you should have the following files:

  tflovg.f
  tfexe
  room.dat
  grid.dat

1. If the files are not already on the rs6000, then copy them to the rs6000 using ftp or
   the doswrite command. To use the doswrite command, first insert a floppy disk with
   the files on it into an rs6000 computer that you are logged onto. Then type the
   following in one of the windows with a return at the end of each line:

     doswrite  tflovg.f  tflovg.f
     doswrite  tfexe  tfexe
     doswrite  room.dat  room.dat
     doswrite  grid.dat  grid.dat

   This will copy the files to the rs6000 directory that you are in when you issue the
   command.

2. Compile the code (tflovg.f in this case)
xlf -O3 -qmaxmem=-1 -NQ20000 -o tflovg tflovg.f

For debugging use:  xlf -g -o tflovg tflovg.f

and then type xde when sitting at an rs6000 terminal.

And for quick edits: xlf -o tflovg tflovg.f

which does not optimize the code. This will allow the code to compile quicker, but the executable will run much slower.

3. Check the grid input file (grid.dat)

Let's say you have another grid file named, 'grid1.dat'. You want to use grid1.dat in your simulation. Then you would copy grid1.dat to grid.dat by typing the following command.

    cp grid1.dat  grid.dat

Check the input file (room.dat)
Let's say you have another input file named 'wfu15.dat' which is a turbulent Wall Function file with a Uniform grid and an airflow of 15 ach. You want to use wfu15.dat as the input file to tflovg.f. Then issue the following command:

    cp  wfu15.dat  room.dat
4. Run the program:

Make sure tfexe has execute permissions.
(If not, or if you aren't sure, then type:)

```
chmod 700 tfexe
```

Then type the following line to start the code:

```
nice nohup tfexe &
```

This will run the code even if you log off.
The nice command allows runs your process at a lower priority.
Output that would normally goto the screen will be written to a nohup.out file.

To check the processes on a certain machine type:

```
ps -ef
```

To kill a process of yours that is running:
First read the PID # of the process (given from the above ps -ef command.) and then type:
9.0 Summary of Equations

This section presents finite difference equations found in the TFLOVG.F code.
9.1 Finite Difference Equations for U Velocity Components

\[
U_{i,j,k}^{\text{new}} = U_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i+1,j,k}}{\Delta x_i} + \beta \left( Q_{i,j,k} - T_{i,j,k} \right) g_x + \text{VISX} - \text{FUX} - \text{FUY} - \text{FUZ} \right]
\]

The viscous flux term in expanded finite difference form is given below.

\[
\text{VISX} = \nu \left[ \frac{\left( U_{i+1,j,k} - U_{i,j,k} \right) - U_{i,j,k} - U_{i-1,j,k}}{\Delta x_i} \right] + \frac{\left( U_{i,j+1,k} - U_{i,j,k} \right) - U_{i,j,k} - U_{i,j-1,k}}{\Delta y_j} + \frac{\left( U_{i,j,k+1} - U_{i,j,k} \right) - U_{i,j,k} - U_{i,j,k-1}}{\Delta z_k} \]

The convective flux terms are as follows.

\[
\text{FUX} = \frac{1}{4} \left\{ \left( U_{i,j,k} + U_{i+1,j,k} \right)^2 + \overline{\alpha} \left| U_{i,j,k} + U_{i+1,j,k} \right| \left( U_{i,j,k} - U_{i+1,j,k} \right) / \Delta x_i \right. \\
- \left\{ \left( U_{i,j,k} + U_{i,j,k} \right)^2 + \overline{\alpha} \left| U_{i,j,k} + U_{i,j,k} \right| \left( U_{i,j,k} - U_{i,j,k} \right) / \Delta x_i \right\}
\]

\[
\text{FUY} = \frac{1}{4} \left\{ \left( V_{i,j,k} + V_{i,j+1,k} \right)^2 + \overline{\alpha} \left| V_{i,j,k} + V_{i,j+1,k} \right| \left( V_{i,j,k} - V_{i,j+1,k} \right) / \Delta y_j \right. \\
- \left\{ \left( V_{i,j,k} + V_{i,j,k} \right)^2 + \overline{\alpha} \left| V_{i,j,k} + V_{i,j,k} \right| \left( V_{i,j,k} - V_{i,j,k} \right) / \Delta y_j \right\}
\]

\[
\text{FUZ} = \frac{1}{4} \left\{ \left( W_{i,j,k} + W_{i,j,k+1} \right)^2 + \overline{\alpha} \left| W_{i,j,k} + W_{i,j,k} \right| \left( W_{i,j,k} - W_{i,j,k+1} \right) / \Delta z_k \right. \\
- \left\{ \left( W_{i,j,k} + W_{i,j,k} \right)^2 + \overline{\alpha} \left| W_{i,j,k} + W_{i,j+1,k} \right| \left( W_{i,j,k} - W_{i,j,k} \right) / \Delta z_k \right\}
\]
9.2 Finite Difference Equations for V Velocity Components

\[ V_{i,j,k}^{\text{new}} = V_{i,j,k}^{\text{old}} + \Delta t \left\{ \frac{P_{i,j,k} - P_{i,j+1,k}}{\Delta y_j} + \beta \left( g_y - \nabla T_{i,j,k} \right) + V_{i,j,k} - V_{i,j-1,k} \right\} \]

The viscous flux term in expanded finite difference form is given below.

\[ V_{i,j,k}^{\text{vis}} = \nabla \left[ \left( \frac{V_{i+1,j,k} - V_{i,j,k}}{\Delta x_i} - \frac{V_{i,j,k} - V_{i-1,j,k}}{\Delta x_i} \right) / \Delta x_i \right] + \left( \frac{V_{i,j+1,k} - V_{i,j,k}}{\Delta y_{j+1}} - \frac{V_{i,j,k} - V_{i,j-1,k}}{\Delta y_j} \right) / \Delta y_j + \left( \frac{V_{i,j,k+1} - V_{i,j,k}}{\Delta z_k} - \frac{V_{i,j,k} - V_{i,j,k-1}}{\Delta z_{k-1}} \right) / \Delta z_k \]

The convective flux terms are as follows.

\[ F_{VX} = \frac{1}{4} \left\{ (U_{i+1,j,k} + U_{i,j+1,k}) (V_{i,j+1,k} + V_{i+1,j,k}) + |U_{i,j+k} + U_{i,j+1,k}| (V_{i,j+k} - V_{i+1,j,k}) \right\} / \Delta x_i \]

\[ -[(U_{i,j,k} + U_{i,j+1,k}) (V_{i,j,k} + V_{i,j+1,k}) + |U_{i,j,k} + U_{i,j+1,k}| (V_{i,j,k} - V_{i+1,j,k})] / \Delta x_i \}

\[ F_{VY} = \frac{1}{4} \left\{ (V_{i,j+1,k} + V_{i,j+1,k})^2 + |V_{i,j+k} + V_{i,j+1,k}| (V_{i,j+k} - V_{i,j+1,k}) \right\} / \Delta y_{j+1} \]

\[ -[(V_{i,j,k} + V_{i,j,k})^2 + |V_{i,j+1,k} + V_{i,j,k}| (V_{i,j+1,k} - V_{i,j,k})] / \Delta y_j \}

\[ F_{VZ} = \frac{1}{4} \left\{ (W_{i,j,k} + W_{i,j+1,k}) (V_{i,j,k} + V_{i,j,k+1}) + |W_{i,j,k} + W_{i,j+1,k}| (V_{i,j,k} - V_{i,j,k+1}) \right\} / \Delta z_k \]

\[ -[(W_{i,j,k+1} + W_{i,j+1,k}) (V_{i,j,k} + V_{i,j,k+1}) + |W_{i,j+1,k} + W_{i,j+1,k}| (V_{i,j,k} - V_{i,j,k+1})] / \Delta z_{k-1} \}

9.3 Finite Difference Equations for W Velocity Components
\[ W_{i,j,k}^{\text{new}} = W_{i,j,k}^{\text{old}} + \Delta t \frac{P_{i,j,k} - P_{i,j,k+1}}{\Delta z_k} + \beta \left( QT_0 - T_{i,j,k} \right) g_z + \text{VISZ} - \text{FWX} - \text{FY} - \text{FWZ} \]

The viscous flux term in expanded finite difference form is given below.

\[
\text{VISZ} = \nu \left[ \frac{\left( W_{i+1,j,k} - W_{i,j,k} \right)}{\Delta x_i} - \frac{W_{i,j,k} - W_{i-1,j,k}}{\Delta x_{i-1}} \right] / \Delta x_c_i \\
+ \left( \frac{W_{i,j+1,k} - W_{i,j,k}}{\Delta y_j} - \frac{W_{i,j,k} - W_{i,j-1,k}}{\Delta y_{j-1}} \right) / \Delta y_c_j \\
+ \left( \frac{W_{i,j,k+1} - W_{i,j,k}}{\Delta z_{c k+1}} - \frac{W_{i,j,k} - W_{i,j,k-1}}{\Delta z_{c k}} \right) / \Delta z_c_k
\]

The convective flux terms are as follows.

\[
\text{FWX} = \frac{1}{4} \left\{ \left( U_{i,j,k} + U_{i+1,j,k} \right) * \left( W_{i,j,k} + W_{i+1,j,k} \right) + \overline{\alpha} \left| U_{i,j,k} + U_{i+1,j,k} \right| \left( W_{i,j,k} - W_{i+1,j,k} \right) \right\} / \Delta x_i \\
- \left\{ \left( U_{i+1,j,k} + U_{i-1,j,k} \right) * \left( W_{i-1,j,k} + W_{i,j,k} \right) + \overline{\alpha} \left| U_{i-1,j,k} + U_{i-1,j,k} \right| \left( W_{i-1,j,k} - W_{i,j,k} \right) \right\} / \Delta x_{i-1}
\]

\[
\text{FWY} = \frac{1}{4} \left\{ \left( W_{i,j,k} + W_{i,j+1,k} \right) * \left( V_{i,j,k} + V_{i,j+1,k} \right) + \overline{\alpha} \left| V_{i,j,k} + V_{i,j+1,k} \right| \left( W_{i,j,k} - W_{i,j+1,k} \right) \right\} / \Delta y_j \\
- \left\{ \left( V_{i,j+1,k} + V_{i,j-1,k} \right) * \left( W_{i,j-1,k} + W_{i,j,k} \right) + \overline{\alpha} \left| V_{i,j-1,k} + V_{i,j-1,k} \right| \left( W_{i,j-1,k} - W_{i,j,k} \right) \right\} / \Delta y_{j-1}
\]

\[
\text{FWZ} = \frac{1}{4} \left\{ \left( W_{i,j,k} + W_{i,j,k+1} \right)^2 + \overline{\alpha} \left| W_{i,j,k} + W_{i,j,k+1} \right| \left( W_{i,j,k} - W_{i,j,k+1} \right) \right\} / \Delta z_{c k+1} \\
- \left\{ \left( W_{i,j,k+1} + W_{i,j,k} \right)^2 + \overline{\alpha} \left| W_{i,j,k+1} + W_{i,j,k} \right| \left( W_{i,j,k+1} - W_{i,j,k} \right) \right\} / \Delta z_{c k}
\]
9.4 Finite Difference Turbulent Velocity Equations

\[\begin{align*}
U_{i,j,k}^{\text{new}} &= U_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i,j,k+1}}{\Delta x_i} + \beta \left( Q T_0 - T_{i,j,k} \right) \right] g_x + \text{GAMX} - \text{FUX} - \text{FUY} - \text{FUZ} \right] \\
GAMX &= \Gamma_x = \nu_t \left[ 2 \frac{\partial^2 U}{\partial x^2} + 2 \frac{\partial^2 U}{\partial y^2} + 2 \frac{\partial^2 U}{\partial z^2} + \frac{\partial^2 V}{\partial x \partial y} + \frac{\partial^2 W}{\partial x \partial z} \right] + \frac{\partial}{\partial x} \left[ \frac{\partial U}{\partial x} \right] \\
&\quad + \frac{\partial}{\partial y} \left[ \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right] + \frac{\partial}{\partial z} \left[ \frac{\partial U}{\partial z} + \frac{\partial W}{\partial y} \right] - \frac{2}{3} \frac{\partial k}{\partial x} \\
V_{i,j,k}^{\text{new}} &= V_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i,j,k+1}}{\Delta y_j} + \beta \left( Q T_0 - T_{i,j,k} \right) \right] g_y + \text{GAMY} - \text{FVX} - \text{FVY} - \text{FVZ} \right] \\
GAMY &= \Gamma_y = \nu_t \left[ 2 \frac{\partial^2 V}{\partial y^2} + 2 \frac{\partial^2 V}{\partial x^2} + 2 \frac{\partial^2 V}{\partial z^2} + \frac{\partial^2 U}{\partial x \partial y} + \frac{\partial^2 W}{\partial x \partial z} \right] + \frac{\partial}{\partial y} \left[ \frac{\partial V}{\partial y} \right] \\
&\quad + \frac{\partial}{\partial x} \left[ \frac{\partial V}{\partial x} + \frac{\partial U}{\partial y} \right] + \frac{\partial}{\partial z} \left[ \frac{\partial V}{\partial z} + \frac{\partial W}{\partial y} \right] - \frac{2}{3} \frac{\partial k}{\partial y} \\
W_{i,j,k}^{\text{new}} &= W_{i,j,k}^{\text{old}} + \Delta t \left[ \frac{P_{i,j,k} - P_{i,j,k+1}}{\Delta z_k} + \beta \left( Q T_0 - T_{i,j,k} \right) \right] g_z + \text{GAMZ} - \text{FWX} - \text{FWY} - \text{FWZ} \right] \\
GAMZ &= \Gamma_z = \nu_t \left[ 2 \frac{\partial^2 W}{\partial z^2} + 2 \frac{\partial^2 W}{\partial x^2} + 2 \frac{\partial^2 W}{\partial z^2} + \frac{\partial^2 U}{\partial x \partial z} + \frac{\partial^2 V}{\partial z \partial x} \right] + \frac{\partial}{\partial z} \left[ \frac{\partial W}{\partial z} \right]
\end{align*}\]
\[ + \frac{\partial v_i}{\partial x} \left[ \frac{\partial W}{\partial x} + \frac{\partial U}{\partial z} \right] + \frac{\partial v_i}{\partial y} \left[ \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} \right] - \frac{2}{3} \frac{\partial k}{\partial z} \]

9.5 Finite Difference Temperature Equations

\[ T_{i,j,k}^{\text{new}} = T_{i,j,k}^{\text{old}} + \Delta t \left[ \text{VIST} - \text{FTX} - \text{FTY} - \text{FTZ} \right] \]

The viscous flux term in expanded finite difference form is given below.

\[ \text{VIST} = \alpha \left[ \left( \frac{T_{i+1,j,k} - T_{i,j,k}}{\Delta x} - \frac{T_{i,j,k} - T_{i-1,j,k}}{\Delta x} \right) / \Delta x \right] + \left( \frac{T_{i,j+1,k} - T_{i,j,k}}{\Delta y} - \frac{T_{i,j,k} - T_{i,j-1,k}}{\Delta y} \right) / \Delta y \] + \left( \frac{T_{i,j,k+1} - T_{i,j,k}}{\Delta z} - \frac{T_{i,j,k} - T_{i,j,k-1}}{\Delta z} \right) / \Delta z \]

The convective flux terms are:

\[ \text{FTX} = \frac{1}{2} \left\{ \left| U_{i,j,k} \right| (T_{i,j,k} + T_{i+1,j,k}) + \bar{\alpha} \left| U_{i,j,k} \right| (T_{i,j,k} + T_{i+1,j,k}) \right\} / \Delta x_i \]

\[ + \left\{ (U_{i-1,j,k} T_{i,j,k+1} + T_{i,j,k}) + \bar{\alpha} \left| U_{i-1,j,k} \right| (T_{i,j,k} + T_{i,j,k-1}) \right\} / \Delta x_i \]

\[ \text{FTY} = \frac{1}{2} \left\{ \left| V_{i,j,k} \right| (T_{i,j,k} + T_{i,j+1,k}) + \bar{\alpha} \left| V_{i,j,k} \right| (T_{i,j,k} + T_{i,j+1,k}) \right\} / \Delta y_j \]

\[ + \left\{ (V_{i,j-1,k} T_{i,j,k+1} + T_{i,j,k}) + \bar{\alpha} \left| V_{i,j-1,k} \right| (T_{i,j,k} + T_{i,j,k-1}) \right\} / \Delta y_j \]

\[ \text{FTZ} = \frac{1}{2} \left\{ \left| W_{i,j,k} \right| (T_{i,j,k} + T_{i,j,k+1}) + \bar{\alpha} \left| W_{i,j,k} \right| (T_{i,j,k} + T_{i,j,k+1}) \right\} / \Delta z_k \]

\[ + \left\{ (W_{i,j,k-1} T_{i,j,k+1} + T_{i,j,k}) + \bar{\alpha} \left| W_{i,j,k-1} \right| (T_{i,j,k} + T_{i,j,k-1}) \right\} / \Delta z_k \]
9.6 Finite Difference Equations

\[ k_{i,j,k}^{new} = k_{i,j,k}^{old} + \Delta t \left[ \Theta \text{ETAK} + \nu_t \cdot \Pi - F_K X - F_K Y - F_K Z - \varepsilon \right] \]

\[
F_K X = \frac{1}{2} \left\{ \left[ U_{i,j,k}(k_{i,j,k} + k_{i+1,j,k}) + \bar{\alpha} \ |U_{i,j,k}|*(k_{i,j,k} - k_{i+1,j,k}) \right] / \Delta x_i \\
- \left[ U_{i+1,j,k}(k_{i+1,j,k} + k_{i,j,k}) + \bar{\alpha} \ |U_{i+1,j,k}|*(k_{i+1,j,k} - k_{i,j,k}) \right] / \Delta x_{i+1} \right\}
\]

\[
F_K Y = \frac{1}{2} \left\{ \left[ V_{i,j,k}(k_{i,j,k} + k_{i,j+1,k}) + \bar{\alpha} \ |V_{i,j,k}|*(k_{i,j,k} - k_{i,j+1,k}) \right] / \Delta y_j \\
- \left[ V_{i,j+1,k}(k_{i,j+1,k} + k_{i,j,k}) + \bar{\alpha} \ |V_{i,j+1,k}|*(k_{i,j+1,k} - k_{i,j,k}) \right] / \Delta y_{j+1} \right\}
\]

\[
F_K Z = \frac{1}{2} \left\{ \left[ W_{i,j,k}(k_{i,j,k} + k_{i,j,k+1}) + \bar{\alpha} \ |W_{i,j,k}|*(k_{i,j,k} - k_{i,j,k+1}) \right] / \Delta z_k \\
- \left[ W_{i,j,k+1}(k_{i,j,k+1} + k_{i,j,k}) + \bar{\alpha} \ |W_{i,j,k+1}|*(k_{i,j,k+1} - k_{i,j,k}) \right] / \Delta z_{k+1} \right\}
\]
9.7 ε Finite Difference Equations

\[
\epsilon_{ij,k}^{\text{new}} = \epsilon_{ij,k}^{\text{old}} + \Delta t \left[ \text{THETA} E + \frac{\nu_1 C_1 F_1 \epsilon}{k} \pi - C_2 F_2 \frac{\epsilon^2}{k} - \text{FEX} - \text{FEY} - \text{FEZ} + \epsilon_0 \right]
\]

\[
\text{FEX} = \frac{1}{2} \left\{ \left[ U_{i,j,k}(\epsilon_{ij,k} + \epsilon_{ij+1,k}) + \overline{\alpha} \left| U_{i,j,k} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij+1,k}) \right] / \Delta x_i \right. \\
\left. - \left[ U_{i+1,j,k}(\epsilon_{ij,k} + \epsilon_{ij,k}) + \overline{\alpha} \left| U_{i+1,j,k} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij,k}) \right] / \Delta x_{i+1} \right\}
\]

\[
\text{FEY} = \frac{1}{2} \left\{ \left[ V_{i,j,k}(\epsilon_{ij,k} + \epsilon_{ij+1,k}) + \overline{\alpha} \left| V_{i,j,k} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij+1,k}) \right] / \Delta y_j \right. \\
\left. - \left[ V_{i,j+1,k}(\epsilon_{ij,k} + \epsilon_{ij,k}) + \overline{\alpha} \left| V_{i,j+1,k} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij,k}) \right] / \Delta y_{j+1} \right\}
\]

\[
\text{FEZ} = \frac{1}{2} \left\{ \left[ W_{i,j,k}(\epsilon_{ij,k} + \epsilon_{ij,k+1}) + \overline{\alpha} \left| W_{i,j,k} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij,k+1}) \right] / \Delta z_k \right. \\
\left. - \left[ W_{i,j,k-1}(\epsilon_{ij,k} + \epsilon_{ij,k}) + \overline{\alpha} \left| W_{i,j,k-1} \right|^2 (\epsilon_{ij,k} - \epsilon_{ij,k}) \right] / \Delta z_{k-1} \right\}
\]
9.8 Finite Difference Equations for Theta Function and Pi

$$\text{THETAK} = \Theta(k) = \frac{v_t}{\sigma_k} \left( \frac{\partial^2 k}{\partial x^2} + \frac{\partial^2 k}{\partial y^2} + \frac{\partial^2 k}{\partial z^2} \right) + \frac{1}{\sigma_k} \left( \frac{\partial v_t \partial k}{\partial x \partial x} + \frac{\partial v_t \partial k}{\partial y \partial y} + \frac{\partial v_t \partial k}{\partial z \partial z} \right)$$

$$\text{THETAE} = \Theta(\varepsilon) = \frac{v_t}{\sigma_\varepsilon} \left( \frac{\partial^2 \varepsilon}{\partial x^2} + \frac{\partial^2 \varepsilon}{\partial y^2} + \frac{\partial^2 \varepsilon}{\partial z^2} \right) + \frac{1}{\sigma_\varepsilon} \left( \frac{\partial v_t \partial \varepsilon}{\partial x \partial x} + \frac{\partial v_t \partial \varepsilon}{\partial y \partial y} + \frac{\partial v_t \partial \varepsilon}{\partial z \partial z} \right)$$

$$\text{PI} = \Pi = 2 \left( \frac{\partial^2 U}{\partial x^2} + \frac{\partial U}{\partial y} \frac{\partial U}{\partial y} + \frac{\partial U}{\partial z} \frac{\partial U}{\partial z} + \frac{\partial W}{\partial x} \frac{\partial W}{\partial x} \right) + 2 \left( \frac{\partial^2 V}{\partial y^2} + \frac{\partial V}{\partial x} \frac{\partial V}{\partial x} + \frac{\partial V}{\partial z} \frac{\partial V}{\partial z} \right) + 2 \left( \frac{\partial^2 W}{\partial z^2} + \frac{\partial W}{\partial x} \frac{\partial W}{\partial x} + \frac{\partial W}{\partial y} \frac{\partial W}{\partial y} + \frac{\partial V}{\partial z} \frac{\partial V}{\partial z} \right)$$
VITA

Michael R. Ramey

Candidate for the Degree of

Master of Science

Thesis: APPLICATION OF COMPUTATIONAL FLUID DYNAMICS TO INDOOR ROOM AIRFLOW

Major Field: Mechanical Engineering

Biographical:


Education: Graduated from Granite City Senior High School, Granite City, Illinois, in June, 1987; received Bachelor of Science degree in Mechanical Engineering from Oklahoma Christian University of Science and Arts, Oklahoma City, Oklahoma, in December, 1991; completed requirements for the Master of Science degree at Oklahoma State University in December, 1994.